

Universidade do Minho
Departamento de Informática

Algoritmos Genéticos
e
Redes Neurais
na
Previsão de Séries Temporais
por

Paulo Alexandre Ribeiro Cortez

Dissertação apresentada à Universidade do Minho para obtenção do grau de
Mestre em Informática, na especialidade de Informática de Gestão

Orientador:
José Carlos Ferreira Maia Neves
(Professor Associado)

Braga
Dezembro de 1997

*“Forecasting is the art of saying what will happen, and then explaining why
it didn’t!”*
Anonymous

*“I was intrigued by neural networks that could automatically learn a
pattern. I was very excited about finding these magical devices that, once
you discovered the right design, you could turn on and they would get
smarter.”*
David Shaw, 1997

*“Genetic Neural Networks. I believe that this is going to become the core
of connectionism.”*
Frederic Gruau, 1994

Agradecimentos

Esta dissertação vem no seguimento de um projecto desenvolvido para a Opção III da licenciatura de Engenharia de Sistemas e Informática. Este projecto foi desenvolvido conjuntamente com o meu colega e amigo Miguel Rocha, co-autor da ideia de aplicar redes neuronais à previsão de séries temporais, ao qual agradeço todo o seu trabalho, ideias e ajuda prestada. Na altura, não pensávamos que tal projecto pudesse ir longe. Na verdade, aquele trabalho acabou por se tornar uma base importante, a partir do qual surgiram algumas publicações, assim como esta dissertação.

Fico grato ao Professor José Carlos Ferreira Maia Neves por ter influenciado o meu interesse na área da Inteligência Artificial, pelos seus ensinamentos, aconselhamento bem como toda a confiança que depositou em mim.

Não quero deixar de agradecer a estimada contribuição dos colegas de laboratório, Dr. Rui Mendes e Engenheiro Filipe Santos, por toda ajuda prestada.

Finalmente, desejo agradecer à minha família, amigos e à Ângela, por todo o apoio e encorajamento.

Resumo

Hoje em dia assiste-se a uma mundialização da economia, originando um acréscimo da concorrência entre as organizações. Assim, é natural assumir que as organizações estão interessadas em obter previsões fundamentadas de certos tipos de indicadores, especialmente acerca daqueles que possam influenciar a tomada de decisões.

Uma das formas reside na *Previsão de Séries Temporais (PST)*, baseada nas observações anteriores da variável a prever. Os métodos convencionais de *PST* criam os seus modelos com base em componentes da série temporal, como a tendência ou a sazonalidade. Este tipo de abordagem é aplicável a séries bem comportadas, mas falha quando estas apresentam ruído ou componentes não lineares.

Nos últimos anos, os Algoritmos de Optimização, com base em processos de evolução natural das espécies, têm vindo a ser aplicados com sucesso em diferentes domínios. De entre estes, as *Redes Neurais Artificiais (RNAs)* e os *Algoritmos Genéticos (AGs)* assumiram-se como umas das mais populares e importantes ferramentas.

Assim, neste trabalho, pretende-se explorar o uso combinado de *RNAs* e *AGs*, como fundamentos de um sistema inteligente para a *PST*.

Abstract

Nowadays one assists to a globalization of the world economy, with an increasing competition between enterprises. Thus, it is natural to assume that organizations will be interested on obtaining well-grounded forecasts, specially for decision making.

One way is to use *Time Series Forecasting (TSF)*, forecast of a chronological ordered variable. Conventional *TSF* methods base their models on *Time Series (TS)* components such as *trend* or *seasonal* effects. These methods give accurate forecasts on well behaved *TS*, but present some drawbacks on *TS* with noise or some unknown nonlinear relations among the *TS* data.

In the last few years, Optimization Algorithms, based on natural evolutive processes, have been applied with success on wide areas. From all these, *Artificial Neural Networks (ANNs)* and *Genetic Algorithms (GAs)* have been assumed as ones of the most popular and important techniques.

Thus, in this thesis, one will try to explore the combination of *ANNs* and *GAs*, in order to get intelligent systems for *TSF*.

Índice

1	Introdução	1
1.1	Motivação	1
1.1.1	A Importância da Previsão	2
1.1.2	A Previsão de Séries Temporais	2
1.1.3	Paradigmas Sub-simbólicos	3
1.1.4	Redes Neurais Artificiais	3
1.1.5	Algoritmos Genéticos	4
1.1.6	Os Sistemas <i>GANN</i>	4
1.2	Objectivos	5
1.3	Organização	5
2	A Previsão de Séries Temporais	7
2.1	O Papel da Previsão nas Organizações	7
2.2	Métodos de Previsão	8
2.3	O Horizonte Temporal	8
2.4	Etapas da Previsão	8
2.5	Medida do Erro da Previsão	9
2.6	Modelos de Previsão	10
2.6.1	Modelos de Regressão	10
2.6.2	Modelos de Séries Temporais	10

2.7	Métodos de <i>PST</i>	11
2.7.1	O Método de Holt-Winters	11
2.7.2	A Metodologia ARIMA	12
2.7.3	A Selecção do Método Adequado	13
2.7.4	Uso de Redes Neurais	13
2.8	Decomposição de Séries Temporais	13
2.8.1	Tendência	14
2.8.2	Ciclo	14
2.8.3	Sazonalidade	15
2.8.4	Aleatoriedade	15
2.9	Coefficiente de Autocorrelação	16
2.10	Verificação da Aleatoriedade	16
2.11	Identificação da Tendência	17
2.12	Remoção da Tendência	18
2.13	Identificação da Sazonalidade	19
3	Redes Neurais	22
3.1	O Movimento Conexionista	22
3.2	O que é uma Rede Neuronal Artificial?	23
3.3	Estrutura de uma Rede Neuronal Artificial	24
3.4	Activação e Funções de Activação	26
3.5	Treino de Redes Neurais	27
3.6	Algoritmo <i>RPROP</i>	29
3.7	Pré-processamento dos Dados	30
3.8	Dados de Treino	32
3.9	<i>Overfitting</i>	32
3.9.1	Paragem Antecipada	32
3.9.2	Perda de Generalização	33

3.9.3	Progresso no Treino	34
3.9.4	Número Máximo de Iterações	35
4	Algoritmos Genéticos	36
4.1	A Computação Evolucionária	36
4.2	O que são Algoritmos Genéticos?	36
4.3	Estrutura de um Algoritmo Genético	38
4.4	Tamanho da População	39
4.5	Operadores Básicos	39
4.5.1	Seleccção	39
4.5.2	Cruzamento	40
4.5.3	Mutação	41
4.6	Critérios de Paragem	42
4.7	Codificação	42
4.8	Código <i>Gray</i>	43
4.9	Combinando Algoritmos Genéticos com Redes Neurais	44
4.10	Os Sistemas <i>GANN</i>	45
5	Estado da Arte	47
5.1	Uso de <i>RNAs</i> na <i>PST</i>	47
5.1.1	<i>RNAs Feedforward</i>	47
5.1.2	<i>RNAs</i> Recorrentes	49
5.1.3	Combinação de <i>RNAs</i> com Filtragem de Dados	49
5.2	Outras Técnicas Não Convencionais	50
5.3	Comparação entre Técnicas	50
5.4	Estado Actual	50
6	Descrição do Sistema Desenvolvido	51
6.1	Arquitectura das <i>RNAs</i>	51

6.2	Normalização dos Dados	53
6.3	O Sistema <i>GANN</i>	54
6.3.1	Factores que Influenciam a <i>PST</i>	54
6.3.2	Função de <i>Aptidão</i>	55
6.3.3	Estrutura do Sistema <i>GANN</i>	58
6.4	Arquitectura do Sistema	58
6.4.1	Integração entre o <i>C</i> e o <i>Prolog</i>	60
6.4.2	A Interface Gráfica <i>X-Windows</i>	62
6.5	Séries Utilizadas	64
6.6	Um Exemplo para o Sistema <i>GANN</i>	64
6.7	Resultados do Sistema <i>GANN</i>	66
6.7.1	Factores Óptimos	66
6.7.2	Comparação de Resultados	67
6.8	Filtragem de Dados	69
6.9	Janela Temporal Deslizante	69
6.10	O Sistema <i>GANNF</i>	70
6.11	Um Exemplo para o Sistema <i>GANNF</i>	72
6.12	Resultados do Sistema <i>GANNF</i>	73
7	Conclusões e Perspectivas de Trabalho Futuro	77
7.1	Avaliação do Trabalho Desenvolvido	77
7.2	Perspectivas de Trabalho Futuro	80
7.3	O Futuro da <i>PST</i>	82
	Bibliografia	84
	Apêndice: Artigos Publicados de Acordo com o Trabalho Desenvolvido nesta Dissertação	91

Lista de Figuras

2.1	Estrutura de uma <i>série temporal</i>	11
2.2	<i>Série temporal</i> não estacionária	14
2.3	<i>Série temporal</i> cíclica	15
2.4	<i>Série temporal</i> com factor estação	16
2.5	Autocorrelações de uma <i>série aleatória</i>	17
2.6	Autocorrelações de uma série <i>com e sem tendência</i>	18
2.7	Autocorrelações de uma série e das suas diferenças de primeira e segunda ordem	20
2.8	Autocorrelações de uma série com factor estação de 4	20
2.9	Autocorrelações de uma <i>série sazonal não estacionária</i>	21
3.1	Estrutura de uma <i>RNA</i> recorrente	25
3.2	Estrutura de uma <i>RNA feedforward</i>	26
3.3	Activação de uma célula	27
3.4	Mínimo local e global	29
3.5	Exemplo de perda de poder de generalização de uma <i>RNA</i> genérica	33
3.6	Exemplo de progresso no treino de uma <i>RNA</i> genérica	34
4.1	Estrutura de um Algoritmo Genético	38
4.2	Diferenças entre a operação de <i>selecção</i> baseada na <i>aptidão</i> e a operação de <i>selecção</i> baseada na <i>ordem</i>	40

4.3	Cruzamento de <i>um ponto</i>	41
4.4	Cruzamento de <i>dois pontos</i>	41
4.5	Cruzamento <i>uniforme</i>	42
4.6	<i>Mutação</i> no cromossoma de um indivíduo	42
4.7	Tempos de procura para as várias estratégias	45
6.1	Codificação da <i>RNA</i>	55
6.2	Estrutura do sistema <i>GANN</i>	59
6.3	Arquitectura funcional do sistema	60
6.4	A interface gráfica <i>X-Windows</i>	63
6.5	Valores anuais das manchas solares (série 1)	66
6.6	Evolução do MQE_{va} para a série 1	67
6.7	Previsões para a série 1	68
6.8	Períodos anteriores de uma série	70
6.9	Estrutura de um cromossoma no sistema <i>GANNF</i>	71
6.10	Valores mensais de passageiros de uma companhia aérea (série 2)	74
6.11	Previsões de curto e longo prazo para a série 2	75

Lista de Tabelas

2.1	Amostra de uma <i>série temporal</i> com primeiras e segundas diferenças	19
3.1	Funções de activação	27
4.1	Exemplos de código <i>gray</i>	43
6.1	Correlações entre os vários critérios e o erro para a previsão de curto prazo	57
6.2	Correlações entre os vários critérios e o erro para a previsão de longo prazo	58
6.3	Descrição das séries utilizadas	64
6.4	Características das séries utilizadas	65
6.5	A rede óptima para cada série da Tabela 6.3	69
6.6	Fórmulas de recorrência para os modelos ARIMA	70
6.7	Comparação entre as previsões de curto prazo obtidas pelo sistema <i>GANN</i> e as obtidas por outros métodos (usando o <i>MQE</i> como medida de erro)	71
6.8	Comparação entre as previsões de longo prazo obtidas pelo sistema <i>GANN</i> e as obtidas por outros métodos (usando o <i>MQE</i> como medida de erro)	72
6.9	Conjunto de janelas temporais	73
6.10	A rede óptima para cada uma das séries estudadas	74

6.11	Comparação entre as previsões de curto prazo obtidas pelo sistema <i>GANNF</i> e as obtidas por outros métodos (usando o <i>MQE</i> como medida de erro)	76
6.12	Comparação entre as previsões de longo prazo obtidas pelo sistema <i>GANNF</i> e as obtidas por outros métodos (usando o <i>MQE</i> como medida de erro)	76

Glossário

<i>AG</i>	Algoritmo Genético
<i>AID</i>	Análise Inteligente de Dados
<i>CIA</i>	Critério da Informação Akaike
<i>CIB</i>	Critério da Informação Bayesian
<i>GANN</i>	<i>Genetic Algorithms and Neural Networks</i>
<i>IA</i>	Inteligência Artificial
<i>MQE</i>	Média do Quadrado dos Erros
<i>PST</i>	Previsão de Séries Temporais
<i>RNA</i>	Rede Neuronal Artificial
<i>RPROP</i>	<i>Resilient backPROPagation</i>
<i>SQE</i>	Soma do Quadrado dos Erros

Capítulo 1

Introdução

1.1 Motivação

Este trabalho situa-se na área científica da *Inteligência Artificial (IA)*, ou mais propriamente no domínio da *Análise Inteligente de Dados (AID)*.

A *IA* atende à construção e manipulação de entidades inteligentes, tendo como último objectivo a *clonagem* do ser humano. Apesar de ser uma disciplina recente (foi formalmente iniciada em 1956), tem induzido desenvolvimentos significativos em diferentes áreas de conhecimento, nomeadamente na compreensão da língua natural, no raciocínio lógico-matemático, na teoria dos jogos ou em sistemas de apoio à decisão. Mais e mais cientistas se movem em direcção à *IA*, onde encontram as ferramentas para sistematizar e automatizar os seus actos. De um modo similar, também os que trabalham na *IA* têm vindo a aplicar as suas metodologias e ferramentas em outras áreas do empreendimento intelectual. Neste sentido, a *IA* afigura-se como uma verdadeira disciplina universal [RusNor95].

A *Análise Inteligente de Dados (AID)*, com base na *IA* e na *Estatística*, preocupa-se em desenvolver sistemas inteligentes ou de apoio à decisão, que auxiliem o cientista ou o simples utilizador na análise de dados. Três factores, em particular, contribuíram para o interesse na *AID*. Em primeiro lugar, o desenvolvimento da indústria de semi-condutores permitiu disponibilizar artefactos com grande poder computacional e a um baixo custo. Em segundo lugar, há que ter em conta o ritmo explosivo em que a acumulação de

dados se processa. Esta acumulação de dados é uma consequência directa da diminuição dos custos de armazenagem, de parceria com um elevado nível de automação e uso de novos dispositivos electrónicos de recolha, em que pontuam os sensores. O terceiro factor projecta-se através da panóplia de ferramentas desenvolvidas pelas comunidades da *IA*, da *Investigação Operacional* e/ou da *Física*, nomeadamente as extensões à programação em lógica, a lógica *fuzzy*, o raciocínio probabilístico, as árvores de decisão, as redes neuronais e os algoritmos genéticos. Desta forma foi possível a introdução da *AID* em áreas em que tradicionalmente os problemas eram abordados através do palpite e da intuição [DecFoc95].

1.1.1 A Importância da Previsão

Hoje em dia assiste-se a uma mundialização da economia, e por conseguinte a um acréscimo de concorrência entre as organizações cujas estruturas, processos e tecnologias, se optimizados, lhes darão vantagens competitivas [Cor95]. Contudo, inovar num ambiente de incerteza pode ser desastroso, pelo que é natural pressupor que as organizações estão interessadas em obter previsões fundamentadas sobre o seu futuro [HanRei89].

Este desejo de compreender o passado e prever o futuro impulsiona a procura de leis que expliquem o comportamento de dados fenómenos ou acontecimentos. Se são conhecidas equações determinísticas que os explicam, então é possível efectuar previsões, desde que conhecida a sua génese. No entanto, na ausência de regras que definam o comportamento de um sistema, procura-se determinar o seu comportamento futuro a partir de observações do passado [WeiGer94].

1.1.2 A Previsão de Séries Temporais

Uma das técnicas em uso socorre-se da *Previsão de Séries Temporais (PST)*, que se baseia em observações da variável em estudo. A *PST* tem como objectivo prever o comportamento de sistemas e não o seu funcionamento, que nos surge na forma de *alisamento exponencial*, *decomposição* ou *análise de regressão* [MakMcG78][WheMak80]. Este tipo de abordagem é eficaz quando aplicada a séries bem comportadas, tornando-se inadequada, contudo, quando estas apresentam ruído ou têm componentes não lineares, uma situação

deveras comum [WeiGer94] (eg. tem-se a variação dos preços do mercado bolsista ou o número de manchas solares [McC93]).

1.1.3 Paradigmas Sub-simbólicos

Nos últimos anos, algoritmos de optimização com base em processos de selecção natural, entre outros, têm vindo a ser aplicados no domínio da *IA*. O estudo do comportamento de fenómenos como os processos de fundição (*annealing* do inglês), o sistema nervoso central ou a evolução biológica, deu origem a ferramentas como o *Simulated Annealing*, as *Redes Neurais* e a *Computação Evolucionária*.

De entre estes, as *Redes Neurais Artificiais (RNAs)* e os *Algoritmos Genéticos (AGs)*, impuseram-se como paradigmas em diferentes áreas do conhecimento, revelando-se como umas das mais populares e efectivas ferramentas [RusNor95][SanNev95].

1.1.4 Redes Neurais Artificiais

As *Redes Neurais Artificiais (RNAs)* são modelos conexionistas com a capacidade de aprender a partir do conhecimento de situações passadas, dando resposta a novas situações, especialmente em problemas com uma forte componente de ruído e/ou informação incompleta [Pre94]. Esta é a razão principal das vantagens comparativas que alcançaram em áreas tão distintas como os sistemas periciais, a filtragem de dados, a visão por computador ou o planeamento [Gal93][NevCor97b].

Uma nova área científica parece emergir do uso de *RNAs* na *AID* como máquinas de aprendizagem, face ao enorme conjunto de aplicações para a extracção de conhecimento, como os sistemas periciais, os sistemas de apoio à decisão e de diagnóstico, de análise de sensibilidade ou a *PST*, envolvendo um variado conjunto de domínios, que se estendem desde o sector bancário, financeiro, químico, militar até ao de engenharia [DecFoc95]. Trata-se da procura de descrições matemáticas ou lógico-matemáticas, eventualmente de natureza complexa, sobre padrões e regularidades em conjuntos de dados [GarNev91][NevGar91].

Em particular, a emergência das *RNAs*, a partir da década de oitenta,

veio dar um novo fôlego ao campo da *PST*, multiplicando as suas capacidades de previsão em séries complexas, como as dos mercados financeiros [Azo95].

Porém, as *RNAs* não podem ser vistas como a solução universal para todos os problemas [RusNor95]. De facto, uma das dificuldades que resulta do uso de *RNAs* para a resolução de problemas reside no tempo dispendido com a procura do melhor tipo de rede ou topologia. Uma outra dificuldade tem a ver com a melhor forma de alimentar os dados a uma rede, num processo que se designa de *pré-processamento*, com recurso a técnicas de *normalização* e *filtragem* dos dados.

1.1.5 Algoritmos Genéticos

Os *Algoritmos Genéticos (AGs)* são exemplos de máquinas de aprendizagem, que se baseiam em processos de selecção natural. Os *AGs* assumem-se como procedimentos de optimização, passíveis de serem aplicados à panóplia de parâmetros que fazem a solução de um problema, gerando variantes que, através de competição, podem convergir para uma melhor solução. Assim, embora os *AGs* usem processos estocásticos, os resultados obtidos são claramente superiores aos conseguidos por procura aleatória [Gol89].

Os *AGs* são normalmente utilizados em problemas de natureza combinatoria, onde a procura de soluções exige esforços computacionais demasiado severos [Müh93].

1.1.6 Os Sistemas *GANN*

Uma via que tem vindo a ser usada para torneir este último problema, ou seja, a minimização do esforço computacional, centra-se no uso de *AGs* para procurar os parâmetros óptimos da rede neuronal, no espaço vectorial que nos dá as soluções possíveis para o problema em equação [RusNor95]. Através desta associação de *AGs* com *RNAs*, chamada de sistema *GANN*, procura-se maximizar as vantagens de cada método [Koe94]. Neste tipo de sistemas, a aprendizagem da rede neuronal guia a procura evolucionária da melhor rede. Estudos empíricos evidenciam que os sistemas *GANN* ultrapassam quer as redes neuronais, quer os algoritmos genéticos na procura de uma solução [Kit90][Whi93].

1.2 Objectivos

Pretende-se nesta dissertação:

- fazer um apanhado do que foi feito em termos de investigação nesta área; i.e., no uso de alternativas não convencionais para a *PST*, com especial relevo para as *RNAs*;
- estudar a melhor forma de utilizar *RNAs* para a *PST*, em termos de topologia da rede, do seu treino, das funções de activação, da normalização de dados e das técnicas de filtragem;
- explorar a combinação de *AGs* com *RNAs* tendo em vista a sua aplicação na *PST*;
- desenvolver um sistema computacional que integre o pré-processamento de dados, os *AGs*, e as *RNAs* de um modo racional, permitindo realizar de forma automática a *PST*;
- analisar os resultados obtidos pelo sistema para séries temporais reais de diferentes tipos;
- comparar os resultados obtidos com os conseguidos a partir de técnicas convencionais.

1.3 Organização

A dissertação encontra-se estruturada em sete capítulos, passíveis de um agrupamento; i.e., no que vai do Capítulo 2 até ao Capítulo 5 estabelece-se a ligação do trabalho aqui realizado com outros trabalhos já feitos, assim como se dá uma perspectiva do estado da investigação na área. Nos Capítulos 6 e 7 descreve-se o trabalho que foi desenvolvido.

O Capítulo 2 trata a *PST*. É evidenciada a importância desta técnica para as organizações e descrevem-se sucintamente os métodos de previsão mais utilizados.

As *RNAs* são introduzidas no Capítulo 3, expondo-se as razões porque têm sido objecto de estudo por parte da comunidade científica. É descrito

o que é uma *RNA*, quais as suas propriedades, que tipos de *RNAs* existem, como se processa o treino, e em que consiste o pré-processamento dos dados. Finalmente, é apresentado o problema do fenómeno de *overfitting*, assim como técnicas para o evitar.

No Capítulo 4 é feita menção à computação evolucionária, aos *AGs*, às vantagens comparativas sobre outros métodos e ao seu modo de funcionamento. São também apresentadas formas de combinar os *AGs* com as *RNAs*.

O Capítulo 5 apresenta o estado da arte na área: o que tem sido feito e que técnicas não convencionais têm sido utilizadas.

O sistema desenvolvido é descrito no Capítulo 6, sendo aí apresentadas as estratégias e opções tomadas, as experiências efectuadas, a arquitectura do sistema, as séries utilizadas e os resultados obtidos. Também é descrito um sistema de filtragem de dados, sendo analisadas as suas implicações no desempenho do sistema.

Por último, no Capítulo 7, são apresentadas as conclusões, sendo apontadas direcções para trabalho futuro.

Capítulo 2

A Previsão de Séries Temporais

2.1 O Papel da Previsão nas Organizações

Nos últimos anos uma certa ênfase tem sido colocada no melhoramento dos processos de tomada de decisão, e isto quer no que respeita ao mundo empresarial, quer ao universo da política; i.e., quando os métodos de gestão de há vinte ou trinta anos atrás eram baseados nos sentidos e intuição do gestor ou político, no presente tal arquétipo não é mais aplicável. No seu lugar apareceram outras técnicas de tomada de decisão, nomeadamente as oriundas de disciplinas como a *Investigação Operacional*, a *Estatística*, a *Informática* ou *Princípios de Desenho Organizacional* [WheMak80].

As empresas, não raras vezes a contragosto, têm que fazer opções. Ora, um ponto importante da tomada de decisões é a capacidade de prever as circunstâncias que as rodeiam, pelo que é natural que estejam interessadas em obter previsões fundamentadas, da evolução no tempo, das variáveis que determinam o sucesso ou insucesso de determinada opção; i.e, procura-se, deste modo, calcular os riscos que estão associados à tomada de decisões [HanRei89].

