



Faculdade Integrada do Recife
Sistemas de Informação
Trabalho de Diplomação

REDES NEURAIS APLICADAS À PREVISÃO DE AÇÕES DA BOLSA DE VALORES

por

Marcus Vinícius Diniz Barros Braga

orientadora

Profa. Juliana Neiva de Gouveia Ribeiro

Recife, dezembro de 2004

Agradecimentos

- A Deus por sua companhia constante em minha vida.
- À minha mãe Elisabeth e minha avó Elza por todo amor, apoio e compreensão.
- Aos meus irmãos Fábio e Lucas pela amizade, companheirismo, carinho e apoio.
- À minha querida Andréa pelo amor, carinho e incentivo.
- Aos meus amigos Caio e Henrique pelo apoio e compreensão.
- À professora Juliana N. Ribeiro, pela orientação, confiança e estímulo.
- Ao professor André M. Cavalcanti pela amizade, apoio, carinho e exemplo de vida.
- Aos professores Cristine Gusmão, Mônica Bandeira e Telmo Araújo pela presteza, apoio e carinho.
- Aos membros da banca examinadora, professores Sandra Siebra e George Darmiton pela atenção e presteza.
- A Vladimir e todos os meus colegas pela presteza e apoio.
- A FIR – Faculdade Integrada do Recife e todo seu corpo docente que me proporcionaram obter grande crescimento pessoal e profissional, assim como todos que fazem parte da mesma pela cordialidade e apoio técnico dispensados constantemente.
- A toda minha família e amigos pela compreensão e carinho.

Sumário

LISTA DE FIGURAS	VII
LISTA DE TABELAS	IX
LISTA DE ABREVIATURAS E SIGLAS	X
LISTA DE SÍMBOLOS	XI
LISTA DE GRÁFICOS	XII
RESUMO	XIV
<i>ABSTRACT</i>	XV
CAPÍTULO 1 - INTRODUÇÃO	16
1.1 Motivação	16
1.2 Objetivo	17
1.2.1 Objetivo Geral	17
1.2.2 Objetivos Específicos	17
1.3 Organização do Trabalho	18
CAPÍTULO 2 - MODELOS DE REDES NEURAIS	19
2.1 Introdução às Redes Neurais Artificiais	19
2.2 O Neurônio Artificial	23
2.2.1 Funções de ativação	25
2.2.1.1 Função Linear por Partes	25

2.2.1.2	Função Logística (Sigmóide)	26
2.2.1.3	Função Tangente Hiperbólica	27
2.3	Topologias de Redes Neurais	27
2.3.1	Redes Diretas	28
2.3.2	Redes com Ciclos	28
2.4	Métodos de Aprendizagem	29
2.4.1	Aprendizado Supervisionado	30
2.4.2	Aprendizado Não-supervisionado	30
2.4.3	Algoritmos de Aprendizado	30
2.4.3.1	Regra Delta Generalizada	31
2.5	Modelos de Redes Neurais	33
2.5.1	Perceptron Multicamadas - MLP	34
2.5.1.1	O processo de Treinamento da MLP	35
CAPÍTULO 3	- O PROBLEMA DE PREVISÃO	36
3.1	As RNAs e o Problema de Previsão	36
3.2	Mercado de Ações	37
3.2.1	Ação	37
3.3	Formação da Base de Dados	38
3.3.1	O Arquivo de Dados Original	39
3.3.2	Técnicas Aplicadas para Ajustar a Base de Dados	40
CAPÍTULO 4	- A FERRAMENTA PREVISÃO DE ATIVOS	43
4.1	Arquitetura e Ferramentas para o Desenvolvimento	43
4.2	Componente MLP	43
4.2.1	Diagrama de Classes	45
4.2.2	Classe TMVBNeuronio	46
4.2.3	Classe TMVBSinapse	46
4.2.4	Classe TMVBCamada	47

4.2.5	Classe TMVBMultiLayerPerceptron	47
4.2.5.1	Propriedades	47
4.2.5.2	Métodos	49
4.3	Validação do Componente MLP Desenvolvido	52
4.4	Aplicação para Previsão de Ativos da BOVESPA	53
4.4.1	Método Trein 1	53
4.4.2	Método Trein 2	54
4.4.3	Método Trein 3	55
4.4.4	Configuração de Parâmetros	56
4.4.5	Inicialização da MLP	57
4.4.6	Treinamento	59
4.4.7	Testes	61
CAPÍTULO 5	– ANÁLISE COMPARATIVA DOS RESULTADOS	66
5.1	A Ferramenta <i>NeuroSolutions</i>	66
5.2	Treinamento	71
5.3	Testes	73
5.4	Resultados Obtidos	75
5.4.1	Resultados Obtidos com a Ferramenta Previsão de Ativos	75
5.4.1.1	Método Trein 1	75
5.4.1.2	Método Trein 2	78
5.4.1.3	Método Trein 3	80
5.4.2	Resultados Obtidos com a Ferramenta <i>NeuroSolutions</i>	82
5.4.2.1	Método Trein 1	82
5.4.2.2	Método Trein 2	84
5.4.2.3	Método Trein 3	86
5.4.3	Comparação entre os Resultados	88
5.4.3.1	Resumo dos Resultados com a Ferramenta Previsão de Ativos	88
5.4.3.2	Resumo dos Resultados com a Ferramenta <i>NeuroSolutions</i>	88
5.4.3.3	Os Melhores Resultados	88

CAPÍTULO 6 - CONCLUSÃO	89
6.1 Conclusões do Trabalho	89
6.2 Contribuições do Trabalho	90
6.3 Trabalhos Futuros	90
REFERÊNCIAS	92

Lista de Figuras

FIGURA 1: REPRESENTAÇÃO BÁSICA DO NEURÔNIO BIOLÓGICO	20
FIGURA 2: MODELO NÃO-LINEAR DE UM NEURÔNIO ARTIFICIAL	24
FIGURA 3: TRANSFORMAÇÃO PRODUZIDA PELA PRESENÇA DE UM BIAS	25
FIGURA 4: FUNÇÃO LINEAR	26
FIGURA 5: FUNÇÃO LOGÍSTICA	27
FIGURA 6: SENTIDO DA PROPAGAÇÃO DOS VALORES	28
FIGURA 7: REDE COM REALIMENTAÇÃO.	28
FIGURA 8: SENTIDO DA RETROPROPAGAÇÃO DOS VALORES DE ATUALIZAÇÃO DOS PESOS	31
FIGURA 9: ARQUITETURA DA MLP	35
FIGURA 10: MÉTODOS DE TREINAMENTO DA FERRAMENTA	53
FIGURA 11: ARQUITETURA DA REDE - MÉTODO <i>TREIN 1</i>	54
FIGURA 12: ARQUITETURA DA REDE - MÉTODO <i>TREIN 2</i>	55
FIGURA 13: SELEÇÃO DE NEURÔNIOS DA CAMADA DE ENTRADA – MÉTODO <i>TREIN 3</i>	55
FIGURA 14: ARQUITETURA DA REDE - MÉTODO <i>TREIN 3</i>	56
FIGURA 15: TELA DE CONFIGURAÇÃO DE PARÂMETROS DA MLP	56
FIGURA 16: MENU PARA ACESSAR A CONFIGURAÇÃO DOS PARÂMETROS DA REDE	57
FIGURA 17: ÁREA DE SELEÇÃO DO CONJUNTO DE DADOS DO TREINAMENTO	58
FIGURA 18: APRESENTAÇÃO DA FUNCIONALIDADE CONSTRUIR REDE	58
FIGURA 19: LISTAGEM DE VALORES DE CONFIGURAÇÕES DA REDE	59
FIGURA 20: APRESENTAÇÃO DA FUNCIONALIDADE TREINAR REDE	60
FIGURA 21: APRESENTAÇÃO DE INFORMAÇÕES OBTIDAS DURANTE O TREINAMENTO	61
FIGURA 22: MENU TAREFAS, TESTAR REDE	62
FIGURA 23: APRESENTAÇÃO DA FUNCIONALIDADE TESTAR REDE	62
FIGURA 24: RESULTADOS DAS MEDIDAS DE AVALIAÇÃO DO TESTE	63
FIGURA 25: AMOSTRA DO CONJUNTO DE DADOS DO TESTE	64

FIGURA 26: AMOSTRA DO CONJUNTO DE DADOS PREVISTOS NO TESTE	64
FIGURA 27: GRÁFICO DE COMPARAÇÃO ENTRE OS VALORES ORIGINAIS DO TESTE E OS PREVISTOS	65
FIGURA 28: MENU DE ACESSO ÀS FUNCIONALIDADES DE INICIALIZAÇÃO DA REDE NO NEUROSOLUTIONS	67
FIGURA 29: MENU DE ACESSO ÀS FUNCIONALIDADES DE CONSTRUÇÃO DA REDE NO NEUROSOLUTIONS	67
FIGURA 30: CONSTRUÇÃO DA RNA COM O NEUROSOLUTIONS	68
FIGURA 31: SELEÇÃO DA QUANTIDADE DE CAMADAS ESCONDIDAS COM O NEUROSOLUTIONS	68
FIGURA 32: CONFIGURAÇÃO DAS CAMADAS ESCONDIDAS COM O NEUROSOLUTIONS	69
FIGURA 33: CONFIGURAÇÃO DAS CAMADAS DE SAÍDA COM O NEUROSOLUTIONS	69
FIGURA 34: CONFIGURAÇÃO DO CONTROLE DE APRENDIZADO SUPERVISIONADO NO NEUROSOLUTIONS	70
FIGURA 35: CONFIGURAÇÃO DAS CONFIGURAÇÕES DE PROVA NO NEUROSOLUTIONS	70
FIGURA 36: FUNCIONALIDADE PARA TREINAMENTO DA REDE NO NEUROSOLUTIONS	71
FIGURA 37: CONFIGURAÇÕES DO TREINAMENTO NO NEUROSOLUTIONS	71
FIGURA 38: PROGRESSO DO TREINAMENTO NO NEUROSOLUTIONS	72
FIGURA 39: VALORES DOS INDICADORES DO TREINAMENTO NO NEUROSOLUTIONS	72
FIGURA 40: AMOSTRA DE VALORES DE INDICADORES DO TREINAMENTO E VALIDAÇÃO NO NEUROSOLUTIONS	73
FIGURA 41: FUNCIONALIDADE PARA TESTE DA REDE NO NEUROSOLUTIONS	73
FIGURA 42: CONFIGURAÇÕES DO TESTE NO NEUROSOLUTIONS	74
FIGURA 43: COMPARATIVO E INDICADORES DO TESTE NO NEUROSOLUTIONS	74
FIGURA 44: AMOSTRA DE VALORES APÓS O TESTE NO NEUROSOLUTIONS	75

Lista de Tabelas

TABELA 1: DIFERENÇAS ENTRE O COMPUTADOR E O CÉREBRO HUMANO	22
TABELA 2: AMOSTRA DE DADOS DO ARQUIVO GERADO PELO GOL	39
TABELA 3: AMOSTRA DE DADOS APÓS APLICAÇÃO DE TÉCNICAS DE AJUSTES	41
TABELA 4: AMOSTRA DE DADOS APÓS APLICAÇÃO DA QUARTA TÉCNICA DE AJUSTES	41
TABELA 5: EXPERIMENTOS REALIZADOS COM A MLP NO MÉTODO <i>TREIN 1</i>	76
TABELA 6: EXPERIMENTOS REALIZADOS COM A MLP NO MÉTODO <i>TREIN 2</i>	78
TABELA 7: EXPERIMENTOS REALIZADOS COM A MLP NO MÉTODO <i>TREIN 3</i>	80
TABELA 8: EXPERIMENTOS REALIZADOS NO NEUROSOLUTIONS COM A MLP NO MÉTODO <i>TREIN 1</i>	82
TABELA 9: EXPERIMENTOS REALIZADOS NO NEUROSOLUTIONS COM A MLP NO MÉTODO <i>TREIN 2</i>	84
TABELA 10: EXPERIMENTOS REALIZADOS NO NEUROSOLUTIONS COM A MLP NO MÉTODO <i>TREIN 3</i>	86

Lista de Abreviaturas e Siglas

APE – Erro Percentual Absoluto

Backpropagation – Algoritmo de Retropropagação

BOVESPA – Bolsa de Valores do Estado de São Paulo

Extended – Tipo Numérico com Ponto Flutuante do Object Pascal

GOL – Enfoque Cotações, GOL – Gráficos On-line, versão 5.0

Integer – Tipo Numeral Inteiro do Object Pascal

MAE – Erro Médio Absoluto

MAPE – Erro Percentual Absoluto Médio

ME – Erro Médio

MLP – Perceptron Multicamadas com Retropropagação

MSE – Erro Quadrático Médio

NAs – Neurônios Artificiais

NB – Neurônios Biológicos

PS – Pesos Sinápticos

RNA – Rede Neural Artificial

RNNs – Redes Neurais Naturais

TFileName – Tipo literal para nome de arquivos do Object Pascal

TNLP4 – Ação Preferencial Nominativa da Empresa Telemar

TStrings – Tipo Lista de Literais do Object Pascal

Lista de Símbolos

EQUAÇÃO 1: REGRA DE PROPAGAÇÃO _____	25
EQUAÇÃO 2: APLICAÇÃO DA FUNÇÃO DE ATIVAÇÃO _____	25
EQUAÇÃO 3: FUNÇÃO DE ATIVAÇÃO LINEAR POR PARTES _____	26
EQUAÇÃO 4: FUNÇÃO DE ATIVAÇÃO LOGÍSTICA (SIGMÓIDE) _____	26
EQUAÇÃO 5: VARIAÇÃO DA FUNÇÃO DE ATIVAÇÃO LOGÍSTICA _____	27
EQUAÇÃO 6: FUNÇÃO DE ATIVAÇÃO TANGENTE HIPERBÓLICA _____	27
EQUAÇÃO 7: EQUAÇÃO MAIS UTILIZADA PARA CÁLCULO DO CUSTO DO ERRO _____	32
EQUAÇÃO 8: EQUAÇÃO DE AJUSTE DOS PESOS _____	32
EQUAÇÃO 9: EQUAÇÃO DO ERRO PARA OS NEURÔNIOS DA CAMADA DE SAÍDA _____	32
EQUAÇÃO 10: EQUAÇÃO DO ERRO PARA ATIVAÇÃO SIGMÓIDE (SAÍDA) _____	32
EQUAÇÃO 11: EQUAÇÃO DO ERRO PARA OS NEURÔNIOS DAS CAMADAS DE INTERMEDIÁRIAS _____	33
EQUAÇÃO 12: EQUAÇÃO DO ERRO PARA ATIVAÇÃO SIGMÓIDE (INTERMEDIÁRIAS) _____	33
EQUAÇÃO 13: EQUAÇÃO PARA CÁLCULO DO PERCENTUAL DE RETORNO DIÁRIO _____	42
EQUAÇÃO 14: EQUAÇÃO PARA CÁLCULO DO MSE _____	60
EQUAÇÃO 15: EQUAÇÃO PARA CÁLCULO DO ME _____	62
EQUAÇÃO 16: EQUAÇÃO PARA CÁLCULO DO MAE _____	63
EQUAÇÃO 17: EQUAÇÃO PARA CÁLCULO DO MAPE _____	63
EQUAÇÃO 18: EQUAÇÃO PARA CÁLCULO DO APE _____	65

Lista de Gráficos

GRÁFICO 1: REPRESENTAÇÃO GRÁFICA DO TIPO <i>CANDLES</i> DOS VALORES COLETADOS DA TNLP4 _____	40
GRÁFICO 2: VALORES DA BASE DE DADOS NO INTERVALO DE 1 A 890 _____	59
GRÁFICO 3: AMOSTRA DE VALORES OBTIDOS DURANTE O TREINAMENTO _____	61
GRÁFICO 4: COMPARAÇÃO ENTRE OS VALORES ORIGINAIS E OS PREVISTOS NO MELHOR TREINAMENTO (<i>TREIN 1</i>). _____	77
GRÁFICO 5: COMPARAÇÃO ENTRE OS VALORES PREVISTOS NO MELHOR E PIOR TREINAMENTO (<i>TREIN 1</i>). _____	77
GRÁFICO 6: COMPARAÇÃO ENTRE OS VALORES ORIGINAIS E OS PREVISTOS NO MELHOR TREINAMENTO (<i>TREIN 2</i>). _____	79
GRÁFICO 7: COMPARAÇÃO ENTRE OS VALORES PREVISTOS NO MELHOR E PIOR TREINAMENTO (<i>TREIN 2</i>). _____	79
GRÁFICO 8: COMPARAÇÃO ENTRE OS VALORES ORIGINAIS E OS PREVISTOS NO MELHOR TREINAMENTO (<i>TREIN 3</i>). _____	81
GRÁFICO 9: COMPARAÇÃO ENTRE OS VALORES PREVISTOS NO MELHOR E NO PIOR TREINAMENTO (<i>TREIN 3</i>). _____	81
GRÁFICO 10: COMPARAÇÃO ENTRE OS VALORES ORIGINAIS E OS PREVISTOS NO MELHOR TREINAMENTO DO NEUROSOLUTIONS (<i>TREIN 1</i>). _____	83
GRÁFICO 11: COMPARAÇÃO ENTRE OS VALORES PREVISTOS NO MELHOR E PIOR TREINAMENTO DO NEUROSOLUTIONS (<i>TREIN 1</i>). _____	83
GRÁFICO 12: COMPARAÇÃO ENTRE OS VALORES ORIGINAIS E OS PREVISTOS NO MELHOR TREINAMENTO DO NEUROSOLUTIONS (<i>TREIN 2</i>). _____	85
GRÁFICO 13: COMPARAÇÃO ENTRE OS VALORES PREVISTOS NO MELHOR E PIOR TREINAMENTO DO NEUROSOLUTIONS (<i>TREIN 2</i>). _____	85
GRÁFICO 14: COMPARAÇÃO ENTRE OS VALORES ORIGINAIS E OS PREVISTOS NO MELHOR TREINAMENTO DO NEUROSOLUTIONS (<i>TREIN 3</i>). _____	87

GRÁFICO 15: COMPARAÇÃO ENTRE OS VALORES PREVISTOS NO MELHOR E PIOR
TREINAMENTO DO NEUROSOLUTIONS (*TREIN 3*). _____ 87

Resumo

O aumento da aceitação de sistemas que utilizam Redes Neurais Artificiais em algumas áreas da economia tem acontecido basicamente por dois motivos: elas têm a habilidade de analisar dados incompletos e com ruídos e a capacidade de aprendizagem através de dados históricos. Também é verdade que o sucesso de aplicações passadas trouxeram essa popularidade.

