

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

**Estudo sobre a empregabilidade da previsão do índice
BOVESPA usando Redes Neurais Artificiais**

Autor:

Reginaldo do Carmo Roque

Orientador:

Prof. Flávio Luís de Mello, D.Sc.

Examinador:

Examinador:

DEL

Fevereiro de 2009

DEDICATÓRIA

Dedico este trabalho a todas as pessoas que não puderam ter acesso a um ensino de qualidade neste país de exclusão, de modo que por mérito tivessem acesso a Universidade.

AGRADECIMENTOS

A Deus, por ter me dado força e sabedoria para buscar o meu caminho.

A minha família, pelo grande esforço para que eu tivesse uma educação de qualidade, em especial ao meu irmão Reinaldo pela abertura de portas, pela inspiração e pelo suporte sempre que precisei.

A minha cunhada Bianca Ramos, pela força no início desta caminhada e novamente no final da mesma.

À família que me dei: Naiara, Leonardo e Patrick, pela inspiração, pelas alegrias e por fazerem de mim um ser humano muito melhor.

Aos meus grandes amigos Diego Braga, Leandro Vieira, Junior Nogueira e Luiz Henrique Bensin, por serem meus amigos, na concepção máxima que este termo pode ter.

Aos outros amigos que fiz durante o curso pré-vestibular e que tanto me ajudaram para que eu chegasse a este momento, incluindo professores e colegas de turma.

À Mariana, por ser tudo o que é.

À Dyamilla, pela amizade genuína e verdadeira, apesar da distância.

À Marcela Trindade, minha primeira chefe, pela delicadeza e pelo incentivo constante.

Aos amigos e colegas do curso de Engenharia, que compartilharam comigo as alegrias e as tristezas deste período, representados aqui por: Suzana Cândida, Roberto Ilídio, Luiz Felipe Aquino, Luiz Henrique Pereira, Augusto Proença, Gustavo Pereira, Gustavo Bergter, Luiz Carlos Monteso, Cláudio Rogel, Diogo Barros, Diogo 'Profeta', Felipe Santana, Pedro Salazar, Rafael Rosa, Melk, Rafael Costa, Jonathan Pinto, Rafael Amaro, Paulo César da Costa, Rodrigo Brito, Leonardo Cidade, Thiago Arakaki, Marcos Magalhães, Rodrigo Semeghini.

Ao Allan Dieguez pela amizade, paciência e fundamental ajuda no desenvolvimento deste trabalho.

Ao Professor Flávio Mello, por acreditar na minha capacidade de realizar este projeto.

E aos amigos que a vida e a violência me tiraram o prazer de suas companhias.

*“Com trabalho, inteligência e economia só
é pobre quem não quer ser rico.”*

Marquês de Maricá

RESUMO

Este trabalho apresenta um estudo sobre a empregabilidade da previsão de índices de mercado, em especial o IBOVESPA, usando Redes Neurais Artificiais. Para isso, um modelo de previsão usando Redes Neurais Artificiais foi desenvolvido e empregado para modelar o período de estudo. Por fim, foi feita uma análise dos resultados do modelo desenvolvido em relação a sua capacidade de previsão.

Palavras-Chave: Redes Neurais, Previsão de índices de mercado, IBOVESPA.

ABSTRACT

This study presents an attempt to forecast the BOVESPA index. In this way a prediction model was developed using Neural Networks in order to model the study periods. At end an analisys of the obtained results was developed regarding its prediction capacity.

Key-words: Neural Networks, Stock Market Index Forecast, IBOVESPA.

SUMÁRIO

<u><i>Sumário</i></u>	<u>7</u>
<u><i>SIGLAS</i></u>	<u>11</u>
<u><i>GLOSSÁRIO</i></u>	<u>13</u>
<u><i>Lista de Tabelas</i></u>	<u>15</u>
<u><i>1 – Introdução</i></u>	<u>18</u>
<u>1.1 – Tema</u>	<u>18</u>
<u>1.2 – Delimitação</u>	<u>18</u>
<u>1.3 – Justificativa</u>	<u>18</u>
<u>1.4 – Objetivos</u>	<u>19</u>
<u>1.5 – Metodologia</u>	<u>20</u>
<u>1.6 – Descrição</u>	<u>20</u>
<u><i>2 – Fundamentos do mercado financeiro</i></u>	<u>21</u>
<u>2.1 – Eficiência de mercado</u>	<u>21</u>
2.1.1 – Formas de eficiência de mercado.....	22
2.1.2 – Testes de eficiência de mercado.....	23
2.1.2.1 – Casos especiais.....	24
2.1.2.1.1 – Efeito fim-de-semana.....	24
2.1.2.1.2 – Efeito Janeiro.....	24
2.1.2.1.3 – Splits.....	25
<u>2.2 – Métodos analíticos na previsão do retorno de ações</u>	<u>26</u>
2.2.1 – Análise Técnica.....	26
2.2.1.1 – Indicadores.....	27
2.2.1.1.1 – Médias Móveis.....	27
2.2.1.1.2 – Volume.....	28
2.2.1.1.3 – Tendências.....	28
2.2.1.1.4 – Momento.....	28
2.2.1.2 – Teoria de Dow.....	29
2.2.2 – Análise Fundamentalista.....	32
2.2.3 – Séries Temporais.....	33
2.2.3.1 – Modelos de Box-Jenkins.....	34
2.2.3.2 – Modelos neurais de previsão.....	35
2.2.4 – Teoria do Caos.....	35
2.2.5 – Outras técnicas computacionais.....	36
<u>2.3 – previsão do retorno de ÍNDICES DE ações</u>	<u>37</u>
2.3.1 – Previsão do nível e de direção utilizando redes neurais.....	39
<u>2.4 – o IBOVESPA</u>	<u>40</u>
2.4.1 – Metodologia do cálculo.....	41
2.4.1.1 – Participação da ação na carteira teórica.....	41
2.4.1.2 – Apuração do Índice.....	42
2.4.1.3 – Alteração da quantidade teórica em função de proventos.....	43
2.4.1.4 – Cálculo do Preço ‘Ex-Teórico’.....	43
<u><i>3 – As Redes Neurais Artificiais</i></u>	<u>45</u>
<u>3.1 – Introdução</u>	<u>45</u>

3.2 – O que é uma rede neural?	46
3.3 – Cérebro humano	47
3.4 – Modelos de neurônios	49
3.4.1 – Modelo de McCulloch-Pitts.....	49
3.4.2 – Funções de Ativação.....	51
3.5 – O processo de aprendizado	52
3.5.1 – Paradigmas de aprendizado.....	53
3.5.1.1 – Aprendizado supervisionado.....	53
3.5.1.2 – Aprendizado não-supervisionado.....	54
3.5.1.3 – Aprendizado por reforço.....	55
3.5.2 – Regras de aprendizado.....	56
3.5.2.1 – Regra Delta ou Correção de Erros.....	56
3.5.2.2 – Aprendizado Hebbiano.....	56
3.5.2.3 – Aprendizado competitivo.....	58
3.6 – Algumas Arquiteturas de RNAs	59
3.6.1 – Perceptron de camada única.....	60
3.6.2 – Perceptron Multicamadas.....	61
3.6.2.1 – Definição de uma topologia ideal para uma rede multicamadas.....	63
3.6.2.2 – Treinamento de redes multicamadas.....	64
3.6.2.2.1 – Back-propagation.....	64
3.7 – Redes Neurais temporais	65
3.7.1 – redes recorrentes.....	66
3.7.1.1 – Rede Jordan.....	67
3.7.1.2 – Rede Elman.....	68
3.7.2 – Redes com atraso no tempo.....	68
4 – Metodologia	70
4.1 – Introdução	70
4.2 – Dados de entrada e saída	70
4.2.1 – Obtendo os dados da série histórica.....	70
4.2.2 – Variáveis de entrada da RN.....	71
4.2.3 – Pré-processamento das Variáveis de entrada da RN.....	73
4.2.4 – Variável de saída da RN.....	74
4.3 – A rede neural	74
4.3.2 – Arquitetura da Rede.....	74
4.3.3 – INICIALIZAÇÃO DA REDE.....	78
4.3.4 – Treinamento, Validação e teste.....	79
4.4 – Métricas para análise dos resultados	81
4.4.1 – Cálculo da taxa de acerto de direção.....	81
5.1 – INTRODUÇÃO	84
5.1 – rede neural com configuração não satisfatória	84
5.2 – Resultados da Rede neural com configuração satisfatória	91
5.2.1 – taxa acerto de direção e percentual de acurácia da saída do sistema.....	95
5.2.1.2 – Taxa de acerto de direção.....	96
[1] ARMANO, G.; MARCHESI, M.; MURRU, A. A hybrid genetic-neural architecture for stock indexes forecasting. Information Sciences. v. 170, p. 3-33, 2005.	100
[2] HAUGEN, R. A. Modern Investment Theory. Upper Saddle River, New Jersey: Prentice Hall, 1997.	100

- [3] JENSEN, M. C. Some anomalous evidence regarding market efficiency. *Journal of Financial Economics*. v. 6, p. 95-101, 1978. 100
- [4] FAMA, E. F. Efficient capital markets; a review of theory and empirical work. *Journal of Finance*. 1970..... 100
- [5] DISORNTETIWAT, P. Global stock index forecasting using multiple generalized regression neural networks with a gating network. PhD Thesis, University of Missouri-Rolla, 2001. 100
- [6] COSTA JR, N. C. A. Sazonalidades do IBOVEPA. *Revista de Administração de Empresas*, 1990. 100
- [7] Desai, H., Jain, P. Long-run common stock returns following splits and reverse splits. *Journal of Business*, 1997..... 100
- [8] DAVIDSSON, Marcus. Stock Market Anomalies - A Literature Review and Estimation of Calendar affects on the S&P 500 index., 2006..... 100
- [9] LEITE, J. C. Splits: ineficiência no mercado de ações. *Revista de Administração de Empresas*, 1994. 100
- [10] NORONHA, Márcio. *Análise Técnica*, 2005..... 100
- [11] DORFFNER, G. *Neural Networks for Time Series Processing*, *Neural Network World*, 1996..... 100
- [12] Box, Jenkins G. E. Box & G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, CA, 1970..... 100
- [13] PRUDÊNCIO, Ricardo Bastos Cavalcante. *Projeto híbrido de redes neurais*. Universidade Federal de Pernambuco, 2002..... 100
- [14] Portal Capes. Site: www.portalcapes.com.br..... 100
- [15] SOUTO-MAIOR, César Duarte. *Previsão da direção de movimento de índices de ações usando um sistema fuzzy*. 2007..... 100
- [16] TEIXEIRA, J. C.; RODRIGUES, A. J. An applied study on recursive estimation methods, neural networks and forecasting. *European Journal of Operational Research*, 1997..... 101
- [17] GENÇAY, R. The predictability of security returns with simple technical trading rules. *Journal of Empirical Finance*, 1998..... 101
- [18] YU, S. Forecasting and arbitrage of the Nikkei stock index futures: an application of backpropagation networks. *Asia-Pacific Financial Markets*, 1999.. . 101
- [19] PANDA, C.; NARASIMHAN, V. Predicting stock returns: an experiment of the artificial neural network in Indian stock market. *South Asia Economic Journal*, 2006..... 101
- [20] GOMES, F.C. Os modelos ARIMA e a abordagem de Box-Jenkins: uma aplicação na previsão do IBOVEPA a curtíssimo prazo. *Revista de Administração de Empresas*, 1989..... 101
- [21] RIBEIRO, T. S.; SILVA, A. L. C. Do artificial neural networks provide better forecasts? Evidence from latin american stock indexes. In: *XXIX Enampad*, 2005, Brasília..... 101

- [22] GRUDNITSKI, G.; OSBURN, L. *Forecasting S&P and gold futures prices: an application of neural networks. Journal of Futures Markets, 1993.101*
- [23] FERNÁNDEZ-RODRÍGUEZ, F.; GONZÁLEZ-MARTEL, C.; SOSVILLA-RIVERO, S. *On the profitability of technical trading rules based on artificial neural networks: evidence from the Madrid stock market. Economics Letters, 2000.....101*
- [24] HAYKIN, Simmon –*Neural Networks –A comprehensive foundation. 2nd Edition, 2005.....101*
- [25] W.S. McCulloch and W. Pitts. *A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 1943.....101*
- [26] D.O. Hebb's,. *The Organization of Behavior. New York, Wiley, 1949.....101*
- [27] ROSENBLATT, F. (1959). *Principles of Neurodynamics. New York: Spartan Books, 1959.....101*
- [28] JONES, Edward R. *An Introduction to Neural Networks - A White Paper. 2004101*
- [29] G. Cybenko. *Approximation by Superpositons of a Sigmoid Function. Mathematics of Control, Signals and Systems. 1989.....101*
- [30] G. Cybenko. *Continuos Valued Neural Network with two Hidden Layers are Sufficient. Technical Report, Departament of Computer Science, Tufts University. 1988.....101*
- [31] HOPFIELD, J. *Neural Networks and Physical Systems with Emergent Collectives Computational Abilities. Proceedings of the National Academy of Sciences , 1982.....101*
- [32] JORDAN, M. I. *Attractor dynamics and parallelism in a connectionist sequential machine. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, 1986.....102*
- [33] JORDAN, M. I. *Serial OPrder: A Parallel Distributed Processing Approach, San Diego, La Jolla, CA: Institue for Cognitive Science, University of Califórnia, 1986.102*
- [34] FAUSETT, L. *Fundamentals of Neural Networks – Architecture, Algorithms, and Applications. [S.I.]: Prentice Hall International, Inc., 1994.....102*
- [35] MELLO, Marília Terra de. *Aplicação de redes neurais no processo de precificação de ações. Universidade Estadual de Pelotas, 2004.....102*
- [36] ZANNETI Jr. Luiz Antonio e ALMEIDA. Fernando C. - *Exploração do uso de redes neurais na previsao do comportamento de ativos financeiros. Faculdade de Economia Administração e Contabilidade – USP. 1998.....102*
- [38] LAWRENCE, Ramon. *Using Neural Networks to Forecast Stock Market Prices. University of Manitoba. 1997.....102*
- [39] <http://finance.yahoo.com>.....102

SIGLAS

AR – Auto Regressive Model

ARIMA – Auto Regressive Integrated Moving Average

ARMA – Auto Regressive Moving Average

BOVESPA – Bolsa de Valores de São Paulo.

Capes – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

CATS – Computer Assisted Trading System (Sistema de Negociação Eletrônica)

COMIT – Índice da Bolsa de Valores de Milão

DJIA – Dow Jones Industrial Average Index

IA – Inteligência Artificial

IBOVESPA – Índice da Bolsa de Valores de São Paulo.

IGBM – Índice General de la Bolsa e Madrid (Índice Geral da Bolsa de Madri)

LAR –Linear Auto Regressive

MA –Moving Average

MCP –McCulloch-Pitts Perceptron (Perceptron de McCulloch-Pitts)

MLP – Multilayer Perceptrons (Perceptrons Multicamadas)

NIKKEI 225 – Principal índice econômico da Bolsa de Valores de Tóquio.

RNA – Redes Naturais Artificiais

RNN – Redes Neurais Naturais

S&P 500 –Standard Poors

Sensex – Bombay Stock Exchange Sensitive Index

SIMEX – Singapore International Monetary Exchange

SPOT – Sistema Privado de Operações por Telefone

UFRJ – Universidade Federal do Rio de Janeiro

GLOSSÁRIO

Commodities – mercadorias (geralmente matérias-primas ou outros produtos com pequeno grau de industrialização) negociadas nas bolsas de valores (e.g. soja, arroz, etc).

Índice Dow Jones ou Dow-Jones Industrial Average (DJIA) – um dos principais indicadores do movimento de ações no mercado americano.

Home Broker – instrumento que permite a negociação de ações (compra e venda) via Internet.

After-Market – pregão eletrônico realizado após o fechamento do pregão normal.

Random Walk – Segundo alguns pesquisadores, a propriedade de ajustamento instantâneo das informações públicas aos preços das ações em um mercado eficiente implica na independência entre variações de preços sucessivos na seqüência das negociações. Um mercado que apresenta tal comportamento é, por definição, um mercado *random walk*. De acordo com o modelo random walk, uma série de mudanças de preços não tem memória e, portanto, a história passada da série não pode ser usada para prever o futuro de uma maneira significativa.

LISTA DE FIGURAS

<i>Figura 2.1 – Retorno médio por meses do ano.....</i>	<i>25</i>
<i>Fonte: http://www.finint.ase.ro/Materiale/Manuale/Investment%20Valuation_Damodaran/ch6.pdf, cap.6, pag. 45.....</i>	<i>25</i>
<i>Figura 2.2 – Tendências primária no IBOVESPA.....</i>	<i>30</i>
<i>Fonte: www.neologica.com.br, acessado em 19/10/2008.....</i>	<i>30</i>
<i>Figura 2.3 – Tipos de artigos encontrados no Portal Capes.....</i>	<i>38</i>
<i>Fonte: Souto-Maior [15]. Adaptado.....</i>	<i>38</i>
<i>Figura 2.4 – Horizontes de previsão.....</i>	<i>38</i>
<i>Fonte: Souto-Maior [15]. Adaptado.....</i>	<i>38</i>
<i>Figura 3.1 – Diagrama de blocos da representação do sistema nervoso.....</i>	<i>47</i>
<i>Fonte: Haykin[24], pág 28. Adaptado.....</i>	<i>47</i>
<i>Figura 3.2 – Componentes de um Neurônio Biológico.....</i>	<i>48</i>
<i>Fonte: Haykin [24]. pág. 30. Adaptado.....</i>	<i>48</i>
<i>Figura 3.3 – Neurônio de McCulloch-Pitts.....</i>	<i>50</i>
<i>Figura 3.4 – Algumas funções de ativação.....</i>	<i>51</i>
<i>Figura 3.5 – Aprendizado Supervisionado.....</i>	<i>54</i>
<i>Figura 3.6 – Aprendizado Não Supervisionado.....</i>	<i>55</i>
<i>.....</i>	<i>55</i>
<i>Figura 3.7 – Diagrama de blocos de Aprendizado por Reforço.....</i>	<i>55</i>
<i>Fonte: Haykin[24]. Adaptado.....</i>	<i>55</i>
<i>Figura 3.9 – Rede MLP com camada intermediária.....</i>	<i>62</i>
<i>Fonte: Haykin [24], Extraído.....</i>	<i>62</i>
<i>Figura 3.10 – Treinamento do algoritmo back-propagation.....</i>	<i>65</i>
<i>Figura 3.11: Exemplo de rede recorrente sem loop e sem camada escondida.....</i>	<i>66</i>
<i>Fonte: Haykin [24], Extraído.....</i>	<i>67</i>
<i>Figura 3.12 – Exemplo de Rede Jordan.....</i>	<i>67</i>
<i>Fonte: Krose, Smargt [34], Extraído.....</i>	<i>67</i>
<i>Figura 3.13 – Rede Elman.....</i>	<i>68</i>
<i>Fonte: Krose, Smargt [34], Extraído.....</i>	<i>68</i>
<i>.....</i>	<i>69</i>
<i>Figura 3.14 – Rede TLFN.....</i>	<i>69</i>

<i>Fonte: Haykin [24], Extraído.....</i>	<i>69</i>
<i>Figura 4.1 – Parte da planilha gerada pelo script e utilizada na entrada da RN.....</i>	<i>71</i>
<i>Figura 4.2 – Gráfico das variáveis de entrada da RN.....</i>	<i>72</i>
<i>Figura 4.3 – Gráfico da variável Fechamento.....</i>	<i>73</i>
<i>.....</i>	<i>75</i>
<i>Figura 4.4 – Gráfico da função tan-sigmoid.....</i>	<i>75</i>
<i>Fonte: Matlab, documentação.....</i>	<i>75</i>
<i>.....</i>	<i>76</i>
<i>Figura 4.5 – Gráfico da função purelin.....</i>	<i>76</i>
<i>Fonte: Matlab, documentação.....</i>	<i>76</i>
<i>Tabela 5.1 – Parâmetros de configuração do 1o conjunto de redes desenvolvido.....</i>	<i>85</i>
<i>Tabela 5.2 – Características da melhor rede.....</i>	<i>85</i>
<i>Figura 5.1 – Gráfico de saída da rede cujo resultado não foi satisfatório.....</i>	<i>90</i>
<i>Tabela 5.3 – Parâmetros de configuração do 2º conjunto de RNs.....</i>	<i>91</i>
<i>Tabela 5.4 – Características da melhor rede.....</i>	<i>92</i>
<i>Figura 5.2 – Gráfico de saída da melhor rede.....</i>	<i>93</i>
<i>Figura 5.3 – Gráfico de saída da melhor rede com range de 150 dias.....</i>	<i>94</i>
<i>Figura 5.4 – Gráfico do percentual de acurácia da saída do sistema.....</i>	<i>96</i>
<i>Figura 5.5 – Gráfico do percentual de acertos/erros da tendência.....</i>	<i>97</i>

LISTA DE TABELAS

<i>Figura 2.1 – Retorno médio por meses do ano.....</i>	<i>25</i>
<i>Fonte: http://www.finint.ase.ro/Materiale/Manuale/Investment%20Valuation_Damodaran/ch6.pdf, cap.6, pag. 45.....</i>	<i>25</i>
<i>Figura 2.2 – Tendências primária no IBOVESPA.....</i>	<i>30</i>
<i>Fonte: www.neologica.com.br, acessado em 19/10/2008.....</i>	<i>30</i>
<i>Figura 2.3 – Tipos de artigos encontrados no Portal Capes.....</i>	<i>38</i>
<i>Fonte: Souto-Maior [15]. Adaptado.....</i>	<i>38</i>
<i>Figura 2.4 – Horizontes de previsão.....</i>	<i>38</i>

