Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

Identificação Neuro-Difusa

Aspectos de Interpretabilidade

Dissertação submetida para obtenção do grau de Mestre em
Engenharia Informática

Rui Pedro Pinto de Carvalho e Paiva
Licenciado em Engenharia Informática

Coimbra, 1999

Dissertação realizada sob a orientação do

Professor Doutor António Dourado Pereira Correia

Professor Associado (com agregação) do
Departamento de Engenharia Informática da
Faculdade de Ciências e Tecnologia da
Universidade de Coimbra

a Te… ad Te…

AGRADECIMENTOS

Gostaria, em primeiro lugar, de agradecer ao Professor Doutor António Dourado Pereira Correia, na qualidade de orientador científico, pela oportunidade concedida para a realização do trabalho presente, assim como pela amizade, motivação e disponibilidade sempre demonstradas.

Ao Centro de Informática e Sistemas da Universidade de Coimbra (CISUC), nomeadamente ao grupo de Automação e Controlo, agradeço os meios logísticos e informáticos facultados para a elaboração desta dissertação.

A todos os colegas do CISUC, e em particular aos do grupo de Automação e Controlo, endereço os meus agradecimentos pelas discussões estimulantes que muito contribuíram para a minha aprendizagem, assim como pelo seu apoio, estímulo e bom ambiente proporcionado. Gostaria de agradecer de forma muito especial ao Eng. Jorge Henriques, assim como ao Eng. Paulo de Carvalho, ao Eng. Carlos Vaz e ao Eng. Carlos Pereira pelas suas observações e sugestões.

Ao Doutor Engenheiro Belmiro Duarte da Companhia de Celulose do Caima, agradeço a facultação de dados fabris, bem como a sua disponibilidade e sugestões.

Finalmente, gostaria de agradecer, de forma especial, à minha família e aos meus amigos por todo o seu apoio.

Rui Pedro Pinto de Carvalho e Paiva

RESUMO

No momento presente da História, é lícito afirmar-se que a humanidade se encontra em plena era da informação. De facto, em qualquer aspecto da sociedade, desde as actividades de lazer até aos mais complexos sistemas de produção, é notória a presença e influência das tecnologias de informação. Assim, assiste-se presentemente a um forte impulso na investigação, desenvolvimento e aplicação de metodologias de computação aos processos industriais de produção. Na verdade, o elevado grau de complexidade que os caracteriza, acompanhado de uma necessidade crescente de desempenho como forma de dar resposta às leis de mercado, exige a utilização de estratégias cada vez mais sofisticadas. Uma das áreas que tem merecido uma atenção particular tem sido a *soft computing*, a qual engloba metodologias tais como a lógica difusa, redes neuronais e algoritmos genéticos, de forma simples ou combinada, constituindo um dos pilares dos sistemas de informação inteligentes. Neste contexto, a dissertação apresentada pretende contribuir para a compreensão do potencial associado às técnicas neuro-difusas como mecanismo de identificação de sistemas.

Assim, numa primeira fase introdutória, são apresentados e discutidos os princípios básicos da lógica difusa, sistemas difusos e redes neuronais, enquadrados na temática da identificação de sistemas.

Neste trabalho, são estudadas diversas estruturas difusas, nomeadamente os sistemas de Takagi-Sugeno de ordem 0 e 1, bem como sistemas linguísticos. Neste sentido, são abordados dois aspectos essenciais da identificação difusa: a aprendizagem da estrutura e a aprendizagem de parâmetros. No primeiro ponto é prestada especial atenção à utilização de técnicas de agrupamento de classes, destacando-se, de entre estas, o algoritmo de agrupamento subtractivo. Ainda em relação à aprendizagem da estrutura, é abordada a questão da selecção de entradas relevantes. Relativamente à aprendizagem de parâmetros, a mesma é conduzida com recurso ao treino de uma rede neuronal difusa pelo algoritmo de retropropagação do erro, sendo, em algumas situações, utilizados esquemas híbridos baseados em optimização linear e não linear. Ainda em relação a este ponto, o problema da aprendizagem incremental de parâmetros é abordado, ainda que de forma superficial.

Um aspecto relevante no contexto da implementação de modelos difusos prende-se com a exploração do potencial que lhes é inerente em termos de transparência do modelo final. Assim sendo, são apresentados alguns estudos originais em termos de estratégias que visem a manutenção da interpretabilidade dos modelos durante a aprendizagem de parâmetros, as quais se baseiam em medidas de similaridade e na aprendizagem restringida de parâmetros.

As metodologias referidas foram aplicadas a alguns casos de estudo, e.g., a série caótica Mackey-Glass e a fornalha de Gás Box-Jenkins, os quais confirmaram as suas capacidades de modelização, assim com a adequação das técnicas difusas na implementação de modelos interpretáveis. As mesmas técnicas foram aplicadas a um sistema industrial, nomeadamente uma planta de branqueamento de pasta da papel. Contudo, os resultados obtidos não foram totalmente satisfatórios, em virtude da deficiente qualidade dos dados de identificação.

Na realização do estudo efectuado, os algoritmos descritos neste trabalho foram implementados na linguagem de programação C++, preparando-se neste momento a sua integração numa interface gráfica por forma a que a ferramenta computacional desenvolvida possa constituir um auxílio no estudo dos problemas analisados nesta dissertação, tanto com funções didácticas como de investigação científica.

Palavras-chave: identificação de sistemas dinâmicos, lógica difusa, sistemas difusos, modelização difusa, redes neuronais, treino de redes neuronais, redes neuro-difusas, agrupamento de classes, agrupamento subtractivo, selecção de entradas, interpretabilidade, similaridade difusa, aprendizagem restringida, aprendizagem adaptativa.

ABSTRACT

Nowadays, humankind is in the era of information. In fact, in most of the aspects of today's society, from leisure activities to complex production systems, the presence and influence of the technologies of information is clear. Thus, there is presently a strong impulse towards the research, development and application of computing methodologies in industrial production systems. Actually, the high degree of complexity that characterizes those systems, as well as the increasing necessities in terms of performance in order to cope with the rules of the market, demands strategies more and more sophisticated. One of the areas that has deserved a particular attention is *soft computing*, which includes techniques like fuzzy logic, neural networks and genetic algorithms, in a simple or combined fashion, and constitutes itself as the basis for intelligent information systems. In this context, the study carried out aims to contribute to the comprehension of the potential associated to neuro-fuzzy techniques as a mechanism for system identification.

In a first introductory phase, the grounds of fuzzy logic, fuzzy systems and neural networks are presented and discussed, integrated in the problem of system identification.

In this work, several fuzzy structures are analyzed, namely Takagi-Sugeno (zero and first order) systems and linguistic systems. Two major concerns of fuzzy identification are studied: structure learning and parameter learning. Referring to the first item, clustering techniques receive a deeper attention, especially subtractive clustering. Still in the same point, the questions related to relevant input selection are addressed. As for parameter learning, this task is carried out after the determination of a structure, based on the training of a fuzzy neural network via error backpropagation. In some situations, hybrid learning schemes are also utilized, which result from the combination of both linear and nonlinear optimization algorithms. In the point of parameter learning, the problem of online learning is also addressed, though superficially.

A relevant matter in the context of fuzzy identification relates to the use of their potential in terms of model transparency. In this way, some original studies are performed, regarding the construction of interpretable fuzzy models, which are based on similarity measures and restricted parameter learning.

The subjects mentioned above were applied to same case studies, e.g., the Mackey-Glass chaotic time series and the Box-Jenkins gas furnace, which confirmed their modeling capabilities, as well as the adequacy of fuzzy techniques for the building up of interpretable models. The same techniques were applied to an industrial plant, namely a pulp bleaching plant. However, the results obtained so far are not totally satisfactory, due to bad data quality, which resulted from a deficient sampling time, as well as insufficient excitation of some input variables.

The techniques studied are implemented in software, and constitute the core of an application, which is being developed to assist the comprehension and analysis of the main issues regarding fuzzy identification. The resulting software tool will be used both with research and pedagogical goals.

Keywords: identification of dynamical systems, fuzzy logic, fuzzy systems, fuzzy modeling, neural networks, training of neural networks, neuro-fuzzy networks, clustering, subtractive clustering, input selection, interpretability, fuzzy similarity, restricted learning, adaptive learning.

ÍNDICE GERAL

LISTA DE FIGURAS

LISTA DE TABELAS

SIMBOLOGIA

Abreviaturas

ADALINE	ADAptive LInear Neuron
AIC	Akaike's Information Criterion
ANFIS	Adaptive Network-based Fuzzy Inference System
ANN	Artificial Neural Network
ARMAX	Auto Regressive Moving Average with eXogenous inputs
ARX	Auto Regressive with eXogenous inputs
AS	Agrupamento Subtractivo
BIBO	Bounded Input Bounded Output
CC	Consequentes Constantes
CD	Consequentes Difusos
CO1	Consequentes de Ordem 1
FARX	Fuzzy ARX
FCM	Fuzzy C-Means
IA	Inteligência Artificial
LS	Least Squares
LSE	Least Square Estimator
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
NARX	Nonlinear ARX
NFCN	Neural Fuzzy Control Network
RBF	Radial Basis Function
RLS	Recursive Least Squares
RMSE	Root Mean Square Error
SISO	Single Input Single Output

Símbolos

Identificação de Sistemas

$u(t)$	entrada de um sistema, no instante t
$v(t)$	perturbação em um sistema, no instante t
$y(t)$	saída de um sistema, no instante t
S	sistema
E	condição experimental
M	estrutura de modelos
M^*	conjunto de modelos candidatos
I	método de identificação
V	critério de validação
N	número de amostras de dados
Z^N	conjunto de N amostras de dados
q	vector de parâmetros de uma estrutura paramétrica
g	função representativa de um sistema real
q_o	parâmetros reais de um sistema
$\hat{q}$	vector de parâmetros obtidos por um modelo paramétrico
$\hat{v}(t)$	valor do modelo de perturbações no instante t
$\hat{y}(t)$	saída do modelo no instante t
$\hat{g}$	aproximação da função real g efectuada pelo modelo
$e(t)$	erro de predição (ou modelização) no instante t
J^N	critério de erro calculado com base em N amostras de dados
na	número de regressões da saída, num sistema SISO
nb	número de regressões da entrada, num sistema SISO
d	atraso da entrada, num sistema SISO
$j(t)$	matriz de regressões no instante t
$e(t)$	ruído branco
nc	número de regressões da variável $e(t)$ num ruído colorido

Sistemas Difusos

x	variável numérica
X	variável linguística
X	universo de discurso da variável linguística X
LX	termo linguístico associado à variável X

$L\tilde{X}$	conjunto difuso associado ao termo linguístico *LX*
LX_i	*i*-ésimo termo linguístico da variável *X*
$LX^{(k)}$	termo linguístico da variável *X* na regra *k*
A	conjunto clássico
$\tilde{A}$	conjunto difuso
$m_{\tilde{A}}(x)$	função de pertença associada ao conjunto difuso $\tilde{A}$
c	centro de uma função de pertença Gaussiana
s	desvio padrão de uma função de pertença Gaussiana
c_L, c_R	centros esquerdo e direito de uma função Gaussiana generalizada
s_L, s_R	desvios padrões esquerdo e direito de uma função Gaussiana generalizada
$\hat{*}$	norma-T
$\check{*}$	norma-S
$c(\cdot)$	norma-c
$s(\tilde{A}, \tilde{B})$	similaridade entre dois conjuntos difusos $\tilde{A}$ e $\tilde{B}$
S_1	medida de similaridade S_1
x^*	valor numérico
$\tilde{X}^*$	conjunto difuso resultante da fuzificação de x^*
g	número de regras de um sistema difuso (ou número de neurónios escondidos numa rede RBF)

Redes Neuronais

$a_i^{(p)}$	activação do neurónio *i*, relativamente ao padrão *p*
F_i	função de activação do neurónio *i*
w_{ij}	peso da ligação entre o neurónio *i* (camada *k*) e o neurónio *j* (camada *k*+1)
b_i	termo de polarização do neurónio *i*
$x_i^{(p)}$	*i*-ésima entrada da rede, relativamente ao padrão *p*
$y_i^{(p)}$	*i*-ésima saída da rede, relativamente ao padrão *p*
$y_i^{(p)}$	*i*-ésima saída desejada para a rede, relativamente ao padrão *p*
$E^{(p)}$	erro da rede, relativamente ao padrão *p*
E	erro total
$d_i^{(p)}$	sinal delta correspondente ao neurónio *i*, relativamente ao padrão *p*
g	velocidade de aprendizagem
m	número de entradas da rede
n	número de saídas da rede
X	matriz de entradas da rede
W	matriz de pesos

Y	matriz de saídas desejadas
Y	matriz de saídas reais
P	matriz de co-variância (aproximada)

Identificação Neuro-Difusa

$\|T(X_j)\|$	número de termos linguísticos da variável X_j
n_{fpi}	número total de funções de pertença associadas às variáveis de entrada
n_{fpo}	número total de funções de pertença associadas às variáveis de saída
c_{ij}	centro de uma função de pertença de base radial
dr	decay rate
s	parâmetro de sobreposição entre funções Gaussianas
P	função associada a um dado centro (ou candidato)
a	área de influência de cada centro
b	área de influência na redução do potencial de cada centro
r_a	*radii*: raio da vizinhança de cada ponto
r_b	raio da vizinhança de cada centro com reduções sensíveis no seu potencial
e^{up}	limiar de aceitação de centro
e^{down}	limiar de rejeição de centro
$a_i^{(p2)}$	activação do neurónio i (input) da camada 2, relativamente ao padrão de treino p
$a_r^{(p3)}$	activação do neurónio r (regra) da camada 3, …
$a_s^{(p4)}$	activação do neurónio s (norma-S) da camada 4, …
$a_o^{(p5)}$	activação do neurónio o (output) da camada 5, …, num sistema linguístico
$a_o^{(p4)}$	activação do neurónio o (output) da camada 4, …, num sistema Takagi-Sugeno
c_{ij} / s_{ij}	centro / desvio padrão da i-ésima Gaussiana associada à entrada j
c_{os} / s_{os}	centro / desvio padrão da s-ésima Gaussiana associada à saída o
$\Phi^{(p)}$	matriz de dados para optimização linear (coluna p, referente ao padrão p)
b_{orj}	parâmetros de consequentes de ordem 1, referentes à saída o, regra r e entrada j
B	matriz de parâmetros b_{orj}
l	factor de esquecimento
n_{int}	número de pontos no cálculo aproximado do integral
l	limiar de fusão de funções de pertença
m^{up}	factor de aumento da velocidade de aprendizagem
m^{down}	factor de redução da velocidade de aprendizagem
num_{inc}	número de épocas consecutivas com aumento de erro de treino
num_{red}	número de épocas consecutivas com redução de erro de treino
num_{osc}	número de épocas consecutivas com oscilação de erro de treino

Capítulo 1

INTRODUÇÃO

"Na medida em que as leis da matemática se referem à realidade, não são certas. E na medida em que são certas, não se referem à realidade."

Albert Einstein, "Gemotrie und Ergahrung", 1921

Ao longo do caminho percorrido pela humanidade, norteado pelo desejo de progresso e evolução, a ciência tem sempre procurado compreender o universo e os sistemas que o integram, prever os seus comportamentos e tentar, de algum modo, moldá-los segundo os interesses do ser humano, no sentido daquilo que se designa por uma melhor qualidade de vida. A realização destas tarefas baseia-se largamente na construção de modelos representativos dos sistemas a estudar, os quais se podem caracterizar segundo as mais diversas maneiras: sistemas físicos, biológicos, políticos, educativos, económicos, ou ainda sistemas puramente abstractos.

Já em 1921, Albert Einstein afirmara no seu "Geometrie und Ergahrung" que a modelização matemática analítica, baseada em estruturas como as equações diferenciais, equações de diferenças ou equações algébricas, apresenta algumas limitações em termos de capacidade de descrição de sistemas complexos. Neste contexto, assiste-se actualmente a um forte impulso no sentido da investigação e aplicação de metodologias de modelização mais sofisticadas. Uma das áreas que tem vindo a receber um interesse crescente é a da modelização neuro-difusa. Esse interesse deriva do potencial desta abordagem a nível de capacidade de representação. De facto, as metodologias neuro-difusas procuram conjugar a capacidade de aprendizagem de redes neuronais artificiais com a facilidade de interpretação do conhecimento armazenado, que caracteriza os sistemas difusos. Neste contexto, esta dissertação tem por objectivo demonstrar as potencialidades desta tecnologia na identificação de sistemas.

Assim, a primeira secção deste capítulo apresenta as motivações fundamentais da realização deste trabalho. Na segunda secção são descritos os objectivos propostos, bem como as abordagens seguidas para a sua consecução. Seguidamente, são descritas algumas das contribuições originais do presente trabalho. Finalmente, a organização do documento é apresentada na Secção 1.4.

1.1. Motivação e Enquadramento

A modelização matemática analítica aborda os aspectos de construção de modelos com base

nas leis da mecânica, da física, da química ou da termodinâmica, i.e., com base naquilo que se designa por *primeiros princípios*. Esta abordagem, sendo a mais tradicional, apresenta dificuldades a nível de sistemas de análise fenomenológica complexa ou com factores de incerteza associados. De facto, a capacidade de modelização, rigorosa e precisa, de sistemas com base nos primeiros princípios diminui com o aumento da complexidade. Este problema é sintetizado por Lofti Zadeh [Zadeh, 1973] como o *"princípio da incompatibilidade"*:

> *"À medida que a complexidade de um sistema aumenta, a nossa capacidade de descrever o seu comportamento de forma precisa e, além disso, significativa vai diminuindo até que seja atingido um limiar para além do qual a precisão e a relevância se tornam características quase mutuamente exclusivas."*

Paralelamente às dificuldades enunciadas, a indústria tem manifestado a procura de *autonomia* nos processos de produção. Para além desse desejo de autonomia, o sector produtivo debate-se com questões relativas à *complexidade crescente* dos processos de produção, assim como com *requisitos de desempenho* cada vez mais exigentes. Tais necessidades, nomeadamente a necessidade de autonomia, sugerem a utilização de métodos automáticos de aquisição de conhecimento e a sua incorporação nos sistemas de produção. Estes aspectos, juntamente com as dificuldades da modelização analítica, expressas sucintamente pelo princípio da incompatibilidade, sugerem a utilização de técnicas mais sofisticadas. Neste sentido, o momento presente caracteriza-se por uma forte investigação relativamente à aplicabilidade de técnicas da chamada *Inteligência Artificial* (IA) na modelização e controlo de sistemas.

As técnicas da IA, ou técnicas inteligentes, procuram dotar os sistemas onde sejam utilizadas de capacidades iminentemente humanas, nomeadamente aprendizagem, criatividade, abstracção ou adaptabilidade. Além deste aspecto, as limitações do ser humano procuram ser ultrapassadas, particularmente a nível de cansaço e subjectividade.

Neste sentido, as *redes neuronais artificiais* [Pham e Xing, 1995; Haykin, 1994; Hunt et al, 1992; Narendra e Parthasarathy, 1990] têm provado tratar-se de metodologias com boas capacidades de representação. Estas estruturas, inicialmente com o objectivo de emularem o funcionamento do cérebro humano, apresentam como atractivo essencial a sua capacidade de aprendizagem e adaptação. Deste modo, conseguem representar sistemas com dinâmica complexa. De facto, a aplicação de redes neuronais multicamada à identificação de sistemas constitui uma ferramenta de elevado potencial, dado que essas estruturas funcionam como aproximadores universais. No entanto, na identificação de sistemas, linear ou não linear, a informação armazenada não é, em geral, facilmente interpretável de forma qualitativa, i.e., subsiste o problema da falta de *transparência* do conhecimento representado[1]: os modelos obtidos são do tipo caixa-negra, além de que conhecimento prévio disponível não é facilmente incluído no modelo. Neste contexto, surge, então, a modelização difusa.

Neste aspecto, os sistemas difusos apresentam vantagens significativas, na medida em que a informação é representada de forma *transparente*, o *conhecimento a priori* eventualmente

[1] A análise efectuada, refere-se, essencialmente, às redes neuronais MLP (*Multi-Layer Perceptrons*). Existe uma classe de redes neuronais, as redes AMN (*Associative Memory Networks*), nas quais se incluem as redes RBF *(Radial Basis Function*), que podem ser interpretadas como sistemas de inferência difusos, tendo, assim, as vantagens dos sistemas difusos. Estas redes podem ser designadas por redes neuro-difusas, tal como será discutido posteriormente.

disponível é incluído no modelo de forma simples, além de permitirem representar eficientemente sistemas complexos (o ponto essencial do princípio da incompatibilidade). De facto, Castro [Castro, 1995], na sequência do trabalho de autores como Wang [Wang, 1992] e Buckley [Buckley, 1993], prova a propriedade da *aproximação universal* para um número significativo de classes de sistemas difusos, nomeadamente sistemas do tipo Takagi-Sugeno e sistemas linguísticos, estudados neste trabalho de dissertação (Secção 3.4). No entanto, a selecção de uma estrutura adequada, nomeadamente em termos de base de regras e funções de pertença associadas a cada variável não é efectuada de maneira trivial. Neste sentido, é importante utilizar dados experimentais na realização da tarefa de desenvolvimento de sistemas difusos. Este aspecto conduz-nos à modelização difusa baseada em dados, ou *identificação difusa*. Na selecção da estrutura e dos parâmetros de um sistema difuso, várias metodologias são utilizáveis. Uma técnica particularmente interessante baseia-se na implementação de um sistema difuso por meio de uma rede neuronal artificial, sendo a mesma designada, deste modo, por *abordagem neuro-difusa*.

As metodologias neuro-difusas procuram conjugar as vantagens das técnicas neuronais com as das técnicas difusas. Basicamente, as capacidades de uma são as limitações da outra: se as redes neuronais artificiais apresentam a vantagem da capacidade de aprendizagem, a informação nelas armazenada é, geralmente, opaca, dada a natureza quantitativa do conhecimento representado; quanto aos sistemas difusos, sendo a informação armazenada transparente, em virtude da sua natureza qualitativa, a aquisição dessa informação, porém, não é efectuada de forma trivial. O objectivo primordial deste trabalho de dissertação é, então, o estudo e aplicação de técnicas neuro-difusas na modelização de sistemas dinâmicos.

O objectivo citado enquadra-se numa área científica vasta, a qual engloba as metodologias centradas na lógica difusa, redes neuronais e algoritmos genéticos, baptizada por Zadeh com a designação de *soft computing* [Zadeh, 1994]. A combinação de duas ou mais metodologias da *soft computing* conduz aos sistemas híbridos inteligentes, dos quais as estruturas neuro-difusas constituem, possivelmente, a ferramenta mais explorada. A *soft computing* constitui, então, um dos pilares dos sistemas de informação inteligentes, dado possibilitar a obtenção de conhecimento para a tomada de decisão a partir de grandes quantidades de informação, eventualmente de natureza diversa.

Apesar dos muitos casos de sucesso de aplicações industriais da *soft computing*, as quais estimularam significativamente a investigação nesta área, várias críticas têm sido dirigidas por parte dos apologistas das chamadas técnicas clássicas. Fundamentalmente, são levantadas algumas questões quanto ao campo de aplicação das metodologias inteligentes. Por um lado, há quem sugira a aplicação das metodologias referidas como uma panaceia, a utilizar indiscriminadamente em todas as situações. Por outro lado, alguns investigadores da área designada por clássica defendem que todos os problemas que as técnicas inteligentes se propõem tratar são solucionados pelas técnicas clássicas, com a vantagem de, por se tratarem maioritariamente de técnicas lineares, a sua análise ser efectuada com critérios rigorosos e amadurecidos. Naturalmente, uma posição equilibrada parece a mais adequada: as duas filosofias complementam-se - para a classe de problemas para os quais as técnicas convencionais fornecem respostas satisfatórias não faz sentido utilizar técnicas inteligentes; no entanto, para o tipo de problemas enunciados anteriormente, as técnicas da AI, em virtude do seu potencial, afiguram-se mais adequadas. Este conflito entre a teoria clássica e as técnicas inteligentes é abordado por Zadeh de forma humorística, segundo aquilo que o autor designa por *"princípio do martelo"* [von Altrock, 1995]:

> *"Se alguém tiver um martelo na mão, e se isso for a sua única ferramenta, tudo começa a parecer um prego."*

No entanto, no sentido da afirmação definitiva da aplicação das técnicas inteligentes a situações do mundo real, onde segurança, previsibilidade e correcção são requisitos essenciais, é fundamental estabelecer resultados rigorosos e objectivos em termos de convergência e estabilidade dos algoritmos utilizados, o que ainda não acontece de maneira genérica.

1.2. Objectivos e Abordagens

Na obtenção de modelos difusos, a selecção de um conjunto de regras susceptíveis de descrever o sistema em questão, bem como a sintonização dos parâmetros das funções de pertença associadas a cada variável, constituem os pontos fundamentais de projecto. Essas duas tarefas designam-se, respectivamente, por *aprendizagem da estrutura* e *aprendizagem de parâmetros*. Estas tarefas são conduzidas, nesta dissertação, por meio de estruturas neuro-difusas, tal como foi referido.

Na abordagem linguística pura, um sistema a ser modelizado, por exemplo, um sistema de controlo do nível de líquido num tanque, é representado com base num conjunto de regras do tipo (1.1):

SE (*nível é baixo*) ENTÃO (*abertura da válvula é alta*). (1.1)

Uma das desvantagens desta abordagem deriva do elevado número de regras que, em geral, é necessário para descrever um sistema com um grau de precisão elevado. Como tal, Takagi e Sugeno [Takagi e Sugeno, 1985] propuseram um esquema no qual os consequentes das regras não são representados por termos linguísticos, mas sim por funções dos antecedentes, como se segue (1.2):

SE (*nível é baixo*) ENTÃO (*abertura da válvula = f*(*nível*)). (1.2)

A modelização de Takagi-Sugeno tem, então, a vantagem de permitir a descrição de um sistema com recurso a um menor número de regras - ou alternativamente com o mesmo número de regras mas maior precisão - do que aquele necessário na abordagem linguística. Esta razão, por si só, justifica o seu estudo neste trabalho. Porém, em termos de transparência do modelo final, a modelização linguística apresenta vantagens, tal como se pode depreender de (1.1) e (1.2).

Independentemente da abordagem utilizada, os aspectos relativos à aprendizagem da estrutura e dos parâmetros são mantidos.

1.2.1. Aprendizagem da Estrutura

Relativamente à aprendizagem da estrutura, um dos objectivos desta dissertação é apresentar e debater um conjunto de métodos de aproximação à resolução deste ponto fundamental da modelização difusa.

Assim, numa primeira abordagem, mais directa, o universo de discurso de cada variável de entrada é particionado, correspondendo a cada partição um termo linguístico. Na globalidade, todo o espaço de entrada é particionado, constituindo-se uma grelha multidimensional. Esta estratégia apresenta a desvantagem de o número de regras difusas crescer exponencialmente à medida que a dimensão do espaço de entrada aumenta: o problema designado por explosão da base de regras. Assim, na prática, esta abordagem é viável unicamente para sistemas com um número reduzido de entradas, tipicamente não mais de quatro.

De forma a diminuir a dimensão da base de regras, os algoritmos de *eliminação de regras* revelam-se interessantes num contexto de modelização neuro-difusa, uma vez que permitem reduzir o número de regras do sistema de inferência. Neste contexto, apresenta-se e discute-se o algoritmo de aprendizagem de estrutura de Lin [Lin, 1995].

Uma outra estratégia consiste na utilização de algoritmos de agrupamento de classes, os quais particionam o espaço de entrada-saída de forma mais flexível, diminuindo a dimensão da base de regras. Assim, são descritos e analisados alguns algoritmos de agrupamento, bem como as suas possibilidades no que toca à aprendizagem de regras para sistemas de inferência difusos. Uma das desvantagens desta abordagem resulta de, tipicamente, verificar-se um nível elevado de redundância relativamente aos termos linguísticos das variáveis de entrada e saída, i.e., geralmente há um grande número de termos linguísticos idênticos que se repetem.

No sentido de diminuir a redundância dos termos linguísticos, utilizam-se *medidas de similaridade difusa* com a finalidade de detectar e fundir termos linguísticos semelhantes. Deste modo, são descritas algumas medidas de similaridade difusa, assim como técnicas de fusão de conjuntos difusos.

Um outro aspecto importante, comum a qualquer procedimento de modelização, convencional ou inteligente, reside no problema da *selecção das variáveis de entrada relevantes*. É prática corrente utilizar-se conhecimento a priori sobre os sistemas de forma a determinar-se que variáveis utilizar, com que atraso e com que regressões. Consequentemente, um dos objectivos deste trabalho consiste, justamente, em apresentar e discutir algumas técnicas de selecção de entradas relevantes.

1.2.2. Aprendizagem de Parâmetros

Após a tomada de decisão quanto à selecção das entradas e regras do modelo difuso, é importante sintonizar os parâmetros dos termos linguísticos das variáveis de entrada e saída do sistema, e.g., os centros e larguras das funções de pertença em que se baseiem os termos linguísticos. Deste modo, as redes neuronais difusas revelam-se de grande utilidade por permitirem essa sintonização por meio de técnicas de optimização não linear, nomeadamente, pelo *método do gradiente*. Esta metodologia, de utilização geral, apresenta a desvantagem de não garantir a convergência dos parâmetros para o mínimo global, para além de ter associados alguns problemas em termos de velocidade de aprendizagem. Deste modo, descreve-se um esquema híbrido de optimização, o qual é susceptível de ser utilizado em estruturas difusas do tipo Takagi-Sugeno. A estratégia referida baseia-se na optimização linear dos consequentes das regras difusas, e.g., *estimador dos mínimos quadráticos*, e na optimização não linear das premissas.

As metodologias descritas nos parágrafos precedentes têm por denominador comum o desenvolvimento de modelos fora de linha não sendo, portanto, directamente aplicáveis em tempo real. De facto, as estratégias enunciadas requerem que se proceda à aquisição prévia de dados de entrada e saída do sistema, com base nos quais é, então, implementado um modelo difuso, encapsulado numa rede neuronal difusa. Deste modo, a aproximação referida não é indicada para sistemas variantes no tempo, para os quais se requer adaptação dos parâmetros do modelo em tempo real. Assim sendo, os algoritmos de aprendizagem de parâmetros fora de linha são adaptados de forma a possibilitarem o treino incremental de redes neuro-difusas, tarefa esta apoiada pela circunstância das estruturas utilizadas gozarem da propriedade da localidade. Relativamente à aprendizagem incremental da estrutura, o seu estudo não é levado a cabo neste trabalho.

1.2.3. Interpretabilidade

Um aspecto habitualmente ignorado no contexto da modelização difusa prende-se com a verificação da manutenção da transparência linguística dos modelos obtidos. Dado que a questão da interpretabilidade constitui uma das vantagens potenciais do desenvolvimento de modelos difusos, este trabalho de investigação procura avaliar a possibilidade de se atingir o objectivo referido, sem que a precisão do modelo resultante se degrade de forma inaceitável. Neste sentido, foi incorporado um procedimento de monitorização o qual visa manter a interpretabilidade linguística do modelo durante a aprendizagem de parâmetros.

1.3. Contribuições da Dissertação

Do estudo, análise e desenvolvimentos realizados neste trabalho resultou um conjunto de contribuições científicas, algumas de carácter original, segundo o conhecimento do autor.

Assim, em termos do problema global da identificação difusa, algumas das técnicas apresentadas na literatura foram combinadas entre si, no sentido de estabelecer sinergias entre elas. Um exemplo paradigmático, referente à arquitectura NFCN de Lin (Capítulo 5), prende-se com a substituição do algoritmo de aprendizagem da estrutura original pelo algoritmo de agrupamento subtractivo, o qual se revelou bastante mais eficiente. Também relacionado com a arquitectura NFCN, foram efectuadas algumas adaptações, nomeadamente em termos de operadores difusos e tipos de funções de pertença utilizadas, tendo sido efectuado um estudo sobre as suas principais vantagens e limitações, bem como as situações em que cada parâmetro particular é recomendado. Como consequência da generalização das funções de pertença Gaussianas definidas originalmente, houve a necessidade de desenvolver um método de desfuzificação adequado. Assim, tanto o desfuzificador implementado, como as adaptações à estrutura inicial, podem ser consideradas contribuições originais.

Em termos de estruturas difusas utilizadas, efectuou-se um estudo experimental detalhado, o qual possibilitou a obtenção de alguns resultados conclusivos quanto às potencialidades e aplicabilidade de modelos difusos do tipo Takagi-Sugeno e linguísticos.

Em relação ao aspecto da interpretabilidade, o problema da redundância de funções de pertença, resultante da aplicação de métodos de agrupamento de classes na aprendizagem da estrutura, foi detectado, tendo sido solucionado pela aplicação de medidas de similaridade difusa conducentes à fusão de conjuntos difusos semelhantes. Posteriormente, de forma a que a interpretabilidade fosse garantida durante a aprendizagem de parâmetros, projectou-se um esquema de monitorização o qual garante a possibilidade de distinção entre funções de pertença, o qual constitui outra das contribuições originais do trabalho.

Relativamente aos algoritmos utilizados, procurou-se efectuar uma análise tão detalhada quanto possível com base não só na informação bibliográfica recolhida mas também nas ilações retiradas experimentalmente.

Uma outra contribuição do trabalho desenvolvido prende-se com o facto de todos os algoritmos analisados terem sido implementados na linguagem de programação C++, no sentido do desenvolvimento de uma ferramenta computacional com fins pedagógicos e de investigação.

1.4. Organização da Dissertação

O documento presente está organizado em sete capítulos independentes, os quais se pretende que estejam relacionados e interligados, com o intuito de apresentar o conteúdo deste trabalho de forma tão coerente e clara quanto possível.

Os três capítulos iniciais constituem, então, a primeira parte da tese, na qual são apresentados os conceitos base necessários à compreensão dos capítulos seguintes, capítulos esses que abordam questões estritamente relacionadas com a identificação neuro-difusa.

Assim, após o capítulo introdutório, o Capítulo 2 discute os aspectos essenciais da identificação de sistemas, sendo descritos aspectos relativos ao projecto e recolha de dados experimentais, determinação de estruturas para modelos, estimação de parâmetros e validação. Este capítulo procura apresentar, com base numa espinha dorsal comum, aspectos típicos de identificação de sistemas lineares e não lineares.

O Capítulo 3 descreve os princípios fundamentais dos sistemas difusos, indispensáveis à compreensão dos restantes aspectos da dissertação. Não se pretende abordar os formalismos matemáticos mais sofisticados da lógica difusa mas sim apresentar conceitos base tais como conjunto difuso, função de pertença, regra difusa, bem como a estrutura dos sistemas difusos. Em relação ao último item citado, são descritos os parâmetros e funções associados a cada um dos módulos de um sistema difuso.

Na mesma linha do capítulo precedente, o Capítulo 4 apresenta os fundamentos de redes neuronais, utilizados ao longo do texto. Assim, introduzem-se os conceitos básicos de neurónio artificial e rede neuronal artificial, bem como as suas topologias e métodos de treino mais comuns. Neste capítulo descreve-se a estrutura RBF, dada a sua relação com os sistemas difusos, e os algoritmos de aprendizagem dos mínimos quadráticos e retropropagação.

O Capítulo 5 apresenta inicialmente um resumo da história e estado da arte relativamente ao problema da identificação neuro-difusa. Seguem-se então as questões fundamentais a abordar neste trabalho, nomeadamente, a aprendizagem da estrutura, a qual engloba a aprendizagem de regras e selecção de entradas, a aprendizagem de parâmetros fora de linha, a interpretabilidade linguística e alguns aspectos de aprendizagem incremental de parâmetros. Deste modo, pode afirmar-se que o quinto capítulo constitui o núcleo da dissertação presente.

Os formalismos apresentados no Capítulo 5 são ilustrados no Capítulo 6 através da realização de algumas experiências baseadas em casos de estudo utilizados frequentemente na literatura. Com o apoio dos resultados experimentais obtidos são retiradas algumas conclusões relativamente ao desempenho das diferentes estruturas difusas, assim como do tipo de funções de pertença e operadores difusos. É ainda efectuado um estudo relativo a um sistema real, nomeadamente uma planta de branqueamento de pasta de papel.

O Capítulo 7 apresenta as conclusões fundamentais do trabalho realizado, apontando algumas perspectivas de trabalho futuro, no sentido de ser dada uma resposta a algumas das questões que ficaram por responder.

Finalmente, são apresentadas todas as referências bibliográficas citadas ao longo do texto.

Capítulo 2

IDENTIFICAÇÃO DE SISTEMAS

A identificação de sistemas aborda a construção de modelos com base em dados experimentais. Assim, na construção de um modelo com base na teoria da identificação de sistemas, os seus parâmetros são adaptados segundo um determinado critério, com o intuito de se obter uma representação final susceptível de reproduzir com sucesso os dados empíricos utilizados. Neste sentido, a qualidade das amostras utilizadas, a estrutura assumida para o modelo, bem como o estabelecimento de critérios adequados de estimação de parâmetros e de validação revestem-se de grande importância.

Este capítulo começa por enquadrar a identificação de sistemas na área mais abrangente da modelização. Assim, na Secção 2.1 são apresentados os conceitos básicos subjacentes à modelização de sistemas, assim como as diferentes estratégias disponíveis. Na Secção 2.2 introduzem-se os aspectos fundamentais da identificação de sistemas, aspectos esses desenvolvidos nas secções posteriores. Neste sentido, a Secção 2.3 aborda o problema da recolha de dados experimentais e os factores associados à sua qualidade. Na Secção 2.4 são descritas as estruturas mais utilizadas na construção de modelos baseados em dados, assim como os factores a ter em consideração na sua selecção. A Secção 2.5 apresenta as questões essenciais relativamente à estimação de parâmetros e apresenta alguns dos métodos de mais utilizados. O problema da validação de modelos, nomeadamente os critérios utilizados, constitui o objecto da Secção 2.6.

2.1. Fundamentos de Modelização de Sistemas

2.1.1. Sistemas Dinâmicos

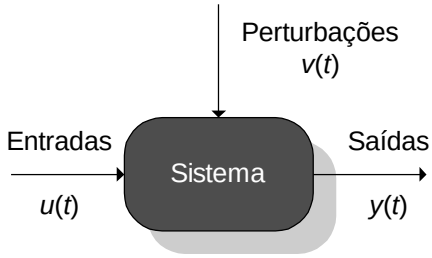


Figura 2.1. Descrição conceptual de sistema.

Em termos genéricos, um sistema é uma entidade na qual interagem variáveis de tipos diferentes, produzindo resultados eventualmente observáveis, tal como é representado na Figura 2.1 [Söderström e Stoica, 1989].

Assim, as *saídas* de um sistema, $y(t)$, são influenciadas por um conjunto de *entradas externas* (ou *controláveis*), $u(t)$, e *perturbadoras*, $v(t)$. Tais perturbações podem derivar, por exemplo, de ruído nos instrumentos de medida ou de factores externos não controláveis. A título de exemplo, considere-se um sistema industrial, designadamente um tanque de mistura (Figura 2.2).

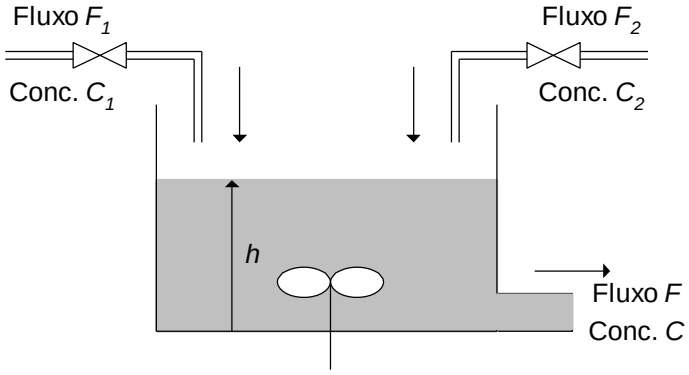


Figura 2.2. Tanque de mistura.

Neste sistema, dois líquidos com concentrações e fluxos variáveis são misturados. Os fluxos $F_1(t)$ e $F_2(t)$ são controlados por válvulas. Já as concentrações, $C_1(t)$ e $C_2(t)$, não são controláveis, pelo que constituem entradas perturbadoras. O fluxo de saída $F(t)$ e a concentração no tanque $C(t)$ são as saídas do sistema.

No exemplo referido, as saídas num dado instante, i.e., o fluxo $F(t)$ e a concentração $C(t)$, não dependem unicamente das entradas nesse instante, controláveis e perturbadoras: dependem também dos seus valores passados. Sistemas deste tipo designam-se por *sistemas dinâmicos* [Ljung, 1987; Martins de Carvalho, 1993]. A compreensão, a previsão e o controlo de sistemas dinâmicos constituem alguns dos objectivos da modelização de sistemas.

2.1.2. Finalidade da Modelização de Sistemas

O objectivo, porventura mais marcante, da modelização de sistemas relaciona-se com o *projecto de sistemas de controlo* [Ogata, 1990; Kuo, 1987; Åstrom e Wittenmark, 1984; Franklin e Powell, 1980]. Habitualmente, o projecto de controladores é efectuado com recurso a um modelo da planta[2] a controlar. Teoricamente, quanto melhor for o modelo do processo, melhor se comportará o sistema de controlo projectado.

A modelização é também importante em problemas de *predição* [Box e Jenkins, 1970; Åstrom, 1970; Kalman e Bucy, 1961]. Por exemplo, a previsão da evolução do mercado de capitais ou a previsão do valor da saída de um variável, necessária num esquema de controlo predictivo.

[2]No contexto da teoria dos sistemas e do controlo, planta é uma peça de equipamento com o objectivo de desempenhar uma determinada tarefa. Tipicamente, qualquer objecto físico susceptível de ser controlado é designado por planta (fornalha, planta de branqueamento, reactor químico, …)

Outra importância da modelização advém do facto de permitir o cálculo de variáveis não mensuráveis directamente. Este aspecto é particularmente importante em situações em que, por questões custo (económico, tecnológico), não é possível medir uma variável importante do sistema.

Na área do *diagnóstico de falhas*, a utilização de modelos do sistema em causa constitui um mecanismo de detecção de comportamentos erróneos.

Uma outra aplicação da modelização está relacionada com a obtenção de um *melhor conhecimento de um sistema*. Neste sentido, é importante que a informação representada seja transparente, i. e., o modelo obtido seja facilmente interpretável.

Dependendo de factores como a complexidade do sistema em análise ou a aplicação desejada, poder-se-ão construir modelos de tipos diversos, com recurso a técnicas variadas.

2.1.3. Tipos de Modelos

Claramente, qualquer sistema dinâmico pode ser modelizado com um maior ou menor grau de formalismo matemático, de acordo com certos factores como o grau de precisão necessário, o custo financeiro, o peso computacional ou a complexidade.

Assim, tomando por exemplo a condução de um carro, o modelo assumido pelo condutor é do tipo *mental* ou *linguístico*. Modelos desta classe não envolvem a utilização de qualquer espécie de formalização matemática do sistema a representar. O acto de conduzir consiste, pois, num conjunto de regras linguísticas do género: pressionar o acelerador aumenta a velocidade, pressionar o travão diminui a velocidade [Söderström e Stoica, 1989]. O mesmo se passa para o caso de operadores industriais. Estes modelos são construídos com base na experiência e são puramente qualitativos.

Outro modo de estudar um sistema consiste na construção de um *modelo físico* no mesmo espaço conceptual, mas a uma escala diferente. Uma estratégia análoga consiste na construção de um circuito analógico de simulação, equivalente ao sistema real. Neste caso, há uma migração do espaço conceptual inicial para o seu equivalente eléctrico. A vantagem desta técnica é a de permitir a modelização física a um custo menor, uma vez que, em geral, o custo de desenvolvimento de um modelo físico no mesmo espaço conceptual é superior ao custo de desenvolvimento do seu equivalente electrónico. Porém, modelos desta natureza sofrem da desvantagem de serem pouco flexíveis.

Existe uma classe de sistemas cujas características essenciais podem ser estudadas com base em *modelos gráficos* ou *tabelas* [Ljung, 1987]. Sabe-se que para determinar as características fundamentais de um sistema linear basta conhecer as suas respostas a impulso ou a degrau, ou a sua resposta em frequência.

A construção de modelos dos tipos enunciados para aplicações mais avançadas, nomeadamente para predição ou projecto de sistemas de controlo, padece de limitações óbvias. Naturalmente, em situações mais exigentes, é necessário determinar-se um *modelo matemático* do sistema.

2.1.4. Modelização Analítica e Identificação

Na modelização matemática, as relações entre as variáveis do sistema são representadas em termos de estruturas matemáticas, e. g., equações diferenciais, equações de diferenças ou equações algébricas. A vantagem da modelização matemática deriva essencialmente, da circunstância de se

tratar de uma ferramenta de modelização genérica, permitindo representar um determinado sistema com maior ou menor grau de rigor, de acordo com os objectivos específicos do modelo.

A construção de um modelo matemático pode ser conduzida por meio de duas estratégias fundamentais: modelização analítica ou modelização experimental (ou sua combinação). Qualquer que seja a metodologia seguida, é fundamental utilizar o máximo de *conhecimento a priori* sobre o sistema em estudo, o qual se pode apresentar de diversas maneiras. Um desses modos, utilizado na modelização analítica [Wellstead, 1979], consiste numa *descrição mecanística* do sistema: o mesmo é descrito, fundamentalmente, com base nas leis da mecânica, leis físico-químicas ou termodinâmicas (modelização *caixa-branca* [Bossley, 1997]). Por exemplo, um circuito eléctrico analógico poderá ser descrito pelas leis de Kirchoff, dos nós e das malhas. É este o tipo de conhecimento a priori utilizado na modelização analítica. Nesta situação, as observações recolhidas não são utilizadas na modelização; são-no, unicamente, na validação do modelo. Esta abordagem é a mais tradicional e, claramente, apresenta dificuldades no tratamento de sistemas complexos, não lineares, estocásticos ou variantes no tempo. A construção de um modelo analítico para sistemas dessa natureza é de difícil realização devido aos aspectos não triviais de análise fenomenológica, ao seu custo, sobretudo a nível de tempo de desenvolvimento e, consequentemente, a nível económico, e à sua reduzida flexibilidade. O problema essencial desta estratégia reside na circunstância de que, à medida que a complexidade de um sistema cresce, a capacidade de o descrever com rigor e precisão diminui, tal como se referiu no capítulo introdutório através do princípio da incompatibilidade. Esta dificuldade sugere a utilização de outro tipo de metodologias, como por exemplo a *identificação de sistemas*.

A identificação de sistemas [Söderström e Stoica, 1989; Ljung, 1987] tem por objectivo a construção de modelos baseados em dados; por este motivo, é também designada por modelização experimental, em contraste com a modelização analítica, baseada em relações matemáticas representativas das leis físico-químicas utilizadas na descrição da realidade. Na identificação de sistemas, são adquiridos dados de entrada e saída do processo, dados esses sujeitos a uma análise posterior, no sentido de se inferir uma sua representação. O modelo obtido é designado por *caixa-negra* [Sjöberg et al, 1994; Ljung, 1987], uma vez que exprime unicamente as relações entre as entradas e as saídas do sistema, ignorando-se o seu interior. Modelos deste tipo são desenvolvidos por meio de estimação de parâmetros de modelos de regressão linear ou não linear - *modelos paramétricos* -, estimação essa levada a cabo unicamente com recurso aos dados de entrada-saída.

A teoria clássica de identificação de sistemas apresenta um rigor e uma base teórica bastante sólida. No entanto, a sua aplicabilidade a sistemas não lineares é limitada. Deste modo, são necessárias técnicas capazes de lidar eficazmente com questões de não linearidade e incerteza. Algumas dessas técnicas são originárias da Inteligência Artificial, pelo que se designam por técnicas inteligentes tal como será exposto na secção seguinte.

2.1.5. Identificação Inteligente

A modelização analítica e a identificação clássica de sistemas apresentam alguns problemas descritos nas secções precedentes. Na tentativa de ultrapassar os problemas inerentes à identificação clássica, assiste-se, actualmente, a um forte impulso no sentido da investigação da viabilidade da aplicação de técnicas inteligentes à modelização de sistemas dinâmicos. Deste modo, à modelização de sistemas utilizando técnicas inteligentes, com base em dados experimentais, dá-se o nome de *identificação inteligente*.

Assim, as técnicas inteligentes procuram dotar de características humanas os sistemas onde a sua utilização seja realizada. Além do referido, procuram ainda diminuir algumas das limitações tipicamente humanas. Assim, por um lado requer-se criatividade, abstracção, aprendizagem, adaptação, capacidade de generalização e transparência do conhecimento representado, e por outro procura-se ultrapassar as limitações humanas no que respeita a cansaço, subjectividade e não repetibilidade.

Uma das áreas que tem vindo a merecer um destaque particular tem sido a das *redes neuronais artificiais* [Haykin, 1994; Kröse e van der Smagt, 1993]. Estas estruturas, inicialmente com o objectivo de emularem o funcionamento do cérebro humano, apresentam como atractivo principal a sua capacidade de aprendizagem e adaptação. Deste modo, conseguem representar sistemas com dinâmica complexa. De facto, a aplicação de redes neuronais artificiais (Capítulo 4) à identificação de sistemas [Pham e Xing, 1995; Hunt et al, 1992; Narendra e Parthasarathy, 1990] constitui uma ferramenta com grandes potencialidades, dado que essas estruturas funcionam como aproximadores universais [Funahashi, 1989]. No entanto, na identificação de sistemas, lineares ou não lineares, a informação armazenada não é facilmente interpretável de forma qualitativa, i.e., subsiste o problema da falta de *transparência* do conhecimento representado: os modelos obtidos são do tipo caixa-negra, além de que conhecimento prévio disponível não é facilmente incluído. Neste contexto, surge a modelização difusa baseada em dados, ou *identificação difusa*, descrita na Secção 2.1.6[3].

Apesar de todo o seu potencial, a afirmação definitiva da aplicação das técnicas inteligentes a situações do mundo real, onde segurança, previsibilidade e correcção são requisitos essenciais, requer o estabelecimento de resultados rigorosos e objectivos em termos de convergência e estabilidade dos algoritmos utilizados. Particularmente, no caso do controlo inteligente, é fundamental estabelecer critérios precisos no que toca à análise da estabilidade do sistema de controlo. Em termos de controlo difuso, von Altrock afirma que a questão da estabilidade é um falso problema, uma vez que um controlador difuso pode ser classificado como um "controlador não linear multivariável", de acordo com a teoria clássica [von Altrock, 1995]. Os problemas encontrados na sua análise são, deste modo, os mesmos que se encontram presentes na análise de sistemas não lineares: estudos analíticos de estabilidade são praticamente impossíveis e requerem modelos precisos. Boas referências de base para a análise das questões de estabilidade de sistemas difusos podem ser encontradas em [Wang, 1994] ou [Tanaka e Sugeno, 1992]. Em termos de modelização e controlo neuronal, o problema maior reside na análise de convergência dos algoritmos utilizados. Neste sentido, existem alguns resultados para redes com camadas lineares, nomeadamente redes RBF.

2.1.6. Identificação Difusa

As vantagens dos sistemas difusos residem no facto de a informação ser representada de forma *transparente*, sendo o *conhecimento a priori* eventualmente disponível incluído facilmente no modelo, além de tais estruturas permitirem representar eficientemente sistemas complexos (o ponto essencial do princípio da incompatibilidade). De facto, uma grande parte destes sistemas

[3] Tal como se verificará no Capítulo 3, nem todos os esquemas de modelização difusa gozam da propriedade da transparência linguística. Neste grupo incluem-se, por exemplo, os sistemas Takagi-Sugeno de ordem 1.

gozam da propriedade da aproximação universal (Secção 3.4).

Na construção de um modelo difuso, a selecção das regras baseia-se, usualmente, no conhecimento heurístico de um ou mais peritos no sistema a modelizar, i.e., num conjunto de regras linguísticas estabelecidas por um perito com base na sua intuição e experiência. Trata-se, pois, de um modelo mental ou linguístico, também designado por algoritmo difuso (Secção 3.3). No entanto, se o conhecimento qualitativo de que um perito humano dispõe apresenta vantagens em termos de transparência da informação, a sua capacidade de quantificar esse mesmo conhecimento é limitada. Por exemplo, o operador de uma cadeia de produção poderá ser capaz de descrever a planta com base num conjunto de regras linguísticas puramente qualitativas, sem dispor de qualquer conhecimento da realidade físico-química subjacente a essas mesmas regras. Tais regras seriam do tipo (para o caso do controlo do nível num tanque) (2.1):

SE (*nível é baixo*) ENTÃO (*abertura da válvula é alta*). (2.1)

No entanto, os sinais *nível* e *abertura da válvula* são grandezas quantitativas. Por conseguinte, de forma a articular a informação de natureza qualitativa fornecida pelos peritos com a informação quantitativa dos sinais do sistema, é fundamental descrever, com rigor, o conceito matemático dos termos linguísticos *baixo* e *alta*, tal como será abordado no Capítulo 3.

Para além da quantificação dos termos linguísticos, o conjunto de regras utilizadas pelo perito não é, em geral, nem completo nem absolutamente rigoroso. Basicamente, as regras periciais constituem uma base valiosa para a construção de um protótipo. No entanto, há que aprimorar esse modelo inicial. Deste modo, põe-se, também, a questão da aprendizagem autónoma da base de regras, com base em dados de entrada-saída.

Assim, poder-se-á definir, genericamente, modelização difusa como a tarefa de representação das características de um determinado sistema por meio dos formalismos dos conjuntos e sistemas difusos [Zadeh, 1971], designando-se, particularmente, identificação difusa como a construção de modelos difusos baseada em dados experimentais. Este tipo de modelização designa-se por modelização *caixa-cinzenta*[4] [Bossley, 1997].

2.1.7. Identificação Neuro-Difusa

A identificação difusa apresenta como tarefas essenciais a aprendizagem de uma estrutura, designadamente, a selecção de um conjunto de regras relevantes, e a atribuição de valores aos parâmetros presentes na estrutura determinada, i.e., a parametrização de funções de pertença. Uma das metodologias utilizáveis na consecução dos objectivos citados, nomeadamente em relação ao segundo ponto, consiste na representação do modelo difuso em questão por meio de uma rede neuronal. Esta estrutura é designada, habitualmente, por rede neuro-difusa, dado constituir uma arquitectura neuronal susceptível de implementar um sistema difuso. A sua função essencial é, então, permitir o ajuste dos parâmetros do modelo da mesma maneira que se treina uma rede neuronal. O estudo das questões fundamentais de identificação neuro-difusa constitui o objectivo principal desta dissertação.

[4] Nesta classe de modelos incluem-se também os modelos analíticos para os quais são utilizados dados empíricos como auxílio à atribuição de valores relativos a parâmetros físicos.

2.2. Aspectos Genéricos de Identificação de Sistemas

Tal como foi referido, o objectivo da identificação de sistemas é a construção de modelos baseados em dados experimentais. De maneira mais formal, esta tarefa é influenciada por cinco factores fundamentais [Söderström e Stoica, 1989; Ljung, 1987]: um *sistema*, S; uma *condição experimental*, C, uma *estrutura*, M; um *método de identificação*, I; e um *critério de validação*, V.

Deste modo, em relação ao sistema a identificar, S, as suas características vão nortear o processo de identificação. Por conseguinte, é importante levar-se em consideração algum conhecimento eventualmente disponível sobre o sistema, nomeadamente em termos de linearidade ou não linearidade, variância ou invariância temporal ou ainda em termos de aspectos de determinismo ou estocasticidade. O sistema em causa irá influenciar, deste modo, a escolha do tipo de modelo para o representar, tal como será referido na Secção 2.4.

A primeira etapa da identificação consiste na aquisição de um conjunto de N amostras de dados de entrada, *u(t)*, e saída, *y(t)* do sistema, com base na condição experimental C (2.2)[5]:

$$Z^N = \{[u(1), y(1)], [u(2), y(2)], \ldots, [u(N-1), y(N-1)], [u(N), y(N)]\} \tag{2.2}$$

Os aspectos genéricos desta tarefa serão apresentados na Secção 2.3.

Na obtenção de um modelo capaz de representar um determinado sistema considera-se, tipicamente, um *conjunto de modelos candidatos*, M^*, sobre os quais incidirá a procura. Assim, a título ilustrativo, poder-se-á definir M^* como o conjunto de todos os modelos lineares, ou, de modo mais restritivo, como o conjunto de todos os modelos lineares de 2ª ordem. No caso particular da identificação de sistemas, são utilizados modelos paramétricos, pelo que se obtém uma estrutura, $M(q)$, em que q designa o conjunto de parâmetros da estrutura M. Em termos genéricos, um modelo paramétrico pode ser representado como em (2.3):

$$y(t) = g\left(t, Z^{t-1}, v(t); q(t)\right) \tag{2.3}$$

onde g representa o mapeamento das entradas e saídas passadas na saída actual, conduzido pelos parâmetros do sistema, e Z^{t-1} denota o conjunto de amostras obtidas até ao instante t-1. Alguns casos particulares de (2.3) serão apresentados na Secção 2.4. Uma vez que a construção de modelos é efectuada com recurso a um conjunto limitado de dados Z^N (2.2) e, habitualmente, na presença de ruído , a função g (2.3), dificilmente será obtida na prática. Assim, o que se tem, em geral, é uma sua aproximação (2.4):

$$\hat{y}(t) = \hat{g}\left(t, Z^{t-1}, \hat{v}(t); \hat{q}(t)\right) \tag{2.4}$$

sendo $\hat{y}(t)$ a saída prevista pelo modelo para o instante t, $\hat{g}$ a aproximação obtida pelo modelo para a função g, $\hat{v}(t)$ o modelo das perturbações e $\hat{q}$ o vector dos parâmetros obtidos com base no conjunto de dados utilizados.

Após a parametrização da estrutura, há que determinar o melhor modelo, com base na procura de valores adequados para os seus parâmetros, por meio de um determinado método de

---

[5] Por simplicidade de notação, serão considerados sistemas com uma entrada e uma saída (SISO - *Single Input Single Output*) . A generalização para sistemas com várias entradas e uma saída (MISO - *Multiple Input Single Output*) ou várias saídas (MIMO - *Multiple Input Multiple Output*) será discutida em situações em que a notação se considere não trivial.

identificação *I*. O objectivo principal consiste em estimar o conjunto de parâmetros *q* de forma a obter-se um modelo tal que o valor por ele previsto, $\hat{y}(t)$, seja o mais próximo possível do valor real *y*(*t*). Por outras palavras, pretende-se que a *capacidade predicitiva* do modelo seja adequada. Um dos modos de o conseguir consiste em estimar os parâmetros de modo que o *erro de predição*, *e*(*t*), seja tão pequeno quanto possível (Figura 2.3). Tal metodologia enquadra-se na classe dos métodos de predição de erro (Secção 2.5).

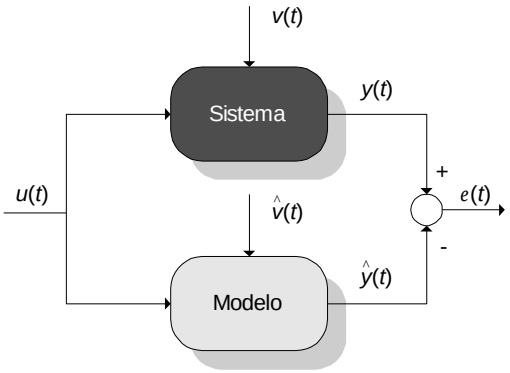


Figura 2.3. Minimização do erro de predição.

O objectivo supracitado pode ser expresso pelo estabelecimento de um critério que conduza à minimização do erro de predição. Tipicamente, utiliza-se o critério dos mínimos quadráticos (LS[6]), descrito na Secção 4.4 (2.5):

$$J_N = \frac{1}{2}\sum_{i=1}^{N}\left(y(t)-\hat{y}(t)\right)^2 \tag{2.5}$$

Na secção 2.5, serão debatidos os aspectos principais da selecção de um critério de identificação.

Finalmente, o modelo obtido terá que ser validado com base num determinado critério de validação *V*. Tipicamente, a validação é efectuada com base no erro de predição do modelo. É importante notar que a minimização de J_N não conduz necessariamente à obtenção de um modelo adequado. De facto, usualmente J_N pode ser minimizado para valores próximos de zero. No entanto, essa circunstância não garante a reprodução satisfatória de dados não incluídos no desenvolvimento do modelo, a qual poderá ser pobre. Deste modo, para que a capacidade de representação do modelo seja adequada, boas propriedades de *generalização* são fundamentais, de forma a que o modelo possa reproduzir, com precisão suficiente, dados nunca antes apresentados.

Alcançar uma capacidade de generalização satisfatória, requisito fundamental para a validação do sistema, nem sempre é um problema trivial. O *princípio da parcimónia*[7] sugere, intuitivamente, que se procurem modelos tão simples quanto possível, dado que:

O modelo mais simples aceitável produz os melhores resultados.

[6] *Least Squares*, em terminologia inglesa.
[7] Ou *Occam's razor*, em terminologia inglesa.

Este princípio heurístico tem por base o facto de que modelos com um grau de flexibilidade desnecessariamente elevado poderem resultar numa má capacidade de generalização, em consequência da susceptibilidade de se ajustarem ao ruído e outras peculiaridades dos dados. Deste modo, verifica-se a necessidade de lidar, simultaneamente, com flexibilidade e simplicidade: pretende-se, por um lado, que o modelo seja suficientemente flexível de modo a captar os aspectos essenciais da dinâmica do sistema, e por outro, que o sistema seja tão simples quanto possível.

Assim, na Secção 2.6 serão apresentados alguns critérios de validação, bem como outros aspectos a ter em conta para além da capacidade de predição.

Em jeito de resumo, este capítulo tem por objectivo principal a apresentação dos aspectos genéricos do ciclo de identificação (Figura 2.4). As questões presentes, quer em esquemas de identificação convencionais, quer inteligentes, ainda que com condicionalismos próprios a cada um dos casos, são apresentadas. Deste modo, referem-se aspectos relativos à condição experimental, à selecção de uma estrutura, à estimação dos parâmetros do modelo e à sua validação.

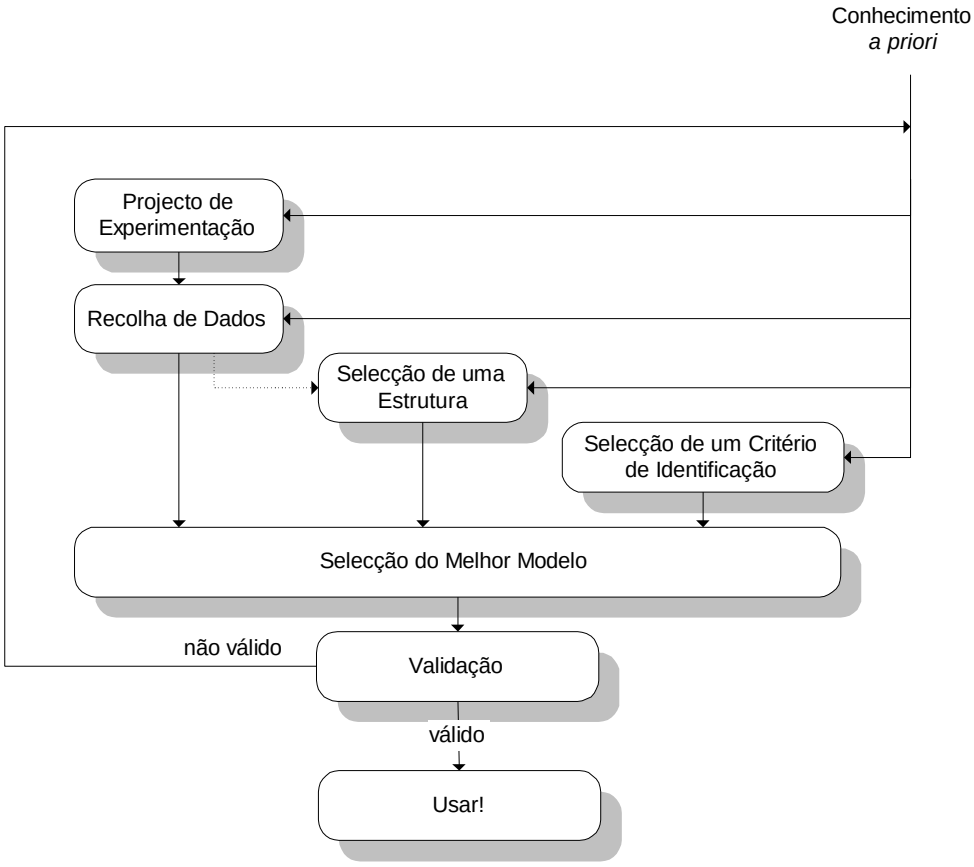


Figura 2.4. O ciclo de identificação.

2.3. Recolha de Dados de Identificação

A identificação de sistemas começa sempre por uma fase de experimentação, na qual é efectuada uma recolha de dados, com o máximo de riqueza informativa possível. Uma vez que, em geral, esta tarefa apresenta custos económicos elevados, além de requerer um tempo de execução considerável, um bom projecto das condições experimentais revela-se de grande importância para a

qualidade final dos dados adquiridos.

Assim sendo, antes de mais há que determinar que sinais se devem considerar como entradas e que sinais como saídas, seguindo-se a escolha do intervalo de amostragem, a manipulação das entradas seleccionadas, a decisão quanto ao número de amostras a adquirir e o pré-processamento dos dados recolhidos.

De forma a que as amostras adquiridas sejam susceptíveis de descrever os aspectos fundamentais da dinâmica do sistema a identificar, os dois primeiros pontos revelam-se fundamentais. Assim, quanto ao intervalo de amostragem, a sua selecção deve ser adequada, havendo alguns critérios que a orientam [Åstrom e Wittenmark, 1984]. Quanto ao modo de manipulação dos sinais, essa decisão deve ser tomada com base na *persistência de excitação*. Essencialmente, quanto mais ricos forem os sinais de entrada, mais rica será a qualidade dos dados obtidos para as saídas.

Na identificação de sistemas lineares, é condição suficiente para uma boa identificação, i.e., para a convergência dos parâmetros para os valores correctos, que a entrada seja persistentemente excitadora de ordem *np*, sendo *np* o número de parâmetros a determinar. Por exemplo, um sistema linear de 2ª ordem (função de transferência com 4 parâmetros) poderá ser identificado com base numa entrada persistentemente excitadora de ordem 4, e.g., uma entrada que seja a soma de dois sinais sinusoidais com frequências distintas. Ljung [Ljung, 1987] analisa detalhadamente as questões relativas a persistência de excitação em sistemas lineares. No entanto, para sistemas não lineares, a conclusão expressa acima não se aplica directamente: é necessário desenvolver novos conceitos para esta classe de sistemas. Em [Gorinevsky, 1995], a análise do problema da persistência de excitação em redes RBF é conduzida[8]. Nesse estudo, prova-se que se consegue persistência de excitação se os sinais de entrada pertencerem à vizinhança dos centros da rede. No entanto, quanto à dimensão da vizinhança, é referido apenas que "pode ser grande", sendo, portanto, uma descrição puramente qualitativa. Estas conclusões poderão ser aplicadas a sistemas difusos Takagi-Sugeno (até à primeira ordem), uma vez que a sua estrutura é, em certas situações, funcionalmente equivalente à das redes RBF (Secção 4.3). Para sistemas difusos linguísticos poder-se-ão generalizar, de maneira meramente intuitiva, as conclusões supracitadas, impondo, sobre os neurónios de saída, as mesmas restrições efectuadas sobre os nós de entrada: os sinais de saída devem pertencer à vizinhança dos centros das funções de pertença de saída. É importante referir que, tanto para redes RBF como para sistemas difusos do tipo Takagi-Sugeno, a persistência de excitação é fundamental para garantir a convergência dos parâmetros da componente linear da sua estrutura. No entanto, para estruturas não lineares, como é o caso dos sistemas difusos linguísticos, o facto de os sinais de entrada e de saída satisfazerem as condições enunciadas não constitui, por si só, uma garantia de convergência. Somente se poderá argumentar que os dados de identificação serão mais ricos, potenciando uma modelização satisfatória. Nesta situação, requer-se que as entradas sejam suficientemente ricas tanto em magnitude como em frequência, de modo a excitar todos os estados da planta, por todo o espaço de entrada. O procedimento habitual consiste em utilizar como sinais de entrada o somatório de sinusoidais de várias frequências e amplitudes, ou sinais aleatórios.

Um aspecto de grande importância na realização dos ensaios de recolha de dados advém do facto de, em certos tipos de sistemas, o projectista não gozar de liberdade absoluta para manipular as variáveis que desejar. Tal situação acontece frequentemente em sistemas de produção contínua,

[8] Alguns dos conceitos abordados na descrição efectuada são apresentados no Capítulo 3 e no Capítulo 4.

nos quais a mesma não pode ser interrompida para se efectuarem as experiências requeridas. Nestas situações, a aquisição de dados tem que ser realizada em malha fechada, durante o funcionamento normal do sistema. Neste caso, a questão da qualidade da informação extraída da amostragem é mais sensível. No caso de sistemas lineares, há uma teoria bem definida, que, essencialmente, se baseia num conjunto de condições de identificabilidade que devem ser satisfeitas. A possibilidade de um sistema linear ser ou não identificável em malha fechada depende, sobretudo, das características da malha de realimentação [Ljung, 1987]. Mais uma vez, para sistemas não lineares, não existe uma teoria de aplicação genérica. No entanto, é referido em [Ljung, 1987] que controladores não lineares, variantes no tempo ou de ordem elevada, conduzem, regra geral, a experiências suficientemente informativas. É, contudo, comum que o índice informativo de dados recolhidos durante o funcionamento normal do sistema seja limitado.

