

"A FEA e a USP respeitam os direitos autorais deste trabalho. Nós acreditamos que a melhor proteção contra o uso ilegítimo deste texto é a publicação online. Além de preservar o conteúdo motiva-nos oferecer à sociedade o conhecimento produzido no âmbito da universidade pública e dar publicidade ao esforço do pesquisador. Entretanto, caso não seja do interesse do autor manter o documento online, pedimos compreensão em relação à iniciativa e o contato pelo e-mail bibfea@usp.br para que possamos tomar as providências cabíveis (remoção da tese ou dissertação da BDTD)."

UNIVERSIDADE DE SÃO PAULO
FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
MESTRADO PROFISSIONALIZANTE
“MODELAGEM MATEMÁTICA EM FINANÇAS”

**REDES NEURAIS APLICADAS AO RECONHECIMENTO E
CLASSIFICAÇÃO DE PADRÕES EM SÉRIES FINANCEIRAS**

SANDRA MARIA CAPUANO DE OLIVEIRA

Orientador: Henrique von Dreifus

São Paulo
2002

REDES NEURAIS APLICADAS AO RECONHECIMENTO E CLASSIFICAÇÃO DE PADRÕES EM SÉRIES FINANCEIRAS

Sandra Maria Capuano de Oliveira

**Dissertação apresentada à
Faculdade de Economia,
Administração e Contabilidade e
ao Instituto de Matemática e
Estatística da Universidade de São
Paulo para obtenção do Título de
Mestre.**

Orientador: Henrique von Dreifus

**São Paulo
2002**

Aos meus pais, a quem tudo devo.

AGRADECIMENTOS

A Luis Gustavo Gasparini Kiatake, por toda a dedicação e companheirismo sempre demonstrados. A ele devo todo o meu amor.

Aos meus pais, César e Luiza, que estiveram sempre presentes, pelo carinho e apoio.

À minha irmã Adriana, pelas lições de vida e de força, que muito tem me ensinado.

Ao Prof. Henrique von Dreifus, pela excelente orientação.

Ao Prof. Renato Vicente, pela imensa contribuição dada a esse trabalho.

Ao Prof. Amaury Fonseca Junior, pelo apoio no início do estudo.

A Roberta Anchieta da Silva, pela amizade preciosa e pelas discussões estatísticas.

A Mauro Morelli, pela compreensão durante esse período e pelos ensinamentos sobre o mercado financeiro no decorrer desses anos.

Ao Banco Itáu pelo suporte financeiro.

“ A pessimist sees the difficulty in every opportunity; an optimist sees the opportunity in every difficulty. ”

Winston Churchill

ABSTRACT

This work presents the performance valuation of a connexionist model for the pattern recognition of financial series. We adopted the approach of classification problem for a back-propagation feed-forward multi-layer network optimized by the scaled conjugated gradient method, applied to the Ibovespa, Telemar PN and Petrobrás PN daily returns series, grouped in weekly periods, as input. Several topologies were trained with different number of hidden units and iterations, which was the stop criterion. The output was the buy or sell predicted order of the specific asset for next week. We evaluated the trained architectures based on the classification and profitability values. The results provided the validation of the proposed model.

OBJETIVO

Neste trabalho avaliamos a habilidade de um modelo conexionista no reconhecimento de possíveis padrões apresentados em séries financeiras. Através da abordagem de um problema de divisão em classes, utilizamos uma rede multicamada do tipo feed-forward tendo como algoritmo de aprendizado back-propagation otimizado pelo método gradiente conjugado escalonado. Tomando como entrada os retornos diários das séries de Ibovespa, Telemar PN e Petrobrás PN em períodos semanais, diversas topologias foram treinadas, variando pelo número de unidades ocultas e o número de iterações no critério de parada, gerando como saída ordens de compra e venda dadas pela previsão do movimento do preço do ativo para a semana seguinte. As arquiteturas treinadas foram avaliadas com base na taxa de classificação e lucratividade, e os resultados obtidos confirmaram a validação do modelo proposto.

ÍNDICE

1	<i>Introdução</i>	1
2	<i>Mercados Eficientes x Previsibilidade</i>	2
3	<i>Redes Neurais</i>	6
3.1	Histórico.....	6
3.2	Redes Neurais Biológicas.....	8
3.3	Redes Neurais Artificiais.....	9
3.3.1	Topologia.....	9
3.3.2	Função de ativação.....	11
3.3.3	Algoritmo de Aprendizado.....	14
3.3.3.1	Cálculo das Derivadas.....	15
3.3.3.2	Função Erro.....	17
3.3.3.3	Métodos de Otimização.....	18
3.3.3.4	Critério de parada.....	21
4	<i>Metodologia</i>	22
4.1	A escolha do ativo objeto.....	22
4.1.1	Índice da Bolsa de Valores de São Paulo.....	22
4.1.2	Telemar PN.....	23
4.1.3	Petrobrás PN.....	24
4.2	A escolha do período.....	24
4.3	Tratamento dos dados de entrada.....	27
4.4	A arquitetura de rede utilizada.....	27
5	<i>Resultados</i>	29
5.1	Taxa de classificação.....	29
5.2	Matriz de Acertos e Erros (Confusion Matrix).....	35
5.3	Lucratividade.....	37
5.4	Consistência.....	41
6	<i>Conclusão</i>	43
	<i>Referência Bibliográfica</i>	45

ÍNDICE DE FIGURAS

<i>Figura 3.1 – Estrutura de uma rede tipo perceptron</i>	7
<i>Figura 3.2 – Representação do neurônio biológico</i>	8
<i>Figura 3.3 – Exemplo de rede multicamada feed-forward</i>	10
<i>Figura 3.4 – Função de ativação linear</i>	12
<i>Figura 3.5 – Função de ativação threshold</i>	13
<i>Figura 3.6 – Função de ativação logística sigmóide</i>	13
<i>Figura 3.7 – Função de ativação tangente hiperbólica</i>	14
<i>Figura 4.1 – Ibovespa em pontos de julho de 1994 a dezembro de 2001</i>	25
<i>Figura 4.2 – Preço da Petrobrás PN (PETR4) de julho de 1994 a dezembro de 2001</i>	26
<i>Figura 4.3 – Preço de Telemar PN (TNLP4) de setembro de 1998 a dezembro de 2001</i>	26
<i>Figura 4.4 – Esquema representativo para os dados de entrada da rede</i>	27
<i>Figura 4.5 – Topologia da rede feed-forward utilizada no treinamento da rede. O número de unidades na camada escondida foi variado de 2 a 17</i>	28
<i>Figura 5.1 – Taxas de classificação obtidas para Telemar PN no período de treinamento – vista 3D</i>	30
<i>Figura 5.2 - Taxas de classificação obtidas para Telemar PN no período de treinamento – vista 2D</i>	30
<i>Figura 5.3 – Taxas de classificação obtidas para Telemar PN no período de teste – vista 3D</i>	31
<i>Figura 5.4 - Taxas de classificação obtidas para Telemar PN no período de teste – vista 2D</i>	31
<i>Figura 5.5 - Taxas de classificação obtidas para Petrobrás PN no período de treinamento – vista 3D</i>	32
<i>Figura 5.6 - Taxas de classificação obtidas para Petrobrás PN no período de treinamento – vista 2D</i>	32
<i>Figura 5.7 - Taxas de classificação obtidas para Petrobrás PN no período de teste – vista 3D</i>	33
<i>Figura 5.8 - Taxas de classificação obtidas para Petrobrás PN no período de teste – vista 2D</i>	33
<i>Figura 5.9 - Taxas de classificação obtidas para Ibovespa no período de treinamento – vista 3D</i>	34
<i>Figura 5.10 - Taxas de classificação obtidas para Ibovespa no período de treinamento – vista 2D</i>	34
<i>Figura 5.11 - Taxas de classificação obtidas para Ibovespa no período de teste – vista 3D</i>	35
<i>Figura 5.12 - Taxas de classificação obtidas para Ibovespa no período de teste – vista 2D</i>	35
<i>Figura 5.13 – Matriz de acertos/erros das respostas da rede com taxa de classificação máxima para Telemar PN</i>	36
<i>Figura 5.14 - Matriz de acertos/erros das respostas da rede com taxa de classificação máxima para Petrobrás PN</i>	36
<i>Figura 5.15 - Matriz de acertos/erros das respostas da rede com taxa de classificação máxima para Ibovespa</i>	37
<i>Figura 5.16 – Retornos acumulados no período de teste obtidos através da rede com taxa de classificação máxima para Telemar PN</i>	39
<i>Figura 5.17 - Retornos acumulados no período de teste obtidos através da rede com taxa de classificação máxima para Petrobrás PN</i>	40
<i>Figura 5.18 - Representação dos retornos acumulados no período de teste obtidos através da rede com taxa de classificação máxima para Ibovespa</i>	41

1 Introdução

De uma maneira geral, o homem tem buscado maximizar seus ganhos estruturando estratégias que o levem a compreender a dinâmica da formação dos preços. Os primeiros instrumentos de análise técnica datam do século XVII, desenvolvidos pelos orientais para avaliar o preço dos contratos de arroz.

Através dos anos, inúmeros estudos têm sido desenvolvidos motivados por essa busca. Apesar dos esforços empregados, o movimento dos mercados continua sendo um enigma. Diante de tal dificuldade, a idéia de que o mercado seria insuperável ganhou força, constituindo por décadas a base de toda teoria financeira.

Com os avanços tecnológicos, novas técnicas computacionais foram desenvolvidas, e modelos mais estruturados começaram a apontar evidências de que o mercado poderia ser previsível. Surge, nas palavras de Haugen, “A Nova Teoria Financeira”.¹

Dentre essas técnicas, as redes neurais artificiais destacam-se pela capacidade de abstração, auto-aprendizado e adaptação, sendo assim adequadas para o estudo do mercado financeiro.

A maioria dos trabalhos desenvolvidos recentemente empregam redes neurais no estudo de fatores fundamentais ao preço dos ativos financeiros (abordagem fundamentalista). Neste trabalho desenvolvemos um modelo de aplicação da estrutura neural em problemas de classificação, procurando aprender e reconhecer padrões de comportamento no movimento dos preços dos ativos.

Através de uma topologia feed-forward, utilizando o algoritmo de aprendizado back-propagation otimizado pelo método gradiente conjugado escalonado, efetuamos o treinamento de diferentes configurações de rede, utilizando como dados de entrada retornos diários de períodos semanais, gerando como resposta ordens de compra ou venda baseadas na previsão de tendência para a semana seguinte.

Utilizando como critério de avaliação a taxa de acertos das respostas apresentadas pela rede e sua lucratividade, verificamos que o modelo proposto apresentou um desempenho acima do mercado, validando a metodologia desenvolvida.

A dissertação está organizada da seguinte forma: no Capítulo 2 discutimos os conceitos de eficiência de mercado, previsibilidade no mercado financeiro, e descrevemos alguns trabalhos desenvolvidos recentemente na área. No Capítulo 3, contextualizamos o conceito de redes neurais, descrevendo todas as estruturas envolvidas. No Capítulo 4, descrevemos a metodologia utilizada para a construção do modelo. No Capítulo 5, demonstramos os resultados obtidos e no Capítulo 6 apresentamos as conclusões e contribuições do trabalho e sugerimos futuros desenvolvimentos.

2 Mercados Eficientes x Previsibilidade

O conceito de eficiência de mercado constitui uma das principais bases da Teoria de Finanças. Um mercado é dito eficiente quando os preços dos ativos refletem todas as informações disponíveis, ajustando-se imediatamente a uma nova informação. Assim, tentativas de previsões seriam inúteis, pois o mercado reagiria a qualquer ação desencadeada por uma nova informação, compensando-a. Nenhum investidor poderia alcançar retornos acima do mercado baseado em dados históricos ou atuais. Variações sucessivas nos preços seriam aleatórias, independentes e identicamente distribuídas, de natureza estocástica, fenômeno denominado *random walk*.

Três tipos de eficiência foram sugeridos por Harry Roberts², de acordo com o nível de disponibilidade de tais informações:

1. *Forma fraca*: o conjunto de informações disponíveis ao investidor se resume ao histórico de preços.
2. *Forma semi-forte*: além dos preços históricos, o investidor tem acesso a informações de disponibilidade pública, como relatórios anuais de empresas e publicações em jornais e revistas.
3. *Forma forte*: em adição aos preços históricos e às informações públicas, o investidor tem acesso à qualquer tipo de informação, incluindo informações privilegiadas e confidenciais.

Vários estudos foram realizados com o objetivo de testar a eficiência dos mercados, como os modelos desenvolvidos por Sharpe, Lintner e Black, Ross, Merton, Rubstein, Lucas, Breeden e Cox, Ingersoll e Ross, sendo o mais difundido e empregado o Capital Asset Pricing Model, ou CAPM, de Sharpe, Lintner e Black. A maioria deles chegou a resultados favoráveis à teoria de mercados eficientes na sua forma fraca e semi-forte, mas não na forma forte.³

Eugene Fama, em 1970, reuniu os principais trabalhos que avaliavam a eficiência dos mercados, analisando teoria e testes empíricos. A conclusão foi que evidências contra mercados eficientes existiam, mas apesar de estatisticamente significantes, apresentavam-se sem grande intensidade e não possuíam um significado econômico bem definido, podendo portanto ser descartadas.⁴

Com o desenvolvimento computacional, técnicas mais avançadas de não-linearidade foram desenvolvidas, criando ferramentas explicativas que teoricamente melhor se aproximariam dos movimentos do mercado, que é claramente não-linear. Dentre as principais técnicas empregadas estão a teoria do caos, modelos de inteligência artificial e os sistemas especialistas baseados em mecanismos de simulação (bootstrap).

Recentemente, inúmeros trabalhos têm apresentado evidências contrárias à hipótese de eficiência dos mercados. Tais evidências são denominadas *anomalias de mercado*, que podem ser classificadas em anomalias de calendário, fundamentais, técnicas entre outras.³

O estudo dessas anomalias pode ser dividido em duas linhas de pesquisa: a *Análise Técnica* e a *Análise Fundamentalista*.

A análise técnica é definida como o estudo dos movimentos do mercado, com o objetivo de prever tendências futuras nos preços dos ativos⁵. Os movimentos do mercado incluem variações no preço, volume e número de contratos abertos. Munidos de tais informações, e principalmente através de gráficos, os analistas técnicos, ou “chartistas”, procuram reconhecer padrões nos dados passados que levem a uma predição de qual será o próximo passo do mercado. Tal abordagem é baseada em três premissas:

1. *O movimento do mercado contém tudo.* Os analistas técnicos acreditam que qualquer informação que poderia afetar o preço de um ativo, seja ela fundamentalista, econômica, política ou psicológica, já está implícita no preço atual do ativo. Assim, basta o estudo das variações no preço para o diagnóstico do humor do mercado, pois todas as outras variáveis, como oferta e demanda, e conseqüentemente, otimismo e pessimismo, estariam refletidas nesse preço.
2. *O preço segue linhas de tendência.* O objetivo da análise técnica é identificar uma tendência de mercado no seu estágio inicial, para então apostar na direção dessa tendência. Essa aposta continua até que surjam indícios de uma reversão. Um corolário dessa premissa seria, parafraseando a Primeira Lei de Newton, “Uma tendência em movimento tem maior probabilidade de continuar do que de reverter”.
3. *A história se repete.* O preço de um ativo representa um consenso baseado nas expectativas dos agentes financeiros, sendo que os padrões apresentados nos movimentos passados do mercado revelam as atitudes desses agentes em situações de otimismo ou pessimismo⁶. Assim, a análise técnica baseia-se também, de certa forma, em um estudo da psicologia humana, acreditando que tais padrões de comportamento tendem a manter-se ao longo do tempo, repetindo-se. Em outras palavras, a chave para a compreensão do futuro estaria no estudo do passado.

A análise fundamentalista, por sua vez, tem como foco os parâmetros econômicos de oferta e demanda que causam o movimento do mercado, e não o movimento em si. Os fundamentalistas examinam os fatores relevantes que afetam o preço de um ativo, como dados econômicos, industriais e as condições da empresa em questão, com o objetivo de calcular seu valor intrínseco. Caso o valor calculado, denominado valor justo, seja inferior ao valor negociado no mercado, os analistas concluem que o mercado está superestimando o valor do ativo e apostam em uma queda no seu preço. Caso contrário, eles apostam em uma alta.

