

REDES NEURAIS ARTIFICIAIS E PREVISÃO DE SÉRIES ECONÔMICAS: UMA INTRODUÇÃO¹

Marcelo S. Portugal²
Luiz Gustavo L. Fernandes³

RESUMO

O objetivo central deste artigo é introduzir um novo método de previsão de séries de tempo, baseado em uma modelagem do cérebro humano. Esta modelagem consiste na criação de Redes Neurais Artificiais (RNA), que têm sido usadas com grande sucesso em diferentes áreas do conhecimento. O cérebro humano é extremamente eficaz no reconhecimento de padrões e regularidades, sendo ainda capaz de generalizar com base no conhecimento acumulado. As RNA incorporam estas qualidades do cérebro sendo, portanto, capazes de realizar previsões. Características particulares de uma série de tempo, tais como sazonalidade, tendência e ciclo, podem ser aprendidas por uma RNA, possibilitando, assim, a realização de previsões. Aplicações práticas deste método à previsão de séries econômicas podem ser encontradas em Portugal (1995) e Fernandes, Portugal e Navaux (1994).

1 INTRODUÇÃO

Muito embora o computador seja considerado uma invenção moderna, sua idealização remonta à época vitoriana. O matemático inglês Charles Babbage talvez tenha sido o primeiro a idealizar conceitualmente a máquina que hoje chamamos de computador. Contratado pela *Royal Astronomical Society* para a elaboração de tabelas astronômicas, Babbage concentrou-se em desenvolver um método de “computação” dos resultados que fosse ao mesmo tempo rápido e sem possibilidade de erro. Inicialmente, Babbage

-
- 1 Os autores agradecem o auxílio dos bolsistas Suzana M. B. Coccaro, Vinícius Carrasco e a César A. S. Follmann.
 - 2 Professor do Curso de Pós-Graduação em Economia, Universidade Federal do Rio Grande do Sul e pesquisador do CNPq (MSP@VORTEX.UFRGS.BR).
 - 3 Pesquisador do Instituto de Informática da Universidade Federal do Rio Grande do Sul.

dedicou-se à construção de uma máquina mecânica de calcular, a qual denominou “máquina de diferenças”, visando, assim, a substituição dos “computadores humanos” utilizados até então na confecção de tabelas de cálculos. Contudo, ele rapidamente vislumbrou a possibilidade de extensão do seu projeto inicial, partindo para a construção de uma máquina ainda mais sofisticada capaz de resolver vários tipos de problemas. Babbage pretendia a construção de uma “máquina universal programável”, capaz, portanto, de realizar diferentes tarefas. Ele concluiu que tal máquina deveria ser composta de três partes: uma unidade de processamento, uma unidade de armazenamento e uma unidade de entrada de instruções. Em linguagem moderna, as três unidades propostas por Babbage correspondem à Unidade de Processamento, à memória e aos programas.

Infelizmente, a tecnologia disponível no final do século XIX não permitiu a construção da máquina universal idealizada por Babbage. Apenas em 1941, através da utilização de relés, que o alemão Konrad Zuse foi capaz de construir algo semelhante à máquina universal idealizada por Babbage. Durante a Segunda Guerra Mundial, o trabalho de Zuse foi praticamente interrompido, pois não foi considerado prioritário para o esforço de guerra alemão. Foi na Inglaterra e nos Estados Unidos que surgiram os primeiros computadores construídos utilizando-se válvulas.

Uma contribuição significativa para o desenvolvimento da ciência da computação foi dada por outro matemático inglês chamado Alan M. Turing. Em seu artigo *On Computable Numbers, with an Application to the Entscheidungs Problem*, publicado em 1936, Turing demonstrou, matematicamente, a possibilidade da existência de uma máquina capaz de desenvolver qualquer tarefa lógica realizada pelo cérebro humano (Turing, 1936).

Turing foi também um pioneiro no campo da inteligência artificial. Inspirado por seu trabalho durante a Segunda Guerra Mundial, quando criou um computador para decifrar o código secreto alemão, Turing acreditava ser possível a construção de um computador dotado de inteligência, isto é, um computador capaz de “pensar”. Para determinar a capacidade de pensamento do computador, Turing sugeriu a aplicação de um teste simples, hoje conhecido como teste de Turing. Segundo este teste, um computador ganharia o título de máquina pensante se conseguisse utilizar linguagem de forma inteligente, isto é, se conseguisse levar um ser humano, com o qual esteja se comunicando, a acreditar que está se comunicando com um outro ser humano.

O rápido desenvolvimento dos computadores levou os pesquisadores, na área de inteligência artificial, a acreditar que a construção de um computador capaz de pensar fosse uma tarefa relativamente simples. A prática foi, contudo, bem diferente. Os pioneiros nesta área costumavam separar o cérebro da mente. O cérebro era visto apenas como o meio físico de

atuação da mente, esta sim responsável pelo pensamento. A analogia entre cérebro/mente e *hardware/software* é imediata. A mente era vista como um *software* que “roda” em um *hardware* chamado cérebro.

O ponto central, portanto, era desenvolver *softwares* que representassem a atuação da mente, sendo a modelagem do cérebro considerada secundária neste processo. Acreditava-se não ser necessário imitar a natureza para obter resultados semelhantes aos gerados por esta. Assim como foi possível construir-se máquinas que voam sem bater as asas, seria também possível criar uma máquina de pensar sem a necessidade de criação de um cérebro artificial. Contudo, os fracassos sucessivos desta abordagem acabaram por mostrar a importância da modelagem do cérebro na criação de uma inteligência artificial. A partir de então, a área da inteligência artificial começou a desenvolver-se no sentido de modelar o cérebro através da criação de redes neurais artificiais (RNA), que possuem as mesmas propriedades cognitivas e associativas do cérebro humano.

Além desta introdução, este artigo conta com mais três seções, onde discutimos, separadamente, o modelo de RNA em formato genérico, o modelo perceptron multi-camada que é o mais apropriado ao problema de previsão de séries temporais e à interpretação das RNA como um tipo de modelo econométrico de previsão.

2 REDES NEURAIS ARTIFICIAIS: CONCEITOS BÁSICOS

As Redes Neurais Artificiais fundamentam-se nos estudos sobre a estrutura do cérebro humano para tentar emular sua forma inteligente de processar informação. Alguns estudos da neurofisiologia consideram que a riqueza computacional do cérebro humano está associada ao grande número de neurônios, interconectados por uma rede complexa de sinapses (Carvalho, 1988).