2.4. Selecção de uma Estrutura

A selecção de uma estrutura a utilizar na identificação do sistema em causa é, sem dúvida, a decisão mais importante e mais complexa de toda a tarefa de identificação. Tal escolha deve ser fundamentada no conhecimento do processo de identificação, assim como no conhecimento e intuição sobre o sistema a identificar. Deste modo, a experiência, intuição e conhecimento do sistema por parte do projectista revestem-se de importância fulcral.

Para além da utilização de conhecimento prévio disponível, a selecção de uma estrutura deve ser norteada segundo o compromisso entre a flexibilidade e simplicidade da classe de modelos considerada, com vista à obtenção de modelos com capacidade de generalização satisfatória. Na verdade, a escolha de uma estrutura demasiado simples, com um número reduzido de parâmetros, i.e., *sobparametrizada*, poderá redundar na incapacidade de representação do sistema - o problema do *sobajustamento*. Por outro lado, uma estrutura demasiado flexível, i.e., *sobreparametrizada*, poderá originar o *sobreajustamento* dos dados aos parâmetros. Um dos casos em que o fenómeno referido ocorre, deriva do facto do número de graus de liberdade do modelo, i.e., o número de parâmetros a ajustar, ser superior ao número de amostras. Nesta situação, verifica-se um bom comportamento do modelo em relação aos dados utilizados no seu desenvolvimento havendo, contudo, uma capacidade de generalização deficiente, para dados nunca antes apresentados. Pelo exposto, torna-se clara a importância de uma selecção adequada da estrutura, o que envolve aspectos como o *tipo* de modelo, a sua *dimensão* e *parametrização*. A Figura 2.5 ilustra os problemas enunciados.

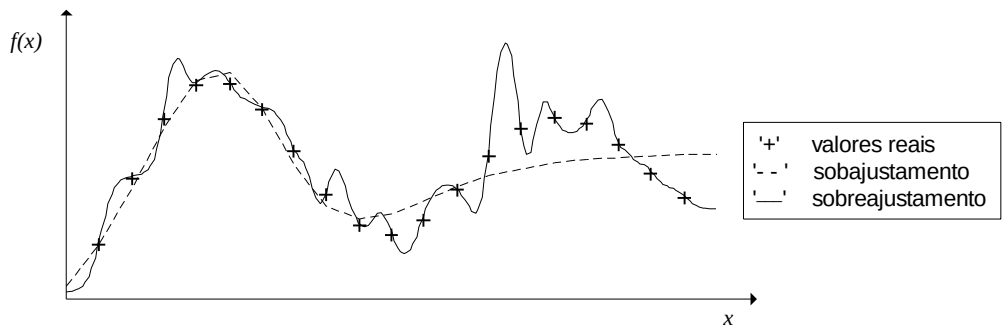


Figura 2.5. Os problemas do sobajustamento e do sobreajustamento.

Para além da importância da selecção adequada da estrutura, bem como do número de amostras a recolher, no contexto das redes neuronais o problema do sobreajustamento pode advir de treino excessivo. Tipicamente, à medida que se ajustam os parâmetros da rede, o erro relativo aos dados de treino diminui, acompanhado pela diminuição no erro face aos dados de teste. Porém, no caso das amostras utilizadas conterem regularidades erróneas derivadas da amostragem, a rede adapta-se no sentido da sua representação, pelo que o erro de treino diminui e o erro de teste começa a aumentar. Este problema é designado por *sobretreino* [Reed, 1993] e é, geralmente, abordado estabelecendo como critério de paragem para o treino (Secção 4.5.1) o aumento do erro em relação aos dados de teste.

2.4.1. Tipo de Modelo

Para além da caracterização apresentada no início do capítulo presente, os modelos matemáticos podem ainda ser caracterizados com base nas relações matemáticas utilizadas: *contínuos* ou *discretos*, caso o modelo assuma tempo contínuo ou discreto, respectivamente; *lineares ou não lineares*, de acordo com a natureza das relações entre as variáveis do sistema; *variantes* ou *invariantes*, se os parâmetros utilizados no modelo apresentarem alguma dependência temporal; *determinísticos* ou *estocásticos*, se a sua caracterização for feita sem qualquer espécie de ambiguidade ou se for requerida a inclusão de informação probabilística (e.g., caracterização das perturbações); de *parâmetros distribuídos* ou *aglomerados*, se a sua caracterização envolver equações diferenciais às derivadas parciais ou às derivadas totais, respectivamente.

No que toca ao tipo de modelos a utilizar, há que decidir, essencialmente, entre modelos contínuos ou discretos, variantes ou invariantes, lineares ou não lineares, e modelos de entrada-saída ou modelos de espaço de estados.

Modelos paramétricos físicos, i.e., modelos cujos parâmetros tenham um significado físico interpretável, favorecem a obtenção de um bom compromisso entre flexibilidade e simplicidade. No entanto, este tipo de modelos padece das dificuldades referidas anteriormente no que respeita à gestão da complexidade. No caso do custo de obtenção de um modelo físico ser viável, o conhecimento a priori deriva das leis físico-químicas, sendo incorporado mais intuitivamente em modelos contínuos. No entanto, o esforço e tempo computacional associados à estimação de parâmetros em sistemas contínuos são elevados. Deste modo, modelos paramétricos discretos são favorecidos, o que conduz ao desenvolvimento de modelos caixa-negra ou modelos difusos (caixa-cinzenta) baseados em dados, utilizados neste trabalho.

Em termos de variância temporal, há que decidir quanto à dependência temporal dos parâmetros do modelo. Naturalmente que sistemas cujas características variem com o tempo são representados mais adequadamente por parâmetros que se adaptem em conformidade com as alterações na dinâmica do sistema. Este aspecto requer a aplicação de algoritmos iterativos, susceptíveis de aplicação em tempo real, tal como será abordado no Capítulo 5.

Quanto à questão da linearidade, é óbvio que, na presença de sistemas lineares (ou sistemas não lineares, cuja dinâmica possa ser aproximada por um modelo linear), não faz sentido utilizar modelos não lineares, sempre mais complexos, em contradição com o princípio da parcimónia. Por outro lado, em sistemas fortemente não lineares, as capacidades de representação de modelos lineares são limitadas. De entre as várias estratégias utilizadas no tratamento de sistemas fortemente não lineares, a mais flexível e genérica consiste no projecto de modelos não lineares globais. Apesar de todo o seu potencial, a sua aplicação prática apresenta algumas dificuldades, resultantes, fundamentalmente, dos problemas inerentes aos métodos de optimização não linear,

como por exemplo, o método do gradiente (Secção 4.4). A dificuldade mais premente prende-se com a análise da convergência dos parâmetros. Enquanto que para técnicas de optimização lineares, como o estimador dos mínimos quadráticos, há resultados estabelecidos em termos de convergência e variância, o mesmo não se verifica para a optimização não linear. De facto, nesta situação os parâmetros de um modelo poderão convergir para um óptimo local e não para o óptimo global. A análise desta e de outras questões é efectuada com maior detalhe na Secção 4.5.1.

Em termos de utilização de modelos de espaço de estados ou de entrada-saída, é sabido que os primeiros são mais vantajosos, uma vez que constituem uma representação mais completa do sistema em causa, dado que não consideram unicamente os sinais de entrada e saída mas também a informação interna do sistema, presente nos seus estados. Além do referido, modelos de espaço de estados permitem uma representação uniforme, tanto para sistemas SISO como para sistemas MIMO. Apesar das vantagens descritas, a utilização deste género de modelos levanta algumas dificuldades. De facto, nem sempre todos os estados do sistema estão acessíveis, sendo, deste modo, necessário implementar um observador, no caso de o sistema ser observável. Assim, os estados inacessíveis são reconstruídos a partir da informação disponível [Friedland, 1986; Luenberger, 1971]. Mais uma vez, para sistemas lineares há uma teoria sólida, sistemática para a construção de observadores. No entanto, a análise de observabilidade e a implementação de observadores para sistemas não lineares constitui um problema complexo, apesar de alguns esforços levados a cabo [Henriques e Dourado, 1998; Thau, 1973]. Deste modo, no contexto da identificação de sistemas não lineares, os modelos de entrada-saída são mais comuns, assumindo-se que as amostras recolhidas, relativas às variáveis observadas, contêm informação suficiente acerca de todos os estados do sistema [Brown e Harris, 1994], sendo, assim, utilizados nesta dissertação.

2.4.2. Dimensão do Modelo

A decisão quanto à dimensão do modelo envolve três aspectos fundamentais: o problema da *selecção da ordem*, o problema da *selecção do atraso* de transporte associado a cada variável de entrada (caso se utilizem modelos de entrada-saída) e o problema da *selecção das variáveis físicas* a incluir no modelo.

No estudo de sistemas lineares, as respostas às questões enunciadas podem ser obtidas com base em algumas técnicas baseadas na análise preliminar dos dados, nomeadamente [Ljung, 1987]: exame da estimação da função de transferência por análise espectral; teste da característica de matrizes de co-variância; correlação de variáveis; e exame da matriz de informação. No problema de determinação da ordem, em sistemas lineares, o critério de Akaike (AIC[9]) [Akaike, 1973] é um dos mais conhecidos.

Se para sistemas lineares, como se tem vindo a verificar, há uma base teórica sólida, o mesmo não se verifica no estudo de sistemas não lineares. De facto, as metodologias de selecção de entradas e estimação da ordem e atraso em sistemas não lineares constituem, geralmente, aproximações heurísticas. Os casos - poucos - em que existem resultados consubstanciados, exigem a introdução de assunções fortemente restritivas relativamente às características do sistema a identificar. Algumas das técnicas utilizadas na tentativa de dar resposta às questões referidas são apresentadas na Secção 5.2.3. Assim, do exposto conclui-se que na selecção da ordem, do atraso e

[9] *Akaike's Information Criterion*, em terminologia inglesa.

das variáveis físicas, em sistemas não lineares, a importância do conhecimento prévio sobre o sistema em causa poderá ser determinante no sentido de ser dada uma resposta válida aos problemas enunciados. Caso esta informação não esteja disponível, utiliza-se frequentemente uma estratégia do tipo *força bruta*: testam-se várias hipóteses e escolhe-se a que permite obter os melhores resultados. É ainda importante realçar que a ordem a escolher, directamente associada ao número de parâmetros do modelo, depende, adicionalmente, do número de amostras recolhidas na medida em que, se este for reduzido, o número de parâmetros terá que ser compatível, de modo a que não ocorram situações de sobreajustamento aos dados, tal como foi referido anteriormente.

2.4.3. Parametrização do Modelo

Após a escolha de uma classe de modelos, a sua parametrização é conduzida, de acordo com a abordagem de modelização seguida. Assim, nos parágrafos posteriores serão descritos esquemas de parametrização típicos em modelos lineares e não lineares, entre os quais os modelos difusos.

Assim, na derivação de um modelo paramétrico de entrada-saída, a incorporação da dinâmica do sistema, i.e., o efeito das entradas e saídas passadas na saída futura, apresenta algumas dificuldades. De facto, as diversas estruturas apresentadas posteriormente não dispõem de memória dinâmica. Assim sendo, é necessário incluir, ainda que de maneira artificial, dinâmica temporal no modelo, o que é conseguido através da introdução de linhas de atraso[10]. Assim, as entradas e saídas passadas do sistema a modelizar são tratadas como entradas do modelo. Deste modo, esta técnica converte um problema de modelização temporal - a inclusão da dinâmica do sistema no domínio temporal - num problema de modelização espacial - o mapeamento estático de entradas e saídas atrasadas na saída futura. A Figura 2.6 ilustra os aspectos referidos.

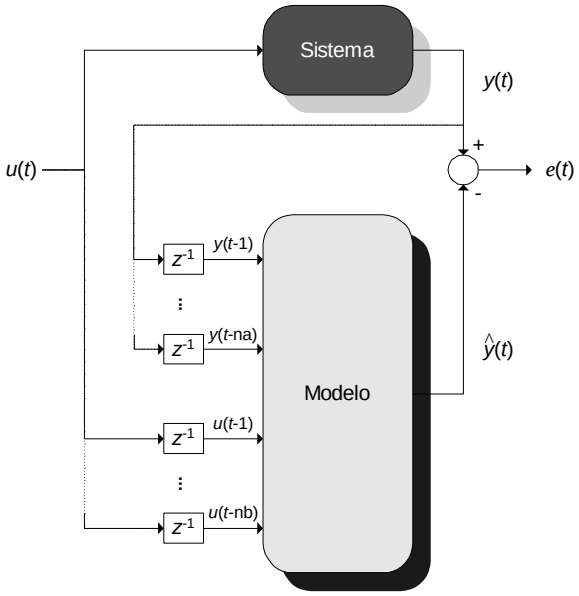


Figura 2.6. Estrutura série-paralela com linhas de atraso.

[10] *Tapped delay lines*, em terminologia inglesa.

Na figura precedente, z^{-1} denota o operador atraso, sendo *nb* e *na*, respectivamente, a memória associada à entrada e à saída, i.e., o número de regressões da entrada e da saída.

O método de incorporação de dinâmica pode ser levado a cabo segundo duas estruturas fundamentais: a estrutura *paralela* e a *série-paralela* [Pham e Xing, 1995; Narendra e Parthasarathy, 1990]. A última, representada na figura anterior, é a mais utilizada, o que se deve ao facto de, na estrutura paralela, o modelo receber as saídas passadas computadas pelo próprio, e não as saídas passadas reais fornecidas pelo sistema. Deste modo, o modelo e o sistema não interferem entre si, funcionando em paralelo. Deste aspecto resulta que não haja garantia da convergência dos parâmetros, mesmo no caso linear [Narendra e Parthasarathy, 1990]. Assim sendo, a estrutura série-paralela é a utilizada neste trabalho.

Desta aproximação estática advêm, naturalmente, alguns aspectos indesejáveis. Uma das suas limitações resulta da necessidade de desenvolvimento de modelos com um número frequentemente elevado de entradas. Este número poderá crescer de modo ainda mais significativo no caso da memória correcta ser desconhecida, situação em que, tipicamente, se sobrestima a ordem do sistema. Do problema da dimensão da camada de entrada resulta, ainda, que o modelo se torne mais susceptível ao ruído externo.

Em virtude dos problemas referidos, resultantes da inexistência de dinâmica temporal, as redes neuronais recorrentes (Secção 4.2.3) afiguram-se, à primeira vista, como mais adequadas ao tratamento de problemas onde a dinâmica seja determinante. Uma estrutura recorrente particularmente adequada aos objectivos referidos é a rede de Elman [Elman, 1990]. No entanto, no contexto de modelização neuro-difusa, as estruturas desenvolvidas são do tipo estático, pelo que as redes recorrentes não serão consideradas neste trabalho.

Modelos lineares

No caso de sistemas lineares consideram-se, por exemplo, um conjunto de funções de transferência susceptíveis de captar a dinâmica do sistema. Neste caso, um modelo linear, SISO, discreto pode ser descrito genericamente por (2.6):

$$Y(z) = G(z)U(z) + V(z) \tag{2.6}$$

onde $G(z)$ denota a *função de transferência* do sistema [Ogata, 1990; Martins de Carvalho, 1993]. Em (2.6), os sinais $U(z)$ e $Y(z)$ representam, respectivamente, as transformadas de Z dos sinais de entrada $u(t)$ e saída $y(t)$, enquanto que $V(z)$ representa o efeito de perturbações, $v(t)$, para as quais se assume, habitualmente, um efeito aditivo. Usualmente, a função de transferência *G(z)* é representada por uma expressão racional do tipo (2.7):

$$G(z) = \frac{b_1 z^{-1} + \cdots + b_{nb} z^{-nb}}{1 + a_1 z^{-1} + \cdots + a_{na} z^{-na}} z^{-d} = \frac{B(z)}{A(z)} z^{-d} \tag{2.7}$$

em que *d* denota o atraso do sistema. Na mesma expressão, a_1, …, a_{na} e b_1, …, b_{nb} constituem os parâmetros da função de transferência (daí a designação de *modelo paramétrico*), cuja identificação constitui o objectivo final.

No que respeita a perturbações, a sua inclusão num modelo é efectuada com recurso a ferramentas probabilísticas, dada a sua natureza estocástica. Uma caracterização completa das perturbações a que um dado sistema é sujeito, obtém-se com base na sua função de densidade de probabilidade condicional conjunta. No entanto, atendendo às dificuldades associadas à sua determinação, uma aproximação simplificada e, em geral, suficientemente versátil é utilizada. Assim, uma perturbação $v(t)$ é representada pela sua transformada de Z do seguinte modo (2.8):

$$V(z) = H(z)E(z) \tag{2.8}$$

onde *E(z)* representa a transformada de Z de *e*(*t*), uma sequência de variáveis aleatórias independentes e identicamente distribuídas (i.i.d.), de média nula, com uma certa função de densidade de probabilidade (FDP), f_e, designada por *ruído branco*. Deste modo, tanto *e*(*t*) como *v*(*t*) constituem processos estocásticos [Papoulis, 1973].

Uma maneira particularmente simples de representar um modelo consiste em assumir que a perturbação num instante *t* é susceptível de ser representada por um ruído branco. Nesta situação, o sistema a modelizar é representado pela equação de diferenças (2.9):

$$y(t) = -a_1 y(t-1) - \cdots - a_{na} y(t-na) + b_1 u(t-d-1) + \cdots b_{nb} u(t-d-nb) + e(t) \tag{2.9}$$

obtida considerando *H*(*z*)=1 em (2.8). O conjunto de parâmetros do modelo poderá ser representado, abreviadamente, por um vector q (2.10):

$$q = \left[a_1 \; a_{21} \; \ldots \; a_{na} \; b_1 \; b_2 \; \ldots \; b_{nb}\right]^T \tag{2.10}$$

Introduzindo o vector de regressões $j\,(t)$ (2.11):

$$j\,(t) = \left[-y(t-1) \cdots - y(t-na) \; u(t-d-1) \cdots u(t-d-nb)\right]^T \tag{2.11}$$

a equação de diferenças (2.9) é descrita, em notação abreviada, por (2.12):

$$y(t) = j^{\,T}(t)q + e(t) \tag{2.12}$$

A estrutura representada em (2.9) e (2.12) é habitualmente designada por *ARX*[11]. De maneira mais genérica, poder-se-á definir uma família de estruturas como em (2.13) [Söderström e Stoica, 1989; Ljung, 1987]:

$$A(z)Y(z) = \frac{B(z)}{F(z)} z^{-d} U(z) + \frac{C(z)}{D(z)} E(z) \tag{2.13}$$

No caso particular em que se considera *F*(*z*) = *C*(*z*) = *D*(*z*) = 1, obtém-se a estrutura ARX, referida no parágrafo anterior.

Na estrutura ARX, a flexibilidade associada ao modelo da perturbação é limitada. De facto, tal como foi referido, assume-se que as perturbações podem ser representadas por um ruído branco, o que é bastante restritivo. Como forma de minorar esta limitação, surge a estrutura *ARMAX*[12], a qual generaliza a estrutura ARX, incorporando no modelo a média móvel do ruído (2.14):

$$Y(z) = \frac{B(z)}{A(z)} z^{-d} U(z) + \frac{C(z)}{A(z)} E(z) \tag{2.14}$$

originando a equação de diferenças (2.15) para o modelo:

$$\begin{aligned} y(t) = &-a_1 y(t-1) - \cdots - a_{na} y(t-na) + b_1 u(t-d-1) + \cdots b_{nb} u(t-d-nb) + \\ &+ e(t) + c_1 e(t-1) + \cdots + c_{nc} e(t-nc) \end{aligned} \tag{2.15}$$

Em (2.15), *nc* designa o número de regressões da variável *e*(*t*). Deste modo, para além dos

[11] *Auto Regressive with eXogenous inputs*, em terminologia inglesa.
[12] *Auto Regressive Moving Average with eXogenous inputs*, em terminologia inglesa.

parâmetros a_1, …, a_{na} e b_1, …, b_{nb} (2.10), há ainda os parâmetros c_1, …, c_{nc}. O problema essencial desta estrutura relaciona-se com o facto dos ruídos $e(t\text{-}1)$, …, $e(t\text{-}nc)$ não serem mensuráveis. Com o objectivo de ultrapassar esta limitação, utilizam-se os erros de predição $e(t\text{-}1)$, …, $e(t\text{-}nc)$ em lugar do ruído.

Modelos não lineares

Claramente, (2.15) não é mais que um caso particular de (2.3), aplicável a sistemas lineares. De facto, a expressão (2.3) constitui uma representação genérica de sistemas, tanto lineares como não lineares. Assim, como extensão à estrutura ARMAX para o caso não linear, surge a estrutura *NARMAX*[13] (2.16):

$$y(t) = g\left(t, Z^{t-1}, e(t-1),...,e(t-nc); q\right) + e(t) \tag{2.16}$$

Tal como no caso ARMAX, são utilizados os erros de predição passados, $e(t\text{-}1)$, …, $e(t\text{-}nc)$, no modelo. Em alternativa a esta estrutura, é usual considerar-se a estrutura *NARX*[14] (2.17):

$$y(t) = g(t, Z^{t-1}; q) + e(t) \tag{2.17}$$

Tal como foi referido anteriormente, na prática o que se obtém é uma aproximação da função g. No caso em que assume a estrutura NARX, sabendo que se considera ruído branco tem-se (2.18):

$$\hat{y}(t) = \hat{g}(t, Z^{t-1}; \hat{q}) \tag{2.18}$$

Esta estrutura é a mais popular no contexto de sistemas não lineares, em consequência da sua simplicidade.

Modelos difusos

Um caso particular da estrutura NARX advém da utilização de estruturas difusas na identificação de sistemas dinâmicos. Neste caso, a estrutura NARX é designada mais adequadamente por *FARX*[15] [Dias e Dourado, 1999]. Os capítulos subsequentes desta dissertação baseiam-se, precisamente, nesta estrutura. Os modelos FARX são representados por um conjunto de regras do tipo R_i (2.19):

$$R_i : Se \;\; y(t-1) \; é \; A_{1i} \;\; e \;\; u(t-d) \; é \; B_{1i} \;\; então \;\; \hat{y}(k) \; é \; C_{1i} \tag{2.19}$$

onde A_{ji}, B_{ji} e C_{ji} denotam os termos linguísticos associados a cada entrada e saída, definidos pelas suas funções de pertença: $m_{A_{ji}}, m_{B_{ji}}, m_{C_{ji}}$. Tal como se pode concluir, o modelo obtido constituirá uma função do tipo (2.18), em resultado da agregação de todas as regras do modelo difuso. A selecção de um conjunto de regras do tipo (2.19) e a definição dos conjuntos difusos A_{ji}, B_{ji} e C_{ji}, para além de outros parâmetros, constituem aspectos de projecto específicos de sistemas difusos (Secção 3.3).

---

[13] *Nonlinear ARMAX*, em terminologia inglesa.
[14] *Nonlinear ARX*, em terminologia inglesa.
[15] *Fuzzy ARX*, em terminologia inglesa.

2.5. Selecção de um Critério de Identificação

Efectuada a parametrização do modelo, a selecção de um critério de identificação, i.e., de um método de estimação dos parâmetros do modelo, constitui o ponto seguinte a abordar. Por outras palavras, o problema essencial reside em, com base na informação contida num conjunto de N amostras recolhidas, Z^N (2.2), numa estrutura e num método de identificação, determinar valores adequados para os parâmetros do modelo (2.20):

$$Z^N \rightarrow \hat{q} \tag{2.20}$$

Pretende-se, assim, que a estimação dos parâmetros $\hat{q}_N$, seja *consistente*. Uma estimação diz-se consistente no caso dos parâmetros estimados tenderem para os reais (2.21):

$$\hat{q} \rightarrow q_0, \quad N \rightarrow \infty \tag{2.21}$$

Assim, um sistema classifica-se como *identificável*, numa dada estrutura, se a estimação dos parâmetros que a constituem for consistente [Söderström e Stoica, 1989]. A consistência (ou inconsistência) de uma estimação é, deste modo, função da estrutura considerada, do método de identificação e das condições experimentais. No que toca à selecção da estrutura, na Secção 2.4 foram considerados aspectos relevantes da sua selecção, bem como dos problemas que poderão advir de uma escolha deficiente, nomeadamente em termos de sobajustamento e sobreajustamento. Quanto às condições experimentais, foram referidas, na Secção 2.3, as questões essenciais relativas à recolha de dados suficientemente informativos, particularmente a necessidade de persistência de excitação. Seguidamente, são descritos alguns dos aspectos relativos aos métodos de identificação.

Ljung [Ljung, 1987] divide os métodos de identificação em dois ramos essenciais: *métodos de predição de erro* e *métodos de correlação*. A primeira metodologia consiste em obter um critério de medida do valor do erro de predição, $e(t)$, e avaliar o modelo de acordo com esse erro. Ao invés, os métodos de correlação baseiam-se no requisito de que o erro de predição não se relacione com a sequência de dados utilizados. Deste modo, num modelo satisfatório, os erros de predição são independentes dos dados passados. A estimação de parâmetros em sistemas não lineares baseia-se, geralmente, em técnicas de optimização não linear. Alguns desses métodos são aplicados ao treino de redes neuronais, tal como se referirá no Capítulo 4. Essas técnicas enquadram-se nos métodos de predição de erro, pelo que, apenas esta classe será referida neste trabalho.

Assim, um dos métodos de predição de erro mais utilizados na estimação de parâmetros é o método dos mínimos quadráticos (LS) [Widrow e Hoff, 1960], descrito na Secção 4.4. Tal como se verificará o método LS, um método de optimização baseado na descida do gradiente, apresenta vantagens importantes em problemas de optimização linear. De facto, o algoritmo dos mínimos quadráticos caracteriza-se pela verificação da propriedade da consistência, com a condição de que a matriz de co-variância seja não singular e de que o ruído presente nos dados seja branco ou a sequência de entrada seja independente da sequência de ruído. A prova deste teorema é apresentada detalhadamente em [Ljung, 1987]. Assim, para que a matriz de co-variância seja não singular, requer-se que a entrada seja persistentemente excitadora de ordem igual ao número de parâmetros do modelo a identificar.

Apesar das suas vantagens, o método dos mínimos quadráticos, na sua versão original iterativa ou versão na analítica, constitui uma técnica de optimização linear. Deste modo, não é aplicável directamente a problemas de optimização não linear, pelo que, neste caso, se aplicam

outras metodologias mais gerais. Nesta situação, é comum utilizar-se o algoritmo de retropropagação do erro, o qual resulta da generalização do método LS, proposto originalmente por Widrow e Hoff para o treino de redes neuronais multicamada. A principal desvantagem do algoritmo reside no facto de não se tratar de um método consistente, uma vez que não há qualquer garantia de que os parâmetros do modelo convirjam para os reais. Nesta dissertação utilizar-se-á o algoritmo de retropropagação do erro com ligeiras modificações, nomeadamente pela definição de uma velocidade de aprendizagem adaptativa. Apesar dos aspectos negativos referidos, em muitas situações a solução (subóptima) obtida é satisfatória.

2.6. Validação

O algoritmo de estimação de parâmetros seleccionado, e.g., mínimos quadráticos ou retropropagação, determina um modelo, eventualmente o melhor, de entre os candidatos expressos numa qualquer estrutura paramétrica. Levanta-se, então, a questão de avaliar a qualidade do modelo obtido, a qual constitui o objecto da etapa de validação.

Idealmente, o objectivo da modelização seria obter um clone perfeito do sistema real. No entanto, na prática esta situação é impossível, quer seja pela complexidade das suas interacções, as quais podem unicamente ser aproximadas por relações matemáticas, quer seja pelas limitações inerentes à identificação de sistemas com base num número finito de amostras. Deste modo, na modelização de um sistema, a avaliação da qualidade do modelo obtido é colocada de uma forma pragmática. É, pois, fundamental conhecer a aplicabilidade do modelo ao propósito para o qual foi desenvolvido, o qual poderá consistir, por exemplo, numa base para o projecto de um controlador ou num predictor. Assim, de acordo com o tipo de modelo em causa, utilizam-se critérios de validação adequados.

Em modelos analíticos, uma das maneiras mais naturais de validar do modelo obtido consiste em confrontar os valores estimados, bem como as suas variâncias, com os valores esperados com base em informação prévia.

No contexto da modelização caixa-negra, a validação baseia-se, fundamentalmente, nas propriedades de entrada-saída do modelo. No caso de sistemas lineares é usual proceder-se a testes de validação baseados na análise estatística dos erros de predição, i.e., resíduos. Naturalmente, a aplicabilidade deste tipo de testes a modelos não lineares é limitada. Assim sendo, em problemas deste tipo, a metodologia de validação mais usual consiste na inspecção do modelo por *simulação*. Neste caso, o modelo é testado com base em dados desconhecidos, i.e., dados não utilizados na construção do modelo, comparando-se as saídas obtidas com as saídas reais do sistema. O desvio do modelo em relação ao sistema é medido, usualmente, com base em critérios de predição de erro. Deste modo, os dados recolhidos do sistema são divididos em dois conjuntos: um para a determinação dos parâmetros do modelo e outro para o seu teste. Assim, no caso do critério apresentar resultados satisfatórios face aos dados de teste, i.e., se o modelo apresentar capacidades de generalização adequadas, poderá ser aprovado. No entanto, é ainda importante colocar a hipótese de uma estrutura mais adequada poder originar melhores resultados, pelo que é usual comparar diversos modelos resultantes de parametrizações distintas, em termos de um critério de predição de erro.

Um aspecto importante da identificação de sistemas prende-se com o facto de que as etapas descritas ao longo deste capítulo, de uma forma sequencial, serem, normalmente, recursivas. De

facto, quando um modelo não é aprovado nos testes de validação, há que procurar as razões que possam ter conduzido a esse insucesso nas diversas etapas enunciadas. Assim, uma vez que se referiu que a escolha de uma estrutura é a tarefa mais complexa e problemática de todo o processo de identificação, o mais natural é procurar falhas nesta etapa, nomeadamente na selecção do tipo de modelo, das variáveis nele incluídas, sua ordem e atraso. Efectivamente, decisões inadequadas relativamente aos aspectos referidos, poderão "obrigar" o modelo a captar uma dinâmica diferente da do sistema real. Além disso, poderão originar problemas, tanto de sobreajustamento (por excesso de parâmetros), como de sobajustamento (por falta de parâmetros). Adicionalmente, o critério de identificação poderá não ser o mais adequado à estrutura considerada. As dificuldades resultantes dos esquemas de optimização não lineares, referidas anteriormente, poderão constituir outra causa. Outra origem potencial de falhas será, porventura, uma aquisição de dados deficiente. Nesta situação, os dados recolhidos não são suficientemente informativos para que se consiga uma identificação aceitável. Por exemplo, uma escolha inadequada do intervalo de amostragem não permitirá captar correctamente a dinâmica do sistema, o que também poderá resultar da não aquisição de variáveis importantes para o processo.

Para além da capacidade de predição, há ainda um conjunto de outros objectivos cujo grau de importância é função da aplicação em causa, tais como *interpretabilidade*, *complexidade algorítmica* e *adaptabilidade* [Bossley, 1997].

Em termos de interpretabilidade, se se pretender construir um modelo como meio de descoberta de conhecimento, é fundamental que a informação final nele contida seja facilmente compreendida, i.e., seja transparente. Definem-se, essencialmente, três níveis de transparência. No primeiro, menos exigente, requer-se unicamente conhecimento sobre as entradas e saídas passadas (regressores) que afectam a saída efectiva. No segundo nível, a interpretação do sistema é feita com recurso a expressões matemáticas simples. No terceiro, o mais exigente, requer-se que o sistema seja descrito por um conjunto de regras linguísticas, de forma a obter-se uma representação qualitativa do sistema. É na consecução deste nível de interpretabilidade que a modelização difusa encontra o seu expoente máximo. Particularmente, modelos difusos baseados em regras com consequentes difusos possibilitam, sob algumas restrições analisadas posteriormente, a obtenção de modelos interpretáveis. Ao invés, modelos difusos do tipo Takagi-Sugeno de ordem 1 não são interpretáveis linguisticamente. Este último nível será o abordado neste trabalho (Secção 5.4).

Para além dos aspectos inerentes à descoberta de conhecimento, modelos que englobem os aspectos de interpretabilidade descritos contribuem para um melhor conhecimento do sistema em causa, além de permitirem validação pericial: o modelo poderá ser avaliado com base na análise, efectuada por um perito ou operador, das regras contidas no modelo. De notar que, frequentemente, se verifica um compromisso entre precisão e interpretabilidade. Esta circunstância deve-se a que, em certas situações de maior complexidade, para se obter um modelo facilmente interpretável, a capacidade de predição diminui, em consequência das restrições de interpretabilidade impostas. Adicionalmente, se tais restrições aumentarem, a capacidade de predição apresentará uma tendência para diminuir (Figura 2.7). Naturalmente, a importância da transparência num modelo com capacidades de representação limitadas é mínima.

Em determinadas situações, o esforço computacional permitido para a obtenção de um modelo é restringido, dada a limitação de recursos disponíveis. Nestes casos, a obtenção de modelos que se coadunem com as limitações impostas é fundamental. Modelos desenvolvidos com base em algoritmos complexos, de grande exigência a nível de capacidade de processamento e com necessidades de memória computacional elevada poderão tornar-se proibitivos em certas ocasiões.

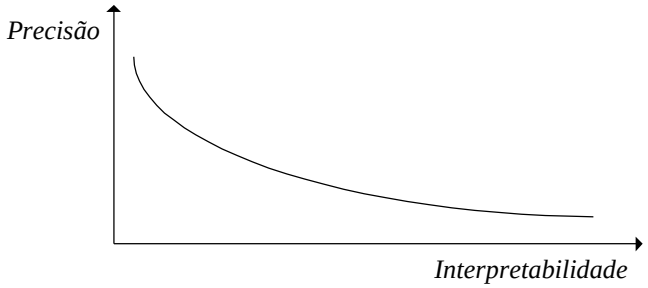


Figura 2.7. O compromisso interpretabilidade/precisão.

Directamente relacionada com a complexidade algorítmica está a eventual necessidade de adaptabilidade em linha, i.e., durante o funcionamento do sistema. Nomeadamente, na modelização e controlo de sistemas variantes no tempo, é importante que o modelo possa adaptar-se em tempo real, adaptação essa conduzida com base nos dados apresentados em linha. Essa capacidade de adaptação depende das características do modelo, assim como do peso computacional do algoritmo utilizado para a aprendizagem dos parâmetros do modelo.

2.7. Sumário

Este capítulo abordou a problemática fundamental da identificação de sistemas, particularmente os aspectos referentes ao ciclo de identificação.

A identificação clássica de sistemas surge como uma primeira resposta às dificuldades subjacentes à modelização analítica em sistemas de complexidade elevada. Apesar das suas vantagens, a sua aplicação a sistemas não lineares não é genérica. Deste modo, esquemas de modelização caixa-negra não linear, tais como redes neuronais, são propostos, os quais apresentam como principal limitação a falta de transparência da informação armazenada. Esta (possível) dificuldade leva à identificação difusa, a qual apresenta a vantagem de permitir a implementação de modelos interpretáveis. Neste sentido, surgem ainda as redes neuro-difusas caracterizadas pela conjugação das vantagens das redes neuronais, a nível de capacidades de aprendizagem e adaptação, com a interpretabilidade dos sistemas difusos, as quais serão o objecto do Capítulo 5.

Na Secção 2.2, apresentaram-se, resumidamente, os aspectos essenciais de identificação de sistemas, nomeadamente a recolha de dados de identificação, a selecção de uma estrutura e de um critério, assim como a validação do modelo obtido, aspectos esses sintetizados no ciclo de identificação.

A Secção 2.3 abordou o problema do projecto das condições experimentais de recolha de dados e os factores associados à sua qualidade, nomeadamente as questões inerentes à persistência de excitação dos sinais de entrada. Verificou-se que, para sistemas não lineares, o problema da persistência de excitação das variáveis incluídas no modelo é abordado de forma heurística.

A Secção 2.4 apresentou os aspectos principais da tarefa mais marcante da identificação de sistemas: a selecção de uma estrutura adequada. Aqui, referiu-se a importância da determinação correcta da dimensão do modelo, i.e., da sua ordem e atraso. Para além deste ponto, foram descritas as estruturas paramétricas mais utilizadas, destacando-se a estrutura NARX e a corresponde FARX.

Na Secção 2.5, foram apresentados os aspectos a considerar na selecção de um critério de identificação de parâmetros. Aqui, destacou-se a importância da consistência de um método de

identificação, a qual se verifica no critério dos mínimos quadráticos, sob determinadas restrições relacionadas com o ruído presente nos dados, a adequação da estrutura considerada e a persistência de excitação dos sinais utilizados. Ao invés, os algoritmos de optimização não linear referidos, e.g., retropropagação, não verificam a propriedade da consistência, uma vez que tais métodos não garantem a obtenção da solução óptima do problema em causa.

Finalmente, os aspectos a ter em conta na validação de modelos, assim como alguns dos critérios utilizados, constituíram o objecto da Secção 2.6. Referiu-se que a propriedade essencial a satisfazer por um modelo é uma capacidade de generalização satisfatória. Deste modo, o modelo é inspeccionado por simulação, com base no seu comportamento face a dados nunca utilizados, com recurso a critérios de medição do erro de predição. Referiu-se ainda que a interpretabilidade, a complexidade computacional e a adaptabilidade constituem parâmetros a considerar em algumas situações, consoante os objectivos de modelização a atingir.

Em face do exposto, e uma vez que o trabalho presente se baseia em aspectos de identificação neuro-difusa, os dois capítulos seguintes introduzem, resumidamente, as questões fundamentais relativas a sistemas difusos e a redes neuronais artificiais.

Capítulo 3

FUNDAMENTOS DE SISTEMAS DIFUSOS

A teoria dos sistemas difusos constitui uma metodologia particularmente adequada ao tratamento de problemas de identificação e controlo de sistemas, nomeadamente em situações com grau de complexidade elevada, onde a presença de não linearidades e de factores de incerteza seja significativa. De facto, o tipo de problemas enunciados enquadra-se no conjunto das limitações mais marcantes das técnicas clássicas de controlo e modelização, pelo que os sistemas difusos se apresentam como um complemento importante das metodologias convencionais.

No desenvolvimento de sistemas difusos, os conceitos de conjunto e lógica difusa são fundamentais. Assim, este capítulo começa por apresentar os princípios fundamentais de conjuntos difusos e lógica difusa. Na Secção 3.3 define-se a estrutura e aspectos essenciais de projecto de sistemas difusos, na óptica da modelização de sistemas. Finalmente, na Secção 3.4 é discutida a propriedade da aproximação universal, no contexto dos sistemas difusos utilizados nesta dissertação.

3.1. Introdução

O princípio da incompatibilidade de Lofti Zadeh [Zadeh, 1973], enunciado no capítulo introdutório, apresenta de forma concisa e profunda as limitações da modelização de sistemas com base nos primeiros princípios, propondo a utilização de mecanismos de processamento qualitativo da informação. Neste sentido, o mesmo autor sugere a construção de modelos ou controladores de sistemas com recurso a um conjunto de regras, expressas em linguagem natural, capazes de descrever qualitativamente a dinâmica de um dado sistema. Esse conjunto de regras constitui um *algoritmo difuso* [Brown e Harris, 1994]. Exemplificando, o controlo de um aquecedor eléctrico doméstico, poderá ser efectuado, de maneira intuitiva, por um conjunto de regras do tipo:

SE (*temperatura é baixa*) ENTÃO (*variação da potência é positiva alta*). (3.1)

A natureza desta representação tem subjacente o conceito de *lógica*, na medida em que o algoritmo difuso consistirá num mecanismo de inferência que, com base num conjunto de premissas, permitirá obter conclusões que se esperam válidas. No caso (3.1), sabe-se que se a temperatura for classificada como baixa, a acção a tomar será aumentar a potência.

Do exposto transparece uma questão natural: uma vez que a modelização e controlo de sistemas lidam com grandezas quantitativas (e.g., medidas de temperatura e potência) e que os

algoritmos difusos se caracterizam pela sua natureza qualitativa, como conjugar estas duas realidades? De facto, a implementação prática de algoritmos difusos requer que as expressões linguísticas do tipo "baixa" sejam quantificadas matematicamente. No entanto, no mundo real a classificação de objectos é por essência vaga e imprecisa. Deste modo, classificar um valor medido de temperatura de maneira binária (baixo ou não baixo) constitui um procedimento inadequado. Como tal, Zadeh [Zadeh, 1965] definiu o conceito de conjunto difuso como forma de superar a classificação dicotómica presente na teoria clássica dos conjuntos. Consequentemente, como resultado da aplicação dos conjuntos difusos aos mecanismos de inferência, o mesmo autor propôs a lógica difusa como generalização da lógica Aristotélica [Zadeh, 1968; Zadeh, 1973].

Do mesmo modo que os termos linguísticos presentes num algoritmo difuso devem ser quantificados por meio de conjuntos difusos, outras operações, tais como a intersecção, união e implicação difusas necessitam de ser definidas. Alguns dos esquemas utilizados são apresentados no decurso deste capítulo. A partir do momento em que os conjuntos difusos e os operadores estejam determinados, as relações expressas qualitativamente pelas regras do tipo (3.1) deixam de ser vagas, passando a constituir uma função não linear *determinística*. Obtém-se deste modo aquilo que se designa por *sistema difuso*: uma implementação específica, dependente do contexto de utilização, de um conjunto de regras qualitativas expressas por meio de um algoritmo difuso. A Figura 3.1 apresenta esquematicamente a distinção e interacção entre algoritmo difuso e sistema difuso.

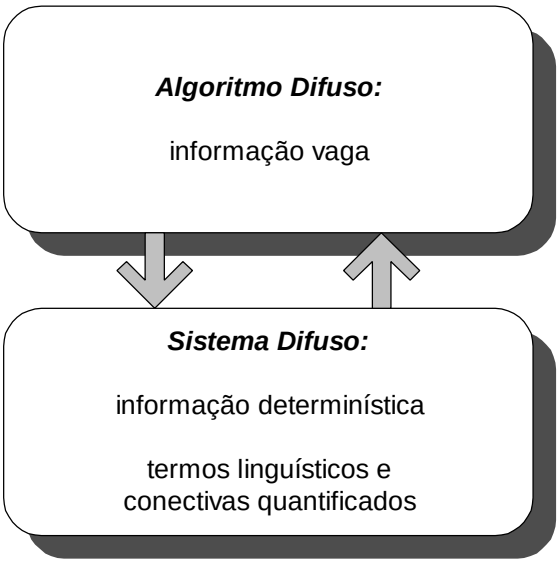


Figura 3.1. Caracterização de algoritmo difuso e sistema difuso.

Apesar do sucesso das aplicações iniciais da lógica difusa, o seu interesse esmoreceu passado algum tempo, em virtude de alguns ataques mais ferozes da área clássica do controlo, assim como de questões culturais, em virtude da associação de conotações negativas à palavra "difuso". No entanto, no Japão, o interesse pela investigação sobre a aplicação de sistemas difusos no tratamento dos problemas mais diversos cresceu significativamente a partir do final do decénio de setenta, altura em que o potencial da lógica difusa foi explorado em grande escala. Tal deveu-se, em grande medida, à simbiose entre a mentalidade e filosofia de trabalho japonesa e os princípios da lógica difusa. De facto, faz parte da cultura de trabalho daquele povo a produção pela criação rápida de protótipos e sua optimização posterior, o que é favorecido pela lógica difusa. Outro aspecto relevante relaciona-se com a filosofia de trabalho em equipa do povo japonês: todos os

elementos desejam conhecer o mecanismo de funcionamento do sistema a ser tratado. Esta mentalidade laboral favorece a utilização de sistemas difusos devido à sua clareza e facilidade de compreensão - *transparência*. Um outro ponto interessante prende-se com o facto de a palavra "difuso" não ter associada, na língua japonesa, qualquer conotação negativa, ao contrário do que acontece nas culturas ocidentais. Pelo contrário, no Japão, à etiqueta "*fuzzy-controlled*" estão associados parâmetros de modernismo e qualidade, bem como de simplicidade e agradabilidade na utilização. Em resultado dos aspectos simbióticos descritos, assim como do incentivo do governo japonês à investigação nesta área, a lógica difusa é utilizada actualmente naquele país em variadíssimas aplicações de controlo inteligente, processamento de dados, assim como em utensílios domésticos e de lazer. Uma das aplicações de maior sucesso consistiu na implementação de um sistema de controlo para o metropolitano de Sendai, composto por dezasseis estações [Oshima et al, 1988]. De facto, como resultado do desenvolvimento referido, o consumo de energia diminuiu 10%, a precisão nos pontos de paragem melhorou duas vezes e meia, o andamento do metropolitano tornou-se bastante suave, além de que o controlador difuso comete menos 70% dos erros de análise cometidos por operadores humanos na aceleração e travagem. Para além desta aplicação, muitas outras tiveram lugar, tanto em aplicações industriais de larga escala, como em pequenos utensílios do quotidiano, tais como controladores de máquinas de lavar, controladores para a focagem de máquinas fotográficas, sistemas de controlo para a indústria automóvel e optimização de processos químicos e biológicos.

Apesar dos muitos casos de sucesso ocorridos no Japão durante os anos oitenta, o nível de interesse manifestado pela comunidade científica Europeia e Americana foi reduzido. Tal deveu-se, possivelmente, à pouca receptividade por parte da comunidade científica ocidental, em virtude do reduzido amadurecimento dos aspectos de análise. Ao contrário, no Japão, a investigação orientou-se sobretudo para a aplicação, tendo-se relegado para segundo plano as questões de análise. No entanto, no início do decénio de noventa, o interesse pela investigação nesta área aumentou significativamente na Europa e Estados Unidos. Tal interesse resultou, em grande parte, do ultrapassar de alguns dos mitos e conotações negativas relacionados com a lógica difusa, em resultado do sucesso de um número significativo de aplicações, essencialmente na área do controlo. Por outro lado, muitas companhias passaram a interessar-se pela sua promoção, como forma de concorrência com as companhias japonesas. Actualmente, é lícito afirmar-se que a lógica difusa ganhou uma ampla aceitação na comunidade científica, ao ponto do estudo desta área do conhecimento, e sua aplicação à modelização e controlo de sistemas, fazer parte de um número significativo de programas educacionais das universidades em todo o mundo. A Figura 3.2 [Bezdek, 1993] apresenta, de forma sintética, a evolução da expectativa à volta da lógica difusa ao longo do tempo. Presentemente, o espectro de aplicação da teoria dos sistemas difusos atinge um número significativo de áreas, tais como reconhecimento de padrões, linguísticas, investigação operacional, redes neuronais artificiais e identificação e controlo de sistemas.

Nos últimos anos tem-se assistido a um interesse crescente pela investigação de mecanismos de aprendizagem em sistemas difusos, o que levou ao despontar da área científica dos sistemas neuro-difusos, a qual procura aproveitar as vantagens resultantes da combinação entre sistemas difusos e redes neuronais artificiais.

Como forma de enquadrar o estudo das tecnologias neuro-difusas, este capítulo e o seguinte debruçar-se-ão, respectivamente, sobre os princípios fundamentais de sistemas difusos e de redes neuronais.

Assim, neste capítulo será apresentada uma síntese introdutória[16] dos conceitos fundamentais de sistemas difusos, necessários à exposição do trabalho presente nos capítulos posteriores.

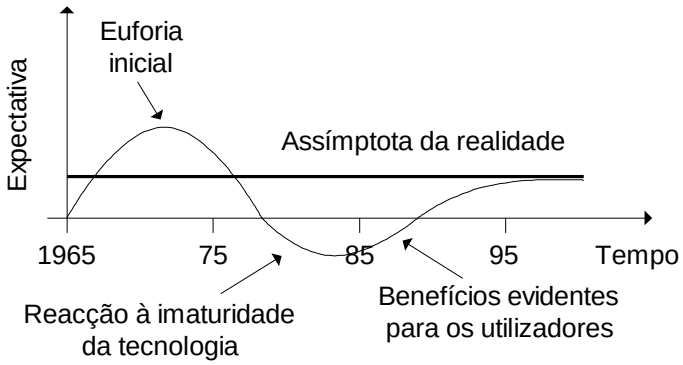


Figura 3.2. Evolução das metodologias difusas.

3.2. Conjuntos Difusos e Lógica Difusa

Com base no problema do controlo de um aquecedor eléctrico doméstico (3.1), suponhamos que, num dado instante, a temperatura do ar apresenta o valor de 14ºC. Segundo a regra (3.1), é necessário determinar o valor lógico da proposição "temperatura está baixa", i.e., é necessário verificar se a proposição é verdadeira ou falsa. Isto implica a definição de um conjunto *A*, cujos elementos são os valores de temperatura que satisfazem o conceito de "baixa". Por exemplo, poder-se-á definir esse conjunto como sendo formado pelos valores de temperatura inferiores a 15ºC. Antes, porém, de definir o conjunto referido, é necessário determinar o campo de referência da variável considerada. Por outras palavras, é necessário definir o seu *domínio* ou *universo de discurso*, X. Suponhamos, então, que o termómetro utilizado funciona na gama $[X_{\min}; X_{\max}] = [0^{o}; 40^{o}]$ (3.2):

$$X = \{x \in \Re : x \geq 0 \;\; e \;\; x \leq 40\} \tag{3.2}$$

onde x representa uma variável numérica à qual estão associados os valores da temperatura. Deste modo, o conjunto A é definido do seguinte modo (3.3):

$$A = \{x \in X \; : x < 15\} \tag{3.3}$$

O mesmo conjunto é representado graficamente na Figura 3.3, adaptada de [von Altrock, 1995]. Assim, o valor de pertença de um dado elemento x no conjunto A poderá ser determinado por (3.4):

$$m_A(x) = \begin{cases} 1 \;, x < 15 \\ 0 \;, x \geq 15 \end{cases} \quad , x \in X \tag{3.4}$$

[16] O texto apresentado não pretende constituir uma compilação dos diversos aspectos associados à lógica difusa. Antes, pretende-se apenas introduzir alguns conceitos chave, necessários aos capítulos subsequentes. Para uma exposição mais detalha, [Driankov et al, 1993], [Harris et al, 1993] ou [Ross, 1995] constituem boas referências.

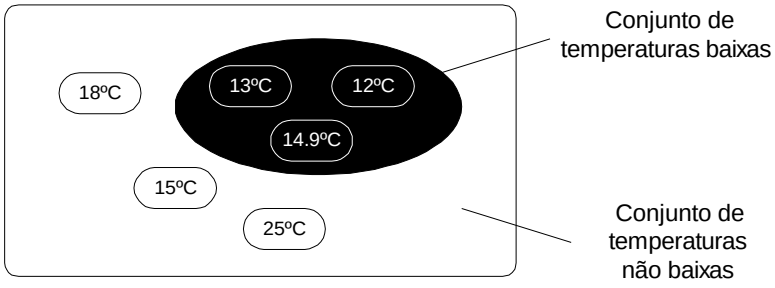


Figura 3.3. O conjunto *A*, segundo a teoria clássica dos conjuntos.

A expressão (3.4) designa-se por *função de pertença*. Esta visão dicotómica, do tipo verdadeiro ou falso, ou 1/0, levanta algumas dificuldades. Nomeadamente, uma temperatura de 14.9ºC pertence ao conjunto *A*, i.e., é classificada como baixa, ao passo que a temperatura 15ºC já não o é. E, contudo, no contexto do problema de aquecimento descrito, tais valores são interpretados de maneira idêntica por um ser humano. De facto, no mundo real, a grande maioria das classes de objectos encontradas não são classificadas de forma binária. Como consequência desta limitação dos conjuntos clássicos - para os quais os elementos de um dado universo são classificados como pertencendo ou não pertencendo ao conjunto - Zadeh introduziu, formalmente, em 1965, o conceito de *conjunto difuso*[17] [Zadeh, 1965]. Segundo o autor, os tipos de conceitos descritos são inerentemente vagos. Deste modo, a pertença dos elementos do universo de discurso a um qualquer conjunto é definida por um grau, não binário como em (3.3), mas sim num intervalo [0; 1]. Deste modo, a função de pertença (3.4) é generalizada para uma função do tipo (3.5):

$$m_{\tilde{A}} : X \rightarrow [0;1] \tag{3.5}$$

em que $\tilde{A}$ denota o conjunto difuso correspondente ao conjunto clássico *A*. Assim, graficamente, o conjunto difuso $\tilde{A}$ poderá ser representado como na Figura 3.4, adaptada de [von Altrock, 1995].

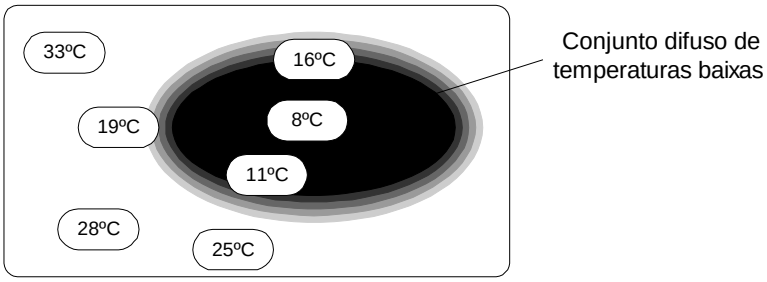


Figura 3.4. O conjunto $\tilde{A}$, segundo a teoria dos conjuntos difusos.

De forma genérica, $\tilde{A}$ é definido, segundo a teoria dos conjuntos difusos, como o conjunto difuso[18] de elementos *x* do universo de discurso X e respectivos graus de pertença (3.6):

---

[17] Na verdade, a ideia de conjunto difuso tem as suas raízes na segunda metade do século XIX (*vide* [Höhle e Neff Stout, 1991]). No entanto, o termo *conjunto difuso*, bem como a sua definição formal, foram introduzidos por Zadeh em 1965.

[18] Na exposição que segue, utilizar-se-á o termo conjunto como forma de denotar conjunto clássico, utilizando-se o termo conjunto difuso, explicitamente, sempre que de tal se trate.

$$\tilde{A} = \{(x, m_{\tilde{A}}(x)) : x \in X\} \tag{3.6}$$

Na representação matemática de conjuntos difusos é comum utilizar-se uma função de pertença geral que define, para cada elemento x do universo de discurso X, o seu grau de pertença $m_{\tilde{A}}(x)$, relativamente ao conjunto difuso considerado (3.5). Assim sendo, as funções de pertença mais frequentes são as do tipo triangular, trapezoidal e em forma de sino.

Relativamente às funções de pertença em forma de sino, uma das mais usuais nesta classe é a função Gaussiana, Ω: $X \rightarrow$[0; 1]. Esta função, que será utilizada frequentemente ao longo deste trabalho, é completamente definida por dois parâmetros: o seu centro, c, e o seu desvio padrão, s. (Figura 3.5).

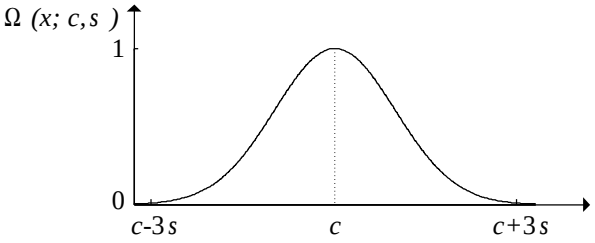


Figura 3.5. Função de pertença Gaussiana.

Analiticamente, vem (3.7):

$$\Omega(x; c, s) = e^{-\frac{(x-c)^2}{2s^2}}, \forall_{x \in X} \tag{3.7}$$

O contradomínio da função Gaussiana não contém o valor 0. De facto, o seu suporte não é compacto[19]. No entanto, é usual assumir-se que a função referida se anula fora do intervalo [c-3s; c+3s], tal como é apresentado na figura anterior.

Pela composição de duas funções Gaussianas, obtém-se a função de pertença Gaussiana generalizada, Ω_g: $X \rightarrow$[0; 1], que se caracteriza pela possibilidade de poder conter um *planalto* e de ser assimétrica. Deste modo, na sua definição recorre-se a quatro parâmetros: c_L e s_L , para a Gaussiana de menor centro, e c_R e s_R, para a componente da direita (Figura 3.6).

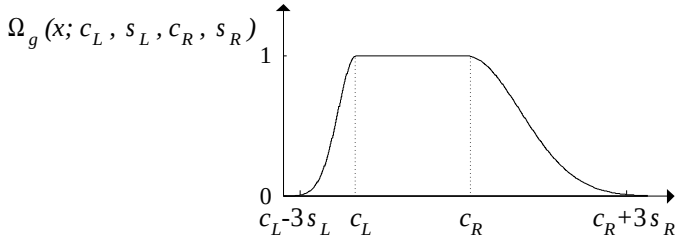


Figura 3.6. Função de pertença Gaussiana generalizada.

[19] O *suporte* do conjunto difuso $\tilde{A}$ é definido pelo conjunto de elementos com grau de pertença não nulo. Um suporte diz-se *compacto* se se tratar de um subconjunto estrito do universo de discurso, i.e., *suporte*($\tilde{A}$) $\grave{I}$ X.

Analiticamente, tem-se (3.8):

$$\Omega_g(x;c_L,s_L,c_R,s_R)=\begin{cases} e^{-\frac{(x-c_L)^2}{2s_L^2}} & ,x<c_L \\ 1 & ,c_L\le x\le c_R \\ e^{-\frac{(x-c_R)^2}{2s_R^2}} & ,x>c_R \end{cases} \quad ,\forall_{x\in X} \tag{3.8}$$

A vantagem fundamental de se utilizarem funções de pertença Gaussianas generalizadas reside na sua maior flexibilidade. Assim, a precisão do modelo difuso poderá ser maior, além de o sistema ser, potencialmente, mais interpretável. Este último aspecto resulta do facto de funções assimétricas permitirem uma sobreposição mais flexível das funções de pertença (Figura 3.7), o que poderá fazer diminuir o excesso de sobreposição em certas áreas do domínio de cada variável. Por outro lado, a sua maior desvantagem prende-se com o facto do número de parâmetros a ajustar duplicar, o que poderá potenciar a ocorrência de situações de sobreajustamento.

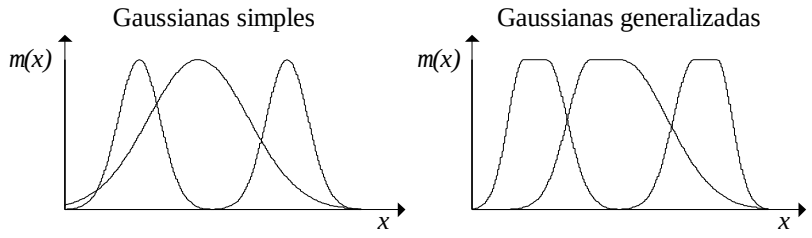


Figura 3.7. Caracterização da sobreposição em funções Gaussianas simples e generalizadas.

3.2.1. Operações Básicas sobre Conjuntos Difusos

Da teoria clássica dos conjuntos derivam algumas noções cuja generalização para o contexto dos conjuntos difusos é efectuada de forma imediata. Nomeadamente, a noção de igualdade é estabelecida para conjuntos difusos de forma natural.

Assim, considerem-se dois conjuntos difusos $\tilde{A}$ e $\tilde{B}$, definidos num universo de discurso X.

Identidade

Os dois conjuntos referidos dizem-se *iguais* ($\tilde{A}=\tilde{B}$) se cada um dos elementos do universo de discurso apresentar o mesmo grau de pertença em ambos os conjuntos. Formalmente, tem-se (3.9):

$$\forall_{x\in X} : m_{\tilde{A}}(x)=m_{\tilde{B}}(x) \tag{3.9}$$

Em contraste com o conceito de igualdade enunciado anteriormente, operações como a união, a intersecção e o complemento não são estendidas para a teoria dos conjuntos difusos de forma inequívoca. Este facto advém das características de pertença contínuas dos conjuntos difusos. Assim, as operações referidas são representadas, segundo a teoria dos conjuntos difusos, como normas e co-normas triangulares.

Intersecção

Uma *norma triangular*, ou *norma-T*, $\hat{*}$ (3.10), constitui o mecanismo básico de representação da operação de *intersecção* difusa ($\cap$) (3.11):

$$\begin{aligned} \hat{*}: [0;1]\times[0;1] &\to [0;1] \\ (a,b) &\to a \,\hat{*}\, b \end{aligned} \tag{3.10}$$

$$\forall_{x\in X}: m_{\tilde{A}\cap\tilde{B}}(x) = m_{\tilde{A}}(x)\,\hat{*}\,m_{\tilde{B}}(x) \tag{3.11}$$

Uma norma-T denota uma classe de funções binárias com as propriedades seguintes (3.12):

$$\begin{aligned} &T-1: && a\,\hat{*}\,b = b\,\hat{*}\,a \\ &T-2: && (a\,\hat{*}\,b)\,\hat{*}\,c = a\,\hat{*}\,(b\,\hat{*}\,c) \\ &T-3: && (a\le c)\wedge(b\le d)\Rightarrow a\,\hat{*}\,b \le c\,\hat{*}\,d \\ &T-4: && a\,\hat{*}\,1 = a \end{aligned} \quad ,\forall_{a,b,c,d\in[0;1]} \tag{3.12}$$

Na definição da operação de intersecção difusa, os operadores *mínimo* (3.13) e *produto algébrico* (3.14) constituem as opções mais comuns. A verificação de que ambos se inserem na classe das normas triangulares é trivial.

$$\forall_{x\in X}: m_{\tilde{A}\cap\tilde{B}}(x) = \min\left(m_{\tilde{A}}(x), m_{\tilde{B}}(x)\right) \tag{3.13}$$

$$\forall_{x\in X}: m_{\tilde{A}\cap\tilde{B}}(x) = m_{\tilde{A}}(x)\cdot m_{\tilde{B}}(x) \tag{3.14}$$

União

No que respeita à operação de *união* difusa ($\cup$), a sua representação é efectuada por uma *co-norma triangular*, ou *norma-S*, $\check{*}$. Formalmente, tem-se (3.15) e (3.16):

$$\begin{aligned} \check{*}: [0;1]\times[0;1] &\to [0;1] \\ (a,b) &\to a \,\check{*}\, b \end{aligned} \tag{3.15}$$

$$\forall_{x\in X}: m_{\tilde{A}\cup\tilde{B}}(x) = m_{\tilde{A}}(x)\,\check{*}\,m_{\tilde{B}}(x) \tag{3.16}$$

Uma norma-S é caracterizada pelas propriedades seguintes (3.17):

$$\begin{aligned} &S-1: && a\,\check{*}\,b = b\,\check{*}\,a \\ &S-2: && (a\,\check{*}\,b)\,\check{*}\,c = a\,\check{*}\,(b\,\check{*}\,c) \\ &S-3: && (a\le c)\wedge(b\le d)\Rightarrow a\,\check{*}\,b \le c\,\check{*}\,d \\ &S-4: && a\,\check{*}\,0 = a \end{aligned} \quad ,\forall_{a,b,c,d\in[0;1]} \tag{3.17}$$

Os operadores *máximo* (3.18), *adição algébrica* (3.19) e *adição limitada* (3.20) constituem representações habituais da operação de união difusa. A verificação de que ambos os casos representam co-normas triangulares é directa.

$$\forall_{x\in X}: m_{\tilde{A}\cup\tilde{B}}(x) = \max\left(m_{\tilde{A}}(x), m_{\tilde{B}}(x)\right) \tag{3.18}$$

$$\forall_{x\in X}: m_{\tilde{A}\cup\tilde{B}}(x) = m_{\tilde{A}}(x) + m_{\tilde{B}}(x) - m_{\tilde{A}}(x)\cdot m_{\tilde{B}}(x) \tag{3.19}$$

$$\forall_{x\in X}\ : m_{\tilde{A}\cup\tilde{B}}(x) = \min\left(1, m_{\tilde{A}}(x) + m_{\tilde{B}}(x)\right) \tag{3.20}$$

Complemento ou negação

Quanto ao *complemento* de um conjunto difuso $\tilde{A}$, $\overline{\tilde{A}}$, a operação referida define-se por uma *norma-c*, tal como aparece, formalmente, em (3.21) e (3.22):

$$\begin{aligned} c : [0;1] &\rightarrow [0;1] \\ a &\rightarrow c(a) \end{aligned} \tag{3.21}$$

$$\forall_{x\in X}\ : m_{\overline{\tilde{A}}}(x) = c\left(m_{\tilde{A}}(x)\right) \tag{3.22}$$

Uma norma-c deve satisfazer os critérios seguintes (3.23):

$$\begin{aligned} c-1: &\quad c(0) = 1 \\ c-2: &\quad a < b \Rightarrow c(a) > c(b) \qquad ,\forall_{a,b\in[0;1]} \\ c-3: &\quad c(c(a)) = a \end{aligned} \tag{3.23}$$

Tipicamente, o operador complemento é definido como (3.24):

$$\forall_{x\in X}\ : m_{\overline{\tilde{A}}}(x) = 1 - m_{\tilde{A}}(x) \tag{3.24}$$

Usualmente, os operadores mínimo e máximo, referidos ao longo desta secção, são designados por *operadores de truncatura*, enquanto que a soma e o produto se designam por *operadores algébricos* [Harris et al, 1993]. A selecção da classe a utilizar reflecte-se, naturalmente, no comportamento do sistema difuso, tal como será analisado posteriormente (Secção 5.3.2).

3.2.2. Similaridade entre Conjuntos Difusos

Em termos genéricos, dois conjuntos difusos dizem-se *similares* se o seu grau de semelhança for elevado, i.e., se as suas funções de pertença tomarem valores aproximados em todos os pontos do domínio. Em termos puramente formais, dois conjuntos difusos $\tilde{A}$ e $\tilde{B}$ são similares no caso de as suas funções de pertença se intersectarem em qualquer ponto do domínio X (3.25):

$$\exists_{x\in X}\ : \min\left(m_{\tilde{A}}(x), m_{\tilde{B}}(x)\right) \neq 0 \tag{3.25}$$

Caso contrário, os conjuntos difusos dizem-se *não similares*. A expressão (3.25) conduz à definição de *similaridade* entre dois conjuntos difusos como o grau de igualdade entre as suas funções de pertença, $m_{\tilde{A}}(x)$ e $m_{\tilde{B}}(x)$. Deste modo, o conceito de similaridade é eminentemente difuso: a similaridade entre dois conjuntos difusos é quantificada em termos de um valor de verdade, *s*, definido no intervalo [0; 1].

Do exposto, transparece a necessidade de se definirem medidas de similaridade. Qualquer que seja a metodologia utilizada, as propriedades abaixo enunciadas devem verificar-se [Setnes, 1995]:

- a similaridade deve ser medida entre instanciações e não tipos de funções de pertença;
- as funções de pertença devem estar definidas no mesmo domínio;
- a posição das funções de pertença no domínio deve ser mais importante do que as suas

formas, e.g., uma função triangular e outra Gaussiana com suportes idênticos são mais semelhantes do que duas funções triangulares com suportes claramente distintos;
- da medida de similaridade entre dois conjuntos difusos $\tilde{A}$ e $\tilde{B}$, $s(\tilde{A},\tilde{B})$ deve resultar um valor $s \in [0; 1]$, correspondente ao grau de igualdade entre os conjuntos em questão (s=1 indica que os conjuntos difusos são iguais);
- a medida de similaridade não deve ser influenciada pela alteração da escala do domínio.

Assim, existem fundamentalmente duas classes de métodos para medida de similaridade entre conjuntos difusos: os métodos geométricos e os métodos baseados na teoria dos conjuntos.

Métodos geométricos

Os *métodos geométricos* baseiam-se, em geral, na utilização de medidas de distância entre conjuntos difusos sobre o eixo das abcissas. Exemplificando, a medida de similaridade entre dois conjuntos difusos através de uma das medidas de distância definidas na métrica de Minkowski é dada por (3.26):

$$s(\tilde{A},\tilde{B}) = \left(\sum_{i=1}^{k} \left| m_{\tilde{A}}(x_i) - m_{\tilde{B}}(x_i) \right|^r \right)^{\frac{1}{r}} \quad , r \geq 1 \tag{3.26}$$

onde k designa o número de pontos considerados num universo de discurso discreto. Neste grupo incluem-se, por exemplo, a distância Euclidiana (r=2) ou a distância City-Block (r=1).

Para além da métrica de Minkowski, outras métricas estão incluídas na classe dos métodos geométricos, tais como as métricas de Hausdorff ou de Goetschel e Voxman. Estes e outros métodos são analisados em detalhe em [Setnes, 1995].

Métodos baseados na teoria dos conjuntos

Os *métodos baseados na teoria dos conjuntos* apoiam-se nas operações sobre conjuntos, e.g., intersecção e união. Desses métodos, merece particular atenção a medida S_1, segundo a qual a similaridade entre dois conjuntos difusos é dada pelo quociente entre a área da sua intersecção e a área da sua união (3.27):

$$S_1(\tilde{A},\tilde{B}) = \frac{\left\| \tilde{A} \cap \tilde{B} \right\|}{\left\| \tilde{A} \cup \tilde{B} \right\|} \tag{3.27}$$

onde as operações de intersecção e união difusas são implementadas pelos operadores *mínimo* e *máximo*. Em (3.27), $\| \tilde{A} \|$ representa a *cardinalidade relativa* do conjunto difuso $\tilde{A}$, definida como (3.28):

$$\left\| \tilde{A} \right\| = \frac{\left| \tilde{A} \right|}{|X|} \tag{3.28}$$

em que X denota o domínio do conjunto difuso $\tilde{A}$, sendo $| \tilde{A} |$ a *cardinalidade escalar* (3.29):

$$\left| \tilde{A} \right| = \sum_{x \in X} m_{\tilde{A}}(x) \tag{3.29}$$

Genericamente, Setnes [Setnes, 1995] conclui que as medidas de similaridade baseadas na teoria dos conjuntos são mais adequadas quando se tem por objectivo simplificar uma base de regras (Secção 3.3.2), tal como acontece no contexto desta dissertação. Esta problemática será abordada na Secção 5.4.