O estudo das Redes Neurais Artificiais é motivado pelo reconhecimento das grandes diferenças entre a forma de processamento feita pelo cérebro humano e os computadores digitais convencionais. À medida que essa tecnologia se torna mais conhecida, ocorre também o surgimento de novas aplicações práticas. A utilização de Redes Neurais Artificiais para previsão, ou regressão, tem apresentado resultados suficientemente bons para motivar a continuidade dos estudos tradicionais e também para novas abordagens. O mercado também vem, paulatinamente, demonstrando que existe demanda para a utilização de serviços baseados em sistemas conexionistas.

O objetivo deste trabalho é investigar a habilidade do modelo MLP em realizar previsões de valores de ações do mercado financeiro. São realizados experimentos com uma ferramenta desenvolvida neste trabalho, e seus resultados são comparados aos obtidos com outra ferramenta já reconhecida e consolidada comercialmente. Também foi criada e ajustada uma base de dados com valores da ação da Telemar PN – ação que atualmente possui a maior representatividade no IBOVESPA (Índice da Bolsa de Valores de São Paulo) – contendo valores do período de 2000 até 2004.

Palavras-chaves: Inteligência Computacional, Redes Neurais, Previsão, Mercado Financeiro, Ações e Multi Layer Perceptron (MLP).

Abstract

The Neural Network study is based in the huge differences between the way human brain and digital computers process data. Such study became popular in specific areas, as Economy, thanks the Neural Network ability in dealing with incomplete information and its capacity in analyze historical data as well. This research intends to analyze the Multi Layer Perceptron dealing with the stock market reality in São Paulo's stock exchange, through the variation of Telemar PN stock in the IBOVESPA (São Paulo's stock exchange Index). The results were faced with the results from another software tool already working.

Key words: *Computational Intelligence, Artificial Neural Networks, Predict, Financial Market, Stocks and Multi Layer Perceptron (MLP).*

Capítulo 1 - Introdução

Nesta monografia, estão descritos os resultados e contribuições alcançados, cujo principal objetivo é investigar a aplicação de redes neurais na previsão de índices de ações na bolsa de valores.

Este capítulo está organizado da seguinte maneira: a seção 1.1 descreve as motivações para o estudo de redes neurais aplicadas à previsão de ações. Na seção 1.2, será feita uma exposição dos objetivos do trabalho. Finalizando, a seção 1.3 apresenta a estrutura de organização desta monografia.

1.1 Motivação

Com a globalização, cada vez mais o mercado acionário vem adquirindo uma crescente importância no cenário financeiro internacional. O crescimento do interesse pelo mercado de ações tem estimulado a realização de estudos de previsão do seu comportamento. Várias técnicas matemáticas, estatísticas (modelos ARIMA, ARMA e SARIMA) e computacionais têm sido aplicadas para este fim. Dentre as técnicas computacionais utilizadas, destacam-se as Redes Neurais Artificiais, uma das mais difundidas da Inteligência Computacional.

A rede neural, por ter a habilidade de lidar com dados ruidosos, incompletos ou imprecisos e de prever sistemas não-lineares, tem trazido resultados bastante satisfatórios em aplicações no mercado financeiro. Ao realizar um investimento em uma carteira de ações¹, o investidor espera que o seu retorno seja grande o suficiente para compensar os riscos que está correndo. Ou seja, seu objetivo é tentar minimizar o risco e maximizar o retorno esperado. Através da estimativa do valor futuro da ação, o investidor adquire maior segurança na

¹ Carteira de ações é o termo utilizado no mercado financeiro que significa o conjunto de ações selecionadas para investimento.

tomada de decisão.

As redes neurais auxiliam na previsão de tendências do mercado financeiro pela sua capacidade de análise de padrões. Embora métodos estatísticos convencionais possam ser utilizados, as redes são bem mais apropriadas para lidar com oscilações e perturbações. As redes podem facilitar bastante o trabalho de um investidor de ações e, ao mesmo tempo, aumentar a possibilidade de ganhos.

“A área das finanças caracteriza-se como uma das principais áreas de aplicação de redes neurais artificiais em negócios, por tratar-se de um ramo onde as decisões são baseadas em informações com um alto grau de variabilidade e incerteza” [Mello, 2004].

A principal motivação para o tema de pesquisa escolhido é a utilização de uma tecnologia já consolidada no meio acadêmico e no mercado na investigação de uma aplicação prática – Previsão de Valores de Ações na Bolsa de Valores.

Neste contexto, surgem as necessidades de obtenção e tratamento de uma base de dados real que possa ser utilizada nos experimentos, bem como o estudo e a utilização de modelos de redes neurais para obtenção dos resultados comparativos.

1.2 Objetivo

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é investigar a aplicação de redes neurais na previsão do preço de ações da Bolsa de Valores de São Paulo (BOVESPA).

1.2.2 Objetivos Específicos

A partir do objetivo geral do trabalho, surgem, naturalmente, os objetivos específicos a seguir:

- Obtenção de uma base de dados históricos;
- Estudo de técnicas de pré-processamento de dados e posterior aplicação das técnicas no tratamento dos dados;
- Estudo dos principais modelos de redes neurais que serão utilizados durante os procedimentos experimentais;
- Implementação de uma ferramenta que suporte um ou mais modelos de redes neurais para realização dos experimentos;
- Utilização de uma outra ferramenta disponível no mercado para realização dos experimentos.

1.3 Organização do Trabalho

Este trabalho contempla seis capítulos. No segundo capítulo, é apresentada uma introdução sobre as redes neurais artificiais, neurônios, funções de ativação e algoritmos de aprendizagem.

No terceiro capítulo, apresenta-se um breve resumo do problema de previsão de valores de ações e detalhes da construção e ajustes da base de dados utilizada nos experimentos.

O quarto capítulo descreve a ferramenta Previsão de Ativos, implementada durante este trabalho, seus componentes e os métodos de treinamento, bem como um manual de utilização.

No quinto capítulo, é apresentada a descrição detalhada dos passos seguidos na utilização da ferramenta *NeuroSolutions*, assim como os resultados comparativos entre as duas ferramentas.

No sexto capítulo, encontram-se as conclusões e a contribuição do trabalho desenvolvido, além de sugestões para o desenvolvimento de trabalhos futuros.

Capítulo 2 - Modelos de Redes Neurais

Neste capítulo, serão abordados os principais conceitos acerca dos neurônios artificiais, das redes neurais artificiais, das topologias de redes neurais e dos métodos de aprendizagem das redes. Também será descrita a rede Perceptron com Multicamadas, que representa o modelo de rede neural mais típico e amplamente utilizado por pesquisadores da área e que foi o modelo adotado neste trabalho.

2.1 Introdução às Redes Neurais Artificiais

O estudo das Redes Neurais Artificiais (RNAs) é motivado pelo reconhecimento das grandes diferenças entre a forma de processamento feita pelo cérebro humano e os computadores digitais convencionais [Haykin, 2001]. Elas são inspiradas nos neurônios biológicos (NB) e no sistema nervoso humano, no entanto é importante informar que são mínimas as semelhanças entre as soluções atuais de RNAs (*hardware* ou *software*) e as Redes Neurais Naturais (RNNs) [Azevedo, 2000].

Para se entender os objetivos das RNAs deve-se, antes, tentar adquirir uma melhor compreensão acerca do funcionamento do cérebro humano. O cérebro está dividido em três partes principais: o tronco encefálico, o cerebelo e o cérebro propriamente dito. O tronco encefálico, conhecido como cérebro reptiliano, é o responsável por transferir informações dos sentidos e controlar funções básicas como respiração e os batimentos cardíacos. Ele não cria raciocínio ou sentimento. O cerebelo ajuda na movimentação do corpo governando a coordenação dos músculos e conservando parte da memória dos movimentos. O cérebro propriamente dito, às vezes chamado de cérebro dos mamíferos, não existia até há cerca de 80 milhões de anos. É nele onde tudo acontece: os pensamentos, as emoções e a memória [Khalsa, 1997].

O cérebro é formado por células, conhecidas como neurônios. Os neurônios funcionam à base de eletricidade. Tudo que acontece no cérebro viaja através dos neurônios por corrente elétrica. Eles são capazes de construir essa corrente de memória devido às suas formas. São mais alongados que outros tipos de células e apresentam sistemas de “ramificações” em uma das extremidades e sistemas de “raízes” na outra. A conexão entre as ramificações de um neurônio com as de um outro é chamado de sinapse. As raízes são conhecidas como axônios. As informações fluem para os axônios a partir das “ramificações” dos neurônios vizinhos. Em seguida, a informação é direcionada para o tronco (corpo celular). Finalmente, alcançam os dendritos que repassam os impulsos nervosos para os axônios de outros neurônios e assim por diante, formando uma cadeia (pensamento ou memória completa), conforme a Figura 1. *“Quanto mais dendritos e conexões você tiver – e também, quanto mais saudáveis forem os dendritos – mais inteligente você é”* [Khalsa, 1997].

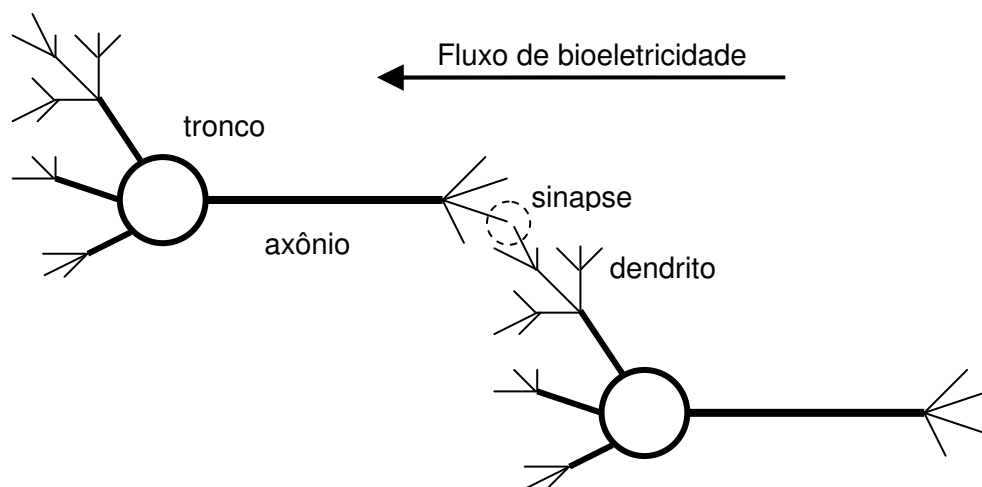


Figura 1: Representação básica do neurônio biológico

Uma RNA, tal como seu paralelo biológico, é composta de um certo número de nós ou unidades, também denominados de neurônios artificiais (NAs) ou simplesmente neurônios, que são conectados por ligações sinápticas (*links*). Cada ligação possui um valor associado (peso). Alguns neurônios são conectados ao ambiente externo, sendo designados como entradas (*inputs*) e saídas (*outputs*)

[Medeiros, 2003]. *“Na sua forma mais geral, uma RNA é uma máquina que é projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse”* [Haykin, 2001].

Os computadores atuais são bem mais velozes que o cérebro. Os circuitos de silício executam eventos em nanosegundos (10^{-9} S), enquanto no cérebro eles acontecem em milissegundos (10^{-3} S). Porém, no cérebro, esse processamento é feito de forma muito mais eficiente que nos computadores, ou seja, ele é capaz de executar tarefas extremamente complexas em curtíssimo tempo [Haykin, 2001]. Isso só é possível graças ao alto grau de paralelismo que ocorre no cérebro. Estima-se que lá existam aproximadamente de 10^{11} a 10^{14} neurônios e cada um com 10^3 a 10^4 conexões (sinapses) [Cottrell, 1985].

Um exemplo que ilustra essa notável eficiência é o seguinte: comparando um neurônio artificial com o biológico pode-se imaginar o neurônio artificial como um avião Concorde (supersônico) e o neurônio biológico como um avião Teco-teco. A tarefa corresponde ao transporte de uma enorme quantidade de carga do Brasil até o Japão. Para dar início à tarefa são carregados cada um dos aviões com os respectivos volumes. O Concorde, por ser supersônico, daria um resultado com maior velocidade que um único Teco-teco. Porém, utilizando a quantidade 10^{14} Teco-tecos o resultado seria bem diferente. Teria-se todos os trilhões de Teco-tecos, em paralelo, percorrendo do Brasil ao Japão por diversas vezes contra um único Concorde. Isso ocasionaria a formação de uma grande ponte de Teco-tecos entre o Brasil e o Japão. O Concorde não poderia competir com os Teco-tecos. A vantagem dos Teco-tecos seria ainda maior no caso de eventuais problemas, ou seja, até mesmo a queda (perda) de alguns Teco-tecos não alteraria o resultado. Já para o Concorde, por ser único, esse impacto seria catastrófico.

O cérebro e sua imensa rede de neurônios também têm a habilidade de desenvolver regras, criar e acumular experiência, além da capacidade de se adaptar ao meio. Essa capacidade de adaptação é de extrema importância no

estudo dos NAs. Pode-se, então, dizer que uma RNA assemelha-se ao cérebro em dois aspectos: o conhecimento é adquirido através da aprendizagem e o conhecimento adquirido é armazenado através dos pesos sinápticos (PS) – que funcionam como as forças de conexões neurais naturais [Haykin, 2001]. O seu processo de aprendizagem é conhecido como *algoritmo de aprendizagem* [Haykin, 2001] – que será visto com mais detalhes adiante, no item 2.4.

As RNAs, através de sua habilidade de aprender, tornam-se capazes de generalizar, ou seja, produzir saídas adequadas para entradas que não estavam no momento do treinamento. Desta forma, as RNAs adquirem um grande poder para resolução de problemas extremamente complexos e, para isso, elas devem funcionar integradas a outros sistemas computacionais [Haykin, 2001].