O recurso à previsão é comum nas organizações, em áreas como o *marketing* ou produção; porém, é nas áreas financeira e contabilística que o seu uso tem sido mais intenso. As empresas não devem, no entanto, encarar a previsão como uma profecia, mas sim como a melhor maneira de extrapolar as relações existentes nos dados, para fazer previsões. Os erros de previsão

são, como é evidente, inevitáveis, tendo-se como objectivo minimiza-los.

2.2 Métodos de Previsão

Como consequência natural do processo de tomada de decisões, as técnicas para a extrapolação de conhecimento têm sido estudadas exaustivamente, e métodos mais objectivos e fiáveis foram sendo desenvolvidos. Estes métodos variam conforme o grau de sofisticação e fins a que se destinam, sendo divididos em duas classes: métodos *qualitativos* e métodos *quantitativos*. Os primeiros são baseados na intuição, e utilizados quando não se dispõe de informação sobre o passado. Os segundos, mais comuns, passam pelo processamento da informação disponível [HanRei89].

Os métodos quantitativos são mais eficazes e a sua aceitação generalizada é uma consequência próxima do uso maciço de computadores. Estes são úteis não só pelo seu poder de cálculo mas também pela sua capacidade de armazenagem de dados [WheMak80].

2.3 O Horizonte Temporal

O período de tempo durante o qual se pretende que prevaleça uma dada tomada de decisão afecta, naturalmente, o processo de selecção do método de previsão. As previsões são feitas normalmente com um ou dois períodos de avanço. Os horizontes temporais podem ser classificados de *curto*, *médio* e *longo prazo*. Os primeiros, até três períodos, são muito usados para decisões de gestão corrente (eg. na gestão de *stocks*), enquanto que as previsões de médio prazo (de três meses a um ano) e longo prazo (mais de dois anos) são usadas em planeamento (eg. em medicina preventiva).

2.4 Etapas da Previsão

Todos os métodos de previsão partem do princípio de que as experiências do passado serão usadas no futuro. Assume-se, assim, que as condições que condicionaram o passado serão válidas no futuro. A previsão envolve quatro

etapas: a *recolha de dados históricos*, a *condensação ou redução dos dados*, a *construção do modelo de previsão*, e a *extrapolação a partir desse modelo* [HanRei89].

A primeira etapa exige um particular cuidado na leitura dos dados do problema em equação de forma a se evitarem distorções. A etapa seguinte, *condensação*, consiste em determinar os dados relevantes para o problema em equação (eg. pode-se estar apenas interessado em prever o volume de vendas de um produto a partir da abertura de loja nova, situação esta em que os dados de vendas anteriores não são relevantes). A tarefa seguinte, a *construção do modelo de previsão*, envolve um ajustamento dos dados a um modelo de previsão, de forma a minimizar o erro. É importante que o modelo seja o mais simples possível, de modo a que tenha uma boa aceitação por parte dos utilizadores. Uma vez definido o modelo segue-se a última etapa, extrapolar do modelo a previsão para períodos futuros.

Neste trabalho apenas são consideradas as duas últimas etapas do processo, partindo-se do princípio que a *recolha e condensação* dos dados foi realizada de forma correcta.

2.5 Medida do Erro da Previsão

Como já foi referido, as previsões estão sujeitas a erros, que se procuram minimizar. Porém, para que tal seja possível, é necessário o recurso a uma métrica. Sendo o *erro* numa previsão dado por

$$e_i = x_i - P_i$$

em que x_i denota o valor da variável em estudo, e P_i o valor previsto para a mesma, à medida que o método de previsão é aplicado, uma sequência de valores de erro é obtida [HanRei89]. Para se avaliar a eficácia da previsão, torna-se necessário analisar esta série, existindo várias formas de o fazer. Uma das mais utilizadas (e que será usada neste trabalho) utiliza a *Média* (M) da potência de expoente dois do erro, ou seja

$$M = \frac{\sum_{i=1}^n e_i^2}{n}$$

em que n é o número de previsões.

2.6 Modelos de Previsão

Existem dois tipos base de modelos de previsão que se complementam e que partem de diferentes premissas, a saber: os *Modelos de Regressão* e os *Modelos de Séries Temporais* [HanRei89].

2.6.1 Modelos de Regressão

Os *Modelos de Regressão* (ou causais) assumem que existe entre a variável a prever, y , e uma ou mais das variáveis independentes, x_1, \dots, x_n , uma relação de causa-efeito. Como primeira tarefa há então que descobrir essa relação, através da análise dos valores da variável dependente ao longo do tempo; i.e., construído o modelo e, sendo conhecidos os valores para as variáveis independentes, o valor da variável dependente é-nos dado por

$$y = f(x_1, \dots, x_n)$$

Este tipo de modelos é bastante usado na tomada de decisões, de que são exemplos a mudança de preço ou o incremento de produção de um produto.

2.6.2 Modelos de Séries Temporais

Uma *série temporal* dá-nos a sequência de observações sobre uma variável em particular, ordenadas cronologicamente [TanFis91] (Figura 2.1). O propósito da *Previsão de Séries Temporais (PST)* consiste em detectar padrões nas observações do passado de modo a extrapolar para o futuro [PapGot93].

Os modelos de *PST* tratam o sistema como se de uma caixa negra se tratasse, não tendo como objectivo determinar os factores que afectam o seu comportamento. Existem, porventura, algumas razões para esta postura. A primeira deve-se ao facto de o sistema não ser perfeitamente determinado, sendo difícil identificar as relações de causa-efeito em jogo. A segunda reside no facto de se desejar prever o que acontece e não como acontece; i.e., existem *séries temporais* reais impossíveis de axiomatizar, tangíveis à previsão (eg. a previsão da ocorrência de manchas solares ou a previsão do comportamento do mercado de acções [McC93]).

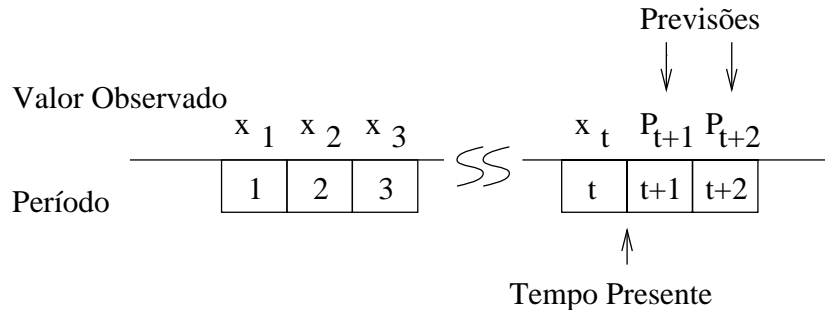


Figura 2.1: Estrutura de uma *série temporal*

2.7 Métodos de *PST*

Ao longo das últimas décadas vários tipos de métodos quantitativos para a *PST* foram desenvolvidos, nomeadamente o *Alisamento Exponencial*, os *Métodos de Decomposição* ou a *Análise de Regressão* [WheMak80]. De entre estes, importa destacar o método de Holt-Winters e a metodologia ARIMA, que serão descritos com pormenor seguidamente.

2.7.1 O Método de Holt-Winters

Um dos métodos de *Alisamento Exponencial* mais em voga é o *Holt-Winters*, que se centra em determinadas características da série, como seja a existência de factores de *tendência* ou *sazonalidade* [HanRei89]. É vantajoso por ser simples de usar, requerer pouco esforço computacional, e ser bastante eficaz em previsões de curto prazo, em particular com séries sazonais. O modelo base obedece ao clausulado:

$$\begin{aligned}
 P_t &= \alpha \frac{x_t}{S_{t-k}} + (1 - \alpha)(P_{t-1} + T_{t-1}) \\
 T_t &= \beta(P_t - P_{t-1}) + (1 - \beta)T_{t-1} \\
 S_t &= \gamma \frac{x_t}{P_t} + (1 - \gamma)S_{t-k}
 \end{aligned}$$

em que P_t representa o *Alisamento Exponencial* para o período t (equivalente à previsão para o mesmo período), T_t a estimativa para a *tendência*, S_t a estimativa *sazonal*, x_t a *série temporal* e k o *factor sazonal*. Previsões a longo prazo são efectuadas de acordo com a fórmula recorrente

$$P_{t+l} = (P_t + lT_t)S_{t-k+l}$$

para uma previsão de l períodos à frente de t . Este método exige que certos parâmetros sejam inicialmente avaliados, nomeadamente as estimativas *sazonais*. Subsiste, contudo, um problema, ou seja, o cálculo dos parâmetros α , β e γ . Ora, ou se usam os valores de $\alpha = 0,2$, $\beta = 0,2$ e $\gamma = 0,6$, ou então procura-se estimar estes parâmetros, através de um processo de *trial and error*, em que a minimização do erro é o objectivo perseguido [HanRei89].

2.7.2 A Metodologia ARIMA

Outro método é o de Box-Jenkins, chamado *ARIMA*, sendo um dos mais populares para a *PST* [BoxJen76]. Trata-se de um método sofisticado e generalista, que se baseia na síntese de padrões a partir de dados históricos, exigindo a caracterização formal do sistema em equação (obtenção de um modelo), estimação dos parâmetros a ele associados e a sua posterior validação. Tem como vantagem principal o facto de ser bastante preciso em previsões de curto prazo. No entanto torna-se difícil conciliar os parâmetros com o modelo inicialmente criado à medida que novos dados são adquiridos; i.e., o modelo tem de ser periodicamente revisto ou, pior ainda, um novo modelo criado. Por outro lado, a construção de um modelo exige o uso de um especialista, sendo também mais exigente em termos computacionais que os métodos de *alisamento*.

Os modelos são criados a partir de uma combinação linear finita de alguns valores passados da série (componente AR do modelo ARIMA). Conforme o tipo de modelo, poderá existir, isoladamente ou em conjunto com o componente anterior, uma combinação linear finita dos erros passados (componente MA do modelo ARIMA); i.e., diferenças entre os valores previstos pelo modelo ARIMA e os valores da série. Assim, o modelo geral, conhecido por $ARMA(M, N)$, obedece ao clausulado [WeiGer94]

$$x_t = \sum_{m=1}^M a_m x_{t-m} + \sum_{n=0}^N b_n e_{t-n}$$

em que x_t representa a *série temporal* e e_t o erro.

2.7.3 A Selecção do Método Adequado

A escolha do método (ideal) de previsão para uma dada série não é um processo simples. McNees [McN82] é de opinião que não existe um único modelo de previsão que se possa assumir *primus inter pares*, no que é secundado por Zarnowitz [Zar84]. Em particular, chegou-se à conclusão de que um aumento de complexidade e sofisticação estatística, não implica necessariamente um melhoramento na eficácia da previsão. Isto indicia que podem existir situações onde métodos simples podem ser aplicados com sucesso, e a baixo custo; i.e., a escolha do método irá depender da situação específica a ser estudada, do horizonte temporal a prever, e da organização envolvida. Whellwright e Makridakis [WheMak80] aconselham as organizações, que porventura não queiram ser pioneiras, à utilização de métodos de *Alisamento Exponencial*, em particular com séries de periodicidade mensal.

2.7.4 Uso de Redes Neurais

Os modelos referidos até ao momento, modelos convencionais, requerem que um conjunto bem definido de factores seja conhecido à partida. Porém, não raras vezes, estes factores são difíceis de ser quantificados, pelo que uma abordagem alternativa e promissora, que tem vindo a ser desenvolvida, passa pelo uso de redes neurais, e por duas boas razões [PapGot93]: em primeiro lugar, as redes neurais conseguem extrapolar padrões a partir dos dados existentes, sendo bastante imunes a ruído; em segundo lugar, as redes neurais adaptam o seu comportamento à medida que novos dados são introduzidos, sem necessidade de alterar a sua estrutura.

2.8 Decomposição de Séries Temporais

O processo de identificação dos factores que influenciam os valores de uma *série temporal* é denominado de *decomposição* [HanRei89], tendo sido identificados quatro factores principais que influenciam de forma sustentada o seu comportamento [WheMak80]: a *Tendência (T)*, o *Ciclo (C)*, a *Sazonalidade (S)* e a *Aletoricidade (A)*, ou seja

$$Y = T \times C \times S \times A$$

onde Y representa o valor da *série temporal*.

2.8.1 Tendência

A *Tendência* dá-nos uma medida do crescimento ou declínio dos valores de uma série. Uma série que possua este tipo de comportamento é chamada de *não estacionária*. Na Figura 2.2 representa-se uma *série não estacionária* típica.

Várias forças se conjugam para que esta componente seja comum em situações que têm a ver com o estudo da inflação, das mudanças tecnológicas, do crescimento da população ou dos aumentos de produtividade.

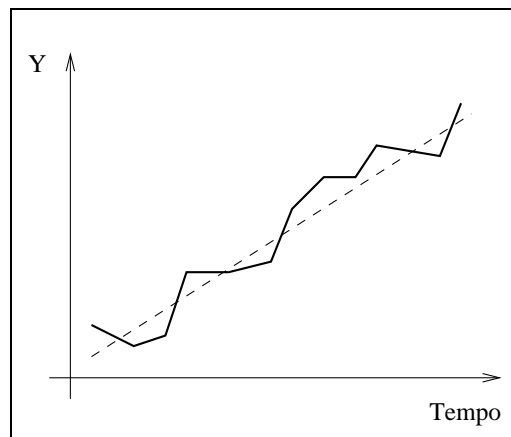


Figura 2.2: *Série temporal* não estacionária

2.8.2 Ciclo

Este padrão é tangível em situações em que os dados apresentam oscilações periódicas, de que são exemplos as séries construídas com base em dados que descrevem a procura de produtos agrícolas, a venda de automóveis ou a evolução de indicadores macro-económicos (Figura 2.3).

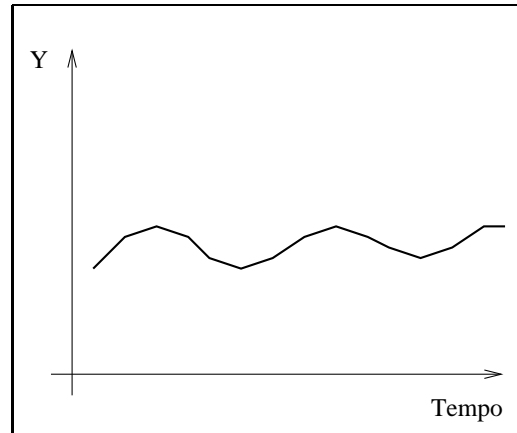


Figura 2.3: *Série temporal* cíclica

2.8.3 Sazonalidade

Este tipo de flutuação é típico de séries periódicas. Estes períodos, mais curtos do que o *ciclo* (normalmente de três a cinco anos), medem-se em semanas ou meses. O padrão repete-se, ao longo do tempo, após cada k períodos, sendo k o *factor sazonal*.

Esta situação é comum, por exemplo, na comercialização de produtos que dependem do clima ou simplesmente de produtos que se vendem em determinados períodos do ano, como por exemplo no Natal. A Figura 2.4 representa uma *série sazonal* com um período de sazonalidade de quatro, correspondendo às quatro estações do ano.

2.8.4 Aleatoriedade

Esta componente representa a parte da série que não é previsível, ou seja, a parte que não é explicada, o ruído. É devido à existência desta componente que as previsões não são correctas a 100%.

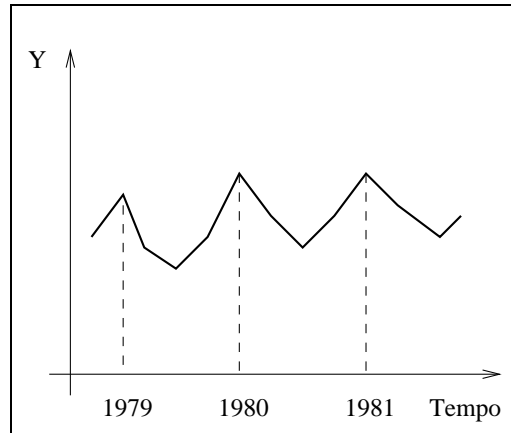


Figura 2.4: *Série temporal* com factor estação

2.9 Coeficiente de Autocorrelação

O *coeficiente de autocorrelação* é um indicador que ajuda na análise de *séries temporais*. É definido como a correlação entre a série e ela própria (daí o termo auto), deslocada de k períodos de tempo [BoxJen76]:

$$r_k = \frac{\sum_{t=1}^{n-k} (x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^n (x_t - \bar{x})^2}$$

onde x_1, x_2, \dots, x_n representam os valores da *série temporal* e \bar{x} a sua média.

2.10 Verificação da Aleatoriedade

As autocorrelações são aplicadas na determinação da aleatoriedade de uma série. Uma série diz-se *aleatória* se os seus termos obedecem à decomposição: uma média constante μ e um erro e_t , o qual é independente de período para período de tempo, ou seja [MakMcG78]

$$x_t = \mu + e_t$$

Uma forma prática de se detectar uma *série aleatória* passa pelo exame das autocorrelações para os diversos deslocamentos k . Teoricamente todas as autocorrelações de uma *série aleatória* deveriam ser nulas. Isto pressuponha,

todavia, uma amostra de tamanho infinito. Porém, se se construírem amostras de n números aleatórios, e se considerarmos k como a autocorrelação da população de onde são retiradas essas amostras, prova-se que os coeficientes das amostras seguem uma distribuição normal de média ρ_k e variância $\frac{1}{\sqrt{n}}$.

Ora esta informação é usada na criação de um teste estatístico para se determinar se uma dada série é ou não aleatória; i.e., devido às características da distribuição normal, pode-se dizer que uma série é *aleatória* se todas as suas autocorrelações estão no intervalo (Figura 2.5) [MakMcG78]

$$-1,96 * \frac{1}{\sqrt{n}} \leq r_k \leq 1,96 * \frac{1}{\sqrt{n}}$$

para um grau de confiança de 95%.

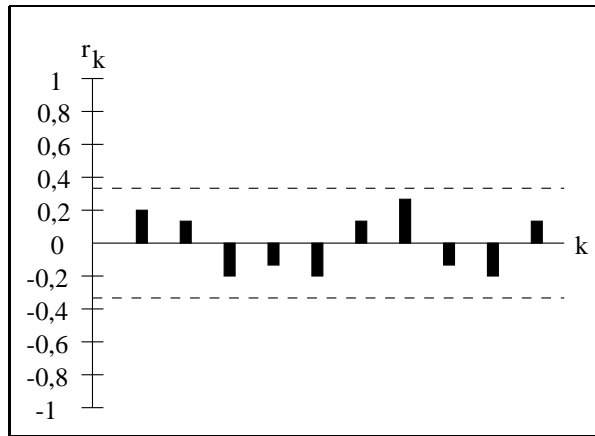


Figura 2.5: Autocorrelações de uma *série aleatória*

2.11 Identificação da Tendência

Uma análise do gráfico de uma série é, por vezes, o suficiente para determinar se a série é ou não *estacionária*. De facto, para uma série *não estacionária*, o gráfico que dá os valores da variável a prever em função do tempo, apresenta sempre com um declive acentuado (Figura 2.2).

Uma outra via, mais eficaz, para determinar se uma série é ou não *estacionária*, consiste numa análise das autocorrelações da série [MakMcG78].

Assim, uma *série estacionária* caracteriza-se pelo facto de as suas autocorrelações possuírem um padrão distinto, com valores próximos de zero a partir de k igual a três ou quatro, enquanto que nas séries *não estacionárias* as autocorrelações são diferentes de zero (Figura 2.6).

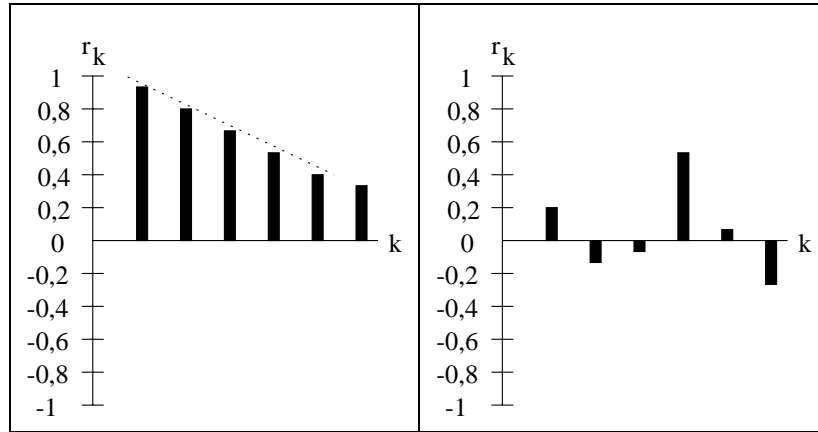


Figura 2.6: Autocorrelações de uma série *com* e *sem* tendência

2.12 Remoção da Tendência

A existência de *tendência* numa série leva a que este padrão se sobreponha a todos os restantes padrões da série. Além disso, muitos dos métodos existentes para a *PST* têm como condição essencial a *estacionariedade* da série. Devido a estas limitações, foi desenvolvido um método para eliminar da série a sua *tendência*, a que se chama de *diferenciação* [MakMcG78]; diferenciar uma série passa por se criar uma nova série, cujos elementos são dados pela diferença entre valores da série inicial, tomados de períodos consecutivos, ou seja

$$x'_t = x_t - x_{t-1}$$

A nova série terá $n - 1$ elementos e será *estacionária* se a *tendência* na série original for de primeira ordem (linear). Se tal não acontecer, repete-se o processo, agora tomando como série original a série diferenciada, obtendo-se assim uma diferença de segunda ordem da série inicial. Na Figura 2.7 dão-se as autocorrelações para os dados da Tabela 2.1 antes da série ser diferenciada, assim como após uma primeira e segunda diferenciações. O

processo poderia continuar até a estacionariedade ser atingida; todavia, em séries reais, raramente é necessário passar-se para além da segunda diferença.

Tabela 2.1: Amostra de uma *série temporal* com primeiras e segundas diferenças

Período	Série	Primeiras Diferenças	Segundas Diferenças
1	2,44	-	-
2	5,3	2,86	-
3	8,97	3,67	0,81
4	13,88	4,91	1,24
5	19,58	5,7	0,79
6	26,99	7,41	1,71
7	35,95	8,96	1,55
8	45,86	9,91	0,95
9	55,7	9,84	-0,07
10	67,36	11,66	1,82
11	79,63	12,27	0,61
12	92,13	12,5	0,23

2.13 Identificação da Sazonalidade

A *sazonalidade* define-se como o padrão que se repete a si próprio em intervalos fixos de tempo [HanRei89]. É comum em séries de periodicidade mensal, de que são exemplos as vendas de produtos sazonais, sejam estes gelados ou guarda-chuvas. Nestes casos é de esperar uma forte tendência sazonal, a qual se reflecte no valor das autocorrelações; i.e., se uma série se apresenta com um valor da autocorrelação que seja expressivo para k períodos, então é de esperar valores significativos para as autocorrelações com deslocamentos de k períodos. Por exemplo, para o caso das séries referidas em epígrafe, será de esperar um valor elevado para a autocorrelação com um deslocamento de doze períodos.

Para *séries estacionárias*, o padrão de *sazonalidade* pode então ser encontrado, conhecidos que sejam as autocorrelações (as mais significativas) para valores de k superiores a três [MakMcG78]. Na Figura 2.8 indicam-se as

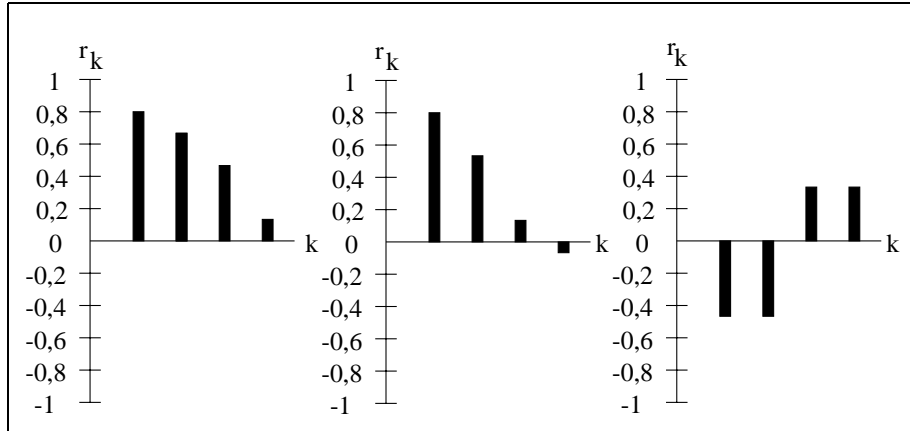


Figura 2.7: Autocorrelações de uma série e das suas diferenças de primeira e segunda ordem

autocorrelações de uma série com um padrão de sazonalidade trimestral, em que é notório o valor elevado da autocorrelação para $k = 4$ e seus múltiplos. Isto representa um comportamento típico das *séries sazonais*.