<i>Fonte: Souto-Maior [15]. Adaptado.....</i>	<i>38</i>
<i>Figura 3.1 – Diagrama de blocos da representação do sistema nervoso.....</i>	<i>47</i>
<i>Fonte: Haykin[24], pág 28. Adaptado.....</i>	<i>47</i>
<i>Figura 3.2 – Componentes de um Neurônio Biológico.....</i>	<i>48</i>
<i>Fonte: Haykin [24]. pág. 30. Adaptado.....</i>	<i>48</i>
<i>Figura 3.3 – Neurônio de McCulloch-Pitts.....</i>	<i>50</i>
<i>Figura 3.4 – Algumas funções de ativação.....</i>	<i>51</i>
<i>Figura 3.5 – Aprendizado Supervisionado.....</i>	<i>54</i>
<i>Figura 3.6 – Aprendizado Não Supervisionado.....</i>	<i>55</i>
<i>.....</i>	<i>55</i>
<i>Figura 3.7 – Diagrama de blocos de Aprendizado por Reforço.....</i>	<i>55</i>
<i>Fonte: Haykin[24]. Adaptado.....</i>	<i>55</i>
<i>Figura 3.9 – Rede MLP com camada intermediária.....</i>	<i>62</i>
<i>Fonte: Haykin [24], Extraído.....</i>	<i>62</i>
<i>Figura 3.10 – Treinamento do algoritmo back-propagation.....</i>	<i>65</i>
<i>Figura 3.11: Exemplo de rede recorrente sem loop e sem camada escondida.....</i>	<i>66</i>
<i>Fonte: Haykin [24], Extraído.....</i>	<i>67</i>
<i>Figura 3.12 – Exemplo de Rede Jordan.....</i>	<i>67</i>
<i>Fonte: Krose, Smargt [34], Extraído.....</i>	<i>67</i>
<i>Figura 3.13 – Rede Elman.....</i>	<i>68</i>
<i>Fonte: Krose, Smargt [34], Extraído.....</i>	<i>68</i>
<i>.....</i>	<i>69</i>
<i>Figura 3.14 – Rede TLFN.....</i>	<i>69</i>
<i>Fonte: Haykin [24], Extraído.....</i>	<i>69</i>
<i>Figura 4.1 – Parte da planilha gerada pelo script e utilizada na entrada da RN.....</i>	<i>71</i>
<i>Figura 4.2 – Gráfico das variáveis de entrada da RN.....</i>	<i>72</i>
<i>Figura 4.3 – Gráfico da variável Fechamento.....</i>	<i>73</i>
<i>.....</i>	<i>75</i>
<i>Figura 4.4 – Gráfico da função tan-sigmoid.....</i>	<i>75</i>
<i>Fonte: Matlab, documentação.....</i>	<i>75</i>
<i>.....</i>	<i>76</i>
<i>Figura 4.5 – Gráfico da função purelin.....</i>	<i>76</i>
<i>Fonte: Matlab, documentação.....</i>	<i>76</i>
<i>Tabela 5.1 – Parâmetros de configuração do 1o conjunto de redes desenvolvido.....</i>	<i>85</i>
<i>Tabela 5.2 – Características da melhor rede.....</i>	<i>85</i>

<i><u>Figura 5.1 – Gráfico de saída da rede cujo resultado não foi satisfatório.....</u></i>	<i><u>90</u></i>
<i><u>Tabela 5.3 – Parâmetros de configuração do 2º conjunto de RNs.....</u></i>	<i><u>91</u></i>
<i><u>Tabela 5.4 – Características da melhor rede.....</u></i>	<i><u>92</u></i>
<i><u>Figura 5.2 – Gráfico de saída da melhor rede.....</u></i>	<i><u>93</u></i>
<i><u>Figura 5.3 – Gráfico de saída da melhor rede com range de 150 dias.....</u></i>	<i><u>94</u></i>
<i><u>Figura 5.4 – Gráfico do percentual de acurácia da saída do sistema.....</u></i>	<i><u>96</u></i>
<i><u>Figura 5.5 – Gráfico do percentual de acertos/erros da tendência.....</u></i>	<i><u>97</u></i>

1 – INTRODUÇÃO

1.1 – TEMA

Este trabalho corresponde a um estudo da empregabilidade da previsão de índices de mercado de capitais, em especial o IBOVESPA, utilizando uma das principais áreas da Inteligência Artificial: as Redes Neurais Artificiais.

1.2 – DELIMITAÇÃO

Este projeto destina-se a todos, estudantes ou não, que queiram familiarizar-se com os alguns conceitos, aplicações e desdobramentos das Redes Neurais Artificiais. Apesar de trabalhar dados históricos da BOVESPA, o método desenvolvido neste estudo pode ser aplicado para outras séries temporais, com algumas adaptações a depender do tipo de aplicação.

1.3 – JUSTIFICATIVA

Há várias motivações para se tentar prever o comportamento de ações do mercado financeiro. A principal delas é, obviamente, o lucro ao se negociar tais ações. Qualquer sistema que possa, de maneira consistente, definir vencedores e perdedores na dinâmica deste mercado, fará, certamente, com que seu desenvolvedor ou proprietário tenha uma vantagem muito grande sobre os outros jogadores. Pensando nisso, muitos investidores e pesquisadores, em especial engenheiros e economistas, estão continuamente buscando por melhores sistemas que possam ajudá-los a receber melhores retornos.

Outra motivação reside no fato de que cada vez mais engenheiros, das mais variadas habilitações, estão trabalhando no mercado financeiro. Sejam em bancos,

consultoras ou mesmo por conta própria, estes profissionais aproveitam o grande conhecimento matemático aprendido durante a faculdade e o avanço de ferramentas computacionais para se desenvolverem e progredirem em suas carreiras neste ramo. Com isso, torna-se cada vez mais necessária a introdução de assuntos ligados a conceitos e estratégias de investimentos no mercado de renda variável durante a vida acadêmica destes engenheiros.

Mais uma motivação está na Hipótese do Mercado Eficiente. Segundo esta teoria, ‘nenhum sistema poderia continuamente ser superior ao mercado, por que, se tal sistema existisse, de alguma forma ele se tornaria público e qualquer individuo poderia utilizá-lo, fato que diminuiria seu ganho potencial’ [37]. Esta teoria tem provocado diversos debates a respeito de sua validade e alguns pesquisadores têm utilizado as Redes Neurais Artificiais para comprovar suas alegações.

1.4 – OBJETIVOS

A partir do problema e do tema já mencionados, foram determinados objetivos a serem atingidos nesta pesquisa. Tais objetivos estabeleceram a direção a ser seguida por este estudo. Desta forma, tem-se primeiramente o objetivo geral, que norteou a pesquisa, e em seguida os objetivos específicos, que operacionalizaram os propósitos deste estudo.

Constituiu-se como objetivo geral deste trabalho verificar a empregabilidade da previsão do índice BOVESPA usando Redes Neurais Artificiais. Para verificar a possibilidade de previsão do índice, foi definido um período de estudo: de 27/06/1997 a 09/05/2008;

Com isso, fizeram-se necessários os seguintes objetivos específicos:

- desenvolvimento de um script para Matlab, com o objetivo de obter os dados da série histórica do índice com mais facilidade;
- desenvolvimento de um script para Matlab para implantação da Rede Neural, incluindo seu treinamento, teste e validação;

- verificação da possibilidade de previsão estatisticamente significativa do IBOVESPA no período de estudo;

1.5 – METODOLOGIA

Neste estudo, as redes neurais são empregadas na previsão de séries temporais. Para isso, foi recolhida, com a ajuda de um script para Matlab, a série histórica de dados correspondentes ao período de estudo. Estes dados foram usados para treinar, testar e validar a rede neural. Por fim, foi feita uma análise dos resultados da previsão baseada na taxa de acertos de direção e no percentual de acertos da saída da rede em relação ao valor original.

1.6 – DESCRIÇÃO

No Capítulo 2 serão mostrados alguns conceitos sobre o mercado financeiro, tais como eficiência de mercado, análise técnica e análise fundamentalista.

O Capítulo 3 apresenta os conceitos de Redes Neurais, desde o funcionamento do cérebro humano, passando por alguns tipos de aprendizado, até algumas arquiteturas de redes.

A metodologia empregada no trabalho é apresentada no Capítulo 4. Nele são descritos os tipos de dados utilizados, o modelo de rede empregado e as métricas para análise dos resultados.

O Capítulo 5 apresenta os resultados finais do estudo e, por fim, o Capítulo 6 conclui a pesquisa e apresenta proposta para trabalho futuros.

2 – FUNDAMENTOS DO MERCADO FINANCEIRO

2.1 – EFICIÊNCIA DE MERCADO

De acordo com a Hipótese de Mercado Eficiente, todos os mercados são eficientes em descobrir oportunidades de lucro tão rapidamente, que elas, na verdade, deixam de ser oportunidades. Um mercado é dito eficiente se os preços deste mercado refletem todas as informações disponíveis.

A Hipótese de Mercado Eficiente - tal qual a teoria do Random Walk – é, segundo Haugen [2], na verdade, um modelo de como o mercado se comporta. Como qualquer modelo, ele se aplica a determinados casos e a outros não.

Suponha que alguém tenha ouvido que determinada empresa obteve lucro acima do esperado por seus analistas. Neste momento, seria normal de se esperar que tal indivíduo se apressasse em comprar ações desta empresa. Porém, tal cidadão se surpreenderia ao saber que o preço da ação por ele desejada já teria sofrido uma valorização substancial desde o fechamento do dia anterior. De acordo com a Hipótese de Mercado Eficiente, qualquer informação que seja suficientemente importante na tomada de decisões já foi incorporada ao mercado nos preços e é inútil tentar prever o comportamento futuro das ações.

De acordo com Armano, Marchesi e Murru [1], o preço de uma ação reflete diretamente, em qualquer tempo, as informações que os investidores possuem. Ou seja, assim que uma informação nova é processada, o preço da ação é alterado.

Segundo Haugen [2], não é necessário que haja um consenso entre todos os investidores para a definição de preços, e sim pelos investidores que negociam mais ativamente no mercado.

Jensen [3] definiu como *mercado eficiente* aquele que, baseado em todas as informações disponíveis, fosse incapaz de ser ‘derrotado’, ou seja, fosse impossível de se obter rendimentos superiores aos rendimentos médios do mercado.

2.1.1 – FORMAS DE EFICIÊNCIA DE MERCADO

Entre 1965 e 1970 muitos testes empíricos foram realizados sobre o comportamento dos preços das ações. Um dos trabalhos mais importantes desta época foi o de Fama [4], intitulado "*Efficient capital markets: A review of theory and empirical work*". Nele, Fama definiu três formas de eficiência de mercado:

- *Eficiência de mercado da forma fraca*: os preços das ações refletem todas e quaisquer informações contidas nos históricos dos preços e volumes já negociados. É essencialmente a hipótese de Random Walk, porém sem uma total caracterização do processo estocástico que descreve o comportamento dos preços e volumes. Nesta forma de eficiência, torna-se impossível ‘vencer o mercado’ com base no histórico das ações, pois os preços já teriam sido ajustados por analistas que tentaram prever tal comportamento;
- *Eficiência de mercado da forma semiforte*: os preços das ações refletem todas as informações disponíveis aos investidores – histórico de preços, notícias econômicas, relatórios contábeis, dados de ganhos, etc. Desta forma, não é possível ‘vencer o mercado’ com base na análise de tais informações públicas, pois outros analistas já fizeram isto e os preços já foram reajustados;
- *Eficiência de mercado da forma forte*: os preços refletem todas as informações públicas e sigilosas. Informações sigilosas são aquelas que ainda não foram publicadas, porém possuem a capacidade de influenciar o mercado. Desta forma, não é possível ‘vencer o mercado’ com base na análise de informações, sejam elas públicas ou sigilosas, pois outros analistas que também possuíam estas informações já as utilizaram e os preços já foram reajustados.

É importante salientar que em um mercado eficiente os preços podem sofrer desvios dos seus valores reais. Segundo esta teoria, tais desvios seriam aleatórios. Também de acordo com esta teoria, o retorno esperado de cada investimento será, de alguma forma, proporcional ao risco que este oferece a longo prazo. Entretanto, estes mesmos retornos podem sofrer desvios a curto prazo.

2.1.2 – TESTES DE EFICIÊNCIA DE MERCADO

Segundo Disornetiwat [4], a teoria de mercado eficiente implica diretamente no conhecimento do modelo *Random Walk*. De acordo com este modelo, os preços das ações não se comportam de maneira previsível e sim de maneira aleatória. Por isso, a melhor expectativa de preços para o período seguinte é o próprio período atual.

Testes de eficiência de mercado são empregados de forma a examinar quais estratégias de investimentos produzem maiores retornos. Alguns testes também avaliam os custos de transações e de prováveis execuções. Sabendo que o *retorno esperado* de um investimento é a diferença entre o *valor atual* e o *valor esperado*, há implicitamente em todo teste de mercado eficiente um modelo para o retorno esperado. Em alguns casos, este retorno advém do risco do investimento, em outros é baseado no retorno de investimentos similares ou equivalentes. Nos dois casos, um teste de eficiência de mercado é a junção deste com o teste de eficiência do modelo usado para calcular o retorno esperado.

Desde a década de 1970, vários estudos têm sido feitos para tentar refutar o modelo de Random Walk. Neste contexto, os testes de Fama serviram apenas para reforçar a idéia de mercado eficiente.

Outros pesquisadores apresentaram evidências do contrário. Um ramo de pesquisas denominado Finanças Comportamentais tem tentado mostrar que, com base em observações comportamentais do mercado, os preços não variam de forma tão aleatória quanto a Teoria do Mercado Eficiente defende, e que eles seguem o princípio de retorno à média, além de serem fortemente influenciadas pela subjetividade do comportamento humano.

De acordo com Costa Jr. [6], devido ao recente desenvolvimento dos computadores e banco de dados mais elaborados, vários estudos começaram a mostrar certas anomalias a respeito do comportamento dos retornos de ativos financeiros, antes não notadas. Estas anomalias, segundo tais estudos, vão de encontro às hipóteses de eficiência de mercado.

2.1.2.1 – Casos especiais

Estudos de eficiência de mercado têm descoberto inúmeros exemplos de comportamentos de mercado que são inconsistentes com os modelos de risco e retorno. A persistência de algumas dessas anomalias sugere que o problema, ou pelo menos parte dele, está muito mais nos modelos de risco e retorno do que no próprio comportamento do mercado financeiro. As seções seguintes mostram algumas dessas anomalias encontradas em mercados financeiros pelo mundo.

2.1.2.1.1. – Efeito fim-de-semana

O *efeito fim-de-semana* é um fenômeno encontrado em inúmeros mercados internacionais. Ele consiste nas diferenças de retorno entre segundas-feiras e os outros dias da semana. Muitas pesquisas mostram que os retornos desses dias são significativamente menores que os do resto da semana.

Este fenômeno pode ser explicado, em parte, pela desigual divulgação de notícias durante a semana. Evidências mostram que há uma tendência de se divulgar boas notícias uniformemente durante toda a semana e guardar as más para o final de semana. Outra explicação está nos investidores com pouca experiência. Estes, durante o fim-de-semana, fariam análises errôneas e, na segunda-feira, investimentos equivocados.

2.1.2.1.2 – Efeito Janeiro

Estudos têm revelado que, nos Estados Unidos e em outros grandes centros financeiros do mundo, existem grandes diferenças com relação ao comportamento dos retornos durante os meses do ano.

De acordo com tais estudos, retornos em Janeiro são significativamente maiores que em outros meses do ano. Este fenômeno, chamado de *Efeito Janeiro* ou *Efeito Fim de ano*, pode ser melhor visto nas duas primeiras semanas do mês de Janeiro.

A figura 2.1 mostra o retorno médio por mês durante os anos de 1926 e 1983.

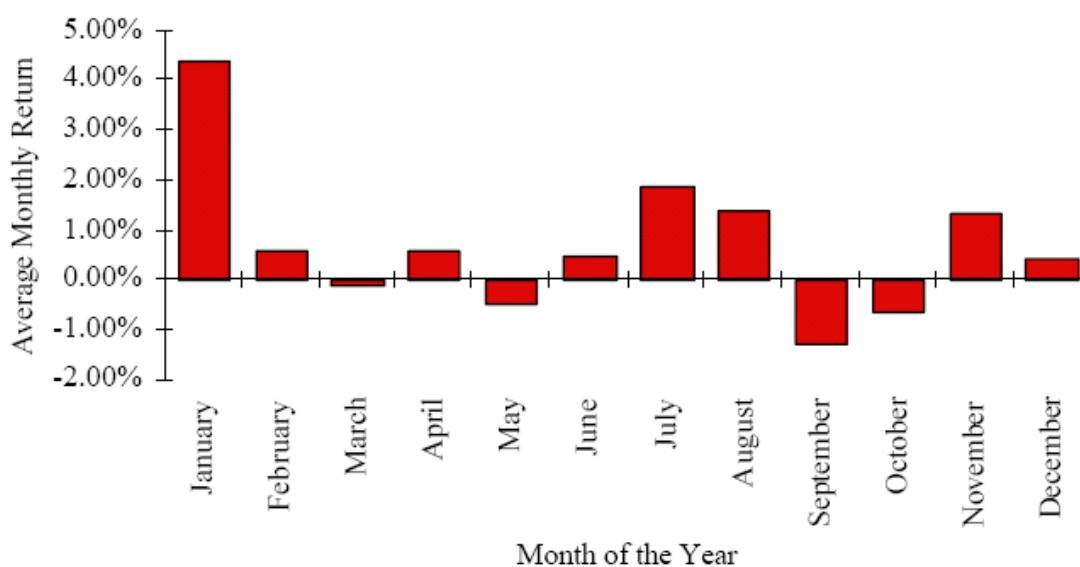


Figura 2.1 – Retorno médio por meses do ano.

Fonte: http://www.finint.ase.ro/Materiale/Manuale/Investment%20Valuation_Damodaran/ch6.pdf, cap.6, pag. 45.

Uma das explicações para este fenômeno está na tentativa de pagar menos impostos pelos investidores. Estes venderiam ativos em Dezembro, a preços menores que seus valores reais, e os comprariam de volta em Janeiro, resultando em retornos maiores. Porém, como esta estratégia depende do sistema fiscal de cada país, há registros de países em que tal fenômeno não ocorre.

2.1.2.1.3 – Splits

Splits são desdobramentos de capital de empresas. Por exemplo, para cada ação de uma determinada empresa o investidor interessado passará a possuir duas ações com a metade do preço original.

Segundo Desai & Jain [7] (*apud* Davidsson [8]) splits tendem a aumentar o preço de uma ação, antes e depois de seu anúncio.

Segundo Haugen [2], ao ocorrer um split, existe, aproximadamente, 80% de chances de, no futuro, ocorrer um aumento no valor dos dividendos.

Também é possível explicar este fenômeno a partir da diminuição do valor da ação. Com isso, haveria um aumento de liquidez devido à presença de pequenos investidores que teriam acesso ao novo valor.

No Brasil, Leite [9] analisou splits de todas as ações listadas na BOVESPA e constatou fortes evidências que contradizem a hipótese de mercado eficiente na forma semiforte.

2.2 – MÉTODOS ANALÍTICOS NA PREVISÃO DO RETORNO DE AÇÕES

Antes da era dos computadores, negociar ações e commodities era uma tarefa muito mais intuitiva. À medida que o nível de investimentos e as negociações cresceram, a necessidade de ferramentas e métodos que aumentassem os ganhos e minimizassem os riscos também cresceu. Estatísticas, análises técnicas, análises fundamentalistas e regressão linear são algumas dessas ferramentas usadas para tentar prever a direção do mercado.

Esses métodos são aqui apresentados como são normalmente utilizados na prática. Com eles é possível ter uma primeira idéia de como as redes neurais devem funcionar.

2.2.1 – ANÁLISE TÉCNICA

Segundo a Análise Técnica, os preços se movem em tendências ditadas pela constante mudança de atitudes de investidores em resposta a diferentes forças. Usando o histórico dos movimentos de preços e volumes e outras estatísticas, e ignorando fatores como ambiente econômico, eventos políticos, política fiscal do governo e tendências industriais, o analista técnico utiliza gráficos para prever o comportamento futuro das ações.

Este tipo de análise ainda é controversa entre estudiosos do assunto. Isto se deve ao fato de ser, segundo alguns, extremamente subjetiva e, também, por contradizer a Hipótese de Eficiência de Mercado. Apesar disso, é utilizada por aproximadamente 90% dos investidores [10].

Os investidores que utilizam esta técnica fazem uso de gráficos para detectar possíveis tendências. Tais tendências, acredita-se, são baseadas na oferta e demanda e geralmente apresentam um padrão cíclico. Há inúmeros indicadores técnicos derivados deste tipo de análise a partir dos quais podem ser retiradas regras de aplicações ou entradas de redes neurais, dentre os quais: indicadores de filtro, indicadores de momento, análise de curva de tendência, indicadores de volume e análise de padrão. Tais indicadores podem fornecer informações de curto ou longo prazo, ajudando a identificar tendências ou ciclos no mercado.

2.2.1.1 – Indicadores

A seguir estão alguns dos indicadores mais utilizados na Análise Técnica:

2.2.1.1.1 – Médias Móveis

Consistem em médias extraídas de um corpo de dados sequenciais numa janela de tempo, com a finalidade de informar sobre um início ou fim de uma tendência de preços. Desta forma, uma média móvel de 10 períodos mostra o preço médio do fechamento dos últimos 10 períodos. No décimo primeiro dia, substitui-se o preço de fechamento do primeiro dia pelo preço de fechamento do décimo primeiro e calcula-se uma nova média, e assim sucessivamente. Ao conectar os pontos das médias móveis de cada dia, cria-se a linha da média móvel. O primeiro ponto de uma média móvel surge de acordo com a periodicidade em que está sendo calculada. Numa média móvel de 10 dias, por exemplo, o primeiro ponto aparece no décimo dia.

Há três tipos de média móvel:

- Médias Móveis Simples: considerando N períodos, é a média aritmética dos últimos N preços de fechamento;
- Médias Móveis Ponderadas: dão pesos diferentes aos preços de fechamento que as formam. O preço do fechamento mais próximo recebe um peso maior, que vai decrescendo nos fechamentos mais antigos, até o último, que recebe o menor peso;
- Médias Móveis Exponenciais: é dada pela equação

$$MME_{HOJE} = \frac{2 \times PF_{HOJE} + (N - 1) \times MME_{ONTEM}}{N + 1}$$

onde, MME_{HOJE} é a média móvel exponencial de hoje, PF_{HOJE} é o preço de fechamento de hoje, N é o número de dias da média móvel exponencial (escolhido pelo analista) e MME_{ONTEM} é a média móvel exponencial de ontem.