Na Tabela 1, baseada em Cottrell [Cottrell, 1985] e Simpson [Simpson, 1990], é apresentada uma comparação entre as principais diferenças existentes nos computadores atuais e no cérebro humano, para que se tenha uma idéia mais clara sobre a capacidade adaptativa do cérebro, em contraste com a rigidez e precisão dos computadores convencionais [Fernandes, 1995].

Características	Computador	Cérebro Humano
Elementos computacionais	processadores	neurônio simples
Velocidade de processamento	10^{-9} segundos	10^{-3} segundos
Tipo de processamento	serial	paralelo
Confiabilidade dos elementos	confiável	não confiável
Tolerância a falhas	quase nenhuma	grande
Tipo de sinal	precisos, simbólicos	imprecisos
Tipo de controle	centralizado	distribuído
Armazenamento da informação	substituível	adaptável

Tabela 1: Diferenças entre o computador e o cérebro humano

Desde o início dos estudos das RNAs, com McCulloch e Pitts (1943 – Perceptron, o primeiro NA), passando por Hebb (1949 – regra de aprendizado), Gabor (1954 – filtros lineares adaptativos), Taylor (1956 – memória associativa), Rosenblatt (1958 – Perceptron Multicamada), Widrow e Hoff (algoritmo da menor média quadrática), Minsky e Papert (1969 – limitações do Perceptron), Grossberg (1980 – aprendizado competitivo), Hopfield (1982 – função de energia para redes recorrentes), Barto, Sutton e Anderson (1983 – aprendizado por reforço), Rumelhart, Hinton e Williams (1986 – algoritmo de retropropagação), Kramer (1989 – gradiente conjugado), entre outros como Saarien (1991) e Baititi (1992), aprendeu-se que todos os conceitos formulados na área são multidisciplinares e sempre foram baseados nos modelos biológicos ou em situações reais. *“A principal idéia sempre foi reproduzir o funcionamento do que está presente fisiologicamente ou simbolicamente no cérebro humano”* [Medeiros, 2003].

2.2 O Neurônio Artificial

Nas RNAs, cada neurônio possui um conjunto de entradas ligadas a outros neurônios e um conjunto de saídas conectadas a outros neurônios. O neurônio artificial apresenta também um nível atual de ativação. Conforme o nível atual das entradas, conjugadas com os pesos sinápticos, o valor resultante irá ativar ou não as saídas deste mesmo neurônio. A Figura 2 apresenta o modelo de um neurônio artificial, que forma a base para o projeto de uma Rede Neural Artificial (RNA). São identificados três elementos básicos do modelo neuronal: sinapses, um somador e função de ativação [Haykin, 2001]:

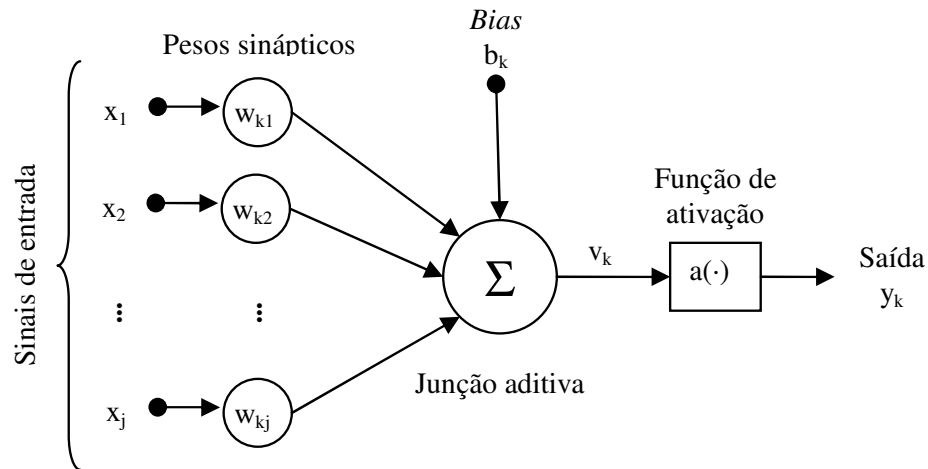


Figura 2: Modelo não-linear de um neurônio artificial

Nos sinais de entrada serão apresentados os valores correspondentes aos padrões (de x_1 até x_j) que, neste trabalho, servirão para previsão de valores financeiros e, em outros casos, por exemplo, poderiam ser valores de estoque, índices pluviométricos, etc. Em aplicações para classificação, poderiam ser valores de padrões para reconhecimento de caracteres ou imagens, identificação de pessoas, localização de alvos, análise de crédito, entre outros [Medeiros, 2003].

Em seguida, estão os pesos ou conexões (de w_{k1} até w_{kj}). Eles assumem valores tais que, quando se aplica um padrão de entrada, obtém-se uma saída intermediária (v_k). Os valores que os pesos assumem representam o aprendizado da rede. O uso do bias (b_k) tem como efeito um ajuste fino para v_k , como pode ser observado na Figura 3. A função de ativação " $a(\cdot)$ " serve para restringir a amplitude da saída y_k de um neurônio artificial. Tipicamente, o intervalo dessa amplitude é escrito como: unitário fechado $[0, 1]$ ou alternativamente $[-1, 1]$.

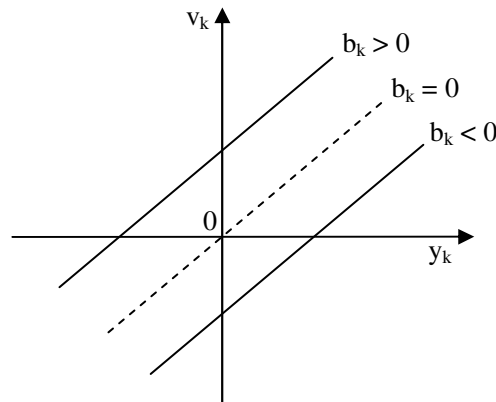


Figura 3: Transformação produzida pela presença de um bias

Pode-se representar o modelo do neurônio artificial através das Equações 1 e 2 [Haykin, 2001]:

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k$$

Equação 1: Regra de Propagação

$$y_k = a(v_k)$$

Equação 2: Aplicação da Função de Ativação

2.2.1 Funções de ativação

Existem três funções de ativação que são mais freqüentemente utilizadas em neurônios individuais, são elas: a função linear por partes, a função logística e a função tangente hiperbólica. Porém qualquer função monotonicamente crescente e contínua tal que $v \in \Re$ e $a(v) \in [-1, 1]$ pode ser usada em modelagem neural [Azevedo, 2000].

2.2.1.1 Função Linear por Partes

A função linear (Figura 4) é expressa pela Equação 3, onde α é uma

constante real que regula a intensidade da atividade de a .

$$a(v) = \alpha \times v$$

Equação 3: Função de ativação linear por partes

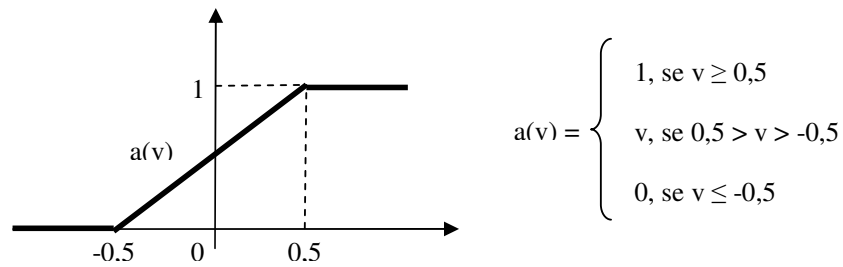


Figura 4: Função Linear

Os valores 0 e 1 indicam respectivamente a saída mínima e máxima e são chamados de pontos de saturação.

2.2.1.2 Função Logística (Sigmóide)

Essa função, também conhecida como função sigmóide (Figura 5), é monótona, não-decrescente e sua resposta é gradual e não-linear. Ela possui os pontos de saturação igual a 0 e 1 e é definida pela Equação 4, onde v deve ser um escalar positivo e d o parâmetro de inclinação da função. No limite, quando $v \rightarrow \infty$, a função é definida por:

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

Quanto mais v cresce, mais abrupta a função é nas proximidades de $d = 0$.

$$a(v) = \frac{1}{1 + e^{-dv}}$$

Equação 4: Função de ativação logística (sigmóide)

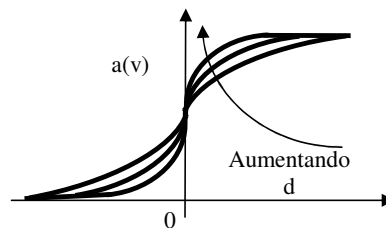


Figura 5: Função Logística

Uma outra variação bastante utilizada da função logística definida na Equação 5.

$$a(v) = \tanh(v/2) = \frac{1 - e^{-v}}{1 + e^{-v}}$$

Equação 5: Variação da função de ativação logística

Ela é a versão simétrica, cujas saídas encontram-se definidas no intervalo $[-1, 1]$.

2.2.1.3 Função Tangente Hiperbólica

É a mais popular função bipolar [Azevedo, 2000]. É definida pela Equação 6, onde d é escalar positivo.

$$a(v) = \tanh(dv) = \frac{e^{dv} - e^{-dv}}{e^{dv} + e^{-dv}}$$

Equação 6: Função de ativação tangente hiperbólica

O fato de permitir que uma função de ativação do tipo logística assuma valores negativos traz benefícios analíticos [Haykin, 2001].

2.3 Topologias de Redes Neurais

A topologia de uma rede neural é o modo como os seus elementos estão interconectados uns aos outros. As mais comumente utilizadas são: as redes diretas (*feedforward*) e as redes com ciclos (retroalimentadas ou recorrentes)

[Azevedo, 2000].

2.3.1 Redes Diretas

Redes Diretas (*Feedforward*) são aquelas cujo grafo não tem ciclos [Azevedo, 2000]. Nestas redes, os dados processados entre os neurônios da rede são propagados em sentido único sem que sejam informados novamente como entradas, conforme ilustra a Figura 6.

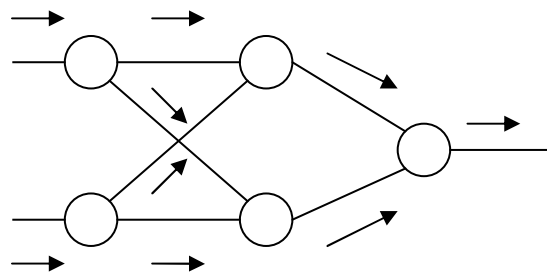


Figura 6: Sentido da propagação dos valores

2.3.2 Redes com Ciclos

Redes com ciclos (ou com realimentação, ou com retroação, ou com 'feedback') são aquelas cujo grafo de conectividade contém, ao menos, um ciclo [Azevedo, 2000]. Nestas redes, pelo menos, um dado processado deve ser propagado novamente para a rede, conforme mostra a Figura 7.

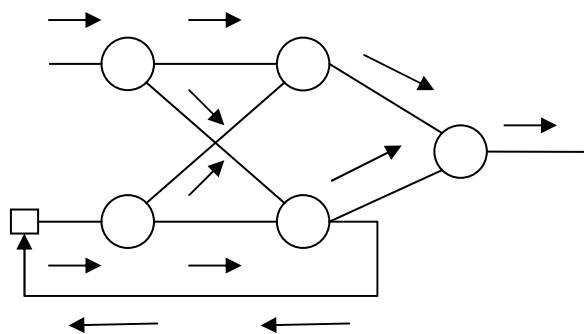


Figura 7: Rede com realimentação.

2.4 Métodos de Aprendizagem

O aprendizado é um processo essencial e tem sido bastante pesquisado e investigado, tanto para os sistemas biológicos quanto para os computacionais. Em relação ao processo de aprendizagem das redes neurais artificiais, há diversas formas de fazê-lo, devendo-se levar em conta os diferentes tipos de redes, ou seja, as redes possuem métodos de aprendizagem distintos.

A aprendizagem pode ser definida operacionalmente como a habilidade de executar novas tarefas que não podiam ser realizadas antes, ou executar melhor antigas tarefas. As novas habilidades são decorrentes das mudanças produzidas pelo processo de aprendizagem. Computacionalmente, pode-se também defini-la como um processo de otimização, isto é, aprendizagem consiste em um processo de busca de solução em um espaço multidimensional de parâmetros, em geral, os pesos das conexões [Oliveira, 2001].

Nas RNAs, o aprendizado estabelece valores para os pesos de forma a dar à rede um sentido de processamento. Ao entrar com dados em uma RNA não treinada, os valores apresentados em sua saída são aleatórios e não previsíveis [Medeiros, 2003]. A aprendizagem em uma RNA é realizada por um procedimento adaptativo, algoritmo ou regra de aprendizagem, envolvendo:

- Captura de estímulos;
- Modificações em seus parâmetros decorrentes dos estímulos do meio ambiente.

Pode-se classificar o aprendizado de duas formas: aprendizado associativo e aprendizado não-associativo. O aprendizado associativo implica em aprender sobre o relacionamento que há entre pares de estímulos. Esse tipo de aprendizado é modelo para RNAs que utilizam aprendizado supervisionado. No aprendizado não-associativo, não há estímulos secundários para associar com estímulos primários. Esse tipo de aprendizado é modelo para RNAs não-

supervisionadas.

2.4.1 Aprendizado Supervisionado

Acontece quando se utiliza um agente externo que indica à rede a resposta desejada para o padrão de entrada, ou seja, durante o treinamento, são apresentados os padrões de entrada e para estes uma saída correspondente. Caso a rede apresente uma resposta diferente à real, a RNA gera um sinal de erro que é utilizado para calcular o ajuste que deverá ser feito nos pesos. Esse agente externo é conhecido como professor ou supervisor, daí o nome Aprendizado Supervisionado.

2.4.2 Aprendizado Não-supervisionado

Acontece quando não se utiliza um agente externo que indica a resposta desejada para os padrões de entrada, ou seja, não existe um professor nem tampouco uma saída desejada. Nesse tipo de aprendizado, a RNA recebe os padrões de entrada e os organiza, arbitrariamente, em categorias. Ao serem apresentadas novas entradas, a RNA verifica a classe correspondente, se não existir, cria-se, então, uma nova classe.

2.4.3 Algoritmos de Aprendizado

“É um conjunto de regras bem definidas para a solução de um problema de aprendizado” [Oliveira, 2001]. Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais. Os algoritmos diferem entre si, principalmente, pelo modo como os pesos são modificados. Alguns exemplos são: Algoritmo de Hebb (postulado de Hebb – 1949), o baseado em memória, o baseado em erros e o competitivo.

2.4.3.1 Regra Delta Generalizada

A regra Delta, desenvolvida por Bernard Widrow e Ted Hoff (conhecida como regra Widrow-Hoff), é bastante utilizada e foi inspirada na Regra de Hebb. Por ela minimizar o erro médio quadrático também é chamada de “*Least Mean Square*” (LMS).

A regra Delta Generalizada é uma variação da regra Delta e também é conhecida como Algoritmo de Retropropagação (*Backpropagation*) [Rumelhart, 1986]. Esse algoritmo é fortemente utilizado em redes diretas com múltiplas camadas e aprendizado supervisionado e funciona fazendo *jus* ao próprio nome, ou seja, após o primeiro passo, que é a propagação dos padrões de entrada, a informação obtida na saída é retropropagada (enviada de volta) às camadas anteriores, para que os pesos possam ser atualizados conforme o resultado obtido, como exemplifica a Figura 8.

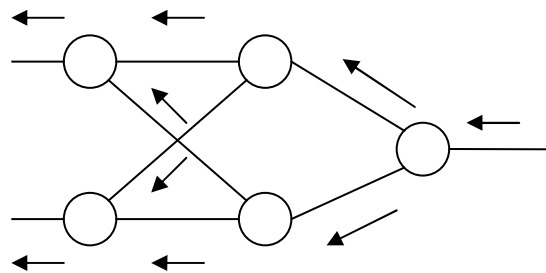


Figura 8: Sentido da retropropagação dos valores de atualização dos pesos

A informação que é retropropagada é obtida por uma função de custo que minimiza o valor do erro que é calculado, verificando a diferença entre o resultado ideal e o obtido. O método minimiza o erro e realiza iterativamente ajustes nos pesos, passando por todos os padrões do conjunto de treinamento. Esse método é conhecido como descida de gradiente (*gradient descent*). “Existem alguns tipos diferentes de funções custo, entretanto a mais utilizada é o mínimo erro quadrado definido na Equação 7 a seguir” [Ribeiro, 1999]:

$$E_p = \frac{1}{2} \times \sum_{j=1}^n (t_{pj} - o_{pj})^2$$

Equação 7: Equação mais utilizada para cálculo do custo do erro

onde o_{pj} é a resposta fornecida pela unidade u_j e t_{pj} corresponde à resposta desejada.