Assim, de entre a classe de métodos em questão, a medida S_1 (3.27) apresenta-se como a mais satisfatória, em virtude de satisfazer os requisitos necessários a uma medida de similaridade expressos anteriormente, além de ser intuitiva e computacionalmente menos exigente que outras. Deste modo, será esta a medida de similaridade utilizada nos capítulos subsequentes.

3.2.3. Lógica Difusa e Raciocínio Aproximado

Em termos genéricos, a lógica baseia-se no estudo da veracidade de proposições, no sentido de se inferirem conclusões com base em premissas. Voltando ao exemplo (3.1), sabe-se que no caso da temperatura ser baixa a variação da potência deve ser alta. Porém, os conceitos *baixa* e *alta* são, inerentemente difusos, tal como se abordou anteriormente. Por conseguinte, surge a noção de *lógica difusa* [Zadeh, 1968; Zadeh, 1973] como um método de inferir conclusões, com base em expressões difusas.

Assim, o objectivo último da lógica difusa consiste em formar a base teórica onde assenta o raciocínio sobre proposições imprecisas ou difusas. Este raciocínio é designado por *raciocínio aproximado* [Zadeh, 1973]. Deste modo, no raciocínio aproximado, a conclusão de um conjunto de proposições difusas, definidas numa regra condicional difusa do tipo (3.1), i.e., uma expressão do tipo *se-então*, depende do significado associado a essas proposições, significado esse determinado com base nos conjuntos difusos definidos.

O conceito de *variável linguística* é fundamental para a representação de conhecimento no raciocínio aproximado [Zadeh, 1973]. De acordo com Zadeh, uma variável linguística é uma variável cujos valores são palavras ou expressões numa linguagem natural ou artificial. Por exemplo, considerando o exemplo introduzido anteriormente, *temperatura* será uma variável linguística, uma vez que toma valores do tipo “baixa” ou “alta”.

3.3. Estrutura e Projecto de Sistemas Difusos

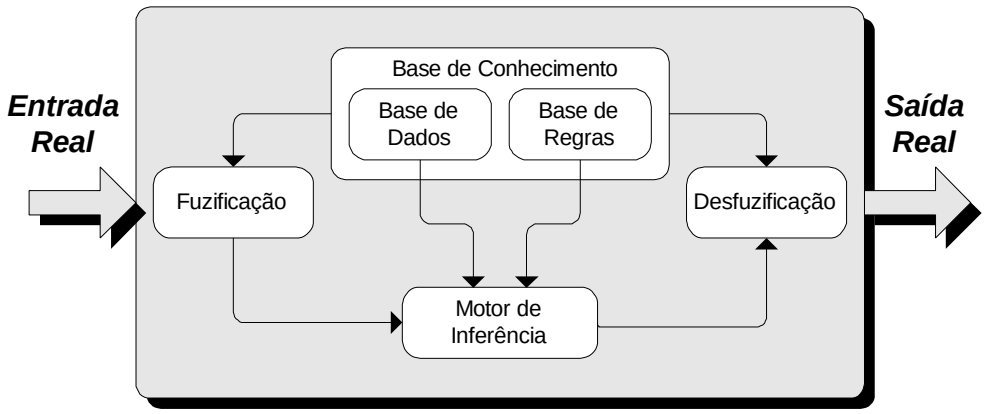


Figura 3.8. Diagrama de um sistema difuso.

Tal como se referiu no início do capítulo presente, um sistema difuso pode ser visto como uma função não linear e determinística que mapeia e quantifica as relações expressas qualitativamente por um algoritmo difuso. Tipicamente, os sistemas difusos são utilizados em

tarefas de modelização e controlo, constituindo, juntamente com o algoritmo associado, modelos ou controladores de natureza qualitativa.

A estrutura básica de um sistema difuso é apresentada na Figura 3.8, de acordo com Mamdani e Assilian [Mamdani e Assilian, 1975][20].

Tal como se pode verificar, qualquer sistema difuso é composto por quatro elementos fundamentais: um *módulo de fuzificação*, uma *base de conhecimento*, um *motor de inferência* e um *módulo de desfuzificação*[21]. A interligação desses elementos permite definir uma determinada função não linear *f: x® y*. Na Secção 3.4, verificar-se-á que essa função constitui um aproximador universal para uma vasta classe de estruturas difusas.

A definição das propriedades de cada um dos módulos enunciados constitui o aspecto fundamental do projecto de sistemas difusos. Seguidamente serão apresentadas as propriedades e parâmetros de projecto para cada um dos elementos enunciados, no contexto em que se insere esta dissertação, i.e., incidindo sobre questões de modelização [Lee, 1990a; Lee, 1990b].

3.3.1. Módulo de Fuzificação

A *fuzificação* é o processo de conversão de entradas numéricas em conjuntos difusos, definidos num dado universo de discurso. Esta operação é fundamental, uma vez que na generalidade das aplicações da lógica difusa os dados observados são numéricos. Como tal, dado que a manipulação de valores num sistema difuso é baseada na teoria dos conjuntos difusos, é necessário, em primeiro lugar, fuzificar todos os dados numéricos. Simbolicamente, o dado numérico x^* é convertido no conjunto difuso $\tilde{X}^*$, por meio de um fuzificador (3.30).

$$\tilde{X}^* = fuzificador(x^*) \tag{3.30}$$

A estratégia de fuzificação a utilizar é condicionada pelo tipo de inferência utilizado. Assim, existem, fundamentalmente, duas regras de inferência: a *regra de inferência composicional* e a *modus ponens generalizada*. Na primeira, a ligação entre proposições difusas é efectuada com base numa relação difusa. Deste modo, a cada uma das regras que compõem um dado sistema difuso, está associada uma relação difusa. Assim, é usual combinar as várias relações numa única, designando-se, por conseguinte, o tipo de inferência em causa por *inferência baseada na combinação de regras*. Na modus ponens generalizada, utiliza-se uma regra condicional difusa que representa, implicitamente, uma relação difusa. Nesta situação, não são utilizadas relações difusas, sendo cada regra accionada separadamente. Deste modo, o tipo de inferência descrito designa-se por *inferência baseada em regras individuais*.

Considere-se, então, um sistema descrito por g regras do tipo (3.31):

$$\text{SE } X \text{ é } LX^{(k)} \text{ ENTÃO } Y \text{ é } LY^{(k)}, \quad k=1,2,\ldots,g \tag{3.31}$$

No caso da inferência baseada na composição, a entrada numérica x^* é fuzificada pelo

[20] A estrutura apresentada foi sugerida no contexto do controlo difuso. No entanto, a sua interpretação poderá ser conduzida de forma genérica, englobando tanto problemas de controlo como de modelização difusa.

[21] À data presente, não se conhece qualquer tradução de aceitação geral para os termos associados à teoria dos sistemas difusos. Assim, utilizar-se-ão ao longo deste trabalho os anglicismos *fuzificação* e *desfuzificação*, que por vezes são traduzidos como *difusificação* e *desdifuzificação*, respectivamente.

fuzificador *singular* (*singleton* [Lee, 1990a]), obtendo-se o conjunto difuso $\tilde{X}^*$, definido pela função de pertença (3.32):

$$\forall_{x \in X} : m_{\tilde{X}^*}(x) = \begin{cases} 1 \, , x = x^* \\ 0 \, , x \neq x^* \end{cases} \tag{3.32}$$

No caso da inferência baseada em regras individuais, a fuzificação é realizada de maneira distinta. Neste caso, a representação difusa de x^* é dada pelo seu grau de pertença, $m_{L\tilde{X}^{(k)}}(x^*)$, no conjunto difuso $L\tilde{X}^{(k)}$. Nesta dissertação, a abordagem baseada no accionamento individual de regras é preferida, pelo que a fuzificação é levada a cabo com base na última metodologia.

Para além da operação de fuzificação, este módulo é também responsável pela transformação da escala, caso seja utilizado um universo de discurso normalizado, o que não se verifica neste trabalho.

3.3.2. Base de Regras

A modelização difusa de um sistema requer a sua representação com base num conjunto de regras susceptíveis de o descreverem. Esta colecção de regras constitui o que se designa por *base de regras*. Tais regras são da seguinte forma (3.33):

$$\text{SE } \underbrace{(\textit{estado do sistema})}_{\text{antecedente}} \text{ ENTÃO } \underbrace{(\textit{saída do sistema})}_{\text{consequente}} \tag{3.33}$$

A primeira parte da regra, a premissa, é habitualmente designada por *antecedente* e descreve o estado do sistema em termos da composição de proposições difusas através de conectivas de conjunção ou disjunção. A segunda parte, a conclusão, designa-se por *consequente*, e representa as saídas do sistema em resultado das condições da premissa.

A construção de uma base de regras para um modelo envolve, essencialmente, a escolha das variáveis linguísticas de entrada e saída do modelo, a selecção do formato das regras condicionais, a selecção dos termos associados a cada uma das variáveis linguísticas e a síntese do conjunto de regras.

Selecção das variáveis linguísticas

A selecção das variáveis linguísticas depende dos factores de análise inerentes à identificação de sistemas. Basicamente, essa selecção é efectuada com base no conhecimento disponível sobre o sistema, bem como na possibilidade de aquisição das variáveis pretendidas, de acordo com os aspectos expostos na Secção 2.4.

Selecção do formato das regras condicionais

De acordo com o formato do consequente, definem-se dois tipos de regras condicionais difusas: regras linguísticas e regras de Takagi-Sugeno.

As *regras linguísticas* caracterizam-se pelo facto de o consequente ser, tal como o antecedente, um conjunto difuso. Nesta situação, as regras condicionais difusas são do tipo (3.34):

$$\text{SE } (X \textit{ é } LX) \text{ ENTÃO } (Y \textit{ é } LY) \tag{3.34}$$

onde aos termos linguísticos *LX* e *LY* estão associados os conjuntos difusos $L\tilde{X}$ e $L\tilde{Y}$.

Por outro lado, nas *regras de Takagi-Sugeno* [Takagi e Sugeno, 1985] só os antecedentes

têm associados conjuntos difusos. As variáveis dos consequentes são definidas como uma função dos antecedentes, tal como se segue (3.35):

$$\text{SE } (X_1 \text{ é } LX1) \; E \; (X_2 \text{ é } LX2) \; E \ldots E \; (X_m \text{ é } LXm) \text{ ENTÃO } y=f(x_1, x_2 ,\ldots ,x_m) \tag{3.35}$$

Aqui, x_1, x_2 ,... ,x_m representam os valores numéricos associados a cada uma das variáveis linguísticas do antecedente X_1, X_2, …, X_m. Neste tipo de regras, o consequente constitui uma variável numérica, cujo resultado é obtido como uma função f dos valores numéricos dos antecedentes. Habitualmente, f é uma função polinomial de ordem 0 ou 1, designando-se o sistema difuso por sistema de Takagi-Sugeno de ordem 0 ou de ordem 1, respectivamente. Para o caso de se tratar de um sistema de ordem 1, vem (3.36):

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_m x_m = b_0 + \sum_{i=1}^{m} b_i x_i \qquad , b_i \in \Re,\, i = 1,2,\cdots,m \tag{3.36}$$

Em sistemas de ordem 0, a expressão (3.36) reduz-se a $y=b_0$.

Resultados apresentados na literatura [Sousa et al, 1997; Chiu, 1994; Jang, 1993] motivam a utilização de sistemas de Takagi-Sugeno de ordem 1, uma vez que o número de regras necessárias à identificação de sistemas, com uma capacidade de representação satisfatória, é inferior. Alternativamente, com o mesmo número de regras (ou até um número inferior) poderão alcançar-se melhores resultados. Este aspecto resulta, fundamentalmente, do facto de tais modelos desempenharem o papel de comutadores suaves entre modelos lineares locais, em virtude da natureza linear dos consequentes. Esta estratégia possibilita um ganho em precisão não alcançável pelos modelos de ordem 0 ou linguísticos, dada a natureza essencialmente interpolativa destes. A sua principal limitação advém de que, dado que os consequentes não são representados por conjuntos difusos, dificilmente se obtém uma representação linguística para modelos deste tipo. Estes e outros aspectos serão analisados posteriormente.

Síntese de regras

A síntese do conjunto de regras constitui, porventura, o parâmetro mais importante deste módulo. Basicamente, a selecção das regras a incluir no modelo de um sistema pode ser efectuada de duas formas: *manual* e *automática*.

Na abordagem manual, as regras são obtidas com base no conhecimento e experiência de peritos, relativamente ao sistema em causa. Esta derivação é puramente qualitativa e apresenta desvantagens decorrentes da subjectividade humana, i.e., peritos diferentes poderão sugerir regras diferentes, em forma e em número. Além deste factor, não é provável que um perito esteja habilitado a quantificar com precisão as grandezas qualitativas por ele expressas. Deste modo, surge a abordagem automática, a qual requer a recolha de dados de funcionamento do sistema. Da análise automática desses dados, resultam relações entre as variáveis do sistema, expressas por um conjunto de regras condicionais. Os métodos de agrupamento de classes[22] constituem um dos mecanismos de selecção de regras mais comuns e serão discutidos posteriormente.

Selecção de termos linguísticos

Quanto à selecção dos termos linguísticos associados a cada variável, na modelização

[22] *Clustering*, em terminologia inglesa.

baseada em dados tal escolha não é, em geral, efectuada previamente. Nesta situação, tanto a definição da base de regras como a determinação dos conjuntos difusos, presentes quer no antecedente quer no consequente, devem ser efectuadas automaticamente. Deste modo, a atribuição de termos linguísticos é efectuada *a posteriori,* após a definição e depuramento da base de regras e da base de dados, descrita no ponto seguinte.

De qualquer modo, os valores linguísticos a utilizar podem ser expressões do tipo "pequeno", "médio" ou "grande", valores esses que podem ser modificados em intensidade, obtendo-se termos como "muito pequeno" ou "muito grande".

Em relação ao número, Valente de Oliveira [Valente de Oliveira, 1995] sugere um valor entre 5 e 9 termos linguísticos, tipicamente 7. Tal heurística deve-se a que, habitualmente, um número inferior resulta numa precisão reduzida do modelo. Por outro lado, um número muito elevado apresenta dificuldades em termos de memorização por parte de um ser humano. Ainda em relação ao número de termos linguísticos, esse valor é geralmente ímpar, dado que as variáveis linguísticas se definem frequentemente com um termo médio, entre dois extremos.

3.3.3. Base de Dados

A função principal da *base de dados* é armazenar e fornecer a informação necessária ao funcionamento adequado dos módulos de fuzificação, base de regras e desfuzificação. Esta informação inclui a definição do tipo de universo de discurso utilizado, contínuo ou discreto. No último caso, é ainda necessário definir os níveis de quantização. Para além da natureza contínua ou discreta, o universo de discurso poderá ser normalizado. Neste caso, é necessário armazenar tanto os domínios físicos das variáveis do sistema, como os seus correspondentes normalizados e respectivos factores de escala. Uma vez que neste trabalho se faz uso unicamente de domínios contínuos não normalizados, os aspectos relacionados com a discretização e normalização não são abordados.

Selecção de funções de pertença

As funções de pertença constituem uma representação atractiva de conjuntos difusos, em virtude da sua descrição funcional paramétrica. Usualmente, são seleccionadas funções triangulares, trapezoidais ou Gaussianas. De todas, as triangulares são as que apresentam maiores vantagens a nível de eficiência computacional, pelo que a sua utilização é predominante em tarefas de controlo. No entanto, em problemas de modelização, o sistema difuso deverá aproximar ao máximo o processo a modelizar, o qual é muitas vezes não linear. Como tal, é frequente recorrer-se, neste contexto, a funções de pertença com um grau de não linearidade maior, pelo que se utilizam funções em forma de sino, como as Gaussianas. Dado o enquadramento da dissertação presente, são estas as funções utilizadas.

Após a escolha da forma das funções de pertença, é necessário distribuí-las pelo universo de discurso e ajustar os seus parâmetros. No caso em que essas tarefas sejam realizadas heuristicamente, é necessário ter em consideração aspectos como o seu grau de sobreposição, a simetria e a largura. No caso presente, a distribuição é efectuada automaticamente por meio de algoritmos de optimização (Secção 5.3). No entanto, após a distribuição das funções de pertença, é, por vezes, importante que o seu grau de sobreposição seja adequado, de forma a que se verifiquem eventuais requisitos de interpretabilidade.

3.3.4. Motor de Inferência

A função do motor de inferência é determinar o valor difuso de saída, com base nos parâmetros estabelecidos nos módulos de fuzificação e base de conhecimento. Neste sentido, existem duas abordagens empregues no mecanismo de inferência de um sistema difuso: a inferência baseada na composição e a inferência baseada em regras individuais, referidas anteriormente.

Assim, o projecto de um motor de inferência envolve essencialmente a selecção das conectivas difusas, a escolha da representação de uma única regra e de um conjunto de regras e a escolha de um método de inferência.

Selecção de conectivas difusas

Na implementação de um sistema difuso, as operações de intersecção e união, assim como de negação difusa devem ser estabelecidas. Assim, tal como foi abordado na Secção 3.2.1, a intersecção difusa é definida por meio de uma norma-T, sendo a união e a negação definidas respectivamente, por normas-S e normas-c. A selecção desses operadores envolve, fundamentalmente, a escolha entre operadores algébricos e operadores de truncatura. Habitualmente os primeiros são preferidos, uma vez que originam modelos suaves, em consequência da sua continuidade [Harris et al, 1993].

Representação de um conjunto de regras

Existem duas abordagens para a representação do conjunto de regras empregue no mecanismo de inferência de um sistema difuso: a inferência baseada na combinação de regras e a inferência baseada em regras individuais.

Na inferência baseada na combinação, as relações difusas, representando o significado de cada uma das regras são agregadas, formando uma única relação a qual descreve o significado global do conjunto de regras. A inferência é, então, conduzida através da composição da entrada fuzificada com a relação global, obtendo-se como resultado um valor difuso para a saída. A utilização desta estratégia conduz-nos à teoria das equações relacionais difusas [Valente de Oliveira, 1992], a qual não será abordada nesta dissertação.

Na segunda abordagem, baseada no accionamento individual de cada regra, a inferência é levada a cabo do modo seguinte: em primeiro lugar, determina-se o grau de pertença do valor numérico em causa em cada um dos conjuntos difusos que descrevem o antecedente da regra; o antecedente fuzificado é obtido pela aplicação dos operadores lógicos de intersecção, união e negação difusa aos graus de pertença obtidos; em seguida, os conjuntos difusos de saída, relativos a cada uma das regras, são transformados de acordo com a operação de implicação definida e com o valor de activação do antecedente correspondente.

A abordagem baseada no accionamento individual é preferida por ser mais eficiente sob o ponto de vista computacional e apresentar um custo reduzido em termos de memória requerida, pelo que será a utilizada neste trabalho.

Selecção de um método de inferência

O significado das implicações presentes nas regras poderá ser dado por métodos diferentes. Um desses métodos, a implicação de Mamdani, é bastante popular em virtude da sua simplicidade, pelo que será utilizado neste trabalho. O operador referido baseia-se, simplesmente, na operação de intersecção, definida pelo operador mínimo. Deste modo, a transformação dos conjuntos difusos de

saída, referida no ponto anterior, é efectuada com base no seu corte, no nível definido pelo grau de pertença do antecedente (Figura 3.9).

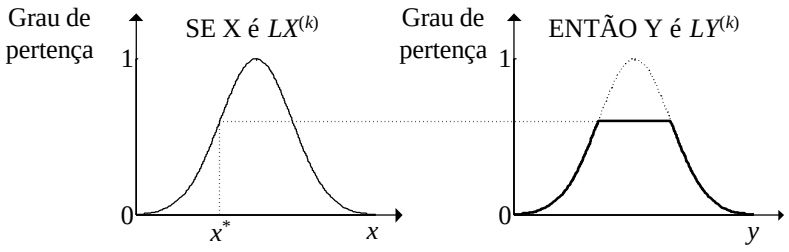


Figura 3.9. Accionamento da regra *k* com base na inferência de Mamdani.

3.3.5. Módulo de Desfuzificação

Em problemas de modelização e controlo, requer-se a obtenção de um valor real para a saída. Deste modo, é necessário definir um mecanismo de transformação do conjunto difuso obtido à saída num valor numérico. Esta operação designa-se por *desfuzificação*.

Existe uma grande diversidade de técnicas de desfuzificação [Driankov et al, 1993; Harris et al, 1993]. Não havendo um procedimento óptimo de selecção da estratégia de desfuzificação, dois dos métodos mais comuns são o método do centro da área e o método da altura. O primeiro, apesar de mais genérico, apresenta algumas desvantagens decorrentes da sua menor eficiência computacional. Por este motivo, o método da altura - ou, mais precisamente, uma sua adaptação - é preferido na dissertação presente.

Método da altura

No método da altura, a saída desfuzificada y^* é calculada com base na soma pesada dos centros de cada conjunto difuso de saída, peso esse determinado pelo valor de pertença do antecedente de cada regra. Formalmente, tem-se (3.37):

$$y^* = \frac{\sum_{k=1}^{g} c_k \cdot m_k}{\sum_{k=1}^{g} m_k} \tag{3.37}$$

onde g indica o número de regras do sistema difuso e c_k e m_k denotam, respectivamente, o pico do conjunto difuso do consequente da regra k, tipicamente o centro de uma Gaussiana, e o valor de pertença do antecedente da mesma regra. Em resultado da definição apresentada, verifica-se que o método não leva em consideração nem o suporte nem a forma dos conjuntos difusos de saída, pelo que equivale à definição de sistemas difusos com consequentes constantes, i.e., sistemas Takagi-Sugeno de ordem 0. Com o objectivo de ultrapassar a limitação enunciada, mantendo a eficiência computacional, em [Lin, 1995] apresenta-se uma extensão ao método referido para funções de pertença Gaussianas, na qual o efeito das larguras é levado em consideração (3.38). Neste método de desfuzificação, o centro da função de pertença é pesado pelo valor do antecedente correspondente, assim como pelo desvio padrão, o que permite incorporar de algum modo o efeito da forma do conjunto difuso na desfuzificação (3.38). Em [Paiva et al, 1999], a expressão (3.38) é,

por sua vez, adaptada para a situação de utilização de funções de pertença Gaussianas generalizadas (3.8), obtendo-se (3.39).

$$y^* = \frac{\sum_{k=1}^{g} c_k s_k m_k}{\sum_{k=1}^{g} s_k m_k} \tag{3.38}$$

$$y^* = \frac{\sum_{k=1}^{g} \left(c_{kL} s_{kL} + c_{kR} s_{kR}\right) \cdot m_k}{\sum_{k=1}^{g} \left(s_{kL} + s_{kR}\right) \cdot m_k} \tag{3.39}$$

O método anterior tem por finalidade fornecer um peso relativo a cada um dos centros da Gaussiana, de acordo com a largura respectiva. Exemplificando, numa função com maior desvio padrão à esquerda, o seu centro esquerdo terá um peso maior na desfuzificação. No caso da função ser simétrica, a qualquer um dos centros será atribuído o mesmo peso. A estratégia de desfuzificação apresentada constitui uma das contribuições originais do trabalho desenvolvido.

Os métodos (3.38) e (3.39) são os utilizados na implementação de modelos linguísticos, uma vez que, nesta dissertação, faz-se uso exclusivamente de funções Gaussianas.

Regras de Takagi-Sugeno

Na utilização de regras de Takagi-Sugeno (3.36), a desfuzificação é efectuada, habitualmente, pela média pesada dos valores numéricos de saída de cada regra, y_i^*, tal como em (3.40):

$$y^* = \frac{\sum_{k=1}^{g} y_k^* m_k}{\sum_{k=1}^{g} m_k} \tag{3.40}$$

Claramente, a expressão anterior é equivalente a (3.37) para sistemas de ordem 0.

3.4. Aproximação Universal

No início do capítulo presente referiu-se que um sistema difuso é susceptível de representar uma função não linear, a qual relaciona de forma determinística as entradas e saídas de um qualquer sistema. No contexto da modelização de sistemas é fundamental que as capacidades de aproximação sejam adequadas. Por outras palavras, o sistema difuso deve ser capaz de representar o processo a tratar, i.e., as saídas do modelo difuso devem aproximar-se tanto quanto possível das saídas reais observadas. A resposta a estas questões, em termos quantitativos, tem sido dada por autores como Li-Xin Wang, James Buckley ou J. L. Castro, tendo o último apresentado os resultados mais genéricos.

Assim, numa primeira aproximação, Wang [Wang, 1992] provou que uma classe particular de sistemas difusos constituem *aproximadores universais*, i.e., são capazes de aproximar qualquer

função com um grau de precisão arbitrário. Essa classe utiliza funções de pertença Gaussianas, intersecção e implicação difusas definidas pelo operador produto e desfuzificação pelo método do centro da área.

Após este primeiro trabalho, Buckley [Buckley, 1993] provou que sistemas difusos do tipo Takagi-Sugeno constituem, também eles, aproximadores universais, com uma pequena modificação no método de desfuzificação. Além disso, a componente associada ao consequente pode conter uma qualquer função polinomial e não apenas funções lineares, tal como em (3.36).

Apesar da importância dos resultados obtidos, é necessário derivar conclusões mais genéricas para outros tipos de funções de pertença, métodos de inferência, estratégias de desfuzificação, etc. Assim, Castro [Castro, 1995] provou a propriedade da aproximação universal para a maior parte das estruturas difusas. Este resultado é aplicável a sistemas difusos linguísticos e de Takagi-Sugeno.

Nos *sistemas difusos linguísticos*, também designados por sistemas de Mamdani, nos quais o consequente é um conjunto difuso, para que a capacidade de aproximação universal seja garantida, é necessário que as condições seguintes se verifiquem:

i) a fuzificação seja do tipo *singleton*;
ii) as funções de pertença tenham suporte compacto;
iii) o sistema de inferência seja do tipo modus ponens generalizado, utilizando-se uma norma-T para a conjunção e para a implicação e uma norma-S para a agregação de regras;
iv) o método de desfuzificação produza um resultado numérico que pertença ao suporte do conjunto difuso do consequente a desfuzificar.

A estrutura definida nas secções anteriores satisfaz a grande maioria dos requisitos. De facto, a fuzificação utilizada é do tipo *singleton*, na inferência são utilizadas normas-T e normas-S e qualquer um dos métodos de desfuzificação apresentados satisfaz a condição iv). No entanto, a condição ii) não se verifica se forem utilizadas funções Gaussianas. Esta limitação pode ser, no entanto, mitigada, no caso de se considerar que a Gaussiana só contém valores significativos no intervalo $[c-3s;\ c+3s]$, sendo nula fora desse intervalo. Tem-se assim um suporte compacto. No entanto, põe-se a hipótese de que a realização deste artifício não seja absolutamente fundamental. De facto, tal como se afirmou anteriormente, Wang provou a propriedade da aproximação universal para uma classe de sistemas difusos com funções de pertença Gaussianas. Eventualmente, poder-se-á provar a propriedade referida sem esta restrição…

Quanto aos *sistemas difusos de Takagi-Sugeno*, nos quais o consequente é definido por uma função real, e.g., uma função polinomial de grau 0 ou 1, Castro provou a propriedade da aproximação universal assumindo as mesmas condições i) a iii) e considerando na condição iv) a desfuzificação expressa em (3.40).

Deste modo, com a ressalva em relação ao facto das funções Gaussianas não apresentarem um suporte verdadeiramente compacto, os sistemas difusos utilizados nesta dissertação - sistemas linguísticos e sistemas de Takagi-Sugeno de ordem 0 e 1 - constituem aproximadores universais.

3.5. Sumário

Este capítulo abordou os aspectos fundamentais da teoria dos sistemas difusos,

nomeadamente conjuntos difusos e raciocínio aproximado, bem como estrutura e projecto de sistemas difusos.

A necessidade de se definirem conjuntos difusos resulta do facto de os objectos do mundo real raramente poderem ser classificados de forma binária, i.e., possuindo ou não possuindo determinada característica. De facto, o grau segundo o qual um objecto pertence a uma determinada categoria não deve ser binário - 1 ou 0, pertence ou não pertence - mas sim contínuo - um valor entre 0 e 1, surgindo então a noção de conjunto difuso (Secção 3.2). Do mesmo modo que na teoria clássica se definem operações entre conjuntos, e.g., intersecção, união ou complemento, essa definição é efectuada na teoria dos conjuntos difusos, com base em operadores genéricos tais como a norma-T, a norma-S e a norma-c.

Após a descrição dos aspectos fundamentais dos conjuntos difusos e raciocínio aproximado, abordou-se, na Secção 3.3, a problemática do projecto de sistemas difusos, nomeadamente os aspectos relacionados com cada um dos seus módulos: fuzificador, base de regras, base de dados, motor de inferência e desfuzificador. Dos vários parâmetros de projecto, realçam-se o problema de síntese da base de regras e da selecção de funções de pertença, que serão analisados em maior detalhe no Capítulo 5.

Finalmente, na Secção 3.4 abordou-se a propriedade da aproximação universal para uma vasta classe de estruturas difusas, o que constitui um formalismo fundamental de suporte e motivação da modelização difusa, discutida nesta dissertação.

Assim, numa palavra, este capítulo poderá ser concluído do mesmo modo que foi iniciado: a teoria dos sistemas difusos constitui uma metodologia particularmente adequada ao tratamento de problemas de identificação de sistemas, nomeadamente em situações com grau de complexidade elevada, onde a presença de não linearidades e de factores de incerteza seja significativa.

Capítulo 4

PRINCÍPIOS DE REDES NEURONAIS

No capítulo precedente foram abordados os aspectos essenciais de sistemas difusos, tendo-se concluído sobre seu interesse na modelização de sistemas dinâmicos, em consequência da propriedade da aproximação universal. Referiu-se ainda que a síntese de uma base de regras, assim como a selecção de funções de pertença constituem dois factores de grande importância, cujo tratamento heurístico é inadequado. Uma das formas de abordar automaticamente os problemas enunciados consiste na utilização de redes neuro-difusas, que não são mais do que sistemas difusos dotados de capacidades de aprendizagem e adaptação. Deste modo, este capítulo descreve os princípios fundamentais das redes neuronais artificiais (ANN)[23], necessários à compreensão dos mecanismos presentes nas estruturas neuro-difusas.

As redes neuronais podem ser caracterizadas como modelos computacionais que procuram emular o funcionamento do cérebro humano. De facto, tais modelos, baseados num conjunto de elementos de processamento simples, fortemente interligados, procuram modelizar a estrutura do córtex cerebral. Assim, as redes neuronais possuem propriedades particulares, tais como capacidade de aprendizagem, adaptação e generalização, inspiradas nos processos cognitivos humanos.

Este capítulo começa por uma breve introdução, na qual são abordados alguns marcos da evolução do interesse por esta área científica, após o que se seguirá, na Secção 4.2, a descrição dos aspectos genéricos de redes neuronais, tais como modelos de neurónios artificiais, topologias de redes e seu treino. As redes RBF serão objecto de tratamento na Secção 4.3. Na Secção 4.4 apresenta-se o método dos mínimos quadráticos, no qual se baseia o algoritmo de retropropagação do erro, descrito na Secção 4.5.

4.1. Introdução

A evolução dos computadores do estádio de "calculadoras automáticas" para "máquinas

[23] *Artificial Neural Networks*, em terminologia inglesa. Habitualmente, o termo *artificial* é abandonado. Assim, deste ponto em diante, a expressão *rede neuronal* (NN - *Neural Network*) significará, mais correctamente, rede neuronal artificial, utilizando-se a notação explícita *rede neuronal biológica* sempre que de tal se trate.

pensantes" constitui, para alguns membros da comunidade científica e da sociedade civil, um problema utópico, enquanto que para outros se apresenta como um desafio aliciante. De qualquer modo, tal hipótese tem servido de motivação para o desenvolvimento de métodos capazes de implementar alguns dos processos cognitivos humanos. As redes neuronais artificiais constituem uma dessas metodologias.

Inicialmente, a motivação para o estudo das redes neuronais artificiais consistia na emulação das estruturas neuro-sinápticas do cérebro humano, as quais armazenam, aprendem e retornam informação com base na experiência. Contudo, a verificação de que se tratava de uma tarefa hercúlea motivou que a investigação se direccionasse maioritariamente para o desenvolvimento de algoritmos capazes de solucionar problemas específicos, tais como modelização e controlo de sistemas, reconhecimento de padrões ou classificação. De facto, o funcionamento dos neurónios biológicos é complexo. No entanto, a investigação conduzida indica que modelos simples, com funções básicas, estão aptos a produzir soluções satisfatórias, ou mesmo excelentes, em resposta a problemas práticos. É com base nestes elementos de processamento simples que tem evoluído o grosso da investigação na área das redes neuronais artificiais.

A origem do desenvolvimento de redes neuronais data, porventura, de meados do século passado, com William James, considerado por muitos o maior psicólogo americano de sempre. James foi pioneiro na publicação de trabalhos relacionados com a estrutura e funcionamento do cérebro humano. Nomeadamente, foi o primeiro a introduzir conceitos como memória associativa e aprendizagem correlacional. Além disso, William James anteviu o facto de que a actividade dos neurónios biológicos pode ser interpretada como uma função da soma das suas entradas.

Em 1943, McCulloch e Pitts [McCulloch e Pitts, 1943] publicaram um dos artigos mais famosos sobre redes neuronais. Aí, os autores desenvolveram teoremas relativos a modelos de sistemas neuronais, com base no conhecimento disponível na altura a nível de estruturas biológicas.

Em 1949, ainda nesta primeira vaga de interesse, Donald Hebb definiu um método de actualização dos pesos sinápticos de neurónios artificiais, no seu livro "*The Organization of Behavior*" [Hebb, 1949]. Este método é hoje designado por *aprendizagem Hebbiana* e constitui a base de muitos dos métodos de aprendizagem utilizados presentemente.

Em 1958, Frank Rosenblatt [Rosenblatt, 1958] definiu uma estrutura neuronal a qual designou por *perceptrão*. Esse trabalho, simulado em detalhe num computador IBM 704, exaltou a imaginação de estudiosos da área da engenharia e do cérebro humano. O perceptrão é considerado o primeiro sistema com capacidade de aprendizagem, uma vez que possibilita a classificação binária de padrões, pela modificação dos pesos das suas ligações sinápticas. O artigo referido lançou as bases de algoritmos de aprendizagem supervisionada e não supervisionada utilizados presentemente, tal como a retropropagação e a aprendizagem de Kohonen. No mesmo artigo, Rosenblatt provou, de forma notável, o teorema da convergência do perceptrão, relativo à aprendizagem deste sistema. Este resultado suscitou um grande interesse e optimismo em relação a esta área.

Outro dos grandes marcos da fase introdutória da investigação sobre redes neuronais, particularmente do ponto de vista de engenharia, foi o artigo "*Adaptive Switching Circuits*", de Widrow e Hoff [Widrow e Hoff, 1960]. Aí, os autores desenvolveram uma estrutura, a qual designaram por *ADALINE* (*ADAptive Linear NEuron*), assim como um algoritmo de aprendizagem para a estrutura referida, designado por *algoritmo de aprendizagem Widrow-Hoff*. Este algoritmo apresenta a vantagem de ser mais rápido e mais preciso do que o algoritmo de aprendizagem do

perceptrão, baseando-se na amplitude do erro à saída do neurónio. Foi demonstrado que o modo de ajuste dos pesos minimiza o somatório do erro quadrático (SSE[24]) sobre todos os exemplos de treino.

A avidez de investigação na área das redes neuronais era, nos 60, enorme, motivada pelos resultados obtidos. No entanto, em 1969 a bomba explodiu: Minsky e Papert [Minsky e Papert, 1969] publicavam o livro "*Perceptrons*", no qual analisavam perceptrões simples, tendo demonstrado que estes elementos eram incapazes de representar uma função tão simples como a função ou-exclusivo (XOR). Os autores sugeriam a utilização de perceptrões com várias camadas de neurónios. No entanto, em face do problema de treino de uma estrutura neuronal multicamada, os autores avaliarem esta área como "estéril". O negativismo patente neste livro originou uma grande desmotivação na comunidade científica, acompanhada da diminuição drástica dos fundos para a investigação. Era o início da "Idade das Trevas" [Eberhart e Dobbins, 1990], que se estendeu até 1982, com o trabalho de Hopfield sobre redes neuronais e sistemas físicos.

Durante esta fase, apenas alguns estudiosos continuaram os seus esforços de investigação na área de redes neuronais, nomeadamente Teuvo Kohonen, Stephen Grossberg, James Anderson e Kunihiko Fukushima.

Em 1972, Kohonen, um engenheiro electrotécnico, e Anderson, um professor de psicologia, publicaram resultados semelhantes sobre desenvolvimentos em redes neuronais. Embora o primeiro tenha designado a sua estrutura por *memória associativa* [Kohonen, 1972], e o segundo por *memória interactiva* [Anderson, 1972], as técnicas utilizadas eram idênticas. Estes trabalhos lançaram as bases das *redes auto-organizadas*, dedicadas a tarefas de classificação, sem supervisão.

Outro dos investigadores resistentes foi Grossberg. O trabalho deste autor centra-se particularmente na plausibilidade fisiológica das estruturas neuronais e não tanto na resolução de problemas práticos, pelo que os seus artigos são, em geral, algo complexos para os estudiosos de áreas da engenharia. Um dos seus trabalhos mais marcantes residiu no desenvolvimento da teoria da ressonância adaptativa (ART[25]) [Grossberg, 1973].

O "último dos bravos" foi o japonês Fukushima, notabilizado pelo desenvolvimento do *neocognitrão* [Fukushima, 1980], com o objectivo de sintetizar uma rede neuronal com a capacidade de reconhecer padrões visuais do mesmo modo que um ser humano.

Apesar dos esforços conduzidos pelos autores referidos, a área das redes neuronais mantinha-se debaixo de uma certa penumbra. No entanto, em 1982, o trabalho de John Hopfield [Hopfield, 1982] desempenhou um papel fundamental no reavivar do campo. Este autor, que granjeava um grande respeito como profissional, não introduziu muitas ideias originais. No entanto, a importância do seu trabalho no despertar do interesse sobre a área deveu-se à forma criativa, mesmo brilhante, como interligou muitos dos aspectos estudados anteriormente. As suas estruturas funcionavam como redes de memória associativa e eram adequadas a problemas de optimização, tendo o autor analisado o seu trabalho com um grande rigor matemático. Além disso, realçou o facto de as suas ideias poderem ser implementadas em circuitos integrados. Assim, a indústria de semicondutores rapidamente se interessou pelos desenvolvimentos de Hopfield. Designadamente, a AT&T Bell Laboratories, anunciou pouco tempo após a publicação do trabalho de Hopfield, as primeiras redes neuronais implementadas em *hardware*. Entrávamos na era do

[24] *Summed Square Error*, em terminologia inglesa.
[25] *Adaptive Ressonance Theory*, em terminologia inglesa.

"Renascimento".

Se o trabalho de Hopfield reacendeu o interesse pelas redes neuronais, a apresentação de uma solução para o problema da aprendizagem em redes multicamada, levantado por Minsky, constituiu o tónico final para a explosão de interesse nesta área. Essa solução baseava-se na retropropagação do erro das camadas externas para as internas, como forma de permitir o ajuste dos pesos nestas últimas. A base conceptual deste método de aprendizagem foi apresentada inicialmente por P. J. Werbos [Werbos, 1974] e reinventada em 1986 por James McClelland e David Rumelhart. Os últimos editaram o livro *"Parallel Distributed Processing"* em dois volumes [Rumelhart e McClelland, 1986; McClelland e Rumelhart, 1986], dividido em capítulos escritos por diferentes elementos do seu grupo de investigação, o Parallel Distributed Process (PDP) Research Group. O livro referido constituiu um sucesso espantoso, o que resultou do facto de dele constar tudo o que havia para conhecer à data sobre redes neuronais, exposto de uma forma simples e interessante. Um dos capítulos do livro que suscitou um maior entusiasmo foi o oitavo, redigido por Rumelhart, Hinton e Williams, intitulado *"Learning Internal Representations by Error Propagation"*.

A partir deste marco, o interesse pela investigação nesta área cresceu rapidamente, o que se reflectiu no número de investigadores, no financiamento, no número e dimensão de conferências internacionais, no número de jornais dedicados ao tema, assim como no número de universidades que integram grupos dedicados à investigação em redes neuronais. As aplicações desta área científica abrangem áreas tão diversas como a indústria aeroespacial e automóvel, a área financeira, medicina, robótica, sistemas de produção, visão computacional e telecomunicações.

Actualmente, a integração das redes neuronais com outros mecanismos de representação do conhecimento tem vindo a ser uma área explorada de modo crescente. Destes sistemas, designados por *sistemas híbridos inteligentes*, merecem particular destaque as redes neuro-difusas que fundem os sistemas difusos com as redes neuronais *standard*, de forma a dotarem os primeiros de capacidades de aprendizagem e adaptação. Como forma de apresentar os fundamentos necessários ao estudo das estruturas neuro-difusas, este capítulo abordará os princípios fundamentais das redes neuronais artificiais[26].

4.2. Aspectos Genéricos

O conceito de rede neuronal pode ser definido do seguinte modo [Haykin, 1994]:

"Uma rede neuronal é um processador distribuído, massivamente paralelo, com uma propensão natural para armazenar conhecimento empírico e torná-lo acessível para uso. Assemelha-se ao cérebro em dois aspectos:

1. *O conhecimento é adquirido pela rede através de um processo de aprendizagem.*
2. *A intensidade das ligações entre neurónios, conhecidas por pesos sinápticos, é utilizada para armazenar o conhecimento."*

[26] Para uma exposição detalhada *vide* [Haykin, 1994] ou [Kröse e van der Smagt, 1993].

Assim, uma rede neuronal é constituída por um conjunto de elementos de processamento simples, os *neurónios*, massivamente interligados e comunicando entre si pelo envio de sinais sobre um número elevado de ligações pesadas. As entidades principais que compõem uma rede neuronal são apresentadas seguidamente, de acordo com a Figura 4.1.

- um conjunto de entradas, x_j, saídas desejadas, y_j, e saídas efectivas, y_j, da rede;
- um conjunto de *unidades de processamento*, neurónios ou células;
- um *sinal de activação*, a_i, para cada unidade *i*, o qual determina a sua saída;
- *ligações pesadas entre as unidades*, definidas por um peso w_{ij}, o qual determina o efeito da unidade *j* na unidade *i*;
- uma *regra de propagação*, que determina a entrada efectiva i_i de uma unidade, em resultado de todas as suas entradas externas;
- uma *função de activação*, F_i, que determina o nível de activação da unidade em função da sua entrada efectiva;
- um termo de *polarização* ou *viés*[27], b_i, para cada unidade;
- um *ambiente de operação*, que forneça sinais de entrada e, eventualmente, sinais de erro.

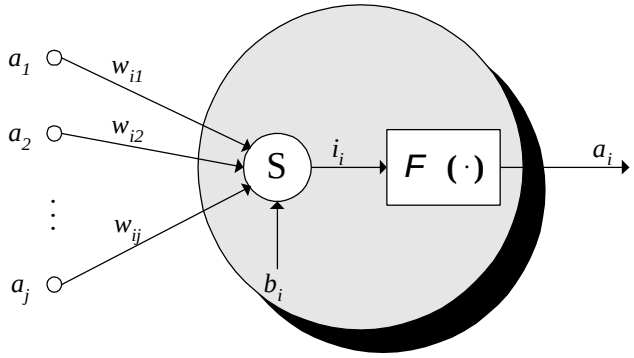


Figura 4.1. Estrutura de um neurónio artificial.

4.2.1. Unidades de Processamento

Tal com é largamente conhecido, o funcionamento dos neurónios biológicos reveste-se de grande complexidade, havendo um número significativo de questões em aberto acerca do seu comportamento. Apesar desta limitação, verificou-se que a resposta a problemas práticos do mundo real pode ser facultada por modelos extremamente simples, desempenhando funções básicas de processamento. A Figura 4.1 representa a estrutura de um neurónio artificial.

Assim, numa rede neuronal artificial, a tarefa de cada neurónio é relativamente simples. Essa tarefa passa pela recepção de sinais de unidades vizinhas ou fontes externas, sinais esses que são utilizados na determinação de um sinal de activação a propagar a outras células, com base numa regra de propagação, numa função de activação e nos pesos das ligações sinápticas.

Habitualmente, distinguem-se três tipos de neurónios: *neurónios de entrada*, que recebem sinais do ambiente exterior; *neurónios de saída*, que enviam dados da rede para o exterior; e

[27] *Bias*, em terminologia inglesa.

neurónios escondidos, que comunicam unicamente com outras unidades da rede. Estes três tipos de células encontram-se nas redes neuronais mais comuns, tais como as redes RBF.

4.2.2. Funções de Activação

A saída de um neurónio depende da função de activação que o caracteriza. De forma genérica, a sua activação depende da entrada líquida i_i (4.1):

$$a_i = F_i(i_i) \tag{4.1}$$

Na maioria das situações, cada uma das entradas de um neurónio i influencia a sua entrada efectiva, i_i, de forma aditiva. Nesse caso, a entrada líquida no neurónio i é determinada pela soma pesada das activações de cada uma das unidades que nela convergem, a_j, juntamente com o termo de polarização b_i (4.2):

$$i_i = \sum_{j=1}^{k} w_{ij} \cdot a_j + b_i \tag{4.2}$$

onde k denota o número de neurónios ligados ao neurónio i, sendo w_{ij} o peso de cada uma dessas ligações. No caso do peso ser positivo, a sua contribuição designa-se por *excitação*. Na situação oposta, ocorre uma *inibição* do neurónio. Os neurónios com uma regra de propagação da forma (4.2) denominam-se *unidades sigma*.

Em estruturas do tipo RBF (Secção 4.3) utilizam-se funções tais como a Gaussiana. Nessa situação, não se procede ao cálculo de qualquer entrada líquida, sendo as ligações da rede responsáveis pelo armazenamento dos parâmetros da função de activação, i.e., centro e desvio padrão, necessários ao cálculo da saída, tal como se verificará posteriormente.

4.2.3. Estruturas de Redes Neuronais

Tal como se referiu, uma rede neuronal consiste num conjunto de neurónios interligados. O tipo de ligações entre as unidades de processamento que constituem uma rede define a sua estrutura. Assim sendo, as redes neuronais podem ser divididas em duas classes principais: as *redes com ligações para a frente*[28] e as *redes recorrentes*.

Redes com ligações para a frente

Nesta classe de redes neuronais incluem-se estruturas como a MLP (Perceptrão Multicamada) [Rumelhart e McClelland, 1986] e as redes RBF (redes com funções de base radial) [Broomhead e Lowe, 1988; Moody e Darken, 1989], as quais são largamente utilizadas em problemas de identificação e controlo. Desta classe constam ainda estruturas como a LVQ[29] [Kohonen, 1989], as redes CMAC[30] [Albus, 1975] e as redes GMDH[31] [Hecht-Nielsen, 1990]. A

[28] Redes *feedforward*, em terminologia inglesa.
[29] *Learning Vector Quantization*, em terminologia inglesa.
[30] *Cerebellar Model Articulation Control*, em terminologia inglesa.
[31] *Group-Method for Data Handling*, em terminologia inglesa.

Figura 4.2 exemplifica a estrutura apresentada.

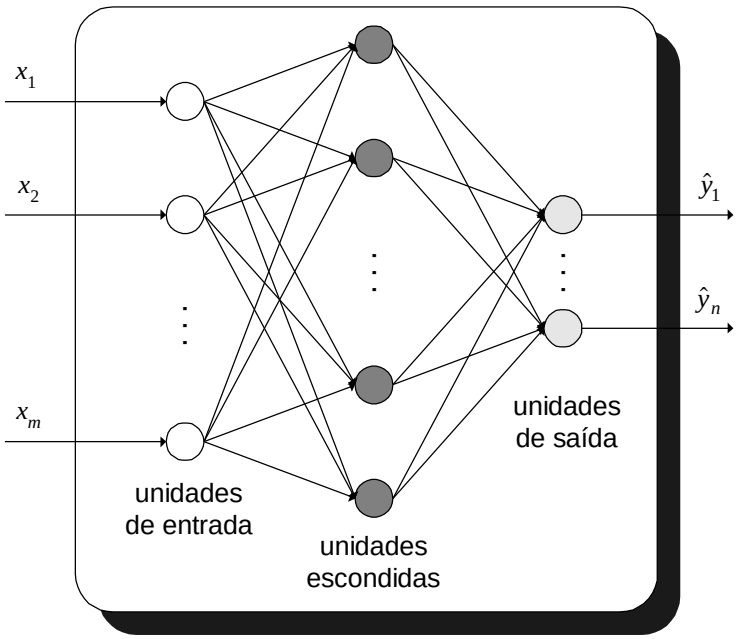


Figura 4.2. Rede neuronal com ligações para a frente.

Redes recorrentes

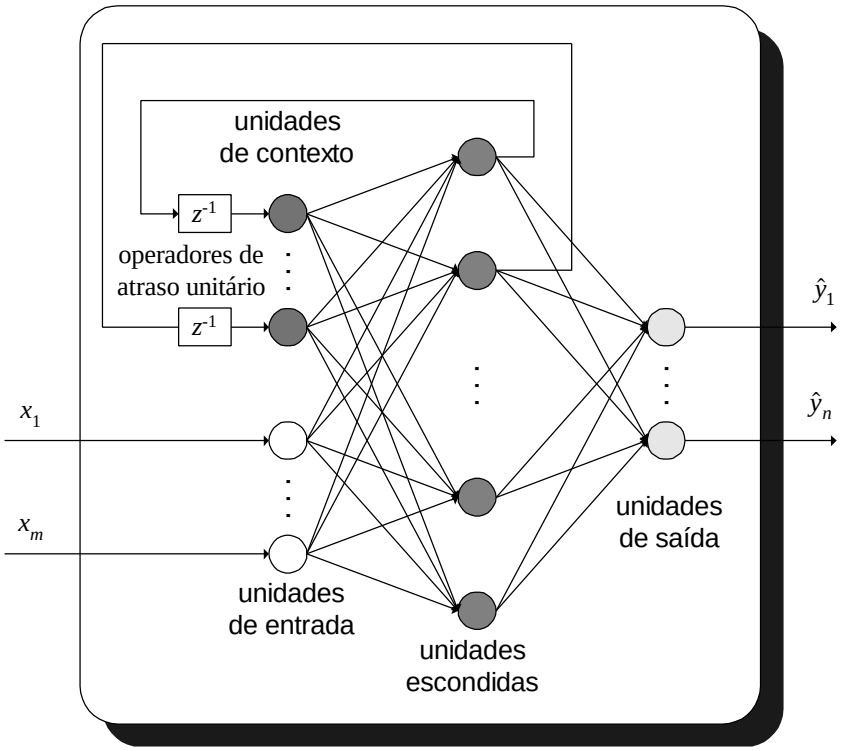


Figura 4.3. Rede neuronal recorrente (rede de Elman).

As redes recorrentes contêm ligações para trás e/ou ligações entre neurónios da mesma camada (e.g., Figura 4.3). Assim, o fluxo dos sinais é bidireccional. Ao contrário das redes com

ligações para a frente, nesta classe as propriedades dinâmicas da rede revestem-se de grande importância. De facto, as redes recorrentes contêm memória dinâmica, pelo que as suas saídas num dado instante reflectem o efeito da entrada corrente, assim como de entradas e saídas passadas. Em virtude desta propriedade, estruturas como a rede de Elman [Elman, 1990] (Figura 4.3) ou a rede de Jordan [Jordan, 1986], são utilizadas na modelização de sistemas dinâmicos [Henriques e Dourado, 1998]. Outra rede incluída nesta classe é a rede de Hopfield [Hopfield, 1982], a qual funciona como memória associativa, sendo também utilizada em problemas de optimização conduzidos por uma função objectivo.

As redes neuro-difusas analisadas na dissertação presente inserem-se na classe das redes com ligações para a frente.

4.2.4. Treino de Redes Neuronais

Para que uma determinada rede neuronal alcance os objectivos desejados, as entradas por si recebidas devem produzir saídas adequadas. Deste modo, é fundamental que os parâmetros da rede sejam configurados convenientemente. Uma das maneiras de o efectuar consiste em atribuir à rede, explicitamente, os pesos sinápticos das suas ligações, com base em conhecimento prévio, o que raramente é possível. Assim, a maneira mais habitual de configurar os parâmetros de uma rede consiste no seu *treino*, guiado por uma determinada regra de aprendizagem, com base num conjunto de padrões de treino com os quais a rede é alimentada.

Genericamente, definem-se duas classes de paradigmas de aprendizagem: a *aprendizagem supervisionada* e a *aprendizagem não supervisionada*.

Aprendizagem supervisionada

Segundo este paradigma, os pesos sinápticos de uma rede neuronal são ajustados com base em exemplos de treino constituídos por pares entrada-saída, obtidos do ambiente de funcionamento da rede. Por exemplo, na modelização de sistemas, os exemplos de treino da rede serão constituídos por dados do funcionamento do sistema, nomeadamente das suas variáveis de entrada e saída. Exemplos de algoritmos de aprendizagem supervisionada incluem a regra delta, ou regra de Widrow-Hoff [Widrow e Hoff, 1960], o algoritmo de retropropagação do erro [Rumelhart e McClelland, 1986] e o algoritmo LVQ [Kohonen, 1989].

Esta classe inclui ainda a *aprendizagem por reforço*, a qual constitui um caso especial de aprendizagem supervisionada [Barto et al, 1983].

Aprendizagem não supervisionada

Este paradigma, também designado por aprendizagem auto-organizada, caracteriza-se pela não necessidade de existência de um supervisor, quer seja na forma de padrões de treino, quer na forma de crítico. Nesta situação, a rede adapta automaticamente os pesos das suas ligações de forma a agrupar os padrões de entrada com característicos semelhantes. Ao contrário da aprendizagem supervisionada, as categorias segundo as quais os padrões de entrada devem ser classificados não são fornecidas, sendo tarefa da rede encontrá-las autonomamente. Deste paradigma constam, por exemplo, o algoritmo de aprendizagem competitiva de Kohonen [Kohonen, 1989] e o algoritmo ART de Grossberg [Grossberg, 1973].

De entre os paradigmas enunciados, será dado ênfase à aprendizagem supervisionada, sem a

inclusão da aprendizagem por reforço. A aprendizagem não supervisionada será também aplicada neste trabalho, ainda que de uma forma breve, pelo que a sua temática não será abordada neste capítulo. Deste modo, os aspectos relacionados com esta temática utilizados na dissertação presente serão expostos posteriormente, no Capítulo 5, no contexto da identificação neuro-difusa.

4.3. Redes RBF

Dentro das estruturas multicamada com ligações para a frente, as estruturas do tipo perceptrão multicamada (redes MLP) e as redes com funções de base radial (redes RBF) têm merecido uma atenção especial, em virtude das suas propriedades de aproximação funcional, que as tornam particularmente atraentes em problemas de modelização e controlo de sistemas. Uma vez que as redes RBF se enquadram no trabalho elaborado, o que não acontece com as arquitecturas MLP, a secção presente irá descrever, sucintamente, os aspectos essenciais destas estruturas.

As redes RBF [Moody e Darken, 1989; Broomhead e Lowe, 1988], caracterizam-se pelo mapeamento funcional com base em campos receptivos locais, inspirados nos campos receptivos biológicos do córtex cerebral.

Na sua forma mais básica, estas estruturas são compostas por três camadas distintas. A primeira, a camada de entrada, tem por único objectivo receber sinais do ambiente exterior e passá-los à camada seguinte. A segunda camada, a camada escondida, é constituída por um conjunto de neurónios, a cada um dos quais se encontra associado um vector de parâmetros designado por *centro*. Cada célula calcula a distância Euclidiana entre o centro respectivo e o vector de entrada, com base numa função de activação não linear de base radial, tal como a função Gaussiana. Quanto à camada de saída, o mapeamento efectuado é linear, ao contrário da transformação não linear da camada escondida. A descrição exposta é apresentada graficamente na Figura 4.4.

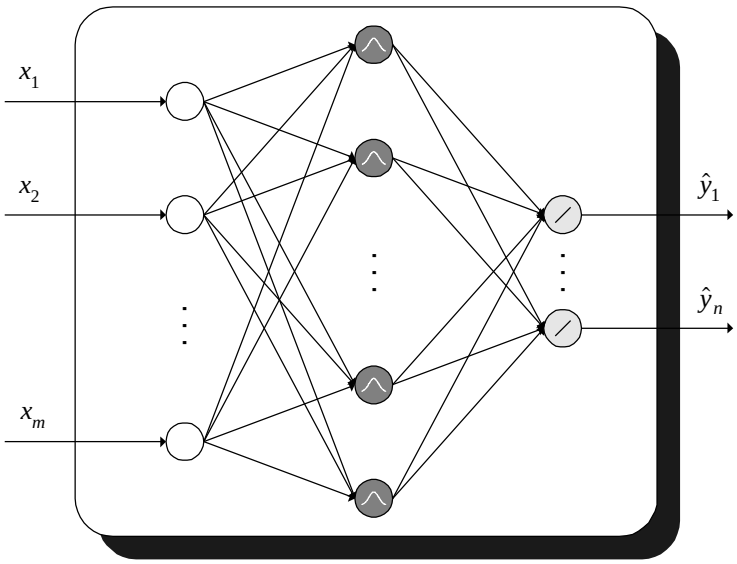


Figura 4.4. Estrutura de uma rede RBF básica.

Resumidamente, dado o *p*-ésimo padrão de entrada, $x^{(p)}$, a saída da rede é dada por (4.3):

$$\hat{y}_o^{(p)} = f_o(x^{(p)}) = \sum_{r=1}^{g} w_{or} F_r\left(\left\| x^{(p)} - c_r \right\|\right) \quad , c_r \in \Re^m, \quad o = 1,2,\cdots,n \tag{4.3}$$

em que *m* e *n* representam, respectivamente, o número de entradas e saídas da rede, *g* designa o número de neurónios escondidos, w_{or} denota os pesos da camada linear da rede, c_r representa cada um dos vectores de centros, $\|\cdot\|$ designa a distância Euclidiana, e F_r constitui uma função de activação radial multivariável, tal como a Gaussiana (4.4). Nessa expressão, o centro e desvio padrão constituem os parâmetros a ajustar, não havendo qualquer termo de polarização, ao contrário das redes MLP.

$$F_r : \Re^m \to \Re, \quad F_r(x) = e^{-\frac{\|x - c_r\|^2}{2\sigma_r^2}} \tag{4.4}$$

Quanto às estratégias de treino de redes RBF, a forma mais geral consistirá na aplicação de um determinado esquema de optimização de parâmetros em estruturas multicamada, tal como o algoritmo de retropropagação do erro. Tal como se verificará posteriormente, este algoritmo apresenta algumas desvantagens centradas essencialmente na não garantia de convergência para o mínimo global de uma função de erro e nos elevados tempos de treino. Deste modo, a particularidade da estrutura das redes RBF, constituídas por uma camada linear e outra não linear, sugere esquemas de treino híbridos, os quais se apresentam vantajosos, sobretudo a nível de tempo de convergência.

Assim, em alternativa ao algoritmo de retropropagação do erro, os pesos da camada saída poderão ser ajustados segundo um qualquer critério de optimização linear, tal como o algoritmo dos mínimos quadráticos, descrito na Secção 4.4. Esta estratégia apresenta as vantagens associadas a esse algoritmo, nomeadamente em termos de convergência para o mínimo global, sob algumas restrições aceitáveis. Coloca-se agora a questão do ajuste dos pesos da camada escondida, i.e., dos centros e larguras das funções radiais. Numa primeira abordagem, os centros são distribuídos uniformemente por todo o espaço de entrada, mantendo-se fixos durante todo o processo de treino. Quanto às larguras, os seus valores são determinados de forma a permitir que as funções se sobreponham de forma adequada.

A estratégia descrita poderá ser melhorada, bastando para tal possibilitar o ajuste dos pesos da camada escondida, em lugar de os manter fixos. Neste caso, utilizam-se, habitualmente, esquemas de aprendizagem não supervisionada, os quais permitem encontrar grupos naturais presentes nos dados, ajustando-se, assim, os centros de acordo com a distribuição de amostras pelo espaço de entrada. No caso de redes RBF, é usual utilizar-se a regra dos *k vizinhos mais próximos* [Moody e Darken, 1989], segundo a qual os *k* centros mais próximos de um padrão de entrada são deslocados no sentido desse padrão.

Uma outra alternativa, utilizada neste trabalho no contexto das redes neuro-difusas, baseia-se num esquema híbrido de optimização da camada de saída e ajuste dos parâmetros das camadas escondidas pela retropropagação (Secção 5.3).

Os algoritmos de treino baseados na optimização linear pelo método LS constituem uma estratégia adequada para aprendizagem em tempo real, em virtude da possibilidade da sua implementação recursiva, descrita na secção seguinte. Esta vantagem, associada à propriedade da localidade (Secção 4.5.1), característica das redes RBF, torna estas estruturas particularmente interessantes no contexto de modelização e controlo em tempo real [Pereira, 1996].

Dos aspectos descritos ressalta, claramente, a equivalência funcional entre redes RBF e sistemas difusos [Jang e Sun, 1993]. De facto, para que tal se verifique basta:

i) interpretar o número de neurónios escondidos como o número de regras da estrutura difusa;
ii) considerar sistemas difusos do tipo Takagi-Sugeno de ordem 0;
iii) estabelecer a equivalência entre as funções de activação dos neurónios da segunda camada e as funções de pertença difusas, o que é imediato, por exemplo, com funções Gaussianas;
iv) definir como operador de conjunção difusa o produto, tal como acontece com a activação dos neurónios escondidos;
v) calcular a saída da rede com base na saída pesada de cada neurónio escondido, o que torna este procedimento idêntico ao método de desfuzificação para sistemas Takagi-Sugeno (3.40).

Como consequência da equivalência funcional entre as duas estruturas, as redes RBF podem ser classificadas como pertencendo à classe mais genérica das redes neuro-difusas, com a vantagem da possibilidade de troca de conhecimento entre ambas as áreas.

4.3.1. Aproximação Universal

Um dos resultados mais importantes no estudo e aplicação de redes neuronais RBF reside no facto de tais estruturas gozarem da propriedade da aproximação universal [Girosi e Poggio, 1990]. De facto, qualquer função real é susceptível de ser aproximada por uma rede RBF com um grau de precisão arbitrário. Para tal, basta que o número de neurónios escondidos seja suficiente e que as funções de activação sejam contínuas e limitadas, o que acontece, por exemplo, com as funções Gaussianas. Adicionalmente, em [Park e Sandberg, 1991] prova-se que larguras idênticas em todas as funções de activação mantêm a propriedade da aproximação universal. No entanto, a variação das larguras poderá favorecer a estrutura utilizada, nomeadamente em termos da diminuição do número de neurónios escondidos necessários. Por outro lado, em consequência da equivalência entre redes RBF e sistemas difusos, a aproximação universal de redes RBF pode ser provada tal como na Secção 3.4.

4.4. Algoritmo dos Mínimos Quadráticos

Tal como se referiu anteriormente, muitas das regras de aprendizagem utilizadas correntemente podem ser consideradas como variações da regra de Hebb [Hebb, 1949]. A ideia geral do seu autor era de que se dois neurónios i e j estiverem activos simultaneamente, a sua ligação deve ser fortalecida pelo aumento do peso sináptico. Assim, na sua versão mais simples, a variação do peso da ligação entre dois neurónios i e j, Δw_{ij}, é efectuada segundo a expressão (4.5):

$$\Delta w_{ij} = \eta a_i a_j \tag{4.5}$$

em que $\eta \geq 0$ é uma constante de proporcionalidade representando a *velocidade de aprendizagem*, sendo a_i a activação do neurónio i e a_j a activação do neurónio j.

Com base na lei de adaptação (4.5), Widrow e Hoff [Widrow e Hoff, 1960] definiram uma regra para aprendizagem da estrutura ADALINE. Nesta arquitectura, constituída por um único

neurónio de saída, a actualização dos pesos das suas ligações com as unidades de entrada baseia-se, não na activação do neurónio de saída, mas sim no seu erro. Deste modo, a implementação do método referido requer que a saída desejada, *d*, seja fornecida por um supervisor, tratando-se, portanto, de um método de aprendizagem supervisionada. Assim sendo, a expressão (4.5) é alterada de forma a obter-se (4.6):

$$\Delta w_j = \eta(y - \hat{y})x_j \tag{4.6}$$

em que x_j representa o valor da *j*-ésima entrada da rede, *y* denota a activação do único neurónio de saída e, como tal, a saída da rede e *y* denota a saída desejada para o neurónio. De notar que o facto de na expressão anterior não se utilizar o índice *i*, deriva do facto da regra ter sido desenvolvida para estruturas com um só neurónio de saída. A expressão (4.6) constitui a base do algoritmo dos mínimos quadráticos, o qual é também designado por regra delta. Este método foi desenvolvido originalmente para o treino da estrutura ADALINE, inspirada no perceptrão de Rosenblatt. Actualmente, a regra delta constitui um bloco importante em áreas como a identificação e controlo de sistemas e processamento de sinal.

O algoritmo baseia-se na minimização de uma *função de erro* que, tal como o nome "least square (LS)" indica, é o somatório do erro quadrático (SSE). Assim, o erro total *E* (4.7), define-se como a soma quadrática dos erros $E^{(p)}$ (4.8), determinados para cada um dos padrões de treino:

$$E = \sum_{p=1}^{N} E^{(p)} \tag{4.7}$$

$$E^{(p)} = \frac{1}{2}\left\| y^{(p)} - \hat{y}^{(p)} \right\|^2 \tag{4.8}$$

onde $\hat{y}^{(p)} \in \Re$ e $y^{(p)} \in \Re$ designam, respectivamente, a activação e saída desejada para o neurónio, relativamente ao padrão de treino *p*, sendo *N* o número de exemplos de treino fornecidos à rede. O erro *E* constitui uma função de todos os parâmetros livres da estrutura neuronal. Deste modo, a actualização dos parâmetros referidos, a qual constitui o objectivo do processo de aprendizagem, é efectuada com base na minimização do critério SSE.

Tal como foi proposto no desenvolvimento original do algoritmo, a minimização é conduzida iterativamente pelo *método do gradiente*. A ideia geral deste método consiste em fazer variar progressivamente os pesos (e eventuais termos de polarização)[32], no sentido da diminuição do erro quadrático (4.9):

$$\Delta w_j = -\eta \frac{\partial E^{(p)}}{\partial w_j} \tag{4.9}$$

O sinal negativo associado à velocidade de aprendizagem relaciona-se com a necessidade da variação dos pesos no sentido da diminuição do erro. De facto, o método do gradiente é designado mais apropriadamente por *método da descida do gradiente*. A Figura 4.5 representa graficamente a dinâmica expressa na equação (4.9).

Do exposto, torna-se óbvia a denominação de velocidade de aprendizagem para o parâmetro η De facto, o seu valor determina o *passo* da rede na descida da superfície de erro.

[32] Neste texto, optou-se por apresentar a expressão da variação dos pesos. No entanto, o método de obtenção da expressão matemática de actualização dos termos de polarização obtém-se de maneira exactamente igual.

Na implementação da regra (4.9), a derivada do erro em relação ao peso é determinada pela aplicação da regra da cadeia (4.10):

$$\frac{\partial E^{(p)}}{\partial w_j} = \frac{\partial E^{(p)}}{\partial \hat{y}^{(p)}} \frac{\partial \hat{y}^{(p)}}{\partial w_j} \tag{4.10}$$

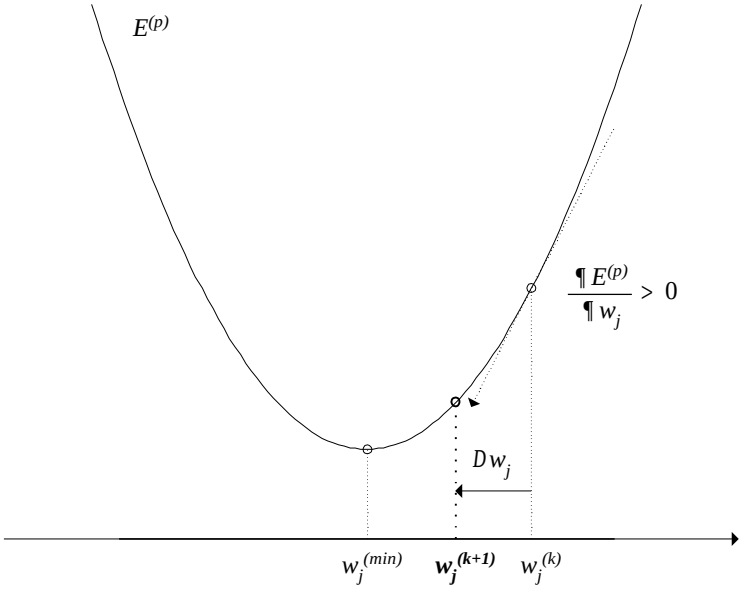


Figura 4.5. Ilustração do método do gradiente.

Na figura anterior, *k* indica o número da iteração. Uma vez que a ADALINE é composta por um único neurónio de saída linear, vem (4.11):

$$\hat{y}^{(p)} = \sum_{j=1}^{m} w_j \cdot x_j^{(p)} + b \tag{4.11}$$

Desta forma, obtém-se (4.12) e (4.13):

$$\frac{\partial \hat{y}^{(p)}}{\partial w_j} = x_j^{(p)} \tag{4.12}$$

$$\frac{\partial E^{(p)}}{\partial \hat{y}^{(p)}} = -\left(y^{(p)} - \hat{y}^{(p)}\right) \tag{4.13}$$

Deste modo, a expressão (4.9) converte-se em (4.14):

$$\Delta w_j = \eta d^{(p)} x_j \tag{4.14}$$

Aqui, $d^{(p)} = y^{(p)} - \hat{y}^{(p)}$ representa o erro entre a saída desejada e a saída real do neurónio.