2.2.1.1.2 – Volume

É uma medida que expressa o valor financeiro negociado num dia de pregão. Pode ser global, referente ao mercado como um todo, ou individual, referente a apenas um ativo, ou expressar o total de títulos negociados.

2.2.1.1.3 – Tendências

A direção dos preços, quando vista em um gráfico, tem notadamente um padrão de ziguezague. Esta direção pode ser ascendente, indefinida ou descendente. À permanência de um preço numa determinada direção, durante um período de tempo, dá-se o nome de tendência.

2.2.1.1.4 – Momento

Mede a diferença entre os preços de fechamento em um determinado intervalo de tempo, e representa a velocidade da evolução dos mesmos em tendências bem definidas.

O momento de “x” dias, dado por M_x , é calculado pela fórmula

$$M_x = PF_{HOJE} - PF_{(HOJE - X)}$$

O gráfico do momento é plotado a partir de uma linha de referência.

2.2.1.2 – Teoria de Dow

Charles Dow foi um dos grandes responsáveis pelo desenvolvimento da análise técnica. Ele entendia que a criação de índices ajudaria os investidores a identificar e confirmar tendências do mercado. Com base nisso, propôs a criação de dois índices: Dow-Jones Ferroviário, que reunia vinte grandes empresas ferroviárias, e o Dow-Jones Industrial, índice que existe até os dias atuais, reunindo doze empresas industriais, número que aumentou para vinte em 1916 e trinta em 1928.

Os estudiosos dos escritos de Charles Dow estabeleceram seis princípios que, reunidos, formam o que hoje conhecemos como a Teoria de Dow. São eles:

- **Princípio 1: Os índices descontam tudo:** Os índices representam a ação conjunta de inúmeros investidores, desde os mais bem informados (que contam com as melhores informações e previsões) até os muito inexperientes. As variações diárias dos preços de um índice, portanto, já têm incluídas (descontadas) no seu valor os eventos que irão acontecer e que são desconhecidos pela maioria dos investidores. Dessa forma, todo fator que afeta a relação de oferta/demanda está refletido no preço do mercado;
- **Princípio 2: As Três Tendências do Mercado:** este princípio afirma que o mercado possui três tendências de movimento: primária, secundária e terciária. O “Mercado”, ou seja, o preço das ações em geral, move-se em tendências das quais as mais importantes são as Primárias. Tendências primárias costumam durar um ano ou mais, e geram grandes valorizações ou desvalorizações de preços. Ao longo de seu percurso, ocorrem as chamadas tendências secundárias, que costumam durar de três semanas a alguns meses, corrigindo a tendência primária precedente em um a dois terços. Dentro das tendências secundárias, há ainda as tendências terciárias, flutuações que costumam durar de seis dias a três

semanas, corrigindo as tendências secundárias também em um a dois terços.

Na figura 2.2, as linhas verticais estão fazendo uma separação entre três tendências primárias no índice Bovespa.



Figura 2.2 – Tendências primária no IBOVESPA

Fonte: www.neologica.com.br, acessado em 19/10/2008

- **Princípio N° 3: As tendências se dividem em três fases:** Dow fez uma série de observações sobre os movimentos de preços, tanto de alta como de baixa, caracterizando aspectos psicológicos marcantes de cada fase:

➤ **Fases do Mercado de Alta**

Fase 1: No início da alta, o mercado começa a ser propulsionado por investidores mais qualificados, que percebem o bom momento. Neste momento, como a grande maioria dos indivíduos não tem opinião formada sobre o momento do papel, isto permite aos investidores de elite comprar papéis muito baratos.

Fase 2: A segunda parte é uma aceleração mais acentuada do movimento. A pressão compradora aumenta bastante.

Fase 3: A terceira fase é marcada por grandes altas. Os participantes do mercado, de maneira geral, estão cada vez mais seguros de seus

lucros e os investidores mais bem preparados começam a vender suas posições. A grande massa de investidores está em clima de euforia que se realimenta diariamente nos noticiários. Está aberta a possibilidade para a fase 1 do mercado de baixa.

➤ **Fases do Mercado de Baixa**

Fase 1: Nesta fase os profissionais e investidores de elite vendem seus ativos, iniciando a retração;

Fase 2: É uma etapa marcada por um grande nervosismo, os investidores percebem o equívoco e tentam se desfazer de suas posições.;

Fase 3: Com as grandes perdas e ativos muito desvalorizados, a pressão vendedora se dissipa e oportunidades para uma nova alta começam a surgir.

- **Princípio 4: O Princípio da Confirmação:** este princípio afirma que para haver uma reversão de tendência, este fato deve ocorrer em dois índices de composições distintas. Assim, um índice confirma o outro, demonstrando que não se trata de uma oscilação temporária do movimento. Divergência entre eles mostra um sinal de mudança iminente na tendência então vigente.

- **Princípio 5: Volume Deve Confirmar a Tendência:** de acordo com a teoria de Dow, o volume está relacionado com as tendências da seguinte maneira:
 - **Tendência de Alta:** Em uma tendência principal de alta é esperado que o volume aumente com a valorização dos ativos e diminua nas reações de desvalorização;
 - **Tendência de Baixa:** Em uma tendência principal de baixa é esperado que o volume aumente com a desvalorização dos ativos e diminua nas reações de valorização.

- **Princípio 6: A tendência continua até surgir um sinal definitivo de que houve reversão:** segundo este princípio, o mercado não vai cair apenas porque atingiu um nível "alto demais" ou subir porque "já caiu demais". Segundo a Teoria de Dow, para trocar de posição em relação a uma tendência, é preciso ter provas definitivas de que ela acabou.

2.2.2 – ANÁLISE FUNDAMENTALISTA

A análise fundamentalista tem por objetivo, tal qual a análise técnica, avaliar alternativas de investimento. Isto é feito a partir de um profundo estudo sobre a empresa, no qual se avalia seu panorama setorial, suas conjunturas macro e microeconômica e outros aspectos mais "tangíveis", como as fontes de recursos existentes, demonstrações financeiras, capital humano da organização, balanços, projeções de resultados, conhecimentos tecnológicos, entre outros.

Cabe à análise fundamentalista estabelecer o valor justo para uma ação, respaldando decisões de compra ou venda. Neste tipo de análise, acredita-se ser possível estabelecer o valor de uma empresa pela definição da sua capacidade de gerar lucros no futuro.

As vantagens da análise técnica estão em sua abordagem sistemática e habilidade de prever mudanças antes que elas apareçam efetivamente nos gráficos. Companhias são comparadas entre si e seus panoramas de crescimento são relacionados de acordo com o ambiente econômico, permitindo ao investidor se tornar mais familiar da companhia.

Infelizmente, este tipo de abordagem é mais difícil de ser implantada para propósitos de automação (utilizando uma rede neural, por exemplo). Existem ainda outros problemas ligados à subjetividade das interpretações das informações coletadas, a possível presença de fatores desconhecidos dos analistas ou mesmo uma demora até que o resto do mercado interprete as informações disponíveis de uma mesma maneira.

Basicamente, a análise fundamentalista assume que 90% dos investidores utilizam a lógica para realizar seus investimentos, enquanto que, para a análise técnica,

90% utilizam fatores psicológicos e reagem às mudanças do mercado de uma maneira previsível.

2.2.3 – SÉRIES TEMPORAIS

Uma série temporal é uma seqüência de vetores $x(t)$, $t = 0,1,\dots$, onde t representa o tempo. Também pode ser definida como um conjunto de observações de um fenômeno ordenadas no tempo. Pode ser contínua, quando medida sem interrupções no tempo, ou discreta, quando medida em intervalos sucessivos no tempo, em geral, em intervalos regulares.

Analisar uma série temporal é identificar as características e propriedades importantes utilizadas para descrever o fenômeno gerador da série. A análise de séries temporais é um processo de indução, uma vez que, a partir de um conjunto de observações, infere características gerais do fenômeno observado.

Segundo Dorffner [11] (apud Prudêncio [12]) a maior aplicação da análise de séries temporais é a geração de modelos de previsão. Os modelos de previsão de séries temporais são capazes de definir, com certo grau de confiabilidade, os valores futuros de uma série a partir de informações passadas da própria série e de outras variáveis significantes no problema. O uso de modelos de previsão é fundamental para diminuir os riscos na tomada de decisões, uma vez que a eficácia de uma decisão depende obviamente dos eventos que se seguem à decisão. Dentre as muitas aplicações de previsão de séries temporais, podemos citar: planejamento de produção, aplicações financeiras, controle de processos e aplicações médicas.

Existem dois enfoques principais para a modelagem e previsão de séries temporais. O primeiro é o uso de uma teoria sobre o domínio de aplicação (econômica, hidrológica, física, etc...) para a construção de um modelo conceitual para o fenômeno gerador da série. Porém, os modelos teóricos têm a desvantagem de necessitar de um conhecimento aprofundado do domínio para a sua construção, além de ter aplicação limitada ao domínio abordado.

O outro enfoque consiste em construir um modelo de previsão a partir dos dados disponíveis, sem recorrer a uma teoria específica. Nessa abordagem, uma série é

modelada através de uma função com parâmetros livres ajustados a partir dos dados da série. Essa abordagem é conhecida como modelos caixa-preta, uma vez que os parâmetros dos modelos gerados não têm uma interpretação direta dentro do domínio do problema.

Dois dos modelos caixa-preta mais difundidos são de Box-Jenkins [12], que modelam as séries através de funções lineares, e o modelo utilizando redes neurais, que utilizam funções não-lineares.

2.2.3.1 – Modelos de Box-Jenkins

Apesar da simplicidade desses modelos, eles têm sido aplicados satisfatoriamente em uma diversidade de problemas reais e seus princípios servem até hoje como base para outros modelos. Segundo Prudêncio [13], a classe de Box e Jenkins contém dois modelos básicos, o modelo *auto-regressivo* e o modelo de *médias-móveis*. Esses modelos podem ser combinados ou usados individualmente.

Em um modelo auto-regressivo puro, a previsão de um valor da série é feita através de uma combinação linear dos valores passados da série. O modelo *AR* (p) (*Autoregressive model*) pode ser descrito pela equação 2.1.

$$Z(t) = \phi_0 + \phi_1 Z(t-1) + \dots + \phi_p Z(t-p) + \varepsilon(t) \quad (2.1)$$

Nesse modelo o valor da série Z no tempo t é previsto usando os últimos p valores da série. O parâmetro p é chamado de ordem do modelo. Os valores dos parâmetros ϕ_i são ajustados conforme a série que se deseja modelar, visando diminuir o erro de previsão. A variável ε representa o erro de previsão obtido pelo modelo.

Um modelo de médias-móveis puro de ordem q , *MA*(q) (*Moving Average*), representa uma série a partir de valores passados dos erros de previsão, podendo ser descrito pela equação 2.2.

$$Z(t) = \theta_0 + \theta_1 \varepsilon(t-1) + \dots + \theta_q \varepsilon(t-q) + \varepsilon(t) \quad (2.2)$$

Nesse modelo, o parâmetro q indica quantos valores passados do erro de previsão são usados como regressores. Os parâmetros θ_i devem ser ajustados conforme a série estiver sendo modelada.

O modelo $ARMA(p,q)$ (*Autoregressive-Moving Average*) é uma generalização dos modelos anteriores, e é formado por uma soma de um componente autoregressivo e um componente de médias móveis. Esses modelos podem ser escritos pela equação 2.3:

$$Z(t) = \phi_0 + \phi_1 Z(t-1) + \dots + \phi_p Z(t-p) + \theta_1 \varepsilon(t-1) + \dots + \theta_q \varepsilon(t-q) + \varepsilon(t) \quad (2.3)$$

Os modelos de Box-Jenkins têm limitações teóricas por serem apenas modelos lineares, sendo, portanto, incapazes de modelar um grande número de fenômenos. As vantagens desses modelos é que eles são simples, fáceis e rápidos de serem implantados.

2.2.3.2 – Modelos neurais de previsão

Outra abordagem utilizada para a previsão de séries temporais é o uso das Redes Neurais Artificiais (RNAs). Estes modelos possuem algumas características que os tornam de grande eficiência para este tipo de tarefa. A primeira é a possibilidade de funcionar como aproximadores universais de funções. Como são modelos não-lineares com um número maior de parâmetros, as RNAs têm o potencial de modelar adequadamente uma quantidade maior de séries em comparação aos modelos lineares clássicos. Outro ponto importante é o relacionamento direto dos modelos de redes neurais com modelos estatísticos, podendo atuar na implantação de modelos lineares e não-lineares.

Existem diversos modelos de redes neurais que podem ser usados para a previsão de séries temporais, tais como *feedforward* e redes recorrentes. Tais modelos serão vistos nos capítulos subsequentes deste trabalho

2.2.4 – TEORIA DO CAOS

Trata-se de uma abordagem relativamente nova para modelagem de sistemas dinâmicos, tais como o mercado de ações, e parte do pressuposto de que não existe ordem em casualidade. Desta forma, segundo Lawrence [38], esta teoria acredita que um processo tem, ao mesmo tempo, características determinísticas e aleatórias.

Vários estudos têm sido desenvolvidos para tentar mostrar que determinados sistemas são caóticos, ou seja, tem *caos* em suas séries temporais. Entende-se por *caos* um processo não-linear com características aleatórias. Com isso, esta teoria contradiz a Hipótese de Mercado Eficiente ao afirmar que o mercado de ações possui comportamento caótico e não simplesmente aleatório.

De acordo com a Teoria do Caos, para a completa modelagem de um sistema caótico é necessário conhecer tanto o comportamento determinístico quanto o comportamento aleatório de tal sistema. Para tanto, os processos determinísticos são caracterizados utilizando informações passadas, enquanto que os aleatórios utilizam parâmetros estatísticos de uma dada função. A habilidade das redes neurais de capturar ambos os aspectos mencionados as torna ideais para a modelagem de tais sistemas.

2.2.5 – OUTRAS TÉCNICAS COMPUTACIONAIS

Muitas outras técnicas baseadas em modelos computacionais têm sido empregadas no mercado financeiro, principalmente no desenvolvimento de softwares para auxiliar negociadores.

Neste ponto, merecem destaque alguns softwares baseados em sistemas especialistas. Tais programas são usados na criação de regras de negociação baseadas em indicadores técnicos, ou, em conjunto com redes neurais, na previsão do comportamento de ações. Em tais sistemas, a rede neural pode realizar as previsões de interesse, enquanto o sistema especialista valida-a baseado em seus conhecimentos de negociação.

O maior obstáculo na utilização de sistemas especialistas encontra-se na dificuldade de formular as regras com as quais o sistema deva trabalhar. Isto se deve ao

fato de que nem mesmo os investidores mais experientes conseguem entender completamente o mercado. Neste momento, as redes neurais apresentam a vantagem de poderem extrair as regras sem que seja necessário o completo entendimento de tais. Em sistemas altamente caóticos ou apenas parcialmente conhecidos, tais como o mercado de ações, este é um fator fundamental.

2.3 – PREVISÃO DO RETORNO DE ÍNDICES DE AÇÕES

Existem dois modelos principais para a previsão de séries financeiras temporais, a saber: modelos de estimação do nível e modelos de previsão por classificação. O primeiro modelo baseia-se na previsão acurada do nível de preços das ações, índices e outros instrumentos de séries financeiras. O segundo é a previsão da direção ou sinal da mudança do nível de preço.

Pesquisa no Portal Capes [14] feita por Souto-Maior [15] por artigos relacionados com previsão de índices e direção de ações, apresentou uma leve tendência pelo primeiro, mostrada na figura 2.3. Dos 48 artigos encontrados, 21 eram sobre previsão de direção; 26 sobre da previsão de nível e somente 1 artigo tratava da previsão de nível e de direção.

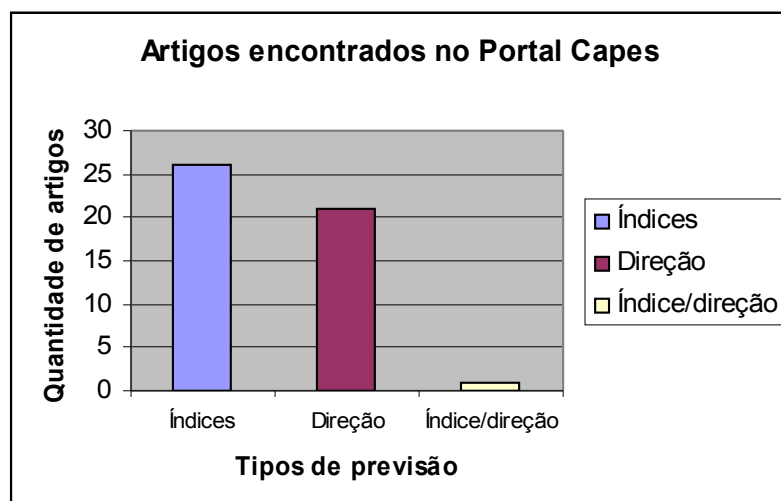


Figura 2.3 – Tipos de artigos encontrados no Portal Capes.

Fonte: Souto-Maior [15]. Adaptado.

Entretanto, pequena parte dos artigos sobre previsão de índices tratava de ações. Em sua maioria, tais artigos possuíam como objetos de estudo índices variados, tais como inflação, taxa de câmbio, desemprego, entre outros.

Em relação ao horizonte de previsão, foram encontrados 34 artigos que utilizavam a previsão diária; 4 que utilizavam a previsão semanal; 3 que utilizavam a previsão mensal; e 7 utilizavam outros horizontes de previsão. A figura 2.4 mostra o gráfico correspondente a estes dados.

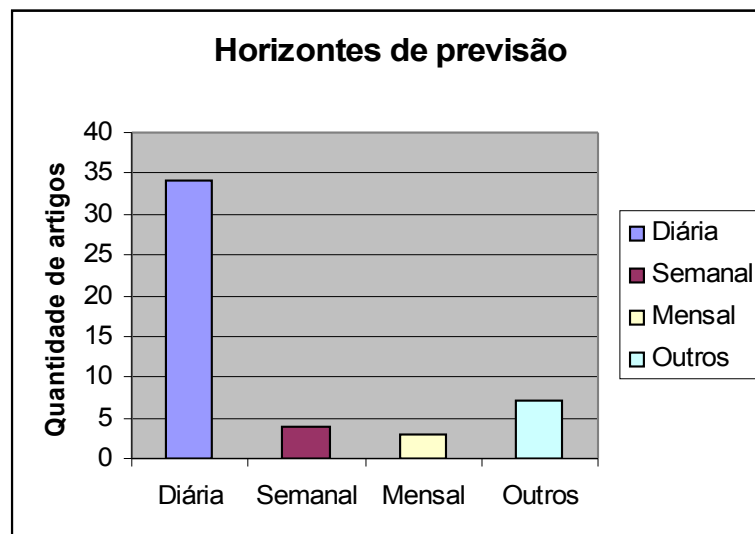


Figura 2.4 – Horizontes de previsão.

Fonte: Souto-Maior [15]. Adaptado.

É interessante destacar que esta mesma pesquisa mostrou que o modelo mais utilizado em tais artigos foi o de redes neurais. Este modelo aparecia sozinho em 18 artigos, ou em modelos híbridos, que juntavam mais de um modelo, em 11 artigos. Percebe-se, dessa forma, o grande emprego das redes neurais em trabalhos de previsão.

2.3.1 – PREVISÃO DO NÍVEL E DE DIREÇÃO UTILIZANDO REDES NEURAIAS

Com relação à previsão de níveis de mercado, destacam-se os seguintes trabalhos encontrados durante a pesquisa mencionada anteriormente:

- o trabalho de Teixeira e Rodrigues [16], que conseguiram bons resultados na previsão do nível diário do *Lisbon Stock Exchange Index* (índice da Bolsa de Valores de Lisboa), Gençay [17], na previsão do nível diário do *Dow Jones Industrial Average Index* (DJIA) entre os anos de 1897 e 1988;
- o trabalho de YU [18], que utilizou redes neurais para previsão do nível diário do NIKKEI 225 futuro negociado no *Singapore International Monetary Exchange* (SIMEX), superando, com isso, o modelo ARIMA,;
- o trabalho de Panda-Narasimhan [19], que aplicaram um modelo *Linear Auto-Regressive* (LAR) e um modelo de redes neurais para previsão do nível diário do *Sensitive Index* (Sensex), obtendo, com este, uma taxa de acerto de até 75,98%;

Dois importantes trabalhos nacionais apareceram durante a pesquisa. O primeiro é o de Gomes [20], na aplicação do modelo ARMA com a abordagem de Box e Jenkins na previsão do nível diário do IBOVESPA. Este modelo obteve um erro percentual médio de 2,67%. E o segundo é o de Ribeiro e Silva [21] que comparando diversos métodos baseados em inteligência artificial com o Random Walk e outros métodos lineares, e concluindo que os métodos de RNA apresentam melhor desempenho.

Com relação à previsão de direção de índices de mercado, destacam-se os trabalhos de:

- Grudnitski e Osburn [22], que utilizaram redes neurais para previsão da direção de movimento mensal do S&P 500 futuro. O desempenho do

modelo foi avaliado pela porcentagem de acerto da direção de movimento, obtendo uma taxa de acerto de até 75% no período de teste;

- Fernández-Rodríguez, González-Martel e Sosvilla-Rivero [23] que utilizaram redes neurais e análise técnica para previsão da direção de movimento diária do Índice General de la Bolsa e Madrid (IGBM). O desempenho dos modelos foi avaliado por dois critérios: (1) porcentagem de acerto da direção de movimento; e (2) retorno obtido por uma estratégia de negociação baseada nesse modelo. Os resultados mostraram que o modelo híbrido superou a estratégia passiva no período de teste e obteve uma taxa de acerto de até 58% no período de teste.
- Armano, Marchesi e Murru [1], estudando a movimentação diária do S&P 500 e do COMIT, propuseram um modelo híbrido usando redes neurais e algoritmos genéticos para prever a sua direção, desenvolvendo duas estratégias usando derivativos dos índices pesquisados. O desempenho dos modelos foi avaliado de acordo com a porcentagem de acertos da direção do movimento e do retorno obtido por uma estratégia de investimentos baseada no modelo desenvolvido. Considerando alguns custos de transação, concluiu-se que o modelo híbrido superava a estratégia passiva no período de testes, porém a taxa de acerto superou o Random Walk para o S&P 500, mas perdeu para o COMIT.