O processo de atualização dos pesos acontece na direção da camada de saída para a camada de entrada, camada após camada. O ajuste dos pesos é dado pela Equação 8. Esse processo é conhecido como convergência e acontece, quando o erro global diminui a cada época de treinamento. Número de épocas de treinamento é a quantidade de vezes que os padrões são apresentados à rede durante o processo de aprendizagem.

$$\Delta w_{ij} = \eta \times \delta_{pj} \times o_i$$

Equação 8: Equação de ajuste dos pesos

onde η representa a taxa de aprendizagem, o_i é a resposta da unidade i que se conecta à unidade j e δ_{pj} corresponde ao erro ocorrido entre a resposta desejada e a resposta fornecida pela unidade j . No caso dos neurônios da camada de saída, o valor de δ_{pj} é dado pela Equação 9 .

$$\delta_{ij} = f_j'(v_{pj}) \times (t_{pj} - o_{pj})$$

Equação 9: Equação do erro para os neurônios da camada de saída

Quando a função de ativação f_j utilizada é a função sigmóide, a Equação 9 torna-se, após o cálculo da derivada primeira f_j' , a Equação 10:

$$\delta_{ij} = o_{pj} \times (t_{pj} - o_{pj}) \times (1 - o_{pj})$$

Equação 10: Equação do erro para ativação sigmóide (saída)

Para as unidades j nas camadas intermediárias, não há como computar diretamente os ajustes em função do erro $(t_{pj} - o_{pj})$, sendo estes pesos calculados

em função dos erros δ_{pj} observados nas unidades das camadas subseqüentes, conforme a Equação 11:

$$\delta_{pj} = f'_j(v_{pj}) \times \sum_k w_{jk} \times \delta_{pk}$$

Equação 11: Equação do erro para os neurônios das camadas de intermediárias

Que no caso da função sigmóide, se transforma na equação 12.

$$\delta_{pj} = o_j \times (1 - o_j) \times \sum_k w_{jk} \times \delta_{pk}$$

Equação 12: Equação do erro para ativação sigmóide (intermediárias)

O processo de aprendizagem prossegue, até que algum critério seja estabelecido como, por exemplo, um mínimo de erro global, ou uma diferença sucessiva mínima entre erros calculados para cada iteração.

O *backpropagation* é, na verdade, um método para implementar o processo de gradiente descendente no espaço dos pesos para o treinamento de uma rede *feedforward* (rede direta). O objetivo do método é, portanto, a computação das derivadas parciais de uma certa função $F(\vec{w}; \vec{x}_p)$ implementada pela rede em relação ao vetor peso \vec{w} para uma dado vetor de entrada \vec{x}_p . O emprego destas derivadas no ajuste dos pesos garante a minimização do erro na camada de saída [Ribeiro, 1999].

2.5 Modelos de Redes Neurais

Existem vários modelos de RNAs. Os principais são: Perceptron (F. Rosenblatt – 1957); Perceptron Multicamadas com Retropropagação ou MLP (P. Werbos, D. Parker, D. Rumelhard – 1974); Rede Kohonen (T. Kohonen – 1980); Rede Hopfield (J. Hopfield – 1982), o Classificador de Carpenter-Grossberg (G. Carpenter, S. Grossberg – 1988), Redes de Função de Base Radial ou RBF (B.

Mulgrew – 1996), entre outros como ART, RAM-Based, Jordan/Elman, Arquitetura Cascade Correlation, [Fahlman & Lebiere, 1990], entre outras.

O modelo de RNA que mais tem sido utilizado em problemas de previsão é o MLP [Fernandes, 1995], [Yunos, 1995], [Mello, 2004]. As Redes Neurais Genéricas para Regressão – GRNN (*Generic Regression Neural Network*), os algoritmos genéticos e outros métodos de otimização são soluções alternativas ou complementares existentes que atualmente estão sendo utilizadas em estudos de previsão.

2.5.1 Perceptron Multicamadas - MLP

O Perceptron Multicamadas (MLP) surgiu como uma evolução do Perceptron do tipo *McCulloch&Pitt*. Ele tem sua arquitetura organizada da seguinte forma: um conjunto de unidades de entrada, uma ou mais camadas intermediárias contendo unidades de processamento (camadas ocultas) e um conjunto de unidades de saída. Essa arquitetura é ilustrada na Figura 9. Com o advento do algoritmo de retropropagação, as redes MLPs tornaram-se muito populares e passaram a ser utilizadas em diversas aplicações do mundo real resolvendo problemas de classificação e previsão.

O MLP possui três características únicas: a não-linearidade é suave, contém uma ou mais camadas de neurônios ocultos (aqueles que não fazem parte da camada de entrada ou de saída) e exibe um alto grau de conectividade. *“É através da combinação destas características, juntamente com a habilidade de aprender da experiência através do treinamento, que a MLP deriva o seu poder computacional”* [Haykin, 2001].

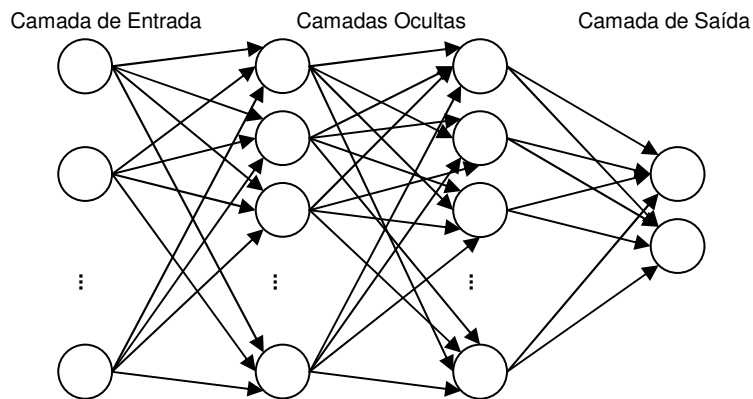


Figura 9: Arquitetura da MLP

2.5.1.1 O processo de Treinamento da MLP

O processo de treinamento da rede MLP apresenta dois passos distintos. No primeiro passo – passo para frente – um padrão de entrada é apresentado para a respectiva camada, fazendo os neurônios das camadas ocultas computarem suas funções de ativação, gerando respostas que são propagadas para a camada seguinte. Os neurônios da camada seguinte, por sua vez, computam as suas funções, e o processo se repete sucessivamente até que a resposta final da rede seja gerada pela camada de saída. As unidades de processamento são definidas, geralmente, por uma regra de propagação (estado de ativação) e por uma função de ativação. Estes elementos variam de acordo com a configuração da rede. No segundo passo, é utilizado o algoritmo de retropropagação, onde o sinal do erro produzido pela rede é retropropagado às unidades da rede, até que não haja mais uma diminuição no sinal do erro produzido.

Capítulo 3 - O Problema de Previsão

Neste capítulo, serão abordados o problema de Previsão utilizando Redes Neurais Artificiais, a aplicação considerada neste trabalho – Previsão de Índices de Ações da Bolsa de Valores – e a base de dados utilizada na investigação experimental.

3.1 As RNAs e o Problema de Previsão

A observação de eventos no decorrer do tempo demonstra que estes eventos, mesmo não sendo exatamente idênticos, também não são completamente diferentes. Há uma linha de continuidade, similaridade e previsibilidade nestes eventos que permite generalizar eventos futuros, freqüentemente de modo correto, a partir de experiências passadas [Mueller, 1996].

As RNAs têm apresentado bons resultados em tarefas envolvendo previsão. Por possuírem a capacidade de generalização, adaptação, correlação e aprendizado, facilitam o tratamento com séries temporais. Elas podem extrair as características mais importantes da série, como sazonalidade e tendência. Também correlacionam outras variáveis que se mostram importantes para a série. Segundo Yonenaga [Yonenaga, 2000], existem vários exemplos de aplicações de redes neurais para previsão das séries temporais, como: previsão de números de passageiros em aviões, previsão de índices de mercado, previsão de vazão de rios, previsão de vendas de produtos no comércio, previsão de cotação de moeda, etc.

A partir de uma série histórica, é possível montar uma rede neural que forneça, com alto grau de certeza, o valor futuro da série. No caso do preço de ações da Bolsa de Valores do Estado de São Paulo (BOVESPA), tal informação é de grande importância para investidores, empresários e consultores, pois, tendo

uma boa previsão para o preço no dia posterior, poderão tomar as decisões apropriadas, como vender ou comprar ações, por exemplo. Também poderá servir para traçar o comportamento da série de preços no mercado futuro. Essas informações podem trazer certas vantagens competitivas para os investidores em geral em relação a seus concorrentes.

3.2 Mercado de Ações

Uma empresa utiliza-se das ações como estratégia de captação de recursos financeiros para apoiar seus projetos de expansão. Estes recursos são devolvidos ao investidor através da distribuição de dividendos e da retenção de lucros. Os investidores também podem obter ganhos com a valorização do preço das ações no mercado. *“O mercado de ações, segmento do mercado de capitais que opera com títulos de renda variável, objetiva canalizar recursos para as empresas através do capital de risco”* [Mueller, 1996].

3.2.1 Ação

“A ação é um título representativo da fração unitária do capital social de uma sociedade anônima ou companhia, denominação deste tipo de empresa na legislação brasileira, conferindo a seu proprietário o direito de participação nessa sociedade” [Mueller, 1996]. Elas estão subdivididas em duas espécies: as ordinárias – dão o direito a voto nas assembleias de acionistas – e as preferenciais – não dão o direito a voto, mas garantem a prioridade e preferência na distribuição de dividendos, no reembolso de capital no caso de dissolução da empresa e na acumulação destas vantagens.

As ações também são classificadas em duas formas: as nominativas – são aquelas cujos certificados identificam seu titular, também inscrito no Livro de Registro de Ações Nominativas – e as escriturais – não são representadas por certificados, sendo títulos mantidos em conta de depósito, em nome de seus

titulares, na instituição financeira que o estatuto da empresa designar.

Os investidores do mercado acionário têm a expectativa de ganhos decorrentes do rendimento das ações adquiridas e de sua valorização no mercado. Quem compra ações, mesmo inconscientemente, especula sobre a possibilidade de rendimentos futuros, o comportamento do mercado e a possibilidade de valorização do papel.

3.3 Formação da Base de Dados

A partir de uma base de dados, é possível modelar uma rede neural para diversas aplicações. *“De posse das informações relevantes do problema e com programa [...] que contenha um algoritmo de treinamento, é possível desenvolver uma rede neural”* [Yonenaga, 2000].

O desenvolvimento de redes neurais artificiais está dividido em dois passos: a coleta de dados relativos ao problema e a sua separação em um conjunto de treinamento e um conjunto de teste. Estas tarefas necessitam de análises cuidadosas sobre o problema para minimizar ambigüidades e erros nos dados. Além disso, os dados coletados devem ser significativos e cobrir amplamente o domínio do problema. Os dados de treinamento serão utilizados para o treinamento da rede e os dados de teste serão utilizados para verificar seu desempenho sob condições reais de utilização. Os dados de testes não devem ser utilizados no momento do treinamento.

Depois de determinados estes conjuntos, eles são, geralmente, colocados em ordem aleatória para prevenção de tendências associadas à ordem de apresentação dos dados. Além disso, pode ser necessário pré-processar estes dados, através de normalizações, escalonamentos e conversões de formato para torná-los mais apropriados à sua utilização na rede.

“Se os dados podem ser representados numericamente, eles podem ser utilizados como entradas para as Redes Neurais Artificiais” [Bosaipo, 2001]. Portanto

as informações técnicas e fundamentais relacionadas a um mercado específico podem ser incorporadas como entradas às redes neurais.

Os dados para se realizar os experimentos, neste trabalho, foram obtidos através do sistema de informação “Enfoque Cotações, GOL – Gráficos On-line, versão 5.0”, que os capturou diretamente dos fornecedores de informação eletrônica da BOVESPA.

3.3.1 O Arquivo de Dados Original

O arquivo de dados originalmente gerado pelo GOL foi exportado em formato texto com os campos separados por tabulações, conforme apresentado na Tabela 2. Este arquivo contém os valores correspondentes às cotações da ação preferencial nominativa da empresa Telemar (TNLP4), conhecida na BOVESPA pela sigla TNLP4, do dia 17 de novembro de 2000 até o dia 01 de outubro de 2004, conforme apresentado no Gráfico 1. O total de registros contidos no arquivo é de 963, pois apenas são arquivados os dados dos dias úteis. Não existem pregões em feriados e finais de semanas.

Data	Hora	Abertura	Máximo	Mínimo	Fechamento	Volume
12/01/2001	00:00	40.7381782531738	41.4115371704102	39.1389465332031	39.3072929382324	2755145216
15/01/2001	00:00	39.4756317138672	39.9806442260742	39.2231254577637	39.6439590454102	497803264
16/01/2001	00:00	39.6018829345703	39.8123168945313	38.6339263916016	38.6760139465332	1601524736
17/01/2001	00:00	39.3072929382324	40.6540145874023	39.1389465332031	40.3173408508301	2528222720
18/01/2001	00:00	40.6119232177734	41.4957046508789	40.2163200378418	40.9906845092773	2631585280
19/01/2001	00:00	41.0748558044434	41.7061309814453	40.5698394775391	41.1674423217773	1693006464

Tabela 2: Amostra de dados do arquivo gerado pelo GOL

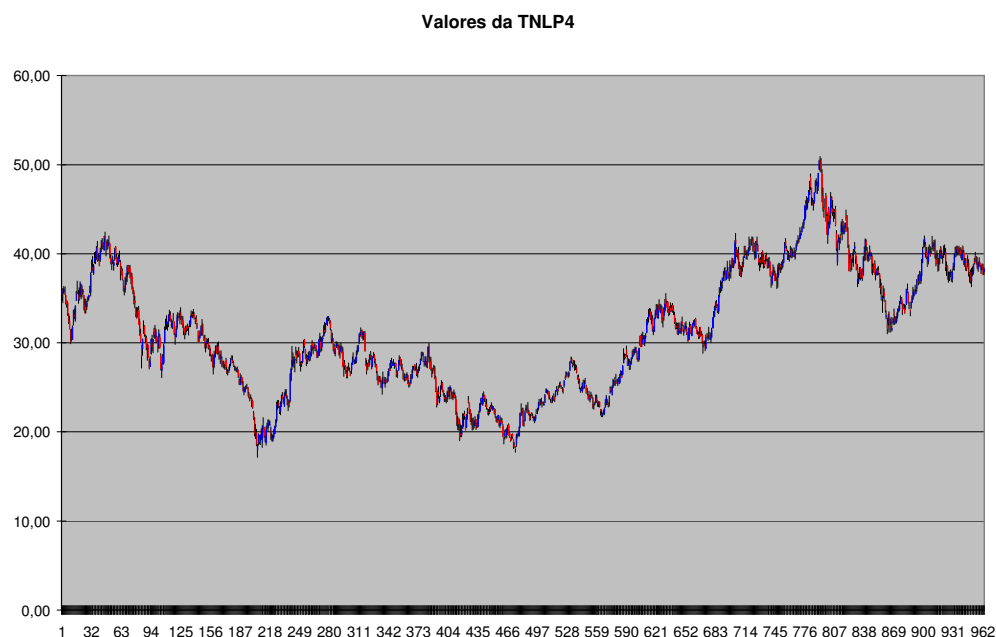


Gráfico 1: Representação gráfica do tipo *Candles* dos valores coletados da TNLP4

Os atributos do arquivo gerado pelo GOL são respectivamente: a data do pregão, a hora da cotação (que vem com valor nulo), o valor da ação no momento da abertura do pregão, o valor máximo negociado durante o pregão, o valor mínimo negociado durante o pregão, o valor da ação no momento do fechamento do pregão e o volume de ações negociadas durante o pregão.