Em lugar do algoritmo iterativo descrito, o procedimento de minimização poderá ser conduzido num só passo, com base na teoria das equações matriciais. De facto, com base em (4.11) obtém-se a equação matricial (4.15):

$$y^{(p)} = \sum_{j=1}^{m} w_j \cdot x_j^{(p)} + b = X^{(p)^T} W \tag{4.15}$$

em os vectores $X^{(p)} \in \Re^{m+1}$ (4.16) e $W \in \Re^{m+1}$ (4.17) representam as entradas da rede e os seus pesos:

$$X^{(p)^T} = \left[x_1^{(p)} x_2^{(p)} \cdots x_n^{(p)} 1\right] \tag{4.16}$$

$$W^T = \left[w_1 w_2 \cdots w_n b\right] \tag{4.17}$$

O objectivo proposto é, então, determinar os parâmetros *W* com base em *N* padrões de treino. Esta situação permite obter um sistema de *N* equações, apresentado seguidamente (4.18):

$$\begin{cases} y^{(1)} = X^{(1)^T} W \\ y^{(2)} = X^{(2)^T} W \\ \vdots \\ y^{(N)} = X^{(N)^T} W \end{cases} \tag{4.18}$$

ou em notação matricial (4.19):

$$Y = X^T W \tag{4.19}$$

A maneira mais intuitiva de obter *W* consiste em utilizar um conjunto de *N=m+1* padrões de treino, de forma a que a matriz *X* seja quadrada. Assim, os pesos da rede determinam-se, de forma simples, com base na expressão (4.20):

$$W = X^{T^{-1}} Y \tag{4.20}$$

No entanto, esta abordagem raramente é aplicável. De facto, é fundamental que a matriz *X* seja não singular. Por outro lado, ruído e perturbações nos dados são razões suficientes para a utilização de um número de amostras superior ao número de parâmetros. Nesta situação, sendo *N*>*m*+1, o sistema de equações em (4.18) torna-se sobredeterminado, o que implica, regra geral, a inexistência de uma solução exacta. Deste modo, define-se o critério SSE (4.7), o qual conduz à determinação dos parâmetros que minimizam o erro quadrático. Neste caso, os parâmetros *W* são determinados como em (4.21), dando origem ao *estimador dos mínimos quadráticos* (LSE[33]):

$$W = \left(X^T X\right)^{-1} X^T Y \tag{4.21}$$

A abordagem anterior exige que a matriz $X^T X$ seja positiva definida. A matriz referida é sempre, pela sua construção, positiva semi-definida. No entanto, no caso de a mesma ser singular, o requisito enunciado não se verifica. Nesta situação, o critério quadrático (4.7) terá um número infinito de soluções. Assim, de forma de garantir que a matriz $X^T X$ seja postiva definida, a recolha de dados deve ser adequada, tanto quanto possível, de forma a que o conjunto de amostras obtidas seja suficientemente informativo, tal como se referiu no Capítulo 2.

A expressão (4.21) pode ser representada, alternativamente, por (4.22):

[33] *Least Square Estimator*, em terminologia inglesa.

$$W = \left[\sum_{k=1}^{N} X^{(k)} X^{(k)^T} \right]^{-1} \left[\sum_{k=1}^{N} X^{(k)} y^{(k)} \right] \tag{4.22}$$

Definindo-se a matriz *P* como (4.23):

$$P(p) = \left[\sum_{k=1}^{p} X^{(k)} X^{(k)^T} \right]^{-1} \tag{4.23}$$

em que o índice *p* denota a apresentação do *p*-ésimo padrão de treino, verifica-se que a expressão (4.22) pode ser obtida recursivamente. De facto, é fácil verificar que (4.24):

$$P^{-1}(p) = P^{-1}(p-1) + X^{(p)} X^{(p)^T} \tag{4.24}$$

resultando, após algumas manipulações simples [Söderström e Stoica, 1989], a expressão (4.25):

$$W(p) = W(p-1) + P(p) X^{(p)} \left[y^{(p)} - \hat{y}^{(p)} \right] \tag{4.25}$$

a qual, tal como seria de esperar, está de acordo com a regra (4.14). De facto, poder-se-á afirmar que o factor $P(p)X^{(p)}$ em (4.25) equivale à definição de uma velocidade de aprendizagem adaptativa óptima para cada um dos parâmetros a ajustar.

Pelas expressões (4.24) e (4.25), verifica-se que os pesos são ajustados após a apresentação de cada uma das amostras de treino. Deste modo, o procedimento descrito designa-se por algoritmo dos *mínimos quadráticos recursivos* (RLS[34]), sendo largamente utilizado no contexto da identificação de sistemas lineares. Um aspecto de grande importância em termos de eficiência computacional deriva do facto de, na implementação recursiva da matriz *P*, ser necessário proceder a uma inversão de matriz em cada iteração, o que se afigura dispendioso a nível de eficiência computacional. Deste modo, utilizando o lema de inversão de matrizes [Söderström e Stoica, 1989], vem (4.26):

$$P(p) = P(p-1) - \frac{P(p-1) X^{(p)} X^{(p)^T} P(p-1)}{1 + X^{(p)^T} P(p-1) X^{(p)}} \tag{4.26}$$

A matriz *P*, é por vezes, designada por matriz de co-variância dos parâmetros *W*. Na verdade, considerando que os dados são afectados por ruído branco, ruído esse descrito por uma sequência de variáveis aleatórias estatisticamente independentes de média nula e de variância l^2, a matriz de co-variância é dada, mais precisamente, por (4.27):

$$\mathrm{cov}(W) = l^2 P \tag{4.27}$$

4.4.1. Análise do Algoritmo dos Mínimos Quadráticos

Verificou-se que, no caso da matriz *P* ser não singular, o algoritmo dos mínimos quadráticos apresenta a importante vantagem de permitir encontrar, de forma eficiente e inequívoca, o mínimo global do critério de erro (4.7) em problemas lineares. Afirmou-se ainda que este requisito depende

[34] *Recursive Least Squares*, em terminologia inglesa.

da estrutura seleccionada, assim como das condições experimentais.

Um aspecto relevante do método prende-se com a propriedade da *consistência*. Assim, é condição necessária para a consistência do algoritmo dos mínimos quadráticos que o ruído seja branco ou que a sequência de entrada seja independente da sequência de ruído, tal como se referiu no Capítulo 2.

Outro aspecto significativo relacionado com este algoritmo prende-se com a sua implementação recursiva. Nesta situação é fundamental inicializar-se a matriz P e o vector de parâmetros W adequadamente. Usualmente, tal informação não se encontra disponível, pelo que é comum inicializar-se os parâmetros com valores nulos. Em relação à matriz P, sendo esta directamente proporcional à matriz de co-variância, os seus valores iniciais devem reflectir a confiança depositada na inicialização dos parâmetros. Assim, se a matriz P for inicializada com valores baixos, na actualização dos parâmetros estes não se afastarão significativamente dos valores iniciais. Na situação inversa, sendo P inicializada com valores elevados, a variação inicial dos parâmetros é considerável. Neste sentido, uma vez que habitualmente não se tem qualquer noção sobre os valores reais dos parâmetros, é prática comum inicializar P como uma matriz diagonal com valores "elevados", tal como se segue (4.28):

$$P = sI \tag{4.28}$$

em que s representa um valor "elevado".

A versão recursiva do algoritmo dos mínimos quadráticos é utilizada habitualmente em tarefas de identificação de sistemas em tempo real. Nesta situação, o algoritmo é modificado ligeiramente de modo a incluir um termo designado por *factor de esquecimento*, o qual possibilita a adaptação do modelo do sistema a dinâmicas variáveis no tempo. Aspectos relacionados com a adaptação em linha de sistemas serão abordados na Secção 5.3.3.

4.5. Algoritmo de Retropropagação do Erro

A regra de Widrow-Hoff, tal como foi expressa pela equação (4.14), baseia-se na utilização de um único neurónio linear de saída, bem como numa rede sem camadas internas. No entanto, estruturas mais complexas, tais como as redes MLP ou RBF, contêm várias camadas de neurónios e utilizam unidades de processamento não lineares, sendo a camada de saída eventualmente composta por mais do que uma célula. Deste modo, a regra delta deve ser generalizada para o conjunto de funções de activação não lineares, camadas de saída com vários neurónios e estruturas multicamada.

Assim sendo, a função de erro, E, a minimizar expressa-se como em (4.7), sendo $E^{(p)}$ definido agora por (4.29), em virtude da eventual existência de vários neurónios de saída i:

$$E^{(p)} = \frac{1}{2}\sum_{i=1}^{n}\left(y_i^{(p)} - \hat{y}_i^{(p)}\right)^2 \tag{4.29}$$

Tal como anteriormente, os pesos sinápticos são adaptados no sentido da diminuição do erro (4.30):

$$\Delta w_{ij} = -g\frac{\partial E^{(p)}}{\partial w_{ij}} \tag{4.30}$$

de onde, após manipulações simples, resulta a expressão para a regra delta generalizada (4.31):

$$\Delta w_{ij} = \eta \delta_i^{(p)} a_j^{(p)} \tag{4.31}$$

A determinação do factor $\delta_i^{(p)}$ é efectuada igualmente com recurso à regra da cadeia. Assim, vem (4.32):

$$\delta_i^{(p)} = -\frac{\partial E^{(p)}}{\partial \text{net}_i^{(p)}} = \left(y_i^{(p)} - \hat{y}_i^{(p)}\right) \cdot \frac{\partial \hat{y}_i^{(p)}}{\partial \text{net}_i^{(p)}} \tag{4.32}$$

No caso em que a função de activação é linear, $\delta_i^{(p)} = y_i^{(p)} - \hat{y}_i^{(p)}$, tal como em (4.14).

Do exposto, uma questão permanece em aberto: como ajustar os pesos das ligações das camadas escondidas, uma vez que para elas não há, directamente, qualquer sinal de erro? Este problema é solucionado pelo algoritmo de retropropagação do erro, baseado na regra delta generalizada.

Assim, a aplicação da retropropagação decorre em duas fases. Na primeira, as entradas são apresentadas e propagadas para a frente, através da rede. Deste modo, são calculadas as activações dos vários neurónios, até à camada de saída. Com base na saída da rede, i.e., nas activações de cada um dos neurónios de saída, e na saída desejada, calculam-se os sinais $\delta_i^{(p)}$ (4.32), para cada uma das unidades de saída. São estes sinais que, numa segunda fase, se propagam para trás - retropropagam - através da rede, de forma a permitir alterações apropriadas dos pesos de todas as ligações interneuronais, incluindo aquelas referentes a camadas escondidas.

Na aplicação do algoritmo da retropropagação distinguem-se dois tipos de neurónios: as unidades de saída e as unidades escondidas.

Neurónios de saída

Em relação às unidades de saída, a resposta desejada é conhecida, uma vez que corresponde à saída pretendida para a rede neuronal. Desta forma, o problema de actualização dos pesos é trivial, sendo essa tarefa realizada com base nas equações (4.31) e (4.32).

Neurónios escondidos

Quanto aos neurónios escondidos, a resposta desejada não é conhecida, o que impossibilita a determinação de qualquer sinal de erro, necessário à adaptação dos pesos. Assim sendo, o erro relativo a um neurónio interno é determinado recursivamente com base nos neurónios da camada seguinte, aos quais se encontra ligado.

Deste modo, considere-se um neurónio escondido *h*. O cálculo do sinal $\delta_h^{(p)}$ é efectuado com base nos sinais $\delta_i^{(p)}$, referentes aos neurónios da camada seguinte à do neurónio abordado. De facto, tal como se verificou em (4.32), $\delta_h^{(p)}$ é calculado, de acordo com a regra da cadeia, por (4.33):

$$\delta_h^{(p)} = -\frac{\partial E^{(p)}}{\partial \text{net}_h^{(p)}} = \frac{\partial a_h^{(p)}}{\partial \text{net}_h^{(p)}} \sum_{i=1}^{no} \delta_i^{(p)} w_{ih} \tag{4.33}$$

onde $a_h^{(p)}$ representa a activação do neurónio escondido *h*.

Em relação aos termos de polarização, o seu ajuste é efectuado de modo em tudo idêntico ao ajuste dos pesos, descrito nos parágrafos precedentes.

4.5.1. Análise do Algoritmo de Retropropagação do Erro

A aplicação da retropropagação ao treino de redes neuronais multicamada apresenta várias vantagens, nomeadamente a sua facilidade de implementação computacional e o facto de permitir, em geral, estruturas com boa capacidade de generalização. No entanto, a sua aplicação prática pressupõe a satisfação de alguns requisitos em termos de diferenciabilidade das funções de activação dos neurónios, sendo necessário definir modos de treino - por lotes ou padrão a padrão - e critérios de paragem. Para além do referido, o algoritmo apresenta algumas dificuldades relacionadas, fundamentalmente, com a selecção da velocidade de aprendizagem e com as suas propriedades de convergência.

Modos de Treino[35]

Na aplicação do algoritmo de retropropagação, a aprendizagem resulta da apresentação repetida do conjunto de padrões de treino. A cada apresentação da totalidade dos exemplos de treino à rede dá-se o nome de *época* ou *iteração*. Deste modo, o processo de treino de uma rede é conduzido durante um determinado número de épocas, apresentando-se à rede, em cada iteração, a totalidade dos padrões de treino. Esta apresentação pode decorrer de dois modos: o modo incremental ou o modo de operação por lotes.

No *modo incremental*, a actualização dos pesos é realizada após a apresentação de cada um dos padrões de treino. A principal limitação desta abordagem reside no facto de que, não sendo utilizados todos os exemplos em simultâneo, o algoritmo de treino não siga verdadeiramente o gradiente, mas sim uma sua aproximação. Uma vantagem deste modo de operação deriva da sua aplicabilidade a problemas de aprendizagem em linha, uma vez que nesta situação os dados de treino surgem sequencialmente. No entanto, é importante notar que a utilização de funções sigmoidais, habituais em redes MLP, cujo suporte se estende por todo o domínio, poderá originar alterações significativas no comportamento da rede entre a apresentação de dois padrões consecutivos. De facto, o carácter global das funções de activação sigmoidais leva a que actualizações nos parâmetros de uma dessas funções alterem, de forma global, o mapeamento efectuado pela rede, alterações essas que se farão sentir em zonas extensas do espaço de entrada-saída. Deste modo, em problemas de aprendizagem em linha, a utilização de estruturas com funções de activação locais, e.g., Gaussianas, afigura-se vantajosa [Brown e Harris, 1994]. Neste caso, a alteração dos parâmetros da função afecta apenas localmente o mapeamento global da rede, em virtude da sua natureza compacta[36]. Este é o caso das redes RBF.

No *modo de operação por lotes*, a adaptação dos pesos é efectuada após a apresentação de todo o conjunto de padrões. Ao contrário da aprendizagem incremental, nesta metodologia os pesos são actualizados segundo a direcção do gradiente. No entanto, a sua aplicação a problemas de aprendizagem em linha não é viável, uma vez que os exemplos de treino não se encontram disponíveis na sua totalidade.

De forma a melhorar o comportamento do algoritmo de aprendizagem em situações de operação em linha, utiliza-se, por vezes, uma abordagem híbrida, baseada na construção de um

---

[35] Por conveniência de exposição, os modos de treino são apresentados no contexto do algoritmo de retropropagação do erro. No entanto, os esquemas descritos são aplicáveis a qualquer outra estratégia de optimização.

[36] De notar o referido na Secção 3.4 relativamente à aproximação do suporte das funções Gaussianas, de forma a torná-lo compacto.

histórico constituído pelas últimas *N* amostras recolhidas, sendo o algoritmo aplicado a esse conjunto em cada iteração [Mills et al, 1996]. Porém, tal estratégia apresenta um custo computacional mais elevado, o qual poderá ser incomportável em tempo real.

Critérios de Paragem

Tipicamente, o número de épocas necessárias ao treino de uma rede neuronal obedece a um conjunto de critérios, designados por *critérios de paragem* [Haykin, 1994]. Estes critérios, definidos de forma heurística, são necessários uma vez que, em geral, a convergência do algoritmo de retropropagação não pode ser provada.

Tal como se sabe, para que se tenha atingido um mínimo, local ou global, da superfície de erro é condição necessária que o gradiente, i.e., a primeira derivada da função de erro em relação aos pesos, seja nulo. Deste aspecto resulta, de forma natural, a definição de um critério de paragem segundo o qual se assume a convergência da retropropagação no caso da norma euclidiana do gradiente atingir um valor inferior a um determinado limiar. No caso enunciado surgem algumas dificuldades que resultam da circunstância de os tempos de treino serem, regra geral, elevados.

Num outro critério admite-se que o algoritmo convergiu se o MSE (4.34) - ou o RMSE[37] (4.35) - for suficientemente pequeno, i.e., se se situar abaixo de um limiar previamente especificado.

$$MSE = \frac{1}{N}\sum_{i=1}^{n}\frac{1}{2}\left(y_i^{(p)} - \hat{y}_i^{(p)}\right)^2 \tag{4.34}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}\frac{1}{2}\left(y_i^{(p)} - \hat{y}_i^{(p)}\right)^2} \tag{4.35}$$

Numa terceira abordagem, porventura a mais utilizada, define-se um critério baseado nos dois anteriores. Assim, assume-se que o algoritmo convergiu se o RMSE (ou o MSE) ou a norma do gradiente, apresentarem valores suficientemente baixos.

Há ainda um critério de paragem baseado nas propriedades de generalização da rede. Neste caso, o seu desempenho em termos de capacidade de generalização é testado no fim de cada iteração, e.g., cálculo do RMSE, pondo-se termo ao treino se o seu valor se revelar adequado ou, alternativamente, no momento em que o critério RMSE começar a aumentar. O aumento do RMSE para os dados de teste, em contraste com a sua diminuição para os dados de treino, indicia um treino excessivo da rede, problema este abordado na Secção 2.4.

Neste trabalho, assume-se que a aprendizagem convergiu no caso do critério RMSE ser suficientemente pequeno, ou ainda, se o comportamento face aos dados de teste se deteriorar.

Velocidade de aprendizagem

A determinação do valor adequado para a velocidade de aprendizagem, g de uma rede neuronal constitui um dos aspectos de maior dificuldade na parametrização de um algoritmo de treino. De facto, uma velocidade baixa tem a vantagem de permitir alterar, de forma suave, os

[37] MSE - *Mean Square Error*: Erro Quadrático Médio; RMSE - *Root Mean Square Error*: Raiz do Erro Quadrático Médio. Os dois critérios são largamente utilizados na computação da medida do erro de aproximação fornecido por uma qualquer técnica, e.g., redes neuronais. O critério RMSE apresenta a vantagem de fornecer um resultado mais intuitivo sobre a magnitude real do erro, em virtude de resultar da raiz do MSE.

pesos sinápticos em cada iteração. No entanto, esta vantagem conduz a uma taxa de aprendizagem menor, o que se manifesta sob a forma de tempos de treino elevados. Por outro lado, uma velocidade de aprendizagem elevada resulta em alterações substanciais nos pesos das ligações, entre duas iterações consecutivas. No entanto, caso a velocidade seja excessiva, os pesos variarão de tal forma que a rede poderá tornar-se instável, i.e., oscilar sem conseguir atingir o mínimo. A dificuldade reside na inexistência de mecanismos teóricos rigorosos para selecção da óptima da velocidade de aprendizagem, o que, na maioria das situações, resulta em tempos de aprendizagem elevados. No sentido de se mitigar os problemas que advêm de uma escolha deficiente, utiliza-se neste trabalho uma velocidade adaptativa, tal como será descrito posteriormente.

Propriedades de convergência

O treino de uma rede neuronal pelo algoritmo de retropropagação do erro pode levar a que se obtenha, para a função de erro a minimizar, uma solução local e não o desejável mínimo global, tal como se apresenta na Figura 4.6.

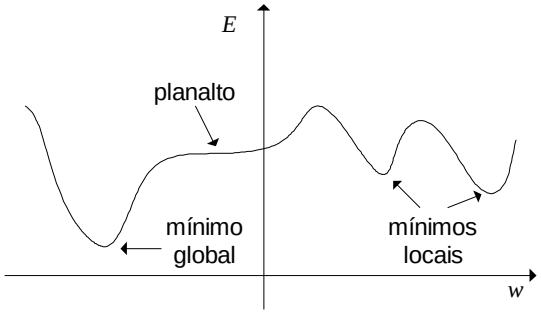


Figura 4.6. Mínimos locais no algoritmo de retropropagação do erro.

De facto, a retropropagação é, basicamente, uma técnica "*trepa colinas*[38]", onde a direcção seguida é a da descida do gradiente, a qual aponta no sentido da diminuição do erro e não necessariamente no sentido do mínimo global. Além disso, a existência de planaltos na superfície de erro pode levar a que o algoritmo aí permaneça. Assim sendo, quanto maior for a complexidade da superfície de erro, maior será a probabilidade do algoritmo bloquear num mínimo local ou planalto. A maior dificuldade associada a este problema prende-se com a falta de ferramentas teóricas que o permitam estudar. Os métodos existentes baseiam-se em restrições limitativas, e.g., o estudo de redes lineares ou a utilização de padrões de treino linearmente separáveis [Haykin, 1994].

Nesta situação, são colocadas duas hipóteses. Na primeira, o mínimo local obtido poderá ser considerado satisfatório, de acordo com determinados requisitos de desempenho, descritos na Secção 2.6. No entanto, nos casos em que tal não ocorra, tenta-se solucionar o problema, habitualmente através de modificações na estrutura.

As dificuldades enunciadas, juntamente com a lentidão da aprendizagem neste algoritmo, sugerem a utilização de métodos mais avançados, alguns baseados em ajustes ao método de retropropagação e outros baseados em técnicas de optimização diferentes.

[38] *Hill-climbing*, em terminologia inglesa.

4.5.2. Alternativas e Modificações à Retropropagação

Tal como se referiu, os dois problemas fundamentais do algoritmo de retropropagação do erro derivam do possível bloqueio da rede em mínimos locais, assim como da lentidão na convergência. Deste modo, foram desenvolvidas algumas extensões ao método, nomeadamente a utilização de um coeficiente de *momentum* e de velocidades de aprendizagem adaptativas. Outras estratégias baseiam-se na utilização de métodos de optimização mais evoluídos, tais como os algoritmos de Gauss-Newton [Widrow e Stearns, 1985] e o gradiente conjugado [Polak, 1971], os quais se caracterizam pela procura do mínimo segundo várias direcções, e não apenas a do gradiente. Durante os últimos anos, a abordagem genética [Holland, 1975], tem crescido em popularidade. O seu interesse reside, fundamentalmente, no facto de possibilitar a obtenção do mínimo global da função de erro, uma vez que o algoritmo efectua uma procura não guiada. No entanto, a razão do seu potencial é também a sua principal limitação, em virtude dos tempos de procura proibitivamente longos, necessários em muitas situações. No trabalho presente, o algoritmo de retropropagação do erro é modificado pela inclusão de uma velocidade adaptativa, tal como se segue.

Velocidade de aprendizagem adaptativa

Os problemas associados às baixas taxas de convergência inerentes ao algoritmo de retropropagação podem ser mitigados pela utilização de velocidades de aprendizagem não constantes.

Assim, Jacobs [Jacobs, 1988] definiu um conjunto de regras heurísticas que fornecem indicações para a variação da taxa de aprendizagem. O que autor sugere, resume-se em quatro regras fundamentais:

i) a atribuição de uma velocidade de aprendizagem para cada peso é vantajosa. Este aspecto resulta do facto de, eventualmente, uma velocidade de aprendizagem adequada para um determinado parâmetro, não o ser para outro;
ii) cada uma das velocidades de aprendizagem deve poder variar ao longo do treino, uma vez que a superfície de erro se comporta de maneira distinta ao longo de regiões diferentes do mesmo peso.
iii) quando a derivada do erro em relação a um dos pesos tem o mesmo sinal algébrico durante várias iterações consecutivas, a velocidade de aprendizagem deve aumentar;
iv) quando o sinal algébrico da derivada do erro em relação a um dos pesos alterna durante várias iterações consecutivas, a velocidade de aprendizagem deve diminuir;

Nesta dissertação, não se definem velocidades distintas para cada um dos pesos, mas sim uma única velocidade global, a qual varia segundo os itens iii) e iv). De notar ainda que a variação da velocidade é guiada pela variação do critério de erro utilizado, e.g., RMSE, e não pela derivada da função de erro, ao contrário do expresso nas regras anteriores.

4.6. Sumário

O capítulo presente abordou os aspectos fundamentais das redes neuronais artificiais.

O interesse pelo desenvolvimento de redes neuronais artificiais partiu do ideal científico de criação de "máquinas pensantes". Assim, as ANN surgiram com o objectivo inicial de emularem a estrutura do cérebro humano, de forma a dotarem os sistemas onde fossem utilizadas de capacidades de aprendizagem, adaptação e generalização. Neste sentido, a sua estrutura baseia-se num conjunto de elementos de processamento ligados entre si, cada um dos quais executando uma tarefa simples, no sentido da consecução do objectivo global de aprendizagem.

Após uma breve resenha histórica relativa à evolução das redes neuronais, foram apresentados, na Secção 4.2, os seus princípios fundamentais, a estrutura dos neurónios artificiais, os tipos de arquitecturas e metodologias de treino mais comuns.

Na Secção 4.3 descreveram-se as redes RBF, tendo-se concluído que as mesmas são funcionalmente equivalentes a sistemas difusos.

Como base para a derivação do algoritmo de treino de redes neuronais unicamada, descreveu-se, na Secção 4.4, o algoritmo dos mínimos quadráticos, ou regra delta. Na mesma secção, referiu-se que os parâmetros da rede são ajustados no sentido da descida do gradiente, tendo-se concluído que o algoritmo é consistente no caso de se verificarem alguns requisitos relativos à estrutura utilizada, ao ruído presente nos dados e às condições experimentais, de acordo com o exposto no Capítulo 2. Nesta secção foram ainda abordados os aspectos essenciais da versão recursiva do algoritmo.

Na Secção 4.5, procedeu-se à generalização da regra delta, de forma a possibilitar o treino de redes multicamada, tendo-se definido o algoritmo de retropropagação do erro, o qual se baseia na propagação de um sinal de erro da camada de saída para as camadas escondidas, para assim ser possível o ajuste dos pesos internos. Verificou-se que este algoritmo de optimização não linear, apesar de algumas vantagens importantes, apresenta duas limitações significativas: o facto de não garantir a convergência para o mínimo global da função de erro a minimizar, bem como a lentidão na convergência. Deste modo, foi considerada a utilização de uma velocidade de aprendizagem adaptativa.

Assim, o capítulo presente abordou os aspectos fundamentais de redes neuronais, como base para a descrição que se segue, relativa a arquitecturas neuro-difusas.

Capítulo 5

IDENTIFICAÇÃO NEURO-DIFUSA

Verificou-se anteriormente que a identificação de sistemas com base em estruturas difusas apresenta propriedades interessantes, decorrentes da aproximação universal de que gozam, da possibilidade de transparência do conhecimento obtido e da sua validação pericial, bem como da facilidade de modificação manual. As duas tarefas essenciais da identificação difusa são, então, a selecção de regras e de funções de pertença associadas às variáveis linguísticas incorporadas no modelo, tarefa esta designada por *aprendizagem da estrutura*, e a sintonização de funções de pertença, designada por *aprendizagem de parâmetros*. Os objectivos enunciados não são atingidos, em geral, de forma trivial, pelo que são necessárias metodologias automáticas. Neste contexto, as redes neuro-difusas apresentam-se como uma alternativa particularmente interessante, uma vez que permitem extrair e conjugar o potencial de aprendizagem das redes neuronais com as vantagens a nível de interpretabilidade dos sistemas difusos.

O capítulo presente começa pela exposição de uma súmula das estratégias mais frequentes de construção automática de sistemas difusos, assim como das várias categorias de redes neuro-difusas.

Na Secção 5.2, as questões associadas à aprendizagem da estrutura em sistemas difusos são endereçadas, bem com alguns aspectos relativos à selecção de entradas relevantes a incluir num modelo.

Na Secção 5.3 são apresentadas arquitecturas e metodologias de treino de redes neuro-difusas, sendo dado especial ênfase a algoritmos de treino fora de linha. Ainda assim, são tecidas algumas considerações relativas ao treino de redes neuro-difusas em linha.

Dado que a interpretabilidade constitui uma vantagem potencial da implementação de modelos difusos, são sugeridas e descritas algumas metodologias seguidas no sentido da sua manutenção durante a determinação de parâmetros.

5.1. Introdução

Dos vários aspectos a ter em conta no projecto de modelos difusos, os quais foram discutidos na Secções 3.3, duas tarefas merecem atenção particular, dada a sua preponderância nas capacidades de aproximação do modelo obtido: a aprendizagem da estrutura e a aprendizagem de parâmetros.

Convencionalmente, modelos difusos são construídos com base no conhecimento e

experiência de um perito, o qual descreve o sistema com base num conjunto de regras linguísticas. Este tipo de modelização padece de algumas limitações importantes. Em primeiro lugar, a existência e disponibilidade de um perito nem sempre se verifica. Ainda que esta dificuldade se solucione, o seu conhecimento é frequentemente incompleto, subjectivo e episódico. Deste modo, o conjunto de regras por ele fornecidas revela-se útil na construção de um modelo inicial, um protótipo. No entanto, é fundamental depurar-se esse mesmo protótipo. Obviamente, a depuração de um modelo difuso não se compadece com processos de tentativa e erro, de custo elevadíssimo a nível do tempo de realização. Nitidamente, há que utilizar métodos que permitam a selecção automática de um conjunto de regras e de um conjunto de funções de pertença para as variáveis utilizadas, além da sua sintonização.

5.1.1. Metodologias de Construção Automática de Sistemas Difusos

No sentido da implementação automática de sistemas difusos baseados em regras[39], duas classes essenciais de métodos se perfilam como aplicáveis: a abordagem baseada em técnicas de agrupamento e a abordagem neuro-difusa.

Uma das estratégias mais comuns na implementação de modelos difusos, consiste na utilização de *técnicas de agrupamento* para a determinação de um conjunto de regras, assim como dos respectivos conjuntos difusos para os antecedentes e consequentes. Particularizando, os algoritmos de agrupamento difuso, e.g., *c-médias difusas* [Bezdek, 1981], permitem, com base num número de regras previamente especificado, determinar os antecedentes e consequentes dessas mesmas regras, tarefa essa guiada por um critério a optimizar. A sua limitação principal prende-se com o facto de os conjuntos difusos obtidos não serem representáveis, frequentemente, por funções matemáticas numa forma fechada, i.e., expressas segundo uma dada expressão paramétrica. Neste sentido, em [Babuška e Setnes, 1998], os conjuntos difusos são projectados e aproximados por funções expressas numa forma fechada. Naturalmente que esta estratégia induz alguma perda em relação à optimização conduzida inicialmente, pelo que seria necessário, em geral, proceder à reoptimização dos conjuntos difusos obtidos.

Uma outra estratégia, a qual tem vindo a merecer um interesse crescente por parte da comunidade científica, consiste na incorporação de mecanismos de aprendizagem na tarefa da identificação difusa, o que conduz à *abordagem neuro-difusa*. Esta metodologia caracteriza-se pela tentativa de conjugação e aproveitamento das capacidades das redes neuronais com as dos sistemas difusos. De facto, a principal vantagem das redes neuronais reside na sua capacidade de aprendizagem a partir de exemplos. No entanto, esta vantagem é reduzida pela sua estrutura caixa-negra (excepto nas redes com funções de base radial), que impede, geralmente, a inclusão de conhecimento prévio na inicialização, a interpretação linguística do estado final obtido e a sua modificação manual. Ao invés, os sistemas difusos caracterizam-se pela sua natureza linguística, o que os torna potencialmente interpretáveis. Porém, tais estruturas não estão intrinsecamente habilitadas a aprender, pelo que a selecção de regras e de funções de pertença se afigura dificultada. Pelo exposto transparece que as duas abordagens se complementam, de forma que a ideia de as combinar numa estratégia neuro-difusa surge naturalmente.

---

[39] Nesta dissertação não se aborda o problema da construção automática de outro tipo de modelo difusos, e. g., relacionais.

Para além das referidas, existem ainda diversas metodologias, essencialmente heurísticas, onde se inclui, a título de exemplo, o trabalho de Wang e Mendel [Wang e Mendel, 1992a]. Aqui, o procedimento genérico consiste em dividir os espaços de entrada e saída em regiões, às quais se atribui uma etiqueta, i.e., um valor linguístico. Seguidamente, determinam-se os graus de pertença de cada padrão em cada uma das regiões, formando-se regras com base na escolha das regiões com grau de pertença mais elevado, tanto para o antecedente como para o consequente. Finalmente, as regras criadas a partir dos dados numéricos e as regras definidas por peritos humanos são combinadas numa base de regras, o que resulta no mapeamento do espaço de entrada no espaço de saída, através de um método de desfuzificação. A principal limitação deste método reside no elevado número de regras geradas, assim, como na necessidade de tratamento de possíveis situações de inconsistência. Além do referido, não são apresentadas garantias de que os conjuntos difusos obtidos sejam os melhores, uma vez que não se implementa qualquer mecanismo de optimização. A sua principal vantagem reside na sua simplicidade e rapidez de implementação. Deste modo, esta classe de métodos poder-se-á apresentar particularmente adequada para a construção de protótipos de sistemas difusos, optimizados numa segunda fase.

Na mesma linha, e no contexto do controlo, incluem-se os controladores adaptativos difusos auto-organizados e auto-ajustáveis. No primeiro caso, em que o controlador altera um conjunto inicial de regras ou parte de uma base vazia, destaca-se o trabalho de Procyk e Mamdani [Procyk e Mamdani, 1979]. No segundo caso, a estratégia seguida consiste, essencialmente, na adaptação de factores de escala (*vide* [Victor e Dourado, 1997]) com base numa base de regras previamente definida e mantida durante a tarefa de controlo. Em [Victor, 1998], o problema do controlo difuso auto-organizado e auto-ajustável é descrito detalhadamente.

Tal como se referiu anteriormente, na identificação difusa distinguem-se dois problemas essenciais: a aprendizagem de regras e a aprendizagem de parâmetros. Na estratégia neuro-difusa, a realização destas tarefas pode ser conduzida, fundamentalmente, de três maneiras: aprendizagem de parâmetros pela rede e aprendizagem da estrutura segundo outra metodologia; aprendizagem da estrutura pela rede, sendo os parâmetros especificados a priori; e aprendizagem tanto da estrutura como dos parâmetros pela rede.

Assim, frequentemente, a rede neuro-difusa dedica-se única e exclusivamente à tarefa de *aprendizagem de parâmetros*, utilizando, para tal, um algoritmo de treino supervisionado, e.g., retropropagação (Secção 4.5). Neste caso, as regras difusas são obtidas separadamente, frequentemente por técnicas de agrupamento de classes. Neste grupo, poder-se-á incluir o trabalho de Takagi e Sugeno [Takagi e Sugeno, 1985]. Os autores referidos desenvolveram um algoritmo de identificação de modelos difusos com aprendizagem de estrutura e parâmetros, o qual provou ser extremamente útil e genérico. Nesse método, as variáveis das premissas são identificadas através de um algoritmo heurístico de procura, o qual consiste, basicamente, na geração progressiva de nós numa estrutura em árvore. A identificação dos conjuntos difusos dos antecedentes é efectuada com base na partição do espaço de entrada, após a selecção das variáveis das premissas. Assim, a tarefa de optimização dos parâmetros das funções de pertença, guiada por um índice de desempenho, reduz-se a um problema de programação não linear. Quanto à identificação dos parâmetros dos consequentes, esta tarefa é levada a cabo com recurso ao método dos mínimos quadráticos. Assim sendo, nesta metodologia podem considerar-se duas fases essenciais: na primeira, define-se a estrutura do modelo difuso, assim como os parâmetros dos antecedentes; na segunda, os parâmetros dos consequentes são optimizados. Esta estratégia apresenta algumas semelhanças com um dos métodos de treino de redes RBF (Secção 4.3), no qual são determinados os pesos da camada escondida (parâmetros das funções de pertença dos antecedentes), os quais se mantêm fixos durante a optimização linear dos pesos da camada de saída (parâmetros dos consequentes). Dado que tais

estruturas são incluídas na classe das redes neuro-difusas (Secção 4.3), também a implementação de modelos difusos do tipo Takagi-Sugeno, assim como outros afins, pode ser considerada parte integrante desta classe de métodos.

Uma outra estratégia consiste em utilizar a rede neuro-difusa para a *aprendizagem de regras*. Nesta situação, os conjuntos difusos são definidos previamente, treinando-se posteriormente a rede de modo não supervisionado, com base em algoritmos de aprendizagem auto-organizada.

Numa terceira abordagem, as duas estratégias referidas nos parágrafos anteriores são conjugadas, obtendo-se uma rede neuro-difusa com capacidade de aprendizagem da estrutura e dos parâmetros. Este esquema é utilizado por Lin [Lin, 1995] e será desenvolvido neste capítulo, juntamente com a primeira abordagem, baseada em técnicas de agrupamento e optimização pela rede.

5.1.2. Classificação de Redes Neuro-Difusas

Tal como se verificou, qualquer trabalho de modelização difusa que utilize métodos de optimização resultantes das redes neuronais, e.g., método de Takagi e Sugeno, pode ser classificado, virtualmente, como uma estratégia neuro-difusa. No entanto, a primeira referência conhecida, na qual esta classificação é efectuada de forma explícita, está ligada ao Japão, nomeadamente ao congresso "International Conference on Fuzzy Logic and Neural Networks - *IIZUKA'88*". Nesse congresso, foram apresentados os trabalhos "Artificial-neural-network driven fuzzy reasoning" [Takagi e Hayashi, 1988] e "NFS: Neuro fuzzy inference system" [Furuya et al, 1988]. A origem geográfica dos trabalhos referidos não está dissociada da liderança conduzida pelo povo japonês na aplicação prática da lógica difusa. De facto, ainda a comunidade científica ocidental começava a vencer os tabus e desconfianças em relação a esta área (Secção 3.1) e já os investigadores japoneses, dois anos após o trabalho do grupo PDP sobre redes neuronais [Rumelhart e McClelland, 1986;McClelland e Rumelhart, 1986], se apercebiam das vantagens potenciais resultantes da conjugação das duas metodologias.

De acordo com a metodologia seguida e com os objectivos propostos, as redes neuro-difusas podem ser classificadas, fundamentalmente, em três categorias, da maneira seguinte:

i) redes neuronais convencionais para raciocínio difuso;
ii) redes neuronais fuzificadas;
iii) sistemas difusos representados por arquitecturas em rede.

A primeira classe inclui redes neuronais convencionais utilizadas em esquemas de raciocínio difuso. Deste grupo constam, por exemplo, o trabalho de Keller [Keller et al, 1992], no qual o treino de redes neuronais é efectuado por conjuntos difusos definidos pelos seus graus de pertença num domínio discreto, sendo a tarefa da rede a de implementar um sistema de raciocínio difuso. Ainda nesta categoria, insere-se o trabalho pioneiro de Takagi e Hayashi [Takagi e Hayashi, 1988].

Na segunda classe estão englobadas as redes neuronais fuzificadas, i.e., redes das quais constam números difusos nas entradas, saídas e/ou pesos sinápticos. Nesta classe, a rede realiza operações difusas, e.g., adição e multiplicação, as quais constituem generalizações das operações clássicas. Nesta área, tem particular preponderância o trabalho de Buckley e Hayashi [Buckley e Hayashi, 1995]. As redes neuronais fuzificadas apresentam-se como as de utilização mais genérica. De facto, a sua aplicação estende-se por áreas como a aproximação funcional [Buckley e Hayashi, 1995], onde as entradas e saídas numéricas podem ser definidas como singulares difusos, sendo os

pesos difusos definidos, geralmente, por níveis-α; a classificação difusa [Ishibuchi et al, 1993a], na qual os pesos numéricos da rede são ajustados de forma supervisionada, de modo a que padrões difusos de entrada sejam categorizados adequadamente; aprendizagem a partir de dados difusos [Lin e Lu, 1996; Paiva, 1997], onde as entradas e saídas da rede são constituídas por conjuntos difusos, sendo os pesos reais ou difusos, o que possibilita concentrar bases de regras redundantes, assim como completar bases de regras incompletas por interpolação; e aproximação funcional difusa, i.e., construção de funções difusas de variável real, em que a rede, constituída por pesos difusos, é treinada com entradas reais e saídas difusas [Ishibuchi et al, 1993b].

No terceiro grupo inserem-se várias arquitecturas neuronais que têm por factor comum o facto de representarem sistemas difusos. Nestas estruturas, as entradas e saídas da rede são reais, o mesmo se passando com os seus pesos, os quais, tipicamente, constituem os parâmetros das funções de pertença do sistema difuso a ajustar. Desta classe constam as arquitecturas ARIC[40] e GARIC[41] de Berenji [Berenji, 1992], as redes neuro-difusas de Horikawa [Horikawa et al, 1992], a arquitectura ANFIS[42] de Jang [Jang, 1993], as redes de Shann e Fu [Shann e Fu, 1995], a arquitectura NFCN[43] de Lin [Lin, 1995], as estruturas NEFCON[44] [Nauck, 1994], NEFCLASS[45] [Nauck e Kruse, 1995] e NEFPROX[46] [Nauck e Kruse, 1999] de Nauck e Kruse, entre muitas outras. No contexto de aprendizagem incremental, há ainda a referir a rede FALCON[47] [Lin et al, 1995], o algoritmo de Figueiredo e Gomide [Figueiredo e Gomide, 1997] e a estrutura SONFIN[48] [Juang e Lin, 1998]. Deste grupo fazem ainda parte os sistemas difusos com capacidade de aprendizagem de Ichihashi e Watanabe [Ichihashi e Watanabe, 1990], Nomura [Nomura et al, 1992] e Wang e Mendel [Wang e Mendel, 1992b]. As estratégias de aprendizagem descritas não abordam explicitamente o conceito de rede neuronal sendo implementadas, no entanto, de maneira funcionalmente idêntica. Do mesmo modo, o trabalho de Babuška e Setnes [Babuška e Setnes, 1998], no qual se utiliza um algoritmo difuso para a determinação de um conjunto de regras e dos respectivos conjuntos difusos, seguindo-se uma fase de optimização dos termos dos consequentes, do tipo Takagi-Sugeno de ordem 1, pode comparar-se ao treino de uma rede RBF com pesos fixos na camada escondida (Secção 4.3).

Uma vez que as redes neuro-difusas nesta última classe implementam sistemas de inferência difusos, as arquitecturas referidas são habitualmente utilizadas em problemas de identificação e controlo. Deste modo, é sobre esta categoria que incidirá o capítulo presente. Assim sendo, o termo "rede neuro-difusa" será utilizado deste ponto em diante, de forma um pouco abusiva, com o intuito de designar esta classe de redes.

5.1.3. Formulação do Problema

Os aspectos expostos no decorrer do capítulo corrente baseiam-se em alguns pressupostos

---

[40] *Approximate Reasoning-based Intelligent Control*, em terminologia inglesa.
[41] *Generalized ARIC*, em terminologia inglesa.
[42] *Adaptive Network-based Fuzzy Inference System*, em terminologia inglesa.
[43] *Neural Fuzzy Control Network*, em terminologia inglesa.
[44] *NEuro Fuzzy CONtrol*, em terminologia inglesa.
[45] *NEuro Fuzzy CLASSification*, em terminologia inglesa.
[46] *NEuro Fuzzy function apPROXimator*, em terminologia inglesa.
[47] *Fuzzy Adaptive Learning COntrol Network*, em terminologia inglesa.
[48] *Self-cOnstructing Neural Fuzzy Inference Network*, em terminologia inglesa.

relativamente aos sistemas a tratar, para além do estabelecimento de alguns dos parâmetros relativamente ao projecto de modelos difusos.

Assim, quanto às propriedades do sistema a identificar, assume-se que se trata de um sistema dinâmico[49] (e causal), estável segundo o critério BIBO[50] [Ogata, 1990] discreto - ou melhor, contínuo discretizado -, caso genérico MIMO (englobando as possibilidades mais simples) e não linear. Quanto à questão da variância temporal, serão assumidos sistemas invariantes, uma vez que será dada particular atenção a métodos de identificação fora de linha (*offline*). Ainda assim, na parte final do presente capítulo, serão abordados alguns aspectos de identificação em linha (*online*), aplicáveis a sistemas variantes no tempo.

No que respeita ao tipo de modelos considerados, utilizam-se modelos difusos, em virtude do contexto em que se insere este trabalho de dissertação. Assim, em termos de estrutura paramétrica, considerar-se-ão modelos do tipo *FARX* (Secção 2.4.3). Como consequência das propriedades dos sistemas difusos, bem como dos aspectos assumidos para os sistemas a abordar, os modelos considerados serão discretos, invariantes (com a salvaguarda do aspecto referido acima), genericamente MIMO, não lineares e do tipo entrada-saída. Esta última assunção baseia-se no pressuposto de que o conjunto de amostras utilizadas na sua identificação seja suficientemente rico, de forma a conter informação suficiente acerca dos estados internos do sistema, tal como se referiu na Secção 2.4.1.

Uma vez que são utilizados modelos difusos, alguns aspectos do seu projecto são preestabelecidos (*vide* Secção 3.3). Assim, quanto à fuzificação, esta operação é efectuada com base no accionamento individual de regras. No que respeita à base de regras, o formato das regras aí definidas será tanto do tipo linguístico (consequentes difusos) como do tipo Takagi-Sugeno de ordem 0 e 1. Na base de dados, assume-se um universo de discurso contínuo não normalizado, utilizando-se funções de pertença Gaussianas (3.7) ou Gaussianas generalizadas (3.8). No que toca ao motor de inferência, utiliza-se accionamento individual de regras e inferência de Mamdani, sendo as conectivas difusas definidas tanto através de operadores de truncatura como algébricos. Finalmente, estabelece-se que a operação de desfuzificação, em sistemas linguísticos, é desempenhada pela modificação do método da altura, definida em (3.38) e (3.39), respectivamente para Gaussianas e Gaussianas generalizadas. Para sistemas de Takagi-Sugeno, utiliza-se o método definido em (3.40). As assunções referidas são apresentadas sucintamente na Tabela 5.1.

Em face dos pressupostos enunciados, o problema de identificação terá por objectivos, no capítulo actual, determinar os restantes parâmetros do projecto automático de sistemas difusos. Assim, em relação à base de regras, o problema central prende-se com aquilo que se designa vulgarmente por *aprendizagem da estrutura*. Este problema consiste na definição de um conjunto de regras e variáveis linguísticas a utilizar. O último ponto tem subjacente os aspectos associados à determinação da dimensão do modelo. De facto, na construção de um modelo de entrada-saída de um sistema dinâmico, as entradas e saídas passadas do sistema constituirão, tipicamente, variáveis de entrada a incluir no modelo, responsáveis pela incorporação de memória. Esta metodologia, i.e., utilização de linhas de atraso (Secção 2.4.3), é necessária, uma vez que as estruturas consideradas não dispõem de memória dinâmica.

---

[49] Embora sejam considerados sistemas dinâmicos, as técnicas descritas posteriormente constituem, fundamentalmente, métodos de mapeamento de um espaço de entrada num espaço de saída, pelo que são também aplicáveis a sistemas estáticos.

[50] *Bounded-Input Bounded-Output*, em terminologia inglesa.

MÓDULO	**PRESSUPOSTOS**
SISTEMA	➢ *Tipo:* - Dinâmico (e causal) - Discreto (contínuo discretizado) - Estável BIBO - Genericamente MIMO (ou mais simples) - Não linear - Invariante no tempo (especialmente, embora também sejam considerados sistemas variantes)
MODELO	➢ *Tipo:* - Equivalente ao sistema - Entrada-saída ➢ *Parametrização:* FARX
FUZIFICAÇÃO	➢ *Fuzificação baseada no accionamento individual de regras*
BASE DE REGRAS	➢ *Formato de regras:* - Linguísticas - Takagi-Sugeno (ordem 0 e 1)
BASE DE DADOS	➢ *Tipo de universo de discurso:* - Contínuo - Não normalizado
MOTOR DE INFERÊNCIA	➢ *Representação do conjunto de regras:* - Accionamento individual ➢ *Conectivas difusas:* - Operadores algébricos (produto, adição) - Operadores de truncatura (mínimo, máximo) ➢ *Método de inferência:* - Mamdani
DESFUZIFICAÇÃO	➢ *Método de desfuzificação:* - Método da altura modificado - Método para sistemas Takagi-Sugeno

Tabela 5.1. Pressupostos considerados na identificação de modelos difusos.

Na aprendizagem da estrutura incluem-se ainda os aspectos relativos à determinação do número de funções de pertença a definir para cada variável linguística.

No que respeita à *aprendizagem de parâmetros*, os objectivos a atingir resumem-se ao ajuste dos parâmetros das funções de pertença e dos parâmetros dos consequentes, no caso da utilização de sistemas do tipo Takagi-Sugeno. Esta tarefa é realizada com base em algoritmos de treino de redes neuronais.

Finalmente, após a derivação do modelo, há que proceder à sua validação. Assim, utilizar-se-á o critério RMSE (4.35) na verificação das capacidades de aproximação do modelo. Tal como se tem vindo a referir, uma das motivações fundamentais da modelização difusa é o desenvolvimento de modelos interpretáveis linguisticamente. Assim, é fundamental que a sintonização de parâmetros garanta a transparência do modelo, pelo que se impõem algumas restrições em relação aos algoritmos de aprendizagem (Secção 5.4.2). Deste modo, será possível atribuir termos linguísticos a cada uma das funções de pertença obtidas.

As tarefas a realizar e os objectivos de identificação enunciados são apresentados na Tabela 5.2.

TAREFA	**OBJECTIVOS**
APRENDIZAGEM DA ESTRUTURA	➢ *Base de regras:* - Selecção de variáveis linguísticas (dimensão do modelo) - Determinação de regras - Determinação do número de funções de pertença
APRENDIZAGEM DE PARÂMETROS	➢ *Base de dados:* - Sintonização de parâmetros das funções de pertença e consequentes do tipo Takagi-sugeno
VALIDAÇÃO	➢ *Capacidade de aproximação* - RMSE ➢ *Base de regras:* - Interpretabilidade: teste da possibilidade de atribuição de termos linguísticos às funções de pertença.

Tabela 5.2. Tarefas e objectivos na identificação de modelos difusos.

5.2. Aprendizagem da Estrutura

Tal como foi referido na secção precedente, a aprendizagem da estrutura envolve, no contexto da identificação difusa, a selecção de variáveis a incluir no modelo, e respectivas regressões, a determinação do número de funções de pertença por variável e a obtenção de um conjunto de regras condicionais difusas. A classe de métodos descritos nesta secção enquadra-se nas metodologias de aplicação fora de linha.

Inicialmente, serão analisados algoritmos que assumem a correcta determinação das variáveis a incluir no modelo. Assim, as tarefas a desempenhar são susceptíveis de serem realizadas por meio de técnicas diversas, algumas baseadas no treino de redes neuro-difusas com capacidade de aprendizagem da estrutura e outras baseadas em algoritmos de agrupamento de classes.

Relativamente à selecção das variáveis, uma vez que se trata de um problema de grande complexidade no contexto de sistemas não lineares, os algoritmos disponíveis são essencialmente heurísticos. Ainda assim, serão referidos os aspectos fundamentais a considerar na realização desta tarefa, sendo apresentado o algoritmo utilizado neste trabalho.

5.2.1. Aprendizagem Neuro-Difusa da Estrutura: a Rede NFCN

O método mais simples de determinação de uma estrutura difusa consiste, pura e simplesmente, na definição das variáveis a utilizar com base em conhecimento *a priori* sobre o sistema, atribuindo-se a cada uma o "número mágico" de funções de pertença, i.e., 7 funções para cada variável linguística (Secção 3.3.2). Assim, as regras a incluir no modelo terão por antecedentes a combinação dos termos linguísticos de cada variável de entrada. Exemplificando, um modelo difuso completo[51], com duas variáveis de entrada X_1, com termos linguísticos $LX1_1$, $LX1_2$ e $LX1_3$, e X_2, definida pelos valores $LX2_1$, $LX2_2$ e $LX2_3$, será composto por $3^2 = 9$ regras, cujos antecedentes serão constituídos como em (5.1). Em relação aos consequentes de cada regra, no caso dos elementos referidos serem do tipo difuso, o procedimento a seguir na sua definição não é evidente. Deste modo, requerem-se métodos para a sua determinação automática. Relativamente a modelos Takagi-Sugeno, os parâmetros dos consequentes são obtidos por meio de métodos de optimização linear, tal como será analisado na próxima secção.

Regra 1*:* SE (X_1 *é* $LX1_1$) *E* (X_2 *é* $LX2_1$) ENTÃO …
Regra 2*:* SE (X_1 *é* $LX1_1$) *E* (X_2 *é* $LX2_2$) ENTÃO …
Regra 3*:* SE (X_1 *é* $LX1_1$) *E* (X_2 *é* $LX2_3$) ENTÃO …
Regra 4*:* SE (X_1 *é* $LX1_2$) *E* (X_2 *é* $LX2_1$) ENTÃO …
Regra 5*:* SE (X_1 *é* $LX1_2$) *E* (X_2 *é* $LX2_2$) ENTÃO … (5.1)
Regra 6*:* SE (X_1 *é* $LX1_2$) *E* (X_2 *é* $LX2_3$) ENTÃO …
Regra 7*:* SE (X_1 *é* $LX1_3$) *E* (X_2 *é* $LX2_1$) ENTÃO …
Regra 8*:* SE (X_1 *é* $LX1_3$) *E* (X_2 *é* $LX2_2$) ENTÃO …
Regra 9*:* SE (X_1 *é* $LX1_3$) *E* (X_2 *é* $LX2_3$) ENTÃO …

Genericamente, o número de regras dependerá exponencialmente do número de funções de pertença de cada variável. Assim, denotando o número de termos linguísticos de cada variável X_j por $|T(X_j)|$, o número total de regras, g, será dado por (5.2):

$$g = \prod_{j=1}^{m} \left| T(X_j) \right| \tag{5.2}$$

Consequentemente, um sistema difuso com 4 entradas e 7 funções de pertença por entrada conterá $7 \times 7 \times 7 \times 7 = 7^4 = 2401$ regras! Este problema é vulgarmente designado por *explosão da base de regras*[52], decorrente da partição do tipo grelha, representada na Figura 5.1.

Contudo, é natural que grande parte das regras definidas sejam desnecessárias. Assim, é importante determinar as regras relevantes e eliminar as restantes. Deste modo, a base de regras não gozará da propriedade da plenitude sendo, contudo, consideravelmente mais simples, o que apresenta vantagens em termos de interpretabilidade e eficiência computacional.

No sentido da resolução dos problemas enunciados, em [Lin, 1995] é apresentada a estrutura neuro-difusa NFCN, utilizada na determinação de regras e inicialização de funções de pertença dos antecedentes e consequentes, em sistemas difusos linguísticos. Este método será descrito nos

[51] Um modelo difuso diz-se completo se a sua base de regras for completa, i.e., se todas as regras possíveis estiverem definidas.
[52] Em terminologia inglesa utiliza-se frequentemente a expressão *"curse of dimensionality"*.

parágrafos seguintes.

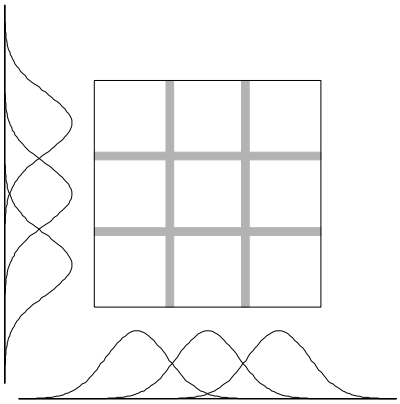


Figura 5.1. Partição do espaço de entrada-saída em grelha.

Estrutura da rede Neural Fuzzy Control Network (NFCN)

A estrutura NFCN tem por objectivo a implementação de um sistema difuso por meio de uma rede neuronal, englobando aspectos de aprendizagem da estrutura e de parâmetros. Nesta secção será abordado o primeiro aspecto, sendo a aprendizagem de parâmetros o tema da secção seguinte. Assim, a Figura 5.2 representa uma possível arquitectura da rede NFCN, resultante da tarefa da aprendizagem da estrutura de um sistema difuso[53]. Por simplicidade, a figura referida representa um sistema com duas entradas, duas saídas e três funções de pertença por variável linguística.

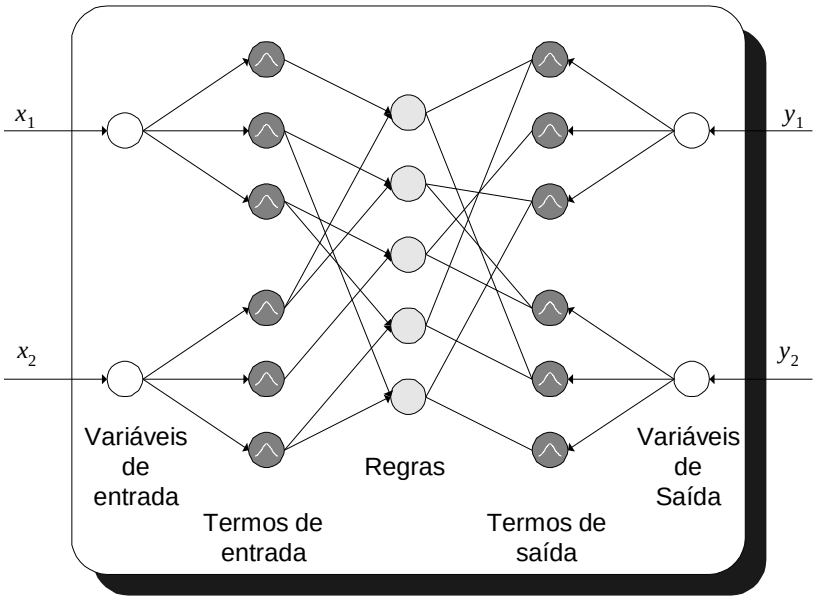


Figura 5.2. Aprendizagem da estrutura na rede NFCN.

[53] A rede apresentada constitui um caso particular da arquitectura NFCN, desenvolvida especificamente para

A rede da figura referida é constituída por cinco camadas, as quais integram os elementos básicos de um sistema difuso, i.e., entradas, funções de pertença, regras condicionais, conectivas difusas e desfuzificação.

Os neurónios da *primeira camada* representam as variáveis de entrada. A função desta camada é unicamente receber os sinais do ambiente exterior e passá-los às camadas posteriores, as quais realizam trabalho útil.

Na *segunda camada* são definidos os *termos linguísticos* associados a cada variável de entrada. Deste modo, as unidades desta camada estão ligadas às entradas correspondentes, sendo o peso de cada ligação constituído pelos parâmetros das funções de pertença utilizadas, neste caso Gaussianas (5.3):

$$a_i^{(p2)} = e^{-\frac{\left(x_j^{(p)} - c_{ij}\right)^2}{2s_{ij}^2}} \quad , i = 1,2,\cdots, n_{fpi} \tag{5.3}$$

em que $a_i^{(p2)}$ representa a activação do neurónio i da camada 2 em relação ao p-ésimo padrão de entrada $x_j^{(p)}$, sendo c_{ij} e s_{ij}, respectivamente, o centro e o desvio padrão da i-ésima Gaussiana associada à entrada j, os quais constituem os parâmetros ajustáveis. Na mesma expressão, n_{fpi} representa o número total de funções de pertença associadas às variáveis de entrada (5.4):

$$n_{fpi} = \sum_{j=1}^{m} \left| T\{X_j\} \right| \tag{5.4}$$

Assim, as células desta camada são responsáveis pelo cálculo do grau de pertença de cada entrada numérica relativamente a cada um dos termos linguísticos.

A *terceira camada* constitui a *camada de regras*. Aqui, cada neurónio representa uma regra condicional difusa. Assim sendo, cada célula na camada presente interliga termos linguísticos de diferentes variáveis de entrada, os quais constituirão o antecedente da regra em causa. Deste modo, cada neurónio desta camada tem por função de activação uma norma-T, a qual corresponde, no desenvolvimento original, ao operador mínimo (5.5):

$$a_r^{(p3)} = norma_{i=1}^{na_r} - T\left(a_i^{(p2)}\right) = \min_{i=1}^{na_r}\left(a_i^{(p2)}\right) \quad , r = 1,2,\cdots, g \tag{5.5}$$

em que $a_r^{(p3)}$ representa a activação da r-ésima regra e na_r designa o número de entradas que constituem o antecedente da regra r. Neste trabalho, utiliza-se também o operador produto.

A quarta e quinta camadas desempenham um papel em tudo idêntico ao das camadas dois e um, respectivamente. De facto, na aprendizagem da estrutura, a quinta camada funciona como uma *"camada de entrada"*, a qual recebe os sinais de saída, passando-os à quarta camada, a *camada de termos linguísticos de saída*. Assim sendo, esta última é responsável pelo cálculo do grau de pertença dos valores numéricos de saída em cada um dos termos linguísticos. Tal como se passava com a segunda camada, utilizam-se funções Gaussianas sendo a activação dos neurónio respectivos dada por (5.6):

$$a_o^{(p4)} = e^{-\frac{\left(y_j^{(p)} - c_{oj}\right)^2}{2s_{oj}^2}} \quad , o = 1,2,\cdots, n_{fpo} \tag{5.6}$$

a aprendizagem da estrutura. A situação de implementação efectiva de um sistema difuso é apresentada na secção de aprendizagem de parâmetros, Secção 5.3.

em que $a_o^{(p4)}$ denota a activação do neurónio *o* da camada 4 em relação ao *p*-ésimo padrão de saída $y_j^{(p)}$, sendo c_{oj} e s_{oj}, respectivamente, o centro e o desvio padrão da *o*-ésima Gaussiana associada à saída *j*. Ainda na expressão (5.6), n_{fpo} representa o número total de funções de pertença associadas às variáveis de saída (5.7):

$$n_{fpo} = \sum_{j=1}^{n} \left| T(Y_j) \right| \tag{5.7}$$

Estrutura inicial da rede

Inicialmente, as ligações entre a camada de termos linguísticos de entrada e a camada de regras são completas, i.e., cada neurónio regra consiste na combinação de termos linguísticos de entrada, de forma a que se obtenham todas as combinações possíveis. Do enunciado, resulta que o número total de regras será dado, inicialmente, pela expressão (5.2). Durante a aprendizagem, alguns neurónios da camada de regras serão eliminados ou combinados, de forma a obter-se a estrutura final da rede e do sistema difuso por ela representado. Do mesmo modo, a camada de regras e a camada de termos linguísticos de saída, estão, no início, completamente ligadas, o que revela o desconhecimento em relação aos consequentes que se deverão associar a cada regra. A correcta atribuição de consequentes a cada uma das regras constitui outro dos objectivos a atingir na aprendizagem da estrutura. Na Figura 5.3 apresenta-se a estrutura inicial da rede neuro-difusa.

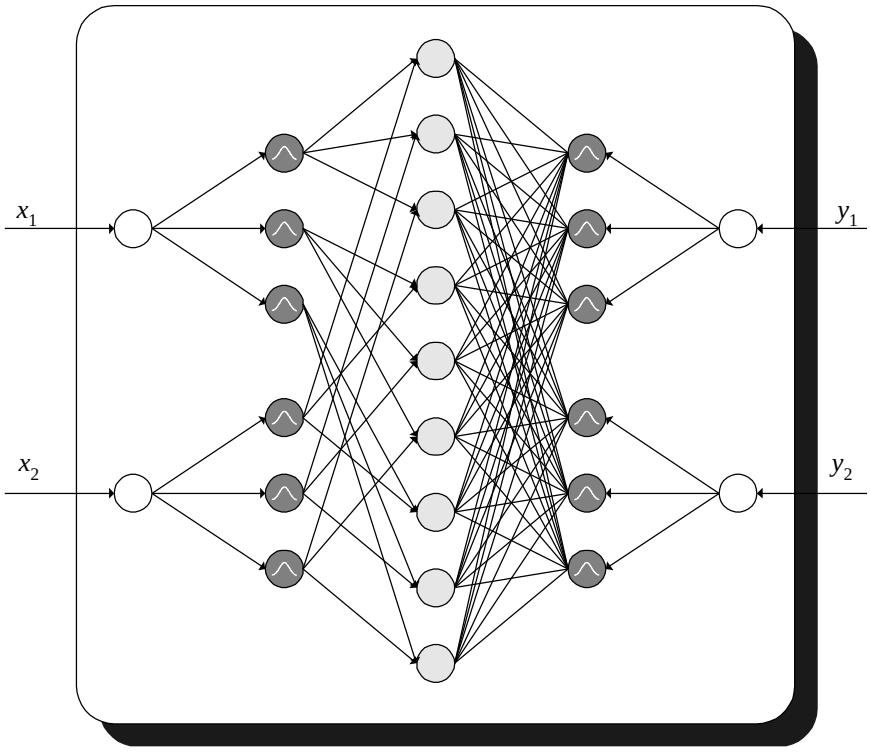


Figura 5.3. Estrutura inicial da rede NFCN.

Algoritmo de aprendizagem da estrutura

A aprendizagem da estrutura na rede NFCN enquadra-se na classe de algoritmos de aprendizagem auto-organizada. O problema geral pode ser enunciado do seguinte modo: dados um conjunto de entradas X_j, j = 1,2,…, m, um conjunto de saídas Y_o, o = 1,2,…,n, as partições difusas associadas a cada variável, $|T(X_j)|$ e $|T(Y_o)|$, e as formas das funções de pertença (Gaussianas), o

objectivo a atingir reside na determinação de funções de pertença iniciais, com base nas quais serão identificadas as regras difusas relevantes.

Assim, em primeiro lugar, os centros e desvios padrões de cada uma das funções de pertença Gaussianas[54] de entrada e saída são determinados por técnicas de aprendizagem auto-organizada. Deste modo, as funções de pertença serão distribuídas pelas áreas mais densas do domínio de cada variável, i.e., pelas áreas onde haja mais amostras. Nesta situação, utiliza-se o algoritmo de Kohonen (*Kohonen's feature maps*) [Kohonen, 1989] na procura do centro de cada função de pertença. O algoritmo referido é em tudo idêntico ao algoritmo de agrupamento neuronal *k-means clustering* [Moody e Darken, 1989].

Deste modo, para cada uma das variáveis de entrada e saída, e de forma independente, o algoritmo começa por determinar o centro *k* mais próximo de um determinado padrão de treino (5.8):

$$k: \quad \forall_{i=1,2,\cdots,|T(X_j)|}: \quad \left|c_{kj} - x_j^{(p)}\right| \le \left|c_{ij} - x_j^{(p)}\right| \tag{5.8}$$

A expressão anterior determina o centro mais próximo, c_{kj}, do padrão de entrada $x_j^{(p)}$, procedendo-se do mesmo modo para os padrões de saída $y_o^{(p)}$. Assim, o centro mais próximo é *deslocado* em direcção ao padrão de entrada, tal como se segue (5.9):

$$\Delta c_{kj} = \eta(t)\left(x_j^{(p)} - c_{kj}\right) \tag{5.9}$$

em que $\eta(t)$ representa uma velocidade de aprendizagem monótona decrescente. De facto, os padrões de treino são apresentados na totalidade, sequencialmente, durante um número de épocas predefinido. Assim sendo, no final de cada época a velocidade de aprendizagem diminui, geralmente de forma exponencial, de acordo com (5.10):

$$\eta(t) = dr \cdot \eta(t-1) \quad , \eta(t) \in]0;1] \tag{5.10}$$

onde *t* designa o número da época e *dr* (*decay rate*) representa um factor de diminuição da velocidade de aprendizagem, ao qual se atribuem geralmente valores próximos de 0.9. A motivação fundamental para a utilização de uma taxa de aprendizagem decrescente resulta da necessidade de que os ajustes efectuados numa iteração não se sobreponham totalmente aos das iterações anteriores, como resultado da apresentação sequencial dos padrões de treino. Nesse caso, as propriedades de convergência do algoritmo seriam deficientes. Quanto à inicialização da velocidade de aprendizagem, Haykin [Haykin, 1994] sugere um valor próximo de 1.

Em relação aos centros derrotados, os seus valores mantêm-se inalterados (5.11):

$$\Delta c_{kj} = 0 \quad , i \neq k \tag{5.11}$$

Após a aplicação do algoritmo durante um determinado número de épocas, os desvios padrões das Gaussianas são atribuídos de acordo com a heurística do *primeiro vizinho mais próximo* [Moody e Darken, 1989], tal como se segue (5.12):

$$\sigma_{ij} = \frac{\left|c_{ij} - c_{kj}\right|}{\sigma} \tag{5.12}$$

---

[54] No caso presente são utilizadas funções de pertença Gaussianas. No entanto, o algoritmo descrito é facilmente generalizável para qualquer outro tipo de funções de base radial.

Aqui, o desvio padrão de uma dada função de pertença é determinado pela distância do seu centro ao centro da função mais próxima, sendo a constante *s* um parâmetro de sobreposição.

No caso de se utilizarem funções Gaussianas generalizadas (3.8), a determinação do desvio padrão de cada função é efectuada com base nas funções vizinhas mais próximas à direita e à esquerda (5.13):

$$s_{ijR} = \frac{\left| c_{ijR} - c_{kjL} \right|}{s}$$
$$s_{ijL} = \frac{\left| c_{ijL} - c_{kjR} \right|}{s} \qquad (5.13)$$

Na expressão anterior, assume-se que a procura de centros define $c_{ijR} = c_{ijL}$, para todas as funções de pertença. A sua determinação final e distinta é conduzida pelo algoritmo de aprendizagem de parâmetros. É importante realçar que a utilização de funções assimétricas permite um grau de sobreposição constante entre todos os pares de funções de pertença, o que não acontece com funções simétricas (*vide* Figura 3.7).