2.4 – O IBOVESPA

A BOVESPA teve sua origem em 1890, com a inauguração da Bolsa Livre, que viria a ser depois a Bolsa de Valores de São Paulo. Em 1935, um ano após se mudar para o Palácio do Café, ganhou o nome de Bolsa Oficial de Valores de São Paulo, e em 1967, passou a se chamar Bolsa de Valores de São Paulo.

A partir da década de 1970, mudanças substanciais aconteceram. Em 1972 começou a funcionar o pregão automatizado e no final desta década houve a introdução de operações com opções. Nos anos 80, implantou-se o Sistema Privado de Operações por Telefone (SPOT) e uma rede de serviços on-line para as corretoras. Em 1990, foram iniciadas as negociações através do Sistema de Negociação Eletrônica – CATS

(Computer Assisted Trading System), que operava simultaneamente com o sistema tradicional de Pregão Viva Voz, e, em 1999, os serviços de *Home Broker* e *After-Market*.

Após o ano de 2006 a BOVESPA passou a concentrar toda a negociação de ações no país, se tornando a única bolsa de valores do Brasil e a maior da América Latina, com cerca de 70% de todo o volume de negócios da região.

O IBOVESPA é o mais importante indicador do desempenho médio das cotações do mercado de ações brasileiro. Ele representa o valor atual, em moeda corrente, de uma carteira teórica de ações constituída em 1968, a partir de uma aplicação hipotética. Supõe-se não ter sido efetuado nenhum investimento adicional desde então, apenas alguns pequenos ajustes. Dessa forma, o índice reflete não apenas as variações dos preços das ações, mas também o impacto da distribuição dos proventos, sendo considerado um indicador que avalia o retorno total de suas ações componentes.

A carteira teórica do Índice é integrada pelas ações que, em conjunto, atenderam a dois critérios: representaram 80% do volume transacionado a vista nos doze meses anteriores à formação da carteira, e apresentaram, no mínimo, 80% de presença nos pregões do período.

A cada quatro meses o mercado é reavaliado, sempre com base nos doze meses anteriores. Isto ocorre para que possíveis alterações na participação relativa de cada ação sejam detectadas. Após a reavaliação, uma nova carteira é montada, com cada papel recebendo um novo peso, segundo a distribuição de mercado.

2.4.1 – METODOLOGIA DO CÁLCULO

2.4.1.1 – Participação da ação na carteira teórica

A participação de cada ação na carteira tem relação direta com a representatividade desse título no mercado a vista – em termos de número de negócios e volume financeiro – ajustada ao tamanho da amostra. Essa representatividade é obtida pelo índice de negociabilidade de ação, calculado através da equação 2.4 a seguir:

$$IN = \sqrt{\left(\frac{n_i}{N} \times \frac{v_i}{V}\right)} \quad (2.4)$$

onde: IN = índice de negociabilidade;

n_i = número de negócios realizados com a ação no mercado a vista (lote-padrão), nos últimos 12 meses;

N = número de negócios total do mercado a vista (lote-padrão) dos últimos 12 meses;

v_i = valor, em moeda corrente, movimentado com a ação no mercado a vista (lote-padrão), nos últimos 12 meses;

V = valor total do mercado a vista (lote-padrão) em moeda corrente dos últimos 12 meses.

2.4.1.2 – Apuração do Índice

O cálculo do IBOVESPA é feito realizando o somatório dos pesos (quantidade teórica da ação multiplicada pelo último preço da mesma) das ações integrantes de sua carteira teórica, como pode ser observado na equação 2.5 a seguir:

$$IBVSP_t = \sum_{i=1}^n P_{i,t} \times Q_{i,t} \quad (2.5)$$

onde: $IBVSP_t$ = Índice Bovespa no instante t;

n = número total de ações componentes da carteira teórica;

$P_{i,t}$ = último preço da ação i no instante t;

$Q_{i,t}$ = quantidade teórica da ação i na carteira no instante t ;

2.4.1.3 – Alteração da quantidade teórica em função de proventos

O mecanismo de alteração é semelhante ao utilizado para o ajuste da carteira como um todo, ou seja, considerando-se que o investidor realizou (vendeu) as ações ao último preço de fechamento anterior ao início da distribuição de provento e utilizou os recursos na compra das mesmas ações sem o provento distribuído ('ex provento').

A equação 2.6, a seguir, mostra o cálculo da alteração da quantidade teórica:

$$Q_n = \frac{Q_o \times P_c}{P_{ex}} \quad (2.6)$$

onde: Q_n = quantidade nova;

Q_o = quantidade antiga;

P_c = último preço de fechamento anterior ao início da negociação 'ex-provento';

P_{ex} = preço ex-teórico, calculado com base em P_c .

2.4.1.4 – Cálculo do Preço 'Ex-Teórico'

O cálculo do preço 'ex-teórico' é dado pela equação 2.7, a seguir:

$$P_{ex} = \frac{P_c + (S \times Z) - D - J - V_{et}}{1 + B + S} \quad (2.7)$$

onde: P_{ex} = preço ex- teórico;

P_c = último preço 'com direito' ao provento;

S = percentual de subscrição, em número índice;

Z = valor, em moeda corrente, de emissão de cada ação a ser subscrita;

D = valor, em moeda corrente, recebido por ação a título de dividendo;

J = juros sobre capital, em moeda corrente;

V_{et} = valor econômico teórico, resultante do recebimento de provento em outro tipo/ativo;

B = percentual de bonificação (ou desdobramento), em número índice.

3 – AS REDES NEURAIS ARTIFICIAIS

3.1 – INTRODUÇÃO

Uma previsão eficiente do futuro pode ser de grande valor em áreas tão diversas quanto o movimento de preços do mercado financeiro, a previsão de estouro de orçamento, o cálculo de riscos de seguro e eficácia de drogas medicinais, entre outras. Muitas técnicas, desde as regressões lineares ao método ARMA para séries temporais, são regularmente usadas para gerar previsões. Uma abordagem por redes neurais permite a utilização de uma técnica que pode operar em circunstâncias onde as técnicas clássicas não podem ou não produzem resultados satisfatórios.

Em uma visão geral, prever significa utilizar dados e modelos existentes para gerar outros que ainda não existem. Os dados existentes podem ser de muitas formas e incluir redundâncias, erros, ausência de valores, ruído e muitas outras características que dificultam a tarefa de geração de previsão. Além disso, qualquer modelo conhecido pode também ter problemas tais como erros estruturais ou tendências. A respeito de todos esses desafios, os dados existentes, em muitos casos, contêm todas as informações necessárias para fornecer uma previsão útil e acurada das variáveis de objetivo. O desafio, então, está em isolar as informações úteis nos dados existentes, separando-as de ruídos e erros.

Alguns dos métodos tradicionais de previsão incluem regressões lineares e não-lineares, séries temporais ARMA, regressão logística, análises de componentes principais, análise discriminante e de clusters. Muitos desses métodos requerem um analista humano para filtrar as informações importantes entre dezenas ou centenas de variáveis. Em circunstâncias onde a quantidade de dados é particularmente grande e complexa, onde somente pequena parte da série temporal está disponível ou quando há a presença de muito ruído, estas abordagens podem ser tornar extremamente difíceis.

As redes neurais oferecem uma abordagem de modelagem e previsão que podem se ajustar a circunstâncias nas quais os dados existentes possuem informações de interesse, porém, estas podem estar mascaradas pelos diversos fatores mencionados anteriormente.

Estes atributos fazem das redes neurais uma excelente ferramenta para tomar o lugar de um ou mais métodos tradicionais, tais como regressões lineares e método dos mínimos quadrados. Com elas é possível gerar previsões úteis em situações onde outras técnicas não poderiam ser aplicadas de forma tão eficiente. Em outras situações, as redes neurais podem melhorar substancialmente as previsões ao levar em consideração mais informações do que as técnicas tradicionais são capazes de processar. Finalmente, o uso de redes neurais na criação de um modelo preditivo de um sistema complexo não necessita da atuação de um estatístico ou outro especialista para analisar todas as combinações possíveis de variáveis, diminuindo sensivelmente o tempo de trabalho.

3.2 – O QUE É UMA REDE NEURAL?

Uma Rede Neural Artificial, ou simplesmente Rede Neural, são sistemas paralelos distribuídos, compostos por unidades de processamentos simples (neurônios), baseadas no funcionamento do cérebro. Estas unidades têm por objetivo computar determinadas funções matemáticas (normalmente não lineares), e são dispostas em uma ou mais camadas interligadas por um grande número de conexões, geralmente, unidirecionais. Na maioria dos modelos, estas conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. As RNAs têm a capacidade de, como as pessoas, aprenderem através de exemplos, sendo configuradas para cada tipo específico de aplicação, tais como reconhecimento de padrão ou classificação de dados.

A solução de problemas através das RNAs é bastante atrativa devido ao seu paralelismo, criando, desta forma, a possibilidade de um desempenho superior em relação à solução de problemas baseados nos modelos convencionais. Em RNAs, o procedimento usual na solução de problemas passa inicialmente por uma fase de aprendizagem, onde um conjunto de exemplos é apresentado para a rede, que extrai automaticamente as características necessárias para representar a informação fornecida. Essas características são utilizadas posteriormente para gerar respostas para o problema.

A capacidade de aprender através de exemplos e de generalizar a informação aprendida são os atrativos principais da solução de problemas através das RNAs. A generalização está associada à capacidade da rede em aprender através de um conjunto

reduzido de exemplos e, posteriormente, dar respostas coerentes a dados não apresentados à rede. Isto é uma demonstração que a capacidade das RNAs vai além de, simplesmente mapear relações de entrada e saída, sendo capazes de extrair informações não-apresentadas de forma explícita através de exemplos. Outras características importantes são a capacidade de auto-organização e processamento temporal que, aliadas àquela citada anteriormente, fazem das RNAs uma ferramenta computacional extremamente poderosa e atrativa para a solução de problemas complexos.

3.3 – CÉREBRO HUMANO

O sistema nervoso humano pode ser visto como um sistema de três estágios, como mostrado na figura 3.1. No centro está uma unidade de processamento, representado pela rede de *neural* (de nervos), a qual continuamente recebe informações, distingue-as e toma as decisões apropriadas. Dois conjuntos de setas são mostrados na figura. Os que apontam da esquerda para direita indicam que a informação está se movimentando numa direção no sistema. Aquelas que apontam da direita para esquerda, indicam a presença de um sistema de realimentação. Os *receptores* convertem estímulos do corpo humano ou do ambiente exterior em impulsos elétricos que carregam as informações até a rede neural (cérebro). Os *executores* convertem os impulsos elétricos gerados pela rede neural em respostas perceptíveis (as saídas do sistema).

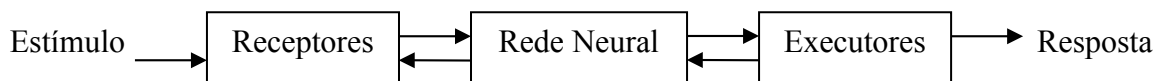


Figura 3.1 – Diagrama de blocos da representação do sistema nervoso.

Fonte: Haykin[24], pág 28. Adaptado.

Em 1911, o estudo de Ramon y Cajál (*apud* Haykin [24]) introduziu a idéia de neurônios como estruturas constituintes do cérebro. Hoje estimam-se que haja 10 bilhões de neurônios no córtex humano e 60 trilhões de sinapses ou conexões. Sinapses são estruturas elementares e unidades funcionais que atuam nas interações entre neurônios. Elas funcionam como válvulas, sendo capazes de controlar a transmissão de impulsos, isto é, o fluxo da informação entre os neurônios na rede neural, inibindo-os ou excitando-os. Devido ao seu efeito variável, as sinapses dão aos neurônios grande capacidade de adaptação.

O neurônio biológico é uma célula que pode ser dividida em três seções: *o corpo da célula, os dendritos e o axônio*, cada uma com funções específicas, porém complementares.

O corpo do neurônio mede apenas alguns milésimos de milímetros e os dendritos apresentam poucos milímetros de comprimento. Os dendritos têm como função receber informações, ou *impulsos nervosos*, oriundas de outros neurônios, e conduzi-las até o corpo celular. Ali a informação é processada e novos impulsos são gerados. Estes impulsos são transmitidos a outros neurônios, passando pelo do axônio e atingindo os dendritos dos neurônios seguintes.

A figura 3.2 ilustra, de forma simplificada, os componentes do neurônio:

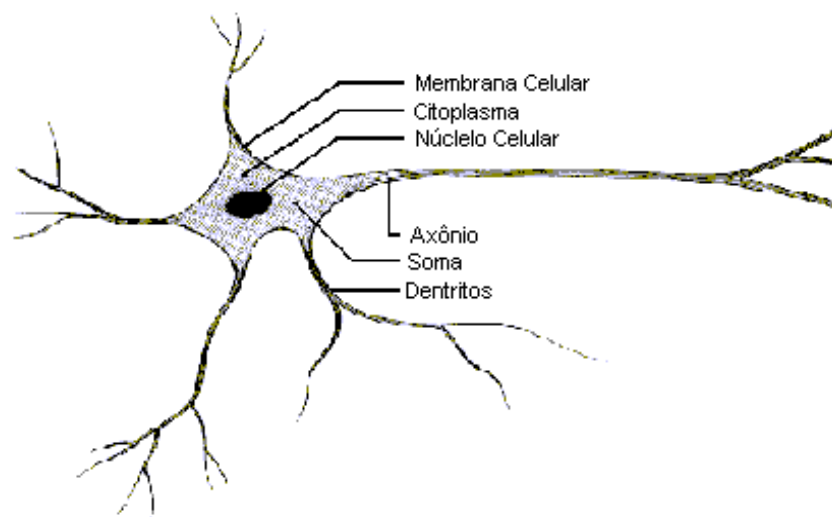


Figura 3.2 – Componentes de um Neurônio Biológico

Fonte: Haykin [24]. pág. 30. Adaptado.

Os sinais oriundos dos neurônios *pré-sinápticos* são passados para o corpo do neurônio, onde são comparados com outros sinais recebidos por este mesmo neurônio. Se o percentual em um intervalo curto de tempo é suficientemente alto, a célula “dispara”, produzindo um impulso que é transmitido para as células seguintes (neurônios *pós-sinápticos*).

3.4 – MODELOS DE NEURÔNIOS

Como já visto, a construção de redes neurais artificiais tem inspiração nos neurônios biológicos e nos sistemas nervosos. Entretanto, é importante compreender que atualmente as RNAs estão muito distantes das redes neurais naturais (RNNs) e freqüentemente as semelhanças são mínimas.

O primeiro modelo de neurônio artificial foi proposto por McCulloch e Pitts [25], em 1943, e tinha a intenção de imitar a realidade biológica. Porém esta preocupação não é totalmente compartilhada por muitos pesquisadores atuais. De fato, dois fatores diferentes motivam a pesquisa hoje em dia:

- modelar o sistema nervoso com suficiente precisão de tal modo a se poder observar um comportamento que, sendo semelhante ao comportamento do ser vivo, possa servir de apoio às hipóteses usadas na modelagem;
- aproveitar as características da RNAs para construir computadores com um alto grau de paralelismo.

3.4.1 – MODELO DE MCCULLOCH-PITTS

O modelo de neurônio proposto por McCulloch e Pitts [25] é uma simplificação do que se sabia a respeito de neurônios biológicos à época do estudo. Sua descrição matemática resultou em um modelo com n terminais de entrada x_1, x_2, \dots, x_n (que representam os dendritos), e apenas um terminal de saída y (representando o axônio). Para emular o comportamento das sinapses, os terminais de entrada do neurônio têm

pesos acoplados w_1, w_2, \dots, w_n , cujo valores podem ser positivos ou negativos, dependendo das sinapses correspondentes serem inibitórias ou excitatórias. O efeito de uma sinapse particular i no neurônio pós-sináptico é dado pelo produto $x_i w_i$. Os pesos determinam “em que grau” o neurônio deve considerar sinais de disparo que ocorrem naquela conexão. Uma descrição do modelo está ilustrada na figura 3.3.

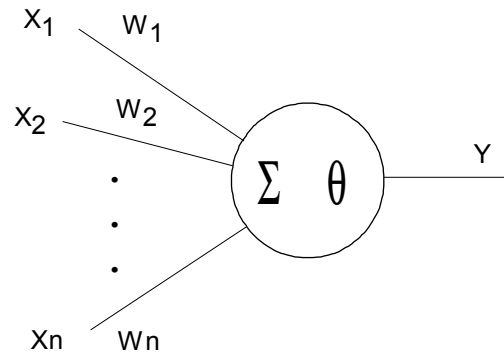


Figura 3.3 – Neurônio de McCulloch-Pitts

Um neurônio dispara quando a soma dos impulsos que ele recebe ultrapassa o seu limiar de excitação, chamado de *threshold*. O corpo do neurônio, por sua vez, é emulado por um mecanismo simples que faz a soma dos valores $x_i w_i$ recebidos pelo neurônio (soma ponderada) e decide se o neurônio deve ou não disparar (saída igual a 1 ou a 0) comparando a soma obtida ao limiar ou *threshold* do neurônio. No modelo MCP, a ativação do neurônio é obtida através da aplicação de uma “função de ativação”, que ativa a saída ou não, dependendo do valor da soma ponderada das suas entradas. Na descrição original do modelo MCP, a função de ativação é dada pela função limiar descrita na subseção seguinte. A inequação 3.1, a seguir, modela a ativação da saída do neurônio MCP:

$$\sum_{i=1}^n x_i w_i \geq \theta \quad (3.1)$$

onde n é o número de entradas do neurônio, w_i é o peso associado à entrada x_i , e θ é o limiar (threshold) do neurônio.

McCulloch e Pitts simplificaram seu modelo assumindo que os neurônios em cada camada da rede disparam sincronamente, isto é, que todos os nós (ou nodos) são avaliados ao mesmo tempo e também que as entradas em um instante de tempo t produzem a sua saída no tempo $t + 1$.

Finalmente, podem-se levantar algumas limitações na descrição do modelo MCP original, como:

- redes MCP com apenas uma camada conseguem implementar funções linearmente separáveis;
- pesos negativos são mais adequados para representar disparos inibidores;
- O modelo foi proposto com pesos fixos, não ajustáveis. Não há a presença de um algoritmo de aprendizagem.

3.4.2 – FUNÇÕES DE ATIVAÇÃO

A partir do modelo proposto por McCulloch e Pitts, foram derivados vários outros modelos que permitem a produção de uma saída qualquer, não necessariamente zero ou um, e com diferentes funções de ativação.

A figura 3.4 ilustra, graficamente, quatro funções de ativação diferentes: (a) função linear, (b) função rampa, (c) função degrau (*step*) e (d) função sigmoideal.

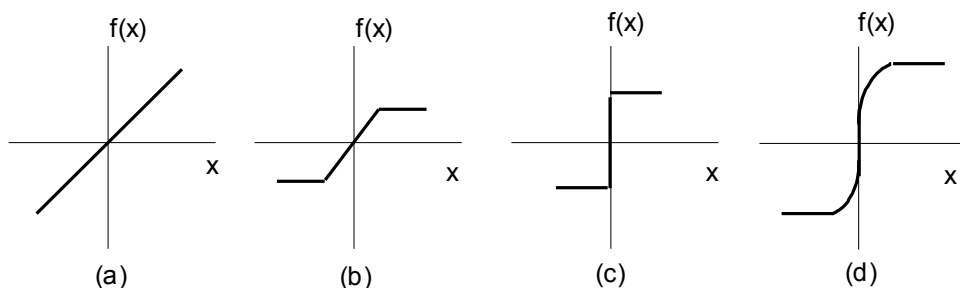


Figura 3.4 – Algumas funções de ativação

A função de ativação linear, mostrada na figura anterior, é definida pela equação 3.2 a seguir.

$$y = \alpha x \quad (3.2)$$

onde α é um número real que define a saída linear para os valores de entrada, y é a saída e x é a entrada.

A função linear pode ser restringida para produzir valores constantes em uma faixa $[-\gamma, +\gamma]$, e neste caso, passa a ser a função rampa mostrada, graficamente, na figura 3.4 item (b) definida pela equação 3.3 abaixo:

$$y = \begin{cases} +\gamma & \text{se } x \geq \gamma \\ x & \text{se } x = \gamma \\ -\gamma & \text{se } x \leq -\gamma \end{cases} \quad (3.3)$$

A função *step*, ilustrada na figura 3.4 item (c), é definida pela equação 3.4 a seguir:

$$y = \begin{cases} +\gamma & \text{se } x > 0 \\ -\gamma & \text{se } x \leq 0 \end{cases} \quad (3.4)$$

A função sigmoideal, conhecida também como *S-shape*, ilustrada na 3.4 item (d), é uma função semi-linear. É possível definir várias funções sigmoideais; uma das mais importantes é a função logística definida pela equação 3.5 a seguir:

$$y = \frac{1}{1 + e^{-x/T}} \quad (3.5)$$

onde o parâmetro T determina a suavidade de curva.

3.5 – O PROCESSO DE APRENDIZADO

Como já visto anteriormente, as RNAs possuem as capacidades de aprender através exemplos e de fazer interpolações e extrapolações do que aprenderam. No aprendizado conexionista não se procura obter regras como na abordagem simbólica da Inteligência Artificial (IA), mas determinar a intensidade das conexões entre os neurônios. Um conjunto de procedimentos bem definidos para adaptar os parâmetros de uma RNA, a fim de que a mesma possa *aprender* uma determinada função, é chamado de *algoritmo de aprendizagem*. Há vários tipos de algoritmos de aprendizagem, cada um com suas vantagens e desvantagens, diferindo, basicamente, pela maneira através da qual o ajuste dos pesos é feito.