Estima-se que a quantidade de neurônios existentes no cérebro humano esteja na casa dos bilhões. Contudo, a velocidade de processamento destes componentes é relativamente baixa, quando comparada aos computadores tradicionais. Esta deficiência na velocidade de processamento dos neurônios é superada pela imensa quantidade de neurônios existentes operando de forma paralela (Simpson, 1990). Estima-se que existam cerca de 10^{11} a 10^{14} neurônios operando em paralelo no cérebro humano. Cada um destes está conectado através de 10^3 a 10^4 sinapses, em média (Cottrell, 1985).

Tais características permitem ao cérebro humano executar rapidamente certas funções (por exemplo, reconhecer fisionomias e sons) que

os computadores convencionais não conseguem realizar com o mesmo desempenho. No Quadro 1 é apresentada uma comparação das principais diferenças existentes entre os computadores tradicionais e o cérebro humano. Esta comparação fornece uma idéia mais clara sobre a capacidade adaptativa do cérebro humano, em contraste com a rigidez e precisão dos computadores convencionais.

Quadro 1

DIFERENÇAS ENTRE O COMPUTADOR E O CÉREBRO HUMANO

	COMPUTADOR TRADICIONAL	CÉREBRO HUMANO
Elementos computacionais	processadores poderosos	neurônio simples
Velocidade de processamento	10^{-9} segundos	10^{-3} segundos
Tipo de processamento	serial	paralelo
Confiabilidade dos elementos	confiável	não-confiável
Tolerância a falhas	quase nenhuma	grande
Tipo de sinal	preciso, simbólico	impreciso
Tipo de controle	centralizado	distribuído
Armazenamento de informação	substituível	adaptável

Fonte: Cottrell (1985) e Simpson (1990).

Uma definição possível de modelos conexionistas de computação, também chamados de redes neuronais artificiais (RNAs), redes neurais ou sistemas de processamento paralelo distribuído (PDP), consiste em encará-los como

“uma estrutura de processamento de informação distribuída e paralela. Ela é formada por unidades de processamento, comumente chamadas de nós, neurônios ou células, interconectadas por arcos unidirecionais, também chamados de ligações, conexões ou sinapses. Os nós possuem memória local e podem realizar operações de processamento de informação localizada. Cada célula possui uma única saída (axônio), a qual pode se ramificar em muitas ligações colaterais (cada ramificação possuindo o mesmo sinal de saída do neurônio). Todo o processamento que se realiza em cada unidade deve ser completamente local, isto é, deve depender apenas

dos valores correntes dos sinais de entrada que chegam dos neurônios através das conexões. Estes valores atuam sobre os valores armazenados na memória local da célula.” (Hecht-Nielsen, 1988, p. 2-3).

Os principais elementos utilizados na descrição de RNAs são a representação distribuída, as operações locais e o processamento não-linear (Simpson, 1990). Estes atributos especificam duas aplicações básicas das RNAs: situações onde poucas decisões têm que ser tomadas a partir de uma grande quantidade de dados e situações onde um complexo mapeamento não-linear deve ser aprendido.

Nesta seção é feita uma revisão sobre as RNAs. Serão descritos os componentes básicos destas redes, bem como suas interrelações.

2.1 Componentes das RNAs

De acordo com Rumelhart (1986), um modelo conexionista pode ser descrito por oito elementos principais:

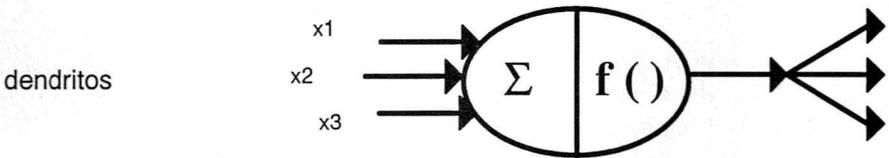
- um conjunto de unidades de processamento;
- um estado de ativação;
- uma função saída;
- um padrão de interconexão;
- uma regra de propagação;
- uma regra de ativação;
- uma regra de aprendizado;
- um ambiente onde o sistema deve funcionar.

i) Unidade de Processamento

A unidade de processamento (neurônio) é o componente básico das RNAs e corresponde ao neurônio humano. Na Figura 1, apresentamos uma ilustração de um neurônio como unidade que representa um limite a ser ultrapassado. As entradas que chegam a ele representam os dendritos. Cada dendrito possui um sinal (x) que é adicionado (Σ). Depois da adição, este sinal é processado através da função que representa o limite ou função de ativação $f(\)$, a qual produz um sinal de saída. Nesta figura, o neurônio pode ser considerado como uma representação simplificada dos neurônios biológicos.

Figura 1

NEURÔNIO COMO UNIDADE LIMIAR



Neste trabalho, as unidades de processamento da rede são designadas pela letra u , seguida de um índice i que indica a posição que o neurônio ocupa na rede. Cada neurônio u_i da rede calcula um estado de ativação, que é um valor numérico líquido de saída. O cálculo desta ativação é realizado a partir dos sinais das saídas dos demais neurônios conectados diretamente a este neurônio, dos correspondentes pesos destas conexões e da função $f()$.

ii) Estado de Ativação

O estado de ativação de todos os neurônios da rede, ou seja, o estado de ativação do sistema, especifica o que está sendo representado na rede em um determinado instante t qualquer. Os valores das ativações existentes na rede podem ser discretos, por exemplo assumindo os valores $\{0,1\}$ ou $\{-1,0,1\}$, como também podem ser contínuos, assumindo valores no intervalo $[0,1]$ ou $[-1,1]$, que são calculados pela regra de ativação (Lippman, 1987; Gallant, 1988).

iii) Função de Saída

Os neurônios interagem entre si através de um valor que é transmitido pelas conexões. Este valor é determinado pela ativação do neurônio estimulador. A função de saída é o mapeamento do estado de ativação num sinal de saída. A função de saída pode ser a função identidade, isto é, o sinal de saída é igual ao estado de ativação. Em muitos casos, a função de saída é uma função porteira (patamar, limiar) que só emite sinal quando o seu estado ultrapassa um certo limite.