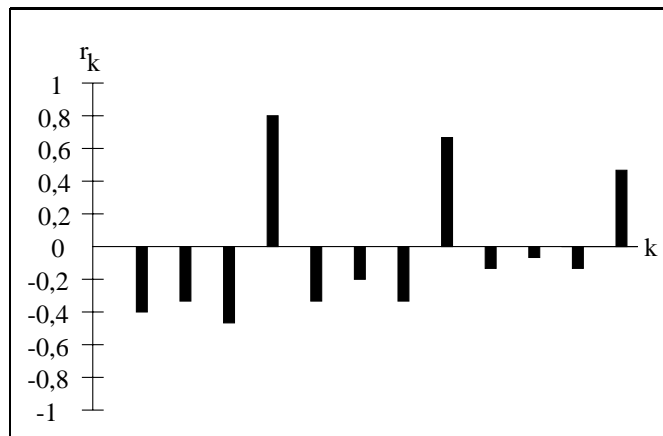


Figura 2.8: Autocorrelações de uma série com factor estação de 4

Quando a série *não é estacionária* dois tipos de padrões vão sobrepor-se, podendo levar a que a identificação do padrão sazonal se torne mais complexa (Figura 2.9). Se a tendência for muito forte e a sazonalidade for fraca, pode mesmo tornar-se impossível esta identificação.

Para contornar este problema, uma das melhores soluções é diferenciar os

dados até se obter uma *série estacionária*, procurando-se depois nesta algum tipo de padrão sazonal.

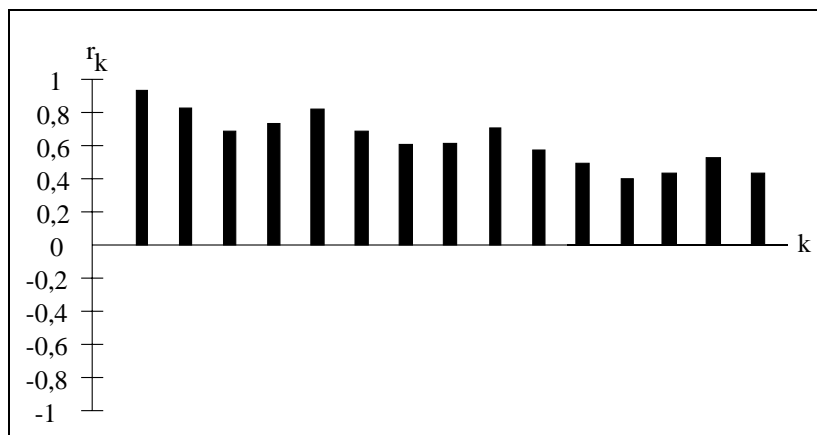


Figura 2.9: Autocorrelações de uma *série sazonal não estacionária*

Capítulo 3

Redes Neurais

3.1 O Movimento Conexionista

Uma *rede neuronal* é o arquétipo tipo utilizado por biólogos e afins, na tentativa de simular e modelar os processos cognitivos da mente. Para distinguir entre o modelo natural (cérebro) e o artificial (rede), é utilizado o termo *Rede Neuronal Artificial (RNA)* para este último.

O grande interesse no uso de RNAs desenvolveu-se a partir do meio da década de oitenta, quando se re-inventou o algoritmo de treino de *backpropagation* [RusNor95]. Desde então as RNAs são objecto de atenção por parte da comunidade científica e usadas na prática. Este ressurgimento também se deve ao facto das RNAs se afirmarem como ferramentas poderosas no reconhecimento de padrões, ultrapassando outras técnicas, quer em fiabilidade, quer em velocidade de processamento [Azo95].

Este movimento conexionista surgiu dentro da *IA* como uma reacção aos modelos simbólicos, considerados demasiado rígidos e especializados; i.e., esperava-se deste novo paradigma a criação de sistemas de aprendizagem mais flexíveis e universais. Contudo, depois de um grande entusiasmo inicial com o modelo conexionista, caminha-se para uma integração de ambos os paradigmas [Min90][CorNev96a].

3.2 O que é uma Rede Neuronal Artificial?

Uma *RNA* implementa um procedimento que relaciona de algum modo informação à entrada com informação (desejada) à saída. As tarefas associadas às *RNAs* podem, por conseguinte, ser divididas em diferentes tipos, nomeadamente de :

- **Classificação**; i.e., para determinar a que categoria pertence um dado padrão de entrada;
- **Associação**; i.e., para actuar como uma memória endereçável, que associa a um dado valor de saída um dado valor de entrada;
- **Codificação**; i.e., para compactar uma entrada, pela produção de uma saída com uma dimensão reduzida. A tarefa oposta, descompactação, também se reveste de interesse.

Por outro lado, devido a uma grande estrutura de conectividade, as *RNAs* têm propriedades únicas, que não se encontram nas técnicas de processamento ditas tradicionais, a saber [TanFis91][Azo95]:

- robustez na manipulação de diferentes tipos de dados;
- degradação suave (permitindo processar ruído ou informação incompleta);
- capacidade de generalização e aprendizagem (responder a novas situações com base nos dados existentes);
- flexibilidade (com um grande domínio de aplicabilidade);
- modelação não linear dos dados;
- adaptação face a alterações.

Uma outra propriedade das *RNAs* é a transparência, uma vez que podem ser vistas como uma *caixa negra*, em que a saída é uma função desconhecida [Gal93]. Em princípio, as redes neuronais podem imitar qualquer função computável; i.e., qualquer função que um normal computador digital pode

calcular, especialmente aquelas que possam ser representadas por uma correspondência entre vectores espaciais. As *RNAs* parecem atingir o seu melhor em problemas de classificação ou associação, em que é necessário um certo grau de tolerância para certos erros, em que há muita informação, e aos quais não se podem aplicar outros procedimentos [Pre94].

Hoje em dia procuram-se não só melhores redes como redes mais eficientes, melhores algoritmos de treino e melhores funções de transferência [ShaRam96]. Por outro lado, espera-se que a aplicação de *RNAs* a outras áreas do conhecimento se generalize, seja à Economia, ao Processamento de Sinal, à Visão por Computador, à Robótica, à Automação ou aos Sistemas Periciais, para além da Estatística [KemMar92][FarCha95]. Neste domínio temos a estimação não paramétrica, a aprendizagem por agentes económicos e a modelação de séries temporais [ShaRam96][CorNev95].

3.3 Estrutura de uma Rede Neuronal Artificial

Uma *Rede Neuronal Artificial (RNA)* (*modelo conexionista*) é, de uma forma simples, um conjunto de unidades computacionais, chamadas de *células* (ou *nodos*), com uma alguma memória local, e um conjunto de *conexões* (ou *ligações*) unindo essas células [Gal93] (Figura 3.1). Estas conexões têm a si associados valores numéricos aos quais se dá o nome de *pesos*. De tempos em tempos cada unidade examina as suas entradas e a partir destas, calcula um valor numérico chamado de *activação*. Este valor de activação é então passado ao longo das conexões, que levam a outras células.

Existem inúmeros tipos de arquitecturas de *RNAs* ou *topologias*, cada um com as suas próprias potencialidades ou senãos. A grande distinção é feita entre redes *recorrentes* e *feedforward*. Numa rede *recorrente* as conexões podem ser feitas entre quaisquer nodos (Figura 3.1), enquanto que numa rede *feedforward* as conexões são unidireccionais, não existindo realimentação, sendo assim formada por camadas (Figura 3.2) [RusNor95]. As últimas são as redes mais vulgarizadas e mais utilizadas [TanFis91].

Numa *RNA* podem-se encontrar três tipos de células: *células de entrada*, cujo valor de saída é fixado externamente; *células de saída*, que devolvem a saída da rede; e finalmente, as *células intermédias*, que não são de entrada

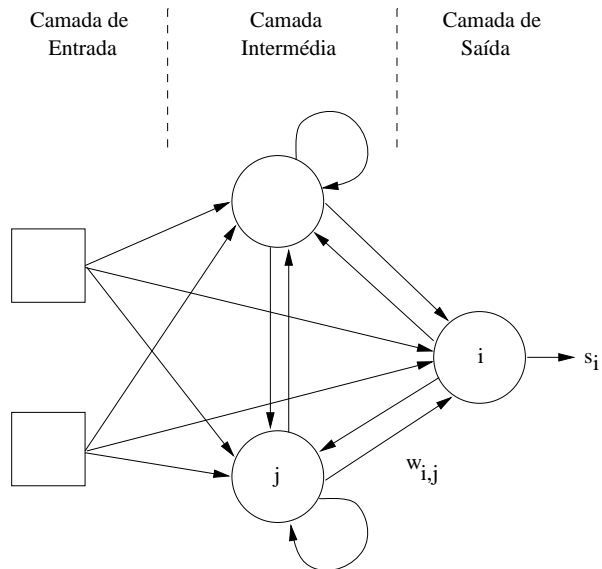


Figura 3.1: Estrutura de uma RNA recorrente

nem de saída [Gal93]. As redes com uma ou mais camadas intermédias são chamadas de redes *multi-camada*. Está provado que uma RNA com uma camada intermédia pode representar uma qualquer função contínua, e que com duas camadas intermédias até funções descontínuas podem ser representadas [RusNor95]. As RNAs com apenas uma camada intermédia respondem pela maioria das aplicações, enquanto que RNAs com mais de duas camadas intermédias raramente são usadas, devido a problemas de desempenho.

Uma RNA *completamente interligada* caracteriza-se por estabelecer todas as ligações possíveis entre os nodos de duas camadas. Como exemplos deste tipo de rede temos as redes das Figuras 3.1 e 3.2. Por vezes existem ligações directas entre as entradas e os neurónios de saída (Figura 3.1). Estas conexões designam-se por *atalhos* [Pre94]. Por convenção existe ainda uma célula especial, cuja saída é sempre 1 (um), e que está ligada a todas as células; o peso desta ligação é chamado de *bias* [Gal93].

É importante mencionar-se que parece não existir nenhum procedimento para a obtenção da melhor topologia de rede, quando apenas se considera o número de entradas e saídas desta, pelo que o número de casos para treino e a complexidade do problema em questão serão determinantes para a sua caracterização.

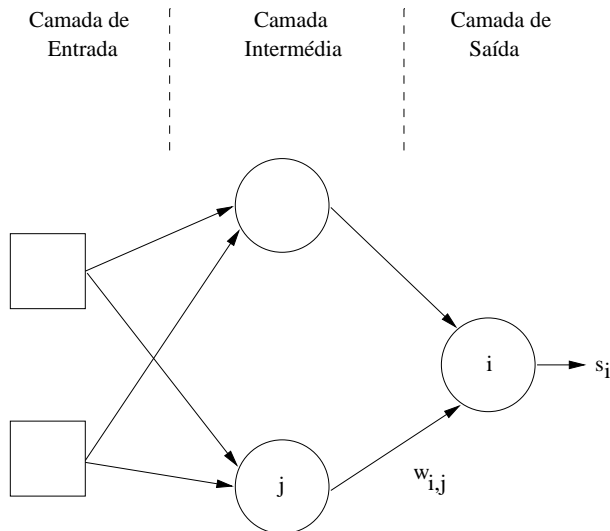


Figura 3.2: Estrutura de uma *RNA feedforward*

3.4 Activação e Funções de Activação

Como já se viu, cada célula calcula a sua activação (ou energia de activação), a partir dos valores das saídas das células que se lhe ligam e dos pesos associados às respectivas conexões. Representando a energia de activação da célula i por s_i , e da célula j por s_j , segue-se que (Figura 3.3):

$$u_i = \sum_{j=0}^n w_{i,j} * s_j$$

$$s_i = f(u_i)$$

onde $w_{i,j}$ representa o peso associado à ligação entre o nodo j e o nodo i .

A função f é chamada de *função de activação*. Os valores da energia de activação podem ser discretos (tomando normalmente os valores -1 , 0 ou 1) ou contínuos. No caso dos valores da energia de activação serem contínuos, f é normalmente uma função não linear, limitada e diferenciável por troços. Neste trabalho serão utilizadas as funções da Tabela 3.1 [Azo95].

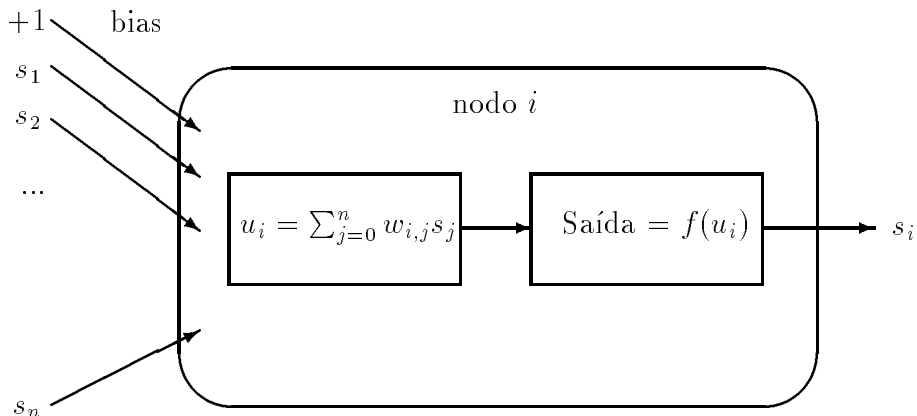


Figura 3.3: Activação de uma célula

Tabela 3.1: Funções de activação

Nome	Função $f(x)$	Contradomínio
<i>linear</i>	x	$] -\infty, +\infty[$
<i>sigmoid</i>	$\frac{1}{1+\epsilon(-x)}$	$[0, 1]$
<i>sigmoid1</i>	$\frac{2}{1+\epsilon(-x)} - 1$	$[-1, 1]$
<i>sigmoid2</i>	$\frac{x}{1+ x }$	$[-1, 1]$
<i>tanh</i>	$\tanh(x)$	$[-1, 1]$
<i>sin</i>	$\sin(x \bmod 2\pi)$	$[-1, 1]$
<i>cos</i>	$\cos(x \bmod 2\pi)$	$[-1, 1]$
<i>gaussian</i>	$e(\frac{-x^2}{2})$	$[-1, 1]$

3.5 Treino de Redes Neurais

Para as *RNAs* aprenderem com a experiência, há que as submeter a um processo chamado de *treino*, para o que existem inúmeros algoritmos de treino, distribuindo-se estes, normalmente, por duas classes: *supervisionados* e *não supervisionados* [Gal93].

No primeiro caso existe um *professor* que fornece um conjunto de *casos de treino*. Um *caso de treino* consiste num vector de entrada e no correspondente vector de saída. A aprendizagem passa então pela minimização do erro da saída, o que é feito através do ajustamento dos pesos das conexões segundo uma determinada regra. No segundo caso apenas existe um conjunto de

entradas, pelo que o algoritmo de treino tenta agrupar os dados a partir de propriedades ou padrões por estes apresentados.

Presentemente, a aprendizagem supervisionada é a mais utilizada na área das *RNAs* [Rie94]. Dentro dos algoritmos supervisionados, o mais popular e mais usado é o algoritmo de *backpropagation* ou seus derivados [Gal93]. Trata-se de um procedimento iterativo, de aprendizagem lenta, que se aplica apenas a redes *feedforward*, usualmente com funções de activação *sigmoidais*. O processo de convergência baseia-se em métodos de gradiente descendente e usa-se a média dos quadrados dos erros como medida do erro da rede, ou seja

$$\Delta w(t) = -\epsilon * \nabla E(t)$$

ou para um único peso

$$\Delta w_{ij}(t) = -\epsilon * \frac{\delta E}{\delta w_{ij}}(t)$$

em que t representa a ordem da iteração e ϵ a taxa de aprendizagem [Rie94]. As derivadas parciais são calculadas segundo o clausulado

$$\frac{\delta E}{\delta w_{ij}} = \frac{\delta E}{\delta s_i} \frac{\delta s_i}{w_{ij}}$$

onde

$$\frac{\delta s_i}{w_{ij}} = f'(u_i) s_j$$

Para se obter $\frac{\delta E}{\delta s_i}$, ou seja a influência da saída s_i do nodo i no erro global E , é necessário atender ao tipo de nodo

$$\frac{\delta E}{\delta s_i} = \begin{cases} -(t_i - s_i) & , \text{ se } i \text{ é um nodo de saída;} \\ \sum_{k \in succ(i)} \frac{\delta E}{\delta s_k} f'(u_k) w_{ki} & , \text{ caso contrário.} \end{cases}$$

onde $succ(i)$ representa o conjunto dos nodos k da camada com que o nodo i se relaciona (ou estabelece ligações). Assim, a informação dada pelo gradiente é sucessivamente propagada desde a camada de saída até à camada de entrada, daí o nome de *backpropagation* (retropropagação).

Antes de se iniciar o treino de uma rede, há que proceder à escolha dos valores iniciais dos pesos associados às ligações entre nodos. Estes deverão ser pequenos e gerados aleatoriamente e, em princípio, pertencendo ao intervalo

$[-\frac{2}{z}; \frac{2}{z}]$, para uma célula com z entradas [Gal93]. Pode-se então partir para o treino da rede, começando-se por seleccionar um caso para treino, calcula-se o gradiente e ajustam-se os pesos. Uma iteração termina quando todos os casos disponíveis tiverem sido considerados. O processo termina quando as mudanças nos pesos e na média dos quadrados dos erros forem insignificantes. De notar que o algoritmo pode convergir para um mínimo local (Figura 3.4); na prática, porém, constata-se que quando se parte de um número elevado de casos de treino, esta questão não se coloca, ou então não se assume como um problema sério.

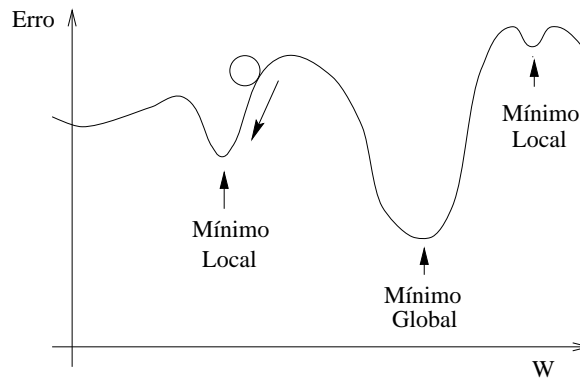


Figura 3.4: Mínimo local e global

Diversas funções de erro podem ser utilizadas no treino de redes neurais, sendo a função mais utilizada a que mede o quadrado dos erros, ou seja

$$E = \sum_i (t_i - s_i)^2$$

para o valor desejado à saída t_i e valor de saída do neurónio s_i .

Esta medida, ao dar um erro por cada caso de treino, produz um grande volume de informação, pelo que normalmente é usada a soma ou a média desta medida; no que se segue será utilizada a *Média do Quadrado dos Erros (MQE)*.

3.6 Algoritmo *RPROP*

O nome do algoritmo, *RPROP*, vem do inglês “*Resilient backPROPagation*” [Rie94], e trata-se de uma variante sofisticada do algoritmo de *backpropaga-*

tion. Ao contrário deste último, o algoritmo *RPROP* utiliza uma aprendizagem de adaptação local; i.e., em cada iteração, o algoritmo primeiro calcula a direcção do gradiente para o conjunto de casos de treino, e só depois efectua a actualização dos pesos das ligações entre neurónios [RieBra93].

A regra de cálculo é dada por

$$\Delta_{ij}(t) = \begin{cases} \eta^+ * \Delta_{ij}(t-1) & , \text{ se } \frac{\delta E}{\delta w_{ij}}(t-1) * \frac{\delta E}{\delta w_{ij}}(t) > 0; \\ \eta^- * \Delta_{ij}(t-1) & , \text{ se } \frac{\delta E}{\delta w_{ij}}(t-1) * \frac{\delta E}{\delta w_{ij}}(t) < 0; \\ \Delta_{ij}(t-1) & , \text{ caso contrário.} \end{cases}$$

para $0 < \eta^- < 1 < \eta^+$, sendo t a iteração. Baseados em considerações de ordem teórica e empírica, estes factores foram fixados em $\eta^+ = 1,2$ e $\eta^- = 0,5$. Os pesos das ligações entre neurónios podem então ser alterados de acordo com a relação

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}(t) & , \text{ se } \frac{\delta E}{\delta w_{ij}}(t) > 0; \\ +\Delta_{ij}(t) & , \text{ se } \frac{\delta E}{\delta w_{ij}}(t) < 0; \\ 0 & , \text{ caso contrário.} \end{cases}$$

Estudos comparativos mostraram que este algoritmo converge mais rapidamente que outros algoritmos do género [Rie94]. Além disso, dependendo de apenas 2 (dois) parâmetros, Δ_0 e Δ_{max} , cuja escolha de valores não é muito crítica, leva-nos a concluir que se está perante um algoritmo deveras robusto; i.e., regra geral, não é necessária nenhuma alteração aos valores aconselhados para Δ_0 e Δ_{max} ($\Delta_0 = 0,1$ e $\Delta_{max} = 50,0$) [RieBra93].

3.7 Pré-processamento dos Dados

Em muitas situações torna-se útil pré-processar os dados de um problema antes de estes serem alimentados à rede. A ideia é ajustar os dados de alguma forma tal que a rede os possa tratar de forma eficiente; i.e., proceder à sua normalização [Azo95]. São várias as fases em que se desdobra uma operação de pré-processamento:

- **verificação da integridade dos dados**, i.e., validar os dados e verificar a existência de erros;

- **representação dos dados**, i.e., escolher a forma de conversão de dados (eg. usando uma codificação binária ou real, usando uma escala linear ou logarítmica);
- **normalização dos dados**, i.e., escalonar os dados para um dado domínio;
- **filtragem dos dados**, i.e., passar à utilização de técnicas de filtragem para a eliminação de ruído.

A normalização de dados torna-se necessária devido a duas razões básicas. Em primeiro lugar, há que atender ao contradomínio das funções de activação (Tabela 3.1). Por exemplo, a função *sigmoid* limita a sua saída ao intervalo $[0, 1]$, pelo que é impossível que a rede tenha como saídas valores fora deste intervalo. A segunda razão prende-se com o processo de assimilação de informação pela rede. Mesmo que os dados de treino a considerar estejam dentro do intervalo requerido pela função de activação, pode acontecer que os mesmos não se distribuam ao longo do domínio da referida função. Neste caso, a operação de normalização permitirá lançar os dados pelo domínio, aumentando a capacidade de assimilação de informação da rede.

Ora, uma das formas de a realizar é através de um escalonamento linear. Aqui, estes antes de serem alimentados à rede, são transformados de acordo com a equação

$$y = \frac{(x - \min)(B - A)}{\max - \min} + A$$

para um escalonamento no domínio $[A, B]$, sendo *max* e *min* os valores máximo e mínimo do conjunto dos dados de treino. Os valores de saída da rede passam pela operação inversa, sendo obtidos pela equação:

$$x = (y - A) \frac{(\max - \min)}{B - A} + \min$$

Esta forma de normalização tem a vantagem de efectuar a distribuição dos dados de treino por todo o intervalo $[A, B]$, a partir de um procedimento bastante simples.

3.8 Dados de Treino

Na aprendizagem de *RNAs* os casos disponíveis devem ser divididos em, pelo menos, duas partes: uma parte na qual o treino é feito, chamada de *casos de treino*, e outra parte que permite medir a eficácia da rede resultante, chamada de *casos de teste*. É óbvio que nenhum dos casos de teste poderá ser utilizado durante o treino, sob pena de estarmos perante uma falácia.