Após a determinação dos parâmetros das funções de pertença, procede-se à *selecção de consequentes* e *eliminação de regras desnecessárias*. Neste sentido, os sinais de ambas as camadas externas da rede são apresentados. Assim, os sinais de entrada fluem pela primeira camada, sendo propagados para a camada de termos linguísticos de entrada e daí para a camada de regras. Quanto aos sinais de saída, estes entram pela quinta camada e são propagados para a camada de termos linguísticos de saída. Assim, com base na activação de cada regra, $a_r^{(p3)}$, e na activação dos neurónios da camada de termos linguísticos de saída, $a_o^{(p4)}$, os consequentes correctos a atribuir a cada regra são determinados com base no ajuste das ligações entre as unidades da terceira e da quarta camada.

A aprendizagem dos pesos referidos é efectuada com base no algoritmo de aprendizagem competitiva definido em [Kosko, 1992]. Assim, tal como se referiu anteriormente, as camadas em causa encontram-se, no início, complemente ligadas. Denotando por w_{or} o peso da ligação entre o neurónio *r* da camada 3 com o neurónio *o* da camada 4, a sua adaptação é conduzida pela expressão (5.14):

$$\Delta w_{or} = a_o^{(p4)} \cdot \left(a_r^{(p3)} - w_{or} \right) \qquad (5.14)$$

Inicialmente, atribui-se aos pesos o valor 0. Assim, a ideia da expressão anterior é fortalecer as ligações nos casos em que uma regra e um consequente sejam simultaneamente activados. Exemplificando, se a regra *r* for activada com o valor 0.5 e o consequente *o* não for activado (valor 0), o peso da ligação entre as duas unidades mantém-se. Por outro lado, se a activação do consequente for não nula, o peso da ligação variará, aumentando ou diminuindo. Assim, supondo $w_{or} = 0.7$ e $a_o^{(p4)} = 0.6$, obter-se-á $\Delta w_{or} = 0.6 \cdot (0.5 - 0.7) = -0.12$. Este resultado deriva do facto da activação simultânea da regra e do consequente em causa ter sido inferior à das ocorrências verificadas na apresentação dos padrões anteriores. Supondo agora que a activação da regra é mais elevada, e.g., 0.8, vem $\Delta w_{or} = 0.6 \cdot (0.8 - 0.7) = 0.06$. Assim, a ligação é fortalecida em virtude dos valores elevados da activação da regra e do consequente. Ainda que a saída do consequente fosse baixa, a ligação seria fortalecida, embora com um valor inferior, tal como é desejável.

Após a apresentação de todo o conjunto de treino, os pesos das ligações entre as células da camada de regras e da camada de consequentes corresponderão à importância da atribuição de um determinado consequente a uma dada regra. Assim sendo, de entre todas as ligações entre uma regra e os consequentes de cada variável de saída, mantém-se unicamente a mais forte,

eliminando-se as restantes. Deste modo, a ligação final corresponderá ao consequente seleccionado para a regra em causa. No entanto, no caso dos pesos de todas as ligações serem bastante pequenos, inferiores a um limiar definido, não faz sentido atribuir qualquer consequente à regra considerada. Deste modo, todas as ligações são eliminadas, o que significará que a regra é desnecessária, sendo, então, também ela, eliminada. Os aspectos referidos são ilustrados na Figura 5.4.

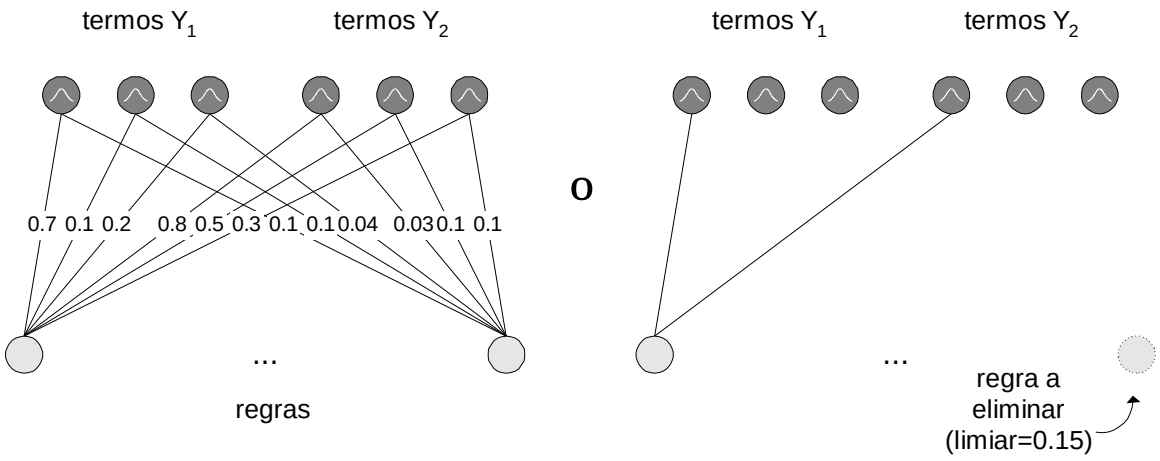


Figura 5.4. Selecção de consequentes e eliminação de regras.

A eliminação de regras conduz, eventualmente, à eliminação de funções de pertença dos antecedentes. De facto, no caso de todas as ligações entre um determinado termo de entrada e a camada de regras corresponderem a regras eliminadas, o neurónio respectivo será removido. Do mesmo modo, termos linguísticos de saída que não sejam incluídos no consequente de qualquer regra serão também eliminados. Em última análise, se todas as funções de pertença de uma variável forem eliminadas, a variável respectiva será removida do modelo.

Após a selecção de consequentes e regras, o sistema difuso representado pela rede neuronal poderá ser ainda mais simplificado através da *combinação de regras*. Neste caso, são estabelecidos alguns critérios com base nos quais a decisão quanto à combinação de um conjunto de regras numa única é tomada, os quais se passam a enunciar:

i) todas as regras a combinar apresentam os mesmos consequentes;
ii) algumas proposições no antecedente são comuns a todas as regras no conjunto;
iii) a união dos termos linguísticos utilizados nas restantes proposições contém todo o conjunto de termos das variáveis linguísticas em questão.

No caso dos três critérios enunciados se verificarem, poder-se-á definir uma nova regra com as proposições comuns no antecedente, a qual substituirá o conjunto de regras em consideração. Exemplificando, no conjunto de regras (5.15):

$$\begin{aligned} &\textit{Regra } 1\textit{:} \quad \text{SE } (X_1 \textit{ é } LX1_1) \textit{ E } (X_2 \textit{ é } LX2_1) \textit{ E } (X_3 \textit{ é } LX3_1) \text{ ENTÃO } Y \textit{ é } LY_1 \\ &\textit{Regra } 2\textit{:} \quad \text{SE } (X_1 \textit{ é } LX1_1) \textit{ E } (X_2 \textit{ é } LX2_1) \textit{ E } (X_3 \textit{ é } LX3_2) \text{ ENTÃO } Y \textit{ é } LY_1 \\ &\textit{Regra } 3\textit{:} \quad \text{SE } (X_1 \textit{ é } LX1_1) \textit{ E } (X_2 \textit{ é } LX2_1) \textit{ E } (X_3 \textit{ é } LX3_3) \text{ ENTÃO } Y \textit{ é } LY_1 \end{aligned} \tag{5.15}$$

verifica-se que todas as regras apresentam o mesmo consequente, Y *é* LY_1, as proposições (X_1 *é* $LX1_1$) e (X_2 *é* $LX2_1$) são comuns a todos os antecedentes e, sabendo que a variável X_3 tem associados 3 termos linguísticos, $LX3_1$, $LX3_2$ e $LX3_3$, as restantes proposições utilizam a totalidade

dos termos linguísticos. Assim sendo, as três regras expressas em (5.15) podem ser combinadas numa única, tal como se segue (5.16):

$$\textit{Regra}\ 1\text{:}\ \text{SE}\ (X_1\ \textit{é}\ LX1_1)\ E\ (X_2\ \textit{é}\ LX2_1)\ \text{ENTÃO}\ Y\ \textit{é}\ LY_1 \tag{5.16}$$

O exemplo de combinação descrito é apresentado esquematicamente na Figura 5.5.

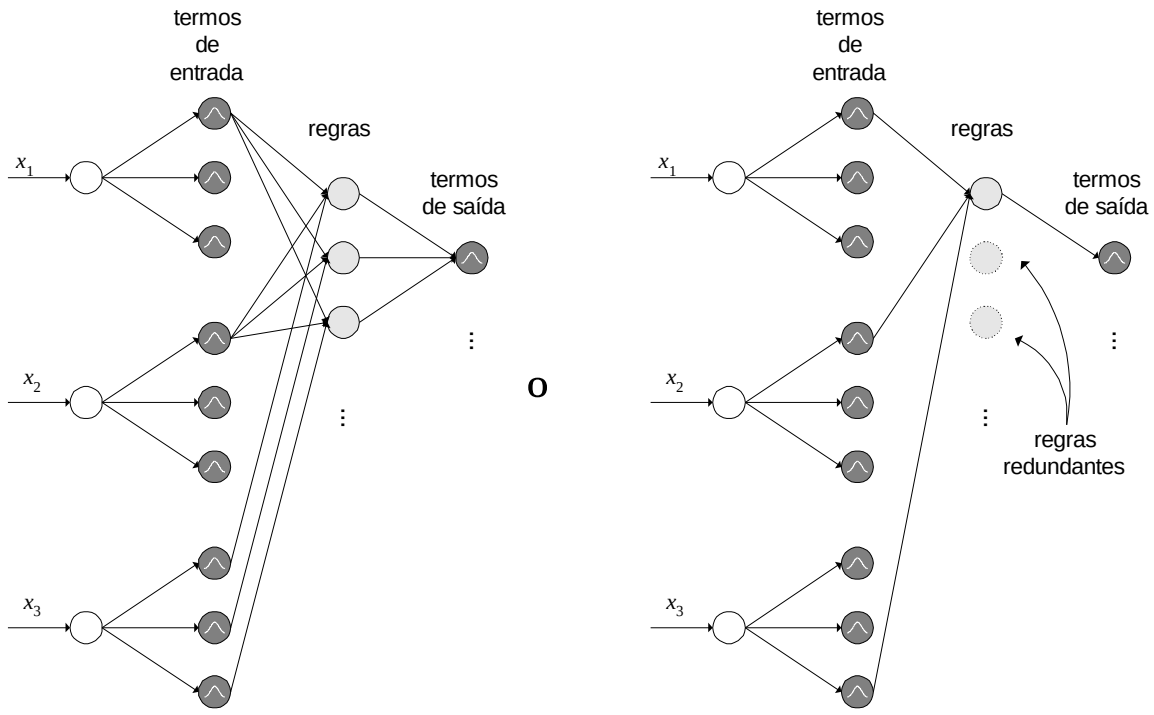


Figura 5.5. Combinação de regras.

A Tabela 5.3 resume o algoritmo de aprendizagem da estrutura na arquitectura NFCN.

1. Aprendizagem de centros e larguras; 2. Aprendizagem dos pesos das ligações entre a camada de regras e a camada de consequentes; 3. Selecção de consequentes e eliminação de regras; 4. Combinação de regras

Tabela 5.3. Algoritmo de aprendizagem da estrutura na arquitectura NFCN.

Análise do algoritmo

O algoritmo de aprendizagem da estrutura na rede NFCN apresenta aspectos interessantes no sentido da simplificação de uma base de regras, nomeadamente em termos de eliminação de regras e termos linguísticos desnecessários.

No método apresentado, é necessário determinar previamente o número de termos linguísticos de cada variável de entrada e saída, o que nem sempre é trivial. Deste modo, utiliza-se habitualmente um número entre 5 e 9 funções de pertença por variável linguística. Para além da determinação dos termos linguísticos, o algoritmo nada refere em relação à selecção de variáveis de

entrada, pressupondo a sua definição com base em conhecimento *a priori*.

Outro aspecto importante deriva do mecanismo de atribuição de larguras a cada função. A heurística utilizada, a dos primeiros vizinhos mais próximos, não garante a escolha ideal dos desvios padrões das Gaussianas. No entanto, esta selecção é extremamente importante para a precisão dos resultados obtidos. Uma hipótese consistiria na optimização prévia das funções de pertença. No entanto, este procedimento não é viável na maior parte das aplicações práticas, em virtude dos problemas decorrentes do particionamento em grelha. De facto, o treino de uma rede com 2401 regras (exemplo referido anteriormente) tornar-se-ia extremamente moroso.

Adicionalmente, o processo de eliminação e combinação de regras é, em geral, lento, o que contribui também para a ineficiência computacional do algoritmo.

Existem outros algoritmos de eliminação de regras baseados em técnicas de poda de redes neuronais, algumas do tipo "força bruta", i.e., baseadas na eliminação iterativa de regras e análise do desempenho da rede, outras derivadas da análise da sensibilidade da rede face à remoção de diferentes elementos e ainda outras com base na adição de termos penalizadores, que levam a rede a associar pesos nulos aos neurónios desnecessários. Em [Reed, 1993] é apresentada uma abordagem geral dos métodos referidos, os quais não serão discutidos neste trabalho.

Em oposição aos métodos de eliminação de regras encontram-se métodos heurísticos, os quais se baseiam na adição iterativa de regras conforme a adequação da rede aos padrões apresentados iterativamente. Neste sentido, alguns autores [Juang e Lin, 1998; Cho e Wang, 1996] propuseram algoritmos designados por *métodos de aprendizagem construtiva*, adequados a aprendizagem em linha, os quais se afiguram bastante promissores. No entanto, tal como se referiu, trata-se de metodologias essencialmente heurísticas, o que denota o seu elevado grau de imaturidade no momento presente.

Dos aspectos expostos resulta a necessidade de utilizar outros métodos de aprendizagem de estrutura. Neste sentido, os métodos de agrupamento de classes afiguram-se particularmente interessantes.

5.2.2. Agrupamento de Classes: Agrupamento Subtractivo

Tal como se referiu no ponto anterior, o particionamento em grelha do espaço de entrada-saída é susceptível de conduzir à explosão da base de regras. Neste sentido, a utilização de técnicas de agrupamento de classes possibilita um particionamento mais disperso, resultando num menor número de regras. A Figura 5.6 ilustra os aspectos referidos num espaço bidimensional.

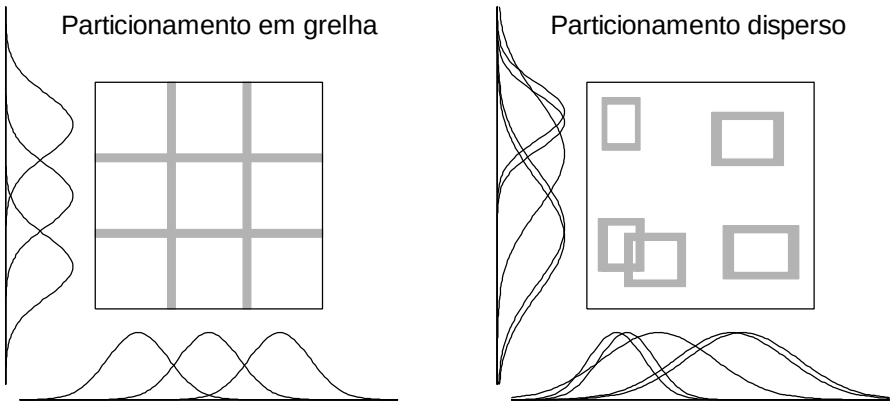


Figura 5.6. Partições difusas num espaço bidimensional.

Apesar das vantagens do particionamento resultante da aplicação de métodos de agrupamento de classes, a figura anterior deixa, desde já, transparecer uma das suas limitações. Na verdade, verifica-se que as técnicas de agrupamento originam, em geral, funções de pertença bastante similares, o que limita a interpretabilidade do modelo difuso obtido. Estas e outras questões serão analisadas posteriormente.

O problema geral do agrupamento de classes

Basicamente, os algoritmos de agrupamento têm por objectivo particionar as amostras de dados num conjunto de grupos naturais[55]. De outro modo, os algoritmos referidos abordam o problema da *extracção de características* significativas em termos da organização estrutural dos dados. Na situação em que os dados são etiquetados, o problema do agrupamento é efectuado trivialmente, de maneira supervisionada. No entanto, na generalidade dos problemas, entre os quais a identificação difusa[56], a procura de grupos naturais é efectuada de forma não supervisionada, constituindo um problema de elevada complexidade.

Assim, o esquema geral de funcionamento dos algoritmos de agrupamento é o seguinte: dado um conjunto de N amostras de dados, Z^N (2.2), pretende-se encontrar um número g de grupos, $g \in [1, N]$, exibindo características homogéneas [Bezdek, 1981]. O caso em que $g = 1$ equivale à inexistência de grupos nos dados, sendo $g = N$, o caso trivial em que cada amostra é utilizada para descrever um grupo, pelo que, tipicamente, o intervalo é aberto.

O ponto fundamental sobre o qual assenta todo o mecanismo de procura de grupos reside na selecção de um *critério de agrupamento*. Obviamente, a identificação de grupos presentes nos dados deve ser levada a cabo com base nas propriedades das amostras recolhidas: distância, ângulo, curvatura, simetria, forma, etc. Independentemente do critério seleccionado, a complexidade da grande maioria dos problemas leva a que nenhum critério se aplique a todos os tipos de problemas, pelo que a sua selecção é sempre subjectiva e, como tal, questionável.

Idealmente, seria desejável que os grupos naturais presentes nos dados fossem facilmente identificáveis, sendo, portanto, compactos, bem separados e com dimensões idênticas. No entanto, em situações reais os dados apresentam características diversas em termos de forma (esférica, elíptica, rectangular), dimensão e geometria (linear, curva). A Figura 5.7 ilustra os aspectos referidos num espaço bidimensional.

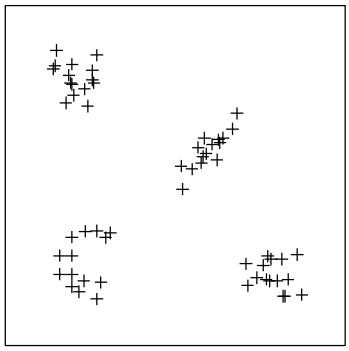


Figura 5.7. Exemplos de distribuições de dados a agrupar.

[55] *Clusters*, em terminologia inglesa.

De acordo com o algoritmo utilizado, cuja selecção é sempre subjectiva, diferentes soluções são obtidas. Deste modo, a questão fundamental que se coloca relaciona-se com a *validação* da solução obtida.

Assim, o número de grupos presentes nos dados não é, em geral, previamente conhecido. Deste modo, é importante identificar o valor mais adequado para *g*, o que nem sempre é trivial. A resolução deste problema torna-se ainda mais complexa na presença de dados com ruído. De facto, a correcta determinação do número de grupos permanece um problema em aberto. Assim sendo, a solução clássica enquadra-se nas técnicas do tipo "força bruta". Aqui, o método consiste no teste de vários valores numa determinada gama, avaliando-se a qualidade dos resultados obtidos em cada um dos casos. Esta abordagem apresenta, obviamente, algumas limitações, sendo a mais óbvia a sua ineficiência computacional. Por outro lado, tal como se verificará, em algoritmos de optimização é comum atingirem-se óptimos locais, pelo que nada garante que a partição obtida para um determinado número de grupos origine os melhores resultados possíveis. Deste modo, desenvolveram-se outras estratégias baseadas em remoção e fusão de grupos. Assim, no *agrupamento progressivo*, o número de grupos é sobrespecificado, sendo encontrados iterativamente os grupos satisfatórios. Na *fusão de grupos compatíveis*, é acrescentado ainda um passo de fusão de grupos. Um dos problemas associados às técnicas referidas resulta da necessidade de definir medidas de validação de grupos individuais, e não da partição como um todo. Assim sendo, é prática corrente definir o número de grupos pela estratégia da "força bruta", o que se verificará neste trabalho. Em [Davé e Krishnapuram, 1997], estes e outros problemas são analisados com algum detalhe.

Algoritmo de agrupamento das *c-médias difusas*: breve descrição

De entre a grande diversidade de algoritmos de agrupamento de classes propostos ao longo do tempo [Davé e Krishnapuram, 1997], o algoritmo das *c-médias difusas* (FCM)[57] [Bezdek, 1981] é, porventura, o mais utilizado. Alguns aspectos deste método serão descritos, com o intuito de enquadrar o algoritmo utilizado neste trabalho. Para uma exposição detalhada, *vide* [Bezdek, 1981].

O algoritmo FCM, o qual se enquadra na classe de *métodos baseados em protótipos* [Davé e Krishnapuram, 1997], consiste numa extensão do algoritmo clássico de agrupamento ISODATA, ou *c-médias crespas*. No algoritmo difuso, dados o conjunto de *N* amostras de dados a particionar, Z^N, e o número g[58] de grupos a formar, determinam-se os protótipos de cada grupo, *C*, e uma matriz de partição *U*, a qual contém os graus de pertença de cada ponto em cada grupo (5.17):

$$(Z^N, g) \xrightarrow{c-médias\ difusas} (C, U)$$

$$C = [c_{ri}] \quad , i = 1,2,\cdots, m+n; \quad r = 1,2,\cdots, g \tag{5.17}$$

$$U = [m_{jr}] \quad , j = 1,2,\cdots, N$$

onde $m+n$ equivale à dimensão do espaço multidimensional de entrada-saída (m entradas e n saídas). Uma vez que cada ponto é definido num espaço de dimensão $m+n$, cada protótipo

[56] Naturalmente, no contexto de identificação difusa, as características significativas presentes nos dados tomam a forma de regras difusas.
[57] *Fuzzy C-Means*, em terminologia inglesa.
[58] Por uma questão de uniformidade, optou-se por utilizar a designação *g* para o número de grupos e não *c* (*cluster*), tal como no desenvolvimento original do algoritmo, de onde vem o nome *c-médias*.

consistirá também num vector de dimensão $m+n$, $c_r = [c_{r1}, c_{r2}, \ldots, c_{r(m+n)}]$, onde c_{ri} denota o centro associado à *i*-ésima variável no *r*-ésimo grupo. Ainda na expressão (5.17), m_{jr} representa o grau de pertença da *j*-ésima amostra no *r*-ésimo grupo. O centro referido é também designado por *protótipo*, daí a designação da classe.

A matriz de partição *U* estabelece a distinção fundamental entre o algoritmo crespo e o difuso. Assim, no algoritmo original, os elementos m_{jr} da matriz de partição apresentam valores binários, 0 ou 1, denotando a pertença ou não a um determinado grupo. No algoritmo difuso, os mesmos elementos podem apresentar valores entre 0 e 1, de acordo com a natureza difusa do algoritmo. Neste caso, os valores da matriz de partição são sujeitos a algumas restrições [Bezdek, 1981].

Os algoritmos difusos baseados em protótipos baseiam-se na minimização de um critério de erro quadrático, tal como se segue (5.18):

$$J(C,U;X) = \sum_{r=1}^{g}\sum_{j=1}^{N}(m_{jr})^{m'} d^2(z_j,c_r) \quad , m' \in [1,\infty[\tag{5.18}$$

em que $d^2(z_j,c_r)$ denota a distância Euclidiana do *r*-ésimo protótipo, c_r, e a *j*-ésima amostra, $z_j = [z_{j1}, z_{j2}, \ldots, z_{j(m+n)}]$. O parâmetro *m'* representa o grau de difusidade do processo de classificação (*m'* = 1 equivale ao algoritmo crespo). Cada um dos protótipos representa uma característica do sistema em causa, a qual pode ser definida como uma regra na identificação difusa. No algoritmo FCM, cada protótipo é, simplesmente, o centro de um grupo.

A optimização da função objectivo apresentada é efectuada iterativamente. Basicamente, em cada passo do algoritmo calculam-se os centros de cada grupo e actualiza-se a matriz de partição. O algoritmo termina quando a variação da norma da matriz de partição entre duas iterações for suficientemente pequena.

Tal como se depreende da expressão (5.18), a função objectivo a minimizar baseia-se na distância entre cada ponto e cada protótipo, procurando minimizar a distância entre pontos do mesmo grupo. O critério enunciado é do tipo gradiente, apresentando, por conseguinte, as limitações decorrentes do método referido, i.e., possibilidade de obtenção de óptimos locais e tempos de convergência elevados, além da elevada sensibilidade ao ruído. Um outro aspecto negativo advém da inicialização da matriz de partição, a qual influenciará a qualidade final da solução obtida, tal como acontece com a generalidade dos algoritmos de optimização.

Apesar do exposto, duas outras razões se mostraram determinantes no sentido da procura de outros algoritmos, as quais se prendem com a determinação de funções de pertença e com o enquadramento do método no contexto da identificação neuro-difusa.

Assim, a primeira questão que se coloca relaciona-se com o *modo de obtenção de funções de pertença* a partir dos protótipos e da matriz de partição obtidos. A maneira clássica de o fazer consiste em projectar os grupos no domínio de cada variável. A dificuldade desta metodologia reside no facto de não se obterem directamente funções de pertença numa forma fechada. Assim, em [Babuška e Setnes, 1998], as projecções obtidas são aproximadas por funções paramétricas, e.g., Gaussianas. No entanto, a projecção efectuada pode ocasionar alguma perda de informação.

Um outro aspecto a considerar, deriva da utilização de modelos do tipo Takagi-Sugeno. Nesta situação, o algoritmo FCM apresenta algumas limitações, uma vez que não permite determinar directamente os termos dos consequentes. De facto, o algoritmo FCM permite encontrar grupos esféricos (hiperesferas) no espaço de dados e não hiperplanos, como é requerido pelos modelos Takagi-Sugeno de ordem 1. Assim, o algoritmo de *agrupamento Gustafson-Kessel* [Gustafson e Kessel, 1979], uma variante das c-médias difusas, é preferível nestes casos, dado

possibilitar a determinação de consequentes do tipo Takagi-Sugeno pela procura de hiperplanos no espaço de dados. Apesar deste algoritmo ser mais adequado, a sua utilização não é determinante, uma vez que os parâmetros dos consequentes podem ser facilmente determinados por algoritmos de optimização linear, tal como será abordado na Secção 5.3. Em relação às restantes limitações apontadas ao algoritmo FCM, estas mantêm-se no algoritmo Gustafson-Kessel.

Quanto ao enquadramento do método no contexto neuro-difuso em que seria aplicado, a sua utilização originaria alguma redundância. De facto, após a aprendizagem da estrutura do modelo difuso, os parâmetros das funções de pertença são optimizados (tal como se discutirá na Secção 5.3), pelo que a implementação do algoritmo FCM com posterior optimização de parâmetros não se afigura uma escolha coerente. Deste modo, a utilização de um algoritmo mais leve a nível computacional e que possibilitasse a obtenção de uma base de regras bem como funções de pertença iniciais seria preferível. É, pois, neste contexto, que se apresenta, o algoritmo de agrupamento subtractivo, o qual consiste numa variação do método da montanha.

Método da Montanha

A classe de *métodos de função de potencial* [Davé e Krishnapuram, 1997], na qual se insere o método da montanha, assenta numa filosofia distinta da seguida pelos algoritmos baseados em protótipos.

Nos métodos referidos, define-se um conjunto de pontos como possíveis centros de grupo, sendo, cada um deles, visualizado como uma fonte de energia. Assim, o potencial gerado por cada um dos candidatos, p_i, é máximo no próprio ponto, decrescendo com a distância. Deste modo, as funções de potencial típicas são do tipo radial, e.g., Gaussianas, em tudo semelhantes às funções de pertença utilizadas na representação de conjuntos difusos (5.19):

$$P(p_i, z_j) = e^{-a\left\|p_i - z_j\right\|^2} \quad , i = 1,2,\cdots,nc; \quad j = 1,2,\cdots,N \tag{5.19}$$

onde o parâmetro a determina a área de influência de cada centro, nc denota o número de candidatos a centros, z_j representa uma das N amostras de dados (5.18) e $\|\cdot\|$ designa a distância Euclidiana. Assim, o potencial total associado a cada candidato pode ser definido como o somatório do potencial resultante da sua vizinhança em relação a todas as amostras (5.20):

$$P(p_i, Z^N) = \sum_{j=1}^{N} e^{-a\left\|p_i - z_j\right\|^2} \quad , i = 1,2,\cdots,nc \tag{5.20}$$

Embora o método da função de potencial não tenha sido proposto originalmente como um algoritmo de agrupamento, a função de potencial definida pode ser utilizada como uma função objectivo. Definindo o potencial de cada candidato como em (5.20), obter-se-á uma função de potencial total, na qual os picos correspondem a protótipos e os vales correspondem a fronteiras de decisão entre grupos. Assim, é fácil concluir que o potencial será mais elevado em zonas densamente povoadas. Exemplificando, assuma-se, num problema bidimensional, um particionamento em grelha do espaço de entrada-saída, tal como se apresenta na Figura 5.8 (adaptada de [Yager e Filev, 1994]).

Na figura referida, cada ponto de intersecção das linhas da grelha define um centro de grupo possível. Deste modo, aparentemente, por inspecção visual, os pontos de coordenadas (0.8; 0.4) e (0.4; 0.8) afiguram-se como as hipóteses mais viáveis. Definindo o parâmetro a = 5.4, obtém-se, graficamente, a função de potencial da Figura 5.9. Aí, destacam-se dois picos fundamentais, equivalentes aos pontos referidos anteriormente, os quais constituirão, de facto, a solução do

problema para a partição considerada.

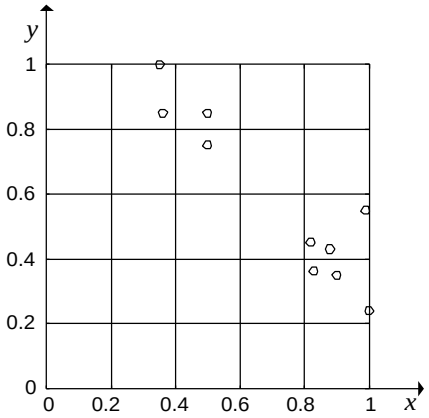


Figura 5.8. Conjunto de dados e partição do domínio.

O exemplo apresentado baseia-se numa das implementações mais conhecidas dos algoritmos de função de potencial: o *método da montanha*[59] [Yager e Filev, 1994]. Neste método, a função de potencial é vista como o relevo de uma superfície com picos e vales, de onde advém a sua designação. A ideia essencial do algoritmo baseia-se na definição de um conjunto de candidatos através de uma partição em grelha do espaço de dados, tal como na Figura 5.8. A cada um dos pontos definidos, associa-se, então, um determinado potencial, calculado com base nas suas distâncias respectivas a cada uma das amostras de dados (5.20).

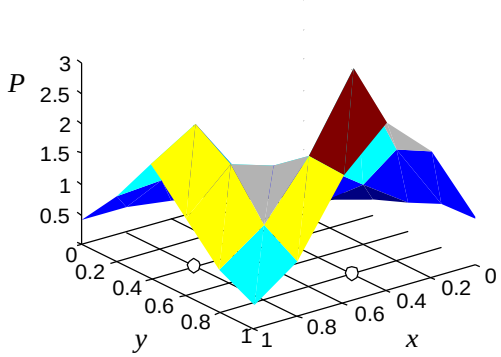


Figura 5.9. Função de potencial.

Assim, um ponto da grelha com um número elevado de amostras na sua vizinhança terá um potencial elevado. Deste modo, o ponto da grelha com maior potencial é escolhido como o primeiro centro de grupo - na Figura 5.9 seria o ponto de coordenadas (0.4; 0.8). Após a selecção do primeiro centro, o potencial de cada ponto da grelha é reduzido, de acordo com as respectivas distâncias ao centro definido. Consequentemente, a redução de potencial será mais notória nos pontos mais próximos do primeiro centro, de forma a que não sejam seleccionados grupos

[59] Nota: no desenvolvimento do método da montanha, a distância Euclidiana expressa em (5.19) e (5.20) não é elevada ao quadrado.

aproximadamente iguais em zonas densas. O novo ponto da grelha com maior potencial é agora seleccionado como centro, repetindo-se iterativamente o procedimento de determinação de centros e redução de potencial até que o potencial de todos os candidatos seja inferior a um limiar previamente especificado. Do exposto sobressai um aspecto interessante: neste algoritmo o número de grupos a encontrar não é estabelecido a priori. No entanto, tal como se verificará, na prática o número de grupos encontrados depende do valor atribuído ao parâmetro a.

Tal como se referiu, o método da montanha baseia-se na definição de uma grelha no espaço de entrada-saída. Porém, esta metodologia torna-se impraticável em problemas de maior dimensão. De facto, a necessidade de precisão conduz à definição de partições mais finas. Este aspecto, em conjugação com espaços de dados de dimensão elevada, leva ao problema designado por "curse of dimensionality". A título ilustrativo, um problema com 4 variáveis e 15 partições do domínio de cada variável originará 15^4 candidatos a centro de grupo. Assim, Chiu [Chiu, 1994] propôs no seu algoritmo de *agrupamento subtractivo*[60] uma alteração simples, contudo significativa: os candidatos a centros são as próprias amostras de dados, não sendo, portanto, necessário definir qualquer grelha. Deste modo, o número de pontos a avaliar iguala o número de amostras recolhidas, independentemente da dimensão do problema.

Algoritmo de agrupamento subtractivo

Com base no exposto, o algoritmo de agrupamento subtractivo é apresentado nos parágrafos seguintes.

Seja Z^N um conjunto de N amostras de dados, z_1, z_2, …, z_N, definidas num espaço de dimensão $m+n$, onde, num contexto de identificação de sistemas, m designa o número de entradas e n o número de saídas. De forma a que a gama dos valores em cada dimensão seja idêntica, assume-se que os dados estão normalizados, sendo, deste modo, limitados por um hipercubo.

Tal como se referiu, admite-se que cada uma das amostras define um eventual centro de grupo. Assim, o potencial associado ao ponto z_i é dado por (5.21):

$$P_i(z_i, Z^N) = \sum_{j=1}^{N} e^{-a\left\|z_i - z_j\right\|^2} \quad , i = 1,2,\cdots,N$$
$$a = \frac{4}{r_a^2} \tag{5.21}$$

onde $r_a>0$ é uma constante designada por *radii*, a qual define o raio da vizinhança de cada ponto. Assim, pontos z_j localizados fora do raio de acção de z_i terão uma influência reduzida no potencial. Ao invés, o efeito de pontos próximos no potencial de z_i será tanto maior quanto maior for a proximidade. Deste modo, pontos com uma vizinhança densa terão associados potenciais elevados. Em relação ao método da montanha, uma outra diferença reside no facto de que a medida de potencial é influenciada pelo quadrado da distância e não pela própria distância.

Após o cálculo do potencial de cada ponto, aquele que apresentar o potencial mais elevado é seleccionado como o primeiro centro de grupo. Tal como no método da montanha, o passo seguinte consiste na redução do potencial dos pontos restantes. Assim, definindo z_1^* como o centro do primeiro grupo encontrado e denotando o respectivo potencial por P_1^*, o potencial dos pontos restantes é reduzido como se segue (5.22):

---

[60] *Subtractive clustering*, em terminologia inglesa.

$$P_i \leftarrow P_i - P_1^* e^{-b\left\|z_i - z_1^*\right\|^2}$$
$$b = \frac{4}{r_b^2} \tag{5.22}$$

onde a constante r_b>0 define o raio da vizinhança com reduções sensíveis no seu potencial. Deste modo, os pontos próximos do centro escolhido verão o seu potencial reduzido da maneira mais significativa, pelo que a probabilidade de serem escolhidos como centros diminui. Este procedimento apresenta a vantagem de evitar a concentração de grupos idênticos em zonas densas. Neste sentido, o valor atribuído a r_b deve ser um pouco superior a r_a, de modo a obterem-se grupos espaçados. Tipicamente, define-se $r_b = 1.5\ r_a$ ou $r_b = 1.25\ r_a$.

Uma vez efectuada a redução de potencial de todos os pontos, aquele que apresentar o maior potencial é seleccionado como segundo centro, efectuando-se, seguidamente a redução do potencial dos restantes. Genericamente, após a determinação do *r*-ésimo grupo, o potencial é reduzido do seguinte modo (5.23):

$$P_i \leftarrow P_i - P_r^* e^{-b\left\|z_i - z_r^*\right\|^2} \tag{5.23}$$

O procedimento de selecção de centros e redução de potencial é repetido iterativamente até que se verifique o critério de paragem descrito na Tabela 5.4.

Se $\boldsymbol{P_r^* > e^{up} P_1^*}$
  Aceitar $\boldsymbol{z_r^*}$ como centro de grupo e continuar
Caso contrário,
  Se $\boldsymbol{P_r^* < e^{down} P_1^*}$
    Rejeitar $\boldsymbol{z_r^*}$ e terminar.
  Caso contrário
    Seja d_{min} = menor distância entre z_r^* e todos os centros já encontrados
    Se $d_{min}/r_a + P_r^*/P_1^* \geq 1$
      Aceitar $\boldsymbol{z_r^*}$ como centro de grupo e continuar
    Caso contrário
      Rejeitar $\boldsymbol{z_r^*}$ e atribuir ao seu potencial o valor 0.0.
      Seleccionar o ponto com o potencial mais elevado como o novo z_r^*.
      Voltar a **testar**.
    Fim Se
  Fim Se
Fim Se

Tabela 5.4. Critério de paragem do algoritmo de agrupamento subtractivo.

Na tabela precedente, o parâmetro e^{up} especifica um limiar para o potencial acima do qual o ponto é aceite como centro sem qualquer espécie de dúvida. Do mesmo modo, e^{down} especifica o limiar oposto, segundo o qual o ponto é rejeitado, pondo termo ao processo de procura. Tipicamente, define-se $e^{up} = 0.5$ e $e^{down} = 0.15$. Na terceira situação, a decisão quanto à aceitação ou rejeição é tomada com base no compromisso entre o potencial do ponto em análise e a sua distância em relação aos grupos já definidos. Assim, pontos com potencial relativamente elevado mas próximos dos centros obtidos tenderão a ser rejeitados. Ao invés, pontos com potencial

aparentemente baixo mas localizados numa zona onde poucos grupos tenham sido definidos tenderão a ser aceites. Em consequência do exposto, para os pontos em relação aos quais se aplica a decisão de compromisso enunciada, afirma-se que o seu potencial se encontra na *região cinzenta*.

Enquadramento do algoritmo na identificação difusa

Aplicado o algoritmo de agrupamento subtractivo, cada um dos grupos obtidos constituirá um protótipo exemplificativo de um determinado comportamento do sistema em causa. Assim sendo, cada grupo poderá ser utilizado para definir uma regra difusa susceptível de descrever o comportamento do sistema numa dada área do espaço de entrada-saída. A questão que se coloca agora reside na definição de um esquema de parametrização das funções de pertença a incluir no sistema difuso.

Assim, assuma-se que foram encontrados g centros $\{z_1^*, z_2^*, \ldots, z_g^*\}$ definidos num espaço de dimensão $m+n$. Cada um dos vectores z_r^* pode ser decomposto em duas componentes, x_r^* e y_r^*, de dimensões m e n, respectivamente, relativas às coordenadas no espaço de entrada e no espaço de saída. Deste modo, poder-se-ão definir g regras condicionais difusas do tipo (5.24):

Regra r:
SE (X_1 *é* $LX1^{(r)}$) *E* (X_2 *é* $LX2^{(r)}$) *E* … *E* (X_m *é* $LXm^{(r)}$) , $r = 1,2,\ldots,g$ (5.24)
ENTÃO (Y_1 *é* $LY1^{(r)}$) *E* (Y_2 *é* $LY2^{(r)}$) *E* … *E* (Y_n *é* $LYn^{(r)}$)

em que cada um dos termos linguísticos, $LXj^{(r)}$, do antecedente tem associada uma função de pertença, tal como se segue (5.25):

$$m_{L\tilde{X}j^{(r)}}(x_j) = e^{-a\left(x_j - x_{rj}^*\right)^2} \quad ,r = 1,2,\cdots,g; \quad j = 1,2,\cdots,m \tag{5.25}$$

Aqui x_j denota um valor numérico referente à dimensão j do espaço de entrada, sendo x_{rj}^* a j-ésima coordenada do vector x_r^*, de dimensão m. A expressão (5.25) resulta do cálculo do potencial associado a cada ponto do espaço de dados. De facto, considerando um conjunto de amostras, x, definidas num espaço de dimensão m, o grau de pertença de cada ponto em cada grupo, x_r^*, é dado por uma função multivariável (5.26):

$$m_{L\tilde{X}}(x) = e^{-a\left\|x - x_r^*\right\|^2} \quad ,r = 1,2,\cdots,g \tag{5.26}$$

Deste modo, verifica-se que a obtenção de funções de pertença univariáveis (5.25) resulta da definição da conjunção difusa pelo produto.

Em relação aos consequentes, será possível associar-lhes directamente um conjunto difuso (5.27) ou uma constante (5.28).

$$m_{L\tilde{Y}j^{(r)}}(y_j) = e^{-a\left(y_j - y_{rj}^*\right)^2} \quad ,r = 1,2,\cdots,g; \quad j = 1,2,\cdots,n \tag{5.27}$$

$$m_{L\tilde{Y}j^{(r)}}(y_j) = y_{rj}^* \quad ,r = 1,2,\cdots,g; \quad j = 1,2,\cdots,n \tag{5.28}$$

onde y_j denota um valor numérico referente à dimensão j do espaço de saída, sendo y_{rj}^* a j-ésima coordenada do vector y_r^*, de dimensão n.

Pelas expressões (5.25) e (5.27), verifica-se que a cada uma das coordenadas de um dado vector centro de dimensão $m+n$ estará associada uma função Gaussiana. Deste modo, a cada dimensão do problema estarão associadas g funções de pertença. Exemplificando, suponha-se que

num problema com duas entradas, X_1 e X_2, e uma saída, Y_1, o algoritmo de agrupamento subtractivo originaria três centros, com as seguintes coordenadas (5.29):

$$\begin{aligned} z_1^* &= \{0.4, 0.6, 0.5\} \\ z_2^* &= \{0.3, 0.7, 0.2\} \\ z_3^* &= \{0.6, 0.4, 0.7\} \end{aligned} \tag{5.29}$$

Do resultado acima, a variável X_1, correspondente à primeira coordenada, teria associadas três funções de pertença com centros respectivamente em 0.4, 0.3 e 0.6, o mesmo se passando em relação às restantes variáveis.

Relativamente ao desvio padrão de cada função, estabelecendo a analogia entre (5.25) e a expressão geral das Gaussianas (3.7), obtém-se, trivialmente, (5.30):

$$s_{rj} = \frac{r_a}{\sqrt{8}} \tag{5.30}$$

Finalmente, após a parametrização das funções de pertença Gaussianas obtidas, os dados de identificação, inicialmente normalizados, são restaurados para os seus valores iniciais. Do mesmo modo, os parâmetros das Gaussianas são ajustados para os domínios definidos em cada dimensão.

Análise do algoritmo de agrupamento subtractivo

O método de agrupamento subtractivo apresenta algumas características interessantes no contexto da aprendizagem da estrutura de um modelo difuso.

Assim, a sua vantagem mais marcante reside no facto de permitir ultrapassar os problemas associados à explosão da base de regras, associados a esquemas baseados na partição em grelha do espaço de entrada-saída. De facto, em problemas reais o número de variáveis físicas a incorporar num modelo é, geralmente, elevado. Este número cresce ainda com a necessidade de inclusão de entradas e saídas passadas, de forma a captar-se a dinâmica do sistema a modelizar. Deste modo, em modelos baseados em partições do tipo grelha, a base de regras obtida facilmente atingirá proporções impraticáveis, com consequências não só em termos de interpretabilidade, mas também de treino e custo computacional do modelo.

O algoritmo descrito pode ser utilizado na estimação do número de regras necessárias à definição de um modelo difuso baseado em dados. De facto, ao contrário de outros algoritmos, como o FCM, no método de Chiu o número de grupos não necessita de ser especificado previamente, sendo determinado automaticamente. No entanto, é importante notar que o parâmetro *radii* está directamente relacionado com o número de grupos encontrados. Assim, um raio pequeno originará um número elevado de regras, o que, no caso de ser excessivo, poderá redundar em problemas de sobreajustamento. Inversamente, um raio maior originará um número menor de grupos, o que poderá originar modelos com capacidades de aproximação reduzidas, no caso do número de regras se mostrar diminuto. Deste modo, em aplicações práticas é necessário testar diversos valores de *radii* e seleccionar o mais adequado em função dos resultados obtidos. Quanto ao parâmetro r_b, referiu-se que habitualmente se define uma relação constante entre este e r_a. Naturalmente, a definição de r_b afecta igualmente o número de centros obtidos, pelo que a necessidade de experimentar valores diferentes também se manifesta em algumas ocasiões.

Em termos de ruído presente nos dados, o algoritmo revela-se robusto em consequência do método de selecção de centros. Tal como foi descrito, pontos isolados terão potenciais baixos, dificilmente sendo escolhidos como centros. Tipicamente, ruído de alta frequência manifesta-se sob

a forma de *outliers*, pelo que a probabilidade de estes pontos serem seleccionados como centros é reduzida.

Quanto à eficiência computacional, o algoritmo apresenta vantagens decorrentes da não utilização de qualquer forma de optimização não linear. No entanto, para um número elevado de amostras, as suas vantagens em termos de eficiência, comparativamente a outros algoritmos com optimização, não é tão notória. O cálculo do potencial de cada ponto requer um número de computações da ordem de N, $O(N)$. Deste modo o cálculo do potencial total, anteriormente à selecção do primeiro centro envolve um número de computações de ordem $O(N^2)$. A selecção de cada centro é também $O(N)$. Assim, para g grupos, a complexidade computacional do algoritmo será $O(N^2+gN)$[61]. Em relação ao algoritmo FCM, a sua complexidade é $O(gNI)$, onde I representa o número de iterações efectuadas. Deste modo, quando $N>>gI$, onde >> se lê "consideravelmente maior", o algoritmo subtractivo terá um custo computacional superior ao FCM. Admitindo um problema típico onde g = 20, I = 200, N = 500, o algoritmo de agrupamento subtractivo será bastante mais eficiente. Caso N = 5000, o algoritmo FCM poderá ser mais eficiente, se o número de iterações necessárias não aumentar consideravelmente. Ainda em relação à questão da eficiência computacional, algoritmos iterativos do tipo *k-nearest neighbours* [Moody e Darken, 1989], apresentam a vantagem de possibilitarem a utilização de uma tabela indexada para a determinação, em cada iteração, do centro mais próximo da amostra em causa. Assim sendo, o seu peso será de ordem $O(NI)$, o que o tornará mais eficiente com a condição de que o número de iterações não seja demasiado elevado. No entanto, são colocados problemas em relação à inicialização, assim como à robustez ao ruído. Comparativamente à aprendizagem da estrutura na rede NFCN, o agrupamento subtractivo revela-se mais eficiente. Na verdade, o algoritmo de Lin, pelo procedimento de procura de centros através da aplicação do algoritmo de Kohonen a cada dimensão, associado ao procedimento de eliminação e combinação de regras, torna-se pesado.

O facto do algoritmo de agrupamento subtractivo não se basear em qualquer esquema de optimização torna-o pouco adequado a situações onde funcione isoladamente (excepto no caso em que os requisitos de precisão não sejam muito exigentes ou em problemas particularmente simples). Assim sendo, o algoritmo afigura-se especialmente interessante como forma de inicialização de algoritmos de optimização tal como o método FCM ou as redes neuro-difusas utilizadas neste trabalho. De facto, a aplicação prévia do método de Chiu permite ultrapassar as limitações do FCM em termos de inicialização da matriz de partição. Do mesmo modo, as regras obtidas e as funções de pertença definidas podem ser posteriormente optimizadas através do treino de uma rede neuro-difusa, o que será abordado na Secção 5.3.

Ao contrário da rede NFCN, o agrupamento subtractivo permite inicializar uma rede neuro-difusa sem que seja necessário definir o número de funções de pertença associadas a cada variável. No entanto, tal como se referiu anteriormente, o número de funções de pertença por variável será igual ao número de centros encontrados. Obviamente que esta situação apresenta limitações em termos da interpretabilidade do sistema difuso, não só pelo número de funções obtidas (25 regras = 25 funções de pertença!) mas também pelo elevado grau de similaridade entre elas (Figura 5.6). Deste modo, é necessário fundir as funções de pertença semelhantes, com base num determinado critério de similaridade, o que será abordado na Secção 5.4.

---

[61] De realçar que os valores expostos resultam de uma análise simplificada. Na verdade, após a selecção de cada centro, o número de pontos a analisar diminui. No entanto, grosso modo, a complexidade será da ordem referida.

Para finalizar, é ainda importante realçar o facto de o método de agrupamento subtractivo se basear na definição de hiperesferas no espaço de dados, todas com o mesmo raio (*radii*). Este aspecto apresenta-se desvantajoso, uma vez que grupos com formas distintas não serão encontrados directamente. Deste modo, a possibilidade de se definirem raios diferentes para cada centro, bem como para cada dimensão, afigura-se interessante uma vez que os grupos encontrados são susceptíveis de apresentarem dimensões e formas diferentes, e.g., helicoidais ou esféricas, com raios distintos. Este problema é ilustrado na Figura 5.10.

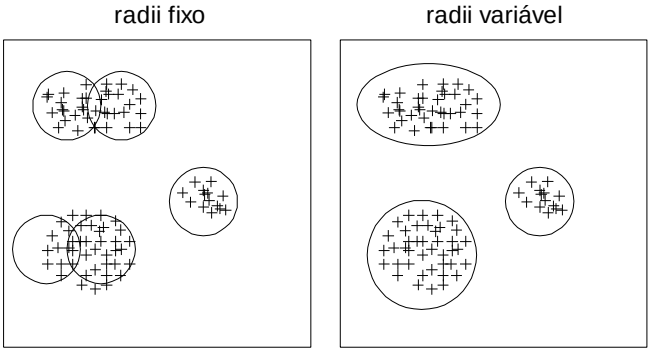


Figura 5.10. Algoritmo de agrupamento subtractivo com raios fixos e variáveis.

Comparando as duas figuras anteriores, verifica-se que a segunda situação, por ser mais genérica, possibilita a determinação de um número inferior de grupos. O seu problema essencial reside na atribuição dos valores correctos a cada um dos *radii*. Este problema é abordado em [Dray et al, 1998], onde a determinação dos raios é efectuada por um esquema de optimização elaborado e algo pesado computacionalmente. O método utilizado baseia-se na definição de um sistema difuso com consequentes do tipo Takagi-Sugeno. O erro do modelo é avaliado, sendo os parâmetros do consequente, bem como os raios, optimizados em comum. Mais uma vez, dado que o objectivo proposto nesta secção reside na procura de um algoritmo eficiente para a definição da estrutura de um modelo difuso, e em virtude do peso do método referido e dos resultados pouco convincentes apresentados, o método proposto em [Dray et al, 1998] não será utilizado.

5.2.3. Selecção de Entradas

Foi referido na Secção 2.4.3 que as redes neuronais utilizadas neste trabalho não dispõem de memória dinâmica. Assim sendo, na implementação de modelos difusos entrada-saída, o problema da selecção das entradas a utilizar envolve não só a escolha das variáveis físicas, mas também a definição da ordem e atraso de cada uma delas.

O problema da selecção de entradas

A determinação do conjunto de variáveis a incluir num modelo constitui um dos maiores desafios na área da modelização de sistemas. Num dado sistema, cada uma das variáveis afecta o seu comportamento a níveis diferentes. Assim sendo, a não inclusão de uma variável importante poderá levar a um comportamento deficiente do modelo. Por outro lado, a inclusão de variáveis com pouco peso conduzirá a modelos desnecessariamente complexos, em oposição ao princípio da parcimónia (Secção 2.2), além de que o sistema de aquisição de dados apresentará custos mais elevados. Do mesmo modo, na perspectiva de controlo, conhecer a importância relativa de cada

variável possibilita que se concentrem esforços sobre as variáveis mais relevantes, reduzindo-se o tempo e custo necessários ao controlo e determinação de referências (*set-points*) referentes a variáveis com menor significado. Do exposto, transparece o elevado grau de importância inerente à tarefa de selecção de entradas.

Convencionalmente, a determinação das variáveis a incorporar num modelo, bem como a sua dinâmica, é efectuada com base em conhecimento prévio sobre o sistema em causa. No entanto, a informação necessária poderá não se encontrar disponível, ou apresentar um nível de fiabilidade reduzido. Este é o caso em que um operador indica que uma determinada variável apresenta um atraso de "mais ou menos 15 minutos". Deste modo, o problema enunciado requer uma abordagem rigorosa, a qual passará pela construção de modelos baseados nos primeiros princípios - com as dificuldades que daí advêm - ou pela procura de uma solução baseada na análise automática de dados. Esta última abordagem é a utilizada.

No estudo de sistemas lineares, a selecção óptima das variáveis de entrada a incorporar no modelo é habitualmente levada a cabo pelo critério de informação de Akaike. A aplicação deste método baseia-se na implementação de modelos incluindo conjuntos diferentes de variáveis e seleccionando o modelo que minimize o critério AIC. A selecção referida rege-se pelo princípio da parcimónia, estabelecendo um compromisso entre a complexidade do modelo e a sua capacidade de representação, com base em princípios estatísticos.

A modelização de sistemas lineares apresenta, em todos os aspectos, um conjunto de fundamentos teóricos sólidos, o que não se verifica em relação a sistemas não lineares. De facto, no problema da selecção de entradas em sistemas não lineares, não se encontram definidos critérios rigorosos, pelo que as abordagens utilizadas assentam sobre princípios heurísticos. As poucas soluções com um grau de rigor mais elevado baseiam-se em assunções fortemente restritivas, que as tornam impraticáveis em aplicações práticas.

Estratégias de selecção de entradas em sistemas não lineares

A identificação do conjunto óptimo de variáveis a incluir num modelo não linear é, virtualmente e no momento presente, um problema de resolução impossível. Assumindo que a recolha de dados fornece um conjunto de amostras suficientemente rico e que o algoritmo de aprendizagem de parâmetros possibilita a determinação do modelo paramétrico óptimo, não há qualquer garantia de que um modelo baseado num determinado conjunto de variáveis seja o ideal, excepto se se analisarem todas as combinações possíveis de variáveis físicas, com todos os valores possíveis para os seus atrasos e ordens. Esta solução apresenta custos impraticáveis a nível computacional, inclusivamente para problemas com um número moderado de variáveis candidatas. De facto, num problema com m variáveis possíveis, o número total de combinações será 2^m-1. Este problema torna-se ainda mais marcante quando o custo de desenvolvimento de cada modelo é elevado. Consequentemente, a esmagadora maioria dos métodos de selecção de entradas constituem algoritmos subóptimos de procura, que não garantem a obtenção da solução óptima. Nestes métodos, distinguem-se duas classes essenciais: as estratégias de selecção para a frente e as de selecção para trás.

Os *algoritmos de selecção para a frente* têm por base a implementação de modelos com um número de variáveis gradualmente maior. Inicialmente, é construído um conjunto de modelos com uma só variável, cada um dos quais contendo uma das entradas candidatas. A variável associada ao melhor modelo obtido é então seleccionada, sendo posteriormente combinada com todas as restantes, formando-se modelos com duas variáveis. Este procedimento de introdução da melhor variável em cada iteração continua até que o desempenho do modelo estabilize (ou, eventualmente, diminua). Nesta classe de metodologias inclui-se, por exemplo, o algoritmo de Takagi e Sugeno

[Takagi e Sugeno, 1985] e as redes neuronais GMDH[62] [Ivakhnenko et al, 1979].

Os *algoritmos de selecção para trás* baseiam-se no princípio oposto. Assim, em lugar de se começar com modelos simples, o algoritmo é iniciado com a construção de um modelo com todas as entradas candidatas. Seguidamente, avalia-se o desempenho do modelo sem cada uma das variáveis. O conjunto de variáveis com o melhor resultado é mantido, continuando o processo, iterativamente, até que o desempenho se degrade de forma inaceitável. De acordo com Chiu [Chiu, 1996], a remoção de uma variável não originará a degradação do desempenho do modelo caso se verifiquem quatro condições: a saída não varia significativamente com a entrada em causa; a variação da saída deve-se a ruído; a entrada é redundante, de forma a que a variação da saída pode ser modelizada por outras variáveis; o modelo sobreajusta-se aos dados de forma significativa. Assim, as três condições iniciais verificam-se quando a variável é irrelevante. Quanto à última condição, da situação descrita transparece a necessidade de não se definirem modelos excessivamente complexos, de forma a evitar-se o problema do sobreajustamento

Tanto a selecção para a frente como a selecção para trás constituem estratégias mais eficientes que a procura exaustiva. No entanto, o seu peso computacional é também elevado, uma vez que as estratégias enunciadas requerem a implementação de um número elevado de modelos na exploração das combinações de variáveis. Em geral, os métodos de selecção para a frente são preferidos, uma vez que começam pela exploração de modelos simples, fáceis de implementar, aumentando-se a complexidade unicamente se tal for necessário. Ao invés, a selecção para trás baseia-se na construção de um modelo inicial (desnecessariamente) complexo. Deste modo, os primeiros são mais eficientes que os segundos.

No sentido da melhoria da eficiência dos algoritmos de selecção de entradas, em [Lin e Cunningham, 1995] é proposto um método de selecção de entradas bastante eficiente. Aqui, os dados de treino são projectados em diferentes planos de entrada-saída, assumindo-se que a saída não depende de uma dada entrada se o gráfico obtido para o mapeamento respectivo for relativamente horizontal. No entanto, como é óbvio, esta visão é algo simplista, uma vez que despreza interacções entre variáveis como causa possível do ocorrido. Além do referido, não se detecta a presença de variáveis redundantes, i.e., variáveis fortemente correlacionadas, cuja incorporação mútua seja desnecessária.

Uma outra estratégia consiste na análise de componentes principais (PCA[63]) [Jackson, 1991] para a redução do número de entradas. Neste método, a eficiência computacional é satisfatória. No entanto, tal como é referido por Chiu [Chiu, 1996], a selecção de variáveis é efectuada com base na sua variabilidade, não no facto da entrada influenciar realmente a saída. Deste modo, em termos puramente teóricos, uma variável com variância elevada pode não apresentar qualquer relação com a saída. Um outro aspecto prende-se com a transparência do modelo obtido. Uma vez que a PCA reduz a dimensão do espaço de entradas pelo seu mapeamento num espaço de menor dimensão, as variáveis originais são perdidas, o que impossibilita a interpretabilidade linguística do sistema.

Nos parágrafos seguintes será apresentado o algoritmo de selecção de entradas de Chiu [Chiu, 1996], o qual se adequa particularmente a problemas de modelização difusa. A razão da sua escolha advém do aspecto referido e, fundamentalmente, da sua eficiência em comparação com outros métodos.

---

[62] *Group Method for Data Handling*, em terminologia inglesa.
[63] *Principal Component Analysis*, em terminologia inglesa.

Algoritmo de Chiu

A ideia genérica do método de Chiu consiste na implementação de um único modelo difuso, incorporando todo o conjunto de entradas possíveis. Posteriormente, e sequencialmente, são eliminadas as proposições do antecedente de cada regra difusa associadas a uma dada variável, de forma a testar a importância relativa das variáveis eliminadas. Desta forma, o algoritmo proposto enquadra-se nos métodos de selecção para trás, com a vantagem de evitar a necessidade de gerar novos modelos repetidamente, do que resulta um ganho significativo em termos de eficiência.

O algoritmo começa pela definição de um modelo difuso contendo todo o conjunto de variáveis de entrada possíveis. Este modelo pode ser obtido por qualquer um dos métodos referidos anteriormente, utilizando ou não redes neuro-difusas. No entanto, métodos que evitem a explosão da base regras são preferíveis, em virtude do elevado número de variáveis normalmente utilizadas no modelo inicial. Assim, a estrutura do modelo difuso é determinada por técnicas de agrupamento de classes, nomeadamente agrupamento subtractivo. Após a aprendizagem da estrutura, os parâmetros são optimizados, tal como se abordará na Secção 5.3. O modelo inicial não deve sobreajustar-se excessivamente aos dados de treino. Deste modo, a optimização deve levar em consideração o desempenho do modelo em relação aos dados de teste, sendo terminada no caso de se verificar qualquer degradação.

Ao contrário de outras abordagens, a arquitectura dos sistemas difusos possibilita o teste simples da importância de cada entrada, sem que com isso seja necessário gerar novos modelos. De facto, a estrutura baseada em regras permite a remoção de uma dada variável através da remoção das cláusulas dos antecedentes a ela associadas, em todas as regras. Exemplificando, num modelo com três entradas e uma saída, e regras da forma (5.31):

$$\begin{aligned}&\text{SE } (X_1 \text{ é } LX1^{(r)}) \text{ E } (X_2 \text{ é } LX2^{(r)}) \text{ E } (X_3 \text{ é } LX3^{(r)})\\&\text{ENTÃO } (Y_1 \text{ é } LY1^{(r)})\end{aligned} \tag{5.31}$$

a importância da variável X_2 poderá ser testada pela remoção temporária da proposição (X_2 é $LX2^{(r)}$) em cada uma das regras do modelo. Desta maneira, as regras são truncadas para a forma (5.32):

$$\begin{aligned}&\text{SE } (X_1 \text{ é } LX1^{(r)}) \text{ E } (X_3 \text{ é } LX3^{(r)})\\&\text{ENTÃO } (Y_1 \text{ é } LY1^{(r)})\end{aligned} \tag{5.32}$$

Este processo equivale à contracção do espaço de entrada-saída.

Deste modo, se o desempenho do modelo não se degradar em termos de um determinado critério, e.g., RMSE, então a variável testada poderá ser eliminada do modelo. Na prática, na eliminação de variáveis não é necessário proceder à actualização das regras, bastando associar o valor de verdade 1 a todas as proposições associadas à variável removida.

Em termos algorítmicos, o funcionamento do método de Chiu é apresentado na Tabela 5.5. Assumindo um modelo com quatro variáveis possíveis, o algoritmo é ilustrado graficamente na Figura 5.11.

Um aspecto interessante do método advém do facto da truncatura de regras constituir unicamente um mecanismo matemático de contracção da superfície de saída do modelo inicial para um espaço de entrada de menor dimensão. Deste modo, as regras truncadas não apresentam qualquer semelhança com as que seriam extraídas dos dados utilizando o subconjunto de variáveis em causa. Assim sendo, após a eliminação de cada variável, o autor optou por não reoptimizar as regras truncadas, exactamente por se tratar de um modelo truncado, o que poderia conduzir a problemas de sobreajustamento. Uma vez que o algoritmo se baseia na comparação do desempenho relativo do modelo após a remoção de cada variável, realizar a operação de optimização não é

fundamental, podendo mesmo revelar-se nefasta.

Em relação à aplicação do algoritmo a modelos Takagi-Sugeno de ordem 1, a remoção de variáveis afecta os termos dos consequentes, o que não acontece em sistemas linguísticos ou de ordem 0. Deste modo, são utilizados modelos linguísticos ou com consequentes constantes na implementação do método de selecção de entradas.

1. Avaliar o desempenho do modelo inicial com todas as variáveis;
2. Para cada variável no modelo, avaliar o desempenho do modelo resultante da eliminação temporária da variável em causa;
3. Eliminar definitivamente a variável associada ao modelo parcial com melhor desempenho. Guardar o conjunto de variáveis obtido e o desempenho do modelo;
4. Se existirem ainda variáveis no modelo, voltar ao passo 2; Caso contrário, ir para o passo 5;
5. Escolher o melhor conjunto de entre todos os guardados no passo 3.

Tabela 5.5. Algoritmo de selecção de entradas.

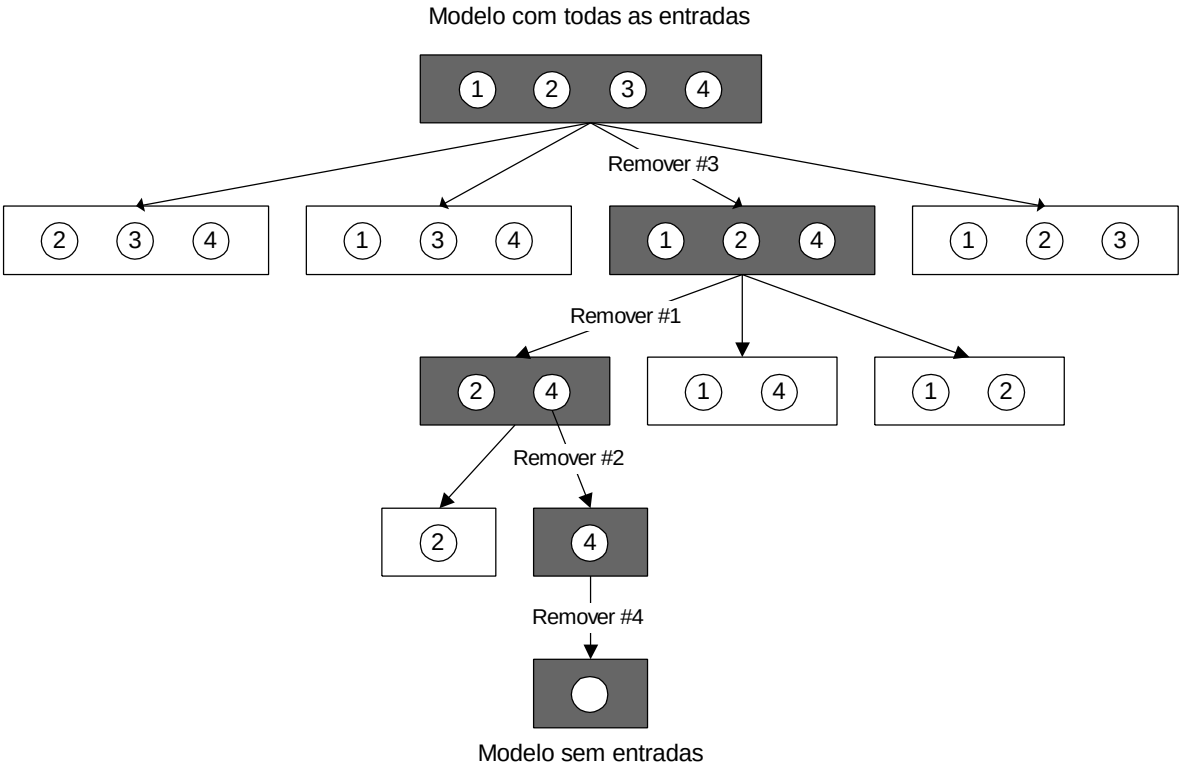


Figura 5.11. Algoritmo de selecção de entradas.

Quanto à medida de desempenho do modelo, várias hipóteses se apresentam, sendo a mais simples a utilização do critério RMSE, aplicado a um conjunto de teste. No entanto, o critério referido, apesar de favorecer a implementação de modelos com capacidades de aproximação satisfatórias, apresenta a desvantagem de ser sensível à escolha dos dados de treino e de teste.

Um outro critério, designado por *critério sem desvio* [Sugeno e Kang, 1988], consiste na

divisão dos dados em dois grupos, *A* e *B*, e na construção de dois modelos, um com base no grupo *A* e outro com base no grupo *B*, sendo definido como se segue (5.33):

$$J_U = \sqrt{\sum_{i=1}^{N_A} \left\| \hat{y}_i^{AB} - \hat{y}_i^{AA} \right\|^2 + \sum_{i=1}^{N_B} \left\| \hat{y}_i^{BA} - \hat{y}_i^{BB} \right\|^2} \tag{5.33}$$

onde N_A e N_B designam o número de amostras dos grupos *A* e *B*, respectivamente. Na mesma expressão, y_i^{AB} representa a previsão para a *i*-ésima amostra do conjunto *A*, efectuada com base num modelo construído com os dados do conjunto *B* (e assim sucessivamente). Os restantes termos são descritos de forma análoga. O critério apresentado tem subjacente a ideia de tornar o modelo obtido insensível aos dados utilizados no seu desenvolvimento. Deste modo, a expressão (5.33) procura minimizar a diferença entre a saída dos dois modelos derivados. No entanto, uma vez que a capacidade de previsão não é levada em consideração, o critério poderá não originar a selecção de variáveis com melhor capacidade de previsão.

Em virtude das desvantagens apontadas aos dois critérios anteriores, o *critério da regularidade* é sugerido em [Sugeno e Yasukawa, 1993], procurando conjugar as vantagens do critério RMSE com as do critério sem desvio. Deste modo, a sua definição baseia-se no cálculo do erro quadrático médio, MSE, para dois modelos distintos (5.34):

$$J_R = \frac{1}{2}\sum_{i=1}^{N_A} \frac{\left\| y_i^A - \hat{y}_i^{AB} \right\|^2}{N_A} + \frac{1}{2}\sum_{i=1}^{N_B} \frac{\left\| y_i^B - \hat{y}_i^{BA} \right\|^2}{N_B} \tag{5.34}$$

Na expressão anterior, y_i^A representa a *i*-ésima amostra presente no conjunto *A*. A vantagem do critério apresentado reside no facto de estabelecer um compromisso satisfatório entre a insensibilidade aos dados de treino e a capacidade de representação dos modelos obtidos. Contudo, dado que os critérios do tipo RMSE dão ao utilizador humano uma noção mais correcta da magnitude do erro, Chiu [Chiu, 1996] sugere a aplicação da raiz quadrada ao critério anterior, tal como se segue (5.35):

$$J_R = \sqrt{\frac{1}{2}\sum_{i=1}^{N_A} \frac{\left\| y_i^A - \hat{y}_i^{AB} \right\|^2}{N_A} + \frac{1}{2}\sum_{i=1}^{N_B} \frac{\left\| y_i^B - \hat{y}_i^{BA} \right\|^2}{N_B}} \tag{5.35}$$

Assim sendo, o critério precedente é o utilizado neste trabalho, na tarefa de validação de entradas a incluir num modelo difuso.