A fase de aprendizagem de uma RNA consiste na extração, por parte da rede, de características relevantes dos padrões de informações apresentados para a mesma. Trata-se de um processo iterativo de ajuste dos parâmetros da rede, ou seja, dos pesos das conexões entre as unidades de processamento. É nestes pesos que a rede guarda o conhecimento adquirido do ambiente em que está operando.

De acordo com Haykin [24], ‘aprendizado [*de uma rede neural*] é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação do ambiente no qual a rede está inserida; sendo determinado pela forma através da qual é efetuada a mudança nos parâmetros’.

3.5.1 – PARADIGMAS DE APRENDIZADO

Existem três métodos básicos de aprendizados em RNAs: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado por reforço.

3.5.1.1 – Aprendizado supervisionado

Este é o método mais comum no treinamento de RNAs. Trata-se de um modelo no qual o aprendizado é obtido através de um *supervisor* (ou *professor*) com o objetivo de ajustar os parâmetros da rede, de forma a encontrar uma ligação entre os pares de entrada e saída fornecidos.

A figura 3.5 ilustra o mecanismo deste paradigma.

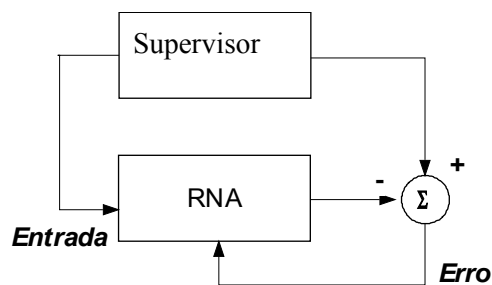


Figura 3.5 – Aprendizado Supervisionado.

O *supervisor* indica um comportamento bom ou ruim para a rede, visando direcionar o processo de treinamento. A rede tem sua saída atual comparada com a saída desejada, recebendo informações do supervisor sobre o erro da resposta atual. A cada padrão de entrada submetido à rede, compara-se a resposta desejada com a resposta calculada, e os pesos das conexões são ajustados de forma a minimizar o erro. A minimização da diferença é incremental, já que pequenos ajustes são feitos nos pesos a cada etapa de treinamento, de tal forma que estes caminhem, se possível, para uma solução.

A desvantagem deste tipo de aprendizado está exatamente na necessidade de um *supervisor*. Na sua ausência, a rede não consegue aprender novas estratégias para situações não vistas nos exemplos de treinamento. Os exemplos mais conhecidos de algoritmos de aprendizagem supervisionada são a regra delta ou correção de erros e a sua generalização para redes de múltiplas camadas, o algoritmo de retropropagação (*back-propagation*). Ambos os algoritmos serão vistos mais adiante neste capítulo.

3.5.1.2 – Aprendizado não-supervisionado

Diferentemente do anterior, este paradigma não necessita de um *tutor* para acompanhar o processo de aprendizado e indicar a saída desejada. Durante o treinamento, somente padrões de entrada são apresentados à RNA, até que ela estabeleça uma harmonia com as regularidades estatísticas da entrada de dados. Porém, este tipo de aprendizado só é possível quando os dados de entrada são, de alguma forma, redundantes. A figura 3.6 a seguir mostra um diagrama deste tipo de aprendizado:

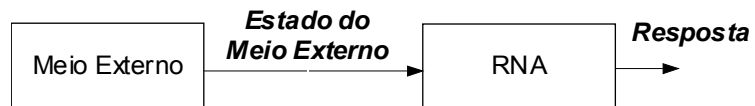


Figura 3.6 – Aprendizado Não Supervisionado

3.5.1.3 – Aprendizado por reforço

No aprendizado por reforço, o aprendizado de um mapeamento de entrada e saída é feito através de interações contínuas com o ambiente, de modo a minimizar um índice escalar de desempenho. Em sistemas deste tipo, somente uma única informação de realimentação é fornecida à rede: se uma determinada saída está correta ou não. A figura 3.7, a seguir, mostra o diagrama de blocos de um sistema de aprendizado por reforço construído em torno de um *crítico*, que converte um sinal de reforço primário recebido do ambiente em um sinal de reforço de maior qualidade, chamado de sinal de reforço heurístico, ambos sendo entradas escalares.

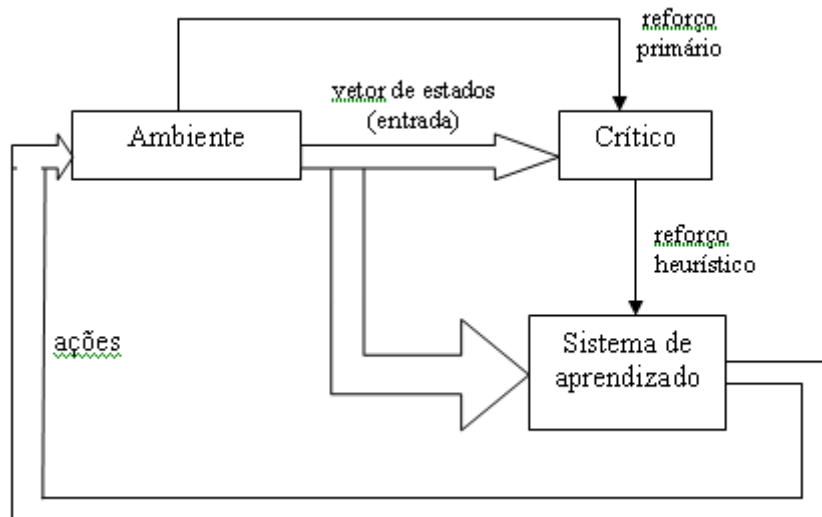


Figura 3.7 – Diagrama de blocos de Aprendizado por Reforço.

Fonte: Haykin[24]. Adaptado.

3.5.2 – REGRAS DE APRENDIZADO

A seguir estão listados alguns dos mais usados *algoritmos de aprendizagem*.

3.5.2.1 – Regra Delta ou Correção de Erros

O objetivo deste processo é minimizar a diferença entre a soma ponderada das entradas pelos pesos (saída calculada pela rede) e a saída desejada, ou seja, o erro da resposta atual da rede.

O erro é calculado de acordo com a equação 3.6, a seguir:

$$e(t) = d(t) - y(t) \quad (3.6)$$

onde: $e(t)$ = erro da resposta atual;

$d(t)$ = é a saída desejada;

$y(t)$ = a resposta atual (calculada) no instante de tempo t .

O ajuste dos pesos deve ser proporcional ao produto do erro pelo valor de entrada da sinapse naquele instante de tempo, segundo a equação 3.7, a seguir:

$$w_i(t + 1) = w_i(t) + \eta \times e(t) \times x_i(t) \quad (3.7)$$

onde: $\eta(t)$ = taxa de aprendizado;

x_i = entrada para o neurônio no tempo t .

3.5.2.2 – Aprendizado Hebbiano

Consiste na mais antiga regra de aprendizagem e foi baseada no postulado de Hebb [26]:

‘Quando um axônio de uma célula A está próximo o bastante para excitar uma célula B e repetidamente ou persistentemente participa de sua ativação, algum processo de crescimento ou alteração metabólica acontece em uma ou ambas as células, tal que a eficiência de A como uma das células que ativa B é aumentada’.

Este postulado foi formulado com base no aprendizado associativo (a nível celular), em um contexto neurobiológico, o que resulta em uma modificação duradoura no padrão de atividade de um conjunto de células nervosas espacialmente distribuído. Expandindo este conceito e separando-o em duas regras, temos:

Se dois neurônios em cada um dos lados da sinapse (conexão) são ativados simultaneamente (i.e sincronamente), então a força daquela sinapse é seletivamente aumentada.

Se dois neurônios em cada um dos lados da sinapse são ativados de forma assíncrona, então aquela sinapse é seletivamente enfraquecida ou eliminada.

Uma sinapse com tais características é chamada de sinapse Hebbiana e possui, segundo Haykin [24], quatro propriedades principais:

- *Mecanismo interativo*: a ocorrência de uma mudança em uma sinapse Hebbiana depende dos sinais em ambos os lados desta sinapse. Isto é, o método de aprendizado Hebbiano depende, estatisticamente ou deterministicamente, de suas ‘interações verdadeiras’ entre os sinais pré-sinápticos e pós-sinápticos;
- *Mecanismo local*: uma sinapse Hebbiana é uma transmissão com sinais contínuos que produz modificações sinápticas locais que são entradas específicas. É a partir do mecanismo local que sinapses Hebbianas efetuam aprendizado não-supervisionado;
- *Mecanismo dependente do tempo*: as modificações em uma sinapse Hebbiana dependem do momento exato de ocorrência das atividades pré e pós-sinápticas;
- *Mecanismo correlacional ou conjuncional*: a sinapse Hebbiana pode ser chamada de sinapse conjuncional pelo fato de que a ocorrência conjunta

de atividades pré e pós-sinápticas é suficiente para que haja uma modificação. Além disso, pode também ser chamada de sinapse correlacional, porque uma correlação entre estas mesmas atividades é suficiente para gerar mudanças.

A equação 3.8 expressa o postulado de Hebb, consistindo em uma regra para a mudança do peso sináptico w_{ij} (regra do produto de atividade):

$$\Delta w_{ij}(t) = \eta \cdot y_i(t)x_j(t) \quad (3.8)$$

onde: η = constante positiva que determina a taxa de aprendizado;

y_i = saída desejada no instante de tempo t ;

x_i = entrada do neurônio j no instante de tempo t .

Esta regra de aprendizagem é classificada com o aprendizado não-supervisionado, mesmo sendo os vetores de entrada e saída fornecidos. Isto se deve ao fato de não haver supervisor externo para verificar a qualidade da resposta da rede para prover ajustes dos pesos.

3.5.2.3 – Aprendizado competitivo

Neste tipo de aprendizado, como o próprio nome indica, os neurônios de saída da rede competem entre si para se serem ativados. Enquanto em redes baseadas em aprendizado Hebbiano vários neurônios de saída podem ser ativados simultaneamente,

no aprendizado competitivo apenas um único neurônio de saída é ativado por vez. É esta característica que torna este tipo de rede altamente indicada para descobrir características estatísticas que possam ser usados para classificar um conjunto de padrões de entrada.

Segundo Haykin [24], há três elementos básicos para uma regra de aprendizado competitivo:

- um conjunto de neurônios que são todos iguais, exceto por alguns pesos sinápticos aleatoriamente distribuídos, e que responde de maneira diferente a um dado conjunto de padrões de entrada;
- um limite imposto ‘à força’ a cada neurônio;
- um mecanismo que permite a competição entre neurônios pelo direito a responder a um dado subconjunto de entradas, tais que apenas um neurônio, ou apenas um neurônio por grupo, é ativado por vez.

A seguir, tem-se um algoritmo simples de aprendizado competitivo:

- apresentar um vetor de entrada;
- calcular a ativação inicial de cada unidade de saída;
- deixar as unidades de saída competirem até que apenas uma permaneça ativa;
- aumentar os pesos sobre as conexões entre as unidades de saída ativa e as unidades de entrada ativas. Com isso, a unidade de saída terá maior probabilidade de ficar ativa na próxima repetição do padrão.

Porém, para evitar que uma unidade de saída torne-se dominante e fique ativa todo tempo, captando para si todo o espaço de entrada, é necessário racionar os pesos de maneira que a soma dos pesos sobre as linhas de entrada de uma unidade seja limitada a 1.

3.6 – ALGUMAS ARQUITETURAS DE RNAS

Nesta seção serão abordados os modelos de perceptron de única e múltiplas camadas. O perceptron foi proposto por Rosenblatt [27], em 1958, como uma extensão

do modelo de neurônio de McCulloch-Pitts, substituindo a função degrau por uma função contínua que mapeia Z em Y , tornando mais fácil o treinamento das redes. Seu modelo consiste, basicamente, de uma estrutura de rede tendo como unidades básicas neurônios MCP e uma regra de aprendizado. Foi somente após este trabalho que o conceito *aprendizado* em RNAs foi introduzido. Alguns anos mais tarde, Rosenblatt demonstrou o teorema de convergência do perceptron: um neurônio MCP treinado com o algoritmo de aprendizagem do perceptron sempre converge, caso o problema em questão seja linearmente separável.

3.6.1 – PERCEPTRON DE CAMADA ÚNICA

A topologia do modelo proposto por Rosenblatt era composta de: unidades de entrada (retina), um nível intermediário (formado pelas unidades de associação) e um nível de saída (formado pelas unidades de resposta). Apesar de esta topologia possuir três níveis, somente o nível de saída (unidade de resposta) possui propriedades adaptativas, o que explica o nome *perceptron camada única*.

Na figura 3.8, a seguir, tem-se um esboço da topologia do perceptron:

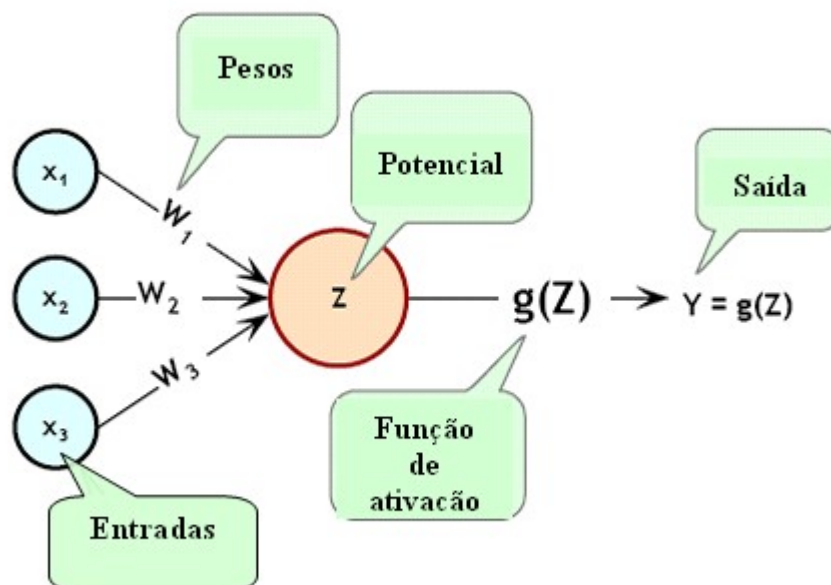


Figura 3.8 – Topologia de um perceptron simples com uma única saída.
Fonte: Jones [28]. Adaptado.

onde, $Z = \sum_{i=1}^3 W_i X_i - \mu$; sendo μ um bias.

Apesar de introduzir importantes novidades e da grande euforia da comunidade acadêmica da época, o modelo de neurônio de Rosenblatt sofreu duras críticas devido a sua capacidade computacional, causando um grande desinteresse por novas pesquisas. Este quadro só mudou com a introdução do algoritmo de backpropagation, alguns anos mais tarde.

3.6.2 – PERCEPTRON MULTICAMADAS

Conforme citado anteriormente, as redes de camada única resolvem apenas problemas linearmente separáveis. Para aplicações em problemas não linearmente separáveis foi necessário o desenvolvimento das redes multicamadas (ou perceptrons multicamadas). Segundo Haykin [24], estas redes consistem de um conjunto de unidades sensoriais (neurônios fontes), que constituem a camada de entrada, de uma ou mais camadas escondidas, e de uma camada de saída. A utilização de uma rede com uma camada escondida permite a implementação de qualquer função contínua, enquanto que uma rede com duas camadas escondidas poder aproximar qualquer função [29,30].

Segundo Haykin [24], um perceptron multicamadas tem três características distintas:

- 1) o modelo de cada neurônio na rede inclui uma função de ativação não-linear. Porém, é importante enfatizar que esta não-linearidade é suave, ou seja, a função deve ser diferenciável em todos os pontos, em oposição ao perceptron original de Rosenblatt;
- 2) a rede contém um ou mais neurônios escondidos que não fazem parte da entrada ou saída da rede. Estes neurônios permitem à rede aprender tarefas complexas, extraíndo progressivamente mais características significativas dos vetores de entrada;

3) a rede exibe um alto grau de conectividade, determinada por suas sinapses. Uma mudança nesta conectividade requer uma mudança nas conexões sinápticas ou em seus pesos.

Na combinação destas características com a habilidade de aprender por experiências através de treinamentos reside o poder computacional das redes MLP (Multilayer Perceptrons). Entretanto, estas mesmas características são responsáveis por deficiências no comportamento do perceptron multicamadas. Primeiro, a presença de uma forma distribuída de não-linearidade e a alta conectividade da rede tornam a análise mais complicada. Segundo, o uso de camadas escondidas faz com que o processo de aprendizagem seja mais difícil de visualizar. De uma maneira geral, o processo de aprendizagem deve decidir quais características dos vetores de entrada deverão ser representadas pelos neurônios da camada escondida. Este processo torna-se, então, mais difícil à medida que a busca por tais padrões tem que ser conduzida em um espaço muito maior de funções de ativação.

A figura 3.9 mostra um exemplo de rede multicamada.

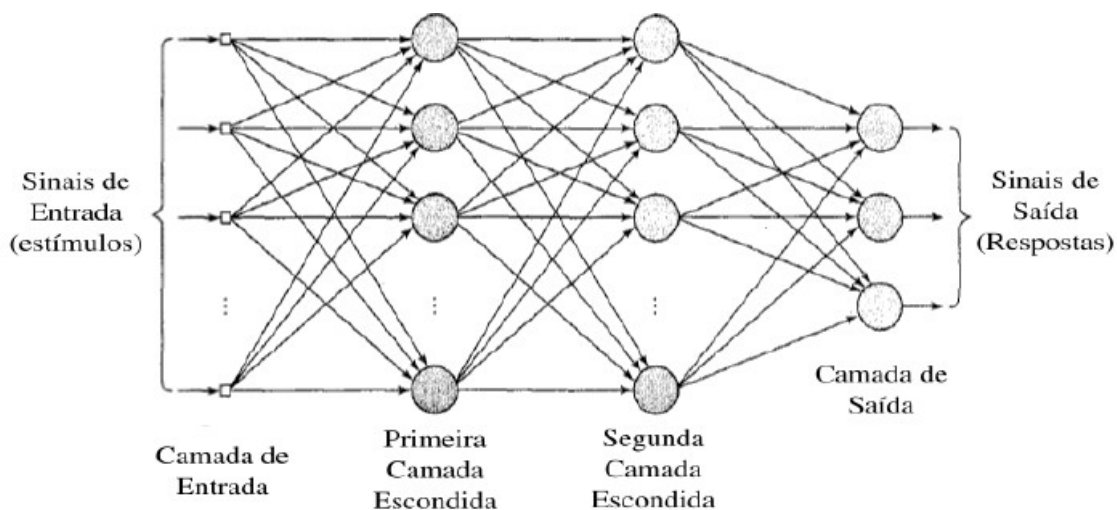


Figura 3.9 – Rede MLP com camada intermediária.

Fonte: Haykin [24], Extraído.

Como já visto, uma função de ativação não-linear é necessária para cada neurônio de entrada da rede. Esta função deve informar os erros cometidos pela rede para as camadas anteriores com a maior precisão possível. Há diversos tipos de função

de ativação; todavia, a mais utilizada é a *sigmoidal logística*, representada na equação 3.9 a seguir:

$$f(v) = \frac{1}{1 + e^{(-\alpha v)}} \quad , \quad \alpha > 0 \quad (3.9)$$

onde α (*parâmetro de inclinação*) é responsável pela inclinação da curva produzida pela função e v é o modelo da ativação da saída do neurônio MCP, definido em 3.4.1.

3.6.2.1 – Definição de uma topologia ideal para uma rede multicamadas

Definir uma topologia de uma rede MLP é, essencialmente, definir o número de neurônios da camada intermediária. Geralmente, este número é definido empiricamente e depende da distribuição dos padrões de treinamento e validação da rede e de alguns outros fatores, tais como número de exemplos a ser aprendido, complexidade da função a ser aprendida, quantidade de ruído presente nos exemplos, entre outros. Todavia, alguns métodos têm sido propostos. Os mais utilizados são:

- Definir o número de neurônios em função do número de entradas e saídas;
- Utilizar um número de conexões em torno de dez vezes menor que o número de exemplos. Este método reduz a incidência de *overfitting* da rede (memorização de padrões de treinamento, ao invés de extração de características). Contudo, caso o número de exemplos seja muito maior que o número de conexões, pode ocorrer *underfitting* (não convergência da rede durante o seu treinamento).

Uma forma de evitar o *overfitting* é estimar o erro de generalização durante o processo de treinamento. Para isto, divide-se o conjunto de dados em *conjunto de treinamento e conjunto de validação*. O conjunto de treinamento é utilizado na modificação dos pesos e o conjunto de validação é utilizado para estimar a capacidade de generalização da rede durante o processo de aprendizagem.

Deve-se interromper o treinamento quando o erro do conjunto de validação começar a aumentar; isto significa que a rede está incorporando o ruído presente nos

dados, degradando, assim, sua capacidade de generalização. Esta alternativa é mais indicada para os casos em que o conjunto de treinamento é muito grande, visto que os dados do conjunto de validação não podem ser utilizados para treinamento.

3.6.2.2 – Treinamento de redes multicamadas

Os algoritmos para treinamento de redes MLP são do tipo supervisionado e podem ser classificados em:

- Estáticos - não alteram a estrutura da rede, apenas variam os valores de seus pesos;
- Dinâmicos. - alteram a configuração da rede, alterando o número de camadas, o número de neurônios nas camadas escondidas e o número de conexões.

O algoritmo de aprendizagem mais conhecido para treinamento destas redes é o *back-propagation*. Este algoritmo é do tipo dinâmico e é utilizado, com variações, em grande parte dos métodos de aprendizado de RNAs do tipo MLP.

3.6.2.2.1 – Back-propagation

O objetivo primário do treinamento de uma rede neural é estimar um conjunto apropriado de pesos baseado nos dados de entrada. O algoritmo de *back-propagation*, também chamado *regra delta generalizada*, trata-se de um algoritmo supervisionado que utiliza pares de entrada e saída esperada para ajustar tais pesos, através do gradiente.

O treinamento ocorre em duas fases (forward e backward), mostradas na figura 3.10 a seguir, onde cada fase percorre a rede em um sentido.