3.9 *Overfitting*

Uma situação que ocorre frequentemente, principalmente quando não existe o número de *casos de treino* adequado, é que após um certo número de iterações, o desempenho da rede começa a piorar para os *casos de teste*, apesar de continuar a melhorar para os *casos de treino*. A este fenómeno chama-se de *overfitting* [GemDou92], e ocorre porque um treino demasiado prolongado faz com que a rede memorize todo e qualquer padrão nos *casos de treino*, deixando, assim, de ter poder de generalização. Este problema pode ser minimizado a partir da utilização de métodos de regularização, que tentam diminuir a complexidade da rede de modo a que esta seja incapaz de memorizar particularidades. A outra maneira de evitar o *overfitting* passa pelo uso de métodos de paragem antecipada do treino da rede [FinZim93][LanHin90].

3.9.1 Paragem Antecipada

Na paragem antecipada os casos disponíveis são divididos em *casos de treino* e *casos de validação*.

Os *casos de validação* são usados em um *pseudo* teste, durante o treino da rede, para medir a qualidade desta. Assim, em cada iteração, a rede é treinada apenas com os *casos de treino*, e é medido o erro com os *casos de validação*. A paragem do treino faz-se quando o erro é mínimo, o que é um tanto problemático dado que a função erro pode apresentar mínimos locais. De notar que os *casos de teste* são na mesma utilizados para medir o desempenho da rede após o treino.

Um outro aspecto que importa aqui realçar tem a ver com a escolha do número de casos de cada tipo. Em princípio, quanto maior for o conjunto

de *casos de treino* melhor se comporta a rede, dado que assimila mais informação. Mas, por outro lado, um conjunto de *casos de validação* demasiado pequeno poderá parar o treino numa altura indesejada; i.e., a rede não será convenientemente testada.

Neste processo há a considerar três critérios de paragem: por *perda de generalização*, por *falta de progresso no treino*, e por *se atingir o número máximo de iterações* [Pre94].

3.9.2 Perda de Generalização

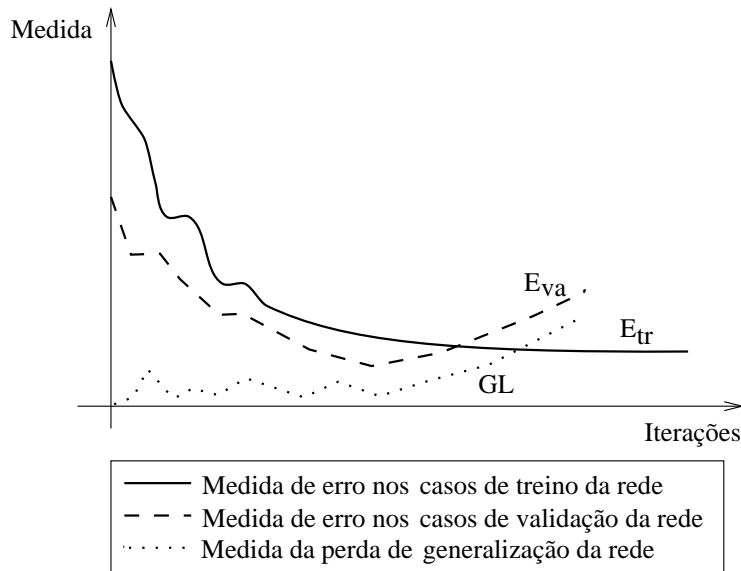


Figura 3.5: Exemplo de perda de poder de generalização de uma *RNA* genérica

A *perda de generalização* ocorre sempre que há uma inversão de sinal nos valores da derivada da função de erro para os *casos de validação*, passando estes de negativos a positivos (Figura 3.5). A função de avaliação toma a forma

$$GL(t) = 100 * \left(\frac{E_{va}(t)}{E_{opt}(t)} - 1 \right)$$

onde E_{opt} e E_{va} denotam, respectivamente, o valor mínimo do erro (neste caso *MQE*) e os erros medidos para os *casos de validação* até à iteração t , ou

seja

$$E_{opt}(t) = \min_{t' \leq t} E_{va}(t')$$

Uma grande perda de poder de generalização por parte da rede é uma boa razão para se parar o treino, aplicando-se o critério de paragem

Parar após primeira iteração t em que $GL(t) > \alpha$

onde α denota a perda de poder de generalização aconselhável. Este critério é medido ao fim de cada k iterações, chamadas de *faixa de treino* com um comprimento k .

3.9.3 Progresso no Treino

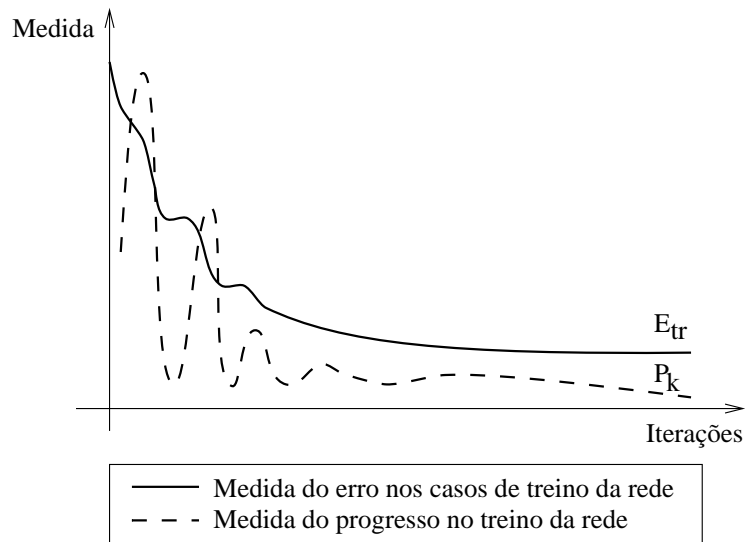


Figura 3.6: Exemplo de progresso no treino de uma *RNA* genérica

Esta medida relaciona a média do erro nos *casos de treino* para uma dada *faixa de treino*, com o mínimo dos erros medidos para a mesma faixa. A função de avaliação toma a forma

$$P_k(t) = 1000 * \left(\frac{\sum_{t' \in t-k+1 \dots t} E_{tr}(t')}{k * \min_{t' \in t-k+1 \dots t} E_{tr}(t')} - 1 \right)$$

em que $E_{tr}(t)$ é o *MQE* dos *casos de treino*, para a iteração t .

O progresso no treino é elevado nas fases instáveis, onde o erro nos casos de treino sobe em vez de descer. No entanto tende para zero a longo prazo, a não ser que o treino se torne oscilante (Figura 3.6).

O treino é parado se $P_k(t) < \beta$, em que β é uma medida de erro de estado estacionário.

3.9.4 Número Máximo de Iterações

Este critério é aplicado quando os outros critérios de paragem do treino da rede falham, sendo dado na forma

Parar quando $t > \gamma$, para a iteração t

em que γ denota a última iteração.

Capítulo 4

Algoritmos Genéticos

4.1 A Computação Evolucionária

Existe todo um conjunto de estratégias de resolução de problemas que foram influenciadas por processos de evolução das espécies, e em particular pelas descobertas de Darwin¹. A famosa *Teoria da Evolução das Espécies*² baseia-se num processo de selecção natural; i.e., indivíduos que herdaram determinadas características dos progenitores podem sofrer mutações, em que os mais capazes são os que tendem a sobreviver [RusNor95]. Os algoritmos evolucionários, baseados no modelo da selecção natural, foram propostos para a resolução de problemas de optimização nos anos sessenta. Desde então o tema central da investigação em *Algoritmos Genéticos* tem sido nos domínios da solidez e do balanceamento entre a eficiência e a eficácia [Gol89][MenNev95].

4.2 O que são Algoritmos Genéticos?

Os *Algoritmos Genéticos (AGs)* são modelos de máquinas de aprendizagem que usam a metáfora da evolução das espécies, numa perspectiva de simulação desses processos, com o fim último de ajudar à resolução de problemas; i.e.,

¹Charles Darwin (1809-1882). Naturalista inglês, célebre pela descoberta da selecção natural.

²Descrita na obra *Sobre a Origem das Espécies por Meio da Selecção Natural*, por Charles Darwin, 1859.

têm como objectivo otimizar um conjunto de parâmetros que, instanciados, nos dão a solução do(s) problema(s).

Em princípio, os *AGs* podem modelar qualquer função que seja computacionalmente representável [Müh93]. Todavia, os *AGs* são ultrapassados em termos de desempenho, em situações em que os procedimentos de resolução de problemas se encontram completamente sistematizados; i.e., algoritmos especializados em determinados domínios do conhecimento geralmente ultrapassam, em eficácia, os *AGs*.

Existe um certo conjunto de problemas de natureza combinatória, denominados de *NP-completos*, onde o domínio das soluções possíveis é tão vasto que a sua apreciação se torna computacionalmente incomportável [RusNor95].

Ora os *AGs* apresentam certas diferenças em relação a esses métodos de procura, nomeadamente:

- trabalham com uma codificação dos parâmetros do problema e não com os próprios parâmetros;
- partem para a procura de uma solução com base num conjunto de possíveis soluções (ou população) e não a partir de uma única solução;
- usam a informação que apenas provém de uma função objectivo e não informação auxiliar;
- usam regras probabilísticas de transição de estado.

Estas diferenças, tomadas em conjunto, contribuem para que os *AGs* encontrem soluções onde outros métodos parecem falhar; i.e., existe um certo conjunto de problemas de natureza combinatória, denominados de *NP-completos*, onde o domínio das soluções possíveis é tão vasto que a sua apreciação se torna computacionalmente incomportável, e que pode definir o campo por excelência para a utilização dos *AGs* [RusNor95][Gol89].

Os *AGs* têm vindo, entretanto, a ser aplicados à Biologia, à Medicina, à Engenharia e à Gestão, com resultados deveras prometedores [PicTom95][Gol89].

4.3 Estrutura de um Algoritmo Genético

Parte-se de um conjunto de *indivíduos*, soluções possíveis para um dado problema, chamado de *população*. A informação que disponibiliza, e que atende aos valores dos parâmetros do problema em equação, é representada por *cromossomas*, de um modo em tudo análogo à estrutura vigente no DNA. Um cromossoma é, por sua vez, composto por um conjunto de *genes* (caracteres). Um valor possível para um *gene* é designado por *alelo*. A qualidade de cada solução (cromossoma) é medida por uma função chamada de *aptidão* [Gol89].

À semelhança do que se passa no mundo real, os indivíduos de uma *população* são sujeitos a uma série de operações, tais como as de *cruzamento*, *mutação* e *selecção*. Como resultado deste processo, uma nova geração de indivíduos é criada. O processo repete-se, então, durante várias gerações, até que um dado objectivo seja atingido [Müh93].

De notar que a evolução, regra geral, não é um processo linear. Na realidade, os processos naturais de evolução passam, não raras vezes, por recuos e avanços, enquanto que os indivíduos de uma população competem pelos recursos disponíveis, o que significa que não há garantia de que a geração seguinte seja melhor do que a anterior, embora se espere que tal suceda a longo prazo.

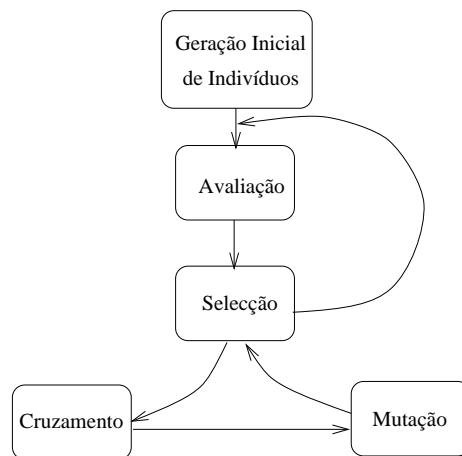


Figura 4.1: Estrutura de um Algoritmo Genético

A Figura 4.1 mostra a estrutura do *AG* [Koe94]. A primeira *população* é gerada aleatoriamente. Um ciclo, chamado de *geração*, inicia-se pela ava-

liação dos indivíduos da população. Uma nova *população* é criada a partir da utilização dos operadores de *cruzamento* e de *mutação*, sendo depois avaliada. A operação de *selecção* é então usada para seleccionar indivíduos de ambas as populações (i.e., inicial e final). A população resultante passa para o ciclo seguinte e o processo repete-se até que uma(s) condição(ões) de paragem seja(m) satisfeita(s).

4.4 Tamanho da População

O tamanho da *população* afecta quer a qualidade da solução final quer o tempo de processamento. Uma *população* pequena é pobre em termos de diversidade dos seus elementos constituintes, pelo que as soluções geradas tendem a não ser as melhores. Uma *população* com muitos indivíduos tem uma maior probabilidade de produzir melhores resultados, embora à custa de um maior esforço computacional.

4.5 Operadores Básicos

4.5.1 Selecção

Pela utilização deste operador, que pode ser aplicado a uma (para o cruzamento) ou a duas populações (para a escolha da geração seguinte), é possível obter os indivíduos mais capazes. Há ainda a mencionar que este processo não é determinístico mas antes estocástico, o que se justifica pela necessidade de se garantir uma grande diversidade de indivíduos em cada população [Gol89].

Perante os princípios que estão subjacentes ao funcionamento da *roleta* e aceitando-se que os mesmos regem a operação de *selecção*, a probabilidade de um indivíduo ser escolhido é proporcional à sua *aptidão* para com a população a que pertence (Figura 4.2) [Koe94]. Embora este método seja bom à partida, degrada-se geração após geração, quando as diferenças de *aptidão* entre os indivíduos que constituem a população são ínfimas. Uma alternativa é dada pelo uso da operação de *selecção baseada na ordem* [Whi89], onde a probabilidade de escolha de um indivíduo depende do seu posicionamento dentro da população. Assim, a probabilidade de *selecção* do indivíduo i é

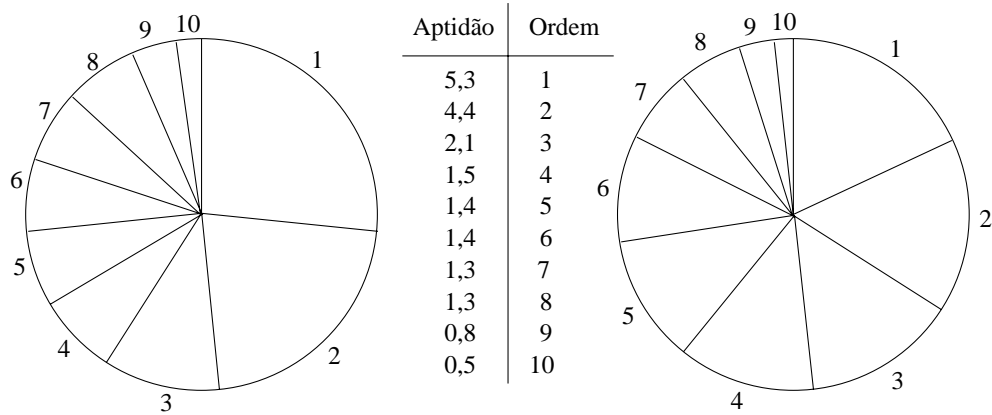


Figura 4.2: Diferenças entre a operação de *selecção* baseada na *aptidão* e a operação de *selecção* baseada na *ordem*

dada por

$$p(i) = \frac{(n + 1) - ordem(i)}{\sum_{j=1}^n j}$$

onde n representa o tamanho da população e $ordem(i)$ é uma função de ordem; i.e., dá o posicionamento de cada indivíduo na população a que pertence (Figura 4.2).

4.5.2 Cruzamento

Ao cruzarem-se, dois indivíduos, os *progenitores*, escolhidos na população, através de uma operação de *selecção*, geram um novo indivíduo (*descendente*). Isto é feito a partir de material genético dos cromossomas dos progenitores, que uma vez combinado, produz o cromossoma de um novo indivíduo ou descendente.

Consideram-se, em princípio, três possibilidades de cruzamento: cruzamento de *um e dois pontos*, e cruzamento *uniforme* [Gol89]. No primeiro caso apenas um ponto, escolhido aleatoriamente, é usado para partir o cromossoma (Figura 4.3). No segundo caso (Figura 4.4) são usados dois pontos, e no terceiro caso cada carácter dos cromossomas dos progenitores tem uma probabilidade real de passagem para o descendente (Figura 4.5).

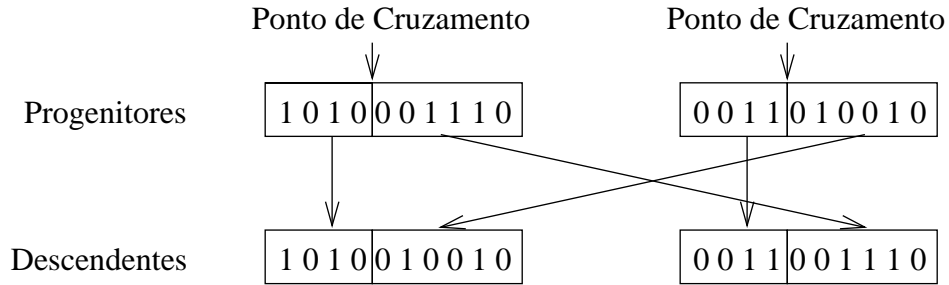


Figura 4.3: Cruzamento de *um ponto*

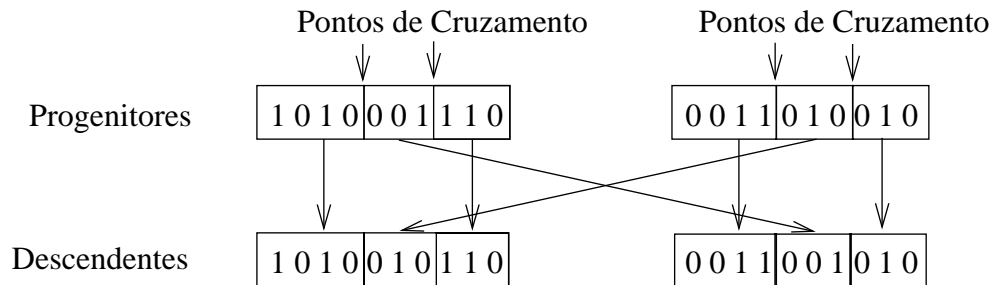


Figura 4.4: Cruzamento de *dois pontos*

4.5.3 Mutação

A *mutação* assume-se como uma operação chave na obtenção de uma solução, embora o seu impacto no processo ainda seja uma questão em aberto [Müh93]. A frequência com que este operador é aplicado é chamada de *taxa de mutação*, que é normalmente baixa para não tornar o processo puramente aleatório. Para uma representação em binário tem-se que (Figura 4.6):

$$\begin{aligned} 0 &\rightarrow \text{Mutação} \rightarrow 1 \\ 1 &\rightarrow \text{Mutação} \rightarrow 0 \end{aligned}$$

De notar que, embora o algoritmo genético use processos estocásticos, o resultado é claramente não aleatório.

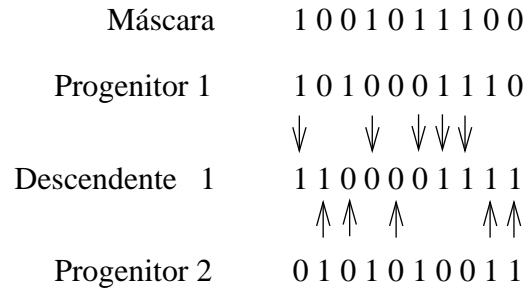


Figura 4.5: Cruzamento *uniforme*

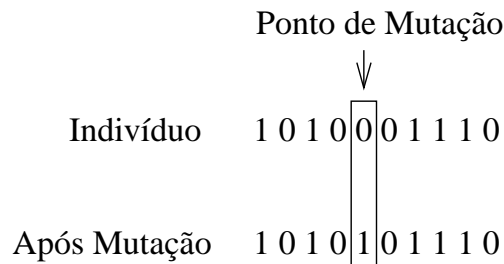


Figura 4.6: *Mutação* no cromossoma de um indivíduo

4.6 Critérios de Paragem

A paragem do *AG* depende do problema em causa e do esforço computacional a ser exigido. É necessário, assim, definir qual a qualidade da solução que se pretende, face ao tempo e aos recursos disponíveis.

4.7 Codificação

O processo de transformação da informação disponível em entidades transaccionáveis (os cromossomas), designado de *codificação*, é uma operação crucial para uma boa resolução dos problemas utilizando *AGs* [Gol89]. A mesma informação pode estar codificada de várias formas de modo a obter-se um espaço de soluções que seja contínuo, o que facilitando a operação de *cruzamento*, se assume como um requerimento básico para um bom funcionamento deste tipo de algoritmos. Algum cuidado há que se ter na apresentação da codificação, isto no caso de se trabalhar com 2 (dois) ou mais bits de informação

(o que normalmente é o caso). Considere-se, por exemplo, o cromossoma em que 2 (dois) dos atributos codificados são centrais para a compreensão do problema em equação. Suponha-se, por outro lado, que esta codificação não é adjacente. Ora, nesta situação torna-se óbvio que a hipótese de cruzar as codificações é mínima, o que poderá influenciar de forma negativa a obtenção de soluções para o problema em causa.

4.8 Código *Gray*

À primeira vista, a codificação binária simples parece ser a melhor forma de codificar em base 2 (dois) quaisquer dos parâmetros do problema. Todavia, nesta representação, números consecutivos podem apresentar diferenças em mais de um bit, ou seja, a *distância de Hamming*³ não é constante; esta é dada pelo número de bits diferentes presentes na codificação de dois números adjacentes. Ora esta propriedade dificulta a acção benéfica do operador de *mutação* dado que, não raras vezes, uma mutação em 1 (um) bit apenas altera significativamente o valor de um parâmetro.

Para colmatar esta lacuna foi desenvolvido o chamado *código gray*, uma representação binária de inteiros, na qual a *distância de Hamming* toma o valor constante de 1 (um) para qualquer n ; i.e., para um incremento (ou decremento) de um número de uma unidade, apenas um bit se altera na sua representação binária, particularidade esta que dá pelo nome de *propriedade de adjacência* [Koe94].

Tabela 4.1: Exemplos de código *gray*

Número	0	1	2	3	4	5	6	7
Código binário	000	001	010	011	100	101	110	111
Código <i>gray</i>	000	001	011	010	110	111	101	100

A Tabela 4.1 dá-nos a correspondência entre os inteiros de 0 (zero) a 7 (sete), para os códigos binário simples e códigos *gray*, com o último a oferecer melhores condições de operacionalidade ao operador *mutação*; i.e., de proceder à operação de *mutação* de uma forma incremental. É certo que também

³Richard Hamming (1915 - -). Cientista norte americano que se especializou na resolução de problemas em larga escala e na concepção de computadores.

podem ocorrer, embora não tão frequentemente, mutações mais radicais (eg. uma mutação no bit mais significativo do número 7 (sete) fará com que esse número se transforme no número 0 (zero)). No entanto, o mesmo fenómeno ocorre na natureza onde, apesar da maioria das mutações provocar variações muito pequenas, há aquelas que induzem mudanças radicais. Estudos há que mostram que o uso do código *gray* melhora o comportamento do *AG*, ou então que dá os mesmos resultados que o do código binário simples [CarSch88].

4.9 Combinando Algoritmos Genéticos com Redes Neurais

As *RNAs* (do tipo *feedforward*) são ferramentas poderosas para a resolução de problemas, em particular os de classificação e de aproximação. No entanto, o esforço colocado na procura da melhor das redes, ou seja, sua estrutura e parâmetros do algoritmo de treino, é normalmente proibitivo [RusNor95].