Enquanto a análise fundamentalista estuda as causas dos movimentos do mercado, a análise técnica estuda os efeitos. Na prática, as duas técnicas geralmente acabam sendo utilizadas simultaneamente, sendo que a análise fundamentalista é

adequada para a seleção de quais ativos devem ser negociados, e a análise técnica para determinar a ocasião em que devem ser negociados.⁷

Nesse estudo empregamos a ideologia da análise técnica, com o objetivo de reconhecer padrões de comportamento no movimento dos preços históricos dos ativos. Como nos baseamos em retornos semanais para a resposta da rede neural, estruturamos os dados de entrada de forma a captar, entre outras anomalias, um possível efeito dia da semana, que consiste na observação de padrões de comportamento diferenciados do mercado de acordo com o dia da semana. A maioria dos estudos comprova que as segundas-feiras tendem a apresentar retornos predominantemente negativos. Uma explicação dada a este fato é que as empresas esperam o fechamento do mercado na sexta-feira para a divulgação de notícias ruins.⁸

Alguns trabalhos podem ser destacados na pesquisa da eficiência dos mercados e previsibilidade dos preços:

Em 1988, Lo e Mackinlay rejeitaram fortemente a hipótese de random walk, analisando retornos semanais, tanto para ações individuais quanto para portfólios de ações e índice de ações, sugerindo inclusive uma má especificação no modelo de Black-Scholes, amplamente empregado na precificação de opções.⁹

Em 1990, Jegadeesh encontrou alta correlação serial negativa nos retornos mensais de ações, e correlação serial positiva nos retornos anuais, utilizando um modelo de regressão por cross-sectional, resultando em retornos anormais no período considerado.¹⁰

Em 1991, Fama voltou a avaliar a teoria de mercados eficientes, analisando os estudos de eventos, informação privilegiada e a previsibilidade dos retornos dos ativos. Novamente voltou a concluir que o mercado ajusta-se de forma eficiente a informações públicas. Quanto às informações privilegiadas, sugeriu que as anomalias encontradas poderiam ser uma evidência da ineficiência do mercado, mas mais provavelmente seriam resultantes de problemas na estruturação ou implementação do modelo. Contudo, constatou evidências de previsibilidade dos retornos dos ativos, principalmente em períodos mais longos.¹¹

Em 1992, Brock, Lakonishok e LeBaron encontraram evidências de previsibilidade nos retornos diários do Índice Dow Jones utilizando ferramentas de análise técnica e *bootstrap*, no período de 1897 a 1986.¹²

Em 1998, em um outro estudo, Fama admite a existência de anomalias nos retornos dos ativos de longo prazo, encontrando correlação serial negativa significativa, mas apesar disso não abandonou a teoria de mercados eficientes, alegando que tais anomalias poderiam ser casuais ou fruto de modelos inadequados.¹³

Em 1999, Andersen, Gluzman e Sornette realizaram testes nos principais índices, títulos e moedas de vários países, utilizando um modelo baseado na velocidade e na aceleração dos preços, e encontraram evidências fortemente significantes de previsibilidade dos retornos.¹⁴

Em 2000, Lewellen e Shanken testaram um modelo de predição baseado no risco de estimação, encontrando retornos divergentes com o modelo do CAPM.¹⁵

No mesmo ano, Lo, Mamayasky e Wang testaram indicadores técnicos nos retornos diários de diversas ações, no período de 1962 a 1996, utilizando regressão por kernel não-paramétrica, concluindo que instrumentos de análise técnica contribuíram de forma positiva para o desempenho do modelo aplicado.¹⁶

Um levantamento mais abrangente dos estudos realizados sobre eficiência e previsibilidade do mercado financeiro pode ser encontrado em Bruni e Famá³.

3 Redes Neurais

O estudo de redes neurais possui uma história de aproximadamente seis décadas. Contudo, aplicações mais consistentes têm sido apresentadas somente nos últimos vinte anos, fruto do maior desenvolvimento dos sistemas computacionais, e continuam se desenvolvendo rapidamente.

Atualmente, redes neurais artificiais têm sido empregadas em diversos campos, como reconhecimento e classificação de padrões, clustering, previsão de séries temporais, aproximação de funções, predição e otimização, em diversos ramos de ciência e tecnologia. Esse amplo crescimento se deve principalmente às características implícitas aos modelos conexionistas, como a habilidade em tratar sistemas não-lineares, capacidade de auto-ajuste (ou auto-aprendizado), abstração dos dados de entrada e generalização dos dados de saída.

Neste capítulo introduzimos um breve histórico do estudo de redes neurais artificiais, construímos um paralelo entre as redes neurais biológicas, e descrevemos em detalhes a sua estruturação.

3.1 Histórico

O primeiro modelo de redes neurais artificiais foi idealizado pelo neurofisiologista McCulloch e o matemático Walter Pitts, em 1943¹⁷. Inspirados nas ligações características de uma rede neural biológica, o modelo simulava o comportamento de um neurônio real, possuindo apenas uma saída, que era uma função soma dos valores de entrada. Esse modelo serviu como base para a maioria dos modelos conexionistas existentes.

Em 1949, o psicólogo Donald Hebb¹⁸ publicou “The Organization of Behaviour”, propondo uma lei de aprendizagem específica para as sinapses dos neurônios. O estudo demonstrou que a capacidade da aprendizagem em redes neurais vem da alteração da eficiência sináptica, isto é, a conexão somente é ativada se tanto as células pré-sinápticas quanto as pós-sinápticas estiverem excitadas. Com isso, Hebb traduziu matematicamente a sinapse dos neurônios biológicos.

Dois anos mais tarde, Marvin Minsky construiu o primeiro neuro computador, denominado Snark. O Snark operava com sucesso a partir de um ponto de partida técnico, ajustando seus pesos automaticamente, entretanto ele nunca executou qualquer função de processamento de informação interessante.

Em 1956, no Dartmouth College, nasceram as duas vertentes da inteligência artificial: a *simbólica*, que tenta simular o comportamento inteligente humano desconsiderando os mecanismos responsáveis por tal, e a *conexionista*, que acredita que construindo-se um sistema que simule a estrutura do cérebro, este sistema apresentará inteligência, sendo capaz de aprender, errar e aprender com seus erros.

Em 1962, pesquisando problemas de classificação tendo como entrada imagens binárias comuns e formas simples, Rosenblatt criou o *perceptron*, constituído

inicialmente por uma única camada de neurônios, com função de ativação threshold, como mostrado na Figura 3.1.

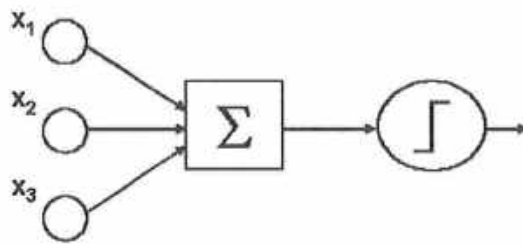


Figura 3.1 – Estrutura de uma rede tipo perceptron, com uma única camada e função de ativação threshold

Ao mesmo tempo que Rosenblatt estava desenvolvendo o perceptron, Widrow e sua equipe trabalhavam em linhas similares utilizando um sistema denominado adalines (ADAPtive LINEar Element), referindo-se a uma única unidade de processamento não-linear. A importância desse modelo está associada à regra de aprendizagem proposta, a regra Delta.

Em 1969, Minsky e Papert analisaram as restrições do perceptron, demonstrando que redes de uma única camada não são capazes de solucionar problemas que não sejam linearmente separáveis, o que fez com que o interesse pelo estudo de redes neurais artificiais praticamente desaparecesse nos anos seguintes. Também a publicação de diversos artigos e livros marcados pelo entusiasmo exagerado de muitos pesquisadores, resultando em modelos de previsão pouco confiáveis, tirou quase toda a credibilidade dos estudos na área.

A partir dos anos 80, com o avanço da tecnologia provendo suporte computacional e o fracasso da escola simbolista na solução de determinados tipos de problema, as redes neurais passaram a atrair substancial atenção novamente.

Em 1982, a introdução do modelo de redes simétricas para otimização, através de um algoritmo de aprendizagem que estabilizava uma rede binária simétrica com realimentação, conhecido pelo nome de seu idealizador, o físico e biólogo John Hopfield, permitiu esclarecer, pelas suas características computacionais e estabilidade, boa parte das dúvidas até então existentes em relação ao processo dinâmico executado por certas redes neurais.

Hinton e Seynowsky, em 1983, estenderam o modelo de Hopfield com a incorporação de dinâmica estocástica, conhecido como a Máquina de Boltzmann.

Dois anos mais tarde, Rumelhart, Hinton e Williams aperfeiçoaram a ideia do perceptron, criando o algoritmo back-propagation, levando a uma explosão de interesse em redes neurais. Idéias similares haviam sido desenvolvidas anteriormente por alguns pesquisadores, incluindo Werbos em 1974, e Parker em 1985. O sucesso desse algoritmo estimulou o desenvolvimento de inúmeras pesquisas em redes neurais artificiais e de uma variedade de modelos cognitivos.

Atualmente, o uso de redes neurais para aproximar funções complexas tem sido empregado em vários campos de aplicação, incluindo reconhecimento de padrões, problemas de identificação, e em sistemas de fala, visão e controle operacional.

3.2 Redes Neurais Biológicas

Historicamente, muitos dos conceitos utilizados em redes neurais artificiais foram inspirados no estudo de redes neurais biológicas. A curiosidade sobre a possível reprodução artificial da capacidade de aprendizado do ser humano foi um dos maiores motivadores do estudo de métodos de inteligência artificial. Dessa forma, um maior conhecimento das redes neurais reais auxilia a compreensão do mecanismo dos modelos conexionistas.

O sistema nervoso biológico é formado por um conjunto extremamente complexo de células, denominadas neurônios. Estes são os maiores responsáveis pela capacidade de raciocínio e aprendizado. A Figura 3.2 mostra um esquema de neurônio biológico.

Os neurônios são formados pelos dendritos, que são um conjunto de terminais de entrada com função de receber os estímulos, pelo corpo celular, que é responsável por coletar e combinar as informações, e pelos axônios, que são longos terminais de saída responsáveis por transmitir os estímulos para outras células. A região onde dois neurônios entram em contato e através da qual os impulsos nervosos são transmitidos entre eles é conhecida como sinapse.

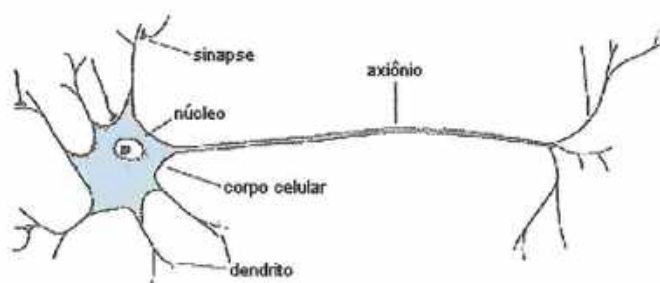


Figura 3.2 – Representação do neurônio biológico¹⁹

A propagação dos impulsos se realiza da seguinte maneira: os pulsos recebidos por um neurônio são continuamente processados até que, atingindo um limiar de ação, são disparados através da produção de uma substância neurotransmissora que flui do corpo celular para o axônio. Este, por sua vez, pode estar conectado a um dendrito de um outro neurônio, que passará pelo mesmo processo, acrescentando novas modificações no pulso e o propagando.

O neurônio que transmite o pulso pode controlar sua intensidade, medida pela frequência com que emite pulsos contínuos e positivos, através da alteração da polaridade na membrana pós-sináptica. Este processo depende de vários fatores, como a geometria da sinapse e o tipo de neurotransmissor.

3.3 Redes Neurais Artificiais

De forma similar ao ser humano, as redes neurais artificiais são capazes de aprender comportamentos através de exposição de exemplos dos mesmos. Tomando como analogia a composição de uma rede biológica, podemos estruturar a arquitetura de uma rede neural computacional de maneira associativa.

Seguindo essa abordagem, os dendritos são representados por *unidades de entrada*, cujas ligações com o corpo celular artificial são realizadas através de elementos chamados *pesos*, simulando as sinapses. Os estímulos são processados por uma *função soma*, e o limiar de disparo por uma *função de transferência*, que então transmite o pulsos por *unidades de saídas*, que representam os axônios.

O processo para o desenvolvimento de aplicações em redes neurais artificiais pode ser dividido nos seguintes passos: inicialmente procedemos a coleta dos dados de entrada, separando-os em dois grupos, um para o período de treinamento e outro para a fase de teste, que devem ser disjuntos. Então estruturamos a arquitetura da rede e efetuamos o treinamento em si, de acordo com o modelo proposto, gerando uma matriz de pesos W . A partir desses pesos, um novo conjunto de dados é propagado pela rede em uma fase de teste, avaliando-se a performance da rede treinada perante padrões inéditos.

Basicamente, uma rede neural artificial é caracterizada pela sua topologia, que especifica a disposição das diversas camadas de neurônios, pelas características de seus neurônios, e pelas regras de treinamento.

3.3.1 Topologia

Redes com apenas uma camada, no modelo do perceptron, são baseadas em uma combinação linear das variáveis de entrada, que são transformadas por uma função de ativação não linear, resultando em redes muito limitadas em relação ao tipo de funções que conseguem representar. Para permitir um mapeamento mais genérico devemos considerar sucessivas transformações correspondentes às redes contendo várias camadas de pesos ajustáveis. De fato, prova-se que duas camadas de pesos são capazes de aproximar qualquer mapeamento de função contínua.²⁰

De um modo mais geral, podemos considerar diagramas de rede arbitrários, que não possuam necessariamente uma estrutura de camadas simples, já que qualquer diagrama de rede pode ser convertido em sua função de mapeamento correspondente. A única restrição é que o diagrama tem que ser alimentado de modo progressivo, ou *feed-forward*, para que não contenha passos em ciclos. Isto assegura que a saída da rede pode ser calculada como uma função explícita dos parâmetros de entrada e dos

pesos. Em geral dizemos que uma rede é feed-forward se, ao numerarmos todas as unidades, das entradas até as saídas, cada unidade somente receba conexões provenientes de unidades que possuam um número de atribuição menor do que o seu.

Devemos pensar em uma rede neural feed-forward como sendo uma ferramenta geral para representar o mapeamento de funções não lineares entre um conjunto de variáveis de entrada e um conjunto de variáveis de saída, representando a função não linear de várias variáveis em termos de composições de funções não lineares de uma variável, chamadas de função de ativação ou transferência.

Cada função multivariada pode ser representada em termos de um diagrama de rede neural tal como a correspondência um-a-um entre os componentes da função e os elementos do diagrama. Da mesma forma, qualquer topologia do diagrama de rede, desde que feed-forward, pode ser transladado em uma função de mapeamento correspondente.

Tais redes são teoricamente mais fáceis de serem analisadas do que topologias mais genéricas, e podem freqüentemente ser implementadas de uma maneira mais eficiente em uma simulação computacional. Um exemplo de rede multicamada feed-forward é mostrado na Figura 3.3. O diagrama representa uma rede de duas camadas, com d unidades de entrada x , M unidades na camada oculta z , e c unidades de saída y . Os pesos w representam as conexões entre as unidades.

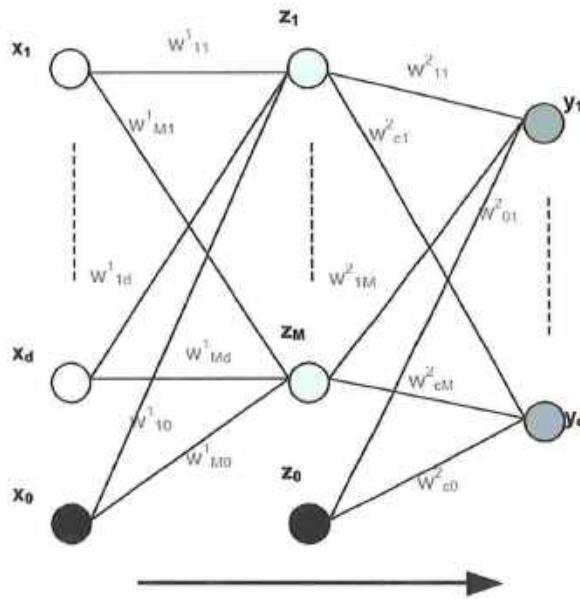


Figura 3.3 – Exemplo de rede multicamada feed-forward, onde x_i representa as variáveis de entrada, z_j as unidades ocultas, y_k unidades de saída e w_{ij} os pesos

Podemos escrever a função analítica correspondente a essa rede como sendo: a saída da j -ésima unidade oculta é obtida formando-se uma combinação linear

ponderada dos valores de entrada (*função discriminante*), e adicionando um bias, ou seja

$$a_j = \sum_{i=0}^d w_{ji}^1 x_i \quad (3.1)$$

onde w_{ji}^1 denota o peso da primeira camada que liga o dado de entrada i à unidade oculta j . Um *bias* foi incluído para a camada de entrada através da soma de uma variável extra x_0 , cujo valor é estabelecido como sendo $x_0 = 1$. A ativação da unidade escondida j é então obtida através da transformação da soma linear na equação (3.1), utilizando a função de ativação

$$z_j = g(a_j). \quad (3.2)$$

As saídas da rede são obtidas transformando as ativações das unidades ocultas através de uma segunda camada de elementos de processamento. Assim, para cada saída k , construímos uma combinação linear de saídas das unidades escondidas na forma

$$a_k = \sum_{j=0}^M w_{kj}^2 z_j \quad (3.3)$$

novamente utilizando um bias $z_0 = 1$ na forma de uma unidade escondida extra.

A ativação (excitação) da k -ésima variável de saída é então obtida transformando esta combinação linear, utilizando uma função de ativação não linear, resultando em

$$y_k = \tilde{g}(a_k). \quad (3.4)$$

onde a função de ativação \tilde{g} não precisa necessariamente ser a mesma função de ativação g utilizada no passo anterior.

Se combinarmos as quatro equações, obteremos uma expressão explícita para a função completa representada pelo diagrama de rede na forma

$$y_k = \tilde{g} \left(\sum_{j=0}^M w_{kj}^2 g \left(\sum_{i=0}^d w_{ji}^1 x_i \right) \right). \quad (3.5)$$

3.3.2 Função de ativação

A função de ativação, ou função de transferência, é uma função que age na soma linear dada pela função discriminante, determinando o grau de importância dessa soma ponderada, gerando uma excitação ou inibição do neurônio.