Após a selecção das variáveis relevantes, o algoritmo de aprendizagem da estrutura é aplicado, utilizando-se agora unicamente o conjunto de entradas escolhidas. De notar que o modelo difuso final poderá ser do mesmo tipo do implementado no algoritmo de selecção ou de outro diferente, e.g., Takagi-Sugeno de ordem 1. Após a aprendizagem da estrutura do modelo difuso final, os parâmetros do modelo são optimizados, aspecto este analisado seguidamente.

Em resumo, o método descrito consiste, basicamente, em ajustar aos dados uma superfície de dimensão elevada e contrair posteriormente essa mesma superfície em cada uma das dimensões, no sentido de se verificar a adequação do mapeamento. Naturalmente, a partir do momento em que a remoção de variáveis se inicia, as regras iniciais deixam de representar as regras reais. Na verdade, as regras não são mais do que um mecanismo através do qual o espaço é contraído, não havendo, como tal, necessidade de as optimizar. Esta conclusão é trivial, uma vez que o seu número é mantido durante todo o processo de redução de dimensões. Desta forma, a optimização das regras não é útil, podendo inclusivamente originar problemas de sobreajustamento, uma vez que o número de regras se torna demasiado elevado para um espaço de dimensão menor. Obviamente, a

alternativa é, após a determinação de cada variável de entrada, extrair um conjunto de regras completamente novo e optimizar o modelo. No entanto, este é o procedimento tradicional o qual apresenta elevados custos computacionais.

É ainda importante realçar o facto do método poder ser apenas utilizado como indicador das entradas relevantes e não mais, o que decorre da falta de formalismo e rigor matemático inerente ao algoritmo descrito, assim como a todos os métodos heurísticos.

5.3. Aprendizagem de Parâmetros

As redes neuro-difusas utilizadas na representação de sistemas difusos são susceptíveis de serem interpretadas com base unicamente na teoria dos sistemas difusos, visão essa na qual a rede é considerada unicamente como um meio de representação. Por outro lado, numa visão inversa, a estrutura neuro-difusa pode ser analisada com base na teoria das redes neuronais, tratando-se, então, apenas de mais uma estrutura neuronal entre tantas outras. Assim, consoante se trate de um problema de aprendizagem de parâmetros ou de aprendizagem de regras, teremos, respectivamente, arquitecturas multicamada com ligações para a frente ou redes competitivas. No contexto em que se insere esta secção serão consideradas redes multicamada. Assim, a topologia das redes neuro-difusas pode ser definida de diversas maneiras, de acordo com os objectivos e com a estrutura desejada para o sistema difuso representado.

Nesta secção, o problema da sintonização de parâmetros de algumas arquitecturas neuro-difusas será abordado, assumindo-se a definição prévia de uma estrutura.

5.3.1. Arquitecturas Neuro-Difusas

Em termos genéricos, distinguem-se, essencialmente, três tipos de estruturas, de acordo com a forma dos consequentes de cada regra, nomeadamente definição de consequentes do tipo Takagi-Sugeno de ordem 0 e 1 e consequentes difusos. Basicamente, as redes apresentadas nos parágrafos seguintes são constituídas por uma camada de entrada, seguindo-se uma camada de funções de pertença e só depois a camada de regras. Após as camadas iniciais segue-se, no caso de consequentes do tipo Takagi-Sugeno de ordem 0 ou ordem 1, uma camada linear de saída. Para o caso de consequentes difusos, segue-se uma camada de integração de regras com o mesmo consequente (*norma-S*) e, finalmente, a camada de saída responsável pela operação de desfuzificação.

Por forma a tornar mais clara a leitura das expressões matemáticas expressas nos parágrafos seguintes, optou-se por apresentar, desde já, a notação relativa à activação dos neurónios de cada camada:

- $a_i^{(p2)}$: activação do neurónio *i* da camada 2, relativamente ao padrão de treino *p* (*i* denota um termo de entrada: "*input*");
- $a_r^{(p3)}$: activação do neurónio *r* da camada 3, relativamente ao padrão *p* (*r* denota "*regra*");
- $a_s^{(p4)}$: activação do neurónio *s* da camada 4, relativamente ao padrão *p* (*s* denota "*norma-S*");
- $a_o^{(p5)} = y_o^{(p)}$: activação do neurónio *o* da camada 5, i.e., saída, relativamente ao padrão *p* (*o*

denota saída: *"output"*);

No caso de consequentes do tipo Takagi-Sugeno, a camada de saída é a quarta, tendo-se:

- $a_o^{(p4)} = y_o^{(p)}$: activação do neurónio *o* da camada 4, i.e., saída, relativamente ao padrão *p* (*o* denota saída: *"output"*);

Consequentes do tipo Takagi-Sugeno

Assim, do exposto resulta, para estruturas difusas do tipo Takagi-Sugeno, a arquitectura representada na Figura 5.12 [Glorennec,1994]. Naturalmente, a rede apresentada serve tanto os modelos de ordem 1 como de ordem 0, bastando para tal considerar pesos constantes ou funções de primeira ordem.

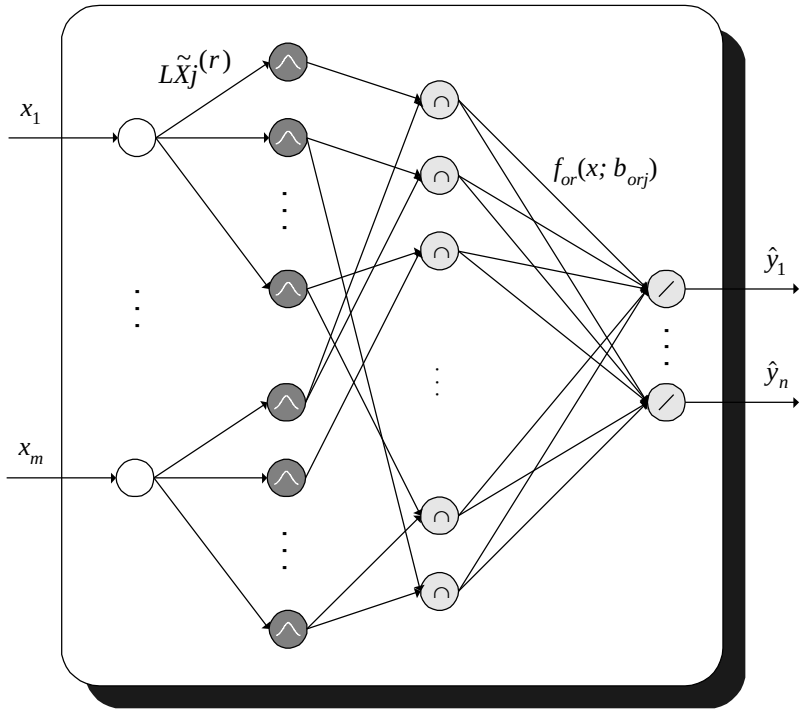


Figura 5.12. Rede neuro-difusa genérica: consequentes de Takagi-Sugeno.

Nesta estrutura e nas seguintes, a *camada de entrada* tem por missão única receber dados do ambiente exterior e passá-los à camada seguinte, não realizando, portanto, processamento útil.

Na segunda camada, a *camada de termos de entrada*, cada uma das células equivale a uma função de pertença associada a uma das entradas. No caso presente, o número total de funções de pertença associadas às variáveis de entrada, n_{fpi}, é dado por (5.4).

Definindo funções de pertença Gaussianas (3.7), a saída de cada um dos neurónios desta camada é dada por (5.36):

$$a_i^{(p2)} = e^{-\frac{\left(x_j^{(p)} - c_{ij}\right)^2}{2s_{ij}^2}}, i = 1,2,\cdots,n_{fpi} \tag{5.36}$$

Alternativamente, poder-se-ão definir funções Gaussianas generalizadas (3.8), fundamentalmente na presença de requisitos de interpretabilidade. Porém, a sua utilização é

passível de originar problemas de sobreajustamento, em virtude da duplicação do número de parâmetros.

Quanto aos neurónios da *camada de regras*, a sua função é, basicamente, efectuar a conjunção dos antecedentes de cada uma das regras, por meio de uma qualquer norma-T, e.g., produto (5.37). No trabalho presente, utiliza-se também o operador mínimo.

$$a_r^{(p3)} = \underset{i=1}{\overset{na_r}{norma-T}}\left(a_i^{(p2)}\right) = \prod_{i=1}^{na_r} a_i^{(p2)} \quad , r = 1,2,\cdots,g \tag{5.37}$$

No que respeita à *camada de saída*, a sua tarefa consiste no cálculo de saídas reais com base no grau de activação de cada regra. Tal como se referiu anteriormente, em modelos de ordem 0, os pesos desta camada denotam os consequentes do sistema difuso, definidos por constantes. Deste modo, a activação de cada um dos neurónios de saída é definida por (5.38):

$$\hat{y}_o(p) = a_o^{(p4)} = \frac{\sum_{r=1}^{g} a_r^{(p3)} \cdot b_{0r}}{\sum_{r=1}^{g} a_r^{(p3)}} \quad , o = 1,2,\cdots,n \tag{5.38}$$

Assim, cada regra será da forma (5.39):

Regra r:
SE (X_1 *é* $LX1^{(r)}$) *E* (X_2 *é* $LX2^{(r)}$) *E* … *E* (X_m *é* $LXm^{(r)}$)
ENTÃO ($y_1^{(p)} = b_{1r}$) *E* ($y_2^{(p)} = b_{2r}$) *E* … *E* ($y_n^{(p)} = b_{nr}$) (5.39)

Na implementação de regras do tipo Takagi-Sugeno de ordem 1, a rede define um sistema difuso com regras do tipo (5.40), definindo-se $f_{or}(x)$ como em (5.41):

Regra r:
SE (X_1 *é* $LX1^{(r)}$) *E* (X_2 *é* $LX2^{(r)}$) *E* … *E* (X_m *é* $LXm^{(r)}$)
ENTÃO [$y_1^{(p)} = f_{1r}(x^{(p)})$] *E* [$y_2^{(p)} = f_{2r}(x^{(p)})$] *E* … *E* [$y_n^{(p)} = f_{nr}(x^{(p)})$] (5.40)

$$\begin{aligned} f_{or}\left(x^{(p)}\right) &= f_{or}\left(x_1^{(p)}, x_2^{(p)}, \cdots, x_m^{(p)}\right) = b_{or0} + b_{or1}x_1^{(p)} + b_{or2}x_2^{(p)} + \cdots + b_{orm}x_m^{(p)} \\ b_{orj} &\in \Re, \quad j = 1,2,\cdots,m; \quad o = 1,2,\cdots,n; \quad r = 1,2,\cdots,g \end{aligned} \tag{5.41}$$

Assim, os neurónios de saída executam uma tarefa em tudo idêntica à do caso anterior (5.38), sendo no contexto presente definidos por (5.42):

$$\hat{y}_o(p) = a_o^{(p4)} = \frac{\sum_{r=1}^{g} a_r^{(p3)} \cdot f_{or}\left(x^{(p)}\right)}{\sum_{r=1}^{g} a_r^{(p3)}} = \frac{\sum_{r=1}^{g} a_r^{(p3)} \cdot \left(\sum_{j=1}^{m} b_{orj} x_j^{(p)} + b_{or0}\right)}{\sum_{r=1}^{g} a_r^{(p3)}} \quad , o = 1,2,\cdots,n \tag{5.42}$$

A estrutura neuro-difusa de sistemas do tipo Takagi-Sugeno mais conhecida é, eventualmente, a arquitectura ANFIS [Jang, 1993]. Aí, a rede é constituída por seis camadas, podendo no entanto, simplificar-se de acordo com a estrutura apresentada na Figura 5.12.

Consequentes difusos

No sentido da definição de redes neuro-difusas genéricas para a utilização de consequentes

difusos, Lin define na sua arquitectura NFCN [Lin, 1995] uma estrutura composta por cinco camadas, a qual se representa na Figura 5.13. A mesma arquitectura é também apresentada em [Shann e Fu, 1995]. De notar que a arquitectura apresentada na figura referida difere da estrutura presente na Figura 5.2, a qual constitui um caso particular da rede para aprendizagem da estrutura.

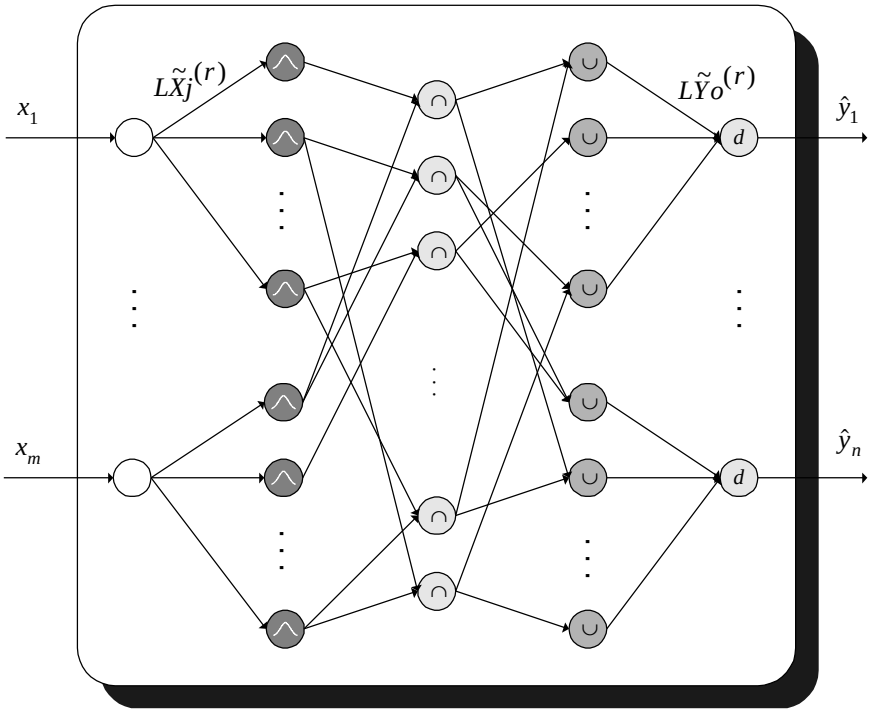


Figura 5.13. Rede neuro-difusa genérica: consequentes difusos.

Na arquitectura anterior, a quarta camada, designada por *camada de união*, é responsável pela integração de regras com o mesmo consequente, por meio de uma norma-S, de onde deriva a sua denominação. Na mesma figura, o símbolo *d* denota a operação de desfuzificação.

O número total de funções de pertença associadas às variáveis de saída, n_{fpo}, é dado por (5.7). Assim, a activação dos neurónios nesta camada, para o operador adição limitada (3.20), será definida por (5.43). Adicionalmente, utiliza-se neste trabalho o operador máximo.

$$a_s^{(p4)} = \underset{r=1}{\overset{nr_s}{norma}} - S\left(a_r^{(p3)}\right) = \min\left(1, \sum_{r=1}^{nr_s} a_r^{(p3)}\right) \quad , s = 1,2,\cdots,n_{fpo} \tag{5.43}$$

onde nr_s designa o número de regras que têm o neurónio *s* por consequente.

Quanto à *camada de saída*, os pesos das ligações entre os seus neurónios e os neurónios da camada de união definem os parâmetros das funções de pertença associadas aos termos de saída. Assim, com base nestas funções de pertença e na activação de cada regra, os seus neurónios devem implementar um determinado método de desfuzificação adequado a consequentes difusos, tal como o método apresentado em [Lin, 1995] (3.38). Deste modo, obtém-se (5.44):

$$\hat{y}_o^{(p)} = a_o^{(p5)} = \frac{\sum_{s=1}^{|T(Y_o)|} c_{os}\, s_{os}\, a_s^{(p4)}}{\sum_{s=1}^{|T(Y_o)|} s_{os}\, a_s^{(p4)}} \tag{5.44}$$

onde c_{os} e s_{os} representam o centro e desvio padrão da *s*-ésima função de pertença associada à saída *o*. No caso de se utilizarem funções Gaussianas generalizadas, vem (5.45), tal como se define em [Paiva et al, 1999] (5.45):

$$\hat{y}_o(p) = a_o(p5) = \frac{\sum_{s=1}^{T(Y_o)} \frac{1}{2}(c_{osL} s_{osL} + c_{osR} s_{osR}) a_s(p4)}{\sum_{s=1}^{T(Y_o)} \frac{1}{2}(s_{osL} + s_{osR}) a_s(p4)} \tag{5.45}$$

Naturalmente, a expressão (5.45), reduz-se à expressão (5.44) no caso de se tratar de uma função Gaussiana usual.

Na utilização de consequentes difusos, a rede define um sistema difuso com regras do tipo (5.46):

Regra r:
SE (X_1 *é* $LX1^{(r)}$) *E* (X_2 *é* $LX2^{(r)}$) *E* … *E* (X_m *é* $LXm^{(r)}$)
ENTÃO (Y_1 *é* $LY1^{(r)}$) *E* (Y_2 *é* $LY2^{(r)}$) *E* … *E* (Y_n *é* $LYn^{(r)}$) (5.46)

Na literatura, encontram-se por vezes arquitecturas mais simples, unicamente com três camadas, semelhantes às redes RBF convencionais (*vide* [Cho e Wang, 1996]). Porém, a sua simplicidade é conseguida à custa de algumas restrições impostas ao sistema difuso definido. Na verdade, verifica-se que, para cada variável de entrada, o número de conjuntos difusos é igual ao número de regras definidas. Esta situação pode levantar problemas quando se deseja que o sistema difuso final seja interpretável. De facto, a definição de um modelo difuso com 25 regras, origina variáveis com 25 conjuntos difusos associados. O problema referido verifica-se igualmente no caso de consequentes difusos[64]. Além do aspecto enunciado, o modelo impõe que cada regra tenha por antecedentes termos referentes a todas as variáveis de entrada. De facto, numa situação em que um sistema com três variáveis linguísticas de entrada, X_1, X_2 e X_3, uma determinada regra fosse, idealmente, da forma (5.47):

Regra r:
SE (X_1 *é* $LX1^{(r)}$) *E* (X_2 *é* $LX2^{(r)}$) ENTÃO (Y_1 *é* $LY1^{(r)}$) (5.47)

o sistema difuso não possibilitaria a sua obtenção (pelo menos não de uma forma trivial), dado que iria sempre incluir um termo associado à variável X_3.

Em virtude do seu carácter mais abrangente, as estruturas genéricas descritas, as quais permitem representar sistemas difusos de forma mais flexível, serão as utilizadas na aplicação a casos de estudo, no próximo capítulo.

5.3.2. Metodologias de Treino

Tal como no ponto anterior, as metodologias de sintonização de parâmetros em redes

[64] É, no entanto, importante realçar que a definição de sistemas difusos com consequentes do tipo Takagi-Sugeno de ordem 1, permite, em princípio, concluir que a interpretabilidade não é um factor em consideração.

neuro-difusas podem ser divididas em duas classes: uma referente a sistemas difusos linguísticos e outra referente a sistemas difusos do tipo Takagi-Sugeno. Nos parágrafos seguintes, abordar-se-á a problemática do treino *offline* de redes neuro-difusas, i.e., o modo de treino por lotes (Secção 4.5.1), sendo referidos alguns aspectos do modo incremental na Secção 5.3.3.

A selecção de um método de treino deve obedecer a critérios de eficiência e convergência, os quais foram discutidos na Secção 4.5.1. Aí, foram apontadas as limitações do algoritmo de retropropagação do erro em termos de velocidade de convergência e da probabilidade de obtenção de soluções subóptimas. Referiu-se ainda, na Secção 4.4, que o estimador dos mínimos quadráticos, utilizado em problemas de optimização linear, possibilita, sob certas condições, a convergência para o mínimo global. O treino de redes neuro-difusas efectuado neste trabalho será baseado nos dois métodos descritos.

De forma a diminuir os tempos de convergência da rede, optou-se por utilizar uma velocidade de aprendizagem adaptativa, com base em [Jang, 1993]. Assim sendo, se o erro diminuir durante num_{red} épocas consecutivas, a velocidade de aprendizagem é aumentada por um factor m^{up}. Se o erro aumentar durante num_{inc} épocas consecutivas ou oscilar num_{osc} vezes consecutivas, a velocidade de aprendizagem é reduzida por um factor m^{down}.

Quanto ao critério de paragem, o treino da rede é terminado no caso do critério RMSE atingir um valor satisfatório, no caso do seu valor estabilizar ou ainda no caso de se verificar um treino excessivo.

Consequentes difusos

Tal como em qualquer rede neuronal, o treino de uma rede neuro-difusa começa pela definição de um critério a optimizar. Mais uma vez, será utilizado o critério SSE (4.7). Assim, pela aplicação do método do gradiente, a adaptação dos pesos da rede neuro-difusa, i.e., centros e larguras das funções de pertença Gaussianas, será efectuada com base na expressão (4.9). Para a rede linguística de 5 camadas, as expressões de adaptação da rede são apresentadas em [Lin, 1995], no contexto da arquitectura NFCN. Neste trabalho, as expressões referidas foram modificadas, no sentido da incorporação de funções Gaussianas generalizadas, bem como de outros tipos de operadores de conjunção e disjunção, para além dos utilizados na arquitectura NFCN. Esta generalização da arquitectura inicial constitui uma das contribuições desta dissertação.

Nestes termos, os pesos associados à *camada de saída* obtêm-se pelas expressões (5.48) e (5.49):

$$d_o^{(p)} = y_o^{(p)} - \hat{y}_o^{(p)} \tag{5.48}$$

$$\frac{\partial E_o^{(p)}}{\partial c_{os}} = -d_o^{(p)} \cdot \frac{s_{os} a_s^{(p)}}{\sum_{k=1}^{|T(Y_o)|} s_{ok} a_k^{(p)}} \tag{5.49}$$

Nas expressões precedentes, $d_o^{(p)}$ representa a *p*-ésima amostra de saída, associada à *o*-ésima variável de saída. Definindo-se funções Gaussianas generalizadas, a expressão (5.49) é substituída por (5.50):

$$\frac{\partial E_o^{(p)}}{\partial c_{osL}} = -d_o^{(p)} \cdot \frac{s_{osL} a_s^{(p)}}{\sum_{k=1}^{|T(Y_o)|} (s_{okL} + s_{okR}) a_k^{(p)}} \tag{5.50}$$

A expressão anterior refere-se ao ajuste da componente esquerda da Gaussiana, o qual será

utilizado ao longo da exposição corrente. Quanto ao lado direito, as expressões são exactamente iguais, a menos do índice *L*, o qual deve ser substituído por *R*. Do mesmo modo, em relação à adaptação do desvio padrão, vem (5.51) para Gaussianas simples e (5.52) para Gaussianas generalizadas:

$$\frac{\partial E_o^{(p)}}{\partial s_{osL}} = -d_o^{(p5)} \cdot \frac{c_{os} a_s^{(p4)} \left(\sum_{k=1}^{T(Y_o)} s_{ok} a_k^{(p4)} \right) - a_s^{(p4)} \left(\sum_{k=1}^{T(Y_o)} c_{ok} s_{ok} a_k^{(p4)} \right)}{\left(\sum_{k=1}^{T(Y_o)} s_{ok} a_k^{(p4)} \right)^2} \tag{5.51}$$

$$\frac{\partial E_o^{(p)}}{\partial s_{osL}} = -d_o^{(p5)} \cdot \frac{c_{osL} a_s^{(p4)} \left(\sum_{k=1}^{T(Y_o)} \left(s_{okL} + s_{okR} \right) a_k^{(p4)} \right) - a_s^{(p4)} \left(\sum_{k=1}^{T(Y_o)} \left(c_{okL} s_{okL} + c_{okR} s_{okR} \right) a_k^{(p4)} \right)}{\left(\sum_{k=1}^{T(Y_o)} \left(s_{okL} + s_{okR} \right) a_k^{(p4)} \right)^2} \tag{5.52}$$

Da análise das expressões anteriores, verifica-se que, na utilização de funções Gaussianas generalizadas, se as componentes direita e esquerda da função forem iguais inicialmente, assim se manterão durante todo o treino. Dado que na aprendizagem da estrutura pelo agrupamento subtractivo se consideram funções Gaussianas convencionais, como tal com componentes esquerda e direita idênticas, é importante modificá-las de algum modo, de forma a que se ultrapasse o problema enunciado. Assim sendo, após a aprendizagem da estrutura, os centros direito e esquerdo de cada função são variados em 1% da amplitude do domínio, tal como segue (5.53):

$$\begin{aligned} c_L^{new} &= c_L - c_L \frac{X_{\max} - X_{\min}}{100} \\ c_R^{new} &= c_R + c_R \frac{X_{\max} - X_{\min}}{100} \end{aligned} \tag{5.53}$$

Na *quarta camada* não há quaisquer parâmetros a ajustar. No entanto, o sinal delta deve ser calculado, de modo a ser propagado para as camadas internas. Tal como foi abordado, o sinal delta da camada actual é calculado com base no mesmo sinal da camada exactamente posterior, resultando (5.54) e (5.55), para funções generalizadas e simples, respectivamente:

$$d_s^{(p4)} = \sum_{o=1}^{n} d_o^{(p5)} \cdot \left(\frac{\left(c_{osL} s_{osL} + c_{osR} s_{osR} \right) \left(\sum_{k=1}^{T(Y_o)} \left(s_{okL} + s_{okR} \right) a_k^{(p4)} \right)}{\left(\sum_{k=1}^{T(Y_o)} \left(s_{okL} + s_{okR} \right) a_k^{(p4)} \right)^2} - \frac{\left(s_{osL} + s_{osR} \right) \left(\sum_{k=1}^{T(Y_o)} \left(c_{okL} s_{okL} + c_{okR} s_{okR} \right) a_k^{(p4)} \right)}{\left(\sum_{k=1}^{T(Y_o)} \left(s_{okL} + s_{okR} \right) a_k^{(p4)} \right)^2} \right) \tag{5.54}$$

$$d_s^{(p4)} = \sum_{o=1}^{n} d_o^{(p5)} \cdot \frac{c_{os} s_{os} \sum_{k=1}^{|T(Y_o)|} s_{ok} a_k^{(p4)} - s_{os} \sum_{k=1}^{|T(Y_o)|} c_{ok} s_{ok} a_k^{(p4)}}{\left(\sum_{k=1}^{|T(Y_o)|} s_{ok} a_k^{(p4)} \right)^2} \tag{5.55}$$

Quanto à *terceira camada*, também aqui não há quaisquer parâmetros a ajustar, pelo que a sua única tarefa reside no cálculo do sinal delta. Genericamente, vem:

$$d_r^{(p3)} = \sum_{o=1}^{no_r} d_o^{(p4)} \frac{\partial a_o^{(p4)}}{\partial a_r^{(p3)}} \tag{5.56}$$

onde o índice *o* é utilizado na representação de neurónios da quarta camada e no_r designa o número de consequentes referentes à regra *r*.

Na versão original do algoritmo, a operação de disjunção é definida pela adição limitada (5.43), a qual não é uma função continuamente diferenciável, o que se apresenta como problemático, em resultado do referido na Secção 4.5.1. Deste modo, o autor optou por simplificar o problema, calculando a derivada correspondente com base unicamente na adição, obtendo-se (5.57):

$$\frac{\partial a_o^{(p4)}}{\partial a_r^{(p3)}} = 1 \tag{5.57}$$

Já a definição da norma-S pelo operador máximo requer a utilização de alguns artifícios no cálculo da derivada. Neste caso, é necessário armazenar o índice associado ao elemento que originou o máximo. Assim, a derivada em ordem ao elemento referido será 1, sendo 0 a derivada em relação aos elementos "derrotados" na obtenção do máximo (5.58):

$$\frac{\partial a_o^{(p4)}}{\partial a_r^{(p3)}} = \begin{cases} 1 & , a_o^{(p4)} = a_r^{(p3)} \\ 0 & , a_o^{(p4)} \neq a_r^{(p3)} \end{cases} \tag{5.58}$$

Em resultado dos aspectos de diferenciabilidade referidos na Secção 4.5.1, a situação ideal corresponderia à definição de um operador contínuo, tal como a adição algébrica (3.19). No entanto, o número de parcelas em (3.19) cresce exponencialmente com o número de operandos, havendo, para *g* operandos, 2^g parcelas. Ao elevado número de parcelas acrescem ainda os problemas associados à obtenção da derivada de maneira eficiente, pelo que o método referido não foi implementado.

Na *segunda camada* há novamente parâmetros a ajustar. A sua adaptação é efectuada, genericamente, com base nas expressões (5.59), para os centros, e (5.60), para as larguras:

$$\frac{\partial E_o^{(p)}}{\partial c_{ij}} = \left(\sum_{r=1}^{nr_i} d_r^{(p3)} \frac{\partial a_r^{(p3)}}{\partial a_i^{(p2)}} \right) \frac{\partial a_i^{(p2)}}{\partial c_{ij}} \tag{5.59}$$

$$\frac{\partial E_o^{(p)}}{\partial s_{ij}} = \left(\sum_{r=1}^{nr_i} d_r^{(p3)} \frac{\partial a_r^{(p3)}}{\partial a_i^{(p2)}} \right) \frac{\partial a_i^{(p2)}}{\partial s_{ij}} \tag{5.60}$$

onde nr_i representa o número de regras que têm o neurónio *i* por antecedente, sendo *j* a entrada referente ao termo *i*.

Em relação ao algoritmo original, a definição das funções Gaussianas é efectuada sem o factor "2" no denominador do expoente. Deste modo, a expressão apresentada em [Lin, 1995] aparece ligeiramente alterada, obtendo-se, para funções generalizadas, os resultados seguintes para os centros, (5.61), e para as larguras, (5.62):

$$\frac{\partial a_i^{(2)}}{\partial c_{ijL}} = \frac{x_j - c_{ijL}}{s_{ijL}^2} e^{-\frac{\left(x_j - c_{ijL}\right)^2}{2 s_{ijL}^2}} \tag{5.61}$$

$$\frac{\partial a_i^{(2)}}{\partial s_{ijL}} = \frac{\left(x_j - c_{ijL}\right)^2}{s_{ijL}^3} e^{-\frac{\left(x_j - c_{ijL}\right)^2}{2 s_{ijL}^2}} \tag{5.62}$$

No caso presente, a adaptação dos centros é efectuada de maneira rigorosamente igual, quer para Gaussianas simples, quer para generalizadas, exceptuando o facto de, no segundo caso, haver dois centros a adaptar.

Tal como na terceira camada, também na segunda há que decidir sobre o operador a utilizar na implementação da norma-T. Na versão original, o autor utiliza o operador mínimo, vindo (5.63):

$$\frac{\partial a_r^{(p3)}}{\partial a_i^{(p2)}} = \begin{cases} 1 & , a_r^{(p3)} = a_r^{(p2)} \\ 0 & , a_i^{(p3)} \neq a_i^{(p2)} \end{cases} \tag{5.63}$$

A expressão anterior resulta da aplicação de um artifício idêntico ao do operador máximo, uma vez que se tratam de operadores de truncatura e, como tal, descontínuos. Alternativamente, a aplicação do operador produto permite evitar o artifício anterior. Nesta situação, vem (5.64):

$$\frac{\partial a_r^{(p3)}}{\partial a_i^{(p2)}} = \prod_{k=1}^{na_r} a_k^{(p2)} \quad , k \neq i \tag{5.64}$$

A utilização de operadores algébricos apresenta vantagens em termos da suavidade da superfície de saída [Harris et al, 1993], além de permitir a aplicação directa do método do gradiente. Uma outra vantagem verificada experimentalmente consiste na obtenção de modelos mais precisos, tal como será analisado no Capítulo 6. Deste modo, dar-se-á preferência à utilização de operadores algébricos.

O treino livre dos parâmetros das funções de pertença poderá redundar na perda de integridade das mesmas. Concretamente, há o perigo dos centros direito e esquerdo de uma Gaussiana generalizada trocarem de posição, assim como os desvios padrões se tornarem negativos. Por conseguinte, após o ajuste dos parâmetros de cada função a integridade é verificada, alterando-se os parâmetros obtidos, em caso de necessidade. Assim, havendo perda de integridade, optou-se por atribuir, tanto ao centro direito como ao esquerdo, o seu valor médio. Quanto ao desvio padrão, esquerdo ou direito, no caso do seu valor se tornar negativo, atribui-se um valor "pequeno", quantificado como 1% da amplitude do domínio. Formalmente, obtém-se (5.65).

Contudo, é importante notar que as alterações efectuadas em função da integridade levam a que não se siga o verdadeiro gradiente, mas sim uma sua aproximação.

$$c_L > c_R \Rightarrow \begin{cases} c_L^{new} = \dfrac{c_L + c_R}{2} \\ c_R^{new} = \dfrac{c_L + c_R}{2} \end{cases} \tag{5.65}$$

$$s < 0 \Rightarrow s = \frac{X_{\max} - X_{\min}}{100}$$

Uma questão relevante no treino de qualquer tipo de rede neuronal relaciona-se com a adequação do número de parâmetros a ajustar com o número de amostras disponíveis. Este aspecto é importante no sentido de se evitarem situações de sobreajustamento. Neste sentido, o número de parâmetros a ajustar num modelo difuso linguístico é dado por (5.66):

$$numPar = numPar_{antecedente} + numPar_{consequente} = \sum_{j=1}^{m} 4\left|T(X_j)\right| + \sum_{o=1}^{n} 4\left|T(Y_o)\right| \tag{5.66}$$

Na utilização de funções de pertença Gaussianas simples, há apenas dois parâmetros por função a ajustar, pelo que o número total de parâmetros diminui para metade.

Resumidamente, a metodologia proposta para construção de modelos difusos linguísticos é a apresentada na Tabela 5.6.

1) Aprendizagem da estrutura (NFCN ou agrupamento subtractivo);
2) Aprendizagem de parâmetros:
   Enquanto não se verificar o critério de paragem
   i) Efectuar o processamento *forward* para cálculo do erro;
   ii) Efectuar o retroprocessamento para ajuste dos pesos pelo gradiente;
   iii) Verificar e garantir a integridade dos conjuntos difusos;
   Fim enquanto;
3) Testar o modelo e repetir 1) e/ou 2), se necessário.

Tabela 5.6. Algoritmo de identificação neuro-difusa para consequentes difusos.

Consequentes do tipo Takagi-Sugeno

Na implementação de modelos difusos Takagi-Sugeno, de ordem 0 ou 1, várias alternativas se afiguram como aplicáveis.

Numa primeira hipótese, o treino da rede poderá ser efectuado pela retropropagação. Deste modo, as únicas alterações em relação ao algoritmo anterior prendem-se com o ajuste dos parâmetros da camada linear de saída e com o cálculo do sinal delta para a camada de regras.

Dado que modelos do tipo Takagi-Sugeno de ordem 0 não são mais que casos particulares de modelos de ordem 1, a derivação do método para os últimos será considerada. Naturalmente, os resultados obtidos aplicam-se directamente a consequentes constantes. Deste modo, o ajuste dos pesos é efectuado com base em (5.67):

$$\frac{\partial E_o(p)}{\partial b_{orj}} = \frac{\partial E_o(p)}{\partial f_{or}(x(p))} \cdot \frac{\partial f_{or}(x(p))}{\partial b_{orj}} \tag{5.67}$$

Na expressão anterior, o primeiro factor é calculado como se segue:

$$\frac{\partial E_o(p)}{\partial f_{or}(x(p))} = -d_o(p) \cdot \frac{\partial a_o(p)}{\partial f_{or}(x(p))} = -d_o(p) \cdot \frac{a_r(p)}{\sum_{k=1}^{g} a_k(p)} \tag{5.68}$$

$$d_o(p) = y_o(p) - \hat{y}_o(p) \tag{5.69}$$

Quanto ao segundo factor, vem:

$$\frac{\partial f_{or}(x(p))}{\partial b_{orj}} = \begin{cases} 1 & , j = 0 \\ x_j(p) & , j \neq 0 \end{cases} \quad , j = 1,2,\cdots,m \tag{5.70}$$

Em relação ao sinal delta a propagar para as camadas internas, este obtém-se como em (5.71):

$$\begin{aligned} d_r(p) &= -d_o(p) \cdot \frac{\partial a_o(p)}{\partial \partial a_r(p)} = -d_o(p) \cdot \frac{f_{or}(x(p)) a_r(p) \sum_{k=1}^{g} a_k(p) - \sum_{k=1}^{g} f_{ok}(x(p)) a_k(p)}{\left(\sum_{k=1}^{g} a_k(p) \right)^2} \\ &= -d_o(p) \cdot \frac{\left(\sum_{j=1}^{m} b_{orj} x_j(p) + b_{or0} \right) a_r(p) \sum_{k=1}^{g} a_k(p) - \sum_{k=1}^{g} \left(\sum_{j=1}^{m} b_{okj} x_j(p) + b_{ok0} \right) a_k(p)}{\left(\sum_{k=1}^{g} a_k(p) \right)^2} \end{aligned} \tag{5.71}$$

Do ponto presente em diante, os cálculos a efectuar são rigorosamente os mesmos que foram apresentados para o caso de consequentes difusos.

Em consequência da linearidade da camada de saída, poder-se-á aplicar o estimador dos mínimos quadráticos na forma matricial. Deste modo, define-se a equação matricial (5.72), cuja solução é dada pela expressão (4.21), verificadas as condições de aplicabilidade.

$$Y^T = \Phi^T \cdot B \tag{5.72}$$

Na expressão (5.72), B denota uma matriz $(m+1)\cdot g \times n$ de parâmetros a identificar, definida do modo seguinte (5.73):

$$B = \begin{bmatrix} b_{110} & b_{210} & & b_{n10} \\ b_{111} & b_{211} & & b_{n11} \\ \vdots & \vdots & \vdots & \vdots \\ b_{11m} & b_{21m} & & b_{n1m} \\ \hline \vdots & \vdots & \cdots\ b_{orj}\ \cdots & \vdots \\ \hline b_{1g0} & b_{2g0} & & b_{ng0} \\ b_{1g1} & b_{2g1} & & b_{2g1} \\ \vdots & \vdots & \vdots & \vdots \\ b_{1gm} & b_{2gm} & & b_{ngm} \end{bmatrix} \tag{5.73}$$

Na expressão anterior, a cada coluna da matriz *B*, i.e., a cada saída do modelo, estão associados $(m+1)\cdot g$ parâmetros. Assim sendo, a mesma matriz divide-se em conjuntos de g grupos de $m+1$ linhas, em que cada um dos quais corresponde aos parâmetros definidos para os consequentes de cada regra. Como tal, o número total de parâmetros no consequente em modelos difusos do tipo Takagi-Sugeno de ordem 1 será $(m+1)\cdot g\cdot n$. Em modelos de ordem 0, a cada regra estará associado um único parâmetro, b_{or0}, pelo que a matriz *B* terá a dimensão $g\times n$.

Ainda na expressão (5.72), *Y* representa a matriz $n\times N$ de saídas reais desejadas, sendo a matriz Φ, de dimensão $(m+1)\cdot g\times N$, definida como em (5.74):

$$\Phi = \begin{bmatrix} \dfrac{a_1^{[13]}}{\sum_{r=1}^{g} a_r^{[13]}} X^{[1]} & \dfrac{a_1^{[23]}}{\sum_{r=1}^{g} a_r^{[23]}} X^{[2]} & \cdots & \dfrac{a_1^{[N3]}}{\sum_{r=1}^{g} a_r^{[N3]}} X^{[N]} \\ \dfrac{a_2^{[13]}}{\sum_{r=1}^{g} a_r^{[13]}} X^{[1]} & \dfrac{a_{21}^{[23]}}{\sum_{r=1}^{g} a_r^{[23]}} X^{[2]} & \cdots & \dfrac{a_2^{[N3]}}{\sum_{r=1}^{g} a_r^{[N3]}} X^{[N]} \\ \vdots & \vdots & \cdots & \vdots \\ \dfrac{a_{g1}^{[13]}}{\sum_{r=1}^{g} a_r^{[13]}} X^{[1]} & \dfrac{a_1^{[g3]}}{\sum_{r=1}^{g} a_r^{[g3]}} X^{[2]} & \cdots & \dfrac{a_{g1}^{[N3]}}{\sum_{r=1}^{g} a_r^{[N3]}} X^{[N]} \end{bmatrix} \tag{5.74}$$

$$X^{[p]} = \begin{bmatrix} 1 & x_1^{[p]} & x_2^{[p]} & \cdots & x_m^{[p]} \end{bmatrix}^T \quad , p = 1,2,\cdots,N$$

Claramente, para modelos Takagi-Sugeno de ordem zero o vector $X^{(p)}$ reduz-se a $X^{(p)} = 1$.

Em consequência da estrutura da matriz *B*, verifica-se que o número total de parâmetros a ajustar em modelos de ordem 0 e de ordem 1 será dado, respectivamente, por (5.75) e (5.76):

$$numPar = numPar_{antecedente} + numPar_{consequente} = \sum_{j=1}^{m} 4\left|T(X_j)\right| + g\cdot n \tag{5.75}$$

$$numPar = numPar_{antecedente} + numPar_{consequente} = \sum_{j=1}^{m} 4\left|T(X_j)\right| + (m+1)\cdot g\cdot n \tag{5.76}$$

Das expressões (5.66), (5.75) e (5.76) conclui-se que, à medida que o número de entradas no sistema aumenta, o número de parâmetros em modelos Takagi-Sugeno de ordem 1 se torna claramente superior ao número verificado nas restantes estruturas consideradas. Nota-se ainda que modelos Takagi-Sugeno de ordem 0 constituem, em geral, a arquitectura com o menor número de parâmetros a ajustar, excepto nas situações em que o número de regras seja significativamente elevado. Nesta situação, o número de parâmetros em modelos linguísticos poderá ser menor, bastando para tal que o número de funções de pertença nos consequentes não seja muito elevado, e.g., se situe nos limites da interpretabilidade.

Em virtude do peso computacional associado ao algoritmo de um só passo, utiliza-se frequentemente a versão recursiva do estimador dos mínimos quadráticos, tal como se segue (5.77):

$$\begin{aligned} B(p) &= B(p-1) + P(p)\Phi^{(p)}\left[Y^{(p)} - \hat{Y}^{(p)}\right]^T \\ P(p) &= P(p-1) - \frac{P(p-1)\Phi^{(p)}\Phi^{(p)T}P(p-1)}{1+\Phi^{(p)T}P(p-1)\Phi^{(p)}} \end{aligned} \quad , p = 1,2,\cdots,N \tag{5.77}$$

Aqui, $\Phi^{(p)}$ representa a *p*-ésima coluna da matriz Φ, sendo $Y^{(p)}$ e $\hat{Y}^{(p)}$ vectores de dimensão $n\times1$ denotando, respectivamente, a saída real e a saída do modelo.

O método descrito apresenta a vantagem de garantir a obtenção da solução óptima, para valores fixos dos pesos da segunda camada. Deste modo, é frequente recorrer-se à sua utilização segundo um esquema híbrido [Jang, 1993]. Assim, a rede executa um passo em que funciona no modo *forward* até à camada de regras, necessário para a determinação da matriz Φ. Neste ponto, os parâmetros óptimos de *B* são obtidos pelo método RLS, após o que o erro de modelização é calculado. Na segunda fase, a rede funciona no modo de retropropagação, adaptando-se os pesos da segunda camada tal como se descreveu anteriormente, com base no cálculo do sinal delta relativo à camada de regras (5.71).

O esquema descrito possibilita a redução significativa do número de épocas para a convergência da rede, pelo que será utilizado neste trabalho. Porém, é importante notar que o tempo de cada época é notoriamente superior ao correspondente à utilização do método do gradiente, em consequência da aplicação do algoritmo RLS. A vantagem fundamental reside na redução significativa do número de épocas necessárias à obtenção de resultados satisfatórios. Uma outra vantagem do método reside na facilidade de implementação em linha, bastando para tal introduzir um factor de esquecimento [Ljung, 1987] (Secção 5.3.3). Para além do referido, os modelos de Takagi-Sugeno de ordem 1 apresentam, regra geral, maior precisão comparativamente aos modelos linguísticos, com recurso a um menor número de regras. Tal facto deve-se à maior flexibilidade no consequente, bem como às melhores propriedades de convergência.

Uma outra abordagem, não implementada neste trabalho, é frequentemente utilizada no contexto das redes RBF. Tal estratégia reside na implementação de um esquema em tudo idêntico ao anterior, à excepção do treino da camada de Gaussianas.

Aqui, os centros das funções de pertença são actualizados por aprendizagem competitiva, sendo as larguras das Gaussianas actualizadas pela heurística dos vizinhos mais próximos [Moody e Darken, 1989]. O método referido tem particular interesse em implementações em linha, em virtude da sua elevada eficiência computacional [Pereira, 1996]. No entanto, o ajuste da largura das funções de pertença constitui um aspecto de grande importância na qualidade da solução obtida. Uma vez que neste método as larguras não são adaptadas livremente, os resultados obtidos não são, naturalmente, os óptimos. No entanto, o algoritmo possibilita, em geral, boas soluções de compromisso entre eficiência e aplicabilidade em tempo real.

Em jeito de síntese, a metodologia proposta para construção de modelos difusos de Takagi-Sugeno é a apresentada na Tabela 5.7.

5.3.3. Aprendizagem em Linha

Na modelização e controlo de sistemas variantes no tempo, é importante que o modelo possa adaptar-se em tempo real, de forma a captar as variações na dinâmica do sistema. Essa adaptação deve ser efectuada em linha, i.e., durante o funcionamento do sistema, sendo portanto, fundamental satisfazer os critérios de tempo real impostos.

1) Aprendizagem da estrutura (NFCN ou agrupamento subtractivo);
2) Aprendizagem de parâmetros:
   Enquanto não se verificar o critério de paragem
   i) Efectuar o processamento *forward*
   ii) Optimizar parâmetros dos consequentes pelo método RLS;
   iii) Calcular o erro;
   iv) Efectuar o retroprocessamento, para ajuste dos parâmetros dos antecedentes pelo gradiente;
   v) Verificar e garantir a integridade dos conjuntos difusos;
   Fim enquanto;
3) Testar o modelo e repetir 1) e/ou 2), se necessário.

Tabela 5.7. Algoritmo de identificação neuro-difusa para consequentes de Takagi-Sugeno.

Assim sendo, os parâmetros do modelo devem ser adaptados à medida que novas amostras vão sendo obtidas. Nesta situação, propriedades de convergência satisfatórias são importantes, uma vez que há que cumprir requisitos de desempenho, segurança e previsibilidade, inerentes aos processos de produção.

No ponto presente deste trabalho, não se considera o problema da aprendizagem da estrutura em linha. Desta maneira, assume-se a obtenção de um modelo inicial *offline*, cujos parâmetros são adaptados incrementalmente.

Aprendizagem recursiva em sistemas lineares

Em sistemas lineares, o problema da adaptação recursiva é bem conhecido, havendo vários métodos efectivos. Neste sentido, é comum utilizar-se o método dos mínimos quadráticos recursivos.

A principal desvantagem do método dos mínimos quadráticos recursivos reside na incapacidade de identificação a partir do momento em que se verifiquem as condições de convergência. De facto, uma vez efectuada a estimação dos parâmetros do sistema, a matriz *P*, de cujos valores se pode extrair informação qualitativa relativamente à magnitude do erro de estimação, tomará valores pequenos. Desta forma, as equações da expressão (5.77) não permitirão que a actualização dos parâmetros se continue a processar. Na verdade, verifica-se que os parâmetros se mantêm aproximadamente constantes em iterações sucessivas, i.e., $B(p) \approx B(p-1)$, mesmo que os parâmetros reais variem. Consequentemente, de forma a que o modelo leve em consideração as características variantes do sistema em causa, é importante que, de alguma forma, as amostras mais remotas tenham um peso menor sobre o modelo. Uma solução simples consiste numa formulação pesada do critério de erro, sendo atribuídos factores de ponderação mais elevados às amostras mais recentes. Esta metodologia tem por consequência a adição de um *factor de esquecimento*, l, à formulação recursiva original (5.77), resultando (5.78) [Ljung, 1987]:

$$P(p) = \frac{1}{l}\left[P(p-1) - \frac{P(p-1)\Phi^{T}(p)\Phi(p)P(p-1)}{l + \Phi(p)P(p-1)\Phi^{T}(p)}\right] \quad , \; l \in]0;1] \tag{5.78}$$

Assim sendo, no caso em que l=1, verifica-se a situação recursiva habitual, em que não há esquecimento. À medida que o factor de esquecimento diminui, o peso das amostras passadas

torna-se cada vez menor. No entanto, um factor de esquecimento demasiado baixo poderá ocasionar problemas de instabilidade numérica. Valores típicos de l situam-se entre 0.95 e 1.

O problema essencial do método dos mínimos quadráticos recursivos com esquecimento reside no facto de que, em intervalos de tempo em que não exista variação da entrada e da saída do sistema, a actualização da matriz P (5.78) pode ser aproximada por (5.79):

$$P(p) = \frac{P(p-1)}{l} \tag{5.79}$$

Claramente, os valores da matriz P aumentarão em cada iteração, podendo atingir números elevados e conduzindo à instabilidade do método.

Assim, pode-se afirmar que, se por um lado valores pequenos de P indicam boas características de estimação, por outro indicam a perda da capacidade de identificação do método. Inversamente, valores elevados de P são indicadores de uma estimação deficiente, garantindo, porém, a capacidade de identificação do método, em virtude da incerteza associada aos valores nos parâmetros. Na tentativa de solucionar os dois problemas referidos, vários métodos têm sido propostos. Uma das metodologias mais comuns consiste na *gestão da matriz de co-variância* [Ljung, 1987], que se caracteriza basicamente pelo controlo da grandeza dos seus elementos, não permitindo valores excessivamente elevados nem reduzidos.

Aprendizagem recursiva em sistemas não lineares

Em relação a sistemas não lineares, mais uma vez não se encontram metodologias rigorosas. Como tal, são aplicadas técnicas decorrentes da Inteligência Artificial, nomeadamente redes neuronais e lógica difusa. Uma vez que a capacidade de aproximação do modelo deve ser maximizada, estruturas do tipo Takagi-Sugeno de ordem 1 são favorecidas. No entanto a necessidade de eficiência computacional, bem como o menor número de parâmetros a ajustar, favorece os sistemas linguísticos descritos neste capítulo, os quais são mais eficientes que as estruturas Takagi-Sugeno. No meio termo, situam-se os modelos de ordem 0, os quais apresentam bons compromissos entre eficiência, precisão e número de parâmetros a ajustar. Ainda no tema da eficiência e parcimónia, não sendo a interpretabilidade um objectivo a atingir, a utilização de funções de pertença Gaussianas simples revela-se mais adequada, em consequência do menor número de parâmetros a ajustar, de onde resultam menores tempos de adaptação. Tal como se verificará experimentalmente no capítulo posterior, as funções Gaussianas simples possibilitam a obtenção de modelos com grau de precisão idêntico aos obtidos através de funções generalizadas.

Na aprendizagem em linha, o modo de treino por lotes não é aplicável directamente, excepto se se considerar uma janela de estimação constituída pelas últimas N amostras. Porém, a estratégia referida tem associado um maior peso computacional, o qual poderá limitar a sua aplicabilidade em tempo real (Secção 4.5.1). Deste modo, é mais eficiente adaptar-se o modelo em modo incremental, à medida que se obtêm amostras de dados do sistema. Por conseguinte, o método do gradiente descrito nos capítulos anteriores é aproximado pela sua versão incremental, na qual o ajuste dos pesos é efectuado após a apresentação de cada amostra. Um aspecto importante deriva do facto de, neste caso, não ser seguida a direcção do gradiente verdadeiro, mas sim uma sua aproximação. Para que o gradiente incremental esteja o mais próximo possível do gradiente real, a velocidade de aprendizagem deve ser mantida em valores baixos [Brown e Harris, 1994]. Deste modo, optou-se por utilizar uma velocidade fixa suficientemente pequena.

Na aprendizagem em linha, requer-se que a *propriedade da localidade* se verifique, i.e., que entradas distintas influenciem diferentes conjuntos de pesos da rede neuronal. Esta propriedade

requer que o ajuste dos parâmetros altere a saída de forma local e não global. Assim sendo, as funções sigmoidais, presentes em redes MLP, não se afiguram adequadas, uma vez que o seu suporte se estende por todo o domínio, podendo originar alterações significativas no comportamento da rede entre a apresentação de dois padrões consecutivos. Funções deste tipo, i.e., funções globais, levam a que modificações nos seus parâmetros alterem, de forma global, o mapeamento efectuado pela rede, alterações essas que se farão sentir em zonas extensas do espaço de entrada-saída. Ao invés, as estruturas com funções de activação locais, e.g., Gaussianas, apresentam vantagens importantes. Neste caso, a alteração dos parâmetros da função afecta apenas localmente o mapeamento global da rede, em virtude da sua natureza compacta. Naturalmente, as arquitecturas neuro-difusas utilizadas gozam da propriedade da localidade.

Aprendizagem em linha de modelos difusos

No treino incremental de redes neuro-difusas, o procedimento genérico é em tudo idêntico ao da aprendizagem por lotes, à excepção do facto de que os parâmetros são agora ajustados após a apresentação de cada padrão. Assim sendo, para modelos linguísticos, o procedimento genérico de aprendizagem de parâmetros em linha é o apresentado na Tabela 5.8.

1) Obter um modelo inicial fora de linha;
2) Apresentar a nova amostra de dados à rede;
3) Efectuar o processamento *forward*;
4) Efectuar o retroprocessamento para ajuste dos pesos;
5) Ajustar os parâmetros das funções de pertença do antecedente pelo gradiente;
6) Verificar e garantir a integridade dos conjuntos difusos;
7) Voltar a 2.

Tabela 5.8. Algoritmo de aprendizagem de parâmetros em linha em modelos linguísticos.

Para modelos de Takagi-Sugeno, a aprendizagem de parâmetros em linha é conduzida de acordo com a Tabela 5.9, com base nos aspectos referidos para a optimização linear.

1) Obter um modelo inicial fora de linha;
2) Apresentar a nova amostra de dados à rede;
3) Efectuar o processamento *forward*;
4) Ajustar os parâmetros dos consequentes pelo método RLS com esquecimento;
5) Efectuar o processamento para trás:
6) Ajustar os parâmetros das funções de pertença do antecedente pelo gradiente;
7) Verificar e garantir a integridade dos conjuntos difusos;
8) Voltar a 2.

Tabela 5.9. Algoritmo de aprendizagem de parâmetros em linha em modelos Takagi-Sugeno.

Um esquema alternativo, bastante utilizado no contexto de redes RBF, consiste no ajuste das funções de pertença no antecedente pela versão recursiva do algoritmo *k-nearest neighbours* [Moody e Darken, 1989]. Aqui, as suas larguras são determinadas pela heurística dos vizinhos mais próximos, o que, no entanto, limita a flexibilidade das funções de pertença. A vantagem do método reside na obtenção, em geral, de boas soluções de compromisso entre precisão e eficiência, garantindo a propriedade da localidade.

Aprendizagem da Estrutura

A problemática da adaptação da estrutura tem vindo a merecer uma atenção por parte da comunidade científica que começa a ser notória. Esta área, designada por *aprendizagem construtiva*, afigura-se bastante promissora, particularmente no sentido do desenvolvimento de sistemas autónomos, na medida em que se auto organizam sem a inclusão de qualquer conhecimento prévio. O seu problema essencial deriva de questões colocadas relativamente à convergência da aprendizagem. De facto, se para estruturas fixas o problema não é simples, várias questões se levantam em estruturas adaptativas. Até ao momento, a grande maioria dos algoritmos desenvolvidos, e.g., [Juang e Lin, 1998; Cho e Wang, 1996], implementam metodologias heurísticas, por vezes pesadas a nível computacional. Apesar do grande potencial científico associado à área referida, a aprendizagem em linha da estrutura não será analisada no trabalho de dissertação apresentado, constituindo matéria de investigação futura.

5.4. Interpretabilidade

A filosofia dos sistemas difusos assenta sobre a possibilidade de interpretação linguística que os caracteriza. Porém, o aspecto citado é frequentemente ignorado, dando-se relevância predominante aos factores associados às capacidades de aproximação funcional. No entanto, tal como afirmam Nauck e Kruse [Nauck e Kruse, 1999], no caso de a interpretabilidade não constituir um objectivo de modelização, coloca-se, naturalmente, a hipótese de utilização de outras classes de metodologias. Uma vez que a possibilidade de desenvolvimento de modelos transparentes constitui uma das motivações do recurso a estruturas difusas, tal aspecto é abordado nesta secção, como uma das contribuições originais do trabalho presente.

Uma interrogação natural, prende-se com a importância de se desenvolverem modelos interpretáveis. De facto, na esmagadora maioria dos problemas de modelização, são procuradas soluções adequadas em termos de capacidades de representação. Porém, em outras situações, como por exemplo a planta de branqueamento estudada na Secção 6.4, procura-se um melhor conhecimento do sistema em análise. Nesta caso, a modelização difusa apresenta-se, claramente, como a metodologia mais viável.

Assim sendo, relativamente à identificação neuro-difusa descrita nas secções precedentes, colocam-se questões associadas à transparência do modelo resultante do treino. Assim, em primeiro lugar, sendo a interpretabilidade linguística um dos objectivos da construção de um modelo difuso, os sistemas de Takagi-Sugeno de ordem 1 revelam-se inadequados, uma vez que, nessa classe, os consequentes não representam conjuntos difusos. Por conseguinte, a satisfação dos objectivos de interpretabilidade requer a utilização de modelos linguísticos, onde os consequentes representam conjuntos difusos. Alternativamente, poder-se-ão recorrer a modelos Takagi-Sugeno de ordem 0, com consequentes constantes, os quais, dada a sua estrutura, apresentam bons compromissos entre

precisão e interpretabilidade.

Em segundo lugar, do ajuste livre dos parâmetros pode resultar um conjunto complexo de funções de pertença, às quais será difícil associar termos linguísticos. Deste modo, é fundamental que se imponham restrições quanto ao ajuste dos parâmetros das funções de pertença, de forma a que o modelo final seja interpretável. Ainda na linha referida, a utilização de funções Gaussianas generalizadas revela-se apelativa, dado que a sua maior flexibilidade permitirá gerir a sobreposição e capacidade de distinção entre funções de maneira mais eficiente.

Assim, poder-se-ão definir três critérios fundamentais a considerar, no sentido da verificação da interpretabilidade de sistemas difusos. O primeiro, e mais importante, relaciona-se com o aspecto acabado de enunciar, ou seja, a capacidade de distinção entre funções de pertença. Os seguintes derivam dos aspectos cognitivos humanos, segundo os quais o número de regras e de funções de pertença associadas a cada variável não deve ser excessivo. No caso presente, o número de regras e funções de pertença obtidas é monitorizado pelo modelizador, recaindo a escolha sobre situações com compromissos aceitáveis entre precisão e interpretabilidade.

5.4.1. Fusão de Funções de Pertença Similares

O primeiro passo na consecução do objectivo da interpretabilidade de um modelo difuso prende-se com a detecção e fusão de funções de pertença semelhantes.

Tal como se verificou, a aprendizagem da estrutura por meio de técnicas de agrupamento conduz à obtenção de funções de pertença com um grau de similaridade elevado, o que não só torna o modelo pouco transparente, como origina um número excessivo de parâmetros a ajustar, e o consequente peso computacional. Assim sendo, é útil fundir funções de pertença que apresentem um grau de similaridade elevado.

Na procura de funções de pertença semelhantes, com o objectivo de se simplificar uma base de regras, Setnes [Setnes, 1995] concluiu que a medida S_1 (3.27) se revela a mais satisfatória. A implementação computacional do integral necessário ao cálculo de S_1 é efectuada pela sua aproximação através de um somatório, recorrendo à definição. Deste modo, quanto maior for o número de pontos discretos a considerar, n_{int}, maior será o rigor na aproximação. Contudo, um valor excessivo poderá redundar num custo computacional elevado. Assim sendo, concluiu-se experimentalmente que a utilização de 100 pontos uniformemente distribuídos pelo universo de discurso constitui um bom compromisso entre precisão e peso computacional.

Após a identificação de um par de funções de pertença, $\tilde{A}$ e $\tilde{B}$, suficientemente semelhantes, i.e., cujo grau de similaridade seja superior a um dado limiar, é importante definir um método para a realização da sua fusão. A selecção da melhor função a definir como resultado da fusão revela-se importante, especialmente no caso em que a redução da base de regras seja efectuada após a optimização do modelo [Setnes, 1995]. Uma vez que, no caso desta dissertação, a fusão de funções de pertença é efectuada anteriormente à optimização, o problema da selecção óptima da nova função não se apresenta com um carácter tão determinante. Assim sendo, optou-se por efectuar a fusão de funções semelhantes pela criação de uma nova função cujos parâmetros se obtêm pela média dos parâmetros correspondentes nas funções originais. Deste modo, para funções Gaussianas generalizadas, a função obtida terá por centro esquerdo a média dos centros esquerdos originais, por desvio padrão esquerdo, a média dos desvios esquerdos originais, e assim sucessivamente, tal como se representa na Figura 5.14. Aí, as funções originais são representadas a tracejado, sendo a função resultante da fusão representada a traço contínuo.

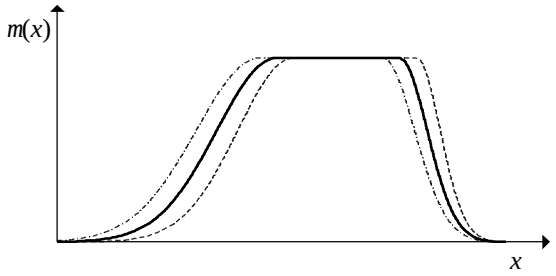


Figura 5.14. Fusão de funções de pertença.

A fusão de funções de pertença no consequente leva à actualização da base de regras. Nesse sentido, as regras referentes aos termos em causa passam a conter o novo termo obtido como um dos seus consequentes. O mesmo se passa em relação à fusão de conjuntos difusos no antecedente. Aqui, nas regras originais, as premissas e conclusões são alteradas de forma a incorporarem os novos termos resultantes da fusão. Os aspectos referidos são ilustrados na Figura 5.15.

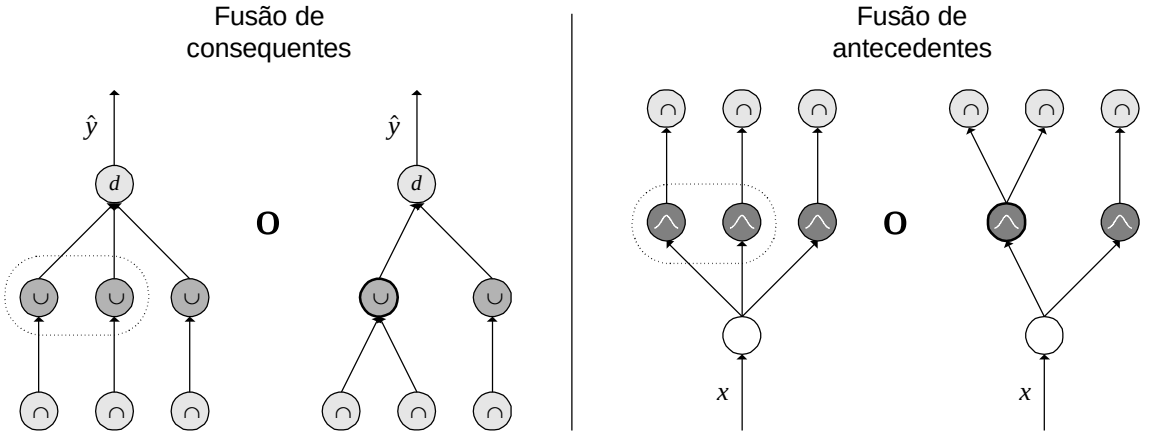


Figura 5.15. Simplificação da base de regras por fusão de conjuntos difusos.

A fusão de funções de pertença pode conduzir à simplificação da base de regras, em consequência da obtenção de regras redundantes. De facto, após a combinação de antecedentes, pode dar-se o caso de algumas regras apresentarem as mesmas premissas. No caso do mesmo se verificar nos consequentes, torna-se nítido que as regras em causa se repetem, sendo, por isso, combinadas numa única. Por outro lado, poderão também ocorrer situações de inconsistência, decorrentes da obtenção de regras com premissas iguais e consequentes distintos. O caso descrito poderá indiciar uma aprendizagem deficiente da estrutura. Na verdade, antecedentes similares deveriam originar consequentes similares. Poder-se-á também dar o caso da "inconsistência" resultar do valor atribuído ao limiar de fusão, podendo este estar bastante próximo do grau de similaridade entre os consequentes, sendo, contudo, ligeiramente superior. De qualquer modo, optou-se por efectuar a fusão de consequentes de forma a serem ultrapassadas as situações de inconsistência. Assim sendo, para que a consistência se mantenha, os consequentes relativos a regras com premissas iguais são fundidos, o que, por sua vez, conduz à combinação das regras em causa numa única (Figura 5.16). De forma a que o modelizador tenha a noção concreta de que a base de regras poderá conter uma inconsistência, é também fornecida informação, relativamente à operação efectuada. Deste modo, o procedimento a aplicar pelo modelizador poderá passar pelo

ajuste do limiar ou pela repetição da aprendizagem da estrutura.

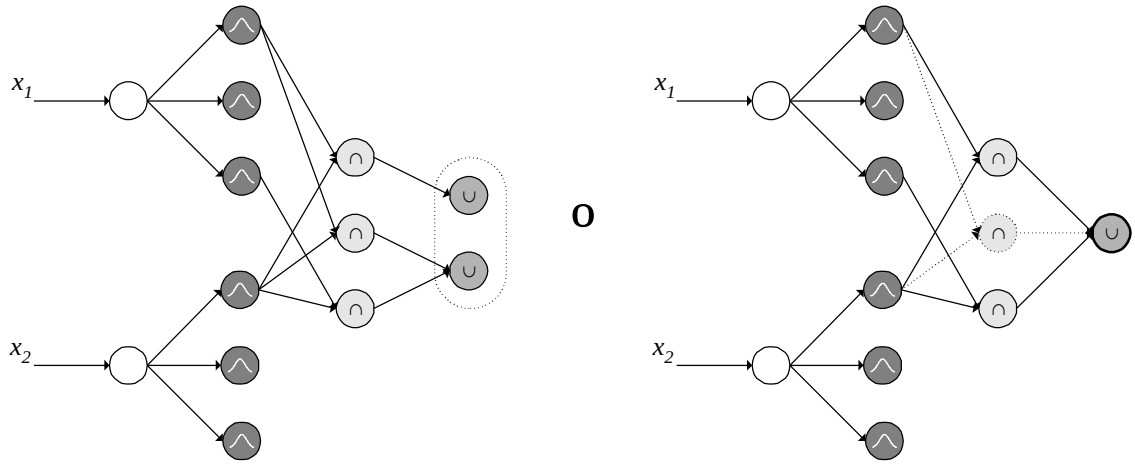


Figura 5.16. Combinação de regras para consistência.

Resumidamente, o algoritmo de simplificação de uma base de regras poderá ser sintetizado como na Tabela 5.10, onde *l* representa o limiar de similaridade para a fusão de conjuntos difusos.

1) Em cada domínio, medir a similaridade entre todos os pares de funções de pertença;
2) Procurar o par mais semelhante: $S_1(\tilde{A}, \tilde{B})$;
3) Se $S_1(\tilde{A}, \tilde{B}) < l$
   i) Terminar.
   Caso contrário
   i) Actualizar a base de regras;
   ii) Voltar a 1.
   Fim Se.

Tabela 5.10. Algoritmo de simplificação da base de regras.

No algoritmo anterior, a base de regras é actualizada após a fusão de cada par de funções. Desta forma, a função criada após a fusão é tida em consideração na iteração seguinte. Assim sendo, em iterações posteriores, a função obtida poderá ser fundida com uma outra. Deste modo, na operação de fusão, é importante dar-se um peso mais significativo às funções obtidas a partir de fusões passadas. Por conseguinte, a fusão de dois conjuntos difusos $\tilde{A}$ e $\tilde{B}$ é efectuada como se segue [Setnes, 1995] (5.80):

$$C_p = \frac{n_A A_p + n_B B_P}{n_A + n_B} \tag{5.80}$$

A expressão anterior descreve o cálculo dos parâmetros do novo conjunto difuso $\tilde{C}$ com base na média ponderada dos parâmetros das funções em causa. Em (5.80), C_p denota o vector de parâmetros do conjunto difuso $\tilde{C}$, e.g., centro e desvio padrão, esquerdo e direito, e n_A e n_B representam o número de funções previamente fundidas, antes da criação do conjunto $\tilde{A}$ e do

conjunto $\tilde{B}$, respectivamente.

5.4.2. Treino Restringido de Parâmetros

Após a simplificação da base de regras pela fusão de funções de pertença, há que garantir a manutenção da interpretabilidade linguística durante a optimização de parâmetros. Claramente, os métodos apresentados na Secção 5.3.2 nada possibilitam nesse sentido. Assim sendo, optou-se por monitorizar o procedimento de optimização de forma a que capacidade de distinção entre as funções de pertença de cada variável seja garantida.