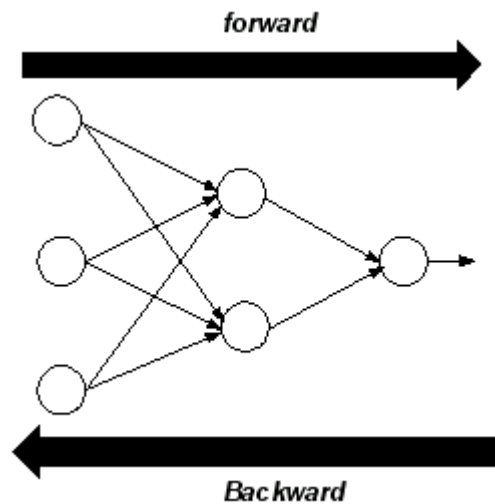


Figura 3.10 – Treinamento do algoritmo back-propagation.

Segundo Haykin [24], a atualização seqüencial de pesos é o método preferido para uma implementação em tempo de execução do algoritmo de back-propagation. Este algoritmo pode ser resumido em:

- Apresenta-se à rede um exemplo e verifica-se a saída;
- Obtém-se o erro do processo através da comparação entre a saída obtida e a saída esperada;
- Calcula-se o gradiente deste erro com relação aos valores sinápticos da camada de saída;
- A partir deste erro, atualiza-se a camada de saída;
- Calcula-se o erro da saída da penúltima camada e assim por diante.
- Apresenta-se outro exemplo à rede e repetem-se os passos anteriores;
- Continua-se o processo até que o erro seja menor que uma tolerância estabelecida.

A limitação deste algoritmo encontra-se no fato de que as funções de ativação devem ser obrigatoriamente deriváveis.

3.7 – REDES NEURAIIS TEMPORAIS

São redes dinâmicas dotadas de memória temporal na camada de entrada, permitindo-lhes a realização de tarefas de caráter temporal. Podem ser de dois tipos, dependendo da maneira como a memória é incorporada à rede:

- Redes com realimentação ou recorrentes;
- Redes com atraso no tempo.

3.7.1 – REDES RECORRENTES

Segundo Hopfield [31], redes recorrentes são aquelas que, além de apresentarem ciclos, envolvem também neurônios dinâmicos. Haykin [24] a definiu como uma rede que possui pelo menos um laço de realimentação. Estas redes possuem a vantagem de serem mais gerais e potencialmente mais poderosas que as redes *feedforward*

A figura 3.11 ilustra um exemplo de rede recorrente;

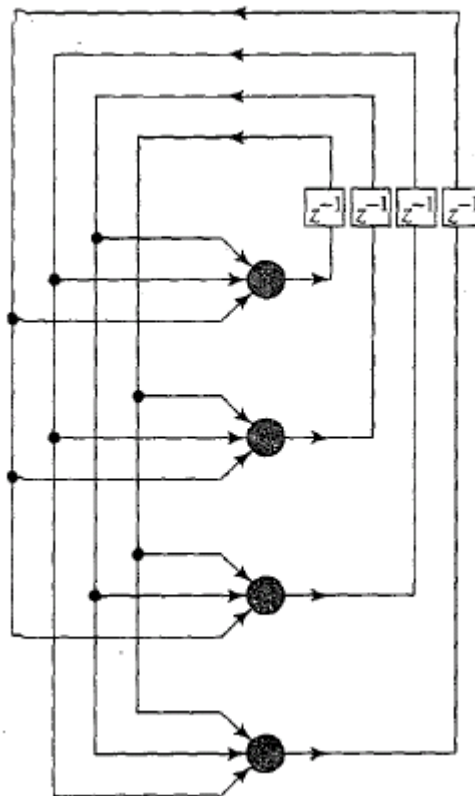


Figura 3.11: Exemplo de rede recorrente sem loop e sem camada escondida

Fonte: Haykin [24], Extraído.

onde Z^{-1} são operadores de atraso unitário, responsáveis por fornecer à rede alterações relacionadas aos períodos em que os dados lhe são apresentados.

3.7.1.1 – Rede Jordan

Desenvolvida por Jordan [32,33], esta foi uma das primeiras redes recorrentes. Um exemplo por ser visto na figura 3.12 a seguir.

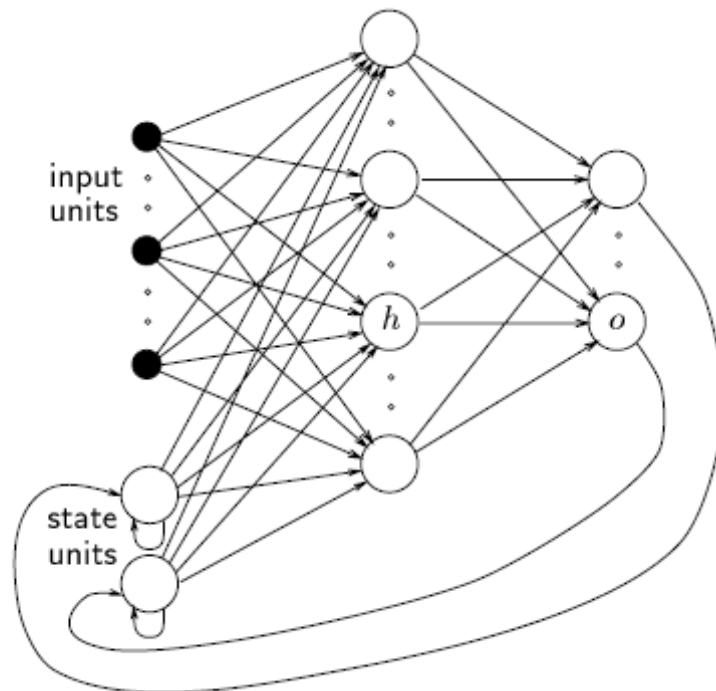


Figura 3.12 – Exemplo de Rede Jordan

Fonte: Krose, Smargt [34], Extraído.

Numa rede Jordan, as saídas são realimentadas na camada de entrada através de um conjunto extra de entradas, chamados *unidades de estado*. O numero de unidades de estado é igual ao numero de saídas na rede. As conexões entre a saída e as unidades de

estado têm um peso fixo igual a +1. A aprendizagem ocorre apenas nas conexões entre a entrada e as camadas escondidas e entre as camadas escondidas e a camada de saída.

3.7.1.2 – Rede Elman

Nesta modelo de rede, introduzido por Elman [35], o conceito de *camadas de contexto* é introduzido. Camadas de contexto são entradas que são realimentadas para as camadas escondidas. Esta rede é muito similar à Rede Jordan, exceto pelo fato da realimentação ocorrer na camada escondida e por que os as entradas extras não possuem conexão entre si, como pode ser visto na figura 3.13 a seguir:

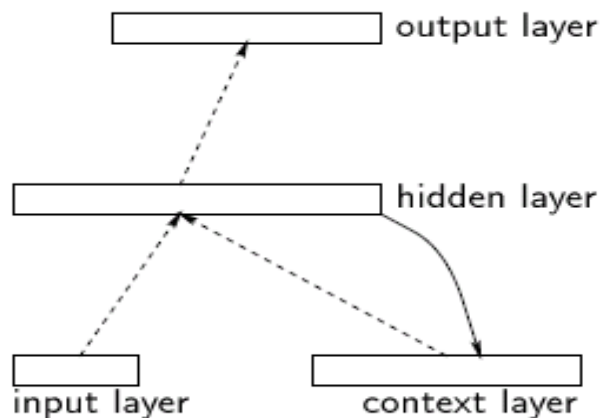


Figura 3.13 – Rede Elman

Fonte: Krose, Smargt [34], Extraído.

3.7.2 – REDES COM ATRASO NO TEMPO

São redes as quais se introduz atrasos, proporcionando aos neurônios valores de entrada atuais e valores temporariamente anteriores a eles. Os atrasos podem ser inseridos tanto na camada de entrada quanto nas camadas escondidas e de saída. Neste contexto, as redes TLFN (Time Lagged Feedforward Network) são as mais utilizadas.

As redes TLFN consistem de uma rede cuja memória é introduzida nos neurônios da camada de entrada, fornecendo-se valores das entradas atuais e também em tempos anteriores. Desta forma, a resposta da rede neural no tempo t é baseada nas entradas no tempo $(t-1), (t-2), \dots, (t-n)$.

A Figura 3.14 a seguir mostra um exemplo de rede TLFN.

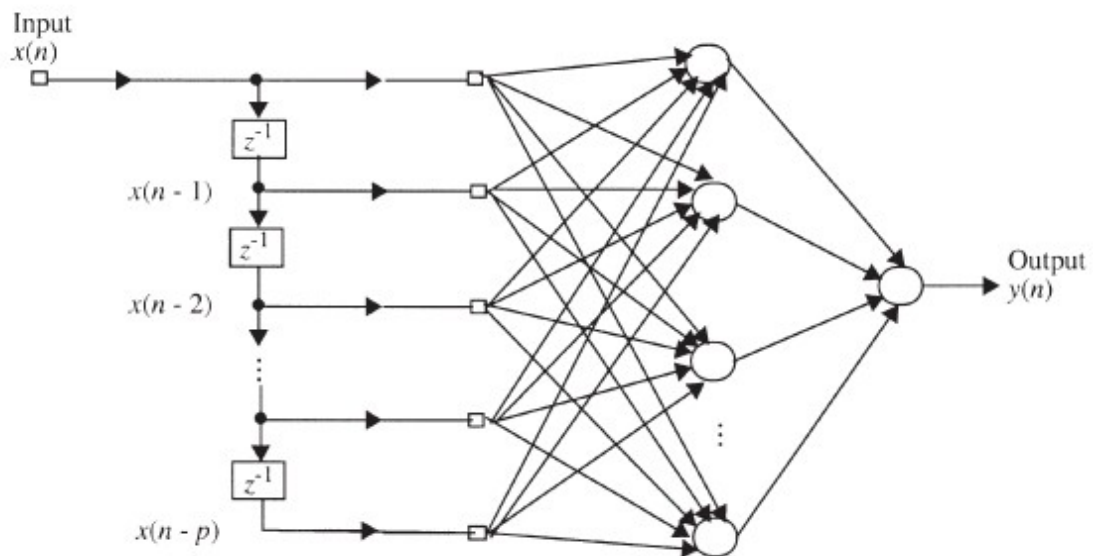


Figura 3.14 – Rede TLFN
Fonte: Haykin [24], Extraído.

4 – METODOLOGIA

4.1 – INTRODUÇÃO

Neste capítulo são apresentados a metodologia e o desenvolvimento de uma rede neural TLFN usada para o estudo da previsão do comportamento do IBOVESPA.

4.2 – DADOS DE ENTRADA E SAÍDA

4.2.1 – OBTENDO OS DADOS DA SÉRIE HISTÓRICA

Os dados de entrada deste trabalho consistem na série histórica do IBOVESPA no período de 27/06/1997 a 09/05/2008, obtidos do site do *Yahoo! Finance* [39].

O site do *Yahoo! Finance* disponibiliza os dados em um arquivo .csv. Desta forma, para facilitar a obtenção e o posterior tratamento dos dados de entrada, foi desenvolvido um script para *Matlab* (gentilmente cedido pelo laboratório de Processamento de Sinais da Universidade Federal do Rio de Janeiro). Tal script tem como parâmetros de entrada a data de início e a data de término do período a ser observado, além do nome da ação que se deseja obter a série (no caso deste trabalho, a entrada foi '27061997', '10052008', '^BVSP'). Como saída, esta aplicação insere os dados obtidos em uma *structure* do próprio *Matlab* e em uma planilha do Excel (arquivo .xls).

O arquivo .xls gerado será utilizado, posteriormente, na entrada da rede neural. Este arquivo consiste em uma planilha com os dados do comportamento do índice dia-a-dia. A figura 4.1 seguir mostra parte da planilha utilizada.

	A	B	C	D	E
1	Data	Abertura	Maximo	Minimo	Fechamento
2	9/5/2008	69722	69722	68764	69646
3	8/5/2008	69021	69877	69014	69722
4	7/5/2008	70201	70545	68599	69018
5	6/5/2008	70175	70215	69561	70195
6	5/5/2008	69366	70435	69366	70175
7	2/5/2008	67869	70973	67865	69366
8	30/4/2008	63826	68038	63826	67868
9	29/4/2008	65678	65678	63706	63826
10	28/4/2008	65191	66025	65191	65678
11	25/4/2008	64576	65253	64172	65187
12	24/4/2008	64946	65086	64223	64576
13	23/4/2008	65413	65689	64914	64948
14	22/4/2008	64922	65689	64922	65413
15	21/4/2008	64554	65384	64554	64923
16	18/4/2008	64554	65384	64554	64923
17	17/4/2008	64150	64945	63599	64552
18	16/4/2008	62625	64314	62625	64152
19	15/4/2008	62151	62825	62151	62618
20	14/4/2008	62592	62733	61467	62153
21	11/4/2008	63526	63526	62459	62585
22	10/4/2008	63475	63646	62712	63527
23	9/4/2008	64540	64621	63167	63477
24	8/4/2008	64180	64833	63453	64540
25	7/4/2008	64447	65409	63919	64176
26	4/4/2008	64177	64630	63906	64446
27	3/4/2008	63367	64731	62793	64175
28	2/4/2008	62775	63816	62775	63364

Figura 4.1 – Parte da planilha gerada pelo script e utilizada na entrada da RN.

Como pôde ser observado, a planilha indica, além da *data*, os valores de *abertura*, *máximo*, *mínimo* e *fechamento* correspondentes a cada dia da série histórica.

4.2.2 – VARIÁVEIS DE ENTRADA DA RN

Como entrada da RN utilizou-se os dados de *abertura*, *máximo*, *mínimo* e *fechamento* na ordem inversa a que aparecem na tabela, ou seja, começando da data mais antiga até a data mais recente. Isto se deve ao fato da Rede Neural trabalhar com dados seqüenciais, ou seja, a disposição dos dados influi na maneira como a rede se comporta.

Como não foi possível obter todos os dados da variável Volume para todo o período de observação, decidiu-se não utilizá-la como entrada da RN.

A Figura 4.2 a seguir mostra o gráfico das variáveis de entrada para o período de estudo.

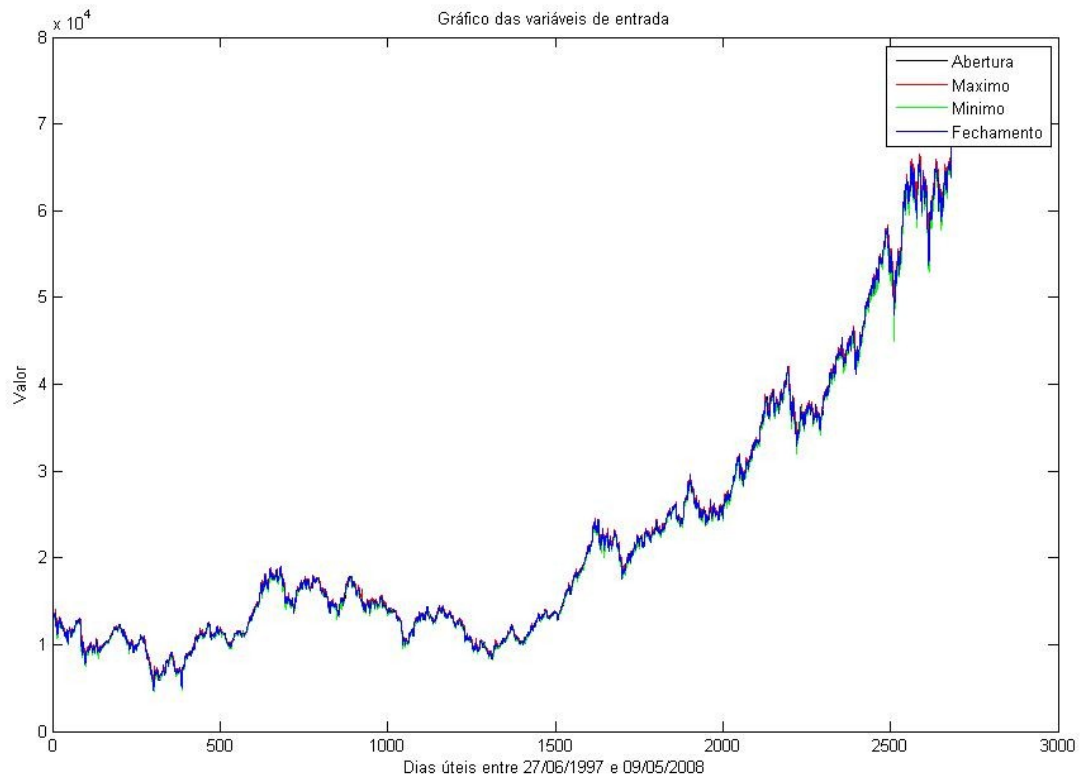


Figura 4.2 – Gráfico das variáveis de entrada da RN.

Para uma melhor visualização, a figura 4.3 a seguir mostra o gráfico da variável *Fechamento* durante o período de estudo.

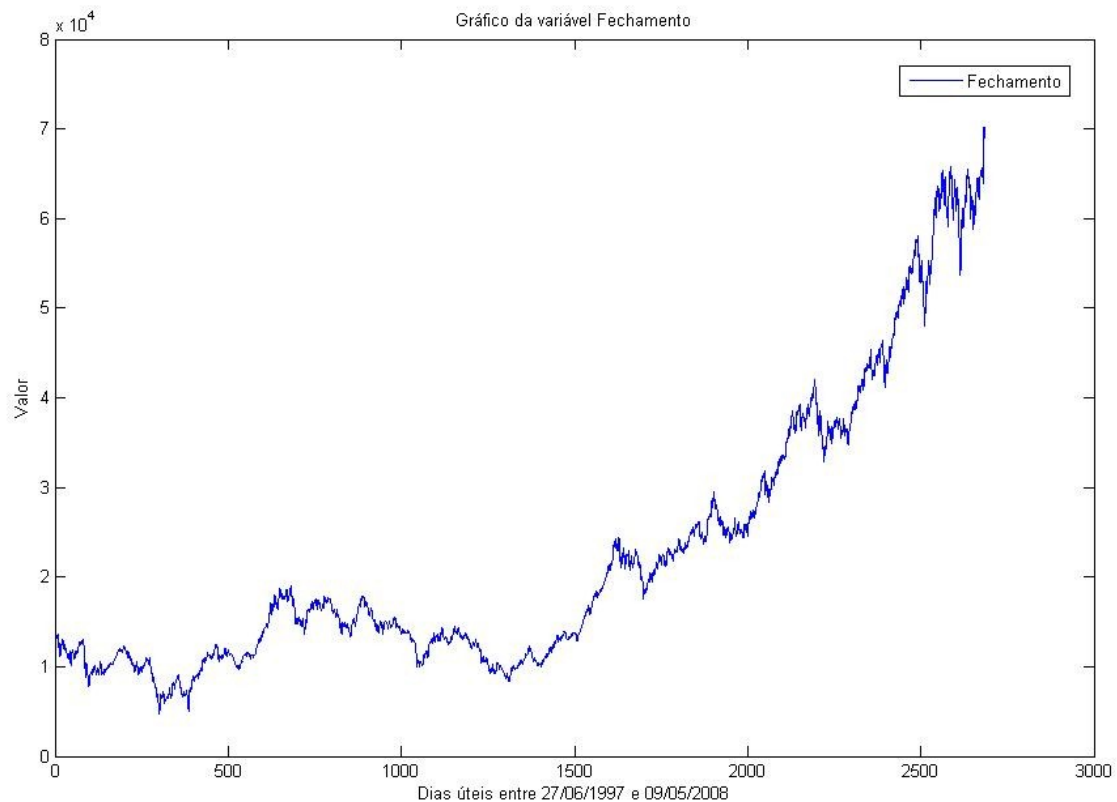


Figura 4.3 – Gráfico da variável *Fechamento*.

4.2.3 – PRÉ-PROCESSAMENTO DAS VARIÁVEIS DE ENTRADA DA RN

Os dados de entrada foram submetidos a um pré-processamento antes de serem apresentados à rede. Este pré-processamento constituiu-se da normalização das variáveis de entrada, cujo objetivo era diminuir a influência causada por valores que se destacam excessivamente em relação aos outros, ou seja, diminuir a distância entre os valores de variáveis muito espaçadas. Para isso, foram utilizadas duas funções da biblioteca do Matlab: *prestd* e *trastd*.

A função *prestd* tem por objetivo normalizar as entradas e saídas (*targets*) de forma que elas possuam média 0 (zero) e desvio-padrão 1 (um). Esta função retorna um vetor com as médias normalizadas e outro com os targets normalizados, ambos com média zero e desvio-padrão igual a 1 (um). Também retorna dois vetores com as médias e desvios-padrão da matriz de entrada original e dois vetores contendo as médias e

desvios-padrão dos targets originais. Após o treinamento da rede, esses vetores são utilizados na transformação de quaisquer outros vetores de entrada, tornando-se, assim, parte integrante da rede, tal como os pesos.

Após o pré-processamento dos dados de treinamento, toda vez que a rede for utilizada com novas entradas é necessário que elas também sejam pré-processadas com as médias e desvios-padrão que foram calculados para o conjunto de treinamento. Isto é realizado com o emprego da função *trastd*, utilizada para o conjunto de testes da RN.

Por fim, para realizar a conversão dos valores de saída para as mesmas unidades dos valores de entrada foi utilizada a função *poststd*.

4.2.4 – VARIÁVEL DE SAÍDA DA RN

Neste trabalho, escolheu-se como saída a ser obtida pela rede neural a variável *Fechamento* no dia posterior ao dia de observação. De acordo com a definição de proporção do conjunto dos valores de entrada, visto mais adiante em 4.3.3, o período de previsão vai do dia 10/03/2006 até 09/05/2008.

4.3 – A REDE NEURAL

4.3.1 – INTRODUÇÃO

A utilização de redes neurais impõe a escolha de alguns parâmetros na definição do modelo que mais se aproxime da resposta ideal. Neste trabalho, buscou-se o desenvolvimento de um algoritmo capaz de, por si só e após sucessivos testes, escolher a rede com a resposta mais satisfatória possível. Tal algoritmo foi desenvolvido e implementado com o auxílio do *Matlab*.