3.3.2 Técnicas Aplicadas para Ajustar a Base de Dados

As RNAs precisam de valores numéricos para que possam funcionar corretamente. As técnicas utilizadas para o ajuste dos dados foram basicamente quatro: a troca do caractere identificador de ponto flutuante de “.” para “,”; a retirada de valores simbólicos nulos; o arredondamento dos valores de ponto flutuante para apenas duas casas decimais e a transformação das datas em valores binários.

Após a aplicação das três primeiras técnicas os dados ficaram formatados de acordo com a Tabela 3.

Data	Abertura	Máximo	Mínimo	Fechamento	Volume
12/1/2001	40,74	41,41	39,14	39,31	2755145216
15/1/2001	39,48	39,98	39,22	39,64	497803264
16/1/2001	39,60	39,81	38,63	38,68	1601524736
17/1/2001	39,31	40,65	39,14	40,32	2528222720
18/1/2001	40,61	41,50	40,22	40,99	2631585280
19/1/2001	41,07	41,71	40,57	41,17	1693006464

Tabela 3: Amostra de dados após aplicação de técnicas de ajustes

Durante a aplicação da quarta técnica, que é a de transformação das datas em valores binários, foram definidos novos campos, são eles: dias da semana com as representações binárias para segunda, terça, quarta, quinta e sexta-feira; a dezena do dia, com as representações binárias para identificar se o número do dia da referente data tem como dezena o numeral 0, 1, 2 ou 3; a unidade do dia, com as representações binárias para identificar se o número do dia da referente data tem como unidade o numeral 0, 1, 2, 3, 4, 5, 6, 7, 8 ou 9 e os meses do ano com suas representações binárias para identificar se o número do mês da referente data tem como unidade os valores 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 ou 12. Os valores referentes aos anos de cada data foram descartados, para que a rede pudesse identificar padrões de comportamento relacionados aos dias da semana, aos dias e aos meses do ano [Fernandes, 1995].

Após a aplicação da quarta técnica, os valores das datas ficaram formatados de acordo com a Tabela 4.

Data	Dias da Semana					Dezena do dia				Unidades do dia									Unidades do mês												
	2	3	4	5	6	0	10	20	30	0	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	10	11	12
12/1/2001	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
15/1/2001	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
16/1/2001	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
17/1/2001	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
18/1/2001	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
19/1/2001	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

Tabela 4: Amostra de dados após aplicação da quarta técnica de ajustes

Após a finalização dos ajustes, a base de dados passou a conter o total de 41 atributos, sendo 31 relacionados aos valores binários da data, 5 aos valores de abertura, máximo, mínimo, fechamento, volume e mais outros 5, com 1 para o percentual de retorno do dia e 4 para os valores dos fechamentos de quatro dias anteriores, ou seja, sendo $Fechamento(t)$ o valor do fechamento do dia atual, foram inseridos os valores de $Fechamento(t-1)$ até $Fechamento(t-4)$, para que a rede possa obter memória de curto prazo [Mello, 2004].

O campo do percentual de retorno foi calculado de acordo com a Equação 13.

$$r_t(f_t, a_t) = \frac{f_t - a_t}{a_t}$$

Equação 13: Equação para cálculo do percentual de retorno diário

onde f_t é o valor da ação no fechamento do pregão do dia t , e a_t é o valor da ação na abertura do pregão do dia t .

À base de dados final foi dado o nome *Trade* e estará disponível, para que outros pesquisadores a possam utilizar em seus trabalhos.

Capítulo 4 – A Ferramenta Previsão de Ativos

Neste capítulo, será descrita a ferramenta implementada durante esta monografia – Previsão de Ativos – a qual foi utilizada na realização dos experimentos com a base de dados *Trade*. A descrição detalhada da ferramenta Previsão de Ativos serve como um tutorial para futuros usuários da mesma.

4.1 Arquitetura e Ferramentas para o Desenvolvimento

A ferramenta Previsão de Ativos foi implementada sob a arquitetura Cliente /Servidor, utilizando os conceitos de orientação a objetos, com a ferramenta de desenvolvimento *Borland Delphi 7*, linguagem de programação *Object Pascal* e banco de dados *MS Access*.

Para apoiar o desenvolvimento do programa, foi criada uma pequena infraestrutura composta por dois pacotes básicos. O primeiro contém as classes básicas e foi nomeado *MVBFrameworkCS*. O segundo, chamado de *MVBRedesNeurais*, incorpora as classes utilizadas para a criação das redes neurais.

4.2 Componente MLP

As classes básicas da infra-estrutura são: *TMVBDadosBase*, *TMVBListaBase* e *EMVBException*. Todas as classes do sistema são especializações destas três classes.

A classe *TMVBDadosBase* foi utilizada como generalização para as todas as outras que têm como objetivo a persistência de valores. Essa classe tem a propriedade *Id* do tipo literal que identifica unicamente cada objeto instanciado.

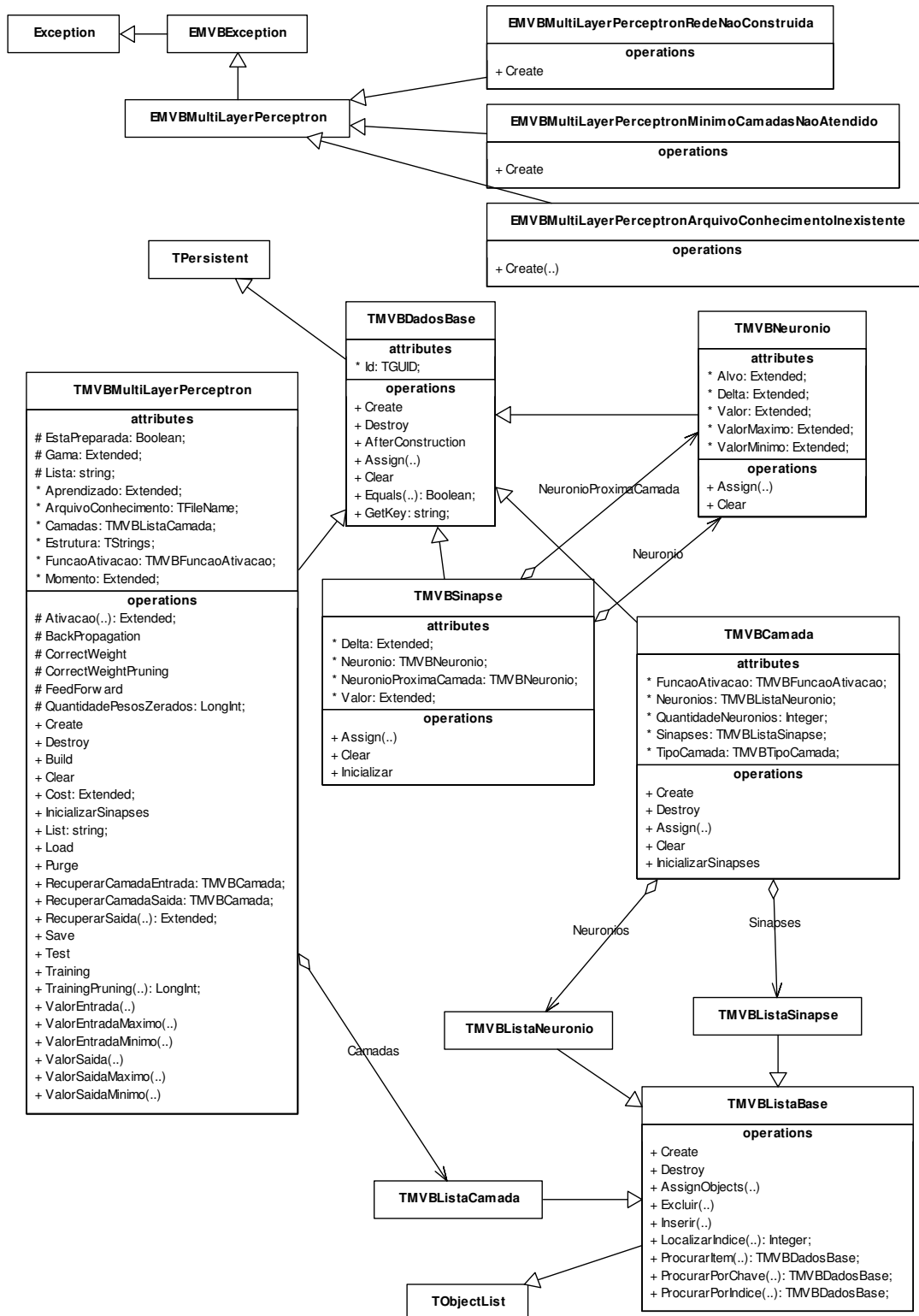
Todas as classes que têm como objetivo persistir um conjunto de valores em forma de listagem foram herdadas da classe *TMVBListaBase*.

As exceções participantes do sistema são herdadas da classe *EMVBException*.

As classes de infra-estrutura desenvolvidas para a implementação da rede MLP são: *TMVBNeuronio*, *TMVBSinapse*, *TMVBCamada*, *TMVBMultiLayerPerceptron*, *TMVBListaCamada*, *TMVBListaNeuronio* e *TMVBListaSinapse*. As quatro primeiras são especializações da *TMVBDadosBase* e as outras especializações da *TMVBListaBase*.

A implementação destas classes foi baseada no componente *TMLP* disponibilizado por Luciano F. de Medeiros em [Medeiros, 2003].

4.2.1 Diagrama de Classes



4.2.2 Classe *TMVBNeuronio*

Essa classe tem como objetivo agrupar as informações de um neurônio artificial. Estas informações são:

- Valor assumido pelo neurônio: representado pela propriedade *Valor* do tipo numérico com ponto flutuante (*Extended*);
- Valor do gradiente descendente: representado pela propriedade *Delta* do tipo *Extended*;
- Valor de alvo: utilizado somente para última camada – representado pela propriedade *Alvo* do tipo *Extended*;
- Valor máximo assumido pelo neurônio: representado pela propriedade *ValorMaximo* do tipo *Extended*;
- Valor mínimo assumido pelo neurônio: representado pela propriedade *ValorMinimo* do tipo *Extended*.

4.2.3 Classe *TMVBSinapse*

Essa classe tem o objetivo de agrupar as informações de uma sinapse artificial. Estas informações são:

- Valor assumido pela sinapse: representado pela propriedade *Valor* do tipo *Extended*;
- Referência ao delta de correção dos pesos sinápticos: representada pela propriedade *Delta* do tipo *Extended*;
- Referência do primeiro neurônio: representado pela propriedade *Neurônio* do tipo *TMVBNeuronio*;
- Referência do neurônio da próxima camada: representado pela propriedade *NeuronioProximaCamada* do tipo *TMVBNeuronio*.

Esta classe também contempla o método *Inicializar* que serve para inicializar os atributos *Valor* e *Delta* da sinapse.

4.2.4 Classe *TMVBCamada*

Essa classe tem o objetivo de agrupar informações de uma camada de uma rede neural artificial MLP. Estas informações são:

- Tipo da camada: representado pela propriedade *TipoCamada* do tipo enumerado *TMVBTipoCamada*, podendo assumir os seguintes valores (desconhecido, entrada, escondida e saída);
- Quantidade de neurônios da camada: representado pela propriedade *QuantidadeNeuronios* do tipo numeral inteiro (*Integer*);
- Lista de neurônios: representada pela propriedade *Neuronios* do tipo *TMVBListaNeuronio*;
- Lista de sinapses: representada pela propriedade *Sinapses* do tipo *TMVBListaSinapse*;
- Função de ativação: representada pela propriedade *FuncaoAtivacao* do tipo enumerado *TMVBFuncaoAtivacao*, podendo assumir os seguintes valores (não informada, linear por partes, sigmóide, sigmóide bipolar, tangente hiperbólica).

4.2.5 Classe *TMVBMultiLayerPerceptron*

Essa classe tem o objetivo de agrupar informações de uma rede neural artificial MLP. Estas informações são:

4.2.5.1 **Propriedades**

- Definição da quantidade de neurônios por camada: representada pela propriedade *Estrutura* do tipo lista de literais (*TStrings*). Cada

valor informado por linha representa a quantidade de neurônios da camada, ou seja, se, na primeira linha for informado o número 41, então, serão criados 41 neurônios na camada de entrada. Se existirem apenas duas linhas, então, a rede será criada sem camadas escondidas, mas somente com a camada de entrada e a de saída.

- Definição do parâmetro de momento para atualização de pesos sinápticos: representada pela propriedade *Momento* do tipo *Extended*. Essa propriedade “[...] define a quantidade que será aproveitada do peso sináptico na época anterior. O valor assumido fica no intervalo [0, 1]. Caso assuma zero, nenhuma proporção do valor do peso assumido na época anterior será considerada” [Medeiros, 2003].
- Definição do parâmetro de aprendizado para atualização de pesos sinápticos: representada pela propriedade *Aprendizado* do tipo *Extended*. Essa propriedade “[...] define a proporção que será aproveitada do gradiente de atualização calculado vezes o valor de saída do neurônio. O valor assumido fica no intervalo [0, 1]. Caso assuma um, o impacto da atualização sobre o peso será máximo” [Medeiros, 2003].
- Definição do nome do arquivo de conhecimento que serão gravados os valores dos pesos sinápticos: representado pela propriedade *ArquivoConhecimento* do tipo literal específico para nome de arquivos (*TFileName*). Aqui estará o nome do arquivo que guardará os valores das conexões sinápticas após o treinamento. São utilizados os métodos *Save* e *Load* para salvar e carregar os valores respectivamente.
- Lista de camadas da rede: representada pela propriedade *Camadas* do tipo *TMVListaCamada*. Nesta lista, estarão as instâncias dos objetos de camadas, neurônios e sinapses que serão criados após a utilização do método *Build*.

- Definição da função de ativação utilizada pela MLP: representada pela propriedade *FuncaoAtivacao* do tipo enumerado *TMVBFuncaoAtivacao*.

4.2.5.2 Métodos

- Função de ativação: representada pela função *Ativacao* que recebe os parâmetros *AValor* do tipo *Extended* e *AFuncaoAtivacao* do tipo *TMVBFuncaoAtivacao*, que são respectivamente o valor do neurônio e a função de ativação selecionada para a camada. Essa função retorna um valor numérico do tipo *Extended*.
- Calcular os pesos e as saídas avançando na rede para frente: representado pelo procedimento *FeedForward*.
- Calcular os deltas após o *feedforward*: representado pelo procedimento *BackPropagation*.
- Correção de pesos após *backpropagation*: representado pelo procedimento *CorrectWeight*.
- Informar os valores para os neurônios da primeira camada: representado pelo procedimento *ValorEntrada* que recebe os parâmetros *AIndiceNeuronio* do tipo *Integer* e *AValor* do tipo *Extended*. O primeiro parâmetro recebe a localização do neurônio na lista de neurônios da camada de entrada – que inicia por zero, ou seja, para informar o valor 10 para o primeiro neurônio de entrada, os parâmetros deverão ser *AIndiceNeuronio* = 0 e *AValor* = 10.
- Informar o valor máximo que poderá ser assumido pelos neurônios da primeira camada: representado pela função *ValorEntradaMaximo* que recebe os parâmetros *AIndiceNeuronio* do tipo *Integer* e *AValor* do tipo *Extended*. “Estes valores são necessários para a

normalização da entrada da rede [...], pois, internamente ela trabalha na faixa de valores [0, 1] [Medeiros, 2003].

- Informar o valor mínimo que poderá ser assumido pelos neurônios da primeira camada: representado pela função *ValorEntradaMinimo* que recebe os parâmetros *AIndiceNeuronio* do tipo Integer e *AValor* do tipo Extended. “*Estes valores são necessários para a normalização da entrada da rede [...], pois, internamente, ela trabalha na faixa de valores [0, 1]*” [Medeiros, 2003].
- Informar os valores para os neurônios da última camada: representado pelo procedimento *ValorSaida* que recebe os parâmetros *AIndiceNeuronio* do tipo Integer e *AValor* do tipo Extended. O primeiro parâmetro recebe a localização do neurônio na lista de neurônios da camada de saída – que inicia por zero, ou seja, para informar o valor 10 para o primeiro neurônio de saída, os parâmetros deverão ser *AIndiceNeuronio* = 0 e *AValor* = 10.
- Informar o valor máximo que poderá ser assumido pelos neurônios da última camada: representado pela função *ValorSaidaMaximo* que recebe os parâmetros *AIndiceNeuronio* do tipo Integer e *AValor* do tipo Extended. “*Estes valores são necessários para a normalização da entrada da rede [...], pois, internamente ela trabalha na faixa de valores [0, 1]*” [Medeiros, 2003].
- Informar o valor mínimo que poderá ser assumido pelos neurônios da última camada: representado pela função *ValorSaidaMinimo* que recebe os parâmetros *AIndiceNeuronio* do tipo Integer e *AValor* do tipo Extended. “*Estes valores são necessários para a normalização da entrada da rede [...], pois, internamente ela trabalha na faixa de valores [0, 1]*” [Medeiros, 2003].
- Recuperar os valores das saídas gerados pela rede: representado

pela função *RecuperarSaida* que recebe o *AIndiceNeuronio* do tipo *Integer* e retorna o valor do tipo *Extended* do neurônio selecionado da camada de saída.