Por conseguinte, estabeleceu-se como condição de interpretabilidade linguística que as funções de pertença de uma mesma variável não se sobreponham em demasia. Assim, heuristicamente, considera-se que o grau de sobreposição entre duas funções de pertença é excessivo no caso do supremo do seu suporte, i.e., o seu zero direito, ultrapassar o zero direito da segunda, procedendo-se analogamente para a componente esquerda da Gaussiana. Formalmente, vem (5.81):

$$\begin{aligned} c_{kR} + 3s_{kR} &\le c_{iR} + 3s_{iR} \\ c_{kL} - 3s_{kL} &\le c_{jL} - 3s_{jL} \end{aligned} \tag{5.81}$$

onde o índice *i* se refere à função mais próxima à direita da função de pertença de índice *k*, sendo o índice *j* relativo à sua vizinha mais próxima à esquerda. Naturalmente, o critério enunciado aplica-se a funções Gaussianas generalizadas, sendo facilmente extensível a Gaussianas simples, situação em que se analisa apenas a sobreposição da função em causa com a sua vizinha mais próxima. No caso da sobreposição ultrapassar os limites estabelecidos pelo critério (5.81), o desvio padrão da função de índice *k* é alterado de forma a que se verifique a restrição apresentada. Assim, para a componente direita da função tem-se (5.82), vindo para a componente esquerda (5.83):

$$s_{kR} = \frac{c_{iR} + 3s_{iR} - c_{kR}}{3} \tag{5.82}$$

$$s_{kL} = \frac{c_{jL} - 3s_{jL} - c_{kL}}{-3} \tag{5.83}$$

Para além da monitorização do desvio padrão, verificou-se ser importante monitorizar também a distância entre funções. Este procedimento tem por base evitar que ocorram situações de inclusão de funções de pertença em outras funções, de tal modo que a sua fusão não se efectuasse, em virtude do reduzido grau de similaridade verificado. Para além deste aspecto, modelos com funções suficientemente espaçadas são mais facilmente interpretáveis. Como tal, definiu-se o critério seguinte para a distância mínima entre funções (5.84):

$$\begin{aligned} c_{iL} - c_{kR} &\le \alpha\left(U_{\max} - U_{\min}\right) \\ c_{kL} - c_{jR} &\le \alpha\left(U_{\max} - U_{\min}\right) \end{aligned} \tag{5.84}$$

onde U_{max} e U_{min} denotam, respectivamente os valores máximo e mínimo do universo de discurso, sendo $\alpha \in [0;1]$ a distância mínima percentual entre duas funções vizinhas, relativamente à amplitude do domínio. Naturalmente, o valor a atribuir ao parâmetro α dependerá do número de funções de pertença definidas para a variável em causa.

O desenvolvimento de modelos difusos interpretáveis e suficientemente precisos requer algum relaxamento quanto às restrições sobre os parâmetros. Por conseguinte, em lugar de se

restringir fortemente o seu ajuste, verificou-se experimentalmente que se obtinham melhores resultados em termos de precisão relaxando um pouco o procedimento de monitorização. Nomeadamente em relação à máxima sobreposição aceitável, foram testados outros critérios, por exemplo de comparação do zero da função monitorizada com o centro do vizinho esquerdo e direito, tendo-se obtido geralmente resultados insatisfatórios em termos de capacidade de aproximação. Assim, verificou-se experimentalmente que o critério definido apresenta um compromisso aceitável entre interpretabilidade e capacidade de previsão.

Em consequência do relaxamento das restrições, pode dar-se o caso de o nível de interpretabilidade do modelo não ser suficiente. Como tal, a base de regras é simplificada periodicamente, i.e., de *x* em *x* épocas, situação em que se efectua a fusão de funções de pertença semelhantes. Uma questão que se coloca naturalmente consiste no porquê da não utilização pura e simples do procedimento de simplificação periódica. Tal facto deve-se a que, em muitas situações, as funções não apresentam qualquer similaridade, estando sobrepostas de forma altamente complexa, e.g., inclusões, funções "atravessando" outras, etc. Deste modo, o algoritmo de simplificação não possibilitaria a solução desses problemas. Ao invés, a simplificação apresenta-se vantajosa partindo do pressuposto de que a sobreposição entre as funções de pertença é suficientemente simples, o que se garante pela aprendizagem restringida de parâmetros.

Em jeito de síntese, a Tabela 5.11 resume o procedimento de implementação de modelos difusos interpretáveis linguisticamente.

1) Aprendizagem da estrutura (NFCN ou agrupamento subtractivo);
2) Aprendizagem de parâmetros:
   Enquanto não se verificar o critério de paragem
   i) Efectuar o processamento *forward* para cálculo do erro;
   ii) Efectuar o retroprocessamento para ajuste dos pesos;
   iii) Efectuar as restrições sobre as larguras e centros das funções de pertença;
   iv) Se tiverem passado *x* épocas, fundir os pares de funções semelhantes;
   Fim enquanto;
3) Testar o modelo e repetir 1) e/ou 2), se necessário.

Tabela 5.11. Algoritmo de desenvolvimento de modelos interpretáveis.

5.5. Sumário

O capítulo presente começou por apresentar algumas das metodologias utilizadas na construção automática de modelos difusos. De entre as metodologias referidas, concluiu-se que as estruturas neuro-difusas se afiguram particularmente interessantes, dado possibilitarem a conjugação da transparência dos sistemas difusos com a capacidade de aprendizagem inerente às redes neuronais.

Ainda na Secção 5.1, foram apresentadas as categorias e utilizações fundamentais das

diferentes estratégias neuro-difusas, definidas de acordo com o tipo de informação processada, numérica ou difusa.

Na Secção 5.2, foi abordado o problema da aprendizagem da estrutura em sistemas difusos, a qual se reveste de importância crucial na qualidade final do modelo. Verificou-se que os métodos baseados em técnicas de agrupamento de classes se revelam particularmente adequados uma vez que possibilitam a obtenção de um conjunto de regras relevantes, evitando o problema da explosão da base regras, inerente a estruturas com partições do tipo grelha. De entre as várias possibilidades disponíveis optou-se pelo algoritmo de agrupamento subtractivo, em virtude da suas características se adequarem à inicialização de estruturas a utilizar em problemas de optimização, tal como sucede na identificação neuro-difusa. O algoritmo apresenta ainda como vantagens a sua robustez ao ruído e o facto de fornecer uma estimação do número de grupos necessários, embora este valor dependa do valor especificado para o raio da vizinhança.

Na mesma secção foi analisada a questão da selecção de entradas relevantes para um modelo. Aqui, verificou-se o carácter heurístico da esmagadora maioria das técnicas disponíveis, tendo-se optado por um método particularmente simples e eficiente, adequado a problemas de modelização difusa. No entanto, dado o carácter heurístico do método, o mesmo poderá ser utilizado apenas como um indicador e não como uma ferramenta rigorosa e definitiva para selecção de entradas.

Quanto à aprendizagem de parâmetros, na Secção 5.3 foram introduzidas algumas arquitecturas neuro-difusas, umas adequadas ao tratamento de sistemas linguísticos e outras relativas a modelos de Takagi-Sugeno. Em ambos dos casos foi apresentado o esquema de treino, baseado no algoritmo de retropropagação do erro. Dadas as limitações inerentes ao método, justificou-se a utilização de metodologias de treino híbridos em sistemas de Takagi-Sugeno pelo aproveitamento da natureza linear dos consequentes nestes modelos. Na mesma secção, o problema da aprendizagem em linha foi endereçado, ainda que de forma algo superficial. Referiu-se que neste caso a precisão e eficiência computacional constituem pontos fulcrais, tendo-se assim dado preferência a modelos de Takagi-Sugeno. Neste sentido, foi descrito um método de treino híbrido, baseado no treino da camada linear pelo método dos mínimos quadráticos recursivos com factor de esquecimento, sendo a camada de funções de pertença treinada pelo algoritmo de retropropagação.
Na Secção 5.4 foi abordada a questão da garantia da interpretabilidade em sistemas difusos linguísticos, tendo sido referidos aspectos de simplificação de bases de regras, resultantes da fusão de funções de pertença similares, após o que se propôs o treino restringido dos seus parâmetros.

Capítulo 6

CASOS DE ESTUDO

Após a exposição, ao longo dos capítulos precedentes, das metodologias seguidas neste trabalho de dissertação, o capítulo presente consistirá na aplicação dessas mesmas técnicas a alguns casos de estudo habitualmente utilizados em trabalhos desta natureza.

Assim, a Secção 6.1 apresenta os pressupostos em que se baseia a identificação dos modelos considerados.

Na Secção 6.2, a série caótica Mackey-Glass será modelizada com recurso ao treino de redes neuro-difusas, tanto do tipo linguístico como do tipo Takagi-Sugeno de ordem 0 e 1. Os problemas do desenvolvimento de modelos interpretáveis linguisticamente e do treino incremental serão também endereçados nesta secção.

Na Secção 6.3 analisar-se-ão alguns aspectos relacionados com a modelização da fornalha de gás de Box e Jenkins. Primeiramente, será abordada a problemática da selecção de entradas relevantes, tomando lugar, seguidamente, o mesmo esquema seguido na identificação da série Mackey-Glass.

Finalmente, a Secção 6.4 descreverá a tentativa de identificação neuro-difusa de um sistema real, nomeadamente, uma planta de branqueamento de pasta de papel. Tal como se verificará, não foram obtidos resultados totalmente satisfatórios, sendo, deste modo, analisadas as causas conducentes a tal situação.

6.1. Introdução

Os algoritmos descritos no capítulo precedente foram implementados computacionalmente na linguagem de programação C++, utilizando o compilador *Borland C++* ®, sendo a visualização gráfica dos resultados efectuada com recurso ao *Matlab* ®. Futuramente, será desenvolvida uma interface, a qual encapsulará os algoritmos desenvolvidos. A aplicação final poderá constituir uma ferramenta a utilizar no estudo de sistemas dinâmicos, sendo utilizável tanto com funções pedagógicas como com objectivos de investigação.

Em termos de plataforma computacional, as experiência foram realizadas numa máquina com processador Pentium II®, 64KB de RAM, correndo o sistema operativo Windows NT 4.0.

Após algum período experimental, chegou-se à parametrização base para os algoritmos descritos, expressa na Tabela 6.1. Naturalmente, os parâmetros apresentados são susceptíveis de sofrerem alterações pontuais, de acordo com os sistemas em causa.

MÓDULO	**PARÂMETROS**		
ARQUITECTURA NFCN	Agrupamento de Kohonen	*ginicial*	1
		dr	0.9
		s	2.8
		Número de épocas	1000
	Aprendizagem de Consequentes	*Peso mínimo*	0.1
	Número inicial de funções de pertença		7
AGRUPAMENTO SUBTRACTIVO	*Domínio de normalização*		[0;1]
	r_a		0.5
	e^{up}		0.5
	e^{down}		0.15
	r_b		1.25
APRENDIZAGEM DE PARÂMETROS	Retropropagação	*ginicial*	0.005
		m^{down}	0.1
		m^{up}	0.05
		num_{red}	4
		num_{osc}	4
		num_{inc}	2
		RMSE máximo	$\frac{U_{max} - U_{min}}{1000}$
		Percent. de amostras para treino	50%
	LSE	*Diagonal inicial da Matriz P*	1000
		Matriz B inicial	0
INTERPRETABILIDADE	Fusão de Funções	*l*	0.6
		n_{int}	100
	Aprendizagem Restringida	*a*	0.15
		Nr. de épocas entre fusões	200
APRENDIZAGEM EM LINHA	RLS	*l*	0.98

Tabela 6.1. Parametrização base dos algoritmos de aprendizagem neuro-difusa.

6.2. Série Caótica Mackey-Glass

Um dos casos de estudo mais utilizados na identificação de sistemas consiste na predição da série temporal de Mackey-Glass, gerada pela equação diferencial caótica com atraso [Mackey e Glass, 1977] a qual se descreve pela expressão (6.1):

$$\dot{x}(t) = \frac{0.2x(t-t)}{1+x^{10}(t-t)} - 0.1x(t) \tag{6.1}$$

A série definida na expressão anterior não apresenta um período definido de forma clara, sendo também bastante sensível às condições iniciais.

O problema a abordar consiste na predição de valores futuros da série para o instante $t+P$ com base em valores conhecidos até ao instante t. O conjunto de amostras a serem utilizadas na predição resulta de um mapeamento de D pontos intervalados segundo um valor Δ. Deste modo, os valores $[x(t-(D-1)\Delta), x(t-(D-2)\Delta), \cdots, x(t-\Delta), x(t)]$ são obtidos, sendo efectuada a predição do valor $x(t+P)$ com base nesses mesmos dados. Tipicamente, considera-se $D = 4$ e $\Delta = P = 6$, utilizando-se os valores $[x(t-18), x(t-12), x(t-6), x(t)]$ na previsão de $x(t+6)$.

A aplicação dos algoritmos estudados à série caótica foi conduzida com base nos dados de identificação disponibilizados pelo "*IEEE Neural Network Council, Standards Committee, Working Group on Data Modelling Benchmarks*", os quais são também utilizados na análise de diversas metodologias, entre as quais a ANFIS [Jang, 1993]. Assim, na integração assume-se $x(t)=0$, $t<0$, e um intervalo temporal de 0.1. Definiu-se ainda a condição inicial $x(0) = 1.2$ e o parâmetro $\tau = 17$. Com base na parametrização descrita, obtiveram-se valores no intervalo $t \in [0; 2000]$, tendo-se seleccionado 1000 pares entrada-saída do intervalo $t \in [118; 1117]$, apresentados na Figura 6.1.

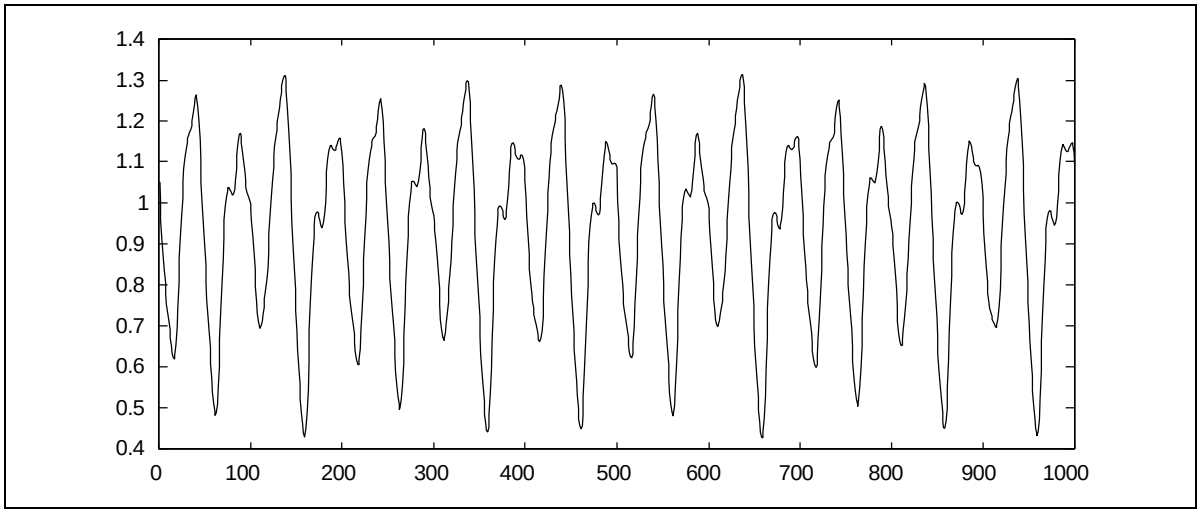


Figura 6.1. Série caótica: dados de identificação.

Com recurso às amostras obtidas, realizou-se um conjunto de experimentações relativas a aprendizagem livre fora de linha e em linha, e análise da interpretabilidade. Em qualquer das situações, os objectivos a atingir apresentam como pano de fundo o treino de uma rede neuro-difusa com quatro entradas e uma saída, de acordo com o conjunto de dados utilizado.

Com base na parametrização base descrita na Tabela 6.1, procedeu-se à previsão da série caótica Mackey-Glass, segundo os vários métodos discutidos.

6.2.1. Aprendizagem Livre Fora de Linha

Numa primeira fase experimental, foram analisadas as capacidades de aproximação das arquitecturas neuro-difusas apresentadas no capítulo precedente. Deste modo, após a aprendizagem da estrutura, os parâmetros foram optimizados livremente, sem a imposição de quaisquer restrições na aprendizagem.

Consequentes difusos: rede NFCN

Inicialmente, aplicou-se a arquitectura NFCN à identificação da série, definindo-se funções de pertença Gaussianas generalizadas, operadores de truncatura na conjunção e disjunção difusas e um conjunto de 500 amostras (as primeiras) para treino, sendo as restantes 500 utilizadas para avaliação da capacidade de generalização do modelo. A rede obtida, composta por 4 entradas e 1 saída, tem associada a cada variável 7 funções de pertença. Deste modo, a rede contém inicialmente uma camada de regras com $7^4 = 2401$ neurónios. Após a aplicação do algoritmo de agrupamento de Kohonen, o qual desloca os centros de cada função de pertença para as zonas mais densas do domínio, aplicou-se o algoritmo de selecção de consequentes e eliminação de regras, do qual resultou um número final significativamente inferior ao inicial: 9 regras. De referir que foram conduzidos esforços no sentido da obtenção do número de regras referido com base em conhecimento prévio sobre o sistema em análise. A eliminação de regras foi também acompanhada pela eliminação de algumas funções de pertença. Assim, para a variável $x(t\text{-}18)$ obtiveram-se 5 funções de pertença, para $x(t\text{-}12)$ também 5 funções e 6 para $x(t\text{-}6)$. Para $x(t)$, o número final de funções de pertenças foi de 4, tendo sido associadas 7 funções à variável de saída $x(t+6)$. Deste modo, resultou um total de 108 parâmetros a ajustar. Verificou-se que o procedimento de aprendizagem da estrutura pela utilização da rede NFCN apresenta alguma morosidade (Tabela 6.2) comparativamente ao método de agrupamento subtractivo testado posteriormente, o que se deve à aplicação do algoritmo de Kohonen durante um número suficiente de épocas, bem como ao procedimento de eliminação de regras, o qual é também algo dispendioso a nível computacional.

Após a aprendizagem da estrutura, os parâmetros dos conjuntos difusos associados a cada variável foram ajustados pela aplicação do algoritmo de retropropagação do erro. Uma vez que se verificaram oscilações significativas no treino, optou-se por reduzir a velocidade de aprendizagem inicial para 0.001. Assim, o modelo inicial apresentava um erro RMS igual a 0.1848 relativamente aos dados de treino, sendo o erro correspondente aos dados de teste de 0.1850. Inicialmente, verificou-se uma redução drástica do erro durante as primeiras épocas de treino, período após o qual a redução se processa de forma progressivamente mais lenta, até à convergência. Assim, após 4000 épocas de treino obteve-se para os dados de treino um erro de 0.0199, sendo o erro para os dados de teste de 0.0206.

Por forma a efectuar-se uma análise comparativa entre a aplicação de operadores de truncatura e algébricos, os últimos foram também experimentados. Assim, uma vez que a activação dos neurónios na camada de regras pelo produto apresenta, em geral, valores inferiores aos obtidos pela utilização do operador mínimo, tornou-se necessário baixar o limiar mínimo de manutenção de consequentes, tendo-se optado pelo valor 0.04 por possibilitar a obtenção das mesmas 9 regras, tal como na situação anterior. Quanto ao número de funções de pertença, obtiveram-se 5, 5, 5, 4 e 7, respectivamente, pela ordem indicada anteriormente, resultando 104 parâmetros a ajustar. Em termos de capacidade de previsão da série, o modelo inicial apresentava inicialmente um erro RMS de 0.1436 para os dados de treino e de 0.1441 para os dados de teste. Após 2000 épocas de treino, os mesmos valores baixaram para 0.0126 e 0.0132, respectivamente, o que se revelou bastante

melhor do que os valores obtidos pela utilização de operadores de truncatura, tal como seria de esperar dada a continuidade das funções utilizadas na implementação dos operadores algébricos.

Consequentes difusos: agrupamento subtractivo

Numa segunda fase, procedeu-se à aplicação do algoritmo de agrupamento subtractivo para a aprendizagem da estrutura. De forma a obterem-se as mesmas 9 regras, atribuiu-se ao parâmetro r_a o valor 0.5. Tal como se referiu no capítulo anterior, o número de funções de pertença associadas a cada variável será igual ao número de regras, ou seja, 9, resultando 180 parâmetros a ajustar, número esse significativamente mais elevado do que o obtido na arquitectura NFCN.

A aplicação do algoritmo de agrupamento subtractivo revelou-se bastante mais rápida que o procedimento de aprendizagem da estrutura na arquitectura NFCN. De facto, enquanto que o último demorou 28s, o primeiro registou um tempo inferior a 1s.

Assim, obteve-se para o modelo, com funções Gaussianas generalizadas e operadores algébricos, um erro inicial de 0.0689 para os dados de treino e de 0.0706 para os dados de teste. Após o treino, conduzido durante 2000 épocas, obtiveram-se os valores de 0.0070 para os dados de treino e 0.0076 para os dados de teste, mais uma vez melhores do que os correspondentes no método anterior. Os resultados alcançados podem ser justificados pelo facto de o procedimento de agrupamento ser mais efectivo, bem como do maior número de graus de liberdade, resultante do maior número de funções de pertença associadas a cada variável. A limitação principal reside no maior tempo de treino, o qual resulta do número de parâmetros a ajustar ser mais elevado.

Da utilização de operadores de truncatura resultou um erro de 0.0111 para os dados de treino e de 0.0121 para os dados de teste. Mais uma vez, os resultados obtidos revelam-se menos precisos do que os verificados com operadores algébricos.

No sentido de comparar a utilização de Gaussianas generalizadas com Gaussianas simples, as últimas foram também testadas com operadores algébricos, tendo-se obtido, após 2000 épocas de treino, os valores 0.0066 para os dados de treino e 0.0071 para os dados de teste. Os resultados obtidos, superiores aos resultantes da utilização de Gaussianas generalizadas, são explicados pelo facto de o número de épocas necessárias à convergência ser menor em Gaussianas simples, tal como se pode depreender das expressões (5.49) e (5.50). Por outro lado, as capacidades das funções Gaussianas generalizadas, sendo maiores do que as Gaussianas simples, não o são de forma arrasadora. De facto, até certo ponto os resultados obtidos através de Gaussianas generalizadas podem ser aproximados por funções simples. Finalmente, o número de parâmetros a ajustar em Gaussianas simples é exactamente metade do número necessário a Gaussianas generalizadas, no caso concreto, 90. Este aspecto constitui um argumento favorável à sua utilização, em virtude dos menores tempos de treino necessários.

Consequentes constantes

A atribuição de valores constantes aos consequentes de cada regra diminuirá, em princípio, o potencial de modelização da rede neuro-difusa utilizada, dada a sua menor flexibilidade. No entanto, o facto da última camada da rede ser agora linear possibilita o uso do estimador dos mínimos quadráticos, o qual garante a obtenção do óptimo global para os parâmetros dos consequentes, com base nas premissas de cada regra e satisfeitas as restrições indicadas na Secção 4.4.

Assim, após o treino da rede com operadores algébricos durante 1500 épocas, obtiveram-se os erros 0.0047 e 0.0050 para os dados de treino e de teste, respectivamente, os quais se revelaram mais satisfatórios que os obtidos na definição de consequentes difusos, requerendo ainda um menor número de épocas de treino. Uma outra vantagem do método reside no menor número de

parâmetros a adaptar: 153. No entanto, é importante notar que o tempo de processamento de cada época é superior ao necessário pela aplicação do método iterativo da descida do gradiente.

Da aplicação de operadores de truncatura resultou, após 1500 épocas, um erro RMS de 0.0097 para treino e de 0.0108 para teste. Mais uma vez, os resultados são superiores aos obtidos com consequentes difusos, sendo, no entanto, inferiores aos obtidos com consequentes constantes e operadores algébricos, tal como se tem vindo a verificar.

Finalmente, foram utilizadas Gaussianas simples com operadores algébricos, obtendo-se, após 1500 épocas, os valores 0.0050 e 0.0052 para treino e para teste. Ao contrário de consequentes difusos, neste caso a utilização de Gaussianas simples deteriora um pouco a capacidade de previsão do modelo. Na verdade, o facto de agora os consequentes serem optimizados permite ultrapassar a limitação das Gaussianas generalizadas em termos de ajuste de consequentes difusos. Quanto ao número de parâmetros, o seu número atinge o valor mínimo de 81.

Consequentes do tipo Takagi-Sugeno de ordem 1

Para finalizar, a série temporal foi ainda identificada com base no treino de um modelo difuso do tipo Takagi-Sugeno de ordem 1. Nesta situação esperam-se resultados superiores aos obtidos pela utilização de consequentes constantes, dada a maior riqueza dos termos no consequente, o que se verificou naturalmente. De facto, esta classe de estruturas difusas é susceptível de ser interpretada como um esquema de comutação entre modelos lineares locais, o que apresenta vantagens face à estrutura interpolativa presente nos modelos de ordem 0 e linguísticos.

Assim, com base em funções generalizadas e em operadores algébricos, a rede foi treinada durante 300 épocas, sendo obtidos os erros RMS 0.0025 e 0.0030 para os dados de treino e teste, o que constitui uma melhoria clara em relação a todos os restantes métodos testados, tanto a nível de capacidade de precisão, como em termos de número de épocas de treino. No entanto, a desvantagem principal prende-se com o facto do tempo de treino de cada época ser consideravelmente mais elevado, bem como com o aumento do número de parâmetros do modelo, o qual atinge agora o valor de 189.

A utilização de operadores de truncatura, mais uma vez, deteriora a capacidade de previsão do modelo, obtendo-se os valores 0.0038 e 0.0043 para treino e teste, respectivamente, ao fim de 300 épocas.

Quanto à utilização de funções Gaussianas simples com operadores algébricos, o erro RMS atinge o valor 0.0030 relativamente às amostras de treino e 0.0033 para os dados de teste. Tal como no caso de consequentes constantes, os resultados revelaram-se inferiores aos das Gaussianas generalizadas, em virtude da optimização dos parâmetros dos consequentes. A sua vantagem reside no facto de que o número de parâmetros baixa para 147.

Em jeito de síntese, os resultados obtidos podem ser sumariados na Tabela 6.2, onde AS denota agrupamento subtractivo e CD, CC e CO1 denotam respectivamente, consequentes difusos, consequentes constantes e consequentes de ordem 1.

Os resultados apresentados sugerem algumas conclusões. Assim, em primeiro lugar a aprendizagem da estrutura por meio do algoritmo de agrupamento subtractivo mostra-se mais adequada; a utilização de operadores algébricos revela-se favorável, relativamente a operadores de truncatura; a utilização de Gaussianas simples apresenta vantagens no caso de serem utilizados consequentes difusos; modelos com consequentes constantes possibilitam melhores resultados do que modelos com consequentes difusos; modelos do tipo Takagi-Sugeno de ordem 1 são os mais precisos, necessitando de um número significativamente menor de épocas de treino. No entanto, tal como se referiu, da optimização linear resultam tempos de processamento elevados, os quais

poderão levantar problemas em termos de aplicabilidade em tempo real. Deste modo, modelos com consequentes constantes constituem, aparentemente, um bom compromisso entre precisão e eficiência.

Método		Tipo de Gaussianas	Nr. de Parâmetros	Op. Difusos	Nr. de Épocas	Tempo p/ Época	RMSE	
							Treino	Teste
1	NFCN	Generalizadas	108	Algébricos	2000	0.11s	0.0126	0.0132
2	"	"	104	Truncatura	4000	0.17s	0.0199	0.0206
3	AS-CD	Generalizadas	180	Algébricos	2000	0.27s	0.0070	0.0076
4	"	"	"	Truncatura	"	0.24s	0.0111	0.0121
5	"	Simples	90	Algébricos	"	0.26s	0.0066	0.0071
6	AS-CC	Generalizadas	153	Algébricos	1500	0.54s	0.0047	0.0050
7	"	"	"	Truncatura	"	0.52s	0.0097	0.0108
8	"	Simples	81	Algébricos	"	0.53s	0.0050	0.0052
9	AS-CO1	Generalizadas	189	Algébricos	300	4.1s	0.0025	0.0030
10	"	"	"	Truncatura	"	4.3s	0.0038	0.0043
11	"	Simples	147	Algébricos	"	4.0s	0.0030	0.0033

Tabela 6.2. Série caótica: resultados de treino livre fora de linha.

Na Figura 6.2 apresenta-se graficamente a saída relativa a dados de treino e dados de teste para o método 3.

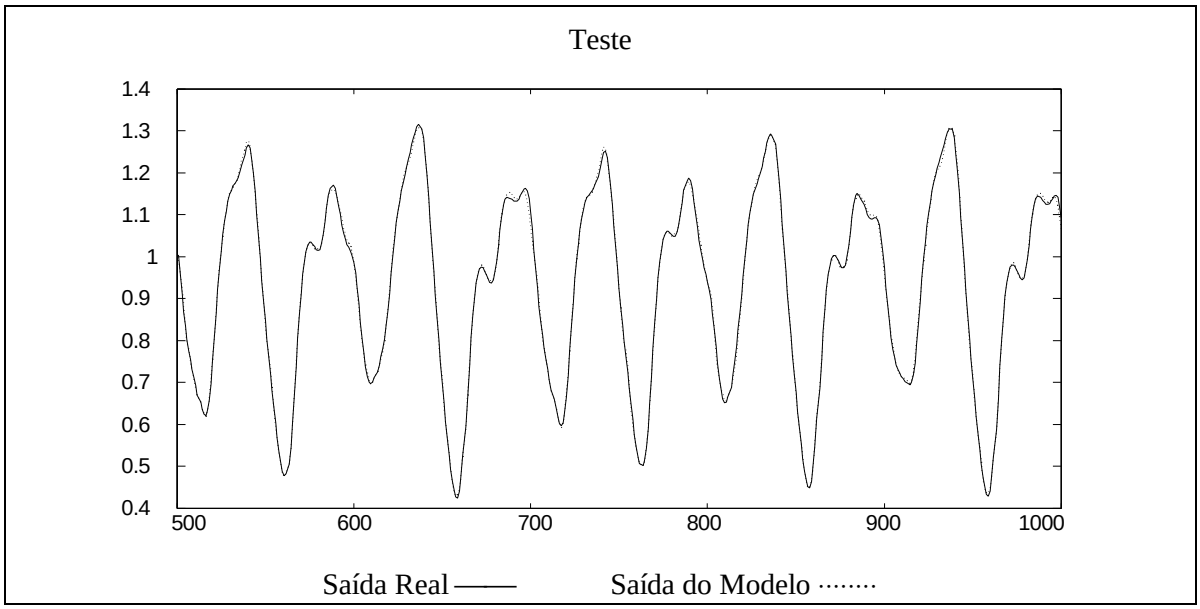


Figura 6.2. Série caótica: previsão da saída num modelo linguístico com operadores algébricos e funções Gaussianas generalizadas.

Relativamente à implementação de modelos difusos do tipo Takagi-Sugeno de ordem 1, os resultados obtidos na simulação 9 são apresentados na Figura 6.3. Na figura referida, a saída do

modelo praticamente não se distingue da saída real, o que prova a grande precisão alcançada.

Para o modelo da simulação 3 (Figura 6.2), as funções de pertença obtidas para cada variável são representadas graficamente na Figura 6.4. Tal como se pode verificar, o modelo obtido não é facilmente interpretável, uma vez que a atribuição de termos linguísticos a cada uma das funções de pertença não se efectua de forma trivial. Desta forma, o problema da construção de modelos transparentes será abordado posteriormente nesta secção.

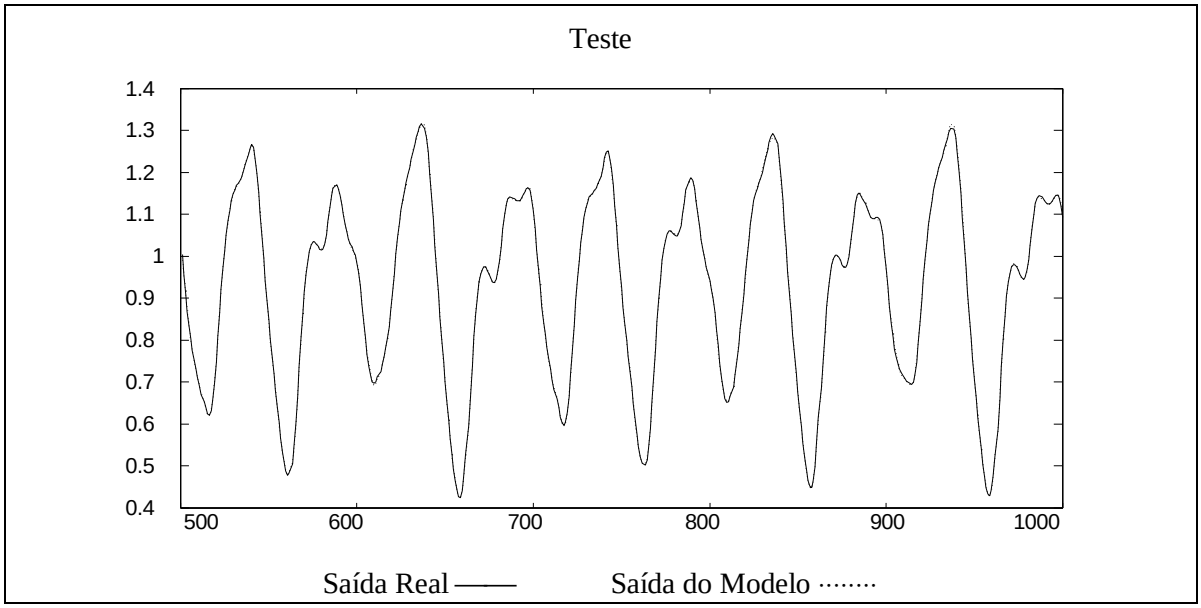


Figura 6.3. Série caótica: previsão da saída num modelo Takagi-Sugeno de ordem 1 com operadores algébricos e funções Gaussianas generalizadas.

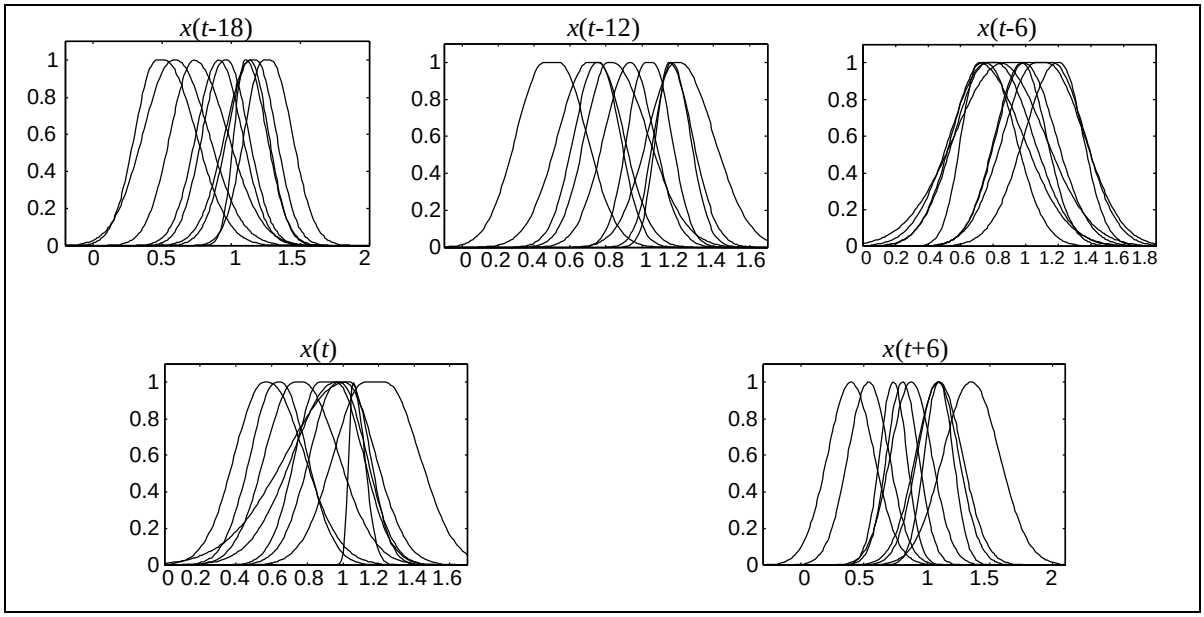


Figura 6.4. Série caótica: funções de pertença com aprendizagem livre.

Em termos de análise comparativa com outros métodos, alguns dos resultados obtidos no passado são apresentados na Tabela 6.3. O índice de erro denota uma grandeza não dimensional

resultante da divisão do erro RMS pelo desvio padrão da série [Jang, 1993], para o qual se obteve o valor 0.2143.

	Método	Número de Regras	Número de Parâmetros Livres	Índice de Erro
1	AS-CD	9	90	0.033
2	AS-CC	9	153	0.023
3	AS-CO1	9	189	0.014
4	Chiu	25	125	0.014
5	ANFIS	16	104	0.007
6	ANN com Retropropagação	----	540	0.02
7	NEFPROX (A)	129	105	0.155
8	NEFPROX (G)	26	38	0.313

Tabela 6.3. Série caótica: comparação do treino não restringido com outras metodologias.

Na tabela referida, as três primeiras entradas referem-se aos melhores resultados obtidos nas simulações realizadas para este trabalho, de acordo com a Tabela 6.2; as entradas 4 a 6 são extraídas de [Chiu, 1994]; finalmente, as entradas 7 e 8 são adaptadas de [Nauck e Kruse, 1999]. Muitos outros resultados são apresentados na literatura, embora a sua inclusão não seja efectuada, dado muitos deles serem baseados em conjuntos de dados diferentes dos utilizados. Dos resultados anteriores, podem ser retiradas algumas ilações. Assim, verifica-se que redes neuronais treinadas pelo algoritmo de retropropagação do erro requerem um número de parâmetros ajustáveis consideravelmente superior ao dos restantes métodos, baseados em sistemas difusos. Esta conclusão já havia sido derivada por Jang [Jang, 1993], tendo este autor apontado essa vantagem dos sistemas neuro-difusos, bem como o facto de se necessitar de um número inferior de épocas de treino. Pela comparação das entradas 3 e 4, ambas baseadas em sistemas difusos com consequentes de ordem 1, verifica-se que no caso AS-CO1, o facto de os parâmetros dos consequentes serem ajustados, o que não acontece em Chiu, possibilita uma redução significativa do número de regras. Embora o número de parâmetros livres seja significativamente superior, esse número pode ser fortemente reduzido pela utilização de funções Gaussianas simples, o que degrada apenas de modo ligeiro o desempenho do modelo. As duas últimas entradas referem-se à implementação de modelos interpretáveis, tendo-se, como tal, obtido modelos menos precisos. Os resultados apresentados serão comparados com os resultantes do algoritmo de interpretabilidade proposto. É ainda importante notar que só os resultados da arquitectura ANFIS se revelaram superiores, à custa de um maior número de regras condicionais difusas.

6.2.2. Aprendizagem em Linha

Tal como se referiu anteriormente, o recurso a funções de activação do tipo das Gaussianas apresenta vantagens em termos de aprendizagem incremental, em virtude da propriedade da localidade de que gozam. De forma a analisar as potencialidades das várias estruturas difusas na identificação em linha, algumas simulações foram conduzidas, as quais se passam a descrever. Em

qualquer dos casos, parte-se de um modelo inicial obtido fora de linha, segundo o esquema de aprendizagem não restringida descrito anteriormente.

Assim, a partir do modelo inicial, obtido com base na primeira metade do conjunto de dados, os seus parâmetros são ajustados em modo incremental, com recurso aos dados de teste, tal como se de uma situação de identificação em linha se tratasse.

Quanto à parametrização dos algoritmos, são utilizados operadores algébricos, dado possibilitarem a obtenção de modelos mais precisos, sendo definida uma velocidade de aprendizagem constante com o valor 0.001.

Consequentes difusos

Assim, para modelos linguísticos, considerou-se o modelo resultante da simulação 3 da Tabela 6.2, com funções Gaussianas generalizadas. Nesta situação, o conjunto de teste é utilizado para treino incremental da rede neuro-difusa, tendo-se obtido um erro segundo o critério RMS de 0.0076, o qual é aparentemente igual ao obtido sem aprendizagem no conjunto de teste. De facto, a alteração é mínima, não sendo notada devido à aproximação numérica. No entanto, o erro diminui, mais precisamente, de 0.007584 para 0.007579. De notar que o ganho em termos de precisão é reduzido em virtude do facto da velocidade de aprendizagem dever ser ela também baixa. Naturalmente que o seu aumento iria produzir melhores resultados, podendo, no entanto, originar instabilidade.

Foi também analisado o comportamento do modelo com funções Gaussianas simples, tendo-se verificado um decréscimo do erro de validação de 0.007132 (simulação 5) para 0.007121. Em termos proporcionais, verifica-se um ganho maior decorrente da utilização de funções simples, o que vai de encontro às conclusões extraídas da análise da identificação fora de linha.

De forma a efectuar-se uma análise mais precisa da eventual maior capacidade de funções simples, realizou-se uma simulação partindo de um modelo com erro de validação o mais próximo possível do valor referente a funções generalizadas, i.e., 0.007584. Assim, a partir de um erro de validação com o valor 0.007583, resultou, com aprendizagem em linha, o valor 0.007574, ligeiramente inferior ao valor obtido com funções generalizadas, tal como seria de esperar.

Consequentes constantes

O treino incremental de modelos difusos com consequentes constantes foi efectuado com base na simulação 6. Neste caso, verificou-se que o erro diminuiu de 0.004998 para 0.004992. De forma a estabelecer-se uma base de comparação com os resultados obtidos através de consequentes difusos, partiu-se de um erro RMS igual a 0.007561, tendo-se chegado ao valor 0.007379, o qual é mais satisfatório. Naturalmente, a melhoria deve-se à optimização linear no consequente.

Da aplicação de Gaussianas simples (simulação 8), resulta a diminuição do erro do valor 0.005186 para 0.005173. Tal como no caso de consequentes difusos, a utilização de funções simples possibilita um maior decréscimo em termos de erro. Definindo um modelo base com erro de validação 0.007585, resulta o erro final 0.007525, o que constitui uma melhoria mais significativa do que no caso linguístico, embora não se tenha uma redução tão drástica como com Gaussianas generalizadas.

Consequentes do tipo Takagi-Sugeno de ordem 1

Em modelos difusos com consequentes de primeira ordem, o estudo efectuado baseia-se na simulação 9. Assim, o erro RMS diminuiu de 0.002951 para 0.002907. Tendo por base um erro de validação com o valor 0.007581, a aprendizagem incremental conduz a um resultado final de 0.007495. Inesperadamente, o seu valor é superior ao obtido com consequentes constantes, o que

poderá ser justificado pela ocorrência de uma situação fortuita, derivada de peculiaridades da superfície de erro.

Aplicando-se funções de pertença Gaussianas simples (simulação 11), o erro RMS decresce de 0.003288 para 0.003164. Novamente, partindo do erro 0.007583, obtém-se o valor 0.007509. Nesta situação, tal como seria de esperar, o ganho verificado é o mais elevado, comparativamente à utilização de consequentes difusos e constantes. Em relação a funções Gaussianas generalizadas, verifica-se que estas apresentam uma melhoria superior à obtida por meio de Gaussianas simples.

Os resultados alcançados são sumariados na Tabela 6.4, com base na qual se podem sugerir algumas conclusões.

Método		**Tipo de Gaussianas**	**Tempo total**	**Resultados reais**		**Comparação**	
				RMSE inicial	**RMSE final**	**RMSE inicial**	**RMSE final**
1	AS-CD	Generalizadas	0.7s	0.007584	0.007579	0.007584	0.007579
2	“	Simples	0.7s	0.007132	0.007121	0.007583	0.007574
3	AS-CC	Generalizadas	1.2s	0.004998	0.004992	0.007561	0.007379
4	“	Simples	1.2s	0.005186	0.005173	0.007585	0.007525
5	AS-CO1	Generalizadas	5s	0.002951	0.002907	0.007581	0.007495
6	“	Simples	5s	0.003288	0.003164	0.007583	0.007509

Tabela 6.4. Série caótica: resultados de treino em linha.

Assim, verifica-se que os modelos do tipo Takagi-Sugeno de ordem 1 constituem a classe com melhorias mais significativas, resultantes do treino incremental dos dados de validação, seguindo-se os modelos de ordem 0 e finalmente os modelos linguísticos. Quanto à utilização de funções Gaussianas generalizadas, verifica-se um desempenho ligeiramente superior, à excepção das situações em que se considerem modelos com consequentes difusos. Em termos de eficiência computacional, os modelos linguísticos são, claramente, os mais eficientes, o que advém do facto de não incluírem o procedimento de optimização linear característico dos modelos Takagi-Sugeno.

Dada a sua maior complexidade, os modelos de ordem 1 são, claramente, os menos eficientes a nível computacional, além do facto do número de parâmetros a ajustar ser, em geral, significativamente maior. Assim sendo, com base na análise da precisão, do número de parâmetros a ajustar e da eficiência computacional, os modelos de Takagi-Sugeno de ordem 0 constituem a melhor solução de compromisso entre os aspectos enunciados.

Considerando agora a questão da escolha de funções de pertença, verifica-se que o ganho em precisão decorrente da utilização de funções Gaussianas generalizadas não justifica a sua utilização, uma vez que o número de parâmetros a ajustar é significativamente superior. Pelo exposto, conclui-se que, para efeitos de aprendizagem em linha, as estruturas com operadores algébricos, funções Gaussianas simples e consequentes constantes, equivalentes às redes RBF clássicas, apresentam-se como as mais adequadas.

Em relação às simulações efectuadas, é importante notar que, em termos reais, alguns dos resultados alcançados não têm grande significado, uma vez que o grau de precisão que se procurou atingir talvez não seja determinante num contexto puro e simples de identificação de sistemas. Porém, as simulações conduzidas visam, acima de tudo, possibilitar a extracção de conclusões pela comparação dos métodos utilizados.

6.2.3. Aprendizagem de Modelos Interpretáveis

De forma a que a interpretabilidade do modelo final obtido seja garantida, é fundamental que se imponham algumas restrições, relativamente ao número de regras, de funções de pertença por variável, bem como do seu grau de sobreposição. Assim, quanto ao primeiro aspecto, o número de regras obtido nas experiências anteriores, i.e., 9 regras, revela-se satisfatório em termos de transparência. Quanto ao número de funções de pertença, embora o mesmo não seja excessivamente elevado, um número inferior fomentaria a interpretabilidade. No que toca à capacidade de distinção entre funções de pertença, verifica-se pela Figura 6.4 que os resultados obtidos são insatisfatórios. Assim, procedeu-se ao treino restringido da rede, de forma a que a interpretabilidade fosse mantida durante o treino, conforme os critérios estabelecidos na Secção 5.4. Uma vez que os modelos Takagi-Sugeno de ordem 1 não são interpretáveis linguisticamente, o seu estudo não é efectuado. Em virtude do objectivo pretendido, são utilizadas funções de pertença Gaussianas generalizadas.

Modelo com consequentes difusos

No que toca à definição de modelos com consequentes difusos, começou-se por testar a obtenção de um modelo transparente com base em operadores algébricos. No entanto, os resultados obtidos revelaram-se insatisfatórios. Nomeadamente, ao fim de 200 épocas de treino o erro RMS estabilizou nos valores 0.0637 e 0.0647 para os dados de treino e de teste, respectivamente. Quanto ao número de funções de pertença, foram obtidas, para cada uma das variáveis segundo a ordem que tem vindo a ser utilizada, 5, 4, 6, 4 e 5 funções, o que origina um total de 96 parâmetros livres.

Posteriormente, realizou-se a mesma experiência com operadores de truncatura, tendo-se obtido, ao fim de 800 épocas de treino, os valores 0.0228 e 0.0239 para as amostras de treino e teste. O resultado obtido constitui desde já uma novidade comparativamente ao treino restringido, decorrente do facto da utilização de operadores de truncatura originar agora melhores resultados. A razão da alteração enunciada deriva do facto do número de parâmetros ajustados em cada iteração ser menor, em virtude das características do operador mínimo na camada de regras. Deste modo, em cada época de treino, as alterações verificadas no modelo são menores, o que implica que o procedimento de monitorização não irá "danificar" de forma significativa os resultados do treino não restringido. Assim sendo, poder-se-á afirmar que a utilização de operadores de truncatura possibilita um menor afastamento da direcção do verdadeiro gradiente. Quanto ao número de funções de pertença, foram obtidas 5, 4, 5, 4 e 5 funções para as variáveis de entrada e saída, o que origina 92 parâmetros livres.

Modelo com consequentes constantes

Tal como na situação anterior, começou por se testar um modelo baseado em operadores algébricos. Assim, após 200 épocas de treino, o erro RMS estabilizou nos valores 0.0419 e 0.0427 para treino e teste, respectivamente. Nesta situação foram obtidas 5, 4, 6, 4 e 9 funções de pertença para cada variável de entrada, o que originou 112 parâmetros ajustáveis.

Seguidamente, realizou-se a mesma experiência com operadores de truncatura, tendo-se obtido, mais uma vez, resultados mais satisfatórios, após as mesmas 200 épocas: 0.0314 para os dados de treino e 0.0327 para os dados de teste. Os valores obtidos confirmam a hipótese levantada anteriormente, relativamente às vantagens de operadores de truncatura no treino restringido. Em termos de funções de pertença de entrada, obtiveram-se 5, 4, 6, 4 e 5, respectivamente, totalizando 96 parâmetros.

Um aspecto curioso da implementação de modelos com consequentes constantes prende-se com o facto de o seu desempenho em termos de precisão ter sido inferior ao obtido através de consequentes difusos. Na verdade, o resultado obtido contraria as expectativas, em função das conclusões retiradas no ponto anterior do trabalho. Para mais, o facto de, em modelos linguísticos, os parâmetros dos consequentes serem restringidos, sugeria, à partida, que a utilização de consequentes constantes optimizados linearmente fosse ocasionar uma melhoria de desempenho superior à obtida no treino não restringido. No entanto, poder-se-á colocar a hipótese de que o treino restringido possibilite que se extraiam as vantagens potenciais dos consequentes difusos, procedentes da sua maior flexibilidade. Até ao momento presente não foram encontradas justificações relativamente seguras para o resultado obtido, o qual se manifestou de forma sistemática em outros casos de estudo analisados.

Sumariando, os resultados obtidos são sintetizados na Tabela 6.5.

Método		**Tipo de Gaussianas**	**Número de Parâmetros**	**Operadores Difusos**	**Número de Épocas**	**RMSE**	
						Treino	**Teste**
1	AS-CD	Generalizadas	96	Algébricos	200	0.0637	0.0647
2	“	“	92	Truncatura	800	0.0228	0.0239
3	AS-CC	“	112	Algébricos	200	0.0419	0.0427
4	“	“	96	Truncatura	200	0.0314	0.0327

Tabela 6.5. Série caótica: resultados de treino fora de linha restringido.

Os resultados apresentados sugerem algumas conclusões. Assim, em primeiro lugar, a utilização de operadores de truncatura mostra-se vantajosa, pelas razões expostas. Quanto à questão da utilização de modelos com consequentes constantes, a sua utilização não apresenta vantagens, o que constitui um resultado de certo modo surpreendente, tal como se referiu.

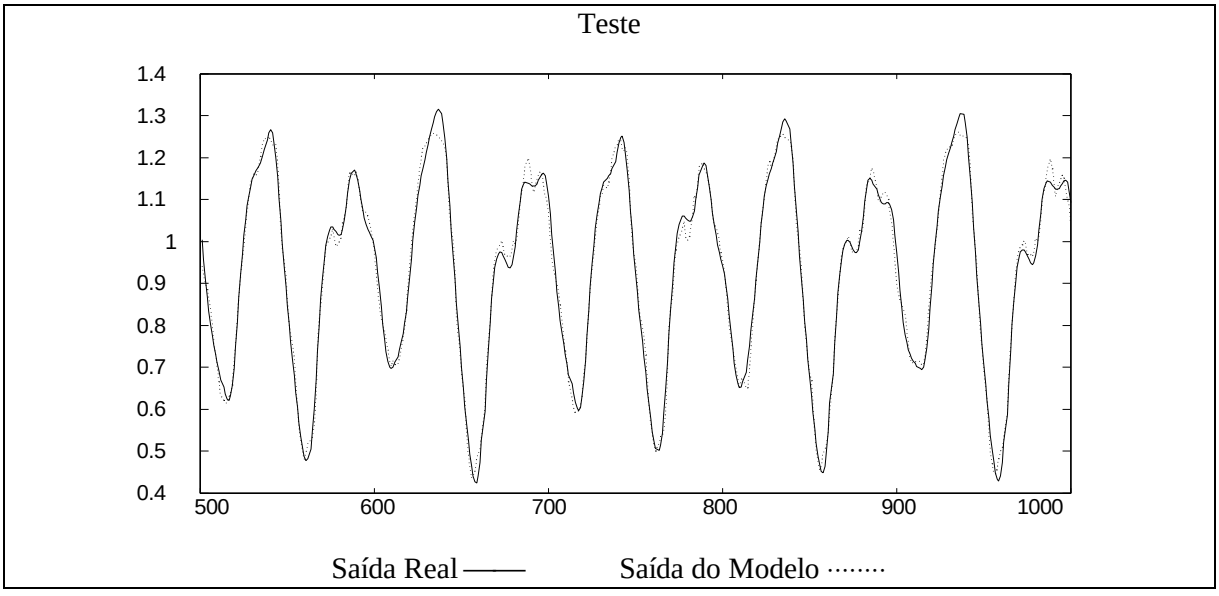


Figura 6.5. Série caótica: previsão da saída num modelo linguístico interpretável.

Da Tabela 6.5, verifica-se que a simulação 2 - consequentes difusos com operadores de

truncatura - possibilitou o melhor desempenho em termos de precisão, o qual se revelou aceitável, de acordo com a Figura 6.5.

Em termos de funções de pertença, os resultados obtidos são apresentados na Figura 6.6. Tal como se pode verificar, a atribuição de termos linguísticos a cada uma das funções é efectuada de forma simples. Na mesma figura, as etiquetas *MP*, *P*, *M*, *G* e *MG* denotam, respectivamente, os termos linguísticos, "muito pequeno", "pequeno", "médio", "grande" e "muito grande". Assim, a dinâmica fundamental da série caótica é interpretada linguisticamente segundo a Tabela 6.6.

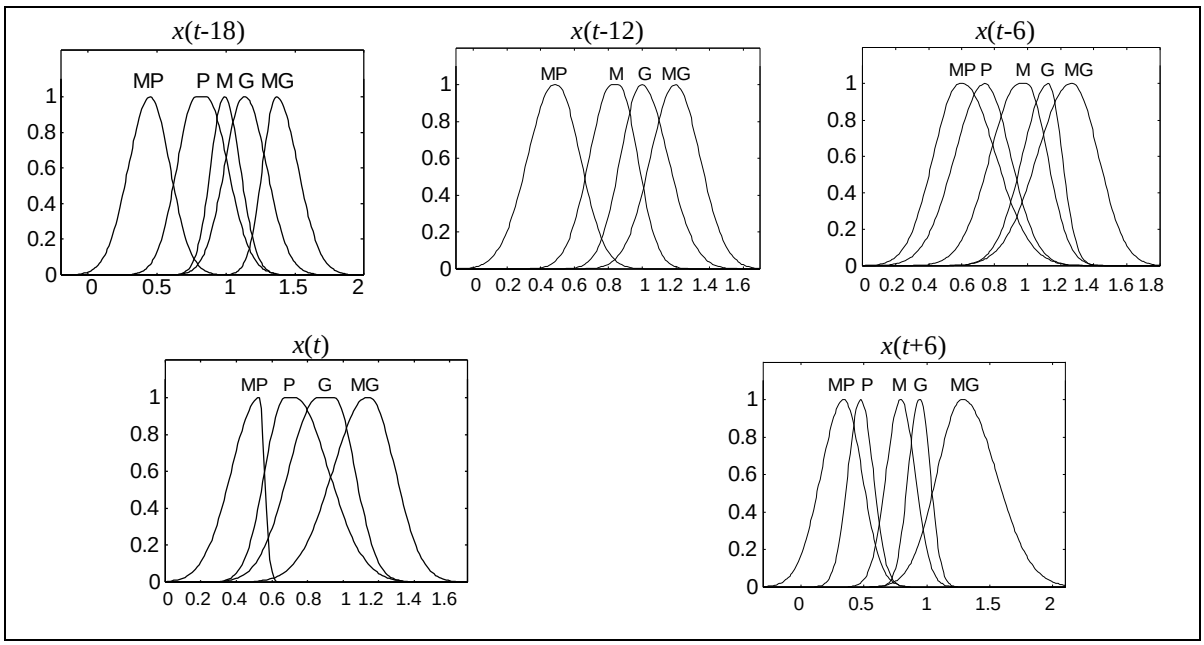


Figura 6.6. Série caótica: funções de pertença com aprendizagem restringida em modelos linguísticos.

Regra	$x(t\text{-}18)$	$x(t\text{-}12)$	$x(t\text{-}6)$	$x(t)$		$x(t+6)$
1	M	MG	G	MG		G
2	G	MG	M	P		P
3	P	M	M	MG		MG
4	M	M	MP	MG		G
5	P	G	P	MP	Þ	M
6	P	MG	MG	G		M
7	P	MP	P	G		G
8	MP	MP	M	G		G
9	MG	MG	MG	G		MP

Tabela 6.6. Série caótica: descrição linguística da dinâmica.

Comparando com a estrutura NEXPROX de Nauck [Nauck e Kruse, 1999] (Tabela 6.7), verifica-se que os resultados obtidos são claramente superiores.

	Método	Número de Regras	Número de Parâmetros Livres	Índice de Erro
1	AS-CD	9	92	0.112
2	AS-CC	9	81	0.152
7	NEFPROX (A)	129	105	0.155
8	NEFPROX (G)	26	38	0.313

Tabela 6.7. Série caótica: comparação do treino restringido com outras metodologias.

6.3. Fornalha de Gás Box-Jenkins

Tal como na série caótica Mackey-Glass, o conjunto de dados da fornalha de gás de Box e Jenkins [Box e Jenkins, 1970] constitui um caso de estudo clássico no contexto da identificação de sistemas.

Os dados utilizados consistem num conjunto de 296 amostras de pares entrada-saída do processo. Tal como no caso de estudo anterior, utiliza-se o mesmo conjunto de dados em que outros autores se basearam. Assim, a entrada é constituída pela taxa de fluxo de gás de alimentação da fornalha, a qual originará à saída uma determinada concentração de dióxido de carbono no gás de escape. Deste modo, o objectivo do problema reside na previsão da concentração de dióxido de carbono à saída, $y(t)$, com base nos valores passados da mesma, bem como da entrada $u(t)$, $\{y(t\text{-}1)$, $y(t\text{-}2)$, $y(t\text{-}3)$, $y(t\text{-}4)$, $u(t\text{-}1)$, $u(t\text{-}2)$, $u(t\text{-}3)$, $u(t\text{-}4)$, $u(t\text{-}5)$, $u(t\text{-}6)\}$. Como consequência da regressão indicada, o número final de amostras utilizadas é de 290. De forma a que se realize um conjunto de experiências idênticas às conduzidas por diversos autores, o treino do modelo é efectuado com recurso a todas as amostras disponíveis. As simulações apresentadas nos pontos seguintes utilizam por base os parâmetros indicados na Tabela 6.1, à excepção de alguns casos pontuais devidamente assinalados

6.3.1. Selecção de Entradas Relevantes

Em virtude do elevado número de entradas que constariam da rede neuro-difusa (10 entradas), parece natural que, numa primeira fase, se tentem detectar e eliminar variáveis redundantes. Assim sendo, o conjunto de dados foi dividido em dois grupos, A e B, o primeiro composto pelos 145 pontos iniciais e o segundo pelos restantes 145 pontos, tendo sido utilizado o critério da regularidade (5.35) na selecção de entradas relevantes.

De forma a minimizar eventuais problemas resultantes de sobreajustamento, foram utilizadas funções Gaussianas simples, em virtude do número de parâmetros livres daí resultante ser inferior. Nesta situação, a construção de modelos com base em 145 pontos, 10 variáveis de entrada e 2 parâmetros por função de pertença origina, efectuados os cálculos, um máximo de 7 regras.

Assim, para os dados do grupo A, atribuiu-se ao parâmetro r_a do algoritmo de agrupamento subtractivo o valor 0.65, donde resultaram 6 regras. Seguidamente, o modelo obtido foi optimizado durante 50 épocas, resultando um erro RMS de 0.489 para os dados de treino e 1.124 para os dados de teste. A razão para o reduzido número de épocas prende-se com a circunstância de a dinâmica da fornalha de gás se alterar, aproximadamente nas últimas 40 amostras. Deste modo, na construção

do modelo A, no caso do treino se prolongar, verificar-se-ia o decréscimo do critério de erro em relação aos dados de treino, o que não sucederia nos dados de teste, originando um sobreajustamento ainda mais elevado do que o verificado.

Analogamente, o conjunto de dados B, correspondente à segunda metade do conjunto inicial de amostras, foi utilizado na implementação de um modelo B. Aqui, atribuiu-se o valor 0.6 ao parâmetro r_a, tendo sido obtidas 6 regras. Novamente, o modelo inicial foi ajustado durante 50 épocas, originando um erro RMS com o valor 0.682 para treino e 1.431 para teste.

Após o desenvolvimento dos modelos A e B, aplicou-se o algoritmo de selecção de entradas de Chiu (Secção 5.2.3) com base no critério da regularidade. Tal como se referiu anteriormente, o critério referido possibilita soluções aceitáveis em termos de compromisso entre precisão e insensibilidade ao conjunto de dados utilizado. Assim, na Tabela 6.8 apresentam-se os resultados obtidos pela aplicação do método.

Todas	$u(t\text{-}6)$	$u(t\text{-}2)$	$u(t\text{-}1)$	$y(t\text{-}4)$	$u(t\text{-}3)$	$y(t\text{-}3)$	$u(t\text{-}5)$	$\mathbf{y(t\text{-}2)}$	$\mathbf{u(t\text{-}4)}$	$\mathbf{y(t\text{-}1)}$
1.286	1.246	1.217	1.194	1.183	1.182	1.181	1.178	1.239	1.437	2.439

Tabela 6.8. Fornalha de gás: remoção de entradas redundantes.

Na tabela anterior, cada coluna apresenta a variável removida em cada iteração, bem como o valor obtido para o critério da regularidade. Exemplificando, inicialmente, com todas as variáveis de entrada incluídas no modelo, o erro apresenta o valor de 1.286, sendo reduzido para 1.246 após a remoção da variável $u(t\text{-}6)$. Seguidamente, o modelo, agora com menos uma variável de entrada, é truncado pela eliminação de $u(t\text{-}2)$, de onde resulta o erro 1.217. Assim, verificou-se que o critério da regularidade atinge o valor mínimo de 1.178 após a remoção da variável $u(t\text{-}5)$. Por conseguinte, concluiu-se numa primeira fase que as variáveis $y(t\text{-}1)$, $u(k\text{-}4)$ e $y(k\text{-}2)$ constituem o subconjunto mais relevante, pela ordem apresentada. No sentido de validar os resultados obtidos foram realizadas outras experiências, variando-se o tipo de operadores difusos utilizados, o número de grupos e o número de épocas de treino. Nas simulações efectuadas constatou-se que o algoritmo de Chiu identifica, sistematicamente, as variáveis $y(t\text{-}1)$ e $u(t\text{-}4)$ como as mais importantes. No entanto, no que respeita às restantes variáveis, a sua ordem de importância variou com as experiências, não tendo sido obtidos resultados conclusivos, o que vai de encontro aos estudos de diversos autores. De facto, de acordo com Chiu [Chiu, 1996], a grande maioria dos métodos de selecção de entradas conclui que $y(t\text{-}1)$ e $u(t\text{-}4)$ constituem as duas variáveis mais relevantes para a previsão de $y(t)$. Quanto à importância relativa das restantes variáveis, os diversos métodos apresentam resultados díspares. Poder-se-á, então, concluir que as duas variáveis referidas, são de facto, as mais importantes, havendo posteriormente um conjunto de variáveis cuja importância relativa não é facilmente determinada. Deste modo, no estudo seguinte os modelos implementados basearam-se nas duas variáveis indicadas, $y(t\text{-}1)$ e $u(t\text{-}4)$.

6.3.2. Aprendizagem Livre Fora de Linha

Concluiu-se na Secção 6.2.1 que o algoritmo de agrupamento subtractivo constitui um mecanismo de aprendizagem da estrutura simultaneamente mais eficiente e preciso do que o procedimento utilizado na arquitectura NFCN. Assim, as experiências realizadas basear-se-ão no método referido. Relativamente ao número de amostras utilizadas, os modelos implementados ao longo deste capítulo recorrem ao conjunto total de amostras disponíveis, de forma a que se possa

estabelecer uma análise comparativa com os resultados obtidos por outros autores. Nas experiências realizadas, definiram-se modelos difusos com 3 regras, as quais resultaram da atribuição do valor 0.5 ao parâmetro r_a.

Consequentes difusos

Na implementação de modelos com consequentes difusos, com o número de regras supracitado, i.e., 3 regras, o número total de parâmetros livres pressupondo funções Gaussianas generalizadas é de 36.

Do treino da rede neuro-difusa inicial com operadores algébricos, o qual foi efectuado durante 4000 épocas, resultou um erro RMS com o valor 0.383. Utilizando-se operadores de truncatura, ao fim de 5000 épocas de treino o erro atingiu o valor 0.364. Ao contrário do que seria de esperar, os resultados obtidos foram mais satisfatórios do que os resultantes do uso de operadores algébricos. Não tendo sido encontrada qualquer justificação inequívoca para o sucedido, admite-se que tal situação tenha resultado de particularidades da superfície de erro. Outra hipótese para o ocorrido relaciona-se com o facto de, com apenas duas entradas, a diferença entre as superfícies de saída obtidas com operadores algébricos e de truncatura não ser tão distinta como em situações onde o número de entradas seja maior.

Tal como no caso da série caótica, efectuou-se uma análise experimental comparativa entre os resultados obtidos através de funções de pertença Gaussianas generalizadas e simples. Assim, do treino de uma rede neuro-difusa com base nas últimas funções e em operadores algébricos, estrutura essa com 18 parâmetros livres, resultou um erro com o valor 0.381 após 5000 épocas de treino. Tal como seria de esperar, os resultados obtidos revelaram um ganho em termos de desempenho, relativamente às funções generalizadas. Com operadores de truncatura, o erro obtido ao fim de 2000 épocas foi 0.396, tendo o modelo acusado um decréscimo a nível de capacidade de previsão, o que está conforme o esperado. O facto de nesta situação a utilização de operadores de truncatura ter sido prejudicial reforça a tese de que o resultado obtido com esse tipo de operadores e funções generalizadas se tratou de um acontecimento fortuito.

Consequentes constantes

Na implementação de um modelo difuso com consequentes constantes, estudou-se, primeiramente, o comportamento do modelo obtido com funções generalizadas e operadores algébricos. Assim, após o treino da rede durante 1000 épocas, obteve-se o erro 0.367, o qual se revelou mais satisfatório do que o verificado com consequentes difusos, necessitando ainda de um menor número de épocas de treino e contendo um número inferior de parâmetros livres: 27. Através do uso de operadores de truncatura resultou, após 900 épocas, o erro 0.368, ligeiramente superior ao verificado com operadores algébricos.

Utilizando-se funções Gaussianas simples, o número de parâmetros livres baixou para 15. Do treino de um modelo deste tipo com operadores algébricos resultou, ao fim de 800 épocas de treino, o valor 0.382 para o erro RMS. Nesta situação, a utilização de funções simples originou alguma degradação do desempenho do modelo, ao contrário do verificado com consequentes difusos, facto este que já havia sucedido na modelização da série caótica. No entanto, o erro obtido parece demasiado elevado, em virtude de ser superior ao verificado com consequentes difusos. Tal facto levou a que fosse realizada uma experiência com operadores de truncatura. Neste caso, obteve-se o erro 0.374 ao fim de 1000 épocas, o qual foi inferior ao obtido com operadores algébricos, sendo também inferior ao obtido na simulação equivalente em modelos difusos. Deste modo, conclui-se que o erro alcançado pela utilização de operadores algébricos e funções simples é superior ao alcançável, o que poderá resultar, mais uma vez, de características particulares da

superfície de erro.

Consequentes do tipo Takagi-Sugeno de ordem 1

Para finalizar, a fornalha de gás de Box e Jenkins foi ainda identificada com recurso a modelos do tipo Takagi-Sugeno de ordem 1. Assim, fazendo uso de funções generalizadas e operadores algébricos, a rede foi treinada durante 700 épocas, tendo sido obtido um erro RMS com o valor 0.348. O resultado verificado constitui uma melhoria clara relativamente aos restantes métodos testados, melhoria essa que era esperada em virtude da natureza dos modelos de ordem 1, bem como dos resultados verificados na identificação da série Mackey-Glass. As duas desvantagens essenciais do método prendem-se com o seu maior custo computacional, assim como com o aumento do número de parâmetros livres, o qual sobe para 33. De notar, no entanto, que este número é inferior ao verificado em modelos linguísticos (36 parâmetros), em virtude do reduzido número de entradas e regras do modelo. Relativamente à utilização de operadores de truncatura, verifica-se, mais uma vez, alguma degradação em termos de desempenho, obtendo-se o erro 0.369 ao fim de 1000 épocas.

Quanto à utilização de funções Gaussianas simples com operadores algébricos (modelo com 21 parâmetros a ajustar), o erro RMS atinge agora, ao fim de 1000 épocas, o valor 0.354, o que constitui uma ligeira perda relativamente a Gaussianas generalizadas, tal como seria de esperar. Quanto à utilização de operadores de truncatura, o erro obtido exibiu o valor 0.356 ao fim de 400 épocas. Nesta situação, verificou-se a degradação esperada, em comparação com operadores algébricos. No entanto, comparando os resultados obtidos com os verificados com operadores de truncatura e funções generalizadas verificou-se uma melhoria, a qual não estava prevista.

Sumariando, os resultados obtidos são resumidos na Tabela 6.9.