Por outro lado, e embora tanto os *AGs* como as *RNAs* se assumam como estratégias de procura universal, há estudos que mostram que o seu comportamento é função do domínio da aplicação [Kit90]. Os *AGs* realizam uma procura globalmente superior à das *RNAs* [Whi93] (Figura 4.7). Por *back-propagation* leva-se mais tempo a atingir a vizinhança de uma boa solução, mas esta é atingida de forma mais precisa. Por outro lado, os *AGs* ao varrerem todo o espaço de soluções, posicionam-se mais rapidamente sobre a região com a melhor destas. Todavia, não convergem necessariamente para essa, uma vez que a optimização da procura depende quase exclusivamente da operação de *mutação* [Koe94].

Assim, combinar ambas as estratégias de procura, nos chamados sistemas ***GANN*** (***Genetic Algorithms and Neural Networks***), aproveitando as vantagens de cada uma das partes, talvez seja o caminho a seguir. Esta via, com raízes nos anos oitenta, tem sido desde então objecto de atenção; há estudos empíricos que mostram que os sistemas *GANN* podem ultrapassar, em eficiência, quer os *AGs* quer as *RNAs*, na resolução de problemas (Figura 4.7) [Kit90].

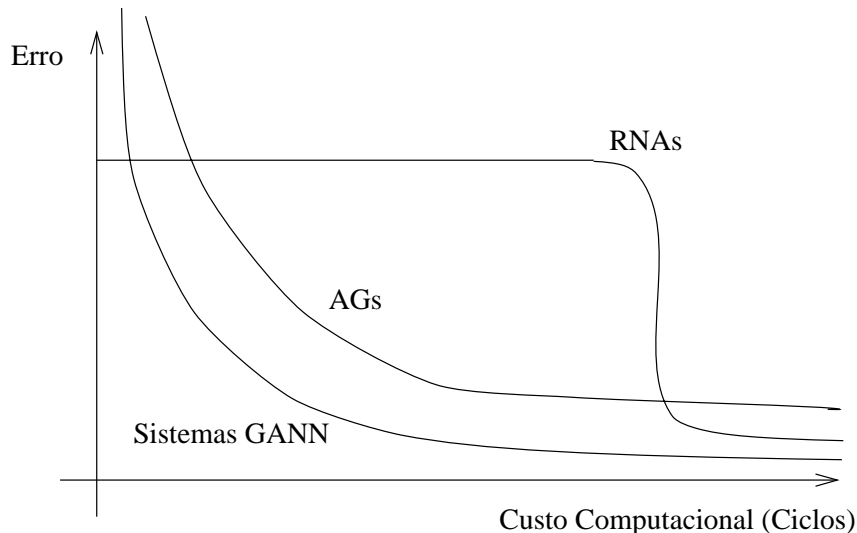


Figura 4.7: Tempos de procura para as várias estratégias

4.10 Os Sistemas *GANN*

Os sistemas *GANN* melhoram o desempenho das *RNAs* de duas formas: por evolução da topologia da rede através dos *AGs* e por aprendizagem via *backpropagation*, processo este que socorre-se em particular das aprendizagens *Lamarckian*⁴ e *Baldwin*⁵ [Koe94].

Pela primeira, os pesos, associados às ligações da *RNA*, são codificados em cromossomas, evoluindo pelo *AG*. Pela segunda, que segue um modelo caro a Darwin, a aprendizagem da *RNA* guia a procura evolucionária. Se uma rede está próxima do óptimo, então por *backpropagation* se chegará a esse óptimo, o que se traduzirá num bom valor para a *aptidão* da rede. Soluções (cromossomas), perto do óptimo, por partilharem muitos padrões, fazem com que o *AG* seja capaz de as explorar por amostragem no hiperplano.

Um dos sinais destes sistemas está no alto custo de avaliação da rede,

⁴Jean Baptiste Lamarck (1744-1829). Naturalista francês, autor da *Teoria do Transformismo*, que postula que características adquiridas por uma espécie, durante o seu tempo de vida, podem ser herdadas por gerações futuras.

⁵James Mark Baldwin (1861-1934). Psicólogo norte-americano. Foi o primeiro a afirmar que a aprendizagem assim como outros ajustes durante o tempo de vida das espécies podem afectar o curso da evolução.

em termos de esforço computacional a desenvolver; i.e., o uso dos sistemas *GANN* em certas situações pode tornar-se impraticável [WinCue95].

A codificação pode ser *directa*, caso esta se realize a partir dos parâmetros da rede (pesos, conectividade, etc.), ou *indirecta*, caso em que são utilizadas regras que comandam a construção da rede. Por sua vez, a codificação *directa* pode ser baseada nas ligações entre nodos ou nos próprios nodos da rede.

Capítulo 5

Estado da Arte

5.1 Uso de *RNAs* na *PST*

As primeiras aplicações de *RNAs* na *PST* surgiram nos fins dos anos oitenta. Esperava-se então ultrapassar as limitações próprias dos métodos convencionais de análise, em particular na área das séries financeiras, dada a sua natureza não linear. De início, as tentativas de aplicação de *RNAs* ao mercado bolsista revelaram-se algo desapontadoras, situação que se foi progressivamente esbatendo. Por exemplo, já no início dos anos noventa, Shoneberg [Sho90] refere a inaplicabilidade dos métodos estatísticos convencionais às previsões de curto prazo dos valores em bolsa. Era então já óbvio que os métodos de *alisamento* eram incapazes de detectar as pequenas variações de preço dos produtos financeiros no dia a dia, justificando-se assim o uso de novas ferramentas neste domínio.

5.1.1 *RNAs Feedforward*

A maior parte do trabalho nesta área passa pelo uso de *RNAs* do tipo *feedforward*, sendo o treino deste tipo de redes realizado normalmente, com recurso aos algoritmos de:

- *quickprop* - um derivado do algoritmo de *backpropagation* [Fah88];

- *cascade correlation* - um novo tipo de arquitectura e procedimento de treino para as *RNAs*. Em vez de um simples ajuste de pesos sobre uma rede fixa, este procedimento de treino de *RNAs* pressupõe a existência de uma configuração mínima da rede, a qual evolui para uma configuração final através da criação de camadas intermédias [FahLeb90].

Estudos há que parecem indicar que as redes deste tipo são as mais indicadas para a análise do comportamento dos mercados financeiros. Já em 93, Papadourakis, Spanoudakis e Gotsias [PapGot93] utilizam *RNAs* deste tipo para a previsão de preços de produtos financeiros, com resultados encorajadores. Um estudo dos índices bolsistas do mercado Kuala Lumpur, revelou que elementos de trabalho interessantes podem ser obtidos sem um uso excessivo de dados do mercado ou do conhecimento deste [YaoPoh95]. Em outro trabalho, Gan e Ng [GanNg95] elaboraram um estudo da evolução dos câmbios das principais moedas em relação ao dólar americano. Neste trabalho utilizaram-se modelos *univariáveis* (que usam os dados históricos de apenas uma variável) e modelos *multivariáveis* (com base em dados históricos de várias variáveis), com os últimos a produzir os melhores resultados. Outros casos de interesse são referenciados por Freisleben e Ripper [FreRip95], em que os resultados obtidos revelaram-se superiores aos obtidos por regressão linear. Para previsões de um período à frente, o modelo ARIMA (do método Box-Jenkins) dá melhores resultados que as *RNAs*. Porém, as *RNAs* impuseram-se ao modelo ARIMA para previsões de longo prazo.

Tang e Fishwick [TanFis91] estudaram as redes *feedforward*, usando 14 (catorze) séries temporais distintas e compararam os resultados obtidos com os dados pelo método de Box-Jenkins, com as *RNAs* a fornecer os melhores resultados, em particular para previsões a longo prazo. Segundo os autores, o desempenho das *RNAs* depende de vários factores: a estrutura da rede, os parâmetros de treino e a natureza da série temporal. Chegou-se igualmente à conclusão de que as *RNAs feedforward* com conexões de *atalho* combinam componentes lineares (dado pelas conexões de *atalho*), e não lineares (dada pela camada intermédia), funcionando como um super conjunto dos modelos de Box-Jenkins. Não é pois de admirar que as *RNAs* possam modelar o comportamento de séries temporais deveras complexas. A dificuldade no uso de *RNAs* está na falta de processos sistemáticos para construir os modelos de previsão (i.e., escolha correcta dos parâmetros da rede), e no facto das *RNAs* funcionarem como caixas negras, não se podendo tirar ilações a partir do conhecimento da sua estrutura.

Um sistema que escolhe o número de entradas da rede de acordo com os coeficientes de autocorrelação da série temporal é apresentado em [CorNev95]. Com este sistema conseguiram-se resultados que suplantam os obtidos pelo método de Box-Jenkins e Holt-Winters em algumas séries. Num outro trabalho, Thiesing, Middelberg e Vornberger [ThiVor94] advogam que as *RNAs* são apropriadas para a previsão de séries temporais caóticas, apresentando-as como uma alternativa aos métodos clássicos de previsão.

Uma posição diferente é defendida por Faraday e Chatfield [FarCha95], que ao estudarem modelos de previsão modelados por *RNAs*, não encontram nestes mais valias relativamente a outras estratégias. Os autores argumentam ainda que uma grande atenção deve ser dada à escolha do conjunto de variáveis de entrada, à arquitectura da rede, às funções de activação, ao ajuste da rede, à escolha dos pesos iniciais e ao escalonamento dos dados.

5.1.2 *RNAs* Recorrentes

Apesar da maior parte do trabalho em previsão usar redes *feedforward*, há autores a defender o uso de outro tipo de redes. Khotanzad e seus colaboradores [KhoMar95] argumentam que esta arquitectura não permite obter resultados fiáveis, pelo que se socorreram de um outro tipo de redes, chamadas redes *recorrentes*, que aplicaram à previsão da carga de um sistema eléctrico, obtendo excelentes resultados. Resultados idênticos são também apresentados por Ulbricht [Ulbr94], que clama que as redes *feedforward* apenas podem tratar, em simultâneo, excertos da série temporal. Ulbricht constata ainda que as redes *recorrentes* têm a vantagem de permitir a formação de estados de memória, justificando assim o uso de *redes recorrentes feedforward* (redes *recorrentes* que são actualizadas como as *feedforward*) para a previsão de curto prazo (eg. na identificação de fluxos de do tráfego).

5.1.3 Combinação de *RNAs* com Filtragem de Dados

Há trabalho a ser desenvolvido na combinação de *RNAs* com técnicas de filtragem de dados. Lisi, Nicolis e Sandri [LisSan95] submetem os dados da série temporal a um pré-processamento, um processo adaptativo de redução de ruído, sendo estes, depois de filtrados, alimentados a uma rede *feedforward*. O estudo mostra que a técnica de filtragem usada melhora a previsão.

5.2 Outras Técnicas Não Convencionais

Também têm sido usadas outras técnicas em estudos de previsão em séries temporais. Pictet e seus colaboradores [PicTom95] utilizaram *AGs* na previsão da evolução da flutuação cambial e outros indicadores financeiros. Em outro trabalho, López, Cembrano e Cellier [LópCel96] usam a *Fuzzy Inductive Reasoning* (uma técnica de modelação e simulação), com resultados comparáveis aos obtidos quer por *RNAs* quer pelo método de Box-Jenkins.

5.3 Comparação entre Técnicas

Uma comparação entre técnicas é feita em [McC93] para as previsões do mercado bolsista, número de manchas solares e séries caóticas de Mackey-Glass. São estudadas redes *feedforward* (sem camadas intermédias; com camadas intermédias e retropropagação, usando o algoritmo *cascade correlation* e funções de activação *radial basis*¹); redes recorrentes de vários tipos; e *AGs*. Neste estudo, McCluskey conclui que a recorrência pode ser útil para algumas séries, mas uma escolha cuidada das entradas da rede (ou janela temporal) é crucial. As redes lineares (ou seja, sem camadas intermédias), assim como os *AGs*, não são modelos a seguir no estudo de séries do mercado bolsista, por não serem imunes ao ruído; ora neste universo as *RNAs* dão bons resultados, desde que haja cuidado com a forma em como se faz a alimentação da rede.

5.4 Estado Actual

Actualmente, o uso de *RNAs* na *PST* continua a ser alvo de atenção por parte da comunidade científica, com a produção de textos sobre a matéria, organização de *workshops* e conferências, e até de uma competição [WeiGer94] [Azo95], o que por si é um óptimo indicador de que as *RNAs* têm vindo a constituir uma boa alternativa à *PST*, existindo ainda espaço para investigação na área.

¹Função de activação do tipo gaussiano, modelada por: $e^{-\left(\frac{x-w}{\sigma}\right)^2}$, onde w e σ designam o centro e a largura do gaussiano [MooDar82].

Capítulo 6

Descrição do Sistema Desenvolvido

6.1 Arquitectura das *RNAs*

Quando se usam *RNAs* algumas escolhas têm de ser feitas à partida, o que é realizado olhando ao tipo de dados e à especificidade do problema [Gal93][Pre94]. Por outro lado, atendendo a que a maior parte dos trabalhos em curso nesta área utiliza *RNAs feedforward*, esta situação é aqui entendida como um bom motivo para se adoptar este tipo de redes neste trabalho.

Tendo como objectivo reduzir os espaço de soluções, foi decidido usar redes completamente interligadas com *bias*, com apenas uma camada intermédia e sem ligações de *atalho* [CorNev96a]. O uso de redes completamente interligadas permite que não haja preocupações com a existência ou não existência de ligações entre nodos, o que simplifica todo o processo. O peso do *bias* aumenta o potencial da *RNA* para assimilar informação sem acrescentar extra complexidade, dado que este valor é definido, de forma automática, pelo algoritmo de treino. Por outro lado, a maior parte das aplicações apenas força à utilização de uma camada intermédia [Gal93]. Por este motivo, achou-se ser esta a topologia a seguir, dado que o uso de camadas extra iria aumentar a complexidade da rede (e com isso o esforço computacional) de forma desnecessária.

A organização da rede em camadas será representada por produções no

formato $C_e - C_i - C_s$ para uma *RNA* com C_e nodos de entrada, C_i nodos intermédios e C_s nodos de saída [Pre94]. Os pesos iniciais das ligações entre nodos da rede serão gerados aleatoriamente, dentro do intervalo $[-\frac{2}{z}; \frac{2}{z}]$, para um nodo com z entradas. Para o treino foi escolhido o algoritmo *RPROP*, devido às suas múltiplas vantagens (Secção 3.6) [RieBra93]. Como funções de activação foram utilizadas as da Tabela 3.1.

Existem várias formas de alimentar uma série temporal a uma rede, para se processar uma previsão. Neste trabalho foi decidido utilizar uma janela temporal deslizante, de comprimento k , para uma *RNA* com k nodos de entrada. O objectivo é obter à saída o valor desejado, em função de k valores dados [NevCor97b]. Esta simplificação faz com que o número de pesos a estimar seja menor [FarCha95]. Assim, a rede toma a forma $k - Ci - 1$. Os casos de treino criados pela janela temporal deslizante são dados por

$$\begin{array}{lcl} x_1, x_2, \dots, x_k & \rightarrow & x_{k+1} \\ x_2, x_3, \dots, x_{k+1} & \rightarrow & x_{k+2} \\ \dots & \rightarrow & \dots \\ x_{p-k}, \dots, x_{p-1} & \rightarrow & x_p \end{array}$$

para a série temporal x_1, \dots, x_p [CorNev96a]. Estes casos de treino são alimentados à rede por esta ordem devido à escala temporal em que os dados são gerados.

Para evitar problemas de *overfitting*, o treino da rede é antecipado, de acordo com o descrito na Secção 3.9.1, pelo que a informação disponível é dividida em dois conjuntos: um conjunto de *casos de treino* (para assimilação), e um conjunto *casos de validação* (para testar se a rede continua ou não com capacidade de generalização). Como foi referido anteriormente (Secção 3.9.1), a escolha da cardinalidade de cada um destes conjuntos é algo crítica. Mais *casos de treino* permitirão à rede assimilar mais informação. Por outro lado, menos *casos de validação* farão com que o treino da rede pare numa altura indesejada, ficando a rede com dificuldade de generalização. Dado que os elementos mais recentes da série afectam fortemente a previsão, foi decidido usar apenas 10% da informação disponível como *casos de validação* [NevCor97b]. Este valor foi escolhido com base em evidência empírica e alguma intuição. Para os critérios de paragem foram usados os valores de $\alpha = 25$, $\beta = 0,1$ e $\gamma = 4000$ (valores escolhidos por experimentação e com base nas indicações dadas por Prechelt [Pre94]).

Após o treino da rede, é então possível efectuar previsões de curto ou longo

prazo de uma forma simples. Previsões de curto prazo são feitas alimentando a rede com os últimos valores conhecidos para a série temporal, ou seja

$$P_{1,p} = RNA(x_{p-k+1}, \dots, x_p)$$

em que $RNA(x_1, \dots, x_k)$ representa a função que é modelada pela rede, sendo $P_{i,j}$ a previsão no período j , para i períodos à frente de j . Assim, uma previsão de curto prazo, para l períodos, será dada pelo conjunto

$$\{P_{1,p}; P_{1,p+1}; \dots; P_{1,p+l-1}\}$$

Considere-se, a título de exemplo, uma série de periodicidade mensal, dada por $\{\dots, 5, 16, 20, 16, 20, 24\}$, sendo os últimos valores da série relativos aos meses de Janeiro a Junho. Admitindo-se que se efectua a previsão do comportamento da série a partir de uma rede com 3 (três) nodos de entrada, e que o mês actual é o mês de Março, ter-se-á que a previsão para o mês de Abril será dada por $P_{1,\text{Março}} = RNA(5, 16, 20)$. De modo análogo, em Abril tem-se que $P_{1,\text{Abril}} = RNA(16, 20, 16)$. Uma previsão de curto prazo, ao longo de três meses ($l = 3$), será então dada por $\{P_{1,\text{Março}}; P_{1,\text{Abril}}; P_{1,\text{Maio}}\}$, em que “{” e “}” é a notação para conjuntos.

Previsões a longo prazo são efectuadas alimentando a rede com os valores conhecidos para a série temporal (dados e previstos) [CorNev96a]

$$\begin{aligned} P_{1,p} &= RNA(x_{p-k+1}; \dots; x_p) \\ P_{2,p} &= RNA(x_{p-k+2}; \dots; x_p; P_{1,p}) \\ P_{3,p} &= RNA(x_{p-k+3}; \dots; x_p; P_{1,p}; P_{2,p}) \\ &\dots \\ P_{l,p} &= RNA(P_{l-k,p}; \dots; P_{l-1,p}) \\ &\dots \end{aligned}$$

Uma previsão de longo prazo, efectuada até l períodos à frente de p , será então dada por

$$\{P_{1,p}; P_{2,p}; \dots; P_{l,p}\}$$

6.2 Normalização dos Dados

Como foi referido, os dados dos casos de treino precisam de ser normalizados, conhecidos que são os contradomínios das funções de activação (Tabela 3.1).

É difícil saber à partida qual o intervalo ideal, tal dependendo de cada série. Testes no terreno levam-nos a decidir pelo intervalo $[0, 1]$, pelo que foi este o intervalo escolhido. Uma forma simples de passar esta decisão à prática poderia passar por uma normalização linear dos dados dos casos de treino, de acordo com o procedimento descrito na Secção 3.7. Todavia, há séries temporais que possuem componentes de *tendência* (Secção 2.8.1), o que faz com que a série possa vir a apresentar valores fora do intervalo mencionado em epígrafe. Isto acontece porque o conjunto de valores possíveis para a rede à saída tem que estar no intervalo $[min, max]$, em que *min* e *max* falam pelos valores máximo e mínimo da série. Sabendo que, em princípio, quanto maior for o intervalo, mais fácil será a assimilação de informação por parte da rede, foi decidido usar o intervalo de $[0, 1]$ para as séries estacionárias, e o intervalo $[0, 2; 0, 8]$ para as séries com *tendência* [NevCor97b].

Para detectar a existência de *tendência* foi usada uma *RNA*, designada por *TEND(ência)*. Esta rede foi treinada com os 10 (dez) primeiros coeficientes de autocorrelação de 18 (dezoito) séries temporais, exigindo-se à sua saída o valor de +1 se existir *tendência* e -1 caso contrário. Após algumas experiências foi decidido usar uma *RNA* com a topologia $10 - 5 - 1$ e função de activação *sigmoid1*, o que acabou por se revelar bastante premonitório.

6.3 O Sistema *GANN*

6.3.1 Factores que Influenciam a *PST*

A partir das experiências com vários tipos de redes, pode-se afirmar com alguma segurança que a previsão depende dos seguintes factores:

- C_e - o número de nodos de entrada;
- Δ_0 e Δ_{max} - os parâmetros do algoritmo *RPROP*;
- f - a função de activação;
- C_i - o número de nodos intermédios;
- s - a semente para a geração aleatória dos pesos iniciais.

Isto pode ser facilmente explicado, atendendo a que o primeiro factor, C_e , determina o tamanho da janela temporal, enquanto que os outros parâmetros afectam o modo como a rede aprende [NevCor97b].

É aqui que entra o *AG*, que irá permitir escolher os valores dos parâmetros que são centrais em cada série temporal. Para o sistema *GANN* foi utilizado o modelo de aprendizagem de Baldwin, o qual guia a rede na procura evolucionária das soluções para o problema a ser objecto de atenção.

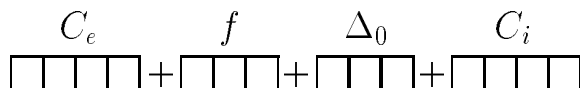


Figura 6.1: Codificação da *RNA*

Estes parâmetros foram codificados usando código *gray* de base 2, localizando-se os mais significativos à esquerda do cromossoma (Figura 6.1). O número de nodos, C_e e C_i , foi restringido ao intervalo $[3, 14]$, por permitir uma codificação de 4 (quatro) bits. Existe alguma evidência de que este é o intervalo mais adequado [CorNev95], e um alargamento deste iria fazer crescer de forma exponencial o espaço de procura. Em relação aos parâmetros do algoritmo *RPROP*, a Δ_{max} foi dado o valor 50 (cinquenta) (aconselhado em [RieBra93]), e Δ_0 foi codificado em apenas 3 (três) bits, com valores tomados do conjunto $\{0, 1; 0, 2; \dots; 0, 8\}$. Por último, a função de activação foi codificada também em 3 (três) bits. Resta dizer que a semente para a geração aleatória dos pesos iniciais da *RNA*, não foi codificada, porque a sua influência na obtenção de uma previsão é imaterial, por definição.

6.3.2 Função de *Aptidão*

Um aspecto crucial do sistema tem a ver com a escolha da função de avaliação do desempenho de um cromossoma ou solução, ou seja, da função de *aptidão*. Normalmente, quando se utilizam *AGs*, a função de *aptidão* já é conhecida, sendo relativamente fácil de calcular. Neste caso, isso não sucede. Não é fácil encontrar um critério, que sendo sólido, garanta que uma dada rede modela uma série temporal melhor (e com isso efectua uma melhor previsão) do que uma outra qualquer. Este critério terá de ser obtido, de algum modo, através do treino da rede neuronal. Em princípio, quanto menor for o erro no fim do

treino, mais informação foi assimilada pela rede, pelo que melhores devem ser as previsões. Porém, atendendo ao que foi dito na Secção 3.9, podem ocorrer fenómenos de *overfitting*, onde a rede, por assimilar em demasia a informação contida nos casos de treino, perde a sua capacidade de generalização. Face a este facto e dado que foi adoptada uma paragem antecipada para o treino da rede, parece lógico utilizar antes o erro dos *casos de validação* (MQE_{va}). O princípio assenta em que se a rede apresenta um valor de erro para os casos de validação, que é menosprezável, então é porque tem uma boa capacidade de generalização e/ou previsão.