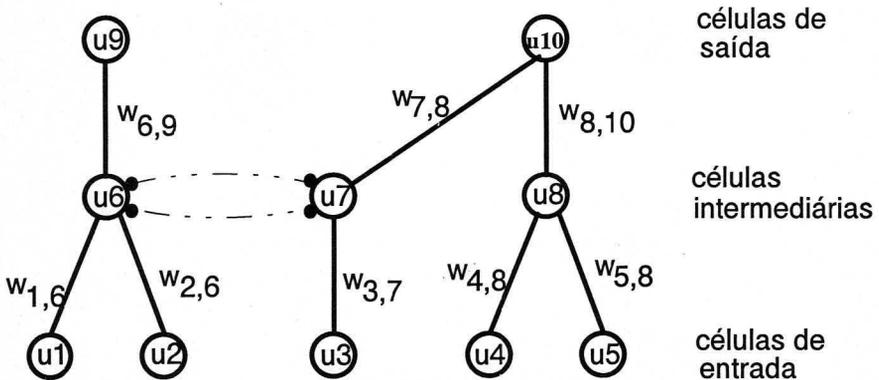
iv) Padrão de Interconexão

Pode-se representar o padrão de interconexão da rede por uma matriz de pesos w , onde um elemento $w_{i,j}$ corresponde à influência do neurônio u_i sobre o neurônio u_j . Conexões com pesos positivos, chamadas de excitatórias, indicam o reforço na ativação do neurônio u_j . Conexões com pesos negativos, chamadas de inibitórias, indicam inibição na ativação do neurônio u_j . O conjunto das ligações excitatórias e inibitórias existentes na rede determina o comportamento da mesma.

Topologicamente, as RNAs podem ser organizadas em camadas. A camada de entrada da rede não recomputa suas saídas, ela é usada somente para a entrada dos dados. Portanto, não existem arcos de entrada em suas células. Os valores resultantes das células pertencentes à camada de saída são considerados os resultados finais da rede como um todo. Na Figura 2, os neurônios u_9 e u_{10} são considerados as saídas da rede. Os neurônios que não pertencem nem à camada de entrada e nem à de saída são chamadas de intermediários ou ocultos.

Figura 2

EXEMPLO DE RNA



Observando a Figura 2 quando u_{10} é ativado o valor de sua ativação é determinado pelas ativações de u_3, u_4, u_5, u_7 e u_8 e os pesos $w_{4,8}, w_{5,8}, w_{3,7}, w_{7,8}$ e $w_{8,10}$. Podem existir conexões que ligam neurônios da mesma camada, sendo chamados de sinapses intra-camadas. As ligações inter-camadas conectam células de camadas diferentes. No caso da Figura 2, as conexões que unem os neurônios u_6 e u_7 são sinapses intra-camadas inibitórias. A ligação que conecta o neurônio u_7 ao u_{10} é uma inter-camada excitatória.

Com relação às conexões que ligam os neurônios u_6 e u_7 elas são chamadas de ligações recorrentes. Estas conexões formam um ciclo, voltando para o neurônio de origem (aquele que foi ativado primeiro). Podem existir conexões que liguem os neurônios de diferentes camadas de modo a formar ciclos entre as mesmas. Por exemplo, uma ligação que conectasse o neurônio u_{10} a u_5 formaria um outro ciclo na rede da Figura 2. Portanto, as RNAs podem ser classificadas em redes cíclicas ou acíclicas, em função da existência ou não de ligações cíclicas (Lippman, 1987).

v) Regra de Propagação

Cada neurônio u_i calcula sua nova ativação através de uma regra de propagação. Em geral, ela é definida como sendo uma função somatório da entrada líquida dos pesos (*net*) dos neurônios u_j que estão diretamente conectados a u_i . Isto é feito através da multiplicação do estado do j -ésimo neurônio (u_j) pelo peso da conexão $w_{i,j}$, do i -ésimo para o j -ésimo neurônio, para cada um dos j neurônios que estão conectados à entrada do neurônio i . A regra de propagação se completa através do limite que deve ser ultrapassado pelo somatório explicado acima. Este limiar, que pode ser nulo inclusive, deve ser superado para que ocorra a ativação da célula.

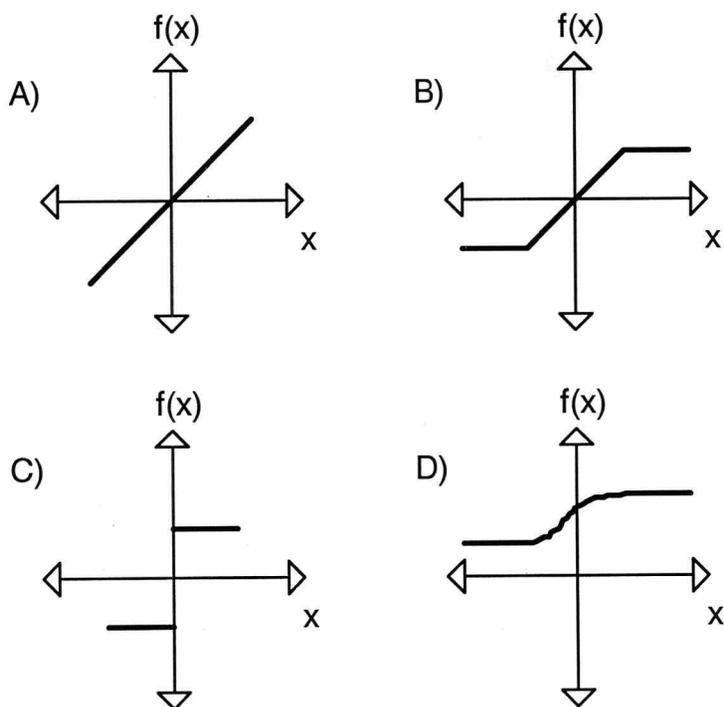
vi) Regra de Ativação

É necessária uma regra que calcule o valor de ativação de um neurônio no instante t . É preciso uma função f que calcule a nova ativação $a(t)$ utilizando as entradas líquidas (*net*). Geralmente, esta função possui a forma $a_i(t+1) = f[a_i(t), net_i(t)]$, onde f é a função de ativação, também chamada de função limiar. Esta função mapeia os neurônios de entrada para um intervalo pré-especificado de saída. As quatro funções de ativação mais utilizadas são linear, rampa, degrau e sigmóide (Simpson, 1990), representadas na Figura 3.

A Figura 3D e a equação abaixo apresentam a função sigmóide ou função logística, cujos pontos de saturação são 0 e 1.

$$S(x) = \frac{1}{1 + e^{-x}}$$

Figura 3
EXEMPLOS DE FUNÇÕES DE ATIVAÇÃO



vii) Regra de Aprendizado

A modificação do processamento ou da estrutura de conhecimento de uma rede neural envolve a alteração do seu padrão de interconexão (Rumelhart, 1986). Em princípio, isto pode ser feito de três maneiras:

- desenvolvimento de novas conexões;
- perda de conexões existentes na rede;
- modificação dos pesos das conexões já existentes.