Existem vários tipos de funções de ativação, como por exemplo a função linear, threshold (conhecida também por Hard Limit, Heaviside ou degrau), sigmóide, radial

basis, que determina a ativação de uma unidade oculta através da distância entre o vetor de entrada e um vetor modelo, tangente hiperbólica, softmax, entre outras.

Dentre essas, as mais empregadas são:

Linear:

A função linear é a mais simples função de ativação, sendo da forma

$$g(a) = a,$$

cuja resposta é representada na Figura 3.4.

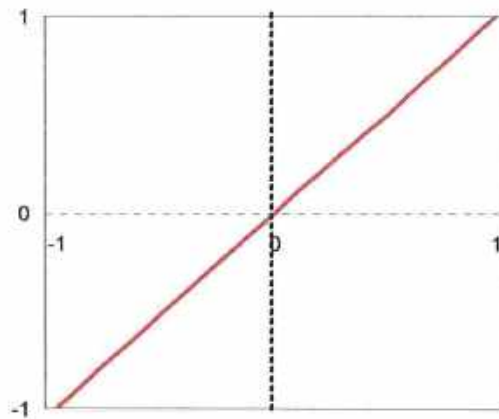


Figura 3.4 – Função de ativação linear

Threshold:

A função threshold, ou Heaviside (Hard-Limit) é dada por

$$g(a) = \begin{cases} 0 & \text{se } a < 0 \\ 1 & \text{se } a \geq 0 \end{cases}$$

e leva o neurônio a tomar uma decisão ou classificação. Pode ser representada como na Figura 3.5.

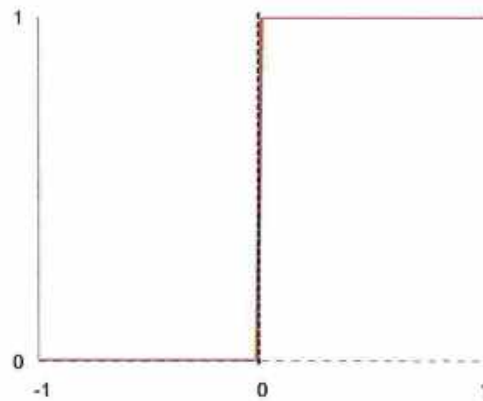


Figura 3.5 – Função de ativação threshold

Logística sigmóide:

A função logística sigmóide gera valores de saída compreendidos no intervalo (0,1), e é representada pela equação

$$g(a) \equiv \frac{1}{1 + \exp(-a)},$$

como apresentado na Figura 3.6.

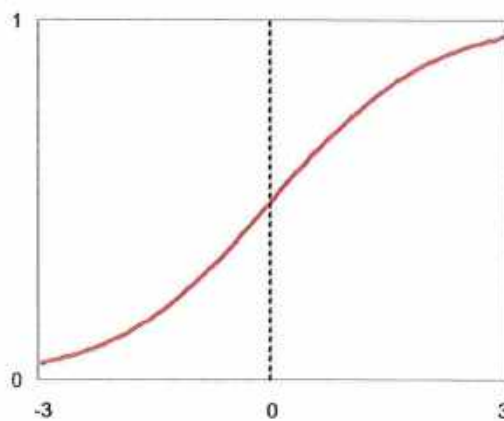


Figura 3.6 – Função de ativação logística sigmóide

A importância desta função na utilização em redes neurais se dá pela interpretação probabilística que ela pode gerar, produzindo respostas que podem ser interpretada como probabilidades a posteriori, sendo indicada para problemas de classificação com duas classes. Em problemas de mais de duas classes, uma extensão da função sigmóide chamada exponencial normalizada ou softmax é utilizada.²⁰

Função tangente hiperbólica:

A função tangente hiperbólica é dada pela forma

$$g(a) \equiv \frac{e^a - e^{-a}}{e^a + e^{-a}},$$

como representado na Figura 3.7.

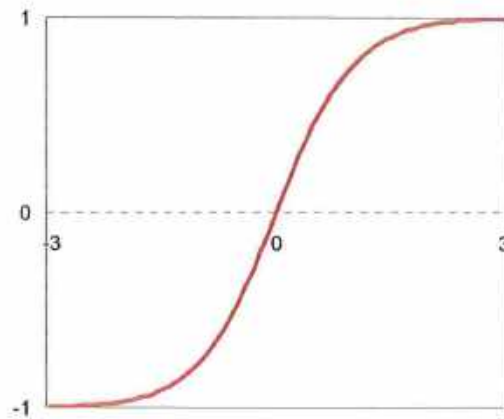


Figura 3.7 – Função de ativação tangente hiperbólica

Podemos notar que a função de ativação tangente hiperbólica é na verdade uma transformação linear da função sigmóide. Assim, as duas redes são equivalentes, porém geram pesos e biases diferentes. Resultados empíricos demonstram que a tangente hiperbólica convergem mais rapidamente no treinamento da rede do que a sigmóide.²⁰

3.3.3 Algoritmo de Aprendizado

Um algoritmo de aprendizado ou treinamento é um conjunto de regras bem definidas que especificam como os pesos iniciais devem ser adaptados para que a rede assimile o comportamento dos dados analisados, baseando-se na definição de uma função erro apropriada, e na sua minimização com respeito aos pesos e bias da rede.

Os valores dos pesos podem ser estabelecidos a priori, ou seja, ajustados em um único passo (redes de Hopfield), ou atualizados interativamente através de um método

de aprendizado. Esse método pode ser *supervisionado*, no qual os pesos são otimizados através de comparações entre os valores de saída gerados pela rede e os valores desejados como resposta (*targets*), ou *não supervisionado*, que ao invés de realizar um mapeamento entre entradas e saídas, procura modelar a distribuição de probabilidade dos dados de entrada, ou agrupar estes dados em classes (clusters). Dessa forma, pares desejados de informação de entrada e saída são obtidos e as relações de causa e efeito podem ser estabelecidas.

Existem vários tipos de algoritmos de aprendizado, diferindo entre si principalmente pelo modo de atualização dos pesos. Certamente, o mais difundido e utilizado é o *Back-propagation Error*.

Segundo Bishop²⁰, o termo back-propagation é utilizado na literatura com diversos sentidos. Como exemplo, uma arquitetura perceptron multicamada simples é referenciada em alguns trabalhos como sendo uma rede back-propagation. Uma rede multicamada que utiliza o método gradiente descendente aplicado à função soma dos quadrados dos erros também pode ser considerada uma rede back-propagation.

Para esclarecer tal terminologia, considere uma rede multicamada do tipo threshold. A camada final de pesos na rede pode ser considerada como um perceptron com valores de entrada fornecidos pelas unidades de saída da camada escondida antecedente. Esta abordagem, contudo, não pode ser aplicada para determinar os pesos das camadas anteriores da rede. Apesar de tais camadas poderem, a princípio, ser consideradas também como perceptrons, não existe um procedimento para associar os valores desejados às suas saídas. Ou seja, se uma unidade de saída produzir uma resposta incorreta, não é possível determinar qual unidade deve ser considerada responsável por gerar tal erro, e, portanto, não conseguimos determinar qual peso deve ser ajustado e nem o quanto.

Agora, se considerarmos uma rede com função de ativação diferenciável, então a ativação das unidades de saída torna-se uma função diferenciável das unidades de entrada, dos pesos e bias. Logo, podemos calcular as derivadas da função erro com respeito aos pesos e estas derivadas podem ser utilizadas para encontrar os valores de pesos que minimizam a função erro, utilizando algum método de otimização. O algoritmo para calcular a derivada da função erro é conhecido como back-propagation já que, como vemos, corresponde a retropropagação dos erros através da rede.

A maioria dos algoritmos de treinamento envolve um processo iterativo para minimizar a função erro, com o ajuste dos pesos sendo feito em uma seqüência de passos, sendo dividido em dois estágios. O primeiro estágio calcula a derivada da função erro com relação aos pesos. O algoritmo back-propagation contribui com um método computacionalmente eficiente para este cálculo. No segundo estágio, as derivadas são utilizadas para computar os ajustes a serem feitos nos pesos.

3.3.3.1 Cálculo das Derivadas

Considere uma rede neural feed-forward qualquer, que utiliza uma função de ativação não linear e diferenciável arbitrária, e uma função de erro diferenciável arbitrária. Como vimos, cada unidade pode ser representada na forma

$$a_j = \sum_i w_{ji} z_i \quad (3.6)$$

onde z_j é a ativação de uma unidade escondida ou de entrada x_j , que envia uma conexão para a unidade j , com peso w_{ji} associado a essa conexão. A soma em (3.6) é então transformada por uma função de ativação não linear g , proporcionando a ativação da unidade j na forma

$$z_j = g(a_j) \quad (3.7)$$

sendo que z_j pode ser uma unidade escondida ou uma saída da rede y_j . O processo descrito acima, calculado para todas as camadas da rede, é conhecido como *forward propagation* (ou propagação de modo progressivo).

Assim, para determinar os pesos apropriados para um melhor desempenho da rede, iremos minimizar uma função erro, que deve ser considerada como uma função soma de todos os padrões no conjunto de treinamento, de um erro definido para cada padrão separadamente

$$E = \sum_n E^n \quad (3.8)$$

onde n é o número de conjuntos de entradas (padrões). Devemos também propor que o erro E_m pode ser expresso como uma função diferenciável das variáveis de saída tal que

$$E^n = E^n(y_1, \dots, y_c).$$

De (3.8) temos que a derivada do erro E pode ser calculada como a soma do erro de cada padrão separadamente. Para cada padrão, tendo já aplicado a *forward propagation* em todas as unidades, podemos aplicar a regra da cadeia para calcular a derivada

$$\frac{\partial E^n}{\partial w_{ji}} = \frac{\partial E^n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}}. \quad (3.9)$$

Como por (3.6) $\frac{\partial a_j}{\partial w_{ji}} = z_i$, e chamando $\delta_j = \frac{\partial E^n}{\partial a_j}$, teremos

$$\frac{\partial E^n}{\partial w_{ji}} = \delta_j z_i. \quad (3.10)$$

Assim, basta calcular o valor de δ_j . Para as unidades de saída, por (3.7) teremos

$$\delta_k = g'(a_k) \frac{\partial E^n}{\partial y_k}. \quad (3.11)$$

Para as unidades escondidas, utilizando a regra da cadeia novamente, teremos

$$\delta_j = \sum_k \frac{\partial E^n}{\partial a_k} \frac{\partial a_k}{\partial a_j}. \quad (3.12)$$

Podemos então escrever a fórmula do back-propagation como sendo

$$\delta_j = g'(a_j) \sum_k w_{kj} \delta_k, \quad (3.13)$$

que nos diz que o valor de δ de uma unidade particular da rede pode ser obtida propagando os δ s das unidades posteriores para trás.

Podemos resumir o método back-propagation para o cálculo das derivadas da função erro E^n com respeito aos pesos no seguintes passos:

1. Aplicar um vetor de entrada x_n na rede e ativar as unidades ocultas e de saída através de (3.6) e (3.7).
2. Calcular δ_k para todas as unidades de saída utilizando (3.11).
3. Retropropagar as δ s utilizando (3.13) para obter δ_j para cada unidade escondida da rede.
4. Utilizar (3.10) para calcular as derivadas desejadas.

A derivada do erro total E pode ser obtida repetindo-se este processo para cada padrão do conjunto de treinamento, e então somando todos os padrões, na forma

$$\frac{\partial E}{\partial w_{ji}} = \sum_n \frac{\partial E^n}{\partial w_{ji}}. \quad (3.14)$$

Apesar de termos assumido que todas as unidades da rede utilizam a mesma função de ativação g , a generalização é facilmente obtida no caso de utilizarem funções de ativação diferentes.

3.3.3.2 Função Erro

A função erro estabelece uma medida de eficiência da rede neural, sendo que o processo de treinamento baseia-se na sua otimização. A escolha da função erro depende do problema a ser tratado, e também da função de ativação escolhida.

Para problemas de regressão, que têm como objetivo básico modelar a distribuição condicional das variáveis de saída às variáveis de entrada, a função de soma dos quadrados dos erros é apropriada. Outras funções podem ser citadas para problemas de regressão, como Minkowski-R ou input-dependent variance.²⁰

Para problemas de classificação, o objetivo seria modelar a probabilidade a posteriori dos membros de uma classe, também condicionados pelas variáveis de entrada. Apesar de podermos utilizar a função erro, outras seriam mais indicadas. Se escolhermos como função de ativação a função sigmóide, por exemplo, uma função muito apropriada seria a função *cross-entropy*, dada pela equação

$$E = -\sum_n \{t^n \ln y^n + (1-t^n) \ln(1-y^n)\}. \quad (3.15)$$

Derivando com respeito a y_n obtemos:

$$\frac{\partial E}{\partial y^n} = \frac{(y^n - t^n)}{y^n(1-y^n)}, \quad (3.16)$$

que para qualquer valor de n tem como mínimo absoluto $y_n = t_n$. Considerando a derivada da função sigmóide, que tem como propriedade $g'(a) = g(a)(1-g(a))$, temos que a derivada da função erro com respeito a a será dada por

$$\delta^n \equiv \frac{\partial E}{\partial a^n} = y^n - t^n, \quad (3.17)$$

onde δ^n é a quantidade de erro que deve ser retropropagada através da rede.

3.3.3.3 Métodos de Otimização

A otimização dos pesos no processo de aprendizado de uma rede neural envolve a escolha de um algoritmo de otimização que minimize a função erro, ajustando os pesos. Tal algoritmo dará uma seqüência de passos através do espaço dos pesos, que devem decidir inicialmente a direção do movimento, e então a distância dada nesta direção.

O algoritmo de treinamento mais simples é o *gradiente descendente*, que a partir de um vetor inicial de pesos w_0 , iterativamente ajusta os pesos movendo a cada passo τ uma distância na direção da maior taxa de decréscimo do erro, ou seja, na direção do gradiente negativo, calculado como w^τ

$$\Delta w^\tau = -\eta \nabla E \Big|_{w^\tau} \quad (3.18)$$

onde E é o valor do erro e η é a taxa de aprendizado, que deve ser suficientemente pequena.

Assim, a direção é dada pelo negativo do gradiente local da função erro, e o tamanho do passo é determinado pelo parâmetro de aprendizado arbitrário η . Um dos limitadores deste método é justamente a necessidade de uma escolha apropriada para η . Caso contrário, o desempenho da rede pode ser amplamente prejudicado. Além disso, o método pode apresentar problemas de convergência.

Alguns métodos foram criados na tentativa de contornar tais problemas. Entre eles podemos citar o *momentum*, que adiciona um parâmetro μ variando no intervalo $[0,1]$, da forma:

$$\Delta w^\tau = -\eta \nabla E \Big|_{w^\tau} + \mu \Delta w^{(\tau-1)} \quad (3.19)$$

Este procedimento, apesar de adicionar inércia e suavizar o movimento através do espaço de pesos, gera a dificuldade de escolha de mais um novo parâmetro por

tentativa e erro, dependendo das particularidades de cada problema. Outras técnicas tentam corrigir tais problemas, como bold driver, a regra delta-delta, a regra delta-bar-delta, quickprop, entre outras.²⁰

Uma maneira eficiente de efetuar tal correção é apresentada no algoritmo conhecido como *gradiente conjugado*, que fornece uma técnica de minimização que exige apenas o cálculo da função erro e de suas derivadas, sendo que, para uma função erro quadrática, garante o encontro do mínimo em no máximo W passos. Este procedimento pode ser resumido como segue:

1. Inicializa-se um vetor de pesos w_j .
2. Calcula-se o vetor gradiente g_j , e estabelece-se uma direção de busca inicial

$$d_j = -g_j.$$

3. No passo j , minimiza-se $E(w_j + \alpha d_j)$ com respeito a α , resultando

$$w_{j+1} = w_j + \alpha_{\min} d_j.$$

4. Verifica-se se o critério de parada já foi atingido.
5. Calcula-se o novo vetor gradiente g_{j+1} .
6. Calcula a nova direção de busca, utilizando $d_{j+1} = -g_{j+1} + \beta_j d_j$, onde β_j pode

ser calculado através da fórmula de Fletcher-Reeves $\beta_j = \frac{g_{j+1}^T g_{j+1}}{g_j^T g_j}$.

O algoritmo de gradiente conjugado utiliza um procedimento de busca da direção a ser seguida no espaço dos pesos denominado *line search*, que dispensa o cálculo da matriz Hessiana no emprego das derivadas de segunda ordem.*

O problema com esse procedimento é que envolve um número de cálculos da função erro muito elevado, exigindo um tempo computacional excessivo. Além disso, a performance do algoritmo fica sensível aos parâmetros escolhidos como critério de parada, podendo interferir no desempenho da rede.