4.3.2 – ARQUITETURA DA REDE

Para este trabalho, a arquitetura de rede escolhida foi a TLFN, freqüentemente utilizada em aplicações financeiras. Como já visto anteriormente, este tipo de rede feedforward permite como entrada valores atuais e também atrasados no tempo. Será mostrado mais adiante que a escolha deste modelo possibilitou a implementação de um algoritmo mais robusto no que diz respeito à otimização da rede.

A rede escolhida foi modelada em 3 camadas, ou seja, uma camada de entrada, uma camada oculta e outra de saída. Esta escolha seguiu a recomendação de Fausett [34]. Segundo ele, duas ou mais camadas ocultas podem ser benéficas em algumas aplicações, porém, uma única camada oculta é considerada suficiente.

Para implementação da rede foram utilizadas duas funções de ativação diferentes, ambas encontradas nas bibliotecas do *Matlab*: *tan-sigmoid* e *purelin*. Como função de ativação para os neurônios da camada oculta, uma função sigmoideal foi escolhida: *tan-sigmoid*. Esta função não-linear, com range entre -1 e 1, é muito utilizada em redes multicamadas e pode ser expressa de acordo com a equação 4.1 a seguir:

$$\varphi(v) = \frac{2}{1 + e^{-2v}} - 1, \quad (4.1)$$

onde v é a soma dos pesos das entradas.

Esta função é matematicamente equivalente a $\tanh(v)$, porém é mais rapidamente implementada no *Matlab*. Seu gráfico é mostrado na figura 4.4 a seguir.

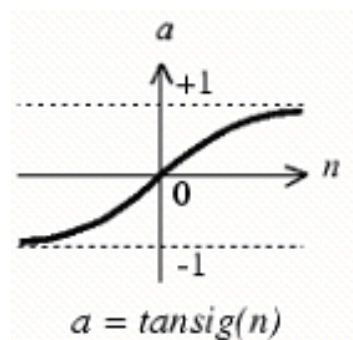


Figura 4.4 – Gráfico da função tan-sigmoid.

Fonte: *Matlab*, documentação.

Para a camada de saída, a função aplicada foi a *purelin*. Trata-se de uma função linear com range entre infinito e cujo gráfico é mostrado na figura 4.5 a seguir:

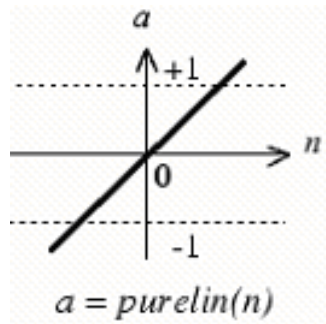


Figura 4.5 – Gráfico da função *purelin*.

Fonte: *Matlab*, documentação.

A função performance utilizada neste trabalho foi a função *erro quadrático médio* (MSE). Esta função, também muito utilizada neste tipo de aplicação, tem por objetivo medir o desempenho da rede através do cálculo de seu erro médio, de acordo com a equação 4.2 a seguir:

$$MSE(y) = E((T - y)^2) \quad (4.2)$$

onde T = valor a ser alcançado (*target*);

y = valor da saída da rede.

O algoritmo desenvolvido permitiu, também, a definição do número de redes que seriam utilizadas para a realização dos testes. Convencionou-se que este número seria igual a 20 (vinte). Assim, durante o processo de criação de redes, o algoritmo criou um total de 20 redes para cada conjunto de janelamento (ver 4.3.3). Desta forma, como

foram definidas 5 janelas para cada conjunto, o total de redes treinadas, testadas e validadas foi igual a 100.

O número de neurônios na camada escondida foi sendo alterado e testado à medida que novas iterações eram feitas. O valor do número inicial de neurônios da camada escondida foi definido, primeiramente, como sendo a média entre o número de entradas e o número de saída, de acordo com a equação 4.3 a seguir:

$$N_H = \frac{N_E + N_S}{2} \quad (4.3)$$

onde N_H = nº inicial de neurônios da camada escondida;

N_E = nº de neurônios na camada de entrada;

N_S = nº de neurônios na camada de saída;

Desta forma, devido à configuração da melhor rede escolhida pelo script (mostrada no Capítulo 5), de início o algoritmo começou a ser executado com um número de neurônios igual a 32 (trinta e dois) e foi configurado para incrementar em 10 (dez) esse número a cada vez que o ciclo de *janelamento* de entrada (explicado no tópico seguinte) fosse completado.

Entretando, este cálculo do número de neurônios da camada escondida foi modificado na segunda configuração utilizada para encontrar a melhor rede neural. Ao invés da média, optou-se por definir o número inicial de neurônios da camada escondida como sendo apenas a soma entre o número de entradas e o número de saídas, de acordo com a equação 4.4. Assim, foi possível verificar o desempenho da rede também para este parâmetro (número de neurônios).

$$N_H = N_E + N_S \quad (4.4)$$

onde N_H = n° inicial de neurônios da camada escondida;

N_E = n° de neurônios na camada de entrada;

N_S = n° de neurônios na camada de saída;

Outro parâmetro importante foi o número de épocas, que consistia no número máximo de iterações que cada rede deveria fazer. Como não foi encontrado em qualquer bibliografia um método ou modelo para definir o valor deste parâmetro, optou-se, inicialmente, por escolher um valor bem acima do normalmente empregado em outros trabalhos, como, por exemplo, o de Mello [35], no qual o número de épocas variava entre 20 (vinte) e 100 (cem). Com isso, definiu-se, inicialmente, o número de épocas deste trabalho em 30000 (trinta mil). Esta escolha, como visto no Capítulo 5, não produziu um resultado considerado satisfatório e uma outra configuração se fez necessária., desta vez utilizando 5000 (cinco mil) épocas.

4.3.3 – INICIALIZAÇÃO DA REDE

Após a obtenção da planilha com os dados das variáveis correspondentes a cada dia de observação, estes foram separados na seguinte proporção: 70% para treinamento da rede (dados de 11/07/1997 a 10/02/2005), 10% para validação (dados de 11/02/2005 a 09/03/2006) e 20% de teste (dados de 10/03/2006 a 09/05/2008).

O algoritmo para implementação da RN foi desenvolvido com a capacidade de variar a janela de entrada dos dados. Como descrito anteriormente, as entradas da rede consistiam nas variáveis *Abertura*, *Máximo*, *Mínimo* e *Fechamento*, referentes a cada dia do período de estudo. Proporcionado pela escolha da arquitetura TLFN, foi possível desenvolver um algoritmo cuja entrada de dados podia ser variada entre os dias D-3 e D-7. Desta maneira, após testar a entrada da rede considerando os três dias imediatamente anteriores ao dia a ser previsto (D-3), o algoritmo passava então a considerar os quatro dias anteriores ao dia da previsão (D-4), e assim por diante, até o dia D-7. Este processo,

chamado algumas vezes de ‘*janelamento*’, juntamente com a variação do número de neurônios da camada escondida, permitiu ao algoritmo, ao final da execução, apontar a rede com a melhor configuração.

Devido à utilização deste processo de ‘*janelamento*’, uma pequena adaptação em parte dos dados de entrada foi necessária. Os dados de entrada foram utilizados a partir do 8º (oitavo) dia a contar da data inicial. Por exemplo, a janela D-3 utiliza como entrada os 3 (três) dias anteriores a esta. Porém, no início do período de observação estes dias não existem. O mesmo ocorre para as outras janelas. Desta forma, optou-se por utilizar os dados a partir do 8º (oitavo) dia, quando todas as janelas poderiam ser efetivamente atendidas.

4.3.4 – TREINAMENTO, VALIDAÇÃO E TESTE

O algoritmo de treinamento utilizado foi o *trainrp*, próprio do *Matlab*. Trata-se de um algoritmo que atualiza os pesos de acordo com um algoritmo de *backpropagation*.

Como já mencionado, redes multicamadas geralmente utilizam funções de transferência sigmoidais em suas camadas escondidas. Devido ao fato dessas funções possuírem um range entre -1 e 1, à medida que os valores de entrada aumentam, a derivada se aproxima de zero. Desta maneira, ao treinar redes desse tipo, o gradiente acabar por possuir valores muito pequenos e, com isso, apenas pequenas mudanças nos pesos são efetuadas, mesmo que estes ainda estejam muito distantes dos seus valores ideais.

A utilização do algoritmo *trainrp* se dá com o intuito de eliminar esses efeitos. Com este algoritmo, o sinal da derivada é utilizado para determinar somente a direção em que o peso deve ser atualizado, e não o seu valor, o qual é atualizado por um valor em separado. O valor a ser alterado para cada peso é acrescido de um fator Δ_{INC} sempre que a derivada da *função performance* tiver o mesmo sinal por duas iterações sucessivas. O valor do peso é decrescido por um fator Δ_{DEC} sempre que o sinal da derivada se alterar entre uma iteração e outra. Caso a derivada seja zero, o valor do peso não se altera.

A definição do valor de validação seguiu as recomendações gerais para este tipo de parâmetro, sendo definido, inicialmente, com o valor de 25%. Porém, como aconteceu com o número inicial de épocas, esta configuração não teve um resultado considerado satisfatório e acabou, também, por ser alterada. Definiu-se então em 10% o valor do parâmetro de validação. Desta forma, o algoritmo, durante o processo de validação, verificava se o MSE da validação aumentou em relação ao erro mínimo estabelecido. Caso este aumento ocorresse em 10% do total de épocas, o algoritmo finalizava o processo de treino e validação da rede correspondente, exibindo a mensagem ‘*Validation Stop*’.

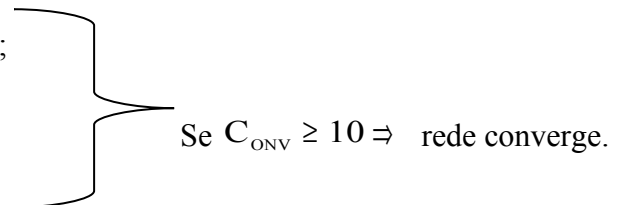
Um teste para verificar a convergência da rede também foi realizado. Ao fim do processo de treino e validação de cada rede, uma comparação entre o 1º e o último MSEs da rede é efetuada, de acordo com a equação 4.5 a seguir:

$$C_{ONV} = \frac{MSE_1}{MSE_N} \quad (4.5)$$

onde, C_{ONV} = convergência da rede;

MSE_1 = 1º MSE da rede;

MSE_N = último MSE da rede.



Após os processos de treinamento e validação, cada rede é, então, submetida ao conjunto de teste. Este, como mencionado em 4.3.3, consiste em 20% dos dados da planilha com a série histórica. Os testes são realizados para cada configuração de rede, ou seja, cada rede com sua respectiva janela de entrada é submetida ao processo de teste. Isto quer dizer que, no presente trabalho, foram realizados teste em 100 (cem) redes diferentes.

Ao término do processo de testes o algoritmo aponta a melhor configuração de rede, baseado no menor MSE de teste obtido.

4. 4 – MÉTRICAS PARA ANÁLISE DOS RESULTADOS

Para avaliação do modelo de previsão utilizado neste trabalho foram empregados dois métodos distintos: cálculo da taxa de acerto de direção e cálculo do percentual de acertos da saída da rede em relação ao valor original.

4.4.1 – CÁLCULO DA TAXA DE ACERTO DE DIREÇÃO

O cálculo da taxa de acerto de direção procurou medir se a rede neural fora capaz de prever com sucesso as tendências do mercado no período de teste. Desta maneira, buscou-se comparar as tendências de saída do sistema de previsão proposto com as tendências reais apresentadas pelo índice IBOVESPA. Isto se deu calculando as diferenças entre dois dias consecutivos de ambos os casos e comparando os sinais entre eles. Caso ambos tivessem sinais iguais, significava que a rede tinha acertado a previsão.

O algoritmo a seguir ilustra o método de cálculo descrito:

INÍCIO

$$\text{Tendencia_Saida}_N = \text{Saida}_{N+1} - \text{Saida}_N$$

$$\text{Tendencia_Real}_N = \text{Real}_{N+1} - \text{Real}_N$$

$$\text{Sinal_Tendencia_Saida}_N = \frac{\text{Tendencia_Saida}_N}{|\text{Tendencia_Saida}_{N+1}|}$$

$$\text{Sinal_Tendencia_Real}_N = \frac{\text{Tendencia_Real}_N}{|\text{Tendencia_Real}_{N+1}|}$$

Se ($\text{Sinal_Tendencia_Saida}_N == \text{Sinal_Tendencia_Real}_N$)

então (*acertou_tendencia*)

senão (*errou_tendencia*)

FIM

onde: Saida_N é o valor da variável fechamento calculado pela Rede Neural para o dia N;

$Saida_{N+1}$ é o valor da variável fechamento calculado pela Rede Neural para o dia N+1;

$Real_N$ é o valor real da variável fechamento para o dia N;

$Real_{N+1}$ é o valor real da variável fechamento para o dia N+1.

4.4.2 – CÁLCULO DO PERCENTUAL DE ACURÁCIA DA SAÍDA DO SISTEMA

Outro método para medição dos resultados do sistema proposto foi o cálculo do percentual de acurácia da saída da rede em relação ao valor original. Como já informado, a rede neural desenvolvida tem como saída a previsão dia-a-dia da variável fechamento. Desta forma, comparou-se o número de vezes em que a rede neural fora capaz de ‘*acertar*’ o valor da variável fechamento considerando uma margem de erro que variava entre $\pm 1\%$ e $\pm 5\%$.

5 – APLICAÇÃO E RESULTADOS

5.1 – INTRODUÇÃO

Este capítulo apresenta a análise dos resultados obtidos no desenvolvimento deste trabalho e está dividido em três partes. Primeiramente, são mostrados os resultados de uma rede neural cuja configuração *não* apresentou um resultado considerado satisfatório e, por isso, fora descartada. Depois, uma segunda rede neural é apresentada, esta, sim, com um resultado satisfatório; atendendo, dessa maneira, o objetivo principal deste trabalho. Por fim, é feita uma breve análise dos resultados desta rede, com o intuito de validar os dados obtidos com a rede neural.

5.1 – REDE NEURAL COM CONFIGURAÇÃO NÃO SATISFATÓRIA

O primeiro conjunto de redes neurais desenvolvido para este trabalho não foi considerado eficiente na previsão da variável *Fechamento*. Os dados de entrada foram os mesmo mencionados no capítulo 4, porém a resposta da rede não foi satisfatória. A tabela 5.1 a seguir contém os parâmetros utilizados para treinamento, validação e teste de tal conjunto:

Parâmetro	Valor
Nº de redes	20
Saídas	1
Janelas	D-3 a D-7
Total de redes	100
Nº. inicial de neurônios na camada escondida	17

Incremento nº de neurônios da camada escondida a cada nova iteração	10
No de épocas	30000
Erro mínimo	10^{-6}
Validação	25%
Algoritmo de treinamento	trainrp
Funções de ativação	tan-sigmoid, purelin

Tabela 5.1 – Parâmetros de configuração do 1º conjunto de redes desenvolvido.

Como pôde ser observado da tabela anterior, o número de épocas escolhido foi bastante elevado, juntamente com o valor de validação. Esta escolha deveu-se à crença de que o algoritmo sempre seria capaz de diminuir o erro à medida que novas iterações fossem feitas. Da mesma forma, acreditava-se que a rede pudesse atingir ou chegar próxima ao erro escolhido (10^{-6}).

Ao final de todo processo de teste, o script apontou como melhor rede àquela com as características mostradas na tabela 5.2 a seguir.

Nº neurônios camada escondida	67
MSE	0.09726
Janela	D-6
MSE médio de teste	12.2035
Desvio padrão do MSE de teste	32.7198

Tabela 5.2 – Características da melhor rede.

O log deste processo encontra-se em um arquivo anexo a este documento.

Porém, esta configuração mostrou-se equivocada, produzindo um resultado insatisfatório, como pode ser visto na análise do gráfico de saída da rede, mostrado na figura 5.1.

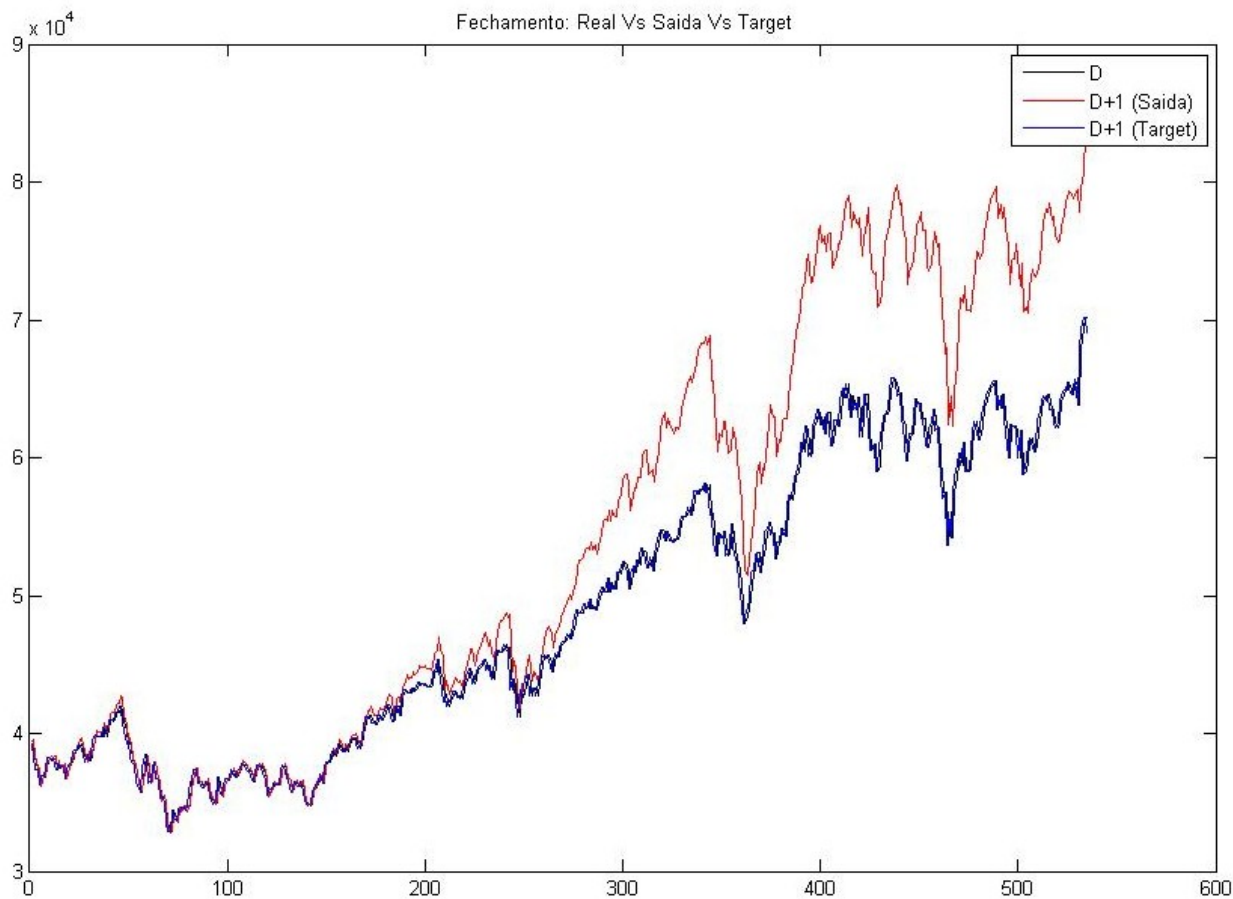


Figura 5.1 – Gráfico de saída da rede cujo resultado não foi satisfatório.

A partir do gráfico anterior percebe-se que a rede não retornou uma saída satisfatória, principalmente a partir do ducentésimo dia, aproximadamente. A escolha de um elevado número de épocas e de um alto valor de validação implicou um aumento do MSE, ou seja, a rede tornou-se muito tolerante a erros.

Outro fator que também contribuiu para o mau resultado foi a escolha de um erro mínimo com valor muito baixo. Isto acabou por fazer com que a rede buscasse sempre esse valor desnecessariamente pequeno com a intenção de otimizar o resultado. Entretanto, o que se viu foi um aumento do MSE, o que acabou por deteriorar a saída da rede.

5.2 – RESULTADOS DA REDE NEURAL COM CONFIGURAÇÃO SATISFATÓRIA

Após o desenvolvimento do primeiro conjunto de RN, percebeu-se que alguns parâmetros deveriam ser alterados de forma a melhorar a saída do sistema. Desta maneira, optou-se pela configuração mostrada na tabela 5.3.

Parâmetro	Valor
Nº de redes	20
Saídas	1
Janelas	D-3 a D-7
Total de redes	100
Nº. inicial de neurônios na camada escondida	33
Incremento nº de neurônios da camada escondida a cada nova iteração	10
No de épocas	5000
Erro mínimo	10^{-3}
Validação	10%
Algoritmo de treinamento	trainrp
Funções de ativação	tan-sigmoid, purelin

Tabela 5.3 – Parâmetros de configuração do 2º conjunto de RNs.

Com a configuração anterior procurou-se diminuir o MSE de saída da rede a partir da alteração do número inicial de neurônios na camada escondida, do número de épocas, do erro mínimo e do valor de validação.

Baseado nas configurações do conjunto de redes anterior a este, decidiu-se por diminuir o número de épocas. Isto fez com que o novo conjunto de redes tivesse, em geral, desempenho superior ao conjunto anterior em relação ao MSE de teste.

O erro mínimo a ser obtido pela rede também teve seu valor alterado em relação ao conjunto anterior. Acreditava-se que o valor anterior (10^{-6}) era desnecessariamente baixo, tornando-se, assim, um parâmetro desprezível pela rede, já que dificilmente seria alcançado. Na nova configuração, optou-se por um valor acima do anterior, porém ainda bastante aceitável para este tipo de aplicação. Desta maneira, escolheu-se o valor 10^{-3} para este parâmetro.

O valor de validação foi outro parâmetro alterado nesta nova configuração. Com o valor anterior (25%), a rede mostrou-se muito tolerante a erros. Com isso, ao substituí-lo pelo valor de 10%, procurou-se diminuir tal tolerância, com a qual a rede não havia apresentado bons resultados anteriormente.