Quando o padrão de interconexão for uma matriz de pesos W , os dois primeiros itens podem ser simulados através do último. Tomando-se uma ligação com peso zero, e modificando-a para um valor positivo ou negativo, equivale a desenvolver esta conexão. Da mesma forma, alterar o peso de uma conexão para zero significa desconectá-la. Portanto, as regras de aprendizado alteram os pesos das conexões das redes através da experiência.

Em geral, as regras de aprendizado podem ser consideradas como uma variante da Regra de Hebb (Hebb, 1949), que estabeleceu o princípio da alteração da eficiência das conexões como base do aprendizado. Segundo o postulado,

“quando o neurônio A se encontra próximo do neurônio B de forma a poder excitá-lo, e o faz repetidas vezes, algum processo desconhecido provoca o crescimento de conexões entre as células A e B, facilitando assim a excitação de B por A”.

Especificamente, se um neurônio u_j recebe uma entrada de outro u_i , e ambos estão fortemente ativos, o peso w_{ij} (de u_i para u_j) deve ser fortalecido.

Devido à sua importância, a questão do aprendizado no contexto das RNAs será tratada em separado na próxima seção.

viii) Ambiente

O último componente de RNAs é o ambiente onde a rede deve funcionar. É necessário especificar a natureza do ambiente, estabelecendo os possíveis padrões de entrada e saída. Em alguns modelos, por exemplo o PDP de Rumelhart, o ambiente é representado como uma função estocástica que varia ao longo do tempo sobre um espaço de padrões de entrada. Geralmente, o ambiente é caracterizado como uma distribuição de probabilidade estável sobre um conjunto de padrões de entrada. Esta distribuição pode ser independente, ou não, de entradas ou de respostas passadas do ambiente.

3 APRENDIZADO EM REDES NEURAIS

Nesta seção, realizamos uma análise das técnicas de aprendizado utilizadas nos modelos conexionistas e apresentamos algumas considerações sobre os principais modelos clássicos encontrados na literatura, destacando-se suas propriedades mais relevantes.

3.1 Diferentes tipos de Aprendizado

Na sua essência, o conceito de aprendizado envolve mudança associada a aperfeiçoamento (Pessoa, 1990). Carbonell (1989) define o conceito de aprendizado, dentro do campo da inteligência artificial, como a habilidade de realizar tarefas novas que não podiam ser realizadas anterior-

mente, ou melhorar a realização de tarefas antigas, como resultado de mudanças produzidas pelo processo de aprendizado.

Uma classificação, apresentada por Lippman (1987), relacionou as redes quanto ao tipo de controle realizado durante o aprendizado (supervisionado ou não-supervisionado). No supervisionado existem, basicamente, rótulos que especificam a classe correta para os padrões de entrada, enquanto que no aprendizado não-supervisionado não existem tais rótulos.

Diversas técnicas de aprendizado podem ser utilizadas nas RNAs (Simpson, 1990), entre elas destacamos as mais relevantes:

- **Correção de erros:** aprendizado supervisionado que ajusta os pesos das conexões entre nós, na proporção da diferença entre os valores desejados e computados de cada neurônio da camada de saída;
- **Reforço:** aprendizado supervisionado onde os pesos são recompensados quando o sistema executa ações apropriadas e punidos caso contrário;
- **Regra de Hebb:** aprendizado onde o ajuste dos pesos das conexões é realizado em função da relação de valores dos dois neurônios que ela conecta. Pode ser aplicado tanto ao aprendizado supervisionado quanto ao aprendizado não-supervisionado.

3.2 Principais Modelos de Redes Neurais

Existem muitos modelos conexionistas, e uma quantidade razoável de publicações que se dedicam a classificá-los (Lippman, 1987; Hinton, 1989; Simpson, 1990). Dentre estes modelos, foram selecionados cinco que são apresentados a seguir em conjunto com os pesquisadores que os desenvolveram.

- Perceptron – proposto por F. Rosenblatt em 1957;
- Perceptron de Multi-Camadas (*backpropagation*) – proposto por P. Werbos, D. Parker, D. Rumelhart em 1974;
- Classificador de Carpenter-Grossberg (Sistema ART) – proposto por G. Carpenter, S. Grossberg em 1978;
- Rede de Kohonen – proposto por T. Kohonen em 1980;
- Rede de Hopfield – proposto por J. Hopfield em 1982.

i) Perceptron

O perceptron de duas camadas que pode ser usado com valores contínuos foi o primeiro modelo conexionista desenvolvido. Suas principais características são:

- **Classe de tarefas:** reconhecimento de padrões (ex.: caracteres impressos);
- **Propriedades dos neurônios:** possuem entradas binárias e saídas que assumem os valores +1 ou -1. A função de ativação é a função degrau;
- **Propriedades das redes:** rede acíclica de duas camadas;
- **Aprendizado:** utiliza a técnica de reforço.

Esta rede gerou muito interesse pela habilidade de aprender a reconhecer padrões linearmente separáveis. Contudo, como a grande maioria dos problemas práticos relevantes não são linearmente separáveis, o perceptron não tem uso generalizado (Simpson, 1990).

ii) Perceptrons multi-camadas (*backpropagation*)

Foram desenvolvidos no começo da década de setenta. São redes acíclicas com uma ou mais camadas de neurônios intermediários entre as camadas de entrada e saída. Um algoritmo capaz de treinar os perceptrons multi-camadas é o *backpropagation* (Rumelhart, 1986). Suas principais características são:

- **Classe de tarefas:** reconhecimento de padrões (ex.: controle adaptativo de braços de robôs) e processamento de fala;
- **Propriedades dos neurônios:** são do tipo perceptron e possuem valores contínuos. A função de ativação é a sigmóide;
- **Propriedades das redes:** rede acíclica de três camadas, no mínimo;
- **Aprendizado:** utiliza técnica de Correção de Erros com o uso da regra delta generalizada.

O algoritmo *backpropagation* foi testado em uma série de problemas clássicos, e em problemas relacionados com reconhecimento de padrões visuais. Na maioria dos casos, ele encontrou boas soluções para os problemas propostos, apesar do algoritmo, às vezes, fornecer uma configuração de pesos correspondente a um mínimo local da função erro. Isto é devido ao fato do *backpropagation* utilizar o método do gradiente e, a princípio, a superfície de erro possuir uma forma qualquer.

iii) Classificador de Carpenter-Grossberg (Sistema ART)

No final da década de setenta, Carpenter e Grossberg (Grossberg, 1986), projetaram uma rede capaz de formar aglomerados de informações (*clusters*), e de ser treinada sem supervisão, chamada de sistema ART.