Um modo de contornar tais problemas, considerando um algoritmo totalmente automatizado, não dependente da escolha de nenhum parâmetro, e que evite o consumo de tempo computacional despendido na *line search*, foi introduzido por Moller²¹ em 1993, denominado *gradiente conjugado escalonado*, que sugere um modo mais simples e computacionalmente eficiente de calcular a matriz Hessiana utilizando o método de Levenberg-Marquardt, com o objetivo de definir o tamanho do passo α . Para garantir

* A matriz Hessiana representa a segunda derivada de uma função, no nosso caso da função erro, e seus

elementos são dados por $\frac{\partial^2 E}{\partial w_{ji} \partial w_{jk}}$.

que a matriz Hessiana seja positiva definida, podemos adicionar a ela algum múltiplo da matriz identidade, na forma

$$H + \lambda I,$$

onde λ deve ser um número positivo suficientemente grande. Se escolhermos $\lambda_j = 0$, o vetor de pesos irá para o mínimo na direção escolhida somente se a função erro for uma função quadrática e se a matriz Hessiana for positiva-definida. Caso contrário, o valor de λ_j deve ser acrescido de acordo.

O algoritmo gradiente conjugado escalonado pode ser resumido como segue:

1. A partir de um vetor de pesos iniciais w_0 , e de valores arbitrários de $\sigma > 0$, $\lambda_j > 0$ e $\bar{\lambda}_j = 0$, estabelece-se $d_j = g_j = -E'(w_0)$, $j = 1$ e uma variável booleana *sucesso* = true.

2. Se *sucesso* = true, calcula-se

$$\sigma_j = \frac{\sigma}{\|d_j\|}, \quad s_j = \frac{E'(w_j + \sigma_j d_j) - E'(w_j)}{\sigma_j} \quad \text{e} \quad \delta_j = d_j^T s_j.$$

3. Escalona-se s_j por $s_j = s_j + (\lambda_j - \bar{\lambda}_j) d_j$ e $\delta_j = \delta_j + (\lambda_j - \bar{\lambda}_j) \|d_j\|^2$.

4. Se $\delta_j \leq 0$, então transforma-se a matriz Hessiana em positiva definida, fazendo

$$s_j = s_j + \left(\lambda_j - 2 \frac{\delta_j}{\|d_j\|^2} \right) d_j, \quad \bar{\lambda}_j = 2 \left(\lambda_j - \frac{\delta_j}{\|d_j\|^2} \right), \quad \delta_j = -\delta_j + \lambda_j \|d_j\|^2 \quad \text{e} \quad \lambda_j = \bar{\lambda}_j$$

5. Calcula-se o tamanho do passo como sendo $\mu_j = d_j^T g_j$ e $\alpha_j = \frac{\mu_j}{\delta_j}$.

6. Calcula-se um parâmetro de comparação

$$\Delta_j = \frac{2\delta_j [E(w_j) - E(w_j + \sigma_j d_j)]}{\mu_j^2}.$$

7. Se $\Delta_j \geq 0$, então diminui-se o erro fazendo

$$w_{j+1} = w_j + \alpha_j d_j, \quad g_{j+1} = -E'(w_{j+1}) \quad \text{e} \quad \bar{\lambda}_j = 0.$$

- a. Se não houve convergência, então reinicia-se o algoritmo com $d_{j+1} = g_{j+1}$. Caso contrário, cria-se uma nova direção de busca

$$\beta_j = \frac{\|g_{j+1}\|^2 - g_{j+1}^T g_j}{\mu_j} \quad \text{com} \quad d_{j+1} = g_{j+1} + \beta_j d_j.$$

- b. Se $\Delta_j \geq 0,75$ então diminui-se o parâmetro de escalonamento
- $$\lambda_j = \frac{1}{2} \lambda_j. \text{ Caso contrário, uma redução no erro não é possível e}$$
- $$\bar{\lambda}_j = \lambda_j, \text{ sucesso} = \text{falso.}$$
8. Se $\Delta_j < 0,25$ então aumenta-se o parâmetro de escalonamento
- $$\lambda_j = 4\lambda_j$$
9. Se a inclinação da direção $g_j \neq 0$, então estabelece-se $j = j + 1$ e volta para o passo 2. Caso contrário, o valor do mínimo desejado w_{j+1} foi encontrado.

Mais métodos de otimização podem ser citados, como o Método de Newton, Quasi-Newton, entre outros.

3.3.3.4 Critério de parada

Diferentes critérios podem ser utilizados para interromper o treinamento da rede, como por exemplo o número de iterações, o tempo de treinamento, um valor específico para a função erro, um valor específico para a variação do erro ou quando o erro na etapa de teste começa a aumentar. O critério a ser escolhido dependerá do problema que está sendo tratado, assim como da própria arquitetura da rede.

No caso de problemas de regressão, uma função erro normalizada é apropriada para avaliar o treinamento da rede, comparando os valores de saída gerados pela rede com os valores desejados. No caso de um problema de divisão em classes, uma taxa de classificação, dada pela proporcionalidade de acertos das respostas da rede pode ser considerada.

4 Metodologia

O objetivo deste trabalho é a avaliação da habilidade de um modelo conexionista no reconhecimento de possíveis padrões apresentados em séries temporais de ativos financeiros. Neste capítulo, descrevemos a metodologia empregada na escolha dos ativos, do período analisado, no tratamento dos dados e na estruturação do modelo de redes neurais.

4.1 A escolha do ativo objeto

O mercado de ações foi escolhido como objeto deste estudo considerando-se que foi o único mercado brasileiro a não sofrer intervenções diretas do Banco Central do Brasil no período considerado, como o caso do mercado de juros e dólar. Os ativos foram escolhidos tomando-se como base a importância refletida no mercado acionário brasileiro, bem como a liquidez, fundamental para um bom modelo explicativo baseado em retornos diários. Um histórico consistente e suficientemente numeroso também foi levado em consideração, garantindo um maior embasamento estatístico. Os ativos escolhidos como objeto desse estudo foram o Índice da Bolsa de Valores de São Paulo (Ibovespa), a ação preferencial da Telemar S.A. (TNLP4) e a ação preferencial da Petrobrás S.A. (PETR4).

4.1.1 Índice da Bolsa de Valores de São Paulo

O Ibovespa é o mais importante indicador do desempenho médio das cotações do mercado de ações brasileiro, já que retrata o comportamento dos principais papéis negociados na BOVESPA (Bolsa de Valores de São Paulo), responsável por aproximadamente 90% do total transacionado em todas as bolsas brasileiras.²² As empresas emissoras das ações integrantes da sua carteira teórica são responsáveis, em média, por aproximadamente 70% do somatório da capitalização bursátil de todas as empresas negociáveis na BOVESPA.

Além disso, o Índice Bovespa é o único dos indicadores de performance de ações brasileiras a ter um mercado futuro líquido, sendo inclusive um dos maiores de contrato de índice do mundo, garantindo assim liquidez na construção de posições vendidas para o modelo.

O valor do índice é o valor atual, em moeda corrente, de uma carteira teórica de ações constituída em 02 de janeiro de 1968. Supõe-se não ter sido efetuado nenhum investimento adicional desde então, considerando-se somente a reinversão dos dividendos recebidos e do total apurado com a venda dos direitos de subscrição, além da manutenção, em carteira, das ações recebidas a título de bonificação. O cálculo do índice se dá pelo somatório dos pesos (quantidade teórica da ação multiplicada pelo último preço da mesma) das ações integrantes da sua carteira teórica. Desde sua implementação, o índice não sofreu modificações metodológicas, sendo adequado às condições do mercado brasileiro, unicamente para efeito de divulgação, ocorrendo

divisão por 100 em 03/10/1983, e divisões por 10 em 02/12/1985, 29/08/1988, 14/04/1989, 12/01/1990, 28/05/1991, 21/01/1992, 26/01/1993, 27/08/1993 e 10/02/1994.

A carteira teórica do índice é integrada pelas ações que, em conjunto, representaram 80% do volume transacionado à vista nos doze meses anteriores à formação da carteira. Como critério adicional, exige-se que a ação represente no mínimo 80% de presença nos preços do período. A participação de cada ação na carteira tem relação direta com a representatividade desse título no mercado à vista – em termos de número de negócios e volume em moeda corrente – ajustado ao tamanho da amostra.

Para que a representatividade do índice se mantenha ao longo do tempo, quadrimestralmente é feita uma reavaliação do mercado, sempre com base nos últimos 12 meses, onde se identificam as alterações na participação relativa e cada ação. Feito isso, monta-se uma nova carteira, atribuindo-se a cada papel um novo peso, segundo a distribuição de mercado, apurada pelo estudo de reavaliação.

4.1.2 Telemar PN

A maior empresa de telecomunicações da América do Sul em número de linhas de serviço, e a maior do Brasil em faturamento e número de telefones instalados, a Telemar é responsável por 65% de todo o território brasileiro, estando presente em 16 estados no Norte, Nordeste e Sudeste do país. Com larga experiência em serviços de telefonia fixa local e de longa distância, disponibiliza também serviços para Internet, transmissão de dados e imagens e videoconferência, entre outros.

Antes da constituição da Telebrás, em 1972, havia mais de 900 companhias de telecomunicações operantes no Brasil. Entre 1972 e 1975 a Telebrás e suas controladoras operacionais adquiriram quase todas essas companhias telefônicas, passando a ter o monopólio na prestação de serviços públicos de telecomunicações em quase todas as regiões do país. A partir de 1995, o Governo Federal deu início a uma ampla reforma da regulamentação do setor de telecomunicações e, em julho de 1997, o Congresso Nacional aprovou a Lei Geral das Telecomunicações, dispendo sobre a criação de um novo marco regulador, a introdução da concorrência e a privatização da Telebrás. Em maio de 1998, a Telebrás foi reestruturada para formar, além da Telebrás, 12 novas empresas controladoras através de uma cisão: três holdings das concessionárias regionais de telefonia fixa, uma holding da operadora de longa distância e oito holdings das concessionárias da telefonia móvel Banda A. Em 29 de julho de 1998, ocorreu o leilão público para a privatização das companhias cindidas, a maior delas sendo a Tele Norte Leste, transformada em Telemar em abril de 1999.

Em maio de 2000, os papéis de Telemar passaram a integrar o Índice Bovespa, sendo a Telemar PN a ação de maior participação no índice atualmente.

Mais de 80% de suas ações são negociadas na Bolsa de Valores de São Paulo e na New York Stock Exchange, através do programa de ADRs representativos de ações preferenciais, estando entre os títulos com maior liquidez de empresas brasileiras.

Além disso, em outubro de 2001 foram lançados contratos futuros de Telemar, que podem ser uma ferramenta útil na construção das posições vendidas no ativo.²³

A Telemar PN é caracterizada pelo mercado financeiro como um ação predominantemente de *trade*, ou seja, objeto de negociações de curta duração que procuram aproveitar as várias oscilações presentes em uma tendência mais longa, por pertencer a um segmento mais volátil (telecomunicações).

4.1.3 Petrobrás PN

A Petrobrás é uma sociedade anônima de capital aberto que atua de forma integrada e especializada em vários seguimentos relacionados à indústria de óleo, gás e energia exploração e produção, como refino, comercialização, transporte e petroquímica, distribuição de derivados, gás natural e energia, tanto no mercado nacional quanto internacional.

Criada em 1953, a Petrobrás é hoje a uma das mais importantes companhias brasileiras, sendo a 13ª maior companhia petrolífera do mundo, e a 5ª maior refinaria mundial.

Os papéis da Petrobrás compõem a carteira do Índice Bovespa desde a sua criação, em 1968. Em maio de 2000, foi realizado um agrupamento das ações que compõem o capital social da companhia, na proporção de cem ações para uma nova ação, propiciando o acesso do pequeno investidor, com aumento de liquidez das ações e a conseqüente expansão da base acionária. Em agosto do mesmo ano, foram agregados cerca de 330 mil novos acionistas brasileiros, que utilizaram seus recursos depositados no FGTS ou optaram por investir diretamente em papéis da Petrobrás.

Desde 1996 o ADR da ação preferencial é negociado no mercado de balcão americano, e a partir de fevereiro de 2001, na NYSE.

Da mesma forma que a Telemar PN, desde outubro de 2001 podem ser negociados contratos futuros da ação preferencial²⁴.

A Petrobrás PN é caracterizada pelo mercado financeiro como uma ação de *posição*, ou seja, negociada principalmente por investidores cujo objetivo é adquirir e manter a ação por um período mais longo de tempo, buscando aproveitar uma tendência, por pertencer a um segmento mais sólido (petrolífero).

Os dois papéis, TNLP4 e PETR4, ocupam atualmente a liderança no Índice Bovespa, apresentando participação relativa de 13% e 9% respectivamente*.

4.2 A escolha do período

O período avaliado para o Ibovespa e Petrobrás PN foi de 04 de julho de 1994, ou o início do Plano Real, que representa uma importante redução da influência da

* Dados referentes à composição da carteira teórica do Ibovespa no dia 02/01/2002.

inflação na atitude de investimento dos agentes econômicos brasileiros, a 28 de dezembro de 2001, totalizando 1.852 dias de negociação. Esse período abrange as grandes crises ocorridas no mercado recentemente – Crise do México (dezembro de 1994), Crise da Ásia (Indonésia em julho de 1997, Tailândia em agosto de 1997, Malásia em setembro de 1997 e Coréia em novembro de 1997), Crise da Rússia (agosto de 1998), Crise Brasileira (desvalorização em janeiro de 1999) e Crise Argentina (iniciando-se em outubro de 2000, com agravamento no final de 2001), o que garante situações de volatilidade diferenciadas para o treinamento da rede, resultando em uma rede mais flexível.

No caso da Telemar PN, o período analisado foi de 21 de setembro de 1998, início da negociação da ação em bolsa, até 28 de dezembro de 2001, em um total de 809 dias de negociação.

Os períodos considerados podem ser avaliados na Figura 4.1, Figura 4.2 e Figura 4.3. A área mais escura abrange o período utilizado no treinamento, correspondente a 2/3 do período total considerado, sendo os dados restantes utilizados na fase de teste da rede. A utilização de dados completamente disjuntos nas fases de treinamento e teste garantem uma maior percepção da capacidade de generalização da rede.

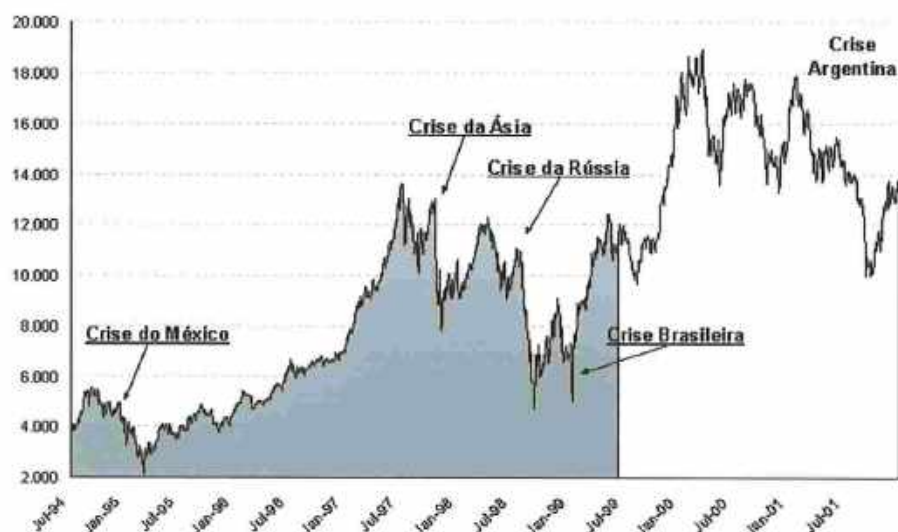


Figura 4.1 – Ibovespa em pontos de julho de 1994 a dezembro de 2001



Figura 4.2 – Preço da Petrobrás PN (PETR4) de julho de 1994 a dezembro de 2001

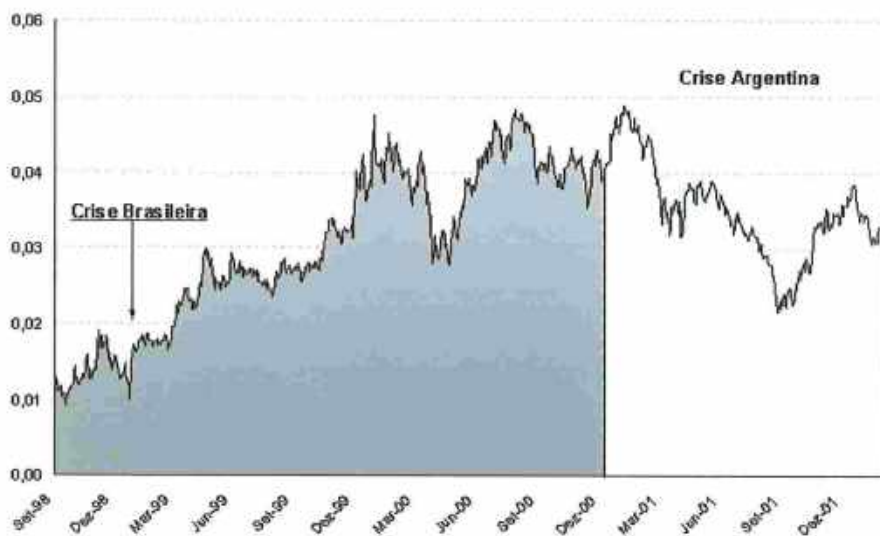


Figura 4.3 – Preço de Telemar PN (TNLP4) de setembro de 1998 a dezembro de 2001

4.3 Tratamento dos dados de entrada

A partir da série de preços, foram calculados os retornos diários dos ativos. Em um primeiro momento a série de retornos foi dividida em períodos de 5 dias úteis, utilizados como entrada do modelo. Contudo, com o objetivo de capturar um possível efeito dia da semana, foram considerados períodos de segunda a sexta-feira, sendo que os dados faltantes, por serem exclusivamente correspondentes a feriados, dada a grande liquidez dos ativos, foram repetidos pela cotação anterior.