- Construir a rede: representado pelo procedimento *Build*. Este método constrói a rede a partir das informações contidas na propriedade *Estrutura*.
- Executar o treinamento da rede: representado pelo procedimento *Training*. Este método executa o treinamento de forma supervisionada, ou seja, é necessário informar os valores das entradas e saídas durante o processo de treinamento. Ele também atualiza o erro global na função *Cost*. “As propriedades *Momento* e *Aprendizado* são parâmetros que influenciam diretamente na convergência do treinamento” [Medeiros, 2003].
- Executar o teste na rede: representado pelo procedimento *Test*. Este método deve ser utilizado após a fase de treinamento ou com uma rede já treinada anteriormente e carregada com o método *Load*.
- Inicializar todos os valores da rede: representado pelo procedimento *Clear*. Este método inicializa cada uma das camadas, liberando seus neurônios e sinapses além de limpar sua estrutura. Ele também atribui valores iniciais de 0,9 e 0,5 para as propriedades *Aprendizado* e *Momento* respectivamente.
- Atribuir valores iniciais para as sinapses: representado pelo procedimento *InicializarSinapses*. Esse método atribui o valor zero para todas as sinapses da rede.
- Recuperar a última camada: representado pela função *Recuperar-CamadaSaida* que retorna um objeto do tipo *TMVBCamada*. Esse método retorna a referência do objeto relacionado à camada de

saída da propriedade *Camadas*.

- Recuperar a camada de entrada: representado pela função *RecuperarCamadaEntrada* que retorna um objeto do tipo *TMVB-Camada*. Esse método retorna a referência do objeto relacionado à camada de entrada da propriedade *Camadas*.
- Salvar o arquivo com os valores da rede das sinapses da rede e suas configurações: representado pelo procedimento *Save*. Esse método deve ser chamado após a fase de treinamento.
- Carregar os valores sinapses do arquivo de conhecimento e suas configurações: representado pelo procedimento *Load*. Esse método carrega a estrutura da rede, os valores de aprendizado, do momento e de todas as sinapses. Se a rede não estiver construída, então, ele o faz.
- Retornar o valor da função custo, ou seja, do erro atual: representado pela função *Cost* que retorna um valor numérico do tipo *Extended*. Essa função retorna o valor do erro na época ou iteração atual. A chamada do método serve para monitorar a convergência do gradiente descendente e também pode sinalizar o término do processo de treinamento.

4.3 Validação do Componente MLP Desenvolvido

Antes de ser utilizado nos testes com a base de dados *Trade*, o componente *TMVBMultiLayerPerceptron* foi testado com outra aplicação mais simples - Previsão de Consultas em *Home-Pages* - cuja base de dados foi obtida através da referência [Medeiros, 2003]. Os resultados obtidos validaram o componente que gerou resultados muito próximos ao reportados na referência.

4.4 Aplicação para Previsão de Ativos da BOVESPA

Para adequar à aplicação de previsão de índices da bolsa de valores, estudo de caso deste trabalho, a ferramenta desenvolvida precisou contemplar três métodos de treinamento distintos, identificados como *Trein 1*, *Trein 2* e *Trein 3*, conforme a Figura 10. Os três métodos utilizam a rede MLP com retropropagação como processo de aprendizagem. O método *Trein 1* foi baseado nos estudos de Luiz Gustavo L. Fernandes em [Fernandes, 1995] que propõe um padrão de entrada com os valores binários das datas e os preços da ação. O método *Trein 2* foi uma alteração do método *Trein 1*, incluindo apenas valores passados de $t-1$ a $t-4$ para dar a rede memória de curto prazo. O terceiro método de treinamento, o *Trein 3*, é baseado nos estudos de Luciano Medeiros [Medeiros, 2003] que, por analogia, propõe um padrão de entrada utilizando apenas os preços da ação.

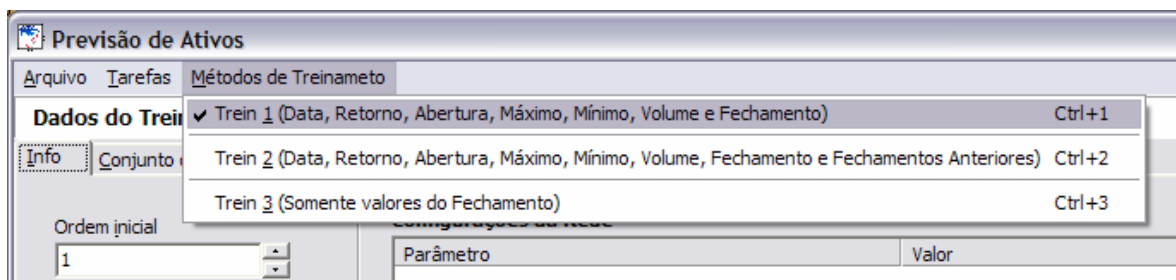


Figura 10: Métodos de treinamento da ferramenta

4.4.1 Método *Trein 1*

Este método recebe como padrões de entrada os valores binários das datas (neurônios de 0 a 30), o retorno obtido no dia (neurônio 31), o preço da ação no momento da abertura do pregão (neurônio 32), o preço máximo alcançado pela ação (neurônio 33), o preço mínimo (neurônio 34), o volume de títulos negociados (neurônio 35) e o preço de fechamento da ação (neurônio 36). Dependendo da configuração dos parâmetros, pode-se incluir uma camada oculta

com um número variável de neurônios. Ele apresenta apenas um neurônio na camada de saída que corresponde ao valor da previsão, conforme apresentado na Figura 11.

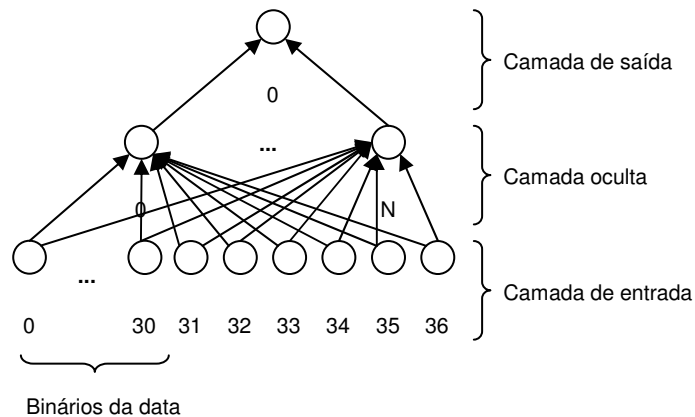


Figura 11: Arquitetura da rede - Método *Trein 1*

4.4.2 Método *Trein 2*

Este método recebe como padrões de entrada os valores binários das datas (neurônios de 0 a 30), o retorno obtido no dia (neurônio 31), o preço da ação no momento da abertura do pregão (neurônio 32), o preço máximo alcançado pela ação (neurônio 33), o preço mínimo (neurônio 34), o volume de títulos negociados (neurônio 35), o preço de fechamento da ação (neurônio 36) e os preços de fechamento nos momentos (t-1) até (t-4) (neurônios 37 a 40). Dependendo da configuração dos parâmetros, pode-se incluir uma camada oculta com um número variável de neurônios. Ele apresenta apenas um neurônio na camada de saída que corresponde ao valor da previsão, conforme apresentado na Figura 12.

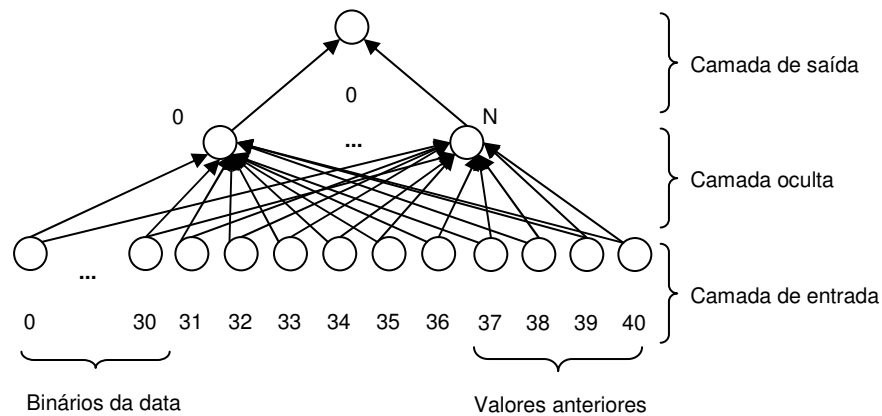


Figura 12: Arquitetura da rede - Método Trein 2

4.4.3 Método Trein 3

Este método recebe como padrões de entrada os valores dos preços de fechamento nos momentos $(t-n)$ até $(t-1)$ e o valor do preço de fechamento da ação, conforme a Figura 14. A quantidade de neurônios da camada de entrada não é fixa e pode ser escolhida no momento da seleção deste método (Figura 13). Dependendo da configuração dos parâmetros, pode-se incluir uma camada oculta com um número variável de neurônios. Ele apresenta apenas um neurônio na camada de saída que corresponde ao valor da previsão.

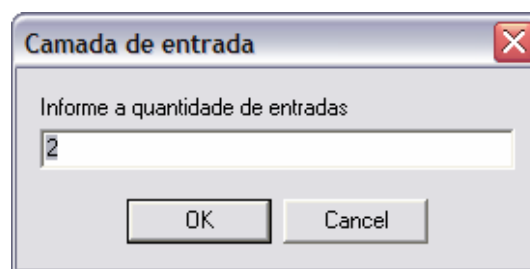


Figura 13: Seleção de neurônios da camada de entrada – Método Trein 3

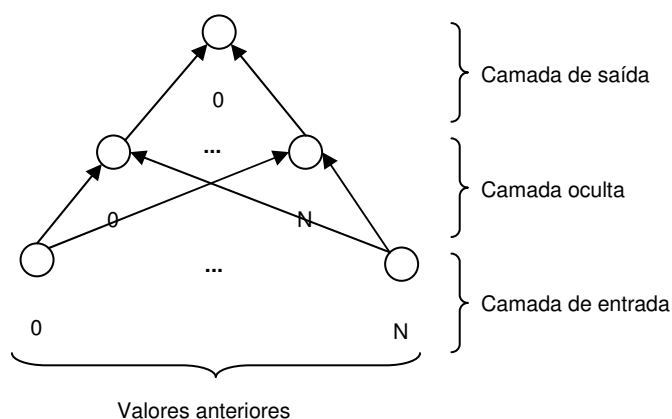


Figura 14: Arquitetura da rede - Método *Trein 3*

4.4.4 Configuração de Parâmetros

Para que sejam alcançados resultados satisfatórios com a rede MLP, faz-se necessária a configuração de parâmetros importantes como: a topologia da rede, a função de ativação, a taxa de aprendizagem, número de ciclos e a taxa de momentum. Para isso, a ferramenta contempla uma interface contendo os parâmetros a serem ajustados, conforme mostra a Figura 15.

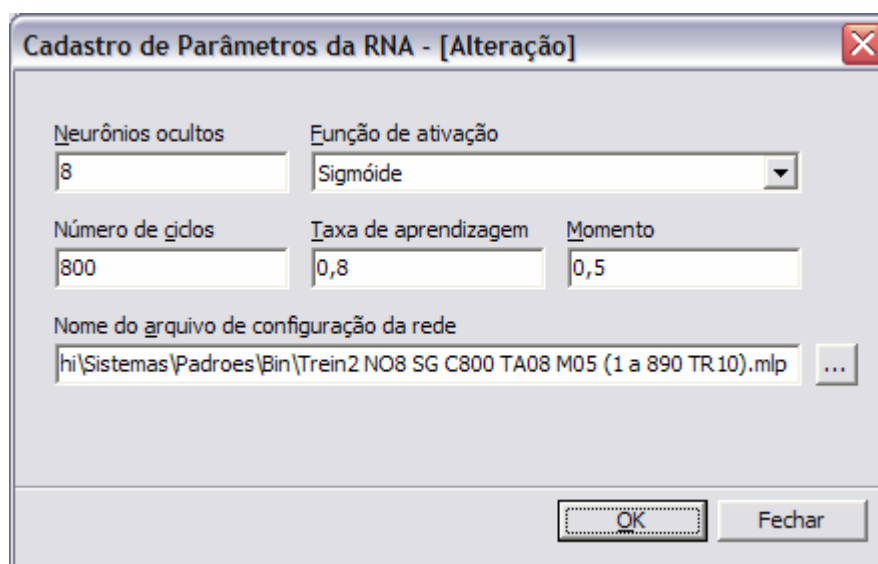


Figura 15: Tela de configuração de parâmetros da MLP

O acesso à funcionalidade pode ser obtido através do menu *Arquivo* ou pela tecla de atalho *F8*, conforme apresentado na Figura 16.

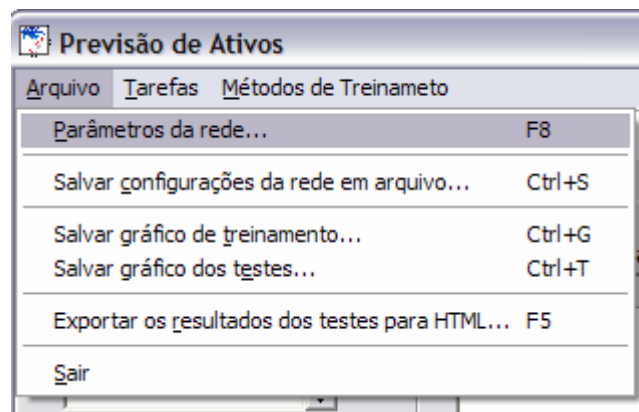


Figura 16: Menu para acessar a configuração dos parâmetros da rede

4.4.5 Inicialização da MLP

Com a base de dados *Trade* devidamente carregada pela ferramenta, para se inicializar a MLP, devem ser executados três passos abaixo:

- Passo 1 – Escolha do método de treinamento;
- Passo 2 – Configuração dos parâmetros²;
- Passo 3 – Escolha do conjunto de dados para o treinamento.

Antes de serem apresentados à rede, os dados são separados. Na execução do Passo 3, faz-se a divisão dos grupos de dados em dois conjuntos: conjunto de dados de treinamento e conjunto de dados de teste. O conjunto de dados que será utilizado no treinamento da rede deve ser informado através dos campos: *Ordem Inicial* e *Final*, localizados na área *Dados do Treinamento* na guia *Info*, conforme ilustra a Figura 17.

² Deve-se sempre observar o nome do arquivo de configuração da rede, pois, se ele já existir, suas configurações serão carregadas e os parâmetros da rede, com exceção da função de ativação, serão descartados.

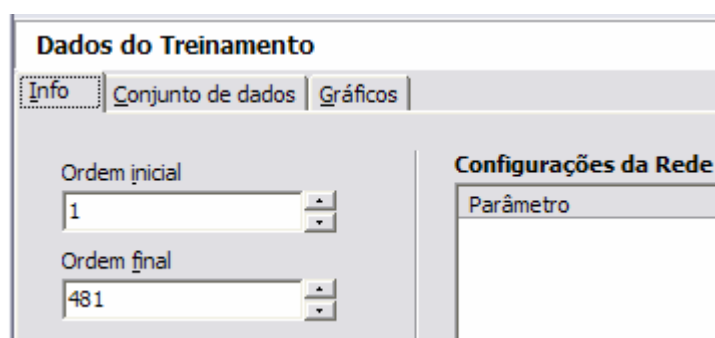


Figura 17: Área de seleção do conjunto de dados do treinamento

A construção da rede só deverá ser realizada após a execução dos três passos de inicialização. Para que a rede seja construída, basta utilizar a funcionalidade *Construir Rede* encontrada no menu *Tarefas* ou no botão localizado na área de *Dados do Treinamento*, conforme apresenta a Figura 18.