Esta via parece ser inovadora, dado que outros estudos ao fazerem a comparação entre os desempenhos de diferentes redes, tomaram como referencial o erro (global) no treino da rede. Uma análise mais cuidada do fenómeno é efectuada em [FarCha95], que apresenta três métricas que devem ser tomadas em consideração, nomeadamente:

- o erro do treino (neste caso a *Soma do Quadrado dos Erros* (SQE));
- o *Crítério da Informação Akaike* (CIA) ($CIA = n \ln \frac{SQE}{n} + 2p$);
- o *Crítério da Informação Bayesian* (CIB) ($CIB = n \ln \frac{SQE}{n} + p + p \ln n$).

onde $p = (C_e + 2) * C_i + 1$ e n denota o número de *casos de treino*. Os dois últimos critérios penalizam certas topologias de rede. O facto está em que estas possuem demasiadas conexões, apresentando por conseguinte um maior número de graus de liberdade para o mesmo conjunto de *casos de treino*, existindo por isso o perigo de que o algoritmo de treino se fixe em demasia nestes casos, o que deve ser evitado. Conclui-se-ia, então, que o critério CIB seria o critério a seguir.

Ora, não é fácil demonstrar que um critério é melhor do que o outro. Para escolher o melhor critério foram feitas algumas experiências com 6 (seis) séries (retiradas da Tabela 6.3), o que levou ao treino de 30 (trinta) redes neuronais, geradas de forma aleatória. O número de períodos previstos, l , foi de 12 (doze) para as séries sazonais (séries 2 e 3) e de 10 (dez) para as restantes séries (séries 1, 6, 8 e 9). Para as medidas MQE , CIA e CIB , o treino da rede foi parado apenas por *Progresso no Treino* e *Número Máximo de Iterações*. A medida de desempenho escolhida foi a correlação, ρ , entre o valor do critério escolhido e os erros de previsão de curto e longo prazo, MQE , para as 30 (trinta) redes. Um valor elevado para a correlação significa que existe uma

relação directa entre o critério e o erro de previsão, o que é justamente o que se pretende. Nas Tabelas 6.1 e 6.2 dão-se os resultados das experiências realizadas, para previsões de curto e longo prazo. Em primeiro lugar, há que salientar o facto de todos os valores das correlações calculadas serem positivos, o que é um bom começo. Por outro lado, e indo contra o exposto em [FarCha95], os resultados indicam que existe uma menor correlação entre os critérios *CIA* e *CIB* com o erro de previsão (exceptuando-se o caso da série 6), com o último dos critérios a dar os piores resultados. Uma explicação para o ocorrido poderá estar no facto de no estudo ter sido apenas estudada uma série. Os resultados mostram que, de um modo geral, os melhores critérios são os do *MQE* e do *MQE_{va}*. O valor elevado para a correlação obtido com o *MQE* em algumas das séries (como é o caso das séries 1 e 3 para as previsões a curto prazo, ou das séries 1 e 8 para as previsões a longo prazo), indica que o fenómeno de *overfitting* não ocorre tanto como seria de esperar. No entanto, o número de séries estudadas não permite tirar conclusões definitivas. Embora pelos dados disponíveis se possa concluir que qualquer um dos critérios poderia ser utilizado, decidiu-se adoptar pelo último, *MQE_{va}*, por ser de mais fácil cálculo (a existência de um terceiro critério de paragem, a *Perda de Generalização*, implica que o treino da rede tenda a terminar prematuramente). Para além disso, esta escolha parece ser a mais lógica para uma solução mais abrangente; i.e., que se aplique a qualquer tipo de série temporal, por se evitar o fenómeno de *overfitting*. Dado que o *AG* retorna o máximo da função de *aptidão*, tem-se que este máximo será dado por

$$aptidão = 1/MQE_{va}$$

Tabela 6.1: Correlações entre os vários critérios e o erro para a previsão de curto prazo

Série	MQE	MQE _{va}	CIA	CIB
1	0,97	0,94	0,72	0,52
2	0,34	0,65	0,51	0,36
3	0,86	0,83	0,76	0,50
6	0,69	0,47	0,71	0,62
8	0,67	0,76	0,26	0,17
9	0,55	0,79	0,34	0,24

Tabela 6.2: Correlações entre os vários critérios e o erro para a previsão de longo prazo

Série	MQE	MQE _{va}	CIA	CIB
1	0,98	0,97	0,72	0,50
2	0,35	0,78	0,50	0,36
3	0,13	0,08	0,17	0,03
6	0,45	0,33	0,38	0,26
8	0,82	0,82	0,19	0,06
9	0,45	0,73	0,26	0,15

6.3.3 Estrutura do Sistema *GANN*

Uma vez que o *AG* desempenha o papel de uma função de optimização de segunda ordem, a escolha dos parâmetros deste não é muito crítica. Baseado em alguma intuição, experiência e tendo em conta o binómio *eficiência computacional - eficácia da solução*, foi decidido usar uma população de 30 (trinta) indivíduos, selecção baseada na ordem, cruzamento de um ponto e uma taxa de mutação de 0,02 [NevCor97b] (Figura 6.2); i.e, inicialmente, um conjunto de indivíduos são gerados aleatoriamente, sendo depois cada rede concebida de acordo com a informação do (seu) genoma. A *aptidão* é função do treino da rede. Após avaliação, todos os indivíduos são ordenados segundo a sua *aptidão*. Os operadores de *cruzamento* e *mutação* irão criar uma nova população, e os seus constituintes serão avaliados. Finalmente, a selecção baseada na ordem irá escolher as melhores redes para ambas as populações, dando origem a uma nova geração. Este processo continua até que algum critério de paragem seja atingido (neste caso após g iterações). De referir que o sistema guarda em memória a melhor rede que entretanto for gerada, devolvendo-a no fim do processo.

6.4 Arquitectura do Sistema

Esta secção descreve de um modo genérico o sistema informático desenvolvido. Certas decisões tiveram de ser tomadas tendo em conta uma medida de qualidade para a solução desejada, e o esforço computacional a exigir ao sistema. Assim, teve que se estabelecer um compromisso entre o que era

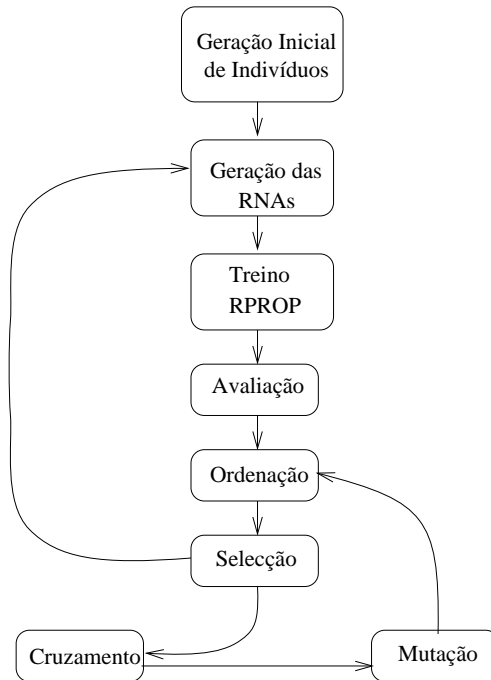


Figura 6.2: Estrutura do sistema *GANN*

teoricamente possível e o que os recursos disponíveis permitiam [CorNev96b].

O sistema é apresentado na Figura 6.3, compreendendo o pré-processamento, o sistema *GANN*, assim como a interface com o utilizador. Em primeiro lugar calculam-se as autocorrelações da série e procede-se ao teste de aleatoriedade, de acordo com o procedimento descrito na Secção 2.10. Caso a série seja aleatória, o valor médio dos elementos da série é passado ao módulo de interface, dado que nestas condições o valor médio é a única previsão possível. Sendo a série previsível, procura-se encontrar uma *tendência*, a partir da rede *TEND*, o que tornará possível à posterior normalização dos elementos da série. A seguir procura-se determinar a melhor topologia para a rede, por recurso ao sistema *GANN* descrito na Secção 6.3. Há ainda que mencionar que a informação acerca da amplitude da janela temporal para as várias redes, C_e , é passada deste para o módulo de pré-processamento, de modo a que se possam criar os *casos de treino* de acordo com esta. Finalmente, o módulo de interface trata da interacção com o utilizador, apresentando-lhe os resultados da previsão.

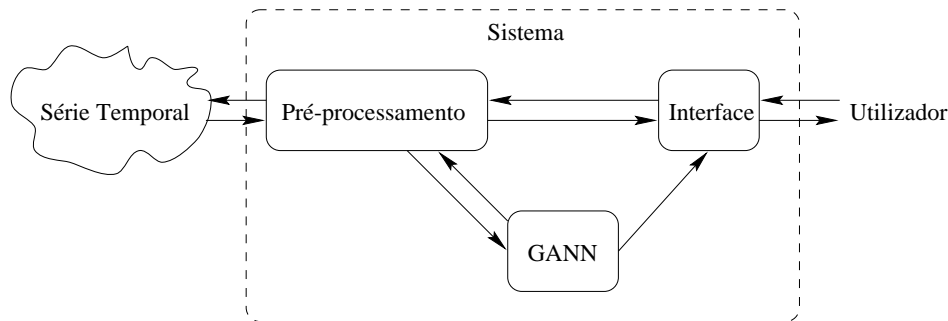


Figura 6.3: Arquitectura funcional do sistema

O sistema foi desenvolvido sobre o sistema operativo *Unix*, sendo usadas as linguagens de programação *C* [KerRit88] e *Prolog* [AlmFri93a]. Enquanto que, por uma questão de eficiência computacional, a linguagem *C* é usada na implementação dos *AGs* e dos procedimentos de treino das *RNAs*, a linguagem de programação em lógica *Prolog* é usada na interface com o utilizador, no módulo de pré-processamento de dados, bem como na integração das diferentes unidades funcionais do sistema [CorNev96b].

6.4.1 Integração entre o *C* e o *Prolog*

Para manipular as *RNAs* foi desenvolvido um programa escrito em *C*, chamado de **nn**. O programa é um dos componentes do *logo GANN*, passível de ser utilizado com qualquer tipo de problema, desde que se usem *RNAs* do tipo *feedforward*, completamente interligadas. Este programa aceita comandos com vários parâmetros, permitindo criar, treinar e obter resultados com qualquer rede neuronal. As redes e os casos de treino são armazenados em ficheiros. A interacção com o *SICStus Prolog* é feita através do comando `unix(system(...))` [AlmFri93b].

No caso do *AG*, todas as operações foram codificadas em *C*, por uma questão de eficiência, como já foi referido, exceptuando-se as de avaliação, que são tratadas usando a linguagem de programação em lógica *Prolog*. A interligação entre o *C* e o *Prolog* é um processo dinâmico, sendo as funções em *C*, a ser acedidas a partir do *Prolog*, declaradas ao *correr da pena* [AlmFri93b].

A título demonstrativo, considere-se o código *C* que implementa as funções que avaliam o estado de uma população e que corporizam o sistema *GANN*:

```

populacao avaliacao(populacao p)
{
    ...
    predicado=SP_predicate("avalia",2,"");/* declarar o predicado */
    populacao_para_lista(p,&P); /* converter a populacao em lista */
    ...
    SP_query(predicado,&P,&PA); /* invocar o predicado Prolog      */
    p=lista_para_populacao(PA); /* converter a lista em populacao */
    return(p);
}
...

void gann(long Populacao, /* Numero de elementos da populacao */
          long Semente,   /* Semente para a geracao aleatoria */
          /* da populacao inicial */
          long G, /* Ultima geracao do AG (criterio de paragem) */
          long *Ce, /* Numero de nodos de entrada */
          char *F, /* Funcao de activacao */
          double *Delta0, /* Delta0 */
          long *Ci) /* Numero de nodos intermedios */
{
    populacao p; /* Declaracao da variavel p, do tipo populacao */
    ...
    p=pop_inicial(Populacao,Semente);/* Criar a populacao inicial */
    p=avaliacao(p); /* Avaliar a populacao p */
    ...
}

```

A função *gann* é invocada a partir do *Prolog* da seguinte forma:

```

foreign(gann,c,gann(+integer,+integer,+integer,
                  -integer,-string(10),-float,-integer)).

gann(RNA,Mse):- gann(30,12345,350,Ce,F,Delta0,Ci),
                melhor_RNA(Ce,F,Delta0,Ci,RNA,Mse).

```

Segue-se a extensão do predicado *avalia(,)*, escrito em *Prolog*:

```

% avalia: Populacao -> Populacao Avaliada

```

```

avalia(P,PA):- cache(P,P1),                % procurar a aptidao dos
                                                % cromossomas em cache
              avaliar(P1,PA), % avaliar os restantes cromossomas
              junta_cache(PA). % juntar novas aptidoes a cache

```

6.4.2 A Interface Gráfica X-*Windows*

As potencialidades gráficas da biblioteca **gmlib** [AlmFri93b] do *SICStus Prolog* foram utilizadas no desenho da interface do sistema *GANN* (Figura 6.4) [CorNev96b]. A janela que apresenta a aplicação serve-se do código na linguagem de programação em lógica *Prolog*

```

janela(menu):- Serie <= button("Series",serie), % criar botao
              ...
              % criar a janela que apresenta a aplicacao
              W <= window("Neuro Previsao",...,Serie,...),
              W => open(500,500), % abrir a janela
              espera_accao(W). % esperar por accao do utilizador

```

Ao utilizador são dadas as opções *Séries*, *Previsão*, *Configuração* e *Saída*. A opção *Séries* permite-lhe manipular as séries temporais, realizar operações como *criar*, *eliminar*, *ler* e *escrever para ficheiros*, *consultar dados*, *analisar a série* ou *criar gráficos*. Pela sub-opção *Analisar* consegue-se uma decomposição da série, ou seja, identificam-se os vectores *tendência* e *sazonalidade*. De certa forma esta sub-opção configura um processo de extracção de conhecimento [DecFoc95]. A identificação da *tendência* obtém-se pelo processo descrito na Secção 6.2, a partir do conhecimento das autocorrelações dos elementos da série e da *RNA TEND*. Por sua vez, a identificação da *sazonalidade* baseia-se no que foi referido na Secção 2.13. Primeiro há que verificar se a série possui *tendência*. Se for este o caso, então a série é *diferenciada*, enviando-se as autocorrelações para posterior processamento. Este passa por se percorrer o conjunto das autocorrelações à procura do um máximo, que terá de ser superior a um dado valor pré-definido (neste caso foi escolhido o valor 0,45, o que se revelou satisfatório). O conjunto de autocorrelações é formado pelas 30 (trinta) primeiras autocorrelações, com exclusão das três primeiras

$$\{r_k \in [-1, 1] : \forall k \in \mathbb{N} \wedge k > 3 \wedge k \leq 30\}$$

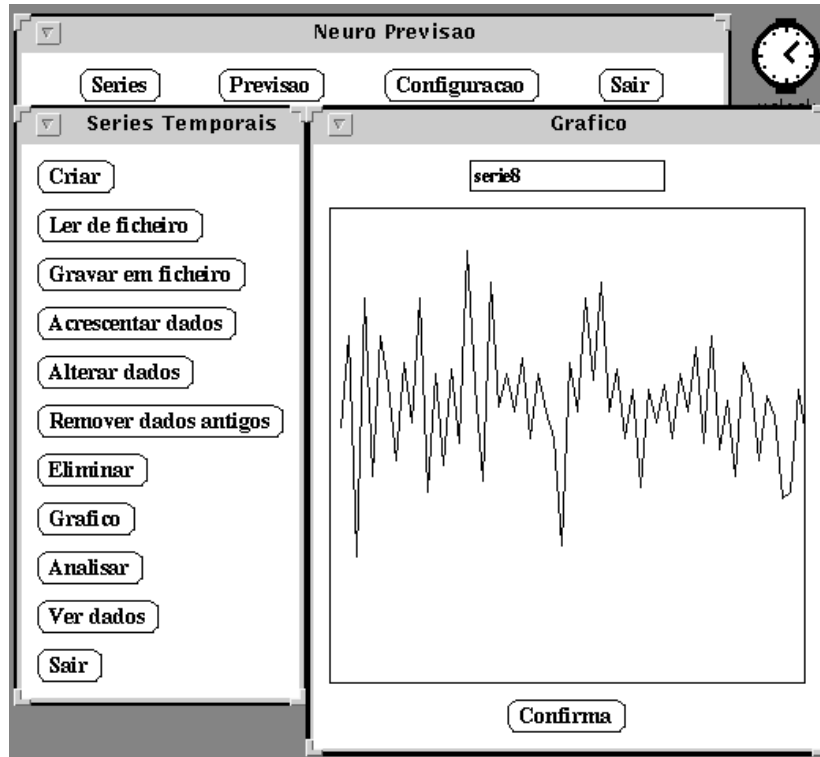


Figura 6.4: A interface gráfica *X-Windows*

O valor de k para o qual r_k é máximo é um bom candidato para ser o factor sazonal. A confirmação deste facto é feita pelo último teste, que passa por verificar se a autocorrelação do deslocamento $2k$ corresponde ao segundo valor mais elevado do conjunto. Caso tal ocorra, é sinal que a série é sazonal com um factor de sazonalidade k . Este processo revelou-se eficaz em todas as séries estudadas.

Através da opção *Previsão* é possível construir-se o modelo para a realização de previsões, assim como realiza-las. Pela sub-opção *Construir Modelo de Previsão* afere-se da aleatoriedade ou não aleatoriedade da série; caso esta seja previsível (não aleatória), invoca-se o sistema *GANN*, caso contrário toma-se como resposta o valor médio dos elementos da série. Por último, a opção *Configuração* serve para configurar certos parâmetros do sistema, como seja a percentagem de elementos usados como *casos de validação* ou os critérios de paragem a utilizar (α , β e γ).

6.5 Séries Utilizadas

Neste trabalho houve o cuidado em seleccionar séries temporais conotadas com problemas reais, que já tivessem sido objecto de análise por métodos convencionais (Tabelas 6.3 e 6.4) [BoxJen76][HanRei89][MakMcG78][Mak82]. Na Tabela 6.4, l representa o número de elementos da série que serão previstos; i.e., que serão usados nos casos de teste. O valor deste será de 12 (doze) para as séries mensais e de 10 (dez) para as restantes. A série 10 apresenta-se com uma forte *tendência*, a qual deve contribuir para esbater o factor sazonal, pelo que apesar de ser uma série com uma periodicidade mensal, não possui sazonalidade.

Tabela 6.3: Descrição das séries utilizadas

Série	Descrição
1	Número de manchas solares de Wolfer
2	Número de passageiros de uma companhia aérea
3	Entregas de equipamento de controlo de poluição
4	Preços de fecho de acções IBM de 29/6/1959 a 30/6/1960
5	Vendas de uma companhia
6	Preços de fecho de acções IBM de 17/5/1961 a 2/11/1962
7	Preços de fecho das acções de uma corporação
8	Resultados de um processo químico
9	Leituras de um processo produtivo de uma empresa
10	Volume de negócios de uma companhia privada dos EUA
11	Vendas de papel de impressão

6.6 Um Exemplo para o Sistema GANN

Aqui será descrito como se processa a criação do modelo de previsão para a série 1 da Tabela 6.4. Esta série possui 100 (cem) elementos (Figura 6.5). Os 10 (dez) últimos serão usados para testar a previsão (como *casos de teste*), e os restantes serão utilizados como *casos de treino*. De entre estes, os últimos 10% (9 (nove) elementos), serão usados como *casos de validação*.

Em primeiro lugar, o sistema calcula os valores para as autocorrelações

Tabela 6.4: Características das séries utilizadas

Série	Tamanho	Periodicidade	Tendência	Sazonalidade	l
1	100	Anual	Não	Não	10
2	144	Mensal	Sim	12	12
3	130	Mensal	Sim	12	12
4	261	Diária	Sim	Não	10
5	115	Mensal	Sim	12	12
6	355	Diária	Sim	Não	10
7	65	Diária	Não	Não	10
8	70	-	Não	Não	10
9	75	-	Não	Não	10
10	138	Mensal	Sim	Não	12
11	120	Mensal	Sim	12	12

dos elementos da série, obtendo-se um primeiro conjunto de valores $\{0,80; 0,43; 0,08; -0,15; -0,23\}$, em que o valor máximo de $0,8 > 0,207 (= 1,96 \frac{1}{\sqrt{90}})$, o que prova que a série é previsível (teste de aleatoriedade). Assim sendo, passa-se à criação do modelo de previsão. A *RNA TEND* retorna o valor $-0,73$ para as primeiras 10 (dez) autocorrelações da série, revelando, como seria de esperar, que se trata de uma série estacionária, a ser normalizada no intervalo $[0, 1]$. A seguir corre-se o *AG*. Na Figura 6.6 é dada a evolução do MQE_{va} ao longo das várias gerações. O valor médio dos elementos da população diminui rapidamente ao longo das primeiras gerações, como seria de se prever. Todavia, o valor óptimo diminui de uma forma lenta, havendo que se considerar várias gerações até que se atinja um valor razoável.

A rede óptima (menor MQE_{va}) foi obtida na geração 34, com o $MQE_{va} = 22$, para o indivíduo com o cromossoma 0001011101010001, com a topologia $4 - 4 - 1$ (0001 = 1 em código *gray*, logo $C_e = C_i = 1 + 3$), com a função de activação *sin* e com $\Delta_0 = 0,7$ (0101 = 7).

Tendo sido encontrada a rede óptima, resta efectuar a previsão. Na Figura 6.7 mostram-se as previsões de curto prazo, que se aproximam dos valores reais da série.

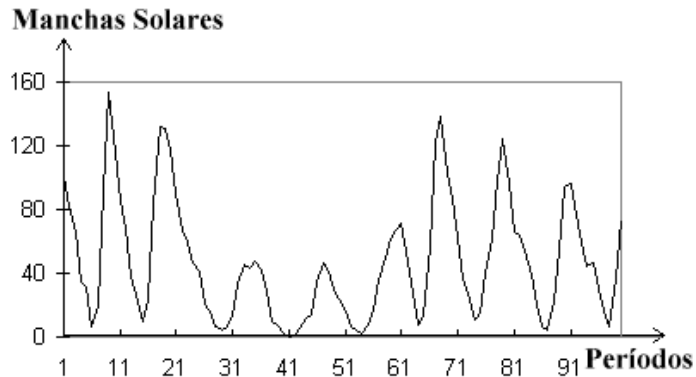


Figura 6.5: Valores anuais das manchas solares (série 1)

6.7 Resultados do Sistema *GANN*

6.7.1 Factores Óptimos

Na Tabela 6.5 são apresentadas as redes modeladas pelo sistema *GANN* para cada uma das séries da Tabela 6.3. Como seria de se esperar, as redes obtidas são bem distintas, não existindo factores dominantes; i.e., a existência de um ou mais parâmetros que sejam óptimos para todas as séries. Este facto atesta da necessidade do uso do *AG*. Há ainda a considerar o facto de as séries sazonais exigirem redes com uma maior janela temporal, com um C_e entre o 12 (doze) e o 13 (treze). Tal facto não surpreende dado que se tratam de séries com um factor de sazonalidade de 12 (doze), pelo que o uso de uma janela temporal inferior não permitiria captar o padrão sazonal. Nas séries não sazonais, quer com ou sem *tendência*, o tamanho da janela temporal situa-se, por sua vez, no intervalo $[3, 7]$. Isto é uma indicação de que para este tipo de séries os valores do passado recente influenciam fortemente a previsão. Para os restantes parâmetros parece não existir nenhuma relação causal entre os valores desses parâmetros e o comportamento da série. A justificação poderá ter a ver com o facto de estes parâmetros apenas afectarem a forma como a rede assimila os dados, e não como estes são criados, o que era o caso na situação anterior.

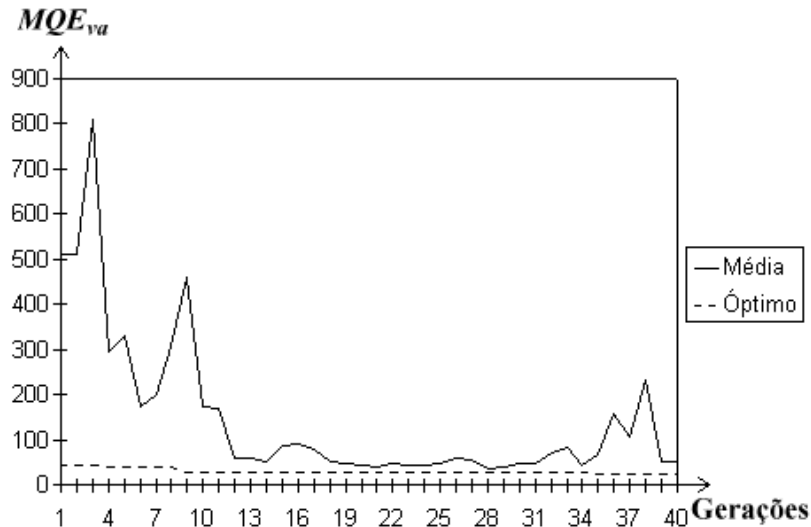


Figura 6.6: Evolução do MQE_{va} para a série 1

6.7.2 Comparação de Resultados

As Tabelas 6.7 e 6.8 espelham os erros de previsão, de curto e longo prazo, para o sistema *GANN* e para os métodos ARIMA [BoxJen76] e Holt-Winters [HanRei89], respectivamente.