A janela de previsão foi estruturada como mostra a Figura 4.4. A partir dos retornos diários da semana corrente, a rede é treinada buscando extrair um padrão de comportamento que explique a variação de preços da semana seguinte, gerando uma ordem de compra ou venda, de acordo com a indicação do movimento do ativo de alta ou baixa.

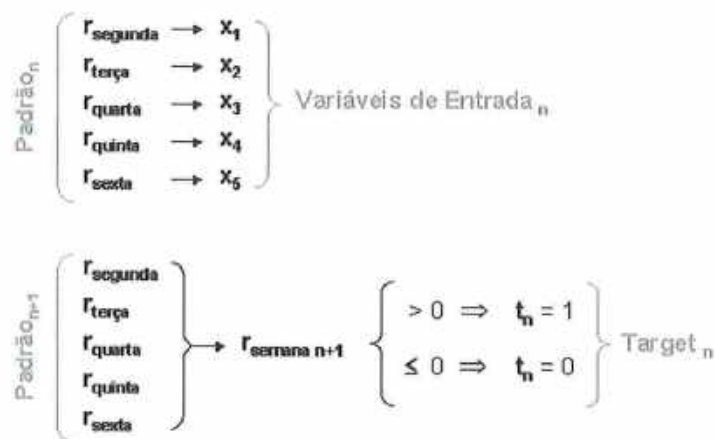


Figura 4.4 – Esquema representativo para os dados de entrada da rede

Caso o retorno da semana referente ao $padrão_{n+1}$ for positivo, indicando um sinal de compra para aquele período, será atribuído o valor 1 para o target t_n do $padrão_n$, a ser comparado com a resposta y_n gerada pela rede. Caso seja negativo ou nulo, o valor do target t_n será considerado como sendo 0.

4.4 A arquitetura de rede utilizada

Os algoritmos foram estruturados na linguagem Matlab, tendo como base a ferramenta Netlab²⁵.

Do período total analisado, dois terços foram destinados ao treinamento da rede e o outro terço restante à fase de teste. Assim, no caso do Ibovespa e Petrobrás PN, 260 padrões com 5 dados de entrada cada, totalizando 1.300 dados, foram utilizados na fase de treinamento e 130 padrões, ou 650 dados, na fase de teste. No caso da Telemar PN,

que possui um histórico mais curto, 114 padrões, ou 570 dados, foram destinados ao período de treinamento e 57 padrões, correspondentes a 285 dados, foram utilizados como entrada no período de teste. Os dados foram apresentados à rede de maneira *offline* (em paralelo), mas atualizações do modelo para a utilização da prática podem ser feitas *online* (um a um)²⁶.

Foi utilizada uma topologia de rede tipo feed-forward multicamada, com duas camadas escondidas, como demonstrado na Figura 4.5. Foram apresentados à rede 5 dados de entrada, sendo os retornos diários do ativo de segunda a sexta-feira, além do valor de *target*, tendo como saída a resposta obtida pela rede sinalizando o movimento do ativo na semana seguinte, 1 para compra e 0 para venda.

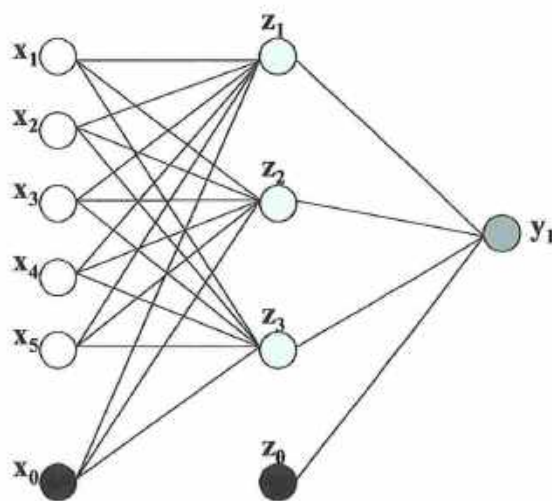


Figura 4.5 – Topologia da rede feed-forward utilizada no treinamento da rede. O número de unidades na camada escondida foi variado de 2 a 17

As redes foram estruturadas tendo de 2 a 17 unidades na camada oculta, com intervalos de 3 unidades. O critério de parada escolhido foi o número de iterações, variando de 100 a 20.100, com intervalos de 1.000 iterações. Assim, combinações entre o número de unidades na camada escondida e o número de iterações foram treinadas, resultando em 126 diferentes arquiteturas de rede.

O algoritmo de aprendizado empregado foi o back-propagation, utilizando como método de otimização o gradiente conjugado escalonado. Dois tipos diversos de função de ativação foram aplicados, a função tangente hiperbólica nas unidades da camada escondida, que, como citado no Capítulo 3, converge mais rapidamente no período de treinamento do que a função logística sigmóide, e a função logística sigmóide nas unidades de saída, por sua característica representação de probabilidades a posteriori. Dadas as funções de ativação, a função erro a ser otimizada escolhida como mais apropriada foi a função cross-entropy.

Diversas outras simulações com as diferentes arquiteturas de redes foram realizadas, de acordo com os resultados obtidos, visando maximizar a compreensão das respostas geradas pelo modelo. Tais resultados são demonstrados a seguir.

5 Resultados

Os critérios utilizados para mensurar a performance das diferentes redes treinadas foram a taxa de classificação e a lucratividade resultantes no período de teste. Neste capítulo descrevemos tais critérios, apresentamos os resultados obtidos e aplicamos um teste de hipótese como medida de consistência.

5.1 Taxa de classificação

Como visto, a abordagem proposta nesse estudo foi contextualizada em um problema de divisão em classes. Assim, uma medida de performance da rede indicada é a taxa de classificação, dada pelo percentual de respostas corretas geradas pela rede.

A partir das combinações entre o número de unidades ocultas e o número de iterações descritas no Capítulo 4, as redes foram treinadas e, medindo-se os acertos percentuais para cada uma delas, resultaram em 126 diferentes taxas de classificação. Para que qualquer efeito indevido ocasionado pelos pesos iniciais fosse expurgado, cada arquitetura foi então simulada 5 vezes, mantendo inalterados os dados de entrada. Os resultados mostraram-se muito consistentes, demonstrando estabilidade em relação às taxas de classificação, como é mostrado nas tabelas apresentadas no Apêndice. Foi calculado um valor médio com os 5 valores das simulações e as configurações de rede que apresentaram valor máximo de taxa de classificação também obtiveram a melhor taxa média.

Os resultados de todas as combinações efetuadas são mostrados a seguir. Na Figura 5.1, podemos ver as melhores taxas de classificação obtidas para Telemar PN no período de treinamento em todas as combinações efetuadas, com o número de unidades escondidas variando de 2 a 17 e o número de iterações variando de 100 a 20.100. A Figura 5.2 mostra os mesmos dados, mas com outra vista. As cores mais frias indicam taxas de classificação mais baixas, enquanto as cores mais quentes indicam valores mais altos. Podemos notar que a taxa de classificação aumenta com o acréscimo de unidades ocultas e, principalmente, do número de iterações. Contudo, a partir de 8 unidades ocultas e 4.000 iterações a rede parece já ter aprendido ou decorado os dados na sua totalidade (*overfitting*), apresentando uma taxa de acertos de 100%, sendo que números maiores se tornam desnecessários e um desperdício computacional.

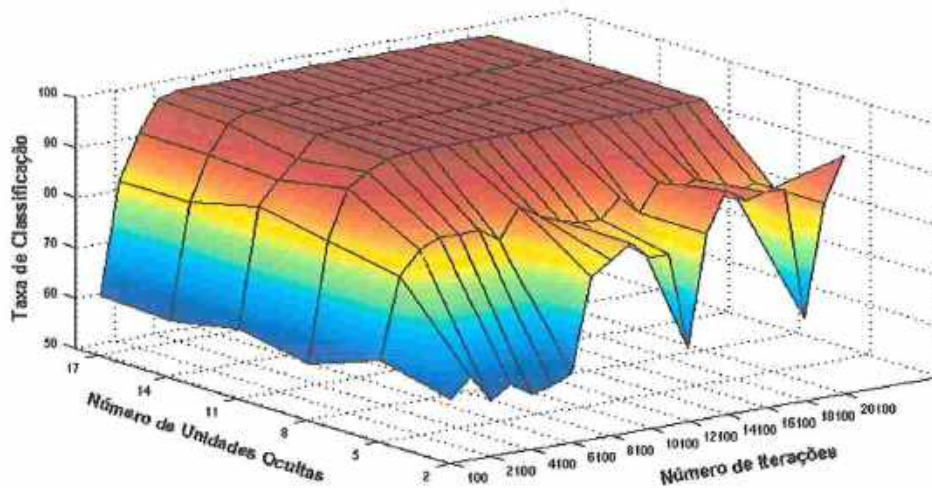


Figura 5.1 – Taxas de classificação obtidas para Telemar PN no período de treinamento – vista 3D

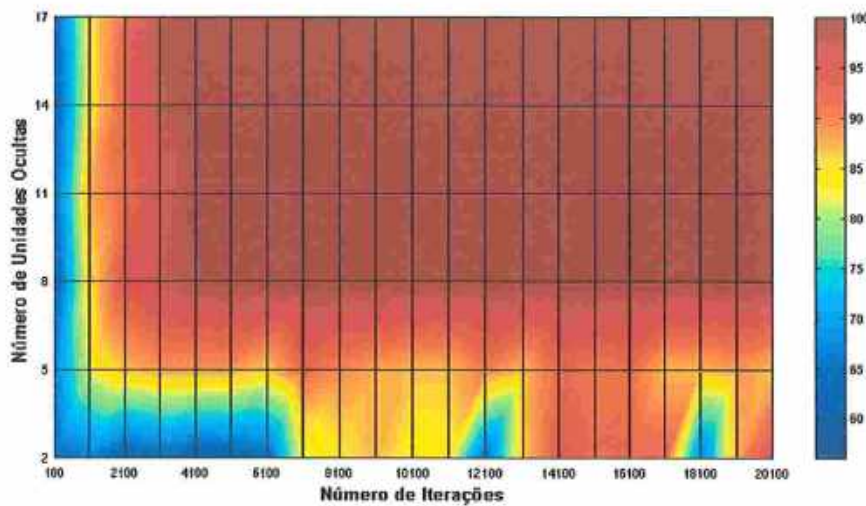


Figura 5.2 - Taxas de classificação obtidas para Telemar PN no período de treinamento – vista 2D

O mais importante para a escolha da melhor rede, contudo, é como a arquitetura se comporta no período de teste. Na Figura 5.3 vemos as taxas de classificação obtidas quando as redes treinadas são propagadas no novo período de teste. Podemos notar na Figura 5.4 que a grande maioria obteve taxas superiores a 52%. Vale notar que uma relação entre número de unidades ocultas utilizada, número de iterações e taxa de acertos não fica bem definida. Configurações que conseguiram taxas de classificação de 100% no período de treinamento, sugerindo um overfitting, demonstraram altas taxas de acerto na fase de teste.

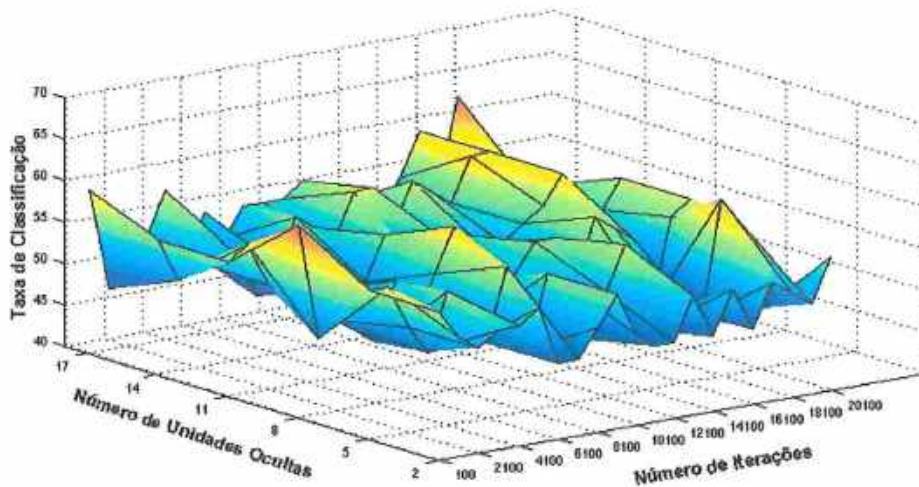


Figura 5.3 – Taxas de classificação obtidas para Telemar PN no período de teste – vista 3D

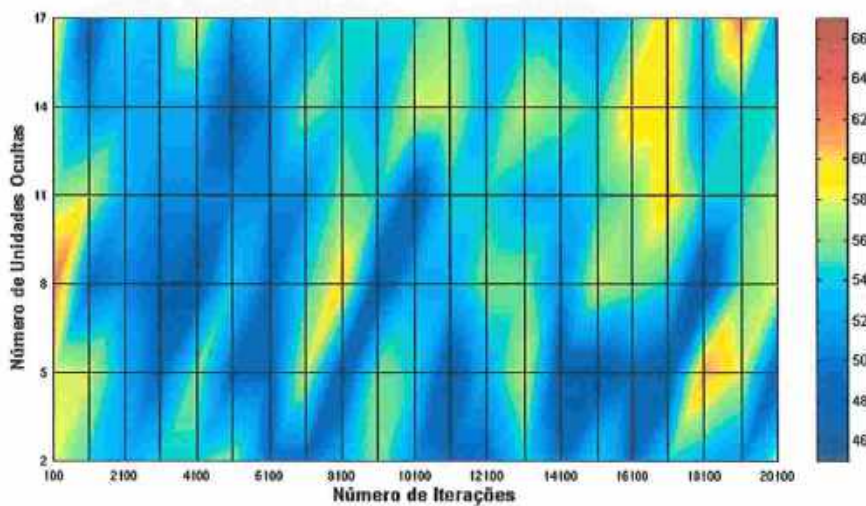


Figura 5.4 - Taxas de classificação obtidas para Telemar PN no período de teste – vista 2D

A rede escolhida, por ter apresentado a melhor taxa de classificação no período de teste, com o valor de 63,16%, possui 8 unidades escondidas e foi treinada com 100 iterações.

Os mesmo dados são mostrados Petrobrás PN, na Figura 5.5 e na Figura 5.6. Podemos notar que são necessárias mais unidades ocultas e um número maior de iterações para que a rede aprenda os dados na fase de treinamento do que no caso da Telemar PN, provavelmente por que o número de padrões apresentados à rede também é maior (260 contra 114).

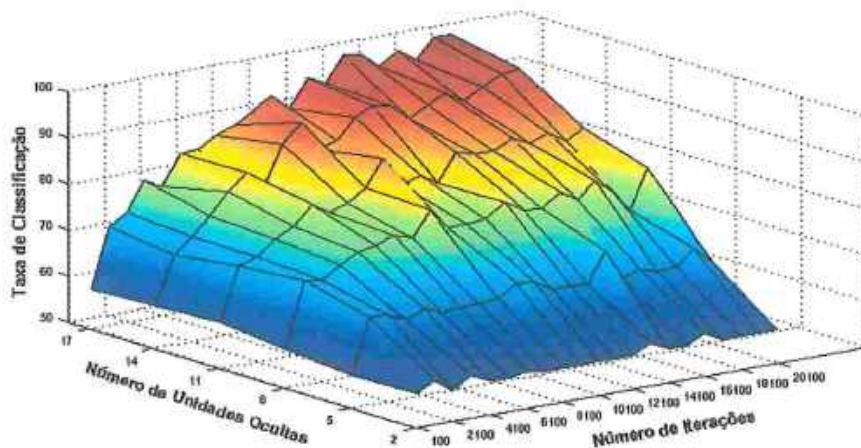


Figura 5.5 - Taxas de classificação obtidas para Petrobrás PN no período de treinamento – vista 3D

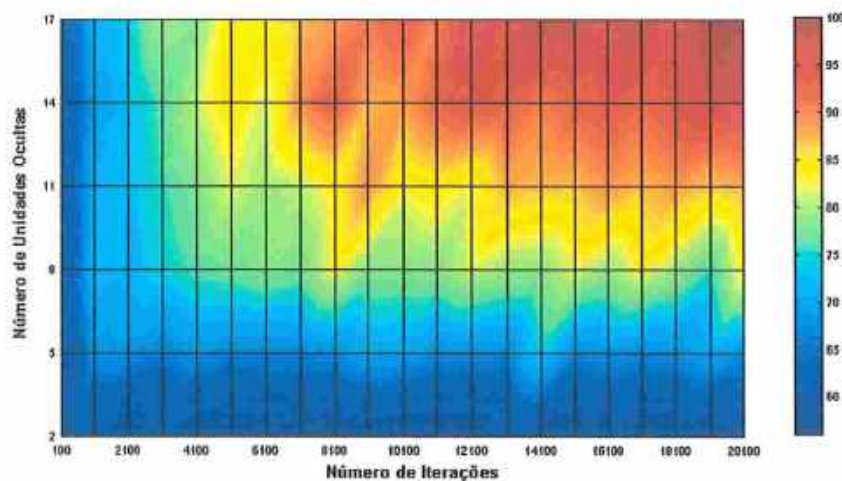


Figura 5.6 - Taxas de classificação obtidas para Petrobrás PN no período de treinamento – vista 2D

Contudo, neste caso, uma relação entre as variáveis avaliadas se mostra mais clara, como mostrado na Figura 5.7 e Figura 5.8, sendo que o percentual de acerto aumenta a medida que o número de unidades ocultas e o número de iterações cresce, atingindo um cume de 67% com 14 unidades ocultas e 18.100 iterações. Novamente, a idéia de overfitting é refutada.