- **Classe de tarefas:** reconhecimento de padrões (ex.: reconhecimento de sinais de radar ou sonar) e processamento de imagens;
- **Propriedades dos neurônios:** possuem entradas binárias, podendo assumir valores contínuos. A função de ativação é a sigmóide;
- **Propriedades das redes:** rede cíclica de três camadas;
- **Aprendizado:** utiliza técnica de aprendizado competitivo e cooperativo que foi introduzida pelo próprio Grossberg.

O sistema ART começa com o algoritmo principal selecionando a primeira entrada como um exemplo para o primeiro aglomerado. A entrada seguinte é comparada com este primeiro exemplo. Ela é agrupada com o mesmo, se a distância para o primeiro for menor que um certo limite, chamado de limiar de vigilância. Caso contrário, este exemplo formará um novo aglomerado. Este processo se repete para todas as entradas existentes. O número de aglomerados cresce em função do limiar e da métrica da distância usada para comparar os exemplos de entrada dos aglomerados.

iv) Rede de Kohonen

No começo da década de oitenta, Kohonen(1982) propôs uma rede onde se corroborou os estudos teóricos sobre a organização dos caminhos de sensoriamento na mente. Suas principais características são:

- **Classe de tarefas:** reconhecimento de padrões (ex.: reconhecimento da fala) e aprendizado da distribuição de probabilidades dos dados (ex.: auto-organização de mapas de características);
- **Propriedades dos neurônios:** possuem entradas contínuas. A função de ativação é a sigmóide;
- **Propriedades das redes:** rede cíclica de duas camadas;
- **Aprendizado:** utiliza técnica de SCA, que foi introduzida pelo próprio Kohonen.

Segundo esta teoria, o cérebro humano foi considerado como uma coleção estruturada de neurônios. Com isto, foi admitida uma ordem espacial das unidades de processamento que permitiu elaborar uma rede neural dotada de mecanismos que permitem formar representações estruturadas dos estímulos de entrada. Após o aprendizado, as unidades respondem a diferentes estímulos de maneira ordenada, formando um sistema de coordenadas de características sobre a rede.

v) Rede de Hopfield

Também na década de oitenta, o trabalho de Hopfield (1982) contribuiu substancialmente para o ressurgimento das pesquisas em RNAs. A rede de Hopfield são mais apropriadas quando representações binárias permitem modelar a situação desejada. Por exemplo, imagens em preto e branco, onde os elementos de entrada podem ser representados pelos valores de cada ponto da imagem (0 = branco, 1 = preto) (Lippman, 1987). Suas características são:

- **Classe de tarefas:** reconhecimento de padrões (ex.: reconhecimento de dados ou imagens completas a partir de fragmentos) e memória associativa;
- **Propriedades dos neurônios:** possuem entradas binárias e saídas que assumem os valores +1 ou -1. A função de ativação é a sigmóide;
- **Propriedades das redes:** rede cíclica de uma camada;
- **Aprendizado:** os padrões são armazenados no começo.

A Rede de Hopfield possui duas limitações quando usada como memória de acesso por conteúdo. Primeiro, apesar dos padrões armazenados, a rede pode convergir para um novo padrão diferente dos padrões exemplo existentes. Isto pode produzir uma situação em que a rede não casa com um padrão já existente. Uma segunda limitação é que o padrão exemplo será considerado instável se ele compartilhar muitos *bits* com outros padrões exemplo, o que pode ocasionar uma convergência da rede para este outro exemplo.

3.3 O modelo perceptron multi-camadas e seu algoritmo de aprendizado

Como já vimos anteriormente, o algoritmo de retropropagação é utilizado para o treinamento do modelo dos perceptrons multi-camadas. Nesta seção, apresentamos o algoritmo de retropropagação (*backpropagation*) o qual é utilizado para a realização de prognósticos de valores futuros de séries temporais.

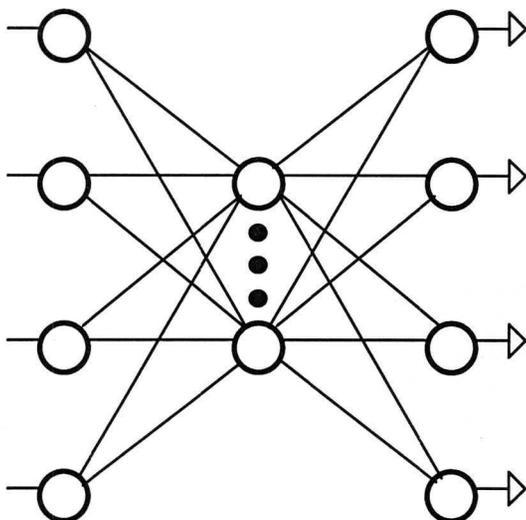
i) Características Básicas

O modelo dos perceptrons multi-camadas com aprendizado baseado no algoritmo de retropropagação é o paradigma mais utilizado em áreas como reconhecimento de padrões, processamento de fala e na previsão de séries temporais.

A base teórica deste modelo foi apresentada originalmente em 1974 por Paul Werbos, David Parker e David Rumelhart. Em 1986 Rumelhart e McClelland publicaram o livro “*Parallel Distributed Processing*”, cujo oitavo capítulo é inteiramente dedicado a análises e aprendizado deste modelo.

O modelo do perceptron multi-camadas utiliza uma topologia de três ou mais camadas. As conexões entre as unidades são do tipo intercadas e são direcionadas da camada de entrada para a camada de saída (Figura 4).

Figura 4
TOPOLOGIA GENÉRICA DO PERCEPTRON MULTI-CAMADAS



No modelo dos perceptrons multi-camadas, entre uma camada e outra, existe uma matriz de pesos. A regra de propagação é a combinação entre as saídas de cada unidade e a matriz de pesos. Ela é realizada através da soma ponderada de cada sinal que chega, via conexões, pelo respectivo peso. O estado de ativação assume valores contínuos e devido a isto a regra de ativação das unidades utiliza como função de ativação uma função do tipo sigmóide.

A função sigmóide também se faz necessária pela regra de aprendizado que utiliza uma função de ativação contínua, não-decrescente e diferenciável.

ii) Aprendizado

Antes de enunciar o procedimento para o aprendizado do modelo perceptron multicamada faz-se necessário estabelecer que:

- **Regra delta:** é o método de treinamento para redes sem camada escondida utilizado no modelo do perceptron;
- **Regra delta generalizada:** procedimento de aprendizado por minimização de erro. Consiste da aplicação do método do gradiente decrescente ao problema de múltiplas camadas;
- **Retropropagação:** propagação do erro gerado na camada de saída para trás na rede. Utiliza a regra da cadeia no cálculo da derivada do erro em relação aos pesos.