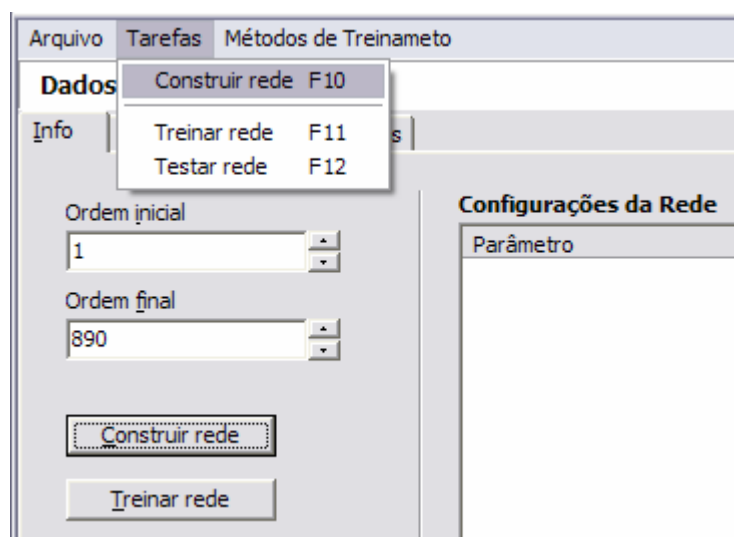


Figura 18: Apresentação da funcionalidade Construir Rede

Como pode ser visto na Figura 18, para o treinamento, selecionou-se o conjunto de dados no intervalo de 1 a 890, pois esse período apresenta os valores máximo e mínimo históricos registrados na base de dados, que vão de 17 de novembro 2000 a 18 de junho de 2004, conforme o Gráfico 2.

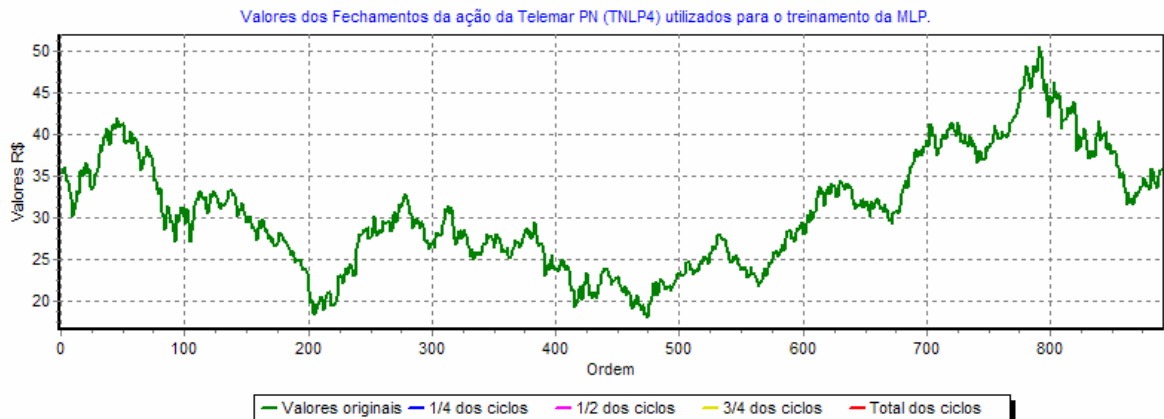


Gráfico 2: Valores da base de dados no intervalo de 1 a 890

Após a construção da rede, os valores são impressos na área de valores de configurações da rede localizado na opção Dados de Treinamento, conforme apresenta a Figura 19.

Configurações da Rede	
Parâmetro	Valor
Arquivo de conhecimento	D:\Docs\Vinicius\Desenv\Delphi\Delphi\Sistem
Quantidade de camadas	3
Quantidade de neurônios de entrada	41
Quantidade de neurônios ocultos	8
Quantidade de neurônios de Saída	1
Função de ativação	Sigmóide
Taxa de aprendizagem	0,800000
Momento	0,500000
Gama	0,000000
Número de Ciclos	800
Conjunto de treinamento	1 até 890

Figura 19: Listagem de valores de configurações da rede

4.4.6 Treinamento

O objetivo da rede neural, nesta aplicação, é a previsão do preço das ações para o dia seguinte. Para tal, o seu treinamento foi realizado da maneira

descrita a seguir.

Depois de finalizados os passos para inicialização e construção da rede, para treiná-la, deve-se apenas utilizar a funcionalidade *Treinar Rede* que pode ser encontrada no menu *Tarefas*, no botão localizado na área de *Dados do Treinamento* ou utilizando a tecla de atalho *F11*, conforme apresentado na Figura 20.

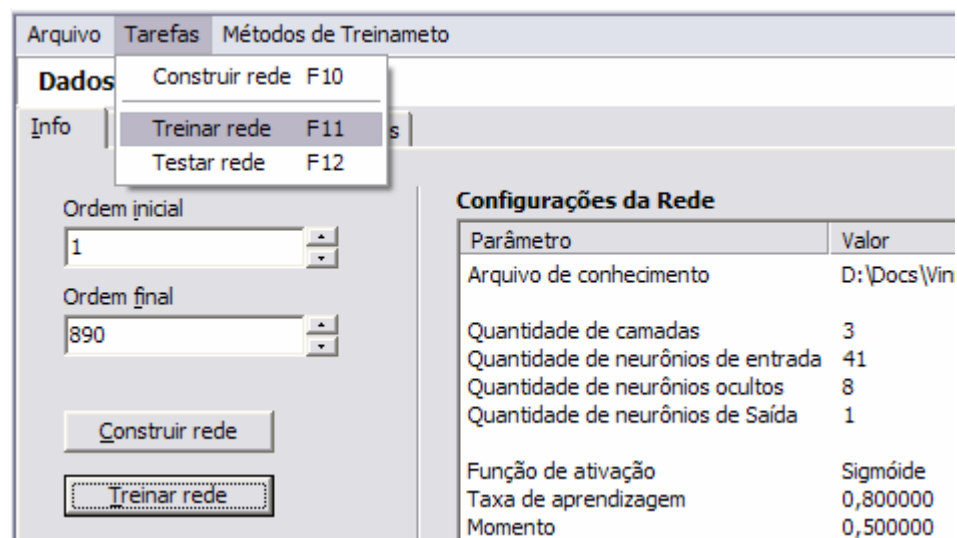


Figura 20: Apresentação da funcionalidade Treinar Rede

Durante o treinamento da rede, nos momentos de 1/4, 1/2 e 3/4 dos ciclos, são apresentadas algumas informações relacionadas ao erro, como o custo e o erro quadrático médio (MSE – *Mean Square Error*), conforme apresentado na Figura 21. Também são apresentadas graficamente as previsões feitas a partir dos valores utilizados no treinamento, conforme apresenta o Gráfico 3. O MSE é calculado de acordo com a Equação 14 e é uma boa medida de desempenho para redes com aprendizado supervisionado [Mello, 2004].

$$MSE(n) = \frac{1}{n} \times \sum_i^n (vo_i - vp_i)^2$$

Equação 14: Equação para cálculo do MSE

Onde n é o total de registros, vo é o valor original, e vp é o valor previsto pela rede.

Configurações da Rede	
Parâmetro	Valor
Erro Total (Cost) em ciclo = 200 e época = 177885. MSE em ciclo = 200 e época = 177885.	0,00002911 0,576510
Erro Total (Cost) em ciclo = 400 e época = 354885. MSE em ciclo = 400 e época = 354885.	0,00002647 0,504608
Erro Total (Cost) em ciclo = 600 e época = 531885. MSE em ciclo = 600 e época = 531885.	0,00002423 0,469665
Erro Total (Cost) em ciclo = 799 e época = 708000. MSE em ciclo = 799 e época = 708000.	0,00001053 0,444106

Figura 21: Apresentação de informações obtidas durante o treinamento

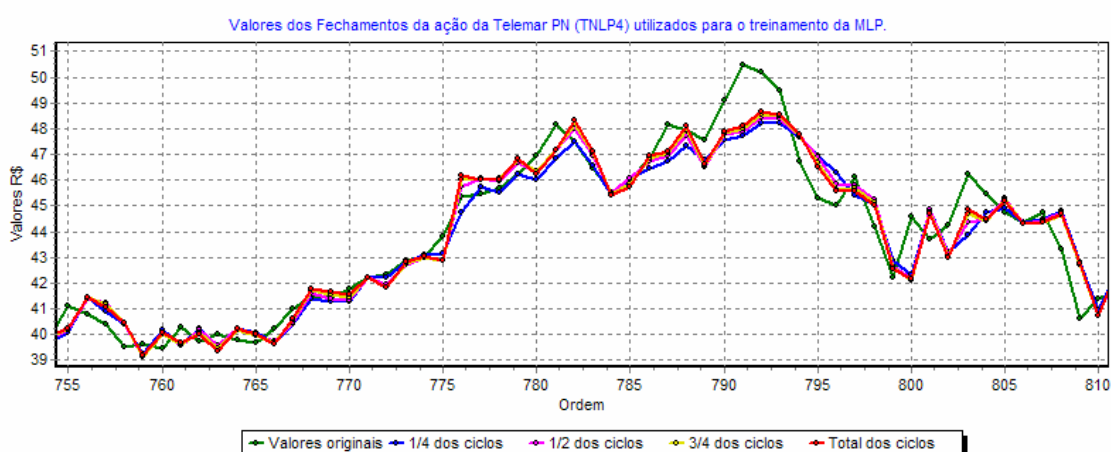


Gráfico 3: Amostra de valores obtidos durante o treinamento

4.4.7 Testes

Com a rede treinada, podem ser iniciados os testes. Para prosseguir, basta apenas informar o intervalo de valores que formam o conjunto de dados do teste através dos campos *Ordem Inicial* e *Final* localizados na guia *Info do Teste*, área de *Dados do Teste* e utilizar a funcionalidade Testar Rede, que pode ser

encontrada no menu *Tarefas*, no botão localizado na área de *Dados do Teste* ou utilizando a tecla de atalho *F12*, conforme ilustram as Figuras 22 e 23.

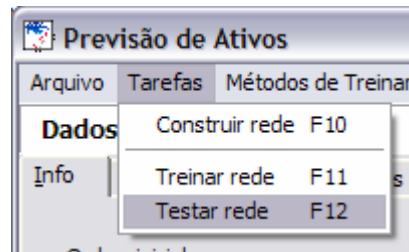


Figura 22: Menu Tarefas, Testar Rede

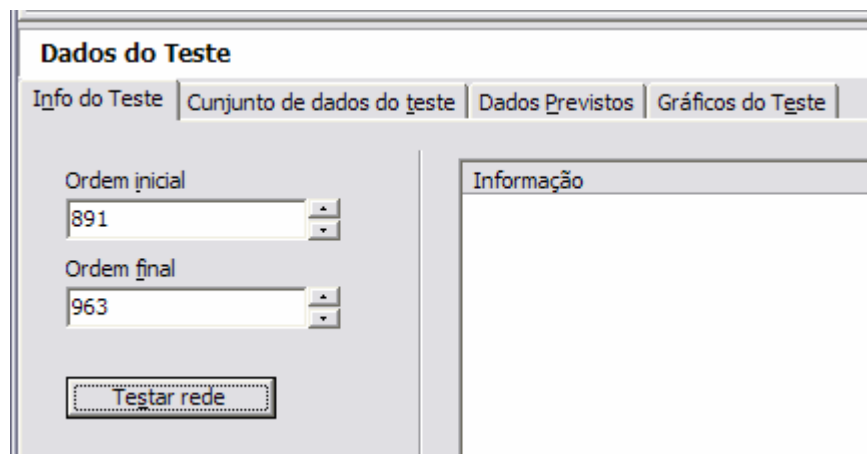


Figura 23: Apresentação da funcionalidade Testar Rede

Durante os testes, são coletadas algumas medidas importantes para avaliação dos resultados absolutos e relativos, são elas: o MSE, o erro médio (ME), o erro médio absoluto (MAE) e o erro percentual absoluto médio (MAPE) [Passari, 2003], como ilustra a Figura 24.

O ME é calculado de acordo com a Equação 15.

$$ME(n) = \frac{1}{n} \times \sum_i^n vo_i - vp_i$$

Equação 15: Equação para cálculo do ME

Onde n é o total de registros, vo é o valor original e vp é o valor previsto

pela rede.

O MAE é calculado de acordo com a equação 16.

$$MAE(n) = \frac{1}{n} \times \sum_i^n |vo_i - vp_i|$$

Equação 16: Equação para cálculo do MAE

Onde n é o total de registros, vo é o valor original e vp é o valor previsto pela rede.

O MAPE é calculado de acordo com a equação 17.

$$MAPE(n) = \frac{100}{n} \times \sum_i^n \left| \frac{vo_i - vp_i}{vo_i} \right|$$

Equação 17: Equação para cálculo do MAPE

Onde n é o total de registros, vo é o valor original e vp é o valor previsto pela rede.

Informação	Valor
ME (Erro Médio)	0,032693
MSE (Erro Médio Quadrado)	1,191888
MAE (Erro Médio Absoluto)	0,900341
MAPE (Erro Percentual Médio Absoluto)	2,29%

Figura 24: Resultados das medidas de avaliação do teste

Depois de finalizado o processo de teste, os resultados obtidos são apresentados em três quadros diferentes:

- Conjunto de dados do teste – Listagem que apresenta os valores do conjunto de dados selecionado para a execução dos testes (Figura 25).
- Dados previstos – Listagem que apresenta os valores previstos pela RNA (Figura 26).

- Gráfico do teste – Apresenta de forma gráfica os valores previstos obtidos durante o processo de teste (Figura 27).

Dados do Teste		
Info do Teste	Cunjunto de dados do teste	D
Ordem	Fechamento	
891	35,75	
892	35,90	
893	37,01	
894	37,00	
895	37,40	
896	36,70	
897	37,79	
898	39,15	
899	40,50	
900	41,30	
901	41,75	

Figura 25: Amostra do conjunto de dados do teste

Dados do Teste			
Info do Teste	Cunjunto de dados do teste	Dados Previstos	Gráficos do Teste
Ordem	Fechamento	Erro Absoluto %	Direção do erro
891	0,00	100,00	para baixo
892	0,00	100,00	para baixo
893	0,00	100,00	para baixo
894	0,00	100,00	para baixo
895	0,00	100,00	para baixo
896	36,93	0,63	para cima
897	36,05	4,60	para baixo
898	36,61	6,48	para baixo
899	38,45	5,07	para baixo
900	40,28	2,47	para baixo
901	40,69	2,53	para baixo

Figura 26: Amostra do conjunto de dados previstos no teste

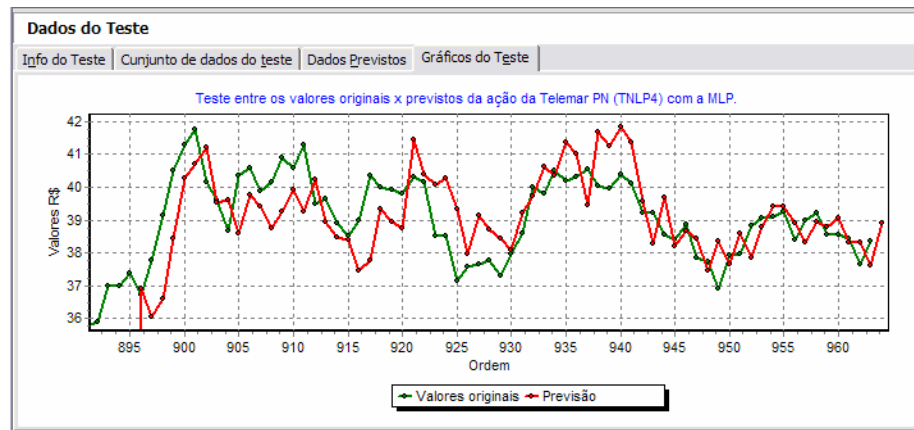


Figura 27: Gráfico de comparação entre os valores originais do teste e os previstos

Na página de dados previstos são apresentadas as colunas: Ordem, Fechamento, Erro Percentual Absoluto (APE) e Direção do Erro. Os valores da coluna Fechamento iguais a zero não são utilizados nos cálculos dos indicadores de desempenho, pois são apresentados à rede, anteriormente, como valores de entrada e não correspondem a resultados de previsão. O APE é calculado de acordo com a Equação 18.

$$APE = 100 \times \left| \frac{vo - vp}{vo} \right|$$

Equação 18: Equação para cálculo do APE

onde *vo* é o valor original e *vp* é o valor previsto pela rede.