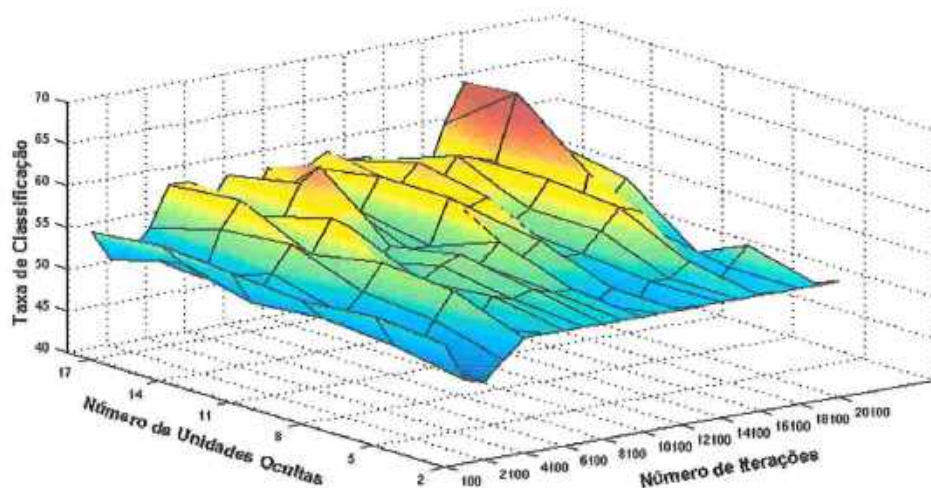


Figura 5.7 - Taxas de classificação obtidas para Petrobrás PN no período de teste – vista 3D

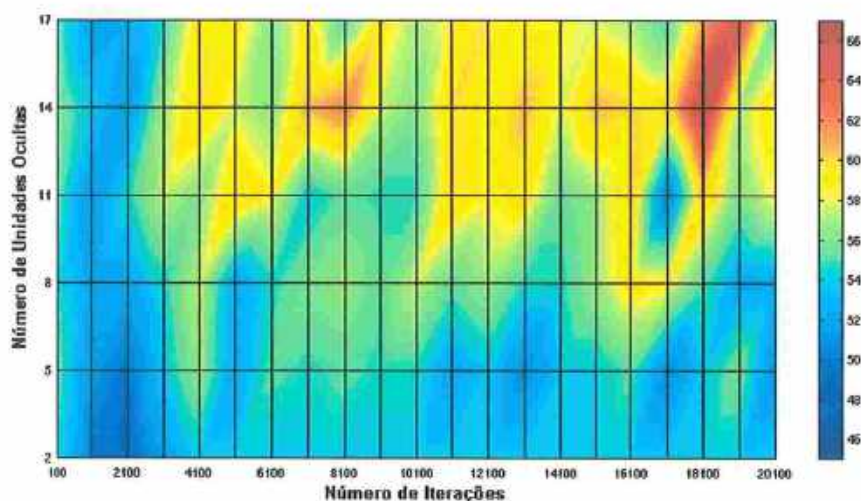


Figura 5.8 - Taxas de classificação obtidas para Petrobrás PN no período de teste – vista 2D

No caso do Ibovespa, a maior taxa de classificação obtida na fase de treinamento foi de 92,69%, inferior a 100% (Figura 5.9 e Figura 5.10), o que pode sugerir que o tempo de treinamento, estabelecido através do número de iterações, foi insuficiente para que o modelo neural fizesse um reconhecimento total dos padrões apresentados

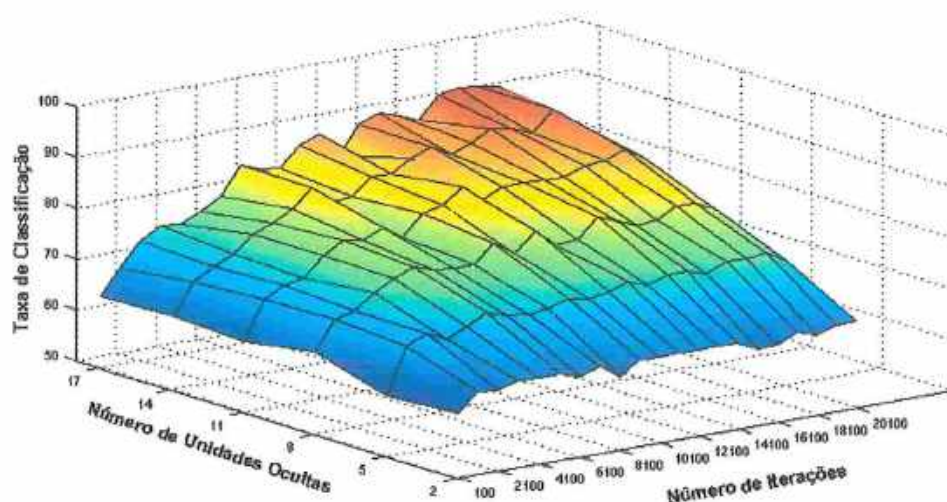


Figura 5.9 - Taxas de classificação obtidas para Ibovespa no período de treinamento – vista 3D

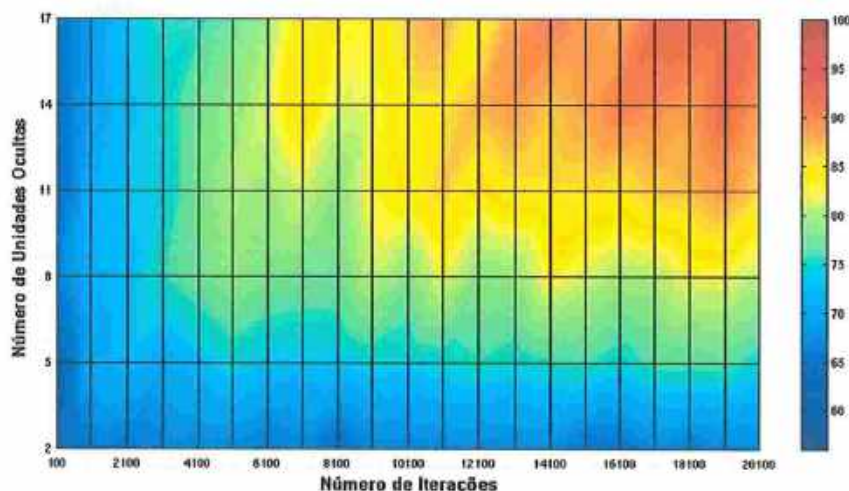


Figura 5.10 - Taxas de classificação obtidas para Ibovespa no período de treinamento – vista 2D

Na fase de teste, configurações com poucas unidades escondidas e/ou iterações mostraram aparentemente melhores performances (Figura 5.11 e Figura 5.12). Contudo, o valor máximo é 55,38%, obtido com 8 unidades ocultas e 14.100 iterações.

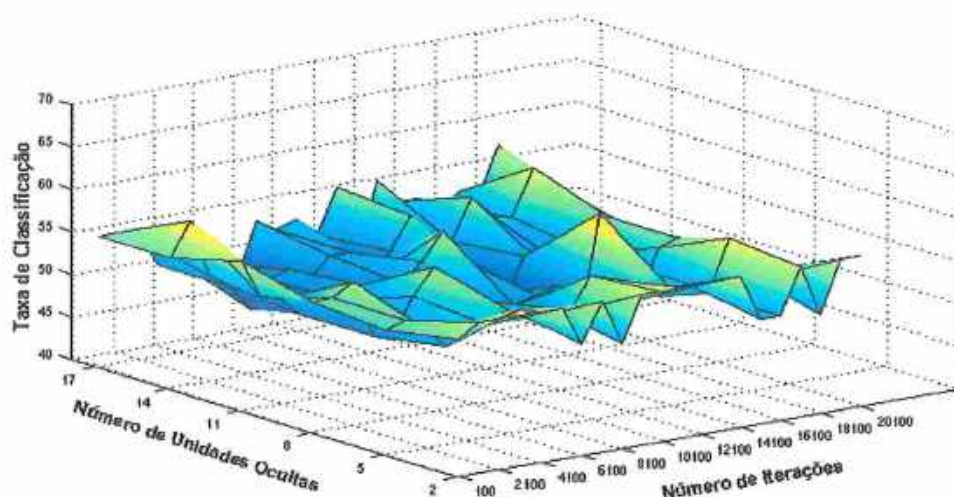


Figura 5.11 - Taxas de classificação obtidas para Ibovespa no período de teste – vista 3D

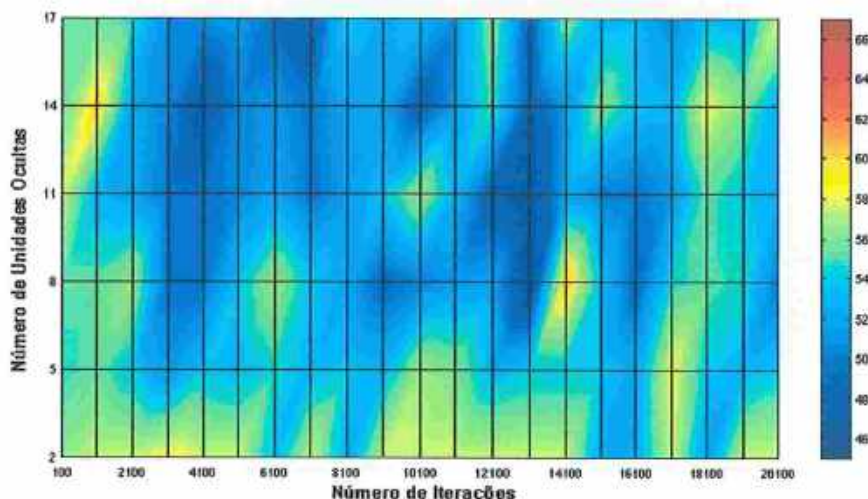


Figura 5.12 - Taxas de classificação obtidas para Ibovespa no período de teste – vista 2D

5.2 Matriz de Acertos e Erros (Confusion Matrix)

A partir das arquiteturas que apresentaram melhores taxas de classificação no período de teste, tomando como critério de avaliação os valores máximos, foram estruturadas as *matrizes de acertos e erros (confusion matrix)* para cada caso.

Os resultados são mostrados a seguir:

		Resposta da rede (y)		Total t	% Acerto
		1	0		
Valores Reais (t)	1	22	6	28	79%
	0	15	14	29	48%
Total y		37	20	57	
% Acerto		59%	70%	63,16%	

Figura 5.13 – Matriz de acertos/erros das respostas da rede com taxa de classificação máxima para Telemar PN

No caso da rede com taxa de classificação máxima na fase de teste para Telemar PN, que como vimos possuía 8 unidades na camada escondida e utilizou 100 iterações no treinamento, podemos estruturar uma matriz de acertos e erros como mostrado na Figura 5.13. Os dados em amarelo referem-se às respostas geradas pela rede (y), enquanto os dados em cinza referem-se aos valores reais desejados (target t). O total de dados do período de teste foi de 57 semanas (ou 57 padrões), sendo que em 28 delas houve um movimento real de alta (1) do ativo e em 29 um movimento de baixa (0). Desses movimentos reais, a rede acertou em 22 semanas de alta (79%) e em 14 de baixa (48%), apresentando assim uma melhor previsibilidade nos períodos altistas, provavelmente explicado pela presença maior de tendências de alta na fase de treinamento. Podemos notar também que a rede sugeriu um maior número de respostas indicando alta nos preços, sendo 37 indicações de alta contra 20 indicações de baixa. Contudo, as indicações de baixa corresponderam melhor à realidade (70% de acertos) do que as indicações de alta (59% de acertos). A taxa de classificação total foi de 63,16% (22 + 14 dados corretos em 57 dados totais).

O mesmo procedimento pode ser aplicado aos demais ativos. A matriz de acertos para a Petrobrás PN pode ser visualizada na Figura 5.14. A rede em questão possuía 14 unidades ocultas e foi treinada com 18.100 iterações, resultando em uma taxa de classificação de 66,92%.

		Resposta da rede (y)		Total t	% Acerto
		1	0		
Valores Reais (t)	1	47	22	69	68%
	0	21	40	61	66%
Total y		68	62	130	
% Acerto		69%	65%	66,92%	

Figura 5.14 - Matriz de acertos/erros das respostas da rede com taxa de classificação máxima para Petrobrás PN

Note que não houveram grandes discrepâncias entre os acertos nos períodos de alta ou baixa. Ao confrontarmos com os gráficos de preços apresentados no Capítulo 4, vemos que foi apresentado ao modelo na fase de treinamento mais exemplos de baixa no caso da Petrobrás PN do que no caso da Telemar PN, o que pode explicar os resultados mais equilibrados.

A matriz de acertos para o Ibovespa é mostrada na Figura 5.15. A configuração de rede que gerou mais respostas corretas foi 8 unidades ocultas com 14.100 iterações, com uma taxa de classificação de 55,38%. A rede acertou mais em períodos de alta do que de baixa (72% contra 36%), obviamente por ter gerado predominantemente indicações de compra (89 para compra contra 41 para venda), fato que não correspondeu à realidade, já que no período considerado os movimentos de alta e de baixa ocorreram em número equilibrado (69 e 61), o que desperta dúvidas sobre a explicabilidade de tal arquitetura.

		Resposta da rede (y)		Total t	% Acerto
		1	0		
Valores Reais (t)	1	50	19	69	72%
	0	39	22	61	36%
Total y		89	41	130	
% Acerto		56%	54%	55,38%	

Figura 5.15 - Matriz de acertos/erros das respostas da rede com taxa de classificação máxima para Ibovespa

5.3 Lucratividade

Conforme visto na seção anterior, das combinações entre número de unidades ocultas e número de iterações foram geradas 126 arquiteturas diferentes, cada uma delas foi simulada 5 vezes para eliminar um possível efeito dos pesos iniciais e, observando-se que as respostas eram estáveis, a arquitetura que obteve maior taxa de classificação foi escolhida para cada ativo.

Nesta seção avaliamos as redes escolhidas quanto à lucratividade referente ao período de teste. Essa análise é de fundamental importância na prática, pois, apesar de a taxa de classificação ser uma boa medida de performance para problemas de separação em classes, na realidade do mercado financeiro ela não nos explica a dimensão dos nossos ganhos ou perdas. De fato, considere uma rede que performa com taxa de classificação de 95%. Então, esse modelo acerta o movimento do ativo em 95% das vezes mas, suponha que os 5% em que houve erro correspondam a um período de stress, por exemplo, no qual ocorre um grande aumento de volatilidade, ampliando a magnitude dos movimentos e, conseqüentemente, os retornos gerados. Na verdade, são

esses períodos os mais propensos a levar o modelo a sugerir posições equivocadas, pois geralmente há uma quebra de padrão. Assim, o retorno resultante desses 5% de erro pode ser negativamente superior ao gerado pelos 95% de acerto.

Dessa forma, tomamos as redes com maior taxa de classificação para cada ativo e calculamos os retornos que teriam sido acumulados no período correspondente à fase de teste. O retorno obtido foi comparado com uma estratégia *buy-and-hold*, ou seja, caso o investidor tivesse efetuado a compra do ativo no início do período e mantido a posição até o final, e com a rentabilidade do ativo livre de risco no mesmo período, que optamos por ser o Certificado de Depósitos Interbancários (CDI).

Inicialmente, foram considerados os retornos resultantes das 5 simulações descritas na seção anterior para cada ativo. Contudo, foi verificada certa instabilidade nos resultados, demonstrando uma sensibilidade aos pesos iniciais, já que todos os outros dados de entrada mantiveram-se constantes. De fato, se apenas uma ordem de compra ou venda for trocada, dependendo dos pesos que inicialmente forem utilizados, o retorno acumulado final pode ser prejudicado ou beneficiado. Procurando uma análise mais justa, as configurações de redes escolhidas foram simuladas 100 vezes, obtendo 100 valores de retornos acumulados, diferenciando-se apenas pelos pesos escolhidos como iniciais.

Os resultados são demonstrados graficamente. Na Figura 5.16 os pontos vermelhos correspondem às rentabilidades acumuladas obtidas nas 100 simulações de Telemar PN, novamente utilizando a rede com melhor taxa de classificação na fase de teste, com 8 unidades ocultas e 100 iterações, no período de 27 de novembro de 2000 a 28 de dezembro de 2001. A linha azul indica a rentabilidade acumulada pela ação no mesmo período, que foi de -5,96%, e a linha verde o retorno acumulado pelo CDI, de 18,89%. Podemos notar que em 90% das vezes a rede gerou retornos superiores ao do ativo, sendo que o retorno máximo obtido pelo modelo foi de 76,24%, cerca de 16 vezes superior ao retorno acumulado da ação, considerando-se que o retorno da ação foi negativo.