O aprendizado representado pelo algoritmo de retropropagação é do tipo supervisionado. Ou seja, apresenta pares de entrada e saída. Utiliza o vetor de entrada e produz sua própria saída, comparando-a com a saída alvo (real). Se não houver diferença, não há mudanças. Caso contrário, os pesos serão modificados com o objetivo de reduzir esta diferença. A regra de modificação dos pesos para um determinado padrão p é dada pela seguinte fórmula:

$$\Delta_p w_{i,j} = \eta \cdot \delta_{p,j} \cdot \theta_{p,i}$$

onde o padrão p corresponde ao conjunto de entradas da rede. No caso de séries temporais, por exemplo, p corresponderá aos valores que influenciarão na realização do prognóstico do próximo valor da série. O termo η representa uma constante de proporcionalidade, também conhecida como taxa de aprendizado. O termo $\delta_{p,j}$ representa a diferença entre a saída alvo e a saída da rede (erro local da unidade j para o padrão p). É esse o termo que será retropropagado para as camadas anteriores e será calculado utilizando-se o algoritmo do gradiente descendente. E finalmente o termo $\theta_{p,i}$ é a saída produzida pela rede para o padrão p na conexão i .

iii) Algoritmo *Backpropagation*

Este algoritmo apresenta duas fases bem identificadas. A primeira fase é a responsável pelo processo de treinamento da rede. É onde ocorre o ajuste dos pesos das conexões entre os neurônios. A segunda fase representa a validação da fase anterior, ou seja é onde ocorre a verificação se o treinamento foi satisfatório através da comparação de saídas geradas pela rede com valores reais.

A fase de treinamento da rede apresenta os seguintes passos:

- iniciar os pesos e tendências com valores aleatórios dentro de certo intervalo;
- apresentar o vetor de entrada com a respectiva saída desejada: $x_p = (x_0, x_1, \dots, x_n)$ e t_p ;
- calcular a saída

$$Y_{p,j} = \sum_{i=0}^{n-1} w_{i,j} \theta_{p,i}$$

em cada camada e o valor $\theta_{p,j}$ na última camada, sendo $\theta_{p,i}$, na camada inicial, igual a x_i .

- adaptar pesos, começando pela camada de saída,

$$w_{i,j}(t+1) = w_{i,j}(t) + \alpha \cdot \delta_{p,j} \cdot \theta_{p,i},$$

onde $w_{i,j}(t)$ é o peso da unidade i para a unidade j no tempo t . O algoritmo do gradiente descendente requer apenas que a modificação dos pesos seja proporcional ao negativo da derivada do erro relativo ao padrão p com relação a cada peso. Esta constante de proporcionalidade é a taxa de aprendizado. Valores altos para esta constante implicam grandes modificações nos pesos. Na prática, a taxa de aprendizado deve ser tomada tão grande quanto possível para acelerar a convergência do algoritmo, mas deve-se evitar as oscilações. O valor ótimo de depende do problema. Para superfícies de erro do tipo planície, um valor maior de resultará numa convergência mais acelerada, enquanto que para superfícies íngremes poderá gerar oscilações. Finalmente, o termo $\delta_{p,j}$ representa o erro para o padrão p na unidade j . Na unidade de saída o erro é calculado por:

$$\delta_{p,j} = (t_{p,j} - \theta_{p,j}) \cdot \theta_{p,j} \cdot (1 - \theta_{p,j});$$

- enquanto que para as unidades das camadas intermediárias (escondidas), o erro é calculado por:

$$\delta_{p,j} = \theta_{p,j} \cdot (1 - \theta_{p,j}) \cdot \sum_k \delta_{p,k} w_{k,j};$$

- apresentar nova entrada até que todos os padrões sejam apresentados e o erro esteja tão pequeno quanto se deseja.

A segunda fase ocupa-se da validação do treinamento Nesta fase deve-se conservar na rede o último conjunto de pesos resultante da fase de treinamento e apresentar vetores de entrada sem valor de saída associado. O valor de saída deverá ser calculado pela rede.

iv) Problemas no Aprendizado

O maior problema da utilização do algoritmo de retropropagação ocorre quando a rede neural obtém uma solução que corresponde a um mínimo local da função do erro. Existem diversas alternativas que visam a diminuir a ocorrência de mínimo local. As mais utilizadas são:

- diminuição da taxa de aprendizado: se a taxa de alteração dos pesos é diminuída progressivamente, então o algoritmo do gradiente decrescente está apto a alcançar uma solução melhor (Beale, 1990).
- adição de unidades internas: é possível considerar que quando um mínimo local é apresentado como solução, duas classes diferentes são classificadas como a mesma. Para evitar isso, ao serem colocadas mais unidades na camada escondida, a rede fará um reconhecimento melhor, diminuindo a ocorrência de mínimo local.
- introdução de um termo de momento: termo introduzido para que se possa aumentar a taxa de aprendizado sem que ocorra oscilação:

$$\Delta w_{i,j}(t+1) = \alpha \cdot \delta_{p,j} \cdot \theta_{p,i} + \beta \cdot [(w_{i,j}(t) - w_{i,j}(t-1))]$$

onde α é o fator de momento ($0 < \beta < 1$) e determina o efeito da modificação do peso passado na direção do espaço de pesos. O termo de momento atua também no aumento da velocidade de convergência. Em muitas simulações realizadas por Rumelhart (1986) o termo de momento assume o valor 0.9.

Outro problema enfrentado pela retropropagação é a quebra de simetria, ou seja, se os pesos começam todos com o mesmo valor e se a solução só pode ser alcançada com pesos diferentes, o sistema não poderá aprender (Rumelhart, 1990). Isto ocorre porque o erro é retropropagado em proporção ao valor dos pesos, o que significa que todas as unidades escondidas conectadas diretamente às unidades de entrada e saída cometem erros iguais. Como a modificação dos pesos depende deste erro, os pesos destas unidades em relação às unidades de saída serão sempre os mesmos.