O número inicial de neurônios na camada escondida também foi alterado, passando de 17 para 33. Esta alteração deveu-se à observação do resultado da primeira configuração da rede; através dela percebeu-se que a melhor rede escolhida pelo script possuía um número de neurônios na camada escondida muito superior ao valor inicial. Concluiu-se, assim, que seria razoável começar o novo treinamento com um valor mais elevado de neurônios na camada escondida.

Com tais configurações, a melhor rede escolhida pelo script teve as seguintes características, mostradas na tabela 5.4 a seguir:

Nº neurônios camada escondida	93
MSE	0.041396
Janela	D-7
MSE médio de teste	7.3575.
Desvio padrão do MSE de teste	11.4235

Tabela 5.4 – Características da melhor rede.

O log deste processo encontra-se em um arquivo anexo a este documento.

O gráfico de saída pode ser visto na figura 5.2. Nele encontram-se duas curvas da variável fechamento: a real e a prevista pela rede neural.

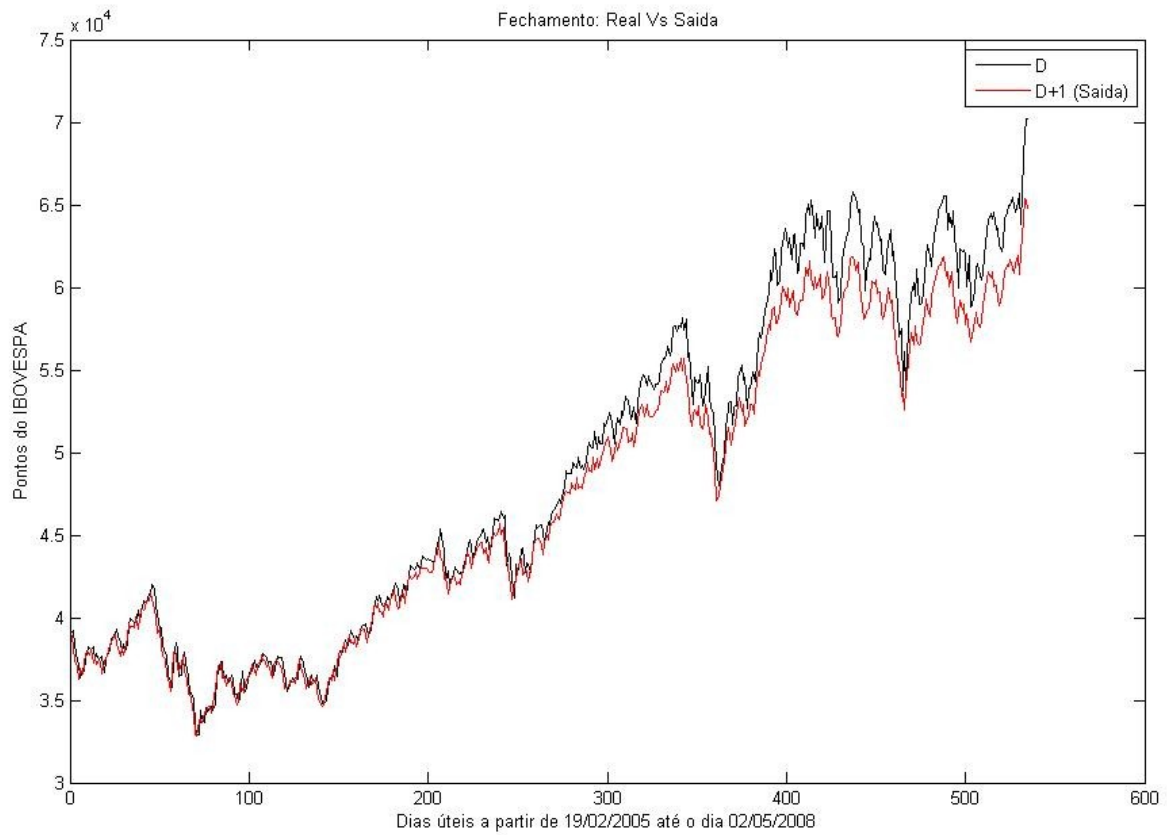


Figura 5.2 – Gráfico de saída da melhor rede.

A figura 5.3 a seguir mostra o mesmo gráfico, porém com um período de 150 dias, para melhor visualização.

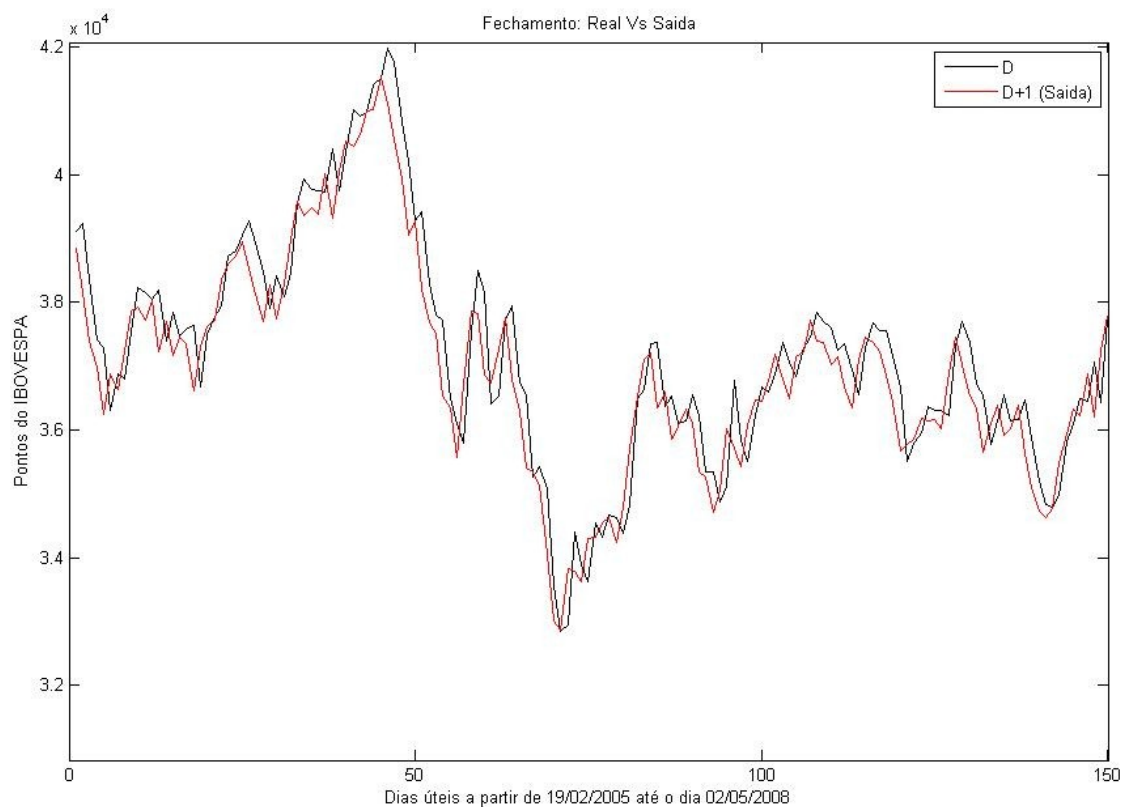


Figura 5.3 – Gráfico de saída da melhor rede com range de 150 dias.

Pode-se perceber pelos dois gráficos anteriores que a rede neural conseguiu realizar uma previsão altamente satisfatória da variável *Fechamento*, principalmente nos trezentos primeiros dias, quando o erro da rede foi menor. A partir daí, a rede neural teve mais dificuldade em realizar a previsão, o que pode ser visto pelo aumento da distância entre os pontos da curva de previsão e a curva real. Apesar disso, o resultado final continuou satisfatório, como será mostrado na seção seguinte.

Vale ressaltar que nesta configuração percebeu-se, também, uma diminuição do tempo de processamento do script em torno de 10 vezes em relação ao primeiro conjunto de redes.

O resultado aqui obtido supera o resultado obtido por Almedia e Zanetti Jr [36]. Ao analisar um período compreendido entre os anos de 1994 e 1998 utilizando o método de aprendizado Vanilla Backpropagation (ou Online Backpropagation) para prever o comportamento diário das ações da Telebrás, estes autores não obtiveram sucesso.

Segundo os mesmos, na conclusão do próprio trabalho, o “(...) método de avaliação e seleção da melhor rede também poderia ser alterado de forma que a melhor rede fosse aquela que apresentasse o menor MSE para o conjunto de dados de treinamento; e em certos períodos o MSE do conjunto de validação seria avaliado, a fim de determinar quando o treinamento deveria ser interrompido.”

5.2.1 – TAXA ACERTO DE DIREÇÃO E PERCENTUAL DE ACURÁCIA DA SAÍDA DO SISTEMA

Para efeitos de validação do sistema e considerando os valores da variável *fechamento* retornados pela rede, calculou-se o percentual de acurácia do modelo de previsão e a taxa de acertos de direção.

5.2.1.1 – Percentual de acurácia do modelo de previsão

Para essa análise, foi desenvolvida uma planilha em Excel com os dados de saída da rede referentes a cada dia de previsão e com os dados reais do mesmo período. Definiu-se, então, as margens de tolerância entre as quais os erros de previsão seriam tidos como aceitáveis; estas foram $\pm 1\%$, $\pm 2\%$, $\pm 3\%$, $\pm 4\%$ e $\pm 5\%$. Desta maneira, verificou-se dia-a-dia se o resultado de previsão da RN estava dentro das margens estabelecidas. Assim, considerou-se como ‘*acerto*’ os resultados de saída da rede que estavam dentro das margens estabelecidas. A figura 5.4 a seguir ilustra os resultados obtidos.

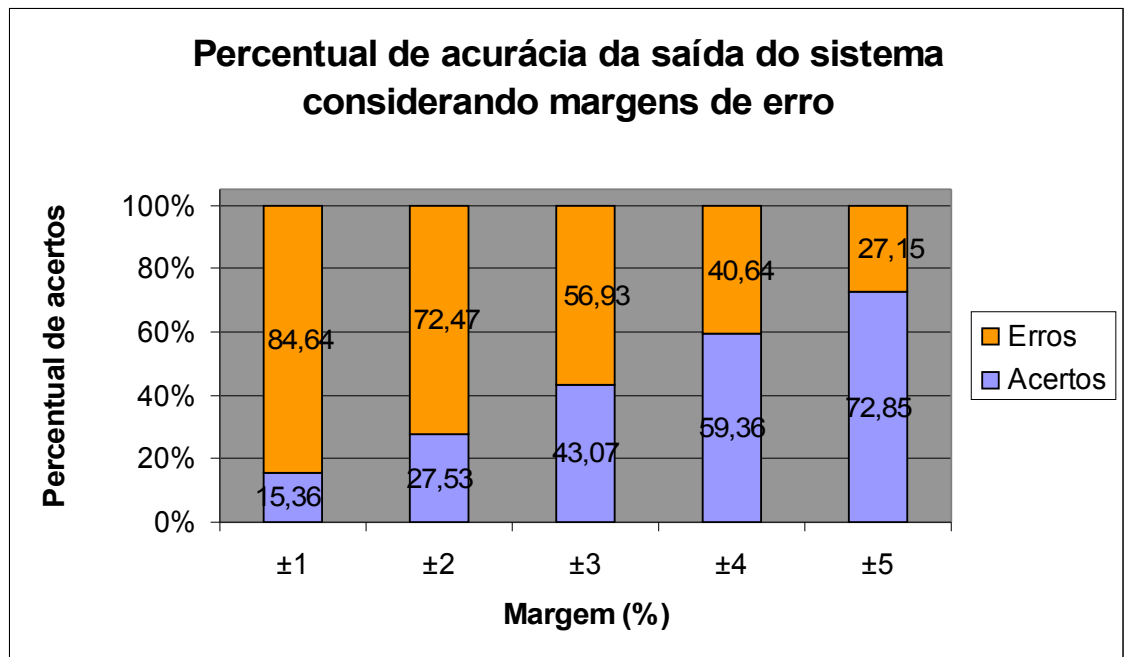


Figura 5.4 – Gráfico do percentual de acurácia da saída do sistema.

A partir do gráfico anterior é possível notar que, com uma margem entre 3% e 4%, o percentual de acertos começa a superar os 50%. Com uma margem de 5%, o percentual de acertos chega a 72,85%.

5.2.1.2 – Taxa de acerto de direção

Nesta análise buscou-se saber qual a capacidade da rede de acertar a previsão diária de tendências. A figura 5.5 a seguir ilustra o resultado obtido.

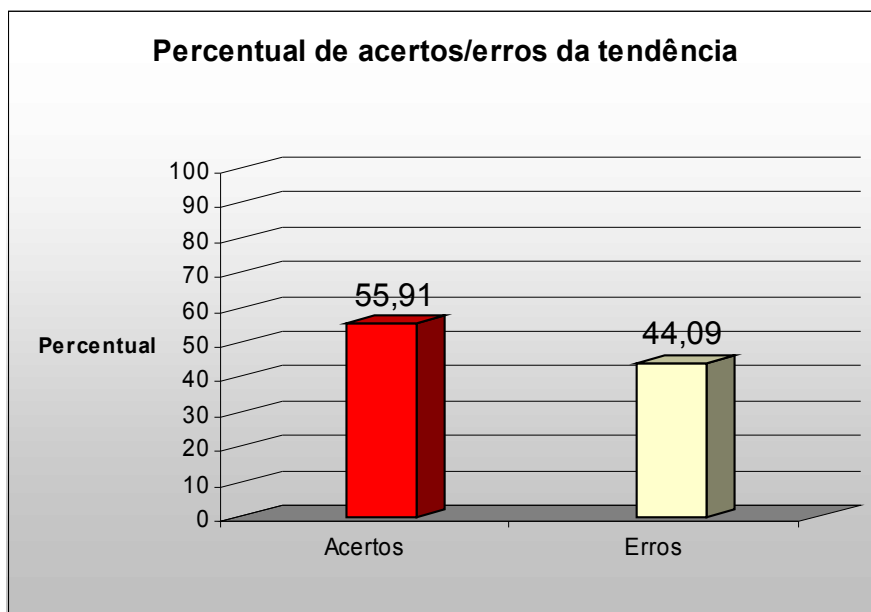


Figura 5.5 – Gráfico do percentual de acertos/erros da tendência.

A rede neural desenvolvida para este trabalho conseguiu prever a tendência correta da movimentação do índice em 55,91%. Isto significa dizer, em outras palavras, que o modelo de previsão desenvolvido tem por característica acertar a previsão de tendências na maioria dos casos.

6 – CONCLUSÕES

Este estudo apresenta um modelo baseado em redes neurais de previsão do índice da Bolsa de Valores de São Paulo. O período a ser previsto foi de 10 de março de 2006 a 9 de maio 2008. Porém, para treinamento e validação da rede foram utilizados dados de 11 de julho de 1997 a 9 de março de 2006.

Os objetivos específicos definidos no início deste trabalho foram todos atendidos:

- Foi desenvolvido um script para obtenção dos dados do IBOVESPA relativos ao período de estudo. Tal script foi fundamental para o posterior desenvolvimento do projeto, pois, ao facilitar a obtenção dos dados, permitiu uma economia de tempo que pôde ser usado em outras fases;
- A implementação da rede se deu, também, através de um script para *Matlab*. A opção por implementar a rede através de linha de comando, ao invés de utilizar uma *toolbox* específica do próprio *Matlab*, mostrou-se acertada, pois permitiu o desenvolvimento de uma lógica capaz de testar inúmeras redes e escolher a melhor entre elas. Esta opção possibilitou também a alteração de parâmetros mais facilmente, de forma a buscar a melhor configuração da rede.

O modelo proposto retornou uma saída satisfatória, tanto em termos do MSE medido entre a saída e o valor real, quanto para a medida de acurácia e da taxa de acerto de direção. Os resultados obtidos mostraram-se superiores aos de dois outros trabalhos.

Desta forma, ficou demonstrado que é possível aplicar o conceito de Redes Neurais para a previsão de séries históricas, em especial séries financeiras, e obter resultados coerentes e satisfatórios.

Há muitas possibilidades de trabalhos futuros. Uma delas é a alteração do algoritmo desenvolvido de forma a realizar previsões semanais e não diárias. Isto tornaria o sistema menos suscetível a ruídos e facilitaria a previsão de tendências.

Outra possibilidade seria o desenvolvimento de um software comercial capaz de obter os dados da série histórica e realizar as previsões para um período de tempo escolhido pelo usuário.

A alteração das configurações da rede neural também poderia ser feita, de forma a tentar melhorar os resultados obtidos. Isto se daria com a modificação dos parâmetros *número de épocas* e *valor de validação*, com a intenção de diminuir o MSE de teste e, com isso, ajudar o sistema a aumentar sua acurácia.

BIBLIOGRAFIA

- [1] ARMANO, G.; MARCHESI, M.; MURRU, A. A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences*. v. 170, p. 3-33, 2005.
- [2] HAUGEN, R. A. *Modern Investment Theory*. Upper Saddle River, New Jersey: Prentice Hall, 1997.
- [3] JENSEN, M. C. Some anomalous evidence regarding market efficiency. *Journal of Financial Economics*. v. 6, p. 95-101, 1978.
- [4] FAMA, E. F. Efficient capital markets; a review of theory and empirical work. *Journal of Finance*. 1970
- [5] DISORNTETIWAT, P. Global stock index forecasting using multiple generalized regression neural networks with a gating network. PhD Thesis, University of Missouri-Rolla, 2001.
- [6] COSTA JR, N. C. A. Sazonalidades do IBOVEPA. *Revista de Administração de Empresas*, 1990.
- [7] Desai, H., Jain, P. Long-run common stock returns following splits and reverse splits. *Journal of Business*, 1997.
- [8] DAVIDSSON, Marcus. *Stock Market Anomalies - A Literature Review and Estimation of Calendar affects on the S&P 500 index.*, 2006
- [9] LEITE, J. C. Splits: ineficiência no mercado de ações. *Revista de Administração de Empresas*, 1994.
- [10] NORONHA, Márcio. *Análise Técnica*, 2005.
- [11] DORFFNER, G. *Neural Networks for Time Series Processing*, *Neural Network World*, 1996.
- [12] Box, Jenkins G. E. *Box & G. M. Jenkins, Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, CA, 1970.
- [13] PRUDÊNCIO, Ricardo Bastos Cavalcante. *Projeto híbrido de redes neurais*. Universidade Federal de Pernambuco, 2002
- [14] Portal Capes. Site: www.portalcapes.com.br
- [15] SOUTO-MAIOR, César Duarte. *Previsão da direção de movimento de índices de ações usando um sistema fuzzy*. 2007.

- [16] TEIXEIRA, J. C.; RODRIGUES, A. J. An applied study on recursive estimation methods, neural networks and forecasting. *European Journal of Operational Research.*, 1997.
- [17] GENÇAY, R. The predictability of security returns with simple technical trading rules. *Journal of Empirical Finance*, 1998.
- [18] YU, S. Forecasting and arbitrage of the Nikkei stock index futures: an application of backpropagation networks. *Asia-Pacific Financial Markets*, 1999.
- [19] PANDA, C.; NARASIMHAN, V. Predicting stock returns: an experiment of the artificial neural network in Indian stock market. *South Asia Economic Journal*, 2006.
- [20] GOMES, F.C. Os modelos ARIMA e a abordagem de Box-Jenkins: uma aplicação na previsão do IBOVEPA a curtíssimo prazo. *Revista de Administração de Empresas*, 1989.
- [21] RIBEIRO, T. S.; SILVA, A. L. C. Do artificial neural networks provide better forecasts? Evidence from latin american stock indexes. In: *XXIX Enampad*, 2005, Brasília.
- [22] GRUDNITSKI, G.; OSBURN, L. Forecasting S&P and gold futures prices: an application of neural networks. *Journal of Futures Markets*, 1993.
- [23] FERNÁNDEZ-RODRÍGUEZ, F.; GONZÁLEZ-MARTEL, C.; SOSVILLA-RIVERO, S. On the profitability of technical trading rules based on artificial neural networks: evidence from the Madrid stock market. *Economics Letters*, 2000.
- [24] HAYKIN, Simmon –Neural Networks –A comprehensive foundation. 2nd Edition, 2005.
- [25] W.S. McCulloch and W. Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 1943.
- [26] D.O. Hebb's., *The Organization of Behavior*. New York, Wiley, 1949
- [27] ROSENBLATT, F. (1959). *Principles of Neurodynamics*. New York: Spartan Books, 1959
- [28] JONES, Edward R. *An Introduction to Neural Networks - A White Paper*. 2004
- [29] G. Cybenko. Approximation by Superpositons of a Sigmoid Function. *Mathematics of Control, Signals and Systems*. 1989.
- [30] G. Cybenko. Continuous Valued Neural Network with two Hidden Layers are Sufficient. Technical Report, Department of Computer Science, Tufts University, 1988.
- [31] HOPFIELD, J. Neural Networks and Physical Systems with Emergent Collectives Computational Abilities. *Proceedings of the National Academy of Sciences* , 1982.

- [32] JORDAN, M. I. Attractor dynamics and parallelism in a connectionist sequential machine. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, 1986
- [33] JORDAN, M. I. Serial Order: A Parallel Distributed Processing Approach, San Diego, La Jolla, CA: Institute for Cognitive Science, University of California, 1986.
- [34] FAUSETT, L. Fundamentals of Neural Networks – Architecture, Algorithms, and Applications. [S.l.]: Prentice Hall International, Inc., 1994.
- [35] MELLO, Marília Terra de. Aplicação de redes neurais no processo de precificação de ações. Universidade Estadual de Pelotas, 2004.
- [36] ZANNETTI Jr. Luiz Antonio e ALMEIDA. Fernando C. - Exploração do uso de redes neurais na previsão do comportamento de ativos financeiros. Faculdade de Economia Administração e Contabilidade – USP. 1998.
- [37] http://www.riskglossary.com/link/efficient_market_hypothesis.htm
- [38] LAWRENCE, Ramon. Using Neural Networks to Forecast Stock Market Prices. University of Manitoba. 1997.
- [39] <http://finance.yahoo.com>