Contudo, apenas um desses retornos poderia ter sido obtido em uma situação real. Assim, adotamos uma estratégia combinando todas as respostas geradas (*committee*). Como os valores de saída calculados pela rede variam entre 0% e 100%, para cada semana foi calculada uma média das 100 respostas, e esse valor foi adotado como uma ordem para efetuar a posição, comprada nos casos superiores a 50% e vendida para valores médios inferiores a 50%. O retorno acumulado obtido através dessas “ordens médias” foi de 36,32%, representado na figura pela linha vermelha.

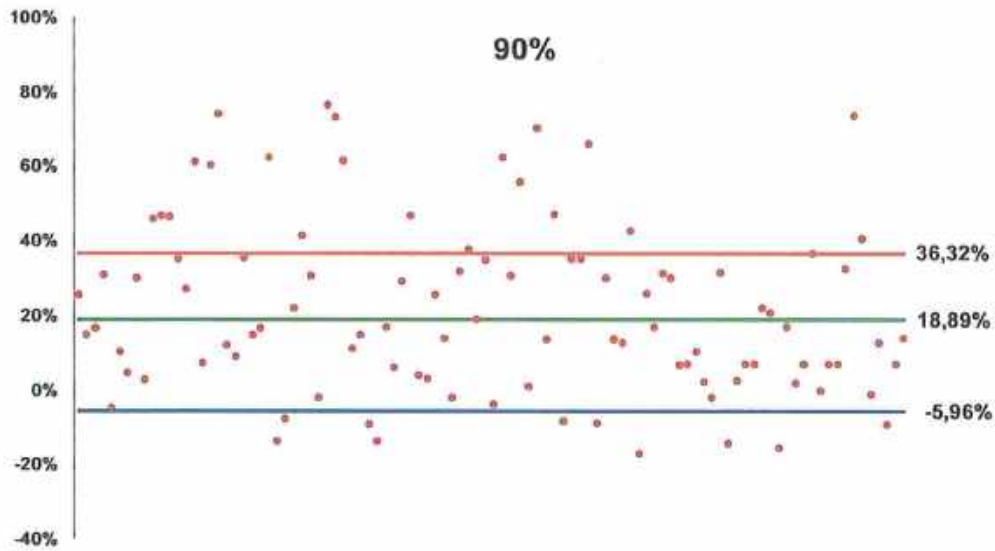


Figura 5.16 – Retornos acumulados no período de teste obtidos através da rede com taxa de classificação máxima para Telemar PN. A linha azul indica o retorno acumulado da ação no mesmo período, a verde o acumulado pelo CDI e a vermelha o retorno acumulado pela média das ordens

Procedendo da mesma forma com a Petrobrás PN, e utilizando novamente a rede com 14 unidades na camada oculta e 18.100 iterações, os retornos foram acumulados no período de teste, de 05 de julho de 1999 a 28 de dezembro de 2001, e são demonstrados na Figura 5.17. Podemos verificar que a rede gerou retornos superiores ao ativo em 69% dos casos, obtendo no melhor caso um retorno acumulado de 749,02%, mais de 750% do valor da rentabilidade acumulada pela ação (98,83%). Utilizando a mesma estratégia de ordens médias, o retorno acumulado no período foi de 404,45%.

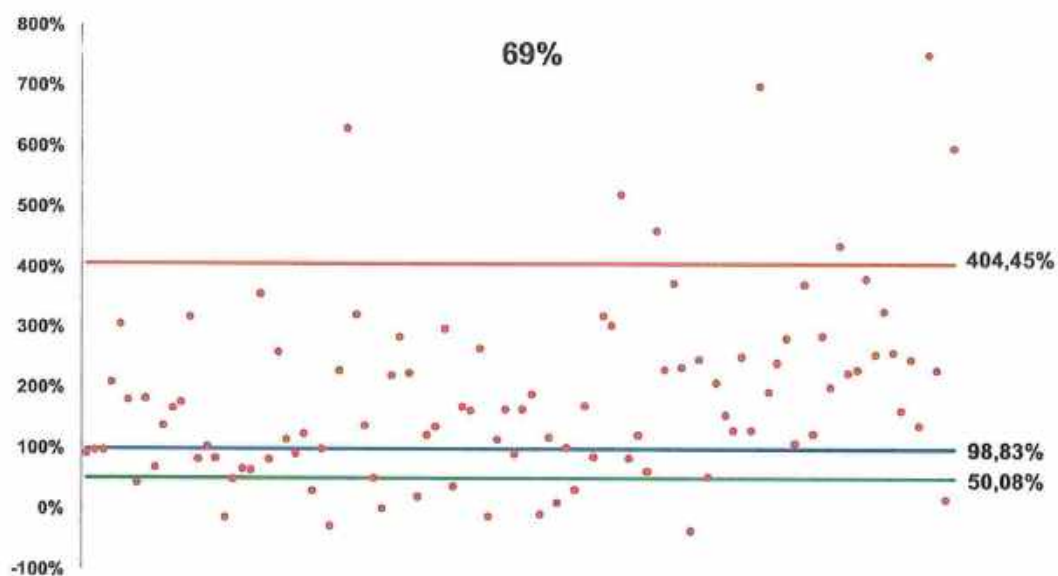


Figura 5.17 - Retornos acumulados no período de teste obtidos através da rede com taxa de classificação máxima para Petrobrás PN. A linha azul indica o retorno acumulado da ação no mesmo período, a verde o acumulado pelo CDI e a vermelha o retorno acumulado pela média das ordens

No caso do Ibovespa, os resultados não foram favoráveis, como mostrado na Figura 5.18. Apenas em 32% das vezes a rede gerou um retorno superior ao acumulado pelo índice (18,68%), e a estratégia de ordens médias resultou em uma rentabilidade acumulada de -15,76%.

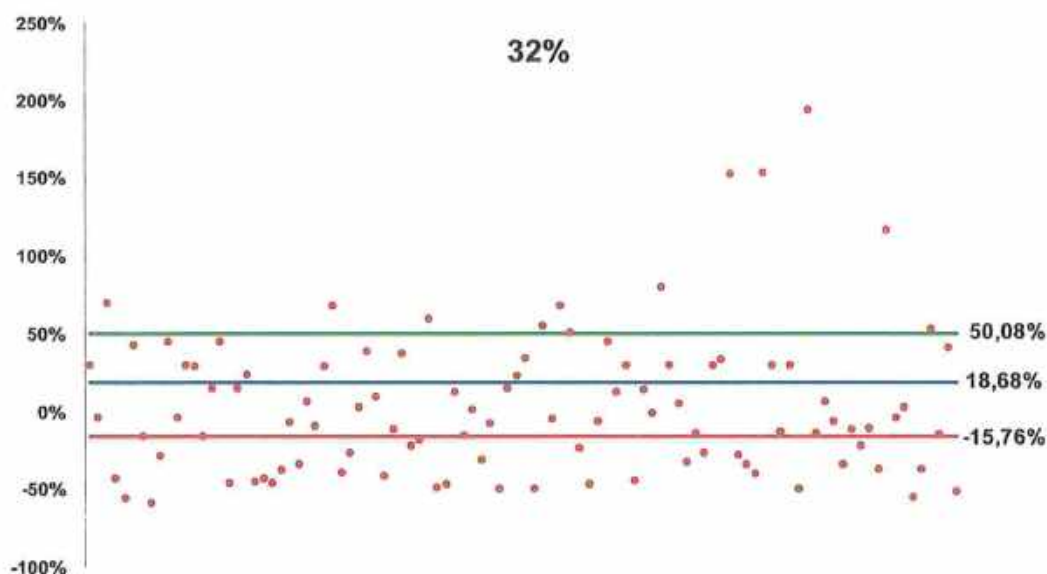


Figura 5.18 - Representação dos retornos acumulados no período de teste obtidos através da rede com taxa de classificação máxima para Ibovespa. A linha azul indica o retorno acumulado da ação no mesmo período, e a linha vermelha o retorno acumulado pela média das ordens

Vale ressaltar que custos operacionais não foram considerados, o que na prática pode representar um impacto considerável sobre a rentabilidade final, de acordo com o número de transações sugeridas pelo modelo.

5.4 Consistência

Como o modelo mostrou-se relativamente sensível aos pesos iniciais, a consistência dos resultados obtidos pode ser avaliada através de um teste de hipóteses simples²⁷. O que queremos saber é se a taxa de classificação obtida realmente representa uma não-aleatoriedade dos dados de saída apresentados pela rede, ou seja, se de um modo efetivo o modelo gera respostas mais corretas do que um processo aleatório qualquer, como por exemplo, “lançar uma moeda”.

Assim, o que queremos testar é se, na média, a taxa de acerto gerada pelo modelo é superior a 50%. Um teste de hipóteses monocaudal sobre a média da população pode ser aplicado, da forma

$$H_0 : \mu \leq \mu_0$$

$$H_a : \mu > \mu_0$$

tomando inicialmente $\mu_0 = 50\%$.

Para tanto, considerando como arquitetura a rede que melhor performou na fase de teste, os dados de entrada do período de teste foram aplicados à rede de modo progressivo 100 vezes, variando apenas os pesos iniciais aleatórios, resultando em $n = 100$ valores de taxa de classificação para cada ativo. Calculando-se a média amostral \bar{x} e o desvio padrão amostral s , podemos encontrar o valor de z dado por

$$z = \frac{\bar{x} - \mu_0}{s / \sqrt{n}}$$

A hipótese nula H_0 é rejeitada se $z > z_{\alpha}$, onde z_{α} é o valor crítico dado pelo nível de significância α especificado.

Os dados aplicados ao teste de hipóteses foram os mesmos utilizados na avaliação da lucratividade.

Assim, para a Telemar PN, uma rede com 8 unidades escondidas e utilizando número de iterações igual a 100 foi propagada 100 vezes, gerando uma média amostral para taxa de classificação $\bar{x} = 54,49\%$ e um desvio amostral $s = 3,52\%$. Calculando z com $\mu_0 = 50\%$, encontramos $z = 12,74$, ou seja, a hipótese nula pode ser rejeitada com segurança. Na verdade, mesmo tomando $\mu_0 = 53\%$ a hipótese nula ainda é rejeitada em um nível de significância de 1%. Ou seja, com 99% de certeza podemos afirmar que a média das taxas de acerto geradas pelo modelo, com a configuração de rede especificada, é superior a 53%.

Procedendo da mesma forma para Petrobás PN, com uma rede contendo 14 unidades ocultas e sendo otimizada em 18.100 iterações, encontramos $\bar{x} = 55,09\%$ e $s = 4,25\%$, o que nos permite afirmar com 99% de confiança que este modelo gera, na média, taxas de classificação superiores a 54%.

Já no caso do Ibovespa, utilizando novamente a rede que apresentou melhor taxa de classificação na fase de teste, com 8 unidades na camada escondida e 14.100 iterações, $\bar{x} = 55,09\%$ e $s = 4,25\%$, obtivemos um valor de $\bar{x} = 48,92\%$ e $s = 3,56\%$, resultando em um valor de $z = -3,02$. Assim, não rejeitamos H_0 , ou seja, não podemos afirmar que essa configuração de rede para os dados de Ibovespa acerte mais suas previsões do que erre.

6 Conclusão

Neste trabalho estudamos a aplicação de redes neurais artificiais em um problema de duas classes separáveis, através do reconhecimento de possíveis padrões apresentados pelas séries históricas dos retornos de Ibovespa, Telemar PN e Petrobrás PN. O modelo foi baseado em uma rede multicamada feed-forward, utilizando back-propagation como algoritmo de aprendizado, otimizado pelo método gradiente conjugado escalonado. Diferentes configurações de rede foram avaliadas, variando no número de unidades na camada escondida e no número de iterações no processo de otimização. Foram utilizados como dados de entrada os retornos diários dos ativos em períodos semanais, gerando como saída a previsão da tendência para a semana seguinte.

Os resultados mostraram que, tanto para Telemar PN quanto para Petrobrás PN, a rede escolhida como tendo a melhor performance para cada ativo, com base na taxa de classificação apresentada no período de teste, mostrou-se consistente, tendo um desempenho significativo no que diz respeito à lucratividade, indicando anomalias no comportamento dos preços. O lucro acumulado pelo modelo no período de teste atingiu 76,24% no caso de TNLP4, equivalente a 16 vezes o valor acumulado pelo ativo no mesmo período, e 749,02% no caso de PETR4, sendo 7,5 vezes o real alcançado pela ação.

No caso do Ibovespa os resultados não foram favoráveis. A configuração de rede que atingiu o melhor desempenho no período de teste não se mostrou consistente e não alcançou uma boa lucratividade, apresentando retorno abaixo do índice na maioria dos casos. Este fato pode ter ocorrido devido a uma interrupção precoce no período de treinamento, o que indicaria uma maior complexidade do Ibovespa em comparação com os outros ativos. Uma possível não-convergência do erro na fase de otimização também pode ser responsável por tais resultados. Além disso, como o Índice Bovespa é composto por uma cesta de ativos, o estudo de padrões de comportamento baseados em preços históricos pode ficar prejudicado, já que os ativos que compõem a carteira são alterados quadrimestralmente.

Vale destacar alguns fatos registrados durante o estudo:

- Redes com um menor número de unidades na camada oculta produzem respostas que giram em torno de 0,5, enquanto topologias com um maior número de unidades ocultas apresentam saídas com valores próximos de 0 ou 1. Concluímos, portanto, que a interpretação das saídas como probabilidades a posteriori, de acordo com o proposto pela teoria, fica prejudicada.
- Redes possuindo um menor número de unidades na camada escondida geraram valores de saída mais estáveis, ou seja, menos dependentes dos pesos iniciais utilizados, provavelmente devido ao fato de possuírem uma matriz de pesos conseqüentemente mais reduzida.
- Muitos autores citam que um overfitting na fase de treino tende a gerar piores resultados no período de teste. Como verificado pelos resultados, tal argumento não foi comprovado, já que no caso de Petrobrás PN, redes com altas taxas de

não foi comprovado, já que no caso de Petrobrás PN, redes com altas taxas de classificação na fase de treinamento obtiveram melhores taxas também na fase de teste.

- Como citados por alguns autores, foi verificada uma relação diretamente proporcional entre o número ótimo de unidades na camada oculta e o número de padrões apresentados à rede.

Dada a validação da metodologia utilizada neste trabalho, e no intuito de refinar e potencializar ainda mais os resultados, identificamos os seguintes trabalhos futuros:

- Neste trabalho empregamos um modelo de duas classes. Problemas com um maior número de estados classificáveis podem ser considerados, através de redes binárias de n saídas. Estados como compra, venda, aplicação em ativo sem risco, zeragem de posição, entre outros, podem ser utilizados.
- Na prática, modelos com entrada de dados online apresentam um menor custo computacional e uma capacidade de adaptação mais dinâmica do que modelos estruturados de forma offline. Uma implementação do modelo proposto, alterando-o para online pode ter grande contribuição em sua performance.
- Diferentes períodos devem ser testados, tanto para estruturação dos dados de entrada, com o objetivo de captar padrões mais longos ou de menor periodicidade, como por exemplo um efeito final de mês, como para os períodos de treinamento e teste, abrangendo uma memória mais curta ou longa.

Referência Bibliográfica

- ¹ Haugen, A. Robert, 1995, "The New Finance – The Case Against Efficient Markets": Prentice Hall.
- ² Roberts, Harry V., 1959, "Stock Market Patterns and Financial Analysis: Methodological Sugestions": *Journal of Finance* 14, 1-10.
- ³ Bruni, Adriano L. e Famá, Rubens, 1998, "Eficiência, Previsibilidade dos Preços e Anomalias em Mercados de Capitais: Teoria e Evidências", *Caderno de Pesquisas em Administração* 1, 7.
- ⁴ Fama, Eugene F., 1970, "Efficient Capital Markets: A Review of Theory and Empirical Work": *The Journal of Finance* 25, 383-417.
- ⁵ Murphy, John J., 1986, "*Technical Analysis of the Future Markets*": New York Institute of Finance.
- ⁶ Achelis, Steven B., 1995, "*Technical Analysis from A to Z*": McGraw-Hill.
- ⁷ Sá, Geraldo T., 1987, "*Mercado de Ações e Bolsa de Valores*": Editora Técnica.
- ⁸ Copeland, Thomas E. e Weston, Fred J., 1992, "*Financial Theory and Corporate Policy*" – 3ª Edição: Addison Wesley.
- ⁹ Lo, Andrew W. e Mackinlay, A. Craig, 1988, "Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test": *The Review of Financial Studies* 1, 41-66.
- ¹⁰ Jegadeesh, Narasimhan, 1990, "Evidence of Predictable Behaviour of Security Returns": *The Journal of Finance* 45, 881-898.
- ¹¹ Fama, Eugene F., 1991, "Efficient Capital Markets: II": *The Journal of Finance* 46, 1575-1617.
- ¹² Brock, William, Lakonishok, Josef e LeBaron, Blake, 1992, "Simple Technical Trading Rules and the Stochastic Properties of Stock Returns": *The Journal of Finance* 17, 1731-1764.
- ¹³ Fama, Eugene F., 1998, "Market Efficiency, Long-Term Returns, and Behavioral Finance": *The Journal of Finance* 49, 283-306.
- ¹⁴ Andersen, J. V., Gluzman, S. e Sornette D., 2000, "Fundamental Framework for Technical Analysis": *The European Physical Journal B*, 579.
- ¹⁵ Lewellen, Jonathan e Shanken, Jay, 2000, "Estimation risk, market efficiency, and the predictability of returns": *National Bureau of Economic Research*, Cambridge.
- ¹⁶ Lo, Andrew W., Mamaysky Harry e Wang, Jiang, 2000, "*Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation*": Laboratory for Financial Engineering, Massachusetts Institute of Technology.