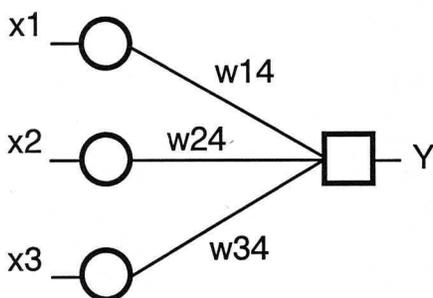
4 REDES NEURAIIS E MODELOS ECONOMÉTRICOS

Ao contrário do que pode parecer à primeira vista, os modelos de redes neurais artificiais têm vários pontos de contato com os modelos econométricos tradicionais. Por modelos econométricos tradicionais referimo-nos tanto ao modelo de regressão, quanto aos modelos mais usuais de

séries de tempo, tais como os modelos ARIMA, o modelo linear dinâmico bayesiano e o modelo estrutural de séries de tempo. Muitas das semelhanças existentes ficam embotadas pelo uso de jargões técnicos diferentes por parte dos economistas e conexionistas.⁴

No caso do modelo de regressão, por exemplo, temos uma variável dita endógena sendo explicada por diferentes variáveis exógenas. Nas RNA, as variáveis exógenas podem ser vistas como os sinais que entram nos neurônios da camada de entrada, enquanto que a variável endógena é representada pelo sinal de saída desejável para a rede. Em outras palavras, a variável endógena é o padrão que é objeto do aprendizado da RNA. Na verdade, uma RNA que tenha apenas uma camada de entrada e outra de saída (perceptron) pode ser facilmente relacionada ao modelo de regressão linear.

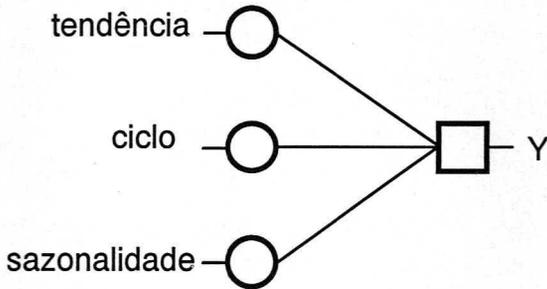
Figura 5



Neste caso, o vetor de pesos da RNA da Figura 5 (w_{14} , w_{24} e w_{34}), que não tem nenhuma camada oculta, nada mais é que o vetor de parâmetros do modelo de regressão. Eles indicam, assim como os parâmetros da regressão, a importância de cada sinal de entrada no que diz respeito a explicação do padrão de saída. Contudo, quando utilizamos uma camada oculta (perceptron multi-camada), como é comum na literatura de RNA, estamos introduzindo não-linearidades na relação entre as entradas x_1 , x_2 e x_3 e a saída y . Portanto, a relação entre as variáveis exógenas e a variável endógena deixa de ser linear, e a comparação, então, tem de ser feita com modelos de regressão não-lineares.

4 Para uma comparação rigorosa das redes neurais e modelos econométricos ver White (1992), principalmente os capítulos 7 e 8.

Figura 6



Mesmo no caso de modelos de séries de tempo, como os ARIMAs ou os modelos de decomposição em componentes não observados, a relação com as RNA segue o mesmo padrão descrito acima. No caso dos modelos de decomposição em componentes não observados, também chamados de modelos estruturais, cada neurônio de entrada pode representar uma característica não observável da série de tempo que se deseja modelar. Temos então, neurônios especializados para representar a sazonalidade, a tendência e o ciclo da série. No caso da sazonalidade, podemos utilizar s neurônios sazonais, onde $s = 4$ ou 12 , dependendo do tipo de dados que se disponha. No caso de dados trimestrais teríamos quatro neurônios sazonais com entrada igual a 1 para o trimestre em questão e zero para os demais. Isto é, teríamos quatro neurônios com sinais de entrada $(1\ 0\ 0\ 0)$, $(0\ 1\ 0\ 0)$, $(0\ 0\ 1\ 0)$ e $(0\ 0\ 0\ 1)$ para cada período de tempo. Alternativamente, poderíamos introduzir a variável dependente defasada em 4 ou 12 períodos conforme o caso, como um neurônio de entrada.

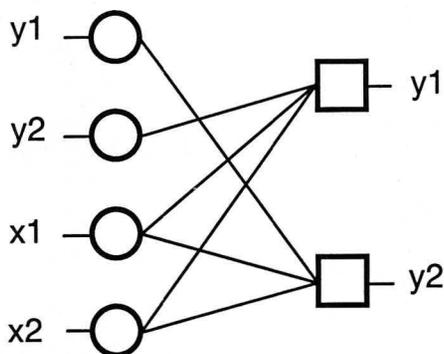
Já a tendência pode ser modelada como uma seqüência de sinais de entrada do tipo $a+bt$, onde $t=1, 2, \dots, T$, para o caso linear. As constantes a e b podem, por exemplo, ser obtidas através de uma regressão da variável dependente contra o tempo, ou estabelecidas em função de conhecimentos *a priori*. No caso de outros formatos de tendência, tais como quadrática ou exponencial, basta utilizar-se uma outra função do tempo para obter os sinais de entrada para o neurônio que representa a tendência. Candidatos óbvios para a geração dos sinais de entrada neste caso são as funções $a+bt+ct^2$ e ae^{bt} . Para a obtenção dos sinais de entrada para o neurônio que represente o componente cíclico podemos, novamente, utilizar funções do tempo. Neste caso, o procedimento mais razoável consiste na utilização de funções periódicas, tais como o seno ou cosseno. Uma possibilidade, neste sentido, é a função $A \cos(\lambda t - \theta)$.

Portanto, na aplicação de RNA ao problema de previsão de séries temporais econômicas, a escolha dos neurônios de entrada pode ser feita utilizando-se a teoria econômica, para determinar as variáveis exógenas relevantes, e os procedimentos comumente utilizados em análise de séries de tempo, tais como funções de auto-correlação e auto-correlação parcial, testes para raízes unitárias, análise gráfica *etc.* Estes procedimentos auxiliam na determinação dos neurônios da camada de entrada, mas não ajudam muito no que diz respeito a determinação da arquitetura global da rede.

A determinação do número de camadas ocultas e o número de neurônios em cada camada oculta é, basicamente, uma questão empírica. Novamente podemos fazer uma analogia com o modelo de regressão, onde a teoria econômica é usada para identificar as variáveis exógenas envolvidas, mas a determinação das relações dinâmicas entre estas e a variável endógena é uma questão empírica. Para a obtenção de uma arquitetura adequada para a RNA devemos utilizar algum critério de poda. Assim como ocorre no modelo de regressão, onde aplica-se o procedimento *general to specific* (Hendry, 1995, cap. 7-8). para o estabelecimento da relações dinâmicas, também nas RNA é usual começar-se com uma rede “grande” que sofre então um processo sucessivo de poda até que se atinja a arquitetura ideal. O processo de poda consiste em eliminar aquelas ligações que mostrem pouca contribuição para a capacidade preditiva da RNA. Isso pode ser feito, por exemplo, utilizando-se algum critério de informação, tais como o AIC, BIC ou SC.