Os resultados obtidos com a aplicação do método de Holt-Winters às séries mencionadas na Tabela 6.3, foram-no com $\alpha = 0,2$, $\beta = 0,2$ e $\gamma = 0,6$ (Secção 2.7.1). Na Tabela 6.6 apresentam-se as fórmulas de recorrência para o modelo ARIMA [BoxJen76][HanRei89][MakMcG78]. No caso da série 10, apenas são conhecidos os valores da previsão para segundo o modelo ARIMA, devidos a Makridakis [Mak82].

Em primeiro lugar, serão analisados os resultados de previsões de curto prazo. Os resultados obtidos com o método de Holt-Winters são superiores aos dos outros métodos para as séries sazonais (séries 2, 3, 5 e 11), o que pode ser facilmente explicado, atendendo-se ao facto deste método ter sido desenvolvido unicamente para a previsão deste tipo de séries (séries bem comportadas). No caso de séries com uma forte *tendência*, séries 4, 6 e 10, o sistema *GANN* dá melhores resultados que o método ARIMA para as séries 6 e 10. Um facto aliás deveras curioso acontece com a série 10, onde o método de Holt-Winters, que tipifica a série como sendo sazonal, apresenta

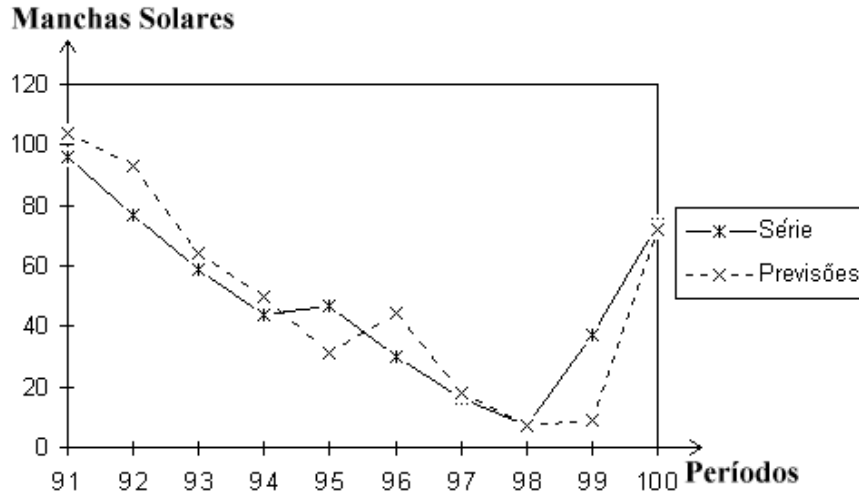


Figura 6.7: Previsões para a série 1

melhores resultados que os das restantes técnicas. Ora, é certo que não foi detectada a existência de sazonalidade através da análise das autocorrelações (Secção 2.13), embora seja bom ter-se em linha de conta que se está perante uma série de periodicidade mensal; i.e., não é de excluir a hipótese de a série apresentar uma certa sazonalidade. Finalmente, para as séries sem *tendência* e sem *sazonalidade* (séries 1, 7, 8 e 9), os resultados obtidos pelo sistema são similares aos obtidos pelo método ARIMA, e mesmo melhores em alguns dos casos (séries 1 e 8).

Nas previsões de longo prazo o cenário mantém-se, se exceptuarmos a perda de supremacia por parte do método de Holt-Winters. De facto, os resultados por este apresentados para as séries 2 e 11 são piores do que os obtidos pelos outros métodos. De um modo algo surpreendente, o sistema *GANN* consegue mesmo superar ambos os métodos para a série sazonal 2. No caso de séries com forte *tendência* não é possível efectuar comparações com os modelos ARIMA, os quais não foram concebidos para efectuar previsões de longo prazo. Na única comparação possível, o sistema *GANN* apresenta resultados que superam os do método de Holt-Winters (série 10). Para as restantes séries, sem *tendência* e sem *sazonalidade*, os resultados obtidos com o sistema *GANN* apenas superam os obtidos com a tecnologia ARIMA para a série 7.

Tabela 6.5: A rede óptima para cada série da Tabela 6.3

Série	Topologia	Função	Δ_0	MQE_{va}
1	4 – 4 – 1	<i>sin</i>	0,7	22
2	13 – 11 – 1	<i>linear</i>	0,4	227
3	13 – 3 – 1	<i>linear</i>	0,8	243966
4	7 – 9 – 1	<i>linear</i>	0,2	49,7
5	12 – 8 – 1	<i>gaussian</i>	0,5	16612
6	4 – 3 – 1	<i>cos</i>	0,8	38,5
7	6 – 14 – 1	<i>sigmoid2</i>	0,3	173,6
8	7 – 6 – 1	<i>sin</i>	0,8	33,3
9	6 – 6 – 1	<i>sigmoid2</i>	0,2	2,72
10	3 – 11 – 1	<i>sigmoid2</i>	0,6	5593
11	13 – 7 – 1	<i>gaussian</i>	0,5	2287

6.8 Filtragem de Dados

Como já foi referido (Secção 3.7), um aspecto a ter em atenção quanto ao uso de *RNAs*, e em particular no que respeita à sua aplicação à previsão, tem a ver com a filtragem de dados, um processo que facilita a assimilação da informação por parte da rede. Na secção anterior viu-se que as séries sazonais exigem uma grande janela temporal deslizando, entre 12 e 13 períodos. Contudo, alimentar uma rede com uma grande janela temporal poderá afectar a eficácia da previsão, devido a um aumento de entropia; i.e., o excesso de informação confunde a rede [NevCor97a]. Por outro lado, os resultados obtidos com o sistema para este tipo de séries não foram muito animadores. Assim, uma selecção correcta dos períodos poderá melhorar de forma significativa a previsão, especialmente para séries sazonais [FarCha95].

6.9 Janela Temporal Deslizante

O processo normal de efectuar a *PST* com *RNAs* é através do uso de uma janela temporal deslizando. A estrutura desta janela é dada pela produção $\langle 1, 2, \dots, n - 1, n \rangle$, em que n denota o tamanho da janela (Figura 6.8). Com a filtragem de dados pretende-se criar janelas temporais deslizantes,

Tabela 6.6: Fórmulas de recorrência para os modelos ARIMA

Série	Fórmulas de recorrência
1	$X_t = 14,9 + 1,32X_{t-1} - 0,63X_{t-2}$
2	$X_t = X_{t-1} + X_{t-12} - X_{t-13} - 0,4e_{t-1} - 0,6e_{t-12} + 0,24e_{t-13}$
4	$X_t = X_{t-1} + 0,09e_{t-1}$
5	$X_t = X_{t-1} + X_{t-12} - X_{t-13} - 0,77948e_{t-1} - 0,61031e_{t-12} + 0,47572e_{t-13}$
6	$X_t = X_{t-1} + 0,09e_{t-1}$
7	$X_t = 268,48 - 0,32218X_{t-1} + 0,20809X_{t-2}$
8	$X_t = 58,3 + 0,32X_{t-1} - 0,18X_{t-2}$
9	$X_t = 115,3 - 0,5295X_{t-1}$
10	Desconhecido
11	$X_t = X_{t-1} + X_{t-12} - X_{t-13} - 0,864e_{t-1} - 0,803e_{t-12} + 0,694e_{t-13}$

que contêm apenas a informação relevante para a previsão. Por exemplo, o método ARIMA [BoxJen76], sugere o uso de janelas do tipo $\langle 1, 12, 13 \rangle$ ou $\langle 1, 2, 12, 13 \rangle$, para séries com sazonalidade mensal, que aqui serão as usadas para testar o sistema (séries 2, 3, 5, 10 e 11 da Tabela 6.4).

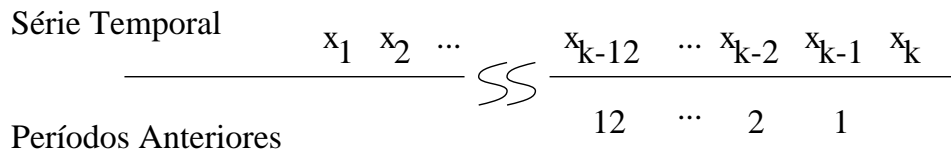


Figura 6.8: Períodos anteriores de uma série

6.10 O Sistema *GANNF*

Ao sistema *GANN* é dada uma nova funcionalidade, dando corpo a um novo sistema que será designado por *GANNF* (sistema *GANN* com **F**iltragem). É o utilizador, quando passa a analisar uma série, que opta entre os sistemas *GANN* e *GANNF*.

Doze tipos de janelas serão utilizados (Tabela 6.9) pelo *AG*, os quais im-

Tabela 6.7: Comparação entre as previsões de curto prazo obtidas pelo sistema *GANN* e as obtidas por outros métodos (usando o *MQE* como medida de erro)

Série	GANN	ARIMA	Holt-Winters
1	139,6	154	-
2	340,3	452	271
3	834578	-	530654
4	63,9	36,3	-
5	20672	15290	11435
6	62,9	72,1	-
7	3670	2939	-
8	66,8	141,5	-
9	265	240,4	-
10	20711	24217	18446
11	6238,7	2581,3	1885

plementam um verdadeiro processo de filtragem de dados [NevCor97a]. Estas janelas foram escolhidas com base em trabalho desenvolvido por Faraday e Chatfield [FarCha95]. De notar que cada janela dá o valor de C_e ; i.e., o número de nodos de entrada da rede. Estas janelas foram codificadas em 4 (quatro) bits, substituindo-se ao factor C_e do sistema *GANN* (Figura 6.9). Exceptuando este facto, o *AG* do sistema *GANNF* comporta-se de modo idêntico ao do sistema *GANN*.

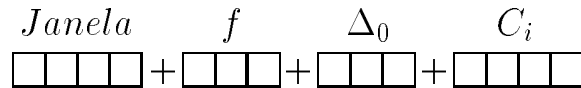


Figura 6.9: Estrutura de um cromossoma no sistema *GANNF*

Os casos de treino são criados de acordo com cada janela. Por exemplo, para uma janela do tipo 2, têm-se que

$$\begin{array}{ll}
 x_1, x_{11}, x_{12} & \rightarrow x_{13} \\
 x_2, x_{12}, x_{13} & \rightarrow x_{14} \\
 \dots & \rightarrow \dots \\
 x_{p-12}, x_{p-2}, x_{p-1} & \rightarrow x_p
 \end{array}$$

Tabela 6.8: Comparação entre as previsões de longo prazo obtidas pelo sistema *GANN* e as obtidas por outros métodos (usando o *MQE* como medida de erro)

Série	GANN	ARIMA	Holt-Winters
1	562,1	267	-
2	370,4	521,8	621,5
3	3888567	-	2927255
4	214,25	-	-
5	21542	20289,3	16954
6	170,4	-	-
7	2867	2897	-
8	270,3	161,3	-
9	316	205,2	-
10	33548	-	52012
11	6258,5	2707,7	3046,4

para a série temporal x_1, x_2, \dots, x_p .

Tal como no sistema anterior, depois de treinada a rede é possível efectuar previsões de um período à frente, bastando para isso alimentar a rede com os dados da última janela temporal. Previsões de longo prazo também são possíveis por realimentação da rede com previsões anteriores. Para o exemplo dado, tem-se que

$$\begin{aligned}
 P_{1,p} &= RNA(x_{p-11}, x_{p-1}, x_p) \\
 P_{2,p} &= RNA(x_{p-10}, x_p, P_{1,p}) \\
 &\dots \\
 P_{l,p} &= RNA(P_{l-12,p}, P_{l-2,p}, P_{l-1,p}) \\
 &\dots
 \end{aligned}$$

6.11 Um Exemplo para o Sistema *GANNF*

A título demonstrativo, será descrito como se processa a previsão, neste caso com o sistema *GANNF* e a série 2 da Tabela 6.3. Trata-se de uma série com *tendência* e *sazonalidade* mensal, um tipo muito comum em situações reais (Figura 6.10). Dos 144 elementos desta série, os primeiros 132 serão usados como *casos de treino*, sendo os últimos 13 elementos usados como *casos de*

Tabela 6.9: Conjunto de janelas temporais

Tipo	Janela	C_e
1	$\langle 1, 12 \rangle$	2
2	$\langle 1, 2, 12 \rangle$	3
3	$\langle 1, 12, 13 \rangle$	3
4	$\langle 1, 2, 3, 12 \rangle$	4
5	$\langle 1, 2, 12, 13 \rangle$	4
6	$\langle 1, 2, 3, 4, 12 \rangle$	5
7	$\langle 1, 2, 12, 13, 14 \rangle$	5
8	$\langle 1, 2, 3, 4, 12, 13 \rangle$	6
9	$\langle 1, 2, 3, 4, 12, 13, 14 \rangle$	7
10	$\langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \rangle$	12
11	$\langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 \rangle$	13
12	$\langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 \rangle$	14

validação, e os restantes 12 para testar a qualidade da previsão. Como a série apresenta *tendência* os *casos de treino* serão normalizados para o intervalo $[0, 2; 0, 8]$.

O passo seguinte passa por se correr o *AG*. A rede óptima foi obtida na geração 263, dada pelo cromossoma 00110001010111, que reporta à janela $\langle 1, 12, 13 \rangle$, a uma topologia $3 - 8 - 1$, a uma função *linear* e a um Δ_0 de 0,7. Na Figura 6.11 dão-se os resultados das previsões de curto e longo prazo.

6.12 Resultados do Sistema *GANNF*

O sistema foi apenas testado com as séries onde o processo de filtragem, tendo em conta um factor sazonal de 12, pudesse trazer alguma vantagem. Assim, foram consideradas as séries sazonais (2, 3, 5 e 11) e a série 10 que, aparentemente, é de alguma forma sazonal.

As redes óptimas modeladas obtidas pelo sistema *GANNF* são dadas na Tabela 6.10. Aqui, também se têm redes são diferentes, com nenhum factor a destacar-se, exceptuando-se as funções de activação, onde só concorrem as funções *linear* e *gaussian*. Tal facto poderá indiciar que estas funções

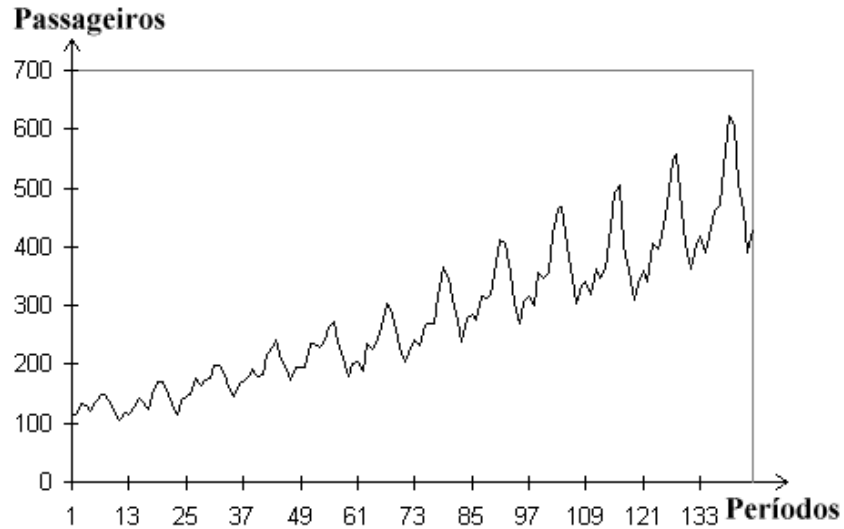


Figura 6.10: Valores mensais de passageiros de uma companhia aérea (série 2)

são as mais indicadas para funcionar com a filtragem de dados, embora o número de séries modeladas não seja significativo para permitir tirar quaisquer conclusões. A grande diferença está nos resultados, com as novas redes a apresentarem valores de C_e a situar-se entre 3 e 6. Este facto evidencia a importância do sistema de filtragem, que ao seleccionar os limites temporais (ou períodos) mais significativos, permitiu a obtenção de redes bastante simples.

Ao comparar-se os valores da métrica de avaliação de rede, MQE_{va} , para os sistemas $GANN$ e $GANNF$, tem-se que este último apresenta melhores

Tabela 6.10: A rede óptima para cada uma das séries estudadas

Série	Janela	Topologia	Função	Δ_0	MQE_{va}
2	1, 12, 13	3 – 8 – 1	<i>linear</i>	0,7	120,4
3	1, 2, 3, 4, 12, 14	6 – 6 – 1	<i>linear</i>	0,2	277101
5	1, 2, 3, 12	4 – 7 – 1	<i>gaussian</i>	0,5	9911
10	1, 12, 12	3 – 13 – 1	<i>linear</i>	0,2	6353
11	1, 2, 3, 12	4 – 3 – 1	<i>gaussian</i>	0,5	1213,7

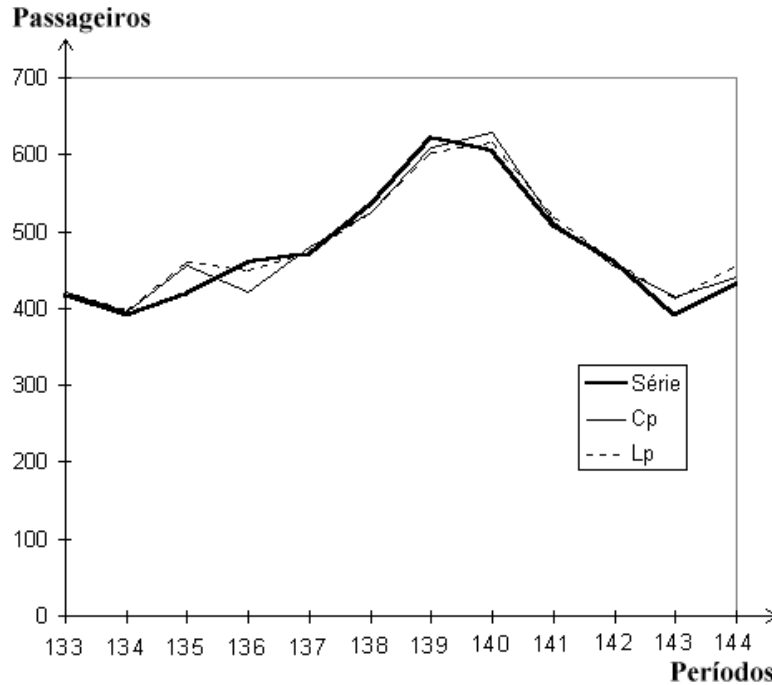


Figura 6.11: Previsões de curto e longo prazo para a série 2

resultados para as séries 2, 5 e 11. Para a série 10, não se estranha que o valor do MQE_{va} seja agora superior, dado que a rede óptima obtida via sistema *GANN*, com a topologia 3–11–1 para a janela $\langle 1, 2, 3 \rangle$, é um tipo de solução que está fora do espaço de soluções do sistema *GANNF*. Contudo, é curioso que tal aconteça também para a série 3, dado que neste caso a rede óptima encontrada pelo sistema *GANN* pertence ao espaço de soluções do sistema *GANNF*. Uma explicação para este facto poderá ser atribuída ao *AG*, que até à última geração, não teria conseguido chegar ao mesmo óptimo.

O método de Holt-Winters dá-nos melhores resultados que os obtidos por outros métodos de previsão de curto prazo, exceptuando-se os da série 10, onde os resultados obtidos com o sistema *GANNF* superam todos os restantes (Tabela 6.11). Para este tipo de previsão, o sistema *GANNF* apresenta um desempenho superior ao do sistema *GANN* para as séries 3, 10 e 11, e ao do método *ARIMA* para as séries 2 e 10. Poderá parecer estranho que com o sistema *GANN* se obtenham melhores resultados para as séries 2 e 5, tendo em atenção que o valor para o MQE_{va} obtido com este sistema, é maior que o

Tabela 6.11: Comparação entre as previsões de curto prazo obtidas pelo sistema *GANNF* e as obtidas por outros métodos (usando o *MQE* como medida de erro)

Série	GANN	GANNF	ARIMA	Holt-Winters
2	340,3	368,3	452	271
3	834578	759581	-	530654
5	20672	25103,8	15290	11435
10	20711	12486,3	24217	18446
11	6238,7	4500	2581,3	1885

Tabela 6.12: Comparação entre as previsões de longo prazo obtidas pelo sistema *GANNF* e as obtidas por outros métodos (usando o *MQE* como medida de erro)

Série	GANN	GANNF	ARIMA	Holt-Winters
2	370,4	308	521,8	621,5
3	3888567	657445	-	2927255
5	21542	31279	20289,3	16954
10	33548	14534	-	52012
11	6258,5	4792	2707,7	3046,4

obtido com o sistema *GANNF*. Ora, o que acontece é que, de um modo geral, existe uma forte correlação entre o MQE_{va} e o valor da previsão (como foi demonstrado na Secção 6.3.2), o que indica que, em princípio, quanto menor for o MQE_{va} , menor será o erro de previsão; porém, nada nos garante que tal conclusão possa passar a axioma.

As vantagens do sistema de filtragem vêm-se nas previsões de longo prazo, com o sistema *GANNF* a apresentar melhores resultados que os restantes métodos para as séries 2, 3 e 10 (Tabela 6.12). De facto, o sistema *GANNF*, em relação ao sistema *GANN*, obtém melhores resultados para quase todas as séries, exceptuando-se a série 5. Por último, o método de Holt-Winters perde o seu lugar de destaque, apresentando melhores resultados que o sistema *GANNF* apenas para as séries 5 e 11.

Capítulo 7

Conclusões e Perspectivas de Trabalho Futuro

Nos últimos anos novos algoritmos de optimização, como as *RNAs* e os *AGs*, tornaram-se objecto de atenção [RusNor95]. Mais recentemente, surgiu a possibilidade de combinar *AGs* com *RNAs*, nos chamados sistemas *GANN*, de forma a se aproveitarem as vantagens inerentes a ambos os paradigmas. Embora exista muito trabalho nesta área, há ainda imenso trabalho a desenvolver [Koe94]. O objecto de estudo tem estado na forma de integrar ambos os paradigmas, não se tendo explorado devidamente o potencial destes sistemas em aplicações práticas. O trabalho aqui desenvolvido vem procurar colmatar essa lacuna, nomeadamente no que respeita à aplicação dos sistemas *GANN* à *PST*, no que se julga ser um sector ainda pouco explorado.

7.1 Avaliação do Trabalho Desenvolvido

A *PST* é um problema clássico na Matemática Aplicada, com particular acuidade no domínio da Economia [NevCor97b]. Neste trabalho teve-se como objectivo fazer uma abordagem ao problema através do uso da aplicação de sistemas *GANN*.

Como foi visto no Capítulo 5, as *RNAs* têm um enorme potencial para a *PST*. O problema está, porém, na escolha da melhor rede, o que faz que se tenha de efectuar a sua procura num largo espaço de possíveis soluções. O

uso de *AGs* permite que essa procura seja mais eficiente. Mesmo assim, os sistemas *GANN* obrigam a um grande esforço computacional, de tal modo que certas decisões tiveram de ser tomadas tendo em conta o compromisso entre a eficiência da procura e a qualidade da solução. Nesta dissertação apresentou-se a forma que se considerou mais correcta de aplicar sistemas *GANN* à *PST*, tendo em conta o conhecimento, o tempo e os recursos disponíveis.