¹⁷ McCulloch, W. S. e Pitts, W., 1943, “A Logical Calculus of the Ideas Immanent in Nervous Activity”: *Bulletin of Mathematical Biophysics* 5, 115-133.

¹⁸ Hebb, Donald O., 1949, “*The Organization of Behaviour*”: John Wiley.

¹⁹ Notas de aula da disciplina Redes Neurais, Programa de Mestrado em Modelagem Matemática Aplicada à Finanças, Universidade de São Paulo, agosto de 2001.

²⁰ Bishop, Christopher M., 1995, “*Neural Networks for Pattern Recognition*”: Oxford University Press.

²¹ Moller, M., 1993, “A scaled conjugate gradient algorithm for fast supervised learning”, *Neural Networks* 6, 525-533.

²² Bolsa de Valores de São Paulo, “Índice Bovespa – Definição e Metodologia”.

²³ Telemar Participações S.A., “*Prospecto Preliminar de Distribuição Pública de Debêntures*”, março de 2002.

²⁴ Petrobrás S. A., “*Relatório Anual 2000*”.

²⁵ Nabney, Ian T., 2002, “*Netlab: Algorithms for Pattern Recognition*”: Springer.

²⁶ Vicente, Renato, 1997, “*Otimização Funcional de Algoritmos em Redes Neurais Multicamada*”: Instituto de Física da Universidade de São Paulo.

²⁷ Morettin, Pedro A. e Bussab, Wilton, 2002, “*Estatística Básica*”: Editora Saraiva.

Apêndice

Apresentamos a seguir as máximas taxas de classificação obtidas na fase de teste a partir das diferentes redes treinadas, e a média calculada com base em 5 simulações para cada configuração, mantendo-se inalterados os parâmetros de entrada e dependendo apenas da aleatoriedade dos pesos iniciais, como descrito no Capítulo 5. Podemos notar que houve consistência nas respostas apresentadas pelo modelo.

Taxa de Classificação Máxima para o Período de Treinamento para Telemar PN

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	63%	66%	61%	64%	61%	62%
	1.100	66%	82%	85%	88%	84%	84%
	2.100	61%	87%	95%	92%	93%	93%
	3.100	64%	88%	97%	96%	98%	99%
	4.100	61%	88%	99%	100%	100%	100%
	5.100	62%	88%	100%	100%	100%	100%
	6.100	64%	86%	100%	100%	100%	100%
	7.100	82%	92%	100%	100%	100%	100%
	8.100	85%	89%	100%	100%	100%	100%
	9.100	88%	88%	100%	100%	100%	100%
	10.100	84%	87%	100%	100%	100%	100%
	11.100	84%	87%	100%	100%	100%	100%
	12.100	65%	90%	100%	100%	100%	100%
	13.100	87%	87%	100%	100%	100%	100%
	14.100	95%	92%	100%	100%	100%	100%
	15.100	92%	90%	100%	100%	100%	100%
	16.100	93%	91%	100%	100%	100%	100%
	17.100	93%	87%	100%	100%	100%	100%
	18.100	66%	87%	100%	100%	100%	100%
	19.100	88%	88%	100%	100%	100%	100%
20.100	97%	87%	100%	100%	100%	100%	

Taxa de Classificação Média para o Período de Treinamento para Telemar PN

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	60%	64%	61%	61%	60%	61%
	1.100	62%	80%	83%	86%	82%	83%
	2.100	60%	82%	90%	90%	92%	92%
	3.100	64%	85%	95%	94%	98%	97%
	4.100	63%	82%	94%	98%	99%	100%
	5.100	63%	86%	97%	100%	100%	100%
	6.100	62%	85%	100%	100%	100%	100%
	7.100	62%	81%	98%	100%	100%	100%
	8.100	62%	87%	98%	100%	100%	100%
	9.100	63%	85%	98%	100%	100%	100%
	10.100	60%	86%	99%	100%	100%	100%
	11.100	62%	85%	100%	100%	100%	100%
	12.100	63%	85%	99%	100%	100%	100%
	13.100	62%	77%	100%	100%	100%	100%
	14.100	63%	85%	99%	100%	100%	100%
	15.100	63%	84%	100%	100%	100%	100%
	16.100	63%	88%	100%	100%	100%	100%
	17.100	64%	84%	100%	100%	100%	100%
	18.100	63%	87%	100%	100%	100%	100%
	19.100	63%	83%	100%	100%	100%	100%
20.100	63%	87%	100%	100%	100%	100%	

Taxa de Classificação Máxima para o Período de Teste para Telemar PN

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	58%	58%	63%	56%	56%	60%
	1.100	54%	58%	49%	58%	51%	47%
	2.100	53%	53%	51%	53%	53%	54%
	3.100	54%	49%	49%	51%	53%	53%
	4.100	54%	56%	47%	51%	53%	58%
	5.100	56%	49%	54%	51%	47%	51%
	6.100	49%	49%	49%	51%	51%	54%
	7.100	49%	58%	54%	51%	56%	51%
	8.100	53%	47%	60%	56%	54%	54%
	9.100	56%	58%	47%	54%	53%	54%
	10.100	51%	54%	53%	47%	58%	54%
	11.100	49%	47%	53%	54%	58%	56%
	12.100	49%	53%	56%	54%	53%	53%
	13.100	54%	58%	56%	53%	58%	53%
	14.100	49%	47%	51%	53%	56%	53%
	15.100	54%	47%	58%	54%	54%	51%
	16.100	49%	49%	56%	56%	60%	54%
	17.100	54%	47%	54%	60%	60%	60%
	18.100	53%	61%	46%	54%	51%	53%
	19.100	51%	58%	56%	54%	54%	63%
20.100	56%	47%	58%	58%	54%	53%	

Taxa de Classificação Média para o Período de Teste para Telemar PN

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	56%	52%	57%	54%	54%	56%
	1.100	51%	52%	44%	54%	50%	45%
	2.100	50%	51%	47%	47%	47%	48%
	3.100	53%	47%	47%	49%	50%	49%
	4.100	52%	52%	46%	47%	52%	52%
	5.100	51%	46%	51%	49%	47%	46%
	6.100	49%	48%	48%	50%	49%	54%
	7.100	49%	54%	52%	50%	53%	46%
	8.100	51%	47%	54%	53%	48%	52%
	9.100	51%	50%	45%	54%	49%	51%
	10.100	50%	51%	50%	46%	50%	50%
	11.100	49%	46%	51%	54%	55%	50%
	12.100	49%	48%	53%	51%	49%	51%
	13.100	53%	55%	51%	50%	50%	51%
	14.100	49%	46%	48%	50%	49%	48%
	15.100	50%	46%	54%	49%	53%	48%
	16.100	49%	47%	53%	47%	48%	53%
	17.100	52%	45%	49%	53%	50%	54%
	18.100	50%	51%	46%	48%	49%	49%
	19.100	50%	54%	50%	51%	52%	54%
20.100	51%	47%	53%	55%	51%	51%	

Taxa de Classificação Máxima para o Período de Treinamento para Petrobrás PN

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	57%	57%	58%	60%	59%	58%
	1.100	60%	69%	73%	72%	70%	72%
	2.100	57%	69%	72%	73%	73%	74%
	3.100	59%	67%	75%	76%	77%	80%
	4.100	57%	69%	77%	78%	81%	78%
	5.100	57%	67%	77%	82%	85%	85%
	6.100	58%	68%	79%	80%	81%	85%
	7.100	57%	68%	78%	81%	89%	87%
	8.100	57%	68%	83%	83%	93%	88%
	9.100	57%	71%	78%	89%	87%	91%
	10.100	57%	70%	78%	83%	88%	93%
	11.100	57%	68%	79%	84%	92%	90%
	12.100	57%	68%	82%	82%	94%	96%
	13.100	59%	68%	80%	88%	93%	94%
	14.100	57%	74%	79%	87%	90%	100%
	15.100	57%	68%	82%	87%	92%	99%
	16.100	59%	69%	80%	89%	94%	96%
	17.100	57%	68%	83%	87%	92%	98%
	18.100	57%	68%	80%	88%	94%	96%
	19.100	57%	71%	75%	86%	96%	100%
20.100	57%	68%	84%	88%	97%	100%	

Taxa de Classificação Média para o Período de Treinamento para Petrobrás PN

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	57%	57%	57%	58%	58%	58%
	1.100	56%	68%	71%	71%	69%	69%
	2.100	56%	68%	71%	72%	72%	72%
	3.100	57%	66%	73%	75%	75%	75%
	4.100	57%	67%	75%	75%	78%	75%
	5.100	57%	66%	75%	80%	80%	81%
	6.100	58%	67%	75%	79%	79%	82%
	7.100	57%	66%	73%	79%	86%	84%
	8.100	57%	67%	79%	82%	86%	85%
	9.100	57%	68%	77%	85%	84%	87%
	10.100	57%	68%	75%	81%	85%	91%
	11.100	57%	68%	77%	81%	90%	88%
	12.100	57%	67%	76%	82%	88%	93%
	13.100	58%	66%	76%	83%	91%	93%
	14.100	57%	69%	77%	85%	86%	95%
	15.100	57%	67%	76%	85%	92%	94%
	16.100	58%	68%	76%	85%	92%	91%
	17.100	57%	66%	78%	85%	89%	95%
	18.100	57%	68%	78%	86%	92%	90%
	19.100	57%	70%	73%	84%	92%	96%
20.100	57%	66%	77%	86%	92%	98%	

Taxa de Classificação Máxima para o Período de Teste para Petrobrás PN

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	55%	55%	55%	55%	56%	55%
	1.100	50%	52%	52%	52%	54%	52%
	2.100	49%	50%	52%	55%	52%	52%
	3.100	52%	54%	55%	58%	57%	55%
	4.100	54%	57%	58%	56%	60%	59%
	5.100	54%	52%	52%	60%	58%	59%
	6.100	54%	55%	55%	58%	56%	56%
	7.100	54%	55%	56%	54%	61%	59%
	8.100	54%	55%	57%	55%	62%	55%
	9.100	54%	55%	55%	55%	58%	59%
	10.100	54%	55%	58%	55%	56%	57%
	11.100	54%	52%	55%	60%	60%	58%
	12.100	54%	54%	57%	58%	58%	60%
	13.100	54%	51%	55%	59%	61%	58%
	14.100	54%	54%	55%	55%	58%	58%
	15.100	54%	54%	57%	57%	61%	58%
	16.100	54%	55%	59%	60%	60%	57%
	17.100	54%	51%	58%	50%	59%	55%
	18.100	54%	54%	55%	60%	67%	61%
	19.100	54%	56%	52%	56%	57%	65%
20.100	54%	51%	52%	58%	60%	55%	

Taxa de Classificação Média para o Período de Teste para Petrobrás PN

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	52%	54%	54%	54%	53%	54%
	1.100	45%	48%	50%	49%	51%	48%
	2.100	47%	49%	50%	54%	49%	50%
	3.100	49%	52%	53%	54%	52%	53%
	4.100	54%	52%	53%	54%	57%	54%
	5.100	54%	51%	51%	55%	54%	56%
	6.100	53%	52%	52%	57%	54%	56%
	7.100	54%	51%	53%	52%	57%	56%
	8.100	54%	53%	52%	53%	57%	52%
	9.100	52%	51%	52%	52%	54%	58%
	10.100	52%	51%	54%	53%	54%	53%
	11.100	54%	51%	52%	58%	54%	54%
	12.100	51%	52%	51%	56%	56%	58%
	13.100	52%	50%	50%	55%	56%	57%
	14.100	52%	51%	50%	51%	53%	52%
	15.100	53%	51%	53%	52%	53%	56%
	16.100	52%	52%	52%	58%	57%	55%
	17.100	52%	50%	54%	48%	56%	53%
	18.100	52%	51%	54%	52%	62%	56%
	19.100	52%	51%	49%	51%	55%	58%
20.100	53%	50%	50%	56%	58%	54%	

Taxa de Classificação Máxima para o Período de Treinamento para Ibovespa

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	63%	62%	66%	64%	65%	64%
	1.100	66%	71%	72%	72%	70%	69%
	2.100	66%	73%	73%	73%	73%	73%
	3.100	67%	71%	76%	74%	75%	76%
	4.100	67%	73%	78%	78%	78%	75%
	5.100	67%	75%	79%	80%	79%	77%
	6.100	66%	73%	78%	80%	82%	80%
	7.100	67%	73%	78%	82%	85%	85%
	8.100	64%	73%	78%	79%	82%	83%
	9.100	67%	75%	81%	83%	82%	83%
	10.100	67%	74%	78%	84%	85%	87%
	11.100	67%	75%	82%	87%	85%	88%
	12.100	67%	76%	79%	83%	87%	83%
	13.100	67%	75%	79%	85%	89%	89%
	14.100	67%	76%	83%	85%	87%	90%
	15.100	65%	76%	82%	85%	88%	89%
	16.100	65%	75%	80%	85%	90%	88%
	17.100	67%	77%	80%	87%	90%	92%
	18.100	66%	77%	82%	87%	88%	93%
	19.100	67%	77%	82%	88%	92%	92%
20.100	67%	75%	80%	86%	89%	92%	

Taxa de Classificação Média para o Período de Treinamento para Ibovespa

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	62%	62%	65%	63%	64%	63%
	1.100	65%	71%	71%	70%	70%	69%
	2.100	65%	71%	72%	73%	72%	72%
	3.100	66%	70%	74%	72%	74%	74%
	4.100	65%	72%	76%	77%	76%	75%
	5.100	65%	74%	77%	78%	77%	77%
	6.100	64%	73%	75%	79%	79%	76%
	7.100	65%	73%	77%	80%	83%	81%
	8.100	64%	73%	76%	77%	80%	83%
	9.100	66%	74%	78%	81%	81%	81%
	10.100	66%	74%	77%	82%	83%	85%
	11.100	66%	71%	78%	83%	83%	86%
	12.100	66%	74%	78%	80%	85%	81%
	13.100	66%	74%	77%	83%	87%	87%
	14.100	67%	76%	81%	84%	85%	88%
	15.100	64%	76%	78%	82%	87%	89%
	16.100	65%	73%	79%	83%	87%	87%
	17.100	65%	76%	78%	83%	88%	90%
	18.100	65%	75%	79%	84%	87%	88%
	19.100	66%	76%	81%	86%	87%	91%
20.100	66%	74%	80%	83%	88%	90%	

Taxa de Classificação Máxima para o Período de Teste para Ibovespa

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	56%	55%	55%	58%	55%	55%
	1.100	58%	55%	55%	53%	59%	55%
	2.100	58%	54%	57%	52%	52%	55%
	3.100	58%	52%	50%	50%	51%	51%
	4.100	58%	54%	50%	50%	48%	51%
	5.100	58%	53%	54%	51%	51%	52%
	6.100	53%	55%	57%	52%	52%	49%
	7.100	58%	53%	54%	50%	50%	48%
	8.100	52%	54%	52%	52%	52%	54%
	9.100	58%	53%	49%	54%	52%	52%
	10.100	58%	57%	51%	57%	48%	53%
	11.100	58%	56%	52%	53%	52%	51%
	12.100	58%	53%	52%	48%	55%	56%
	13.100	58%	54%	46%	47%	48%	48%
	14.100	58%	54%	60%	52%	52%	56%
	15.100	52%	54%	54%	51%	56%	53%
	16.100	52%	52%	49%	50%	54%	53%
	17.100	58%	58%	55%	52%	53%	51%
	18.100	52%	54%	55%	55%	58%	53%
	19.100	58%	53%	55%	54%	56%	52%
20.100	58%	54%	50%	52%	52%	58%	

Taxa de Classificação Média para o Período de Teste para Ibovespa

		Número de Unidades Ocultas					
		2	5	8	11	14	17
Número de Iterações	100	55%	54%	54%	57%	54%	55%
	1.100	55%	51%	53%	51%	56%	54%
	2.100	53%	52%	54%	50%	51%	53%
	3.100	57%	51%	46%	49%	50%	46%
	4.100	54%	53%	49%	49%	46%	49%
	5.100	54%	51%	49%	49%	49%	48%
	6.100	53%	52%	51%	49%	48%	47%
	7.100	54%	51%	49%	48%	48%	47%
	8.100	52%	51%	50%	49%	50%	51%
	9.100	54%	48%	47%	52%	50%	49%
	10.100	56%	54%	48%	53%	46%	49%
	11.100	56%	54%	49%	49%	49%	46%
	12.100	54%	50%	50%	47%	51%	52%
	13.100	54%	50%	46%	47%	48%	46%
	14.100	56%	54%	54%	51%	49%	52%
	15.100	52%	54%	51%	49%	50%	53%
	16.100	52%	50%	47%	49%	50%	51%
	17.100	53%	55%	50%	47%	50%	49%
	18.100	52%	54%	50%	53%	55%	53%
	19.100	55%	53%	51%	52%	52%	51%
20.100	54%	53%	49%	47%	47%	54%	