Podemos ainda incluir, como é o caso da Figura 7, as próprias variáveis endógenas como sinais de entrada para a RNA. Temos apenas de tomar o cuidado de não estabelecer ligações entre variáveis iguais. Caso contrário, todas as demais ligações acabariam por ser eliminadas pois existiria uma correlação perfeita entre um dos sinais de entrada e o sinal de saída.

Figura 7



Nos modelos econométricos tradicionais, os parâmetros do modelo são obtidos através de algum processo de estimação que envolve a minimização da soma do quadrado dos resíduos ou a maximização da função de verossimilhança. Já os pesos das RNA são obtidos, segundo o jargão conexcionista, por um processo de aprendizado. Contudo, a estimação dos pesos das RNA, a partir de um processo de aprendizagem, e a estimação dos coeficientes de modelos tradicionais, tais como os modelos ARIMA, são, do ponto de vista matemático e estatístico, exatamente a mesma coisa. Em ambos os casos, o que se procura atingir é minimização da função de erro médio quadrado, ou qualquer outra função objetivo escolhida. *Backpropagation* é apenas um algoritmo numérico utilizado para obter-se o mínimo local da função objetivo em questão. Existem muitos outros algoritmos que realizam o cálculo numérico para a obtenção de mínimos locais ou globais e que podem ser aplicados. Neste sentido, o processo de aprendizagem das RNAs é equivalente à estimação dos parâmetros realizada nos modelos econométricos. Na verdade, o método de retro-propagação é um caso especial do procedimento de “aproximação estocástica” proposto por Robbins, Monro (1951).⁵

A principal diferença entre os modelos econométricos tradicionais e as RNA, tal como utilizadas atualmente, é que estas não contam com uma base estatística pré-determinada. Enquanto que nos modelos econométricos estamos lidando com variáveis aleatórias que possuem uma determinada distribuição conjunta de probabilidade, nas RNA temos apenas sinais de entrada e saída da rede. A ausência de um modelo estatístico bem especificado impede, por exemplo, a construção de intervalos de confiança para as previsões geradas pelas RNA. A previsão gerada pelas RNA é sempre pontual, ao contrário do que ocorre com os modelos econométricos.

5 CONSIDERAÇÕES FINAIS

Vários experimentos de previsão, realizados nas mais diferentes áreas do conhecimento, têm mostrado a eficácia das RNA em gerar previsões de boa qualidade. No caso específico de séries de tempo econômicas, Zandonade (1993), Portugal (1995) e Fernandes, Portugal, Navaux (1994) entre outros, apresentam estudos comparativos das RNA com modelos econométricos tradicionais e concluem que estas representam um instrumento indispensável para aqueles economistas que trabalham com previsão. Especialmente para previsões com mais de um passo à frente, vários estudos têm mostrado que as RNAs são em geral superiores aos modelos econométricos convencionais.

⁵ Para maiores detalhes sobre a comparação entre os métodos de aproximação estocástica e retro-propagação, ver White (1992, p. 87-88).

REFERÊNCIAS

- BEALE, R., JACKSON, T. *Neural computing: an introduction*. Adam Hilger Publishers. 1990.
- CARBONELL, J. G. Introduction: paradigms for machine learning. *Artificial Intelligence*, v. 40, p. 1-9, 1989.
- CARVALHO, L. A. V. Redes neuronais e a tradição conexionista da inteligência artificial. *Relatório Interno*. Programa de Eng. Mecânica, COPPE/UFRJ, 1988.
- COTTRELL, G. W. A connectionist approach to word sense disambiguation. University of Rochester, 1985. (Tese de Doutorado).
- FERNANDES, L. G. L., PORTUGAL, M. S. NAVAUUX, P. O. A. Um estudo de poder preditivo das redes neurais artificiais comparado a métodos econométricos tradicionais. *Anais do I Simpósio Brasileiro de Redes Neurais*. Caxambu, p. 139-144, 1994.
- GALLANT, S. I. connectionist expert systems. *Communications of the ACM*, v. 31, n. 2, p. 152-168, fev. 1988.
- GROSSBERG, S. *the adaptative brain I: cognition, learning, reinforcement, and rhythm and the adaptative brain II: Vision, Speech, Language and Motor Control*, Elsevier North Holland, Amsterdam, 1986.
- HEBB, D. O. *The organization of behavior: a neuropsychological theory*. Science Editions Inc., 1949.
- HECHT-NIELSEN, R. Applications of counterpropagation networks. *Neural Networks*, v. 1, p. 131-140, 1988.
- HENDRY, D. F. *Dynamic econometrics*. Oxford: Oxford University Press, 1995.
- HINTON, G. E. Connectionist learning procedures. *Artificial Intelligence*, v. 40, p. 185-234, 1989.
- HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Science, USA*, v. 79, p. 2554-2558, 1982.
- KOHONEN, T. Clustering taxonomy and topological maps of patterns. *Proc. of the Sixth Intern. Conference on Pattern Recognition*, p. 114-128, 1982.
- LIPPMAN, R. P. An introduction to computing with neural nets. *IEEE ASSP Magazine*, p. 4-22, 1987.
- PESSOA. Aprendizado não-supervisionado em redes neuronais. COPPE Sistemas, UFRJ, 1990. (Tese de Mestrado).

- PORTUGAL, M. S. Neural networks versus time series methods: a forecasting exercise. *Revista Brasileira de Economia*, v. 49, n. 4, p. 611-629, 1995.
- ROBBINS, H., E MONRO, S. A Stochastic approximation method. *Annals of Mathematical Statistics*, v. 25, p. 737-744, 1951.
- RUMELHART, D. E. , MCCLELLAND, J. *Parallel Distributed Processing*, v. 1, MIT Press, 1986.
- RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. Learning internal representatious by error propagation. *Parallel Distributed Processing*, v. 1, MIT Press, p. 318-362, 1986.
- RUMELHART, D. E., WEIGEND, S. A. Predicting the future: A connectionist approach. *Stanford PDP-90-01*, PARC-SSL-90-20, 1990.
- SIMPSON, P. *Artificial neural systems: foundations, paradigms, applications and implementations*, pergamon press inc., 1990.
- TURING, A. M. On computable numbers, with an application to the entscheidungs problem. *Proceedings of the London Mathematical Society 2*, v. 42, p. 230-265, 1936.
- WHITE, H. *Artificial neural networks: approximation and learning theory*, blackwell publishers, Oxford: 1992.
- ZANDONADE, E. *Aplicação da metodologia de redes neurais em previsão de séries temporais*. Dep. de Engenharia Elétrica, PUC-RJ, 1993. (Tese de Mestrado).