Método		Tipo de Gaussianas	Número de Parâmetros	Operadores Difusos	Tempo p/ Época	Número de Épocas	RMSE
1	AS-CD	Generalizadas	36	Algébricos	0.048s	4000	0.384
2	“	“	“	Truncatura	0.045s	5000	0.364
3	“	Simples	18	Algébricos	0.048s	1000	0.381
4	“	“	“	Truncatura	0.045s	2000	0.396
5	AS-CC	Generalizadas	27	Algébricos	0.095s	1000	0.367
6	“	“	“	Truncatura	0.095s	900	0.368
7	“	Simples	15	Algébricos	0.092s	800	0.382
8	“	“	“	Truncatura	0.092s	1000	0.374
9	AS-CO1	Generalizadas	33	Algébricos	0.20s	700	0.348
10	“	“	“	Truncatura	0.19s	1000	0.369
11	“	Simples	21	Algébricos	0.19s	1000	0.355
12	“	“	“	Truncatura	0.19s	400	0.356

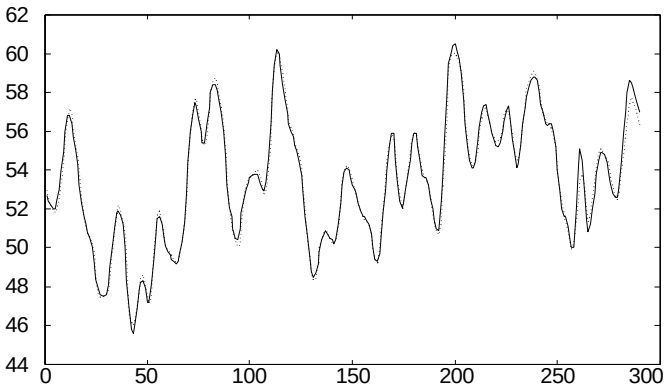
Tabela 6.9. Fornalha de gás: resultados de treino livre fora de linha.

Os valores apresentados sugerem ilações idênticas às retiradas na análise da série caótica. Porém, ao contrário do caso de estudo anterior, as conclusões obtidas não se verificaram de

maneira sistemática, tal como foi descrito nos parágrafos precedentes. Assim, a utilização de operadores algébricos revela-se, em geral, benéfica; o uso de Gaussianas simples apresenta vantagens no caso de serem utilizados consequentes difusos; modelos com consequentes constantes possibilitam melhores resultados do que modelos com consequentes difusos; modelos do tipo Takagi-Sugeno de ordem 1 são os mais precisos, necessitando de um número significativamente menor de épocas de treino.

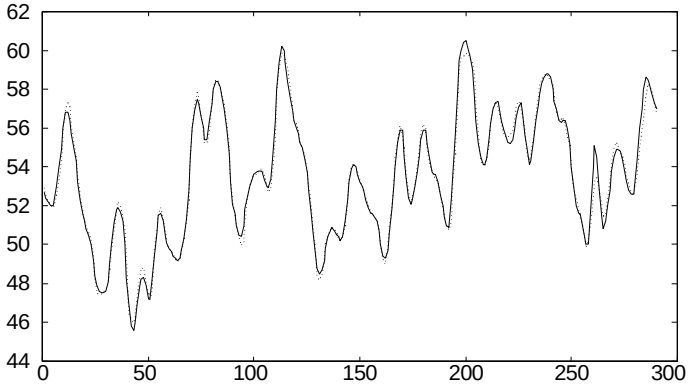
Por forma a esclarecer as dúvidas suscitadas durante as experiências efectuadas, realizou-se um conjunto de simulações idênticas às descritas, definindo agora 3 entradas para o modelo. Nesta situação, os resultados obtidos foram bastante mais sistemáticos, o que sugere que as conclusões obtidas se verificam em modelos com um maior número de entradas, situação em que as vantagens decorrentes da utilização de operadores algébricos são mais significativas. De facto, a utilização de operadores algébricos, nomeadamente o operador produto, apenas com duas entradas, não é muito distinta da utilização do operador mínimo, o que, em certas situações, poderá inclusivamente originar superfícies de erro mais complexas. Ao invés, em modelos com várias entradas, a superfície de saída em operadores algébricos é, em princípio, mais suave do que em operadores de truncatura, o que conduz às conclusões retiradas relativamente às vantagens dos primeiros.

Na Figura 6.7 apresenta-se graficamente a saída relativa à simulação 1.



Saída Real ——— Saída do Modelo ·····

Figura 6.7. Fornalha de gás: modelização linguística com operadores algébricos e funções Gaussianas generalizadas.



Saída Real ——— Saída do Modelo ·····

Figura 6.8. Fornalha de gás: modelização Takagi-Sugeno de ordem 1 com operadores algébricos e funções Gaussianas generalizadas.

Relativamente à implementação de modelos difusos do tipo Takagi-Sugeno de ordem 1, os resultados obtidos na simulação 9 são apresentados na Figura 6.8.

Para o modelo da simulação 1 (Figura 6.7), as funções de pertença obtidas para cada variável são representadas graficamente na Figura 6.9.

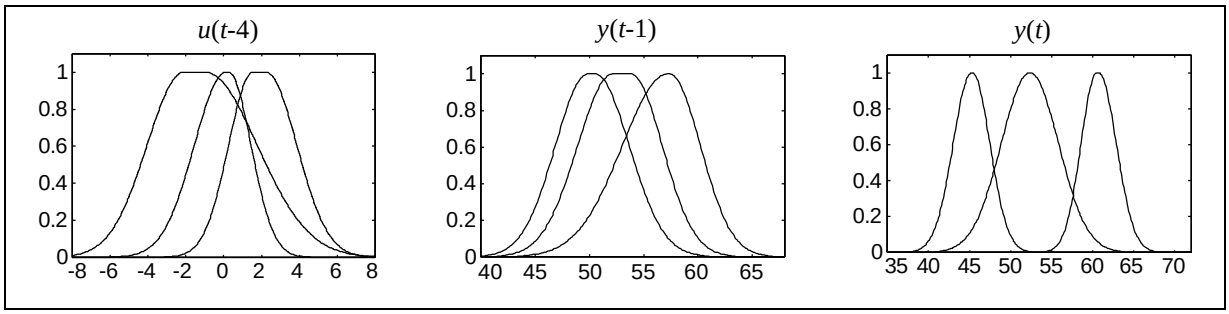


Figura 6.9. Fornalha de gás: funções de pertença com aprendizagem livre.

Tal como se pode verificar facilmente, o modelo obtido, não sendo especialmente complexo, levanta algumas dificuldades relativamente à etiquetagem de funções de pertença, nomeadamente para a variável $u(t\text{-}4)$. Desta forma, o problema da construção de modelos transparentes será abordado posteriormente.

Na Tabela 6.10 são apresentados alguns resultados de comparação com outras metodologias.

	Método	Variáveis de Entrada	Número de Regras	RMSE
1	AS-CD	$y(t\text{-}1)$, $u(t\text{-}4)$	3	0.364
2	AS-CC	"	"	0.367
3	AS-CO1	"	"	0.348
4	Tong	"	19	0.685
5	Pedrycz	"	81	0.5657
6	Xu e Yong	"	25	0.5727
7	Sugeno e Tanaka	$y(t\text{-}1)$, $y(t\text{-}2)$, $y(t\text{-}3)$, $u(t\text{-}1)$, $u(t\text{-}2)$, $u(t\text{-}3)$	2	0.2608
8	Sugeno e Yasukawa	$y(t\text{-}1)$, $u(t\text{-}4)$, $u(t\text{-}3)$	6	0.4359
9	Chiu	$y(t\text{-}1)$, $u(t\text{-}3)$	3	0.3821
10	Chiu	$y(t\text{-}1)$, $u(t\text{-}3)$, $y(t\text{-}3)$	3	0.2683

Tabela 6.10. Fornalha de gás: comparação do treino livre com outras metodologias.

As três primeiras entradas da tabela referem-se aos melhores resultados alcançados neste trabalho, com base na Tabela 6.9, sendo as restantes adaptadas de [Chiu, 1996]. Assim, para modelos com duas entradas, verifica-se que os resultados obtidos no presente trabalho de dissertação são os mais positivos. Verifica-se, no entanto, que o método 7, de Sugeno e Tanaka, apresenta claramente os melhores resultados da listagem, recorrendo, no entanto, a um maior número de entradas. Particularmente interessante é o resultado de Chiu, na linha 10 da tabela, o qual se aproxima significativamente dos valores exibidos pelo modelo de Sugeno e Tanaka, com recurso a apenas três variáveis de entrada. De notar ainda, que não é utilizada a variável $u(t\text{-}4)$.

6.3.3. Aprendizagem em Linha

Tal como na identificação da série Mackey-Glass, serão conduzidas, nos pontos seguintes, algumas simulações no sentido de aferir sobre as potencialidades das estruturas definidas, na problemática da aprendizagem em linha. Novamente, parte-se de um modelo inicial implementado fora de linha, com recurso à primeira metade do conjunto de dados, sendo posteriormente realizada a aprendizagem incremental com os dados não utilizados no modelo inicial, i.e., a segunda metade do conjunto. Em termos de parametrização, são utilizados operadores algébricos e uma velocidade de aprendizagem constante com o valor 0.005.

Consequentes difusos

No estudo da aprendizagem incremental em modelos linguísticos procedeu-se ao treino da rede neuro-difusa durante 200 épocas, obtendo-se um erro RMS com o valor 0.317 para os dados de treino e 0.615 para os dados de teste. A razão fundamental para o reduzido número de épocas prende-se com questões relacionadas com o sobretreino da rede. De facto, uma vez que os dados da fornalha apresentam uma variação da dinâmica nas últimas amostras de teste, o ajuste excessivo de parâmetros conduz a uma redução do erro RMS relativamente aos dados de treino, a qual não se manifesta nos dados de teste, onde o erro cresce.

Fazendo uso do conjunto de teste para aprendizagem incremental, o erro associado decresce para 0.546, o que é significativo e, simultaneamente esperado, uma vez que o modelo se adapta agora à dinâmica distinta da segunda metade do conjunto de dados.

Relativamente a funções Gaussianas simples, ao fim de 149 iterações o erro RMS situou-se em 0.327 e 0.639, respectivamente para os dados de treino e teste. Após a aprendizagem incremental, o erro relativo ao conjunto de teste diminuiu para 0.574, verificando-se um decréscimo da mesma ordem de grandeza do ocorrido em modelos com funções generalizadas.

De forma a proceder-se a um estudo mais detalhado relativamente à precisão resultante da escolha das funções de pertença, seria desejável realizar um conjunto de experiências comparativas, partindo de um erro base comum, tal como o efectuado na série caótica. No entanto, em virtude das particularidades do problema em questão, as quais originam alguns resultados singulares, tais como os verificados na aprendizagem por lotes, verificou-se que tal análise seria inconclusiva. De facto, em certas metodologias existe um maior sobreajustamento ao conjunto de treino do que noutras, o que desde logo levanta problemas na construção de modelos susceptíveis de serem utilizados como base de comparação.

Consequentes constantes

No treino incremental de modelos difusos com consequentes constantes o erro atingiu, após 120 épocas de treino, os valores 0.288 e 0.656 para os dados de treino e teste. Nesta situação, o sobreajustamento ocorrido é claramente superior ao verificado nas simulações precedentes. Tal facto resulta das melhores capacidades de optimização associadas às estruturas do tipo Takagi-Sugeno. Naturalmente, foram experimentadas diferentes parametrizações, tendo-se obtido sistematicamente níveis elevados de sobreajustamento. Após a realização da aprendizagem em linha, o erro RMS relativo à segunda metade do conjunto de dados diminuiu para 0.593. Em termos teóricos, seria de esperar uma melhoria mais significativa do que a verificada com consequentes difusos, o que, contudo, não sucedeu. A justificação mais plausível sugere que tal seja uma consequência do maior sobreajustamento aos dados de treino, o que leva a que o modelo inicial em que se baseia a aprendizagem incremental necessite de maiores alterações no sentido de captar a

dinâmica da segunda metade do conjunto de dados.

Da aplicação de Gaussianas simples resulta, após 200 épocas, um modelo com erro 0.291 para os dados de treino e 0.627 para os dados de teste. Nesta situação o nível de sobreajustamento é menor, pelo que seria de esperar uma melhoria mais significativa, decorrente do treino incremental. De facto, o erro diminuiu para 0.577.

Consequentes do tipo Takagi-Sugeno de ordem 1

Em virtude do ocorrido no treino de modelos com consequentes constantes em termos de sobreajustamento, esperam-se também níveis elevados, em consequência das propriedades de aprendizagem dos modelos de ordem 1. De facto, fazendo uso de funções generalizadas, a rede neuro-difusa foi treinada durante 100 épocas, de onde resultou um erro RMS com o valor 0.276 para os dados de treino e 0.626 para os dados de teste. Agora, dada a obtenção de melhores resultados num menor número de iterações em modelos de ordem 1, espera-se uma incremento significativo do desempenho do modelo, o que se veio a verificar pela diminuição do erro para 0.451.

Aplicando-se funções de pertença Gaussianas simples, o erro RMS obtido ao fim de 100 épocas de treino é de 0.276 para os dados de treino e 0.626 para os dados de teste, valores esses exactamente iguais aos obtidos por meio de funções generalizadas. Naturalmente, esperavam-se valores um pouco mais elevados. Quanto à aprendizagem incremental, o erro decresce para 0.452, o que constitui uma degradação mínima em relação à situação precedente.

Os resultados alcançados são sumariados na Tabela 6.11. Com base na tabela referida, a única conclusão que parece clara prende-se com as vantagens da utilização de estruturas de Takagi-Sugeno de ordem 1. Quanto à comparação entre modelos difusos de ordem 0 e linguísticos, os resultados apresentados, não sendo conclusivos, sugerem que os últimos são preferíveis, o que contraria as expectativas. Relativamente ao tipo de funções utilizadas, verifica-se que os erros exibem valores da mesma ordem de grandeza, pelo que as funções simples serão preferíveis em virtude da sua maior simplicidade. Em termos de eficiência computacional, o tempo de computação em qualquer dos modelos foi inferior a 1s, em virtude da reduzida dimensão do modelo bem como do conjunto de dados. No entanto, os modelos linguísticos são os mais eficientes, não sendo, no entanto, essa mais valia tão clara como no caso da série caótica, em virtude dos aspectos enunciados relativamente ao número de amostras de dados e entradas e parâmetros do modelo.

Método		**Tipo de Gaussianas**	**RMSE**		
			Treino	**Teste**	**Teste final**
1	AS-CD	Generalizadas	0.317	0.615	0.546
2	“	Simples	0.327	0.639	0.574
3	AS-CC	Generalizadas	0.288	0.656	0.593
4	“	Simples	0.290	0.627	0.577
5	AS-CO1	Generalizadas	0.276	0.626	0.451
6	AS-CO1	Simples	0.276	0.626	0.452

Tabela 6.11. Fornalha de gás: resultados de treino incremental.

6.3.4. Aprendizagem de Modelos Interpretáveis

No sentido da garantia da interpretabilidade linguística do modelo obtido, constatou-se que tanto o número de regras como o número de funções de pertença por variável obtidas na aprendizagem não restringida se afigura adequado. No entanto, verificou-se que o grau de sobreposição entre algumas das funções de pertença é excessivamente elevado (Figura 6.9), pelo que será desejável proceder ao treino restringido da rede. Tal como na série Mackey-Glass, não são implementados modelos do tipo Takagi-Sugeno de ordem 1, uma vez que os mesmos não são passíveis de interpretação linguística. Pela mesma razão, definem-se funções de pertença Gaussianas generalizadas, uma vez que fomentam a interpretabilidade final do modelo.

Modelo com consequentes difusos

Na definição de modelos com consequentes difusos, começou por se fazer uso de operadores algébricos, tendo-se obtido, ao fim de 4000 épocas de treino um erro RMS com o valor 0.392. Quanto ao número de funções de pertença, no final do treino a variável $u(t\text{-}4)$ apresentava 2, a variável $y(t\text{-}1)$ continha também 2 e variável de saída $y(t)$ era representada por 3 funções de pertença, o que originou um total de 28 parâmetros ajustáveis.

Seguidamente, foi efectuada a mesma experiência, agora com operadores de truncatura, tendo-se obtido, ao fim de 3000 épocas de treino, o valor 0.390 para o erro de modelização. Tal como seria de esperar, o resultado alcançado é ligeiramente melhor que o verificado com operadores algébricos. De acordo com a justificação apresentada no estudo da série caótica, tal situação deve-se ao facto de o número de parâmetros ajustados em cada iteração ser inferior, em virtude das características do operador mínimo na camada de regras. No entanto, dado que o modelo é composto unicamente por duas entradas, a interacção verificada na camada de regras pelo uso de operadores algébricos é reduzida. Daí que, nesta situação, o número de parâmetros ajustados em cada época seja também baixo, o que justifica o facto de os resultados obtidos não terem sofrido uma degradação significativa, ao invés do ocorrido na série caótica. Quanto ao número de funções de pertença, obteve-se exactamente o mesmo número que na situação precedente, i.e., 2, 2, 3, de acordo com a ordem em que as variáveis têm vindo a ser citadas.

Modelo com consequentes constantes

Tal como na situação anterior, foram utilizados primeiramente operadores algébricos. Deste modo, após 500 épocas de treino, o erro RMS convergiu para o valor 0.397, sendo definidas 3 funções de pertença para cada variável. Posteriormente, recorreu-se a operadores de truncatura, tendo-se obtido, contrariamente ao esperado, resultados menos satisfatórios, ao fim de 1000 épocas: um erro RMS com o valor 0.406. Quanto ao número de funções de pertença, obtiveram-se 2, 3 e 3 para as variáveis na ordem citada. Tal como no caso de treino livre, a situação verificada advém da simplicidade do modelo, a qual leva a que operadores de truncatura ou algébricos sejam aproximadamente equivalentes. De forma a analisar mais profundamente o problema, testou-se um modelo com três entradas, tendo sido obtidos resultados claramente melhores fazendo uso de operadores de truncatura.

Tal como no estudo da série caótica, chegou-se à conclusão de que a definição de estruturas com consequentes constantes não apresenta qualquer vantagem em termos da capacidade de previsão do modelo.

Sumariando, os resultados obtidos são sintetizados na Tabela 6.12. Assim, em virtude dos resultados apresentados, não há, aparentemente conclusões definitivas a retirar, em termos do tipo

de operadores utilizados. No entanto, de acordo com as justificações apresentadas nos parágrafos anteriores, conclui-se, mais uma vez, a especial adequação de operadores de truncatura ao problema da implementação de modelos interpretáveis. Quanto à questão da utilização de modelos com consequentes constantes, do seu uso não advém qualquer vantagem, tal como já havia sucedido no caso de estudo anterior.

Método		Tipo de Gaussianas	Número de Parâmetros	Operadores Difusos	Número de Épocas	RMSE
1	AS-CD	Generalizadas	28	Algébricos	4000	0.392
2	"	"	28	Truncatura	3000	0.390
3	AS-CC	"	36	Algébricos	500	0.397
4	"	"	32	Truncatura	1000	0.406

Tabela 6.12. Fornalha de gás: resultados de treino fora de linha restringido.

Da Tabela 6.12, verifica-se que a simulação 2 - consequentes difusos com operadores de truncatura - possibilitou o melhor desempenho em termos de precisão, o qual se revelou aceitável, de acordo com a Figura 6.10.

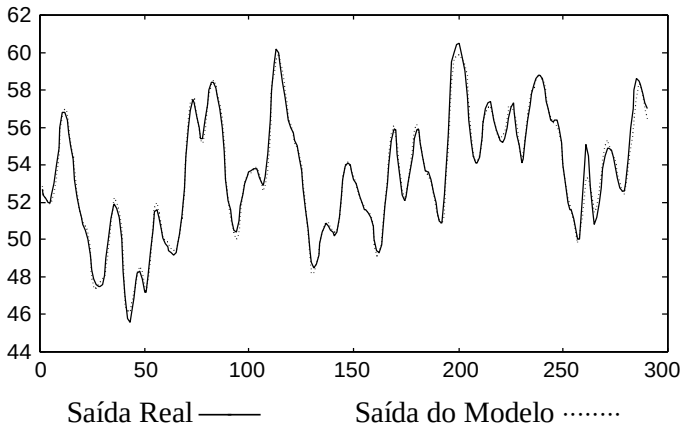


Figura 6.10. Fornalha de gás: modelização linguística interpretável.

Em relação às funções de pertença associadas a cada variável, os resultados obtidos são apresentados na Figura 6.11, onde se verifica a facilidade de etiquetagem que as caracteriza.

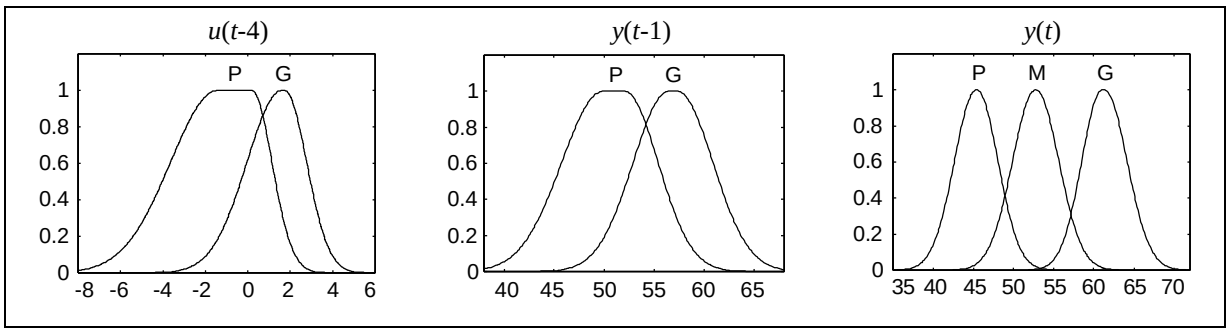


Figura 6.11. Fornalha de gás: funções de pertença com aprendizagem restringida em modelos com consequentes difusos.

Na mesma figura, as etiquetas *P*, *M* e *G* denotam, respectivamente, os termos linguísticos "pequeno", "médio" e "grande". Deste modo, obtêm-se as regras condicionais difusas definidas na Tabela 6.13.

Regra	***u*(*t*-4)**	***y*(*t*-1)**		***y*(*t*)**
1	P	P	Þ	M
2	P	G		G
3	G	P		P

Tabela 6.13. Fornalha de gás: descrição linguística da dinâmica.

6.4. Planta de Branqueamento de Pasta de Papel

Um dos objectivos iniciais deste trabalho de dissertação consistia no desenvolvimento de um modelo neuro-difuso para a planta de branqueamento de pasta de papel da Companhia de Celulose do Caima, S. A. No entanto, em virtude de algumas dificuldades encontradas, associadas à qualidade dos dados disponíveis, tal não foi concretizado com o sucesso desejado. Ainda assim, o presente trabalho deu origem à publicação de dois artigos científicos, nas conferências *EUFIT'98* [Paiva et al, 1998] e *ECC'99* [Paiva et al, 1999]. Nos pontos seguintes, apresenta-se o trabalho desenvolvido, as dificuldades encontradas e as conclusões retiradas.

6.4.1. Breve Descrição da Planta

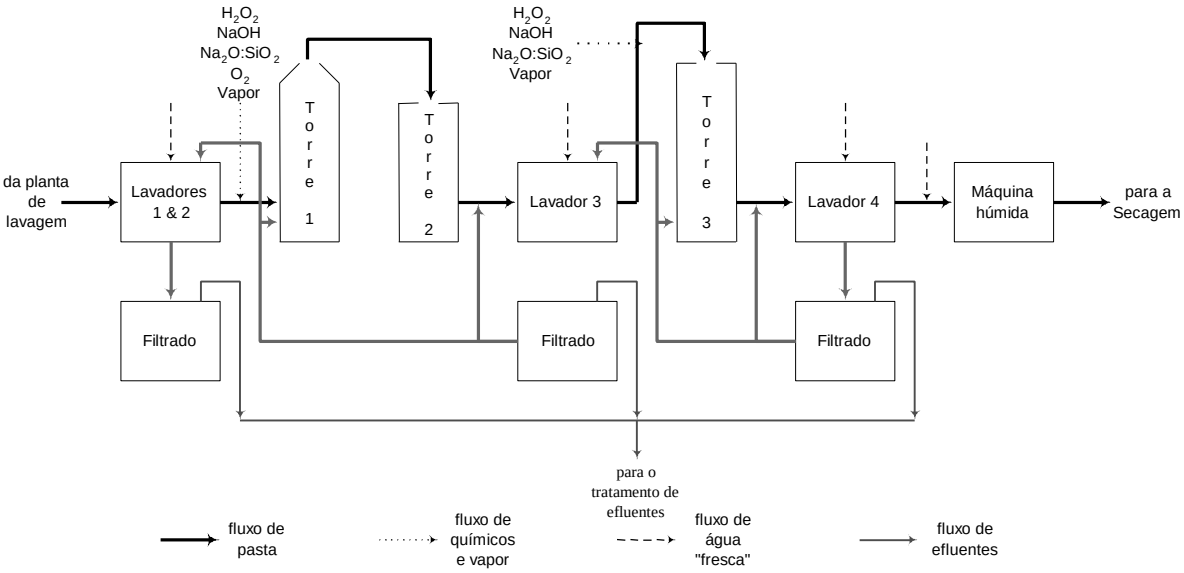


Figura 6.12. Esquema da secção de branqueamento da Companhia de Celulose do Caima, S.A.

Por questões de privacidade inerentes às leis de mercado, a descrição da planta é efectuada com grande brevidade, sendo incluídos unicamente os seus aspectos mais relevantes.

Assim, o objectivo principal do branqueamento prende-se com a descoloração da leninha presente nas fibras da madeira, a qual lhes confere um tom acastanhado. Deste modo, são adicionados alguns químicos, os quais, pela reacção com os cromóforos não branqueados, produzem cromóforos branqueados, de forma a que as características da pasta em termos de brilho, bem como de outras propriedades, satisfaçam os níveis exigidos pela indústria de papel.

Desta maneira, a planta de branqueamento da Companhia de Celulose do Caima, S.A. é totalmente isenta de cloro (TCF[65]), sendo composta por dois estágios correspondentes a 3 torres. No primeiro estágio, dá-se a extracção (E) com hidróxido de sódio e a oxidação com oxigénio (O) e com peróxido de hidrogénio (P). No segundo estágio ocorre apenas uma extracção e uma oxidação com peróxido de hidrogénio. Trata-se, pois, de uma sequência EOP_{NaOH} EP_{NaOH}, tal como se apresenta na Figura 6.12, adaptada de [Caima, 1994].

A pasta proveniente da secção de depuração, com consistência de aproximadamente 4%, é lavada em dois filtros rotativos, em funcionamento paralelo, com água limpa e filtrado proveniente de um terceiro filtro situado a jusante. A consistência da pasta à entrada dos filtros é de 1%, sendo de cerca de 12% à saída. O filtrado desta lavagem é parcialmente recirculado para os próprios lavadores, sendo o restante enviado à estação de tratamento de efluentes. Após esta primeira lavagem a pasta é misturada com os agentes branqueadores adicionados ao primeiro estágio (hidróxido de sódio, peróxido de hidrogénio e oxigénio). Simultaneamente, a sua temperatura é aumentada para cerca de 80ºC pela adição de vapor directo, após o que a mistura é bombeada para duas torres em série, onde as reacções de branqueamento (coloração dos cromóforos e corte das cadeias coradas) ocorrem em maior extensão. A primeira torre apresenta um tempo de retenção de 180 minutos, enquanto na segunda o mesmo é de 60 minutos. Após as acções descritas, a pasta é lavada no lavador 3 usando como corrente de lavagem o filtrado de um outro lavador situado a jusante. O filtrado do lavador 3 destina-se à lavagem nos primeiros dois filtros, situados antes do primeiro estágio de branqueamento. Esta recirculação visa recuperar água, mas sobretudo os agentes químicos branqueadores no estágio EP. Após esta lavagem a pasta é misturada com peróxido de hidrogénio e hidróxido de sódio, sendo igualmente a sua temperatura aumentada, pelo uso de vapor directo, para cerca de 80ºC. A pasta é posteriormente enviada à terceira torre, cujo tempo de residência é de 120 minutos. Da torre 3, a pasta é bombeada para o lavador 4 e lavada com o filtrado proveniente da máquina de formação. O filtrado desta lavagem é enviado ao lavador 3, enquanto que a corrente de pasta é enviada à máquina húmida e posteriormente à secagem, onde a sua humidade é reduzida para cerca de 10%.

Análise do brilho à saída

Existem algumas regras genéricas, as quais permitem prever de forma difusa o brilho final alcançado [Duarte, 1995]. Assim, esta variável é influenciada pelo caudal de peróxido de hidrogénio, o qual contribui para o seu aumento; do mesmo modo, o brilho aumenta com o pH da pasta, bem como com a sua consistência e temperatura, até um certo limiar; inversamente, decresce com o número de permanganato à entrada. Naturalmente, a informação referida é susceptível de ser comparada com a base de regras obtida na implementação dos algoritmos descritos, para efeitos de validação.

Ainda em [Duarte, 1995], é fornecida informação relativamente a alguns aspectos da dinâmica do processo, segundo a qual o atraso verificado entre a variável brilho à entrada e a

[65] *Totally Chlorine Free*, em terminologia inglesa.

variável brilho à saída é de aproximadamente 7 a 8 horas, sendo o mesmo para o caudal de entrada e o número de permanganato. Em relação ao caudal de peróxido, o atraso correspondente à sua adição na primeira torre é de aproximadamente 6.5 a 7.5 horas, sendo de 3 a 5 horas na segunda torre.

6.4.2. Resultados de Identificação

A qualidade final do branqueamento é influenciada por um número elevado de variáveis. De acordo com o conhecimento pericial, aquelas que revelam uma influência mais significativa sobre a qualidade final da pasta são o brilho à entrada, o caudal de pasta à entrada, o número de permanganato à entrada (o qual é uma medida de concentração da leninha) e o caudal de peróxido de hidrogénio em ambas as etapas. De entre as variáveis enunciadas, apenas o caudal de peróxido e o brilho à entrada apresentam uma excitação suficiente, pelo que foram as utilizadas na construção do modelo. Numa primeira fase, foi utilizada informação pericial, não tendo sido aplicado o algoritmo de selecção de entradas descrito no capítulo anterior, o qual seria utilizado posteriormente para comparação com a informação disponibilizada. Desta maneira, o modelo será composto por 4 entradas - o brilho à entrada e o caudal de peróxido na torre 1 com um atraso de 8 horas, o caudal de peróxido na torre 2 com um atraso de 4 horas e o brilho à saída uma hora antes - e uma saída - o brilho à saída da planta de branqueamento.

Os dados utilizados foram recolhidos com a planta em funcionamento em malha aberta (o controlo é do tipo *feedforward*, não havendo realimentação) com um intervalo de amostragem de 1 hora. Naturalmente, uma das primeiras questões colocadas aos peritos e engenheiros da companhia prendeu-se com a adequação do intervalo de amostragem, o qual parecia, à partida, demasiado elevado. No entanto, foi afirmado que tal escolha era suficiente, dado tratar-se de um processo com uma dinâmica bastante lenta.

Em relação aos dados obtidos, os mesmos foram analisados manualmente, tendo-se procedido à eliminação de *outliers*. Foi ainda aplicado um filtro passa-baixo com o intuito de suavizar o conjunto de dados original.

Assim, efectuada a aquisição de dados e o seu pré processamento, com as limitações referidas em termos de excitação de algumas variáveis, e determinada uma estrutura para o modelo, procedeu-se à sua identificação com base nas metodologias descritas neste trabalho.

As simulações foram conduzidas com um total de 1464 amostras, tendo sido utilizados 2/3, i.e., 976 amostras, para treino e as restantes para teste. O parâmetro r_a do agrupamento subtractivo foi definido com o valor 0.4, originando 11 regras condicionais difusas, após o que se procedeu à optimização do modelo utilizando-se operadores algébricos, funções Gaussianas generalizadas e uma velocidade de aprendizagem inicial de 0.005. Desta maneira, após 200 épocas de treino o erro RMS atingiu os valores 0.119 e 0.168 para os dados de treino e teste, os quais parecem, à primeira vista bastante satisfatórios, a menos de um pequeno efeito de sobreajustamento. Na Figura 6.13 são apresentados os resultados de modelização do brilho à saída para os dados de treino e de teste.

De facto, em termos visuais, os resultados obtidos parecem bastante satisfatórios, para mais tratando-se de um sistema de elevada complexidade. No entanto, uma análise mais detalhada permite chegar a uma conclusão totalmente diferente. Na verdade, restringido os dados de treino da Figura 6.13 a um intervalo limitado, e.g., 450 a 550, verifica-se a ocorrência de um efeito de perseguição da saída do modelo relativamente à saída real (Figura 6.14). Por outras palavras, conclui-se que o modelo aprendeu uma função do tipo $y(t) \approx y(t\text{-}1)$, pelo que o mesmo praticamente ignora o efeito das entradas. Procuraram-se, então, as causas do comportamento ocorrido.

Naturalmente, pensou-se, em primeiro lugar, tratar-se de um problema de escolha da estrutura, pelo que foram testadas diversas combinações de entradas, com atrasos e memórias variadas, sem que os resultados se alterassem. Aplicou-se, então, o método de selecção de entradas de Chiu a um conjunto bastante vasto de entradas e respectivos atrasos e memórias, tendo o método chegado à conclusão de que a única variável relevante para a previsão do brilho à saída é o próprio brilho verificado uma hora antes, i.e., $y(t) \approx y(t\text{-}1)$, de acordo com os resultados anteriores.

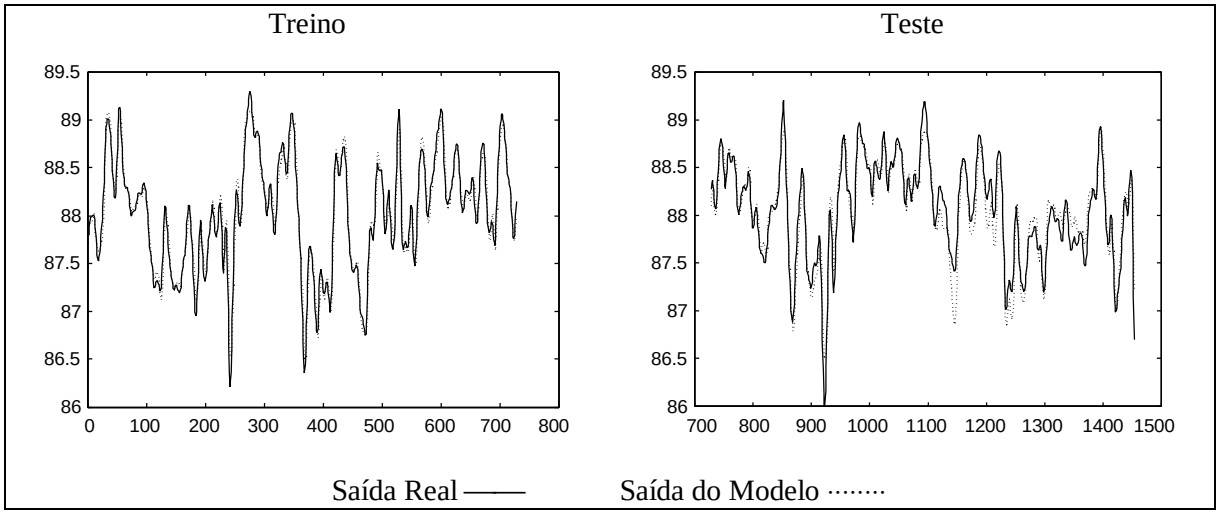


Figura 6.13. Planta de branqueamento: resultados de identificação.

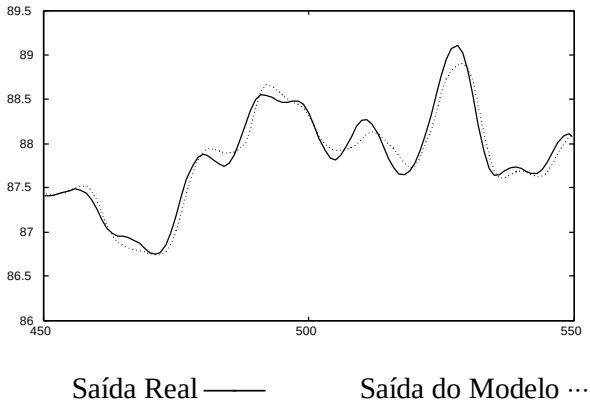


Figura 6.14. Planta de branqueamento: efeito de perseguição.

Em virtude da estranheza da justificação encontrada, colocou-se a hipótese de os problemas derivarem de alguma variabilidade da dinâmica do sistema em termos de atrasos, em consequência de variações no caudal de pasta à entrada. Esta hipótese, embora viável, pareceu não ser a principal razão do ocorrido, uma vez que as variações referidas não eram muito significativas.

Finalmente, foi encontrada a causa mais provável para o comportamento verificado. Na verdade, de acordo com [Silva, 1994], após a adição de agentes branqueadores na pasta, o seu brilho aumenta até estabilizar ao fim de aproximadamente 10 minutos! O resultado apresentado vai totalmente contra a escolha efectuada para o intervalo de amostragem, o qual foi de 1 hora! Assim, a deficiência na escolha do intervalo não permite captar a evolução dinâmica do sistema, pelo que os dados não apresentam qualquer consistência, o que conduz à conclusão de que somente a variável de saída é relevante.

Em face do exposto, conclui-se naturalmente da impossibilidade de modelizar a planta de branqueamento com os dados obtidos, sendo necessário efectuar recolhas suficientemente informativas, o que implicava alterações no sistema de aquisição de dados, bem como algumas paragens na produção, incomportáveis para a companhia. Deste modo, esperam-se evoluções futuras favoráveis ou a possibilidade de serem efectuados testes com base em outros processos.

6.5. Sumário

O capítulo presente apresentou a aplicação das metodologias descritas ao longo do trabalho de dissertação presente a alguns casos de estudo comuns na literatura afim, nomeadamente a série caótica Mackey-Glass e a fornalha de gás de Box e Jenkins.

Com base no conjunto de experiências conduzidas, verificou-se que a aplicações de técnicas neuro-difusas ao problema da identificação de sistemas constitui, de facto, uma abordagem a levar em consideração. Essas mesmas experiências possibilitaram que se retirassem algumas ilações em relação aos aspectos de modelização neuro-difusa abordados. Assim, quanto à selecção de entradas, embora fosse desejável efectuar um estudo experimental exaustivo, os resultados obtidos revelaram-se satisfatórios, sugerindo a possibilidade do método constituir um bom indicador das entradas relevantes em sistemas de larga escala. Em relação à aprendizagem da estrutura, verificou-se que o algoritmo de agrupamento subtractivo constitui um esquema mais eficiente, tanto a nível computacional como a nível do desempenho final dos modelos, do que o algoritmo proposto por Lin na sua arquitectura NFCN.

Em termos de estruturas difusas, os resultados obtidos permitem retirar conclusões em conformidade com as apresentadas por outros autores. Nomeadamente, verificou-se que os modelos do tipo Takagi-Sugeno de ordem 1 originam modelos mais precisos em termos de erro de modelização. Este aspecto deriva, fundamentalmente, do facto de tais estruturas constituírem uma abordagem baseada na comutação suave entre vários modelos lineares locais, a qual apresenta maiores possibilidades do que a abordagem interpolativa inerente aos modelos de ordem 0 e linguísticos. O ponto negativo resultante desta estrutura prende-se com o facto de o custo computacional daí resultante ser consideravelmente superior ao das duas outras estruturas referidas. Assim sendo, a sua aplicação à identificação em tempo real deve ser conduzida com cautelas especiais. Neste aspecto, concluiu-se que os modelos linguísticos constituem a abordagem mais eficiente a nível computacional, originando, contudo, modelos um pouco menos precisos. Por conseguinte, chegou-se à conclusão de que as arquitecturas Takagi-Sugeno de ordem 0 constituem soluções de compromisso entre precisão e eficiência satisfatórias.

Em qualquer dos aspectos supracitados, a utilização de operadores algébricos provou ser preferível, tal como seria de esperar pelo referido nos capítulos anteriores. Quanto ao tipo de funções de pertença, embora as funções Gaussianas generalizadas tenham originado, em geral, resultados um pouco melhores do que os produzidos por Gaussianas simples, o incremento considerável que se verificou no número de parâmetros não justifica o seu uso. Assim, as funções simples permitem a obtenção de modelos mais simples em termos de número de parâmetros livres, o que é conseguido apenas com uma pequena degradação do desempenho.

Em relação à problemática da interpretabilidade, o algoritmo proposto possibilitou a obtenção de resultados aceitáveis. De facto, chegou-se, para os modelos considerados, a uma boa solução de compromisso entre precisão e transparência. Neste aspecto, a utilização de operadores

de truncatura revelou-se fundamental, dado originar o ajuste de um número mais reduzido de parâmetros em cada iteração, o que constitui uma vantagem no sentido da aplicação do procedimento de monitorização proposto. O desenvolvimento de modelos interpretáveis linguisticamente constitui, também, a motivação fundamental para a utilização de funções de pertença Gaussianas generalizadas.

Capítulo 7

CONCLUSÕES E PERSPECTIVAS

Desde que Ebrahim Mamdani desenvolveu a primeira aplicação prática da lógica difusa, um vasto caminho, recheado de vales e montanhas, de trevas e luz, tem sido percorrido. Como resultado das evoluções verificadas nas tecnologias de informação, é agora possível implementar algoritmos mais sofisticados num tempo de execução aceitável, tais como os sistemas híbridos inteligentes descritos neste trabalho. Muito embora o número de aplicações de sucesso da lógica difusa, ou, mais genericamente, da *soft computing*, seja crescente, muitos desafios são ainda colocados, particularmente no campo do estudo de metodologias de análise e síntese, sistemáticas, rigorosas e generalizáveis, fundamentais para o desenvolvimento de aplicações robustas, fiáveis e seguras.

O trabalho apresentado ao longo deste documento visou, acima de tudo, descrever e analisar os aspectos fundamentais relacionados com a temática da identificação neuro-difusa, análise essa efectuada enfatizando sobretudo aspectos experimentais, sem, no entanto, ignorar alguns dos aspectos teóricos subjacentes.

Neste capítulo apresentam-se as principais conclusões retiradas do trabalho realizado, assim como possíveis direcções para investigação futura.

7.1. Conclusões Gerais

Genericamente, concluiu-se que as arquitecturas híbridas neuro-difusas constituem uma abordagem de elevado potencial na identificação de sistemas dinâmicos, não só em virtude de gozarem da propriedade da aproximação universal, mas também por apresentarem vantagens importantes em termos de transparência do conhecimento armazenado. De facto, foram estudadas várias estruturas difusas, nomeadamente, sistemas linguísticos e sistemas do tipo Takagi-Sugeno de ordem 0 e 1, sintonizadas recorrendo ao treino de redes neuronais. Essas mesmas estruturas foram utilizadas na identificação de alguns casos de estudo frequentes na literatura. Qualquer um dos esquemas analisados possibilitou resultados aceitáveis, tendo-se concluído sobre as vantagens de utilização de operadores algébricos e funções Gaussianas simples (no caso da interpretabilidade não ser um dos objectivos de modelização). Verificou-se que modelos com consequentes constantes apresentam bons compromissos entre capacidades de representação e eficiência computacional. No entanto, a sua implementação e integração em sistemas industriais reais requer o desenvolvimento de metodologias sistematizadas de análise e síntese, de forma a que se

satisfaçam alguns requisitos fundamentais a nível de estabilidade da aprendizagem, necessários à consecução dos objectivos de robustez, segurança e desempenho das aplicações, particularmente em tempo real.

No que concerne à aprendizagem de regras, foram analisadas algumas estratégias baseadas essencialmente em métodos de eliminação de regras e algoritmos de agrupamento. Neste sentido, concluiu-se que o método de agrupamento subtractivo apresenta propriedades que o tornam particularmente interessante na inicialização de estruturas às quais se apliquem esquemas de optimização não linear. Concluiu-se ainda sobre a sua maior eficiência, tanto a nível computacional como a nível do desempenho final dos modelos, comparativamente ao algoritmo proposto por Lin na sua arquitectura NFCN.

Ainda no contexto da aprendizagem da estrutura, mais particularmente na selecção de entradas relevantes, constatou-se que a grande maioria dos métodos se caracterizam por uma forte componente heurística, pelo que a sua utilização não apresenta o grau de robustez e fiabilidade requerido, sendo unicamente utilizáveis como indicadores. Dos vários algoritmos possíveis, optou-se pela implementação do método de Chiu, dado constituir uma técnica simples e eficiente, particularmente adequada ao desenvolvimento de modelos difusos.

Relativamente ao aspecto da interpretabilidade, propôs-se a integração de um esquema de monitorização de parâmetros na aprendizagem, o qual possibilitou a obtenção de resultados aceitáveis, em termos de precisão dos modelos e interpretabilidade linguística, com a condição de se utilizarem operadores difusos de truncatura. Neste trabalho, optou-se por não deixar passar em claro a questão da transparência do modelo obtido, uma vez que este aspecto constitui a filosofia dos sistemas difusos.

Quanto à aprendizagem em linha, não se deu uma atenção particular a este ponto, tendo-se, unicamente, adaptado as técnicas de identificação por lotes, por forma a tirar-se partido da propriedade da localidade, inerente às estruturas neuro-difusas.

7.2. Perspectivas de Desenvolvimento

Um dos objectivos inicias do trabalho presente, o qual não foi totalmente atingido, consistia na aplicação das metodologias estudadas a sistemas de larga escala, uma vez que os mesmos poderão ser mais conclusivos, dado serem susceptíveis de apresentar algumas dificuldades não sentidas nos casos de estudo analisados. Assim, uma vez que a tentativa de modelização de uma planta de branqueamento de pasta de papel se revelou infrutífera em virtude da deficiente qualidade dos dados disponibilizados, a análise da viabilidade de aplicação das técnicas descritas a sistemas de larga escala permanece uma questão em aberto.

Um outro aspecto, relacionado com o anterior, prende-se com o estudo de sistemas MIMO, com acoplamento. Embora a generalidade dos autores afirmem que as arquitecturas neuronais possibilitem a modelização de forma trivial da classe de sistemas referida, efectuou-se um pequeno estudo, não documentado neste trabalho, o qual indicou ser preferível o desenvolvimento de um conjunto de sistemas MISO independentes, um para cada saída do sistema.

Relativamente à aprendizagem da estrutura, constatou-se que uma das limitações do algoritmo de agrupamento utilizado se prende com a falta de flexibilidade relativamente à forma e dimensão dos grupos encontrados. Por conseguinte, seria interessante investigar algoritmos susceptíveis de ultrapassarem a desvantagem enunciada.

No contexto da aprendizagem em tempo real, verificou-se que o esquema de aprendizagem incremental exposto neste trabalho requer o desenvolvimento prévio, fora de linha, de um modelo inicial. O aspecto referido constitui uma limitação no caso de se ter por objectivo a implementação de sistemas autónomos, que aprendam sem qualquer conhecimento prévio, unicamente com base na interacção com o ambiente exterior. Neste sentido, o problema da aprendizagem em linha da estrutura, ou aprendizagem construtiva, constitui uma área de elevado potencial científico a investigar.

Na sequência do trabalho de identificação apresentado coloca-se, naturalmente, a hipótese da aplicação das estratégias descritas a problemas de controlo, nomeadamente em esquemas do tipo modelo inverso ou controlo por modelo interno. Deste modo, são colocadas as questões referidas anteriormente, relativas a robustez, segurança, desempenho e eficiência computacional, necessárias ao controlo e adaptação em tempo real.

Quanto à questão da interpretabilidade, embora os resultados obtidos tenham sido satisfatórios, verificou-se que os mesmos só foram possíveis com a restrição relativa à utilização de operadores de truncatura, em virtude do esquema de monitorização proposto. Adicionalmente, prevê-se que as dificuldades aumentem em sistemas de maior complexidade que os testados, uma vez que o seu maior número de parâmetros livres poderá não se coadunar com a estratégia de monitorização descrita. Deste modo, o desenvolvimento de um algoritmo do tipo gradiente restringido, segundo o qual o ajuste de parâmetros siga a direcção do gradiente imposta pelas restrições consideradas, apresenta vantagens claras. Na verdade, a implementação de uma metodologia desta natureza constituiria uma melhoria significativa relativamente ao método proposto.

Finalmente, uma das extensões ao trabalho elaborado consistirá no desenvolvimento de uma interface gráfica, a qual encapsulará os vários algoritmos implementados ao longo deste trabalho. A aplicação final poderá constituir uma ferramenta computacional útil no estudo e identificação de sistemas dinâmicos, tanto a nível didáctico como de investigação científica, fundamentalmente numa vertente mais experimental.

BIBLIOGRAFIA

Akaike H. (1973). "Information theory and an extension of the maximum likelihood principle", *2nd International Symposium on Information Theory*, pp. 267-281.

Albus J. S. (1975). "A new approcah to manipulator control: cerebellar model articulation control (CMAC), *Transactions of ASME, Journal of Dynamics Systems, Measurement and Control*, Vol. 97, pp. 228-233.

Anderson J. A. (1972). "A simple neural network generating on interactive memory", *Mathematical Biosciences*, Vol. 14, pp. 197-220.

Åstrom K. J. (1970). *Introduction to Stochastic Control Theory*. Academic Press, New York.

Åstrom K. J., Wittenmark B. (1984). *Computer Controlled Systems: Theory and Design*. Prentice Hall, Englewood Cliffs.

Babuška R. e Setnes M. (1998). "Data-driven construction of transparent fuzzy models: methods and applications", *Proceedings of the European Congress on Fuzzy and Intelligent Technologies - EUFIT'98*, pp. 594-602.

Barto A. G., Sutton R. S. e Anderson C. W. (1983). "Neuronlike adaptive elements that can solve difficult learning problems", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 13, pp. 834-846.

Berenji H. R. e Khedkar P. (1992). "Learning and tuning fuzzy logic controllers through reinforcements", *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, pp. 724-740.

Bezdek J.C. (1981). "Pattern recognition with fuzzy objective function algorithms", *J. Math. Biol.*, Vol. 1, pp. 57-71.

Bezdek J.C. (1993). "Fuzzy models: what are they and why?", *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 1, pp. 3-13.

Bossley K. M. (1997). *Neurofuzzy Modelling Approaches in System Identification*, PhD Thesis, Department of Electronics and Computer Science, Faculty of Engineering and Applied Science, University of Southampton, United Kingdom.

Box G. E. P. e Jenkins G. W. (1970). *Time Series Analysis, Forecasting and Control*. Holden Day,

San Francisco.

Broomhead D. S. e Lowe D. (1988). "Multivariable function interpolation and adapative networks", *Complex Systems*, Vol. 2, pp. 321-355.

Brown M. e Harris C. (1994). *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, Hemel Hempstead.

Buckley J. J. (1993). "Sugeno type controllers are universal approximators", *Fuzzy Sets and Systems*, Vol. 53, pp. 299-304.

Buckley J. J., Hayashi Y. (1995). "Neural nets for fuzzy systems", *Fuzzy Sets and Systems*, Vol. 71, pp. 265-276.

Caima's Work Group on Best (1994). "TCF bleaching at Caima: past and future", Relatório Técnico, Companhia de Celulose do Caima.

Castro J. L. (1995). "Fuzzy logic controllers are universal approximators", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 4, pp. 629-635.

Chen S. e Billings S. A. (1992). "Neural networks for nonlinear dynamic system modelling and identification", *International Journal of Control*, Vol. 56., No. 2, pp. 319-346.

Chiu S. L. (1994). "Fuzzy model identification based on cluster estimation", *Journal of Intelligent and Fuzzy Systems*, Vol. 2, No. 3, pp. 267-278.

Chiu S. L. (1996). "Selecting input variables for fuzzy models", *Journal of Intelligent and Fuzzy Systems*, Vol. 4, pp. 243-256.

Cho K. B. e Wang B. H.(1996). "Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction", *Fuzzy Sets and Systems*, Vol. 83, pp. 325-339.

Davé R. N. e Krishnapuram R. (1997). "Robust clustering methods: a unified view", *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 2, pp. 270-293.

Dias J. M. e Dourado A. (1999). "A self-organizing fuzzy controller with a fixed maximum number of rules and an adaptive similarity factor", *Fuzzy Sets and Systems*, Vol. 103, pp. 27-48.

Dray G, Peton N. e Pearson D. W. (1998). "Centre influence modification in subtractive clustering", *Proceedings of the 3rd Portuguese Conference on Automatic Control - CONTROLO'98*, pp. 703-706.

Driankov D., Hellendoorn H e Reinfrank M. (1993). *An Introduction to Fuzzy Control*, Springer-Verlag, Berlin.

Duarte B. (1995). "Bleaching plant fuzzy model", Relatório Técnico, Companhia de Celulose do

Caima.

Eberhart R. C. e Dobbins R. W. (1990). *Neural Networks PC Tools - A Practical Guide*, Academic Press, San Diego, U.S.A.

Elman J. L. (1990). "Finding structure in time", *Cognitive Science*, vol. 14, pp. 179-211.

Figueiredo M. e Gomide F. (1997). "Adaptive neuro fuzzy modeling", *Proceedings of FUZZ-IEEE'97*, pp. 1567-1572.

Franklin G. F. e Powell J.D. (1980). *Digital Control of Dynamic Systems*, Addison-Wesley, Reading.

Friedland B. (1986). *Control System Design: An Introduction to State-Space Methods*, McGraw-Hill, New York.

Fukushima K. (1980). "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", *Biol. Cybernetics*, Vol. 36, pp. 193-202.

Funahashi, K. (1989). "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, Vol. 2, pp. 183-192.

Furuya T., Kokubo A. e Sakamoto T. (1998). "NFS: Neuro fuzzy inference system", *Proceedings of the International Conference on Fuzzy Systems ansd Neural Networks - IIZUKA'88*, pp. 219-230.

Girosi F. e Poggio T. (1990). "Networks and the best approximation property", *Biological Cybernetics*, vol. 63, pp. 169-176.

Gorinevsky D. (1995). "On the presistency of excitation in radial basis function network identification of nonlinear systems", *IEEE Transactions on Neural Networks*, Vol. 6, No. 5, pp. 1237-1244.

Glorennec P. Y. (1994). "Learning algorithms for neuro-fuzzy networks", in Kandel A., Langholz G., *Fuzzy Control Systems*, CRC Press, Boca Raton, U.S.A.

Grossberg S. (1973). "Contour enhancement, short memory an constancies in reverberating neural networks", *Studies in Applied Mathematics*, Vol. 52, no. 3, pp. 213-257.

Gustafson D. E. e Kessel W. C (1979). "Fuzzy clustering with a fuzzy covariance matrix", *Proceedings of IEEE CDC*, pp. 761-766.

Harris C. J., Moore C. G. e Brown, M. (1993). *Intelligent Control - Aspects of Fuzzy Logic and Neural Nets*, World Scientific Publishing, Singapore.

Haykin S. (1994). *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing

Company.

Hebb D. O. (1949). *The Organization of Behavior*, Wiley, New York.

Hecht-Nielsen R. (1990). *Neurocomputing*, Addison-Wesley, Reading.

Henriques J. e Dourado A. (1998). "Adaptive control using a recurrent neural network observer", *Proceedings of the 3rd Portuguese Conference on Automatic Control*, Coimbra, pp. 583-589.

Höhle U e Neff Stout L. (1991). "Foundations of fuzzy sets", *Fuzzy Sets and Systems*, Vol. 40, pp. 257-296.

Holland J. M.(1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, U.S.A.

Hopfield, J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences*, Vol. 79, pp. 2554-2558.

Horikawa, S. Furuhashi T. e Uchikawa Y. (1992). "On fuzzy modeling using neural networks with the back-propagation algorithm", *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, pp. 801-806.

Hunt K. J., Sbarbaro D., Zbikowski R. e Gawthrop P. J. (1992). "Neural networks for control systems - a survey", *Automatica*, Vol. 28, No. 6, pp. 1083-1112.

Ichihashi H. e Watanabe T. (1990). "Learning control by fuzzy models using simplified fuzzy reasoning", *Journal of Japan Society for Fuzzy Theory and Systems*, Vol. 2, No. 3, pp. 429-437 (Em Japonês).

Ishibuchi H., Fujioka R. e Tanaka H. (1993a). "Neural networks that learn from fuzzy if-then rules", *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 2, pp. 85-97.

Ishibuchi H., Tanaka H. e Okada H. (1993b). "An architecture of neural networks with interval weights and its application to fuzzy regression analysis", *Fuzzy Sets and Systems*, Vol. 57, pp. 27-39.

Ivakhnenko A. G., Krotov G. I. e Visotsky V. N. (1979). "Identification of the mathematical model of a complex system by the self-organization method", in Halfon E., *Theoretical Systems Ecology: Advances and Case Studies*, Academic Press, New York.

Jacobs R. A. (1988). "Increased rates of convergence through learning rate adaptation", *Neural Networks*, Vol. 1, pp. 295-307.

Jackson J. E. (1991). *A User's Guide to Pricipal Components*, John Wiley & Sons.

Jang J.-S. R. (1993). "ANFIS: Adaptive Network-based Fuzzy Inference System", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 3, pp. 665-685.

Jang J.-S. R. e Sun C.-T. (1993). "Functional equivalence between radial basis function networks and fuzzy inference systems", *IEEE Transactions on Neural Networks*, Vol. 4, No. 1, pp. 156-159.

Jordan M. I. (1986). "Attractor dynamics and parallelism in connectionist sequential machines, *Proceeddings of the 8th Annual Conference of the cognitive Science Society*, pp. 531-546.

Juang C.-F. e Lin C.-T. (1998). "An on-line self-constructing neural fuzzy inference network and its applications", *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 1, pp. 12-32.

Kalman R. E. e Bucy (1961). "New results in linear filtering and prediction theory", *Transactions of ASME, Journal of Basic Engineering (ser. D)*, Vol. 83, pp. 95-108.

Keller J. M., Yager R. R. e Tahani H. (1992). "Neural network implementation of fuzzy logic", *Fuzzy Sets and Systems*, Vol. 45, pp. 1-12.

Kickert W. e van Nauta Lemke H. R. (1976). "The application of fuzzy theory to warm water process", *Automatica*, Vol. 12, No. 4, pp. 301-308.

Kohonen T. (1972). "Correlation matrix memories", *IEEE Transactions on Computers*, Vol. 21, No. 4, pp. 197-220.

Kohonen T. (1989). *Self-Organization and Associative Memory*, 3rd edition, Springer-Verlag, Berlin.

Kosko B. (1992). *Neural Networks and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs.

Kröse B. J. A. e van der Smagt P. P. (1993). *An Introduction to Neural Networks*, 5th edition, The University of Amsterdam, The Netherlands.

Kuo B. C. (1987). *Automatic Control Systems*, Holt, Rinehart and Winston, New York.

Lee C. C. (1990a). "Fuzzy logic in control systems: fuzzy logic controller - part I", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 2, pp. 404-418.

Lee C. C. (1990b). "Fuzzy logic in control systems: fuzzy logic controller - part II", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 2, pp. 419-435.

Lin C.- T. (1995). "A neural fuzzy control scheme with structure and parameter learning", *Fuzzy Sets and Systems*, Vol. 70, pp. 183-212.

Lin C.-T., Lin C.-J. e George Lee C. S. (1995). "Fuzzy adaptive learning control network with on-line neural learning", *Fuzzy Sets and Systems*, Vol. 71, pp. 25-45.

Lin C.-T., Lu Y.-C (1996). "A neural fuzzy system with fuzzy supervised learning", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 26, No.5, pp. 744-763.

Lin Y. e Cunningham III G. A. (1995). "A new approach to fuzzy-neural modelling", *IEEE Transactions on Fuzzy Systems*", Vol. 3, No.2, pp. 190-198.

Ljung L. (1987). *System Identification - Theory for the User*, Prentice Hall, Englewood Cliffs.

Luenberger D. (1971). "An introduction to observers", *IEEE Transactions on Automatic Control*, Vol. 16, pp. 596-603.

Mackey M. C. e Glass L. (1977). "Oscillation and chaos in physiological control systems", *Science*, vol. 197, pp. 287-289.

Mamdani E. H. (1974). "Applications of fuzzy algorithms for control of a simple dynamic plant", *Proceedings of the IEE*, Vol. 121, No. 12, pp. 1585-1588.

Mamdani E. H. e Assilian S. (1975). "An experiment in linguistic synthesis with a fuzzy logic controller", *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13.

Martins de Carvalho J. L. (1993). *Dynamical Systems and Automatic Control*, Prentice-Hall, Hemel Hempstead.

McCulloch W. S. e Pitts W. (1943). "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133.

McClelland J. L. e Rumelhart D. E. (1986). *Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Vol. 2: Psychological an Biological Models*, MIT Press, Cambridge, U.S.A.

Mills P. M., Zomaya A. Y. e Tadé M. O. (1996). *Neuro-Adaptive Process Control: A Practical Approach*, John Wiley & Sons, Chichester, England.

Minsky M. e Papert S. (1969). *Perceptrons: An Introduction to Computational Geometrie*, the MIT Press.

Moody J. E. e Darken C. J. (1989). "Fast learning in networks of locally-tuned processing units", *Neural Computation*, Vol. 1, pp. 281-294.

Narendra K. e Parthasarathy K. (1990). "Identification and control of dynamical systems using neural networks", *IEEE Transactions on Neural Networks*, Vol.1. No.1, pp. 4-27.

Nauck D. (1994). "Building neural fuzzy controllers with NEFCON-I", in Kruse R., Gebhardt J. e Palm R., *Fuzzy Systems in Computer Science*, Vieweg, Braunschweig, Germany.

Nauck D. e Kruse R. (1995). "NEFCLASS – A neuro-fuzzy approach for the classification of data", in George K. M., Carrol J. H., Deaton E., Oppenheim D., Hightower J., *Applied*

Computing 1995, Proceedings 1995 ACM Symposium on Applied Computing, ACM Press, New York, pp. 461-465.

Nauck D. e Kruse R. (1999). "Neuro-fuzzy systems for function approximation", *Fuzzy Sets and Systems*, Vol. 101, pp. 261-271.

Nomura H., Hayashi I. e Wakami N. (1992). "A learning method of fuzzy inference rules by descent method", *Proceedings of the IEEE Conference on Fuzzy Systems*, pp. 203-210.

Ogata K. (1990). *Modern Control Engineering*, Prentice Hall, Englewood Cliffs.

Oshima W., Yasunobu S. e Sekino S. (1988). "Automatic train operation system based on predictive fuzzy control", *International Workshop on Artificial Inteligence for Industrial Applications*, pp. 485-489.

Paiva, R. P. (1997). *Sistema Neuro-Difuso com Aprendizagem Supervisionada Difusa*, Trabalho realizado no âmbito da cadeira "Controlo Inteligente" do curso de Mestrado em Engenharia Informática, Departamento de Engenharia Informática, Faculdade de Ciências e Tecnologia, Universidade de Coimbra.

Paiva R. P., Dourado A. e Duarte B. (1998). "A neuro-fuzzy system for modelling of a bleaching plant", *Proceedings of the European Congress on Fuzzy and Intelligent Technologies - EUFIT'98*, Vol. 3, pp. 1539-1543.

Paiva R. P., Dourado A. e Duarte B. (1999). "Applying subtractive clustering for neuro-fuzzy modelling of a bleaching plant", *Proceedings of the European Control Conference - ECC'99*, CD-ROM.

Papoulis A. (1973). *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, New York.

Park J. e Sandberg I. W. (1991). "Universal approximation using radial-basis-function networks", *Neural Computation*, vol. 3, pp. 246-257.

Pedrycz W. (1995). *Fuzzy Sets Engineering*, CRC Press.

Pereira C. (1996). *Aprendizagem em Tempo Real de Redes Neuronais Aplicada à Identificação e Controlo de Sistemas*, Tese de Mestrado, Departamento de Engenharia Informática, Faculdade de Ciências e Tecnologia, Universidade de Coimbra.

Pham D. T. e Xing L. (1995). *Neural Networks for Identification, Prediction and Control*, Springer-Verlag, London.

Polak E. (1971). *Computational Methods in Optimization*, Academic Press, New York.

Proczyk T. J. e Mamdani E. H. (1979). "A linguistic self-organizing process controller", *Automatica*, Vol. 15, pp. 15-30.

Reed R. (1993). "Pruning algorithms - a survey", *IEEE Transactions on Neural Networks*, Vol. 4, No. 5, pp.740-747.

Rosenblatt F. (1958). "The perceptron: a probabilistic model for information storage and organization in the brain", *Psychological Review*, vol. 65, pp. 386-408.

Ross T. J. (1995). *Fuzzy Logic with Engineering Applications*, McGraw-Hill.

Rumelhart D. E. e McClelland J. L. (1986). *Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, MIT Press, Cambridge, U.S.A.

Setnes M. (1995). *Fuzzy Rule-Base Simplification Using Similarity Measures*, MSc Thesis, Department of Electrical Engineering, Delft University of Technology, The Netherlands.

Shann J. J. e Fu H. C. (1995). "A fuzzy neural network for rule acquiring on fuzzy control systems", *Fuzzy Sets and Systems*, Vol. 71, pp. 345-357.

Silva G. (1994). *Modelação, Identificação e Controlo na Indústria da Pasta de Papel*, Tese de Mestrado, Departamento de Engenharia Electrotécnica e de Computadores, Instituto Superior Técnico, Universidade Técnica de Lisboa.

Sjöberg J., Hjalmarsson H. e Ljung L. (1994). "Neural networks in system identification", *Proceedings of the 10th IFAC Symposium on System Identification (SYSID'94)*, pp. 49-72.

Söderström, T. e Stoica P. (1989). *System Identification*, Prentice Hall, Hemel Hempstead.

Sousa J. M., Babuska R. e Verbruggen H. B. (1997). "Internal model control with a fuzzy model: application to an air-conditioning system", *Proceedings of FUZZ-IEEE'97*, pp. 207-212.

Sugeno M. e Kang G. T. (1988). "Structure identification of fuzzy model", *Fuzzy Sets and Systems*, Vol. 28, pp. 15-33.

Sugeno M. e Yasukawa T. (1993). "A fuzzy-logic-based approach to qualitative modeling", *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 1, pp. 7-31.

Takagi H. e Hayashi I. (1988). "Artificial-neural-network driven fuzzy reasoning", *Proceedings of the International Conference on Fuzzy Systems ansd Neural Networks - IIZUKA'88*, pp. 183-184.

Takagi T. e Sugeno M. (1985). "Fuzzy identification of systems and its applications to modelling and control", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, No. 1, pp. 116-132.

Tanaka K. e Sugeno M. (1992). "Stability analysis and design of fuzzy control systems", *Fuzzy Sets and Systems*, Vol. 45, pp. 135-156.

Thau F. E. (1973). "Observing the state of nonlinear dynamic systems", *International Journal of*

Control, Vol. 17, pp. 471-479.

Valente de Oliveira, J. (1992). *Identificação e Modelação Difusa de Sistemas Dinâmicas*, Tese de Mestrado, Departamento de Engenharia Electrotécnica e de Computadores, Instituto Superior Técnico, Universidade Técnica de Lisboa.

Valente de Oliveira, J. (1995). "A design methodology for fuzzy system interfaces", *IEEE Transactions on Fuzzy Systems*, Vol. 3, No. 4, pp. 404-414.

Victor J. e Dourado A. (1997). "Adaptive scaling factors algorithm for the fuzzy logic controller", *Proceedings of FUZZ-IEEE'97*, Vol. 2, pp. 1021-1026.

Victor J. (1998). *Projecto e Aplicação de Controladores Difusos em Tempo Real*, Tese de Mestrado, Departamento de Engenharia Informática, Faculdade de Ciências e Tecnologia, Universidade de Coimbra.

von Altrock, C. (1995). *Fuzzy Logic and NeuroFuzzy Applications Explained*, Prentice Hall, Upper Saddle River, New Jersey.

Wang L. X. (1992). "Fuzzy systems are universal approximators", *Proceedings of the IEEE Conference on Fuzzy Systems*, pp. 1163-1170.

Wang L. X. (1994). *Adaptive Fuzzy Systems and Control: Design and Stabilty Analysis*, Prentice Hall, Englewood Cliffs.

Wang L. X. e Mendel J. M. (1992a). "Generating fuzzy rules by learning from examples", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 6, pp. 1414-1427.

Wang L. X. e Mendel J. M. (1992b). "Back-propagation fuzzy system as nonlinear dynamic system identifier", *Proceedings of the IEEE Conference on Fuzzy Systems*, 1409-1418.

Wellstead P. E. (1979). *Introduction to Physical System Modelling*. Academic Press, New York.

Werbos P. J. (1974). "Beyond regression: new tools for prediction and analysis in the behavioral sciences", *MSc Thesis*, Harvard University, U.S.A.

Widrow B. e Hoff M. E. (1960). "Adaptive Switching Circuits", *1960 IRE WESCON Convention Record*, New York, pp. 96-104.

Widrow B. e Stearns S. D. (1985). *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, U.S.A.

Yager R. R. e Filev D. P. (1994). "Approximate clustering via de mountain method", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, No. 8, pp. 1279-1284.

Zadeh L A. (1965). "Fuzzy sets", *Information and Control*, Vol. 8, pp. 338-358.

Zadeh L A. (1968). “Fuzzy algorithms”, *Information and Control*, Vol. 12, pp. 94-102.

Zadeh L. A. (1971). “Toward a theory of fuzzy systems”, in Kalman R. E. e De Claris N., *Aspects on Network and Systems Theory*, Eds. New York: Holt, Rinehart and Winston.

Zadeh L A. (1973). “Outline of a new approach to the analysis of complex systems and decision processes”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3, No.1, pp. 28-44.

Zadeh L A. (1994). “Soft Computing and Fuzzy Logic”, *IEEE Software*, November 1994, pp. 48-56.