A coluna direção do erro apresenta a informação retirada com a utilização da função de valor absoluto, ou seja, um valor indicando “para baixo” significa que o valor previsto está com a diferença em percentual abaixo do valor de fechamento original, funcionando com a mesma idéia na indicação “para cima”.

Capítulo 5 – Análise Comparativa dos Resultados

Neste capítulo, são apresentados os resultados obtidos através do procedimento experimental realizado com a ferramenta Previsão de Ativos (Capítulo 4) e a ferramenta *NeuroSolutions*³. Assim como no Capítulo 4, é realizada uma descrição detalhada dos passos seguidos na utilização da ferramenta *NeuroSolutions*, de forma a servir como um tutorial para futuros usuários da mesma.

5.1 A Ferramenta *NeuroSolutions*

A ferramenta *NeuroSolutions* foi escolhida para a realização dos experimentos, pois se trata de um aplicativo já consolidado no meio acadêmico e de fácil utilização. Esta ferramenta é capaz de simular vários tipos de arquiteturas e algoritmos e possui recursos poderosos para obtenção de informações da rede. Para iniciar a ferramenta, deve-se seguir os passos a seguir:

- Carregar a base *Trade*;
- Informar as colunas correspondentes às entradas;
- Informar a coluna relaciona à saída, para efetuar o treinamento supervisionado;
- Separar as linhas para treinamento;
- Separar as linhas para teste.

O acesso a cada uma destas funcionalidades pode ser feito através do menu *NeuroSolutions Tag Data* como ilustra a Figura 28.

³ A ferramenta *NeuroSolutions* é de autoria da empresa *Neuro Dimension*. Mais informações pode ser obtidas através da URL: <http://www.nd.com>.

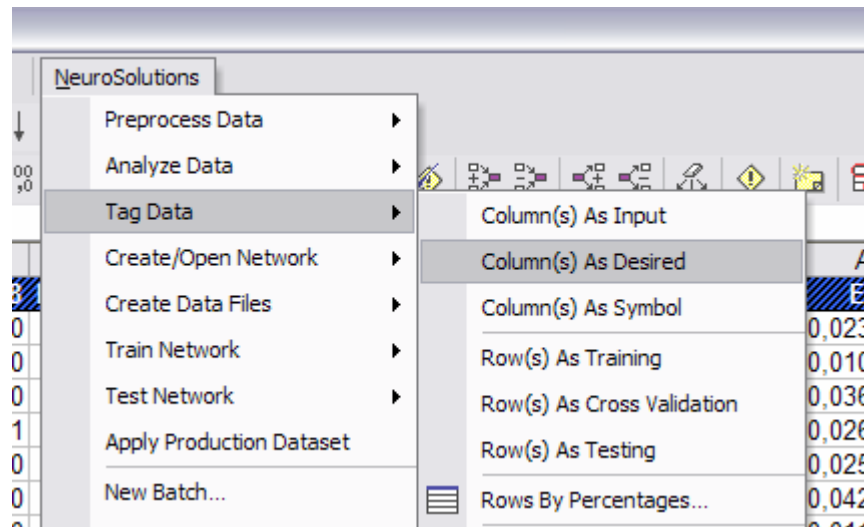


Figura 28: Menu de acesso às funcionalidades de inicialização da rede no NeuroSolutions

Depois de informadas as colunas de entrada e de saída, assim como as linhas de treinamento, validação e teste, pode-se, então, construir a rede utilizando o menu *NeuroSolutions, Create / Open Network, New*, conforme ilustra a Figura 29.

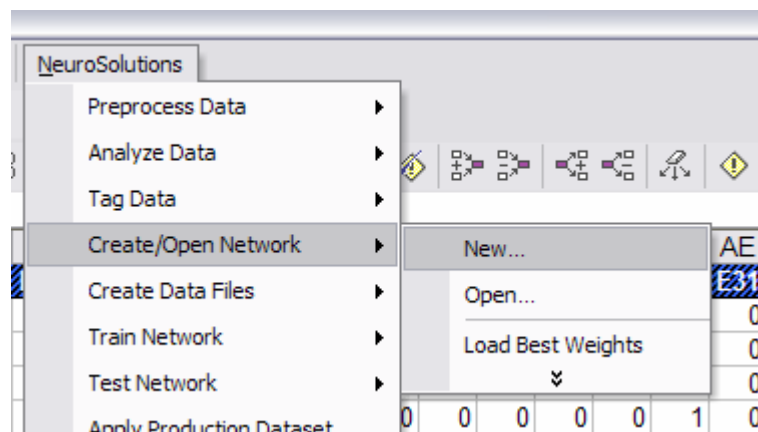


Figura 29: Menu de acesso às funcionalidades de construção da rede no NeuroSolutions

A funcionalidade *New* apresenta a tela para configuração da rede e é acessada através do menu *NeuroSolutions*, conforme mostra a Figura 30.

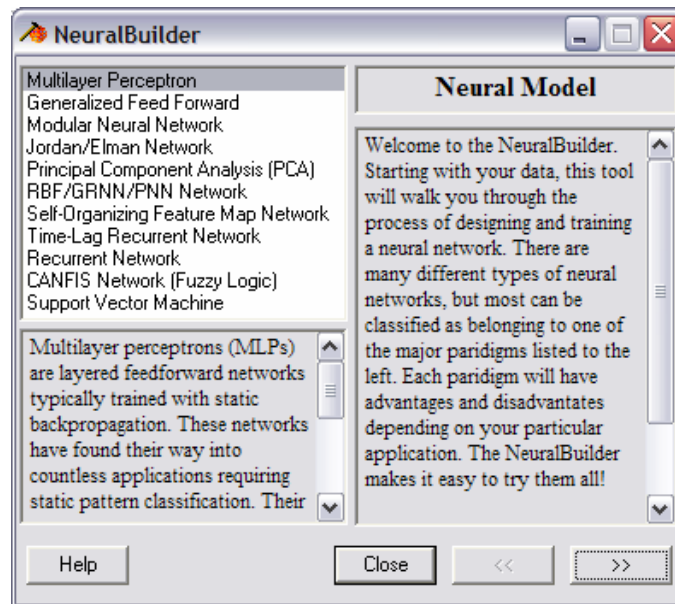


Figura 30: Construção da RNA com o NeuroSolutions

A rede selecionada, dentre as opções mostradas na Figura 30, foi a MLP. O próximo passo consiste em informar a quantidade de camadas escondidas (Figura 31).

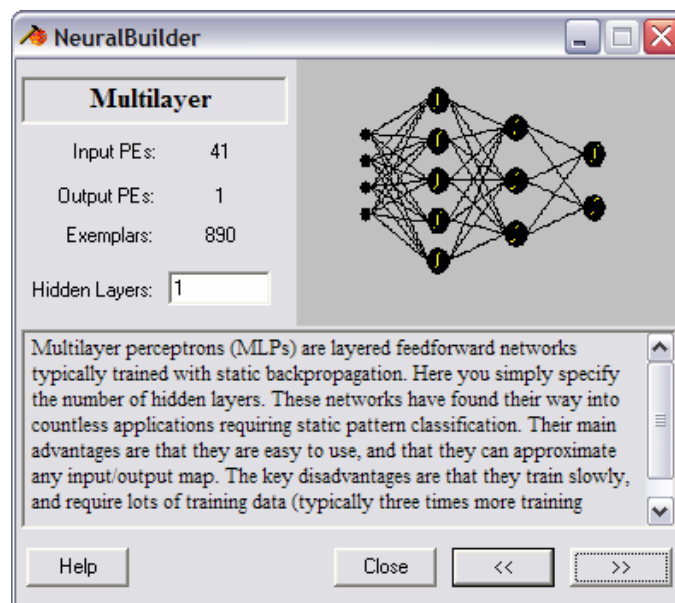


Figura 31: Seleção da quantidade de camadas escondidas com o NeuroSolutions

No passo subsequente, serão informadas as quantidades de neurônios que irão formar cada uma das camadas escondidas, assim como a função de ativação, a medida de aprendizagem, o tamanho do passo e a taxa de momentum, conforme apresenta a Figura 32.

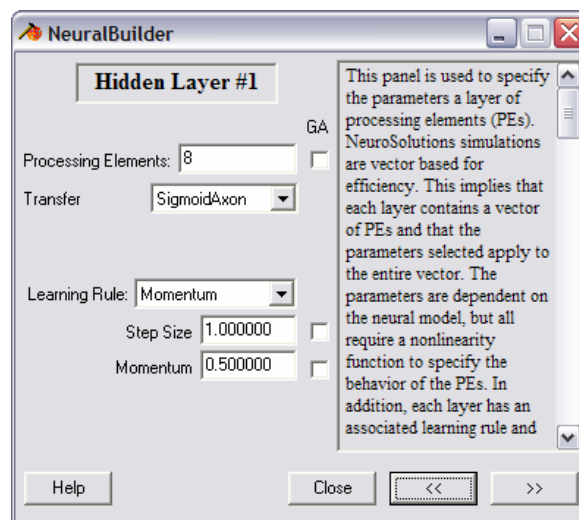


Figura 32: Configuração das camadas escondidas com o NeuroSolutions

Em seguida, deve-se configurar a camada de saída informando a função de transferência, a medida de aprendizagem, o tamanho do passo e a taxa de momentum, conforme a Figura 33.

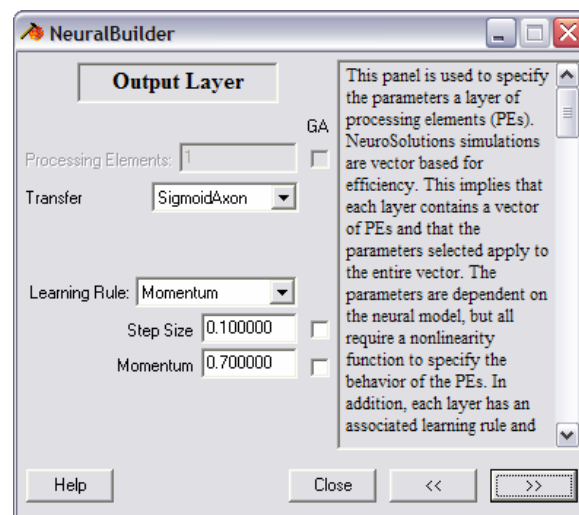


Figura 33: Configuração das camadas de saída com o NeuroSolutions

O último passo é configurar o Controle de Aprendizado Supervisionado: informar o número de épocas para o treinamento e as configurações de prova (Figuras 34 e 35).

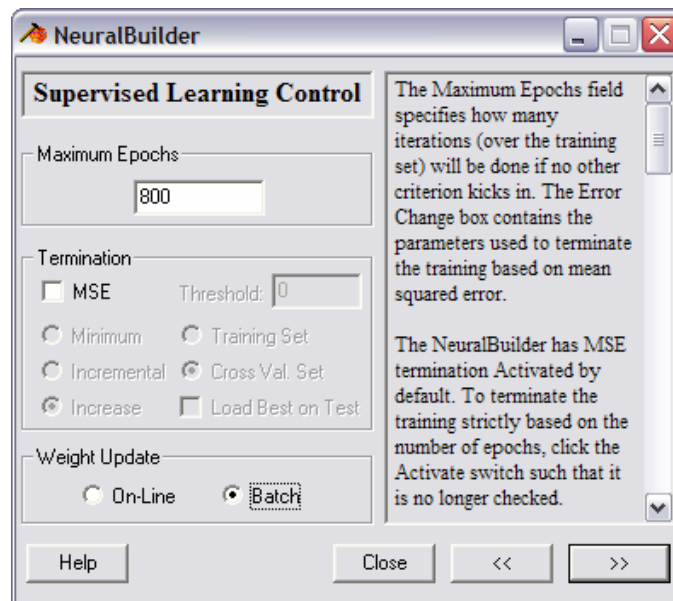


Figura 34: Configuração do Controle de Aprendizado Supervisionado no NeuroSolutions

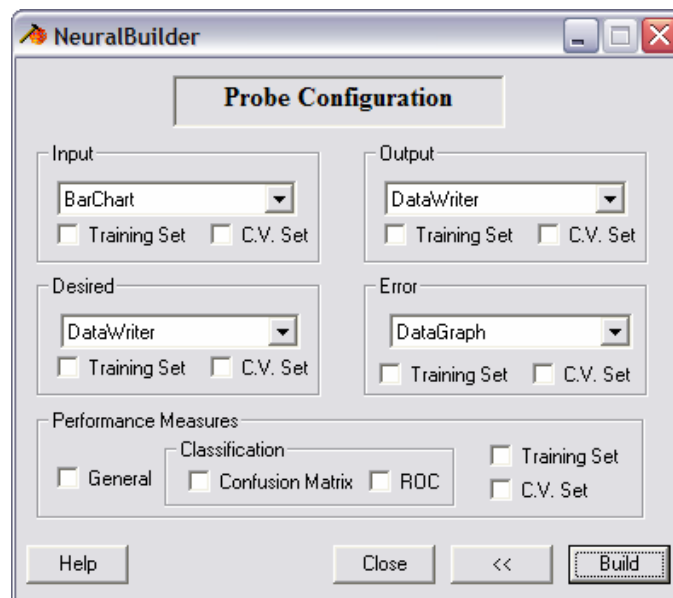


Figura 35: Configuração das configurações de prova no NeuroSolutions

O botão *Build* irá construir a RNA que terá seu esquema apresentado na área do *NeuroSolutions*.

5.2 Treinamento

Para iniciar o treinamento, deve-se utilizar a funcionalidade *Train*, localizada no menu *NeuroSolutions, Train Network*, como ilustra a Figura 35.

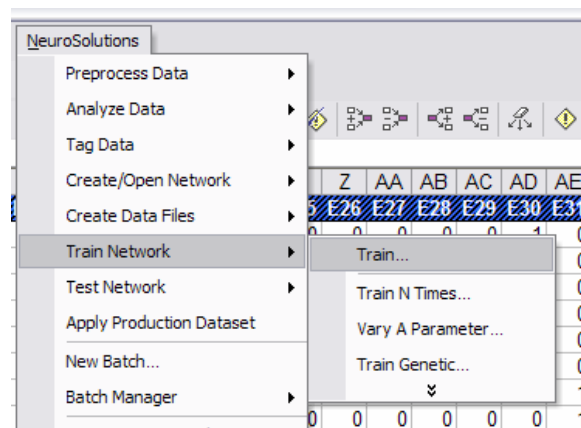


Figura 36: Funcionalidade para treinamento da rede no NeuroSolutions

A funcionalidade *Train* apresenta alguns ajustes nas configurações, como: nome do treinamento e opções do treinamento (Figura 37).

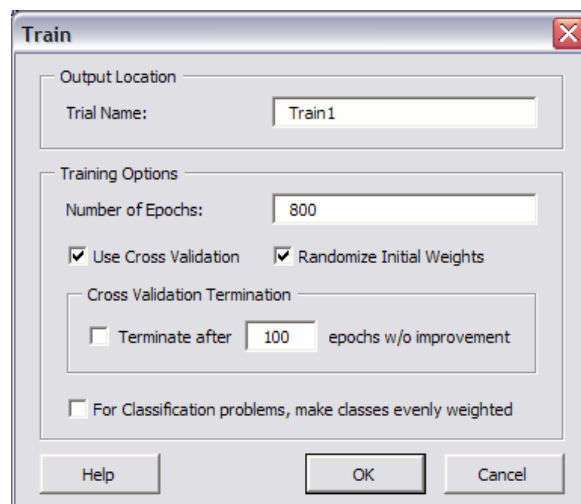


Figura 37: Configurações do treinamento no NeuroSolutions

O treinamento pode ser acompanhado ou cancelado através da