De acordo com o trabalho realizado, a previsão de uma série temporal através de uma *RNA*, depende de quatro factores [NevCor97b]: o número de nodos de entrada (C_e), a função de activação (f), um dos parâmetros do algoritmo de treino *RPROP* (Δ_0) e o do número de nodos intermédios (C_i). O *AG* funciona como um algoritmo de optimização de segunda ordem, pelo que a escolha dos parâmetros deste não foi considerado crítico. O afinamento dos valores para os parâmetros foi efectuado com base em alguma intuição e experiência. As redes que foram modeladas pelo *AG* revelaram-se bem distintas entre si, não havendo factores dominantes, justificando-se assim, a necessidade do *AG*. A excepção surgiu com as séries sazonais, que exigem uma maior janela temporal (C_e entre 12 e 13), para captar o padrão sazonal. Dado que os resultados obtidos pelo sistema *GANN* não se revelaram muito animadores para este tipo de séries, houve que desenvolver um sistema de filtragem. Este sistema teve como objectivo possibilitar a escolha dos espaços temporais que realmente influenciam a previsão, reduzindo-se assim o tamanho da janela temporal e, conseqüentemente, eliminando-se uma parte da entropia no sistema. De facto, as redes modeladas pelo sistema *GANNF*, apresentam uma janela temporal reduzida, com C_e a tomar valores no intervalo [3, 6].

Comparando os resultados obtidos, com os dados por métodos convencionais, tem-se que:

Séries Sazonais

- o método de Holt-Winters revelou-se o melhor nas previsões de curto prazo, o que não é de se estranhar, dado que foi desenvolvido especificamente para a previsão deste tipo de séries (i.e., séries bem comportadas);
- neste tipo de previsão, de curto prazo, os efeitos do sub-sistema de filtragem não são muito visíveis, fazendo contudo que os resultados obtidos com o sistema *GANNF* superem os do sistema *GANN* em três

das cinco séries estudadas;

- nas previsões de longo prazo o cenário altera-se; i.e., os resultados obtidos com o sistema *GANNF* são melhores que os obtidos com o sistema *GANN* (exceptuando-se o caso da série 5), superando mesmo os resultados obtidos via ARIMA e Holt-Winters em 3 das séries.

Séries com Forte Tendência

- nas previsões de curto prazo, os resultados obtidos via sistema *GANN* são da mesma ordem de grandeza que os obtidos pelo método ARIMA, conseguindo o sistema *GANN* superar este em duas das séries estudadas;
- nas previsões de longo prazo não foi possível efectuar comparações com os modelos ARIMA, os quais não foram criados para efectuar este tipo de previsão. Na única comparação possível, o sistema *GANN* apresenta melhores resultados que os obtidos pelo método de Holt-Winters (série 10).

Séries Estacionárias e Sem Sazonalidade

- nas previsões de curto prazo, e de um modo geral, os resultados obtidos por aplicação do sistema *GANN* são similares aos obtidos via ARIMA, e mesmo melhores do que estes para duas das séries estudadas;
- nas previsões de longo prazo, a eficácia do sistema *GANN* perde quando comparada com ARIMA, apresentando resultados que em apenas uma das séries estudadas ultrapassam os obtidos via ARIMA.

Estes resultados sugerem que as *RNAs* podem ser uma alternativa promissora para a *PST*, especialmente para séries temporais com um elevado grau de não linearidade (como no caso da série 1) [NevCor97a]. Comparando o desempenho dos vários métodos, tem-se que o sistema *GANN* apresenta resultados ao nível dos obtidos via ARIMA, superando mesmo estes em alguns casos. A filtragem de dados apresentada melhora a previsão para as séries sazonais, principalmente para as previsões de longo prazo. Porém, nestas séries, e para previsões de curto prazo, o método de Holt-Winters ultrapassa

todos os restantes. Por falta de tempo e de acesso a séries temporais com as características exigidas (serem reais e já objecto de estudo por métodos convencionais), apenas foi possível estudar onze séries. Seria interessante, contudo, estudar o comportamento do sistema desenvolvido em um maior número de séries.

As *RNAs* (e em particular os sistemas *GANN*) são uma forma expedita de efectuar a *PST*, embora se encontrem ainda na *infância*, sendo necessário que atinjam uma maior maturidade, antes que possam ser aplicadas à resolução de problemas de uma forma generalizada. Os sistemas *GANN* têm ainda o senão de serem exigentes em termos computacionais, esperando-se, contudo, que no futuro este problema tenda a desaparecer [NevCor97b]. O modelo Holt-Winters, bastante simples, continua a apresentar-se como o método ideal para a previsão de séries sazonais. Porém, em previsões de longo prazo, e nos casos onde uma melhor previsão seja mandatária, a utilização de sistemas *GANN* pode ser a única solução. Um aspecto positivo do sistema apresentado é que funciona de forma automática, com um mínimo de intervenção humana, ao contrário, por exemplo, do ARIMA, que exige a presença de um perito.

7.2 Perspectivas de Trabalho Futuro

O trabalho desenvolvido mostra acima de tudo que as *RNAs* (e em particular os sistemas *GANN*) podem constituir uma alternativa à *PST*, tendo-se obtido resultados num mesmo patamar, e até melhores em alguns casos, que os dos métodos convencionais (neste caso ARIMA e Holt-Winters). Devido às limitações de tempo, de recursos e à dificuldade em aceder a mais informação, não foi possível ir mais longe. Todavia, importa indicar algum trabalho do que se irá desenvolver num futuro próximo, nomeadamente:

Uma Maior Maturação do Sistema Desenvolvido

O sistema proposto deu bons resultados. Porém, ainda se encontra pouco robusto, pelo menos para ser aplicado à resolução de problemas de uma forma generalizada. O processo de maturação terá de passar pelo teste do sistema com um maior número de séries temporais de forma a que se possam tirar ilações com outro tipo de fundamento e, por outro lado, nos permita afinar parte ou a totalidade dos seus parâmetros. Também será interessante

atender à evolução do sistema em função do tempo; i.e., ver até que ponto uma topologia de uma rede se altera com o andar do tempo.

Um Melhor Uso de Redes Neurais

Neste trabalho usaram-se exclusivamente *RNAs* redes *feedforward*, completamente interligadas e sem ligações de *atalho*. Ora, seria bastante interessante efectuar um estudo mais exaustivo sobre a forma ideal de realizar a *PST* através de *RNAs*, o que passará pela utilização de outro tipo de arquitecturas, nomeadamente redes *feedforward* com ligações de atalho e não completamente interligadas, redes recorrentes ou redes auto-organizáveis como as *RNAs* de Kohonem [Gal93]. Outro aspecto a considerar seria o uso de novos algoritmos de treino (como o *quickprop* [Fah88] ou o *cascade correlation* [FahLeb90]), outras funções de activação (por exemplo funções *radial basis* [MooDar82]), assim como técnicas de simplificação de topologias de rede [LuLiu95].

Uso Mais Elaborado de Sistemas *GANN*

Muitas foram as restrições impostas ao sistema, em particular devido à falta de poder computacional. Como já foi referido, este problema tenderá a desaparecer em parte com o aparecimento de novas tecnologias. Assim, um cromossoma com mais parâmetros, com uma extensão de domínios, permitirá alargar o espaço de soluções. Por outro lado, neste trabalho, e por uma questão de prioridades, não foram estudadas as melhores formas de utilizar os *AGs*. Ora, um aumento da eficiência da procura da solução óptima depende do uso de melhores e mais sofisticados *AGs*, assim como do uso do modelo das ilhas para a distribuição da população [Müh91]. Estes algoritmos facilitam o processo de distribuição da carga computacional, pois usam procedimentos de selecção descentralizados.

Outro aspecto deveras importante tem a ver com a utilização da função de *aptidão*. Como já foi referido, esta questão é crucial para qualquer sistema *GANN*. A solução encontrada, uso do critério MQE_{va} , não parece completamente satisfatória; i.e., dados os resultados de duas previsões, com valores de MQE_{va} diferentes, não se pode garantir que a que apresenta o menor valor de MQE_{va} seja a melhor das previsões. O ideal seria obter uma métrica

que tivesse uma correlação com o erro de previsão muito próxima de 1. Tal objectivo é, porém, difícil de ser atingido. Contudo, existe uma alternativa que poderá ser estudada e que tem a ver com a paragem antecipada do treino das *RNAs*. O problema com esta solução é que uma parte substancial dos dados disponíveis (neste caso 10%) não são efectivamente utilizados no treino. Todavia, actualmente este senão tem vindo a ser ultrapassado através da introdução de novas formas de paragem antecipada, que passam pela divisão dos *casos de treino* em vários conjuntos, treinando a rede com alguns desses conjuntos e verificando o desempenho da aprendizagem com os restantes [KwoYeu95]. A novidade está em que estas técnicas se baseiam numa rotação dos conjuntos que são usados no treino com os conjuntos que são usados para a avaliação.

Diferentes Técnicas de Filtragem

A técnica de filtragem utilizada revelou-se de certo modo útil, principalmente nas previsões de longo prazo. No entanto, trata-se de uma forma simples de filtragem através de uma escolha cuidada dos períodos da série. Técnicas mais elaboradas têm de ser consideradas, de que são exemplo os *filtros de Kalman* [NabLow96] ou a *transformação de onda* [AusMur97].

7.3 O Futuro da *PST*

Nos últimos anos tem havido um certo interesse no que concerne à problemática da *PST*. Existirá um limite para o seu desenvolvimento? Será possível uma solução universal que permitirá fazer previsão acerca de toda e qualquer série temporal? A resposta a estas questões é um claro não. A procura de um procedimento universal de *PST* está relacionado com a visão de Hilbert de reduzir toda a matemática a um conjunto de axiomas, sonho esse que foi terminado por Gödel e depois por Turing [WeiGer94].

Um objectivo mais realista é encontrar o melhor modelo para uma dada série temporal, onde *melhor* significa o modelo que exige menos informação para descrever de forma correcta um dado sistema. O desenvolvimento da informática permitiu uma nova disponibilidade de recursos computacionais, tornando possível um aumento da informação armazenada, a aplicação de algoritmos mais complexos e uma visualização interactiva dos resultados. No

futuro, esperam-se novos desenvolvimentos nos domínios da análise de sistemas não lineares, da informática e das redes neuronais que possam contribuir para uma aproximação a este objectivo [WeiGer94].

Bibliografia

- [AlmFri93a] Almgren, J., Anderson, S., Carlsson, M., Floyd, L., Haridi, S., Frisk, C., Nilson, H. and Sundberg, J. *SICStus Prolog User's Manual*. Swedish Institute of Computer Science, Sweden, 1993.
- [AlmFri93b] Almgren, J., Anderson, S., Carlsson, M., Floyd, L., Haridi, S., Frisk, C., Nilson, H. and Sundberg, J. *SICStus Prolog Library Manual*. Swedish Institute of Computer Science, Sweden, 1993.
- [AusMur97] Aussem, A. and Murtagh, F. Combining Neural Network Forecasts on Wavelet-transformed Time Series. *Connection Science*, vol. 9, no. 1, March, 1997.
- [Azo95] Azoff, E. *Neural Network Time Series Forecasting of Financial Markets*. John Wiley & Sons, New York, USA, 1995.
- [BoxJen76] Box, G. and Jenkins, G. *Time Series Analysis: Forecasting and Control*. Revised Edition, Holden Day, San Francisco, USA, 1976.
- [CarSch88] Caruana, R. and Schaffer, J. Representation and Hidden Bias: Gray vs. Binary coding for Genetic Algorithms. In *Proceedings of the Fifth International Conference on Machine Learning*, Morgan Kaufmann, pp. 153-161, 1988.
- [Cor95] Correia, M. Tecnologias Avançadas de Produção e Gestão das Pessoas. *Comportamento Organizacional e Gestão*. vol. 1, no. 2, pp. 265-284, 1995.
- [CorNev95] Cortez, P., Rocha, M., Machado, J. and Neves, J. A Neural Network Based Forecasting System. In *Proceedings of ICNN'95, 1995 IEEE International Conference on Neural Networks*, Perth, Western Australia, November, 1995.

- [CorNev96a] Cortez, P., Machado, J. and Neves, J. An Evolutionary Artificial Neural Network Time Series Forecasting System. In *Proceedings of IASTED International Conference on Artificial Intelligence, Expert Systems and Neural Networks*, Honolulu, Havai, USA, August 19-22, 1996.
- [CorNev96b] Cortez, P., Machado, J. and Neves, J. Time Series Forecasting in a Distributed Environment. In *Proceedings of IBERAMIA 96, V Congreso Iberoamericano de Inteligencia Artificial*, Cholula, Puebla, Mexico, October - November, 1996.
- [DecFoc95] Decker, K and Focardi, S. Technology Overview: A Report on Data Mining. Technical Report, CSCS-ETH, Swiss Scientific Computing Center, May, 1995.
- [Fah88] Fahlman, S. An Empirical Study of Learning Speed in Back-Propagation Networks. Research Report, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA, 1988.
- [FahLeb90] Fahlman, S. and Lebiere, C. The Cascade-Correlation Learning Architecture. Research Report, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA, 1990.
- [FarCha95] Faraday, J. and Chatfield, C. Time Series Forecasting with Neural Networks: A Case Study. Research Report, Statistics Group, University of Bath, UK, 1995.
- [FinZim93] Finnoff, W., Hergert, F. and Zimmermann, H. Improving model selection by nonconvergent methods. *Neural Networks*, no. 6, pp. 771-783, 1993.
- [FreRip95] Freisleben, B. and Ripper, K. Economic Forecasting Using Neural Networks. In *Proceedings of ICNN'95, 1995 IEEE International Conference on Neural Networks*, Perth, Western Australia, November, 1995.
- [Gal93] Gallant, S. *Neural Network Learning and Expert Systems*. MIT Press, Cambridge, USA, 1993.
- [GanNg95] Gan, W. and Ng, K. Multivariate FOREX Forecasting using Artificial Neural Networks. In *Proceedings of ICNN'95, 1995 IEEE International Conference on Neural Networks*, Perth, Western Australia, November, 1995.

- [GarNev91] Garrido, P. and Neves, J. Mapping a Prolog Subset to Neural Networks, In *Proceedings of NEURO-NIMES'91*, Nimes, France, 1991.
- [GemDou92] Geman, S., Bienenstock, E. and Doursat, R. Neural Networks and the bias/variance dilemma. *Neural Computation*, no. 4, pp. 1-58, 1992.
- [Gol89] Goldberg, D. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, USA, 1989.
- [HanRei89] Hanke, J. and Reitsch, A. *Business Forecasting*. 3rd. Ed., Allyn and Bacon, Massachusett, USA, 1989.
- [KemMar92] Kemsley, D. and Martinez, T. A survey of Neural Network Research and Fielded Applications. In *International Journal of Neural Networks: Research and Applications*, vol. 2, no. 2/3/4, pp. 123-133, 1992.
- [KerRit88] Kernigham, B and Ritchie, D. *The C Programming Language*, Prentice Hall, 1988.
- [Kit90] Kitano, H. Empirical Studies on the Speed of Convergence of Neural Network Training using Genetic Algorithms. In *Proceedings of the International Joint Conference on Neural Networks*, vol. II, pp. 397-404, IEEE, 1990.
- [Koe94] Koen, P. Combining Genetic Algorithms and Neural Networks. Thesis for Master Science Degree, University of Tennessee, Knoxville, USA, 1994.
- [KhoMar95] Khotanzad, A., Abaye, A. and Maratukulam, D. An Adaptive and Modular Recurrent Neural Network Based Power System Load Forecaster. In *Proceedings of ICNN'95, 1995 IEEE International Conference on Neural Networks*, Perth, Western Australia, November, 1995.
- [KwoYeu95] Kwok, T. and Yeung, D. Efficient Cross-Validation for Feed-forward Neural Networks. In *Proceedings of ICNN'95, 1995 IEEE International Conference on Neural Networks*, Perth, Western Australia, November, 1995.
- [LanHin90] Lang, K., Waibel, A. and Hinton, G. A time-delay neural network architecture for isolated word recognition. *Neural Networks*, no. 3, pp. 33-43, 1990.

- [LisSan95] Lisi, F., Nicolis, O. and Sandri, M. Combining Singular-Spectrum Analysis and Neural Networks for Time Series Forecasting. *Neural Processing Letters*, vol. 2, no. 4, July, 1995.
- [LópCel96] López, J., Cembrano, G. and Cellier, F. Times Series Prediction Using Fuzzy Inductive Reasoning: A Case Study. In *Proceedings of ESM 96, European Simulation Multiconference 96*, Budapest, Hungary, June, 1996.
- [LuLiu95] Lu, H., Setiono, R. and Liu, H. NeuroRule: A Connectionist Approach to Data Mining. In *Proceedings of the 21st VLDB Conference*, Zurich, Switzerland, 1995.
- [MakMcG78] Makridakis, S., Wheelwright, S. and McGee, V. *Forecasting: Methods and Applications*. John Wiley & Sons, New York, USA, 1978.
- [Mak82] Makridakis, S. The accuracy of extrapolation (times series) methods: Results of a forecasting competition. *Journal of Forecasting*, no. 1, pp. 111-153, 1982.
- [McC93] McCluskey, P. Feedforward and Recurrent Neural Networks and Genetic Programs for Stock Market and Time Series Forecasting. Thesis for Master Science Degree, Brown University, Rhode Island, USA, 1993.
- [McN82] McNees, S. The Role of Macroeconomic Models in Forecasting and Policy Analysis in the United States. *Journal of Forecasting*, vol. 1, no. 1, pp. 37-48, 1982.
- [MenNev95] Mendes, R. and Neves, J. Genetic Algorithms, Classifiers and Parallelism - An Object Oriented Approach. In *Proceedings of The Second World Congress on Expert Systems*, Lisbon - Estoril, Portugal, January, 1994.
- [MooDar82] Moody, J. and Darken, C. Fast learning in networks of locally-tuned processing units. *Neural Computation*, no. 1, pp. 281-294, 1989.
- [Müh91] Mühlenbein, H. Evolution in Time and Space - The Parallel Genetic Algorithm. In G.Rawlins ed. *Foundations of Genetic Algorithms*, Morgan-Kaufmann, pp. 316-337, San Mateo, 1991.

- [Müh93] Mühlenhein, H. Evolutionary Algorithms: Theory and Applications. Technical Paper, German National Research Center for Computer Science, Germany, 1993.
- [Min90] Minsky, M. Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy. In *Artificial Intelligence at MIT, Expanding Frontiers*, Patrick H. Winston (Ed.), MIT Press, vol. 1, 1990.
- [NabLow96] Nabney, I., McLachlan, A. and Lowe, D. Practical methods of tracking of nonstationary time series applied to real-world data. *Applications and Science of Artificial Neural Networks II*, SPIE - The International Society for Optical Engineering, pp. 152-163, USA, March, 1996.
- [NevCor97a] Neves J. and Cortez, P. Artificial Intelligence in Knowledge Management for Times Series Forecasting. In *Proceedings of ASC'97 - IASTED International Conference on Artificial and Soft Computing*, Canada, 27 July - 1 August, 1997.
- [NevCor97b] Neves J. and Cortez, P. An Artificial Neural-Network Genetic Based Approach for Time Series Forecasting. In *Proceedings of IV Brazilian Symposium on Neural Networks*, Goiania, Brazil, 3-5 December, 1997.
- [NevGar91] Neves J. and Garrido, P. Amalgamating The Neural And Logic Computing Paradigms, In *Proceedings of ICANN-91*, Finland, 1991.
- [PapGot93] Papadourakis, G., Spanoudakis, G. and Gotsias, A. Application of Neural Networks in Short-Term Stock Price Forecasting. In *Proceedings of First International Workshop on Neural Networks in the Capital Markets*, London, UK, November, 1993.
- [PicTom95] Pictet, O., Dacorogna, M., Chopard, B., Oussaidene, M., Schirru, R. and Tomassini, M. Using Genetic Algorithms for Robust Optimization in Financial Applications. *Neural Network World*, vol. 5, no. 4, pp. 573, 1995.
- [Pre94] Prechelt, L. PROBEN1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules. Research Report, Fakultät für Informatik, Universität Karlsruhe Germany, 1994.

- [RieBra93] Riedmiller, M. and Braun, H. A Direct Adaptive Method for Faster Backpropagation Learning: The Rprop Algorithm. In *Proceedings of the IEEE International Conference on Neural Networks*, San Francisco, CA, March 28 - April 1, 1993.
- [Rie94] Riedmiller, M. Advanced Supervised Learning in Multi-Layer Perceptrons - From Backpropagation to Adaptive Learning Algorithms. *Int. Journal of Computer Standards and Interfaces on Neural Networks*, 1995.
- [RusNor95] Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, New Jersey, USA, 1995.
- [SanNev95] Santos, M. and Neves, J. A Distributed Computational Environment for Genetic-Based Classifier Systems, In *Proceedings of The Eight Australian Joint Conference on Artificial Intelligence*, Camberra, Australia, 1995.
- [ShaRam96] Sharda, R. and Rampal, R. Neural Networks and Management Science/Operations Research: A Bibliographic Essay. *Encyclopedia of Library and Information Science*, Marcel Dekker, Inc., 1996.
- [Sho90] Shoneburg, E. Price Prediction using Neural Networks: A Project Report *Neurocomputing*, vol. 2, pp. 17-27, 1990.
- [TanFis91] Tang, Z. and Fishwick, P. Feed-forward Neural Nets as Models for Time Series Forecasting. Technical Report, University of Florida, Gainesville, USA, 1991.
- [ThiVor94] Thiesing, F., Middelberg, U. and Vornberger, O. Parallel Back-Propagation for the Prediction of Time Series. Talk presented: *1st European PVM Users' Group Meeting*, Rome, October, 1994.
- [Ulbr94] Ulbricht, C. Multi-recurrent Networks for Traffic Forecasting. In *Proceedings of AAAI'94 Conference*, Seattle, Washington, USA, July-August, 1994.
- [WeiGer94] Weigend, A. and Gershenfeld, N. Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley, USA, 1994.

- [WheMak80] Wheelwright, S. and Makridakis, S. *Forecasting Methods for Management*. Third Edition, John Wiley & Sons, New York, USA, 1980.
- [Whi93] White, D. *GANNet: A Genetic Algorithm for Searching Topology and Weight Spaces in Neural Network Design*. Dissertation at the University of Maryland, 1993.
- [Whi89] Whitley, D. The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. In *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 116-121, 1989.
- [WinCue95] Winter, G., Périaux, J., Galán, M. and Cuesta, P. *Genetic Algorithms in Engeneering and Computer Science*, John Wiley & Sons, England, 1995.
- [YaoPoh95] YAO, J. and POH, H. Forecasting the KLSE Index Using Neural Networks. In *Proceedings of ICNN'95, 1995 IEEE International Conference on Neural Networks*, Perth, Western Australia, November, 1995.
- [Zar84] Zarnowitz, V. The Accuracy of Individual and Group Forecasts from Business Outlook Surveys *Journal of Forecasting*, vol. 3, no. 1, pp. 11-26, 1984.

Apêndice

Artigos Publicados de Acordo com o Trabalho Desenvolvido nesta Dissertação

- Cortez, P., Machado, J. and Neves, J. An Evolutionary Artificial Neural Network Time Series Forecasting System. In *Proceedings of IASTED International Conference on Artificial Intelligence, Expert Systems and Neural Networks*, Honolulu, Havai, USA, August 19-22, 1996.
- Cortez, P., Machado, J. and Neves, J. Time Series Forecasting in a Distributed Environment. In *Proceedings of IBERAMIA 96, V Congresso Iberoamericano de Inteligencia Artificial*, Cholula, Puebla, Mexico, October - November, 1996.
- Neves J. and Cortez, P., Artificial Intelligence in Knowledge Management for Times Series Forecasting. In *Proceedings of ASC'97 - IASTED International Conference on Artificial and Soft Computing*, Canada, 27 July - 1 August, 1997.
- Neves J. and Cortez, P., An Artificial Neural-Network Genetic Based Approach for Time Series Forecasting. In *Proceedings of IV Brazilian Symposium on Neural Networks*, Goiania, Brazil, 3-5 December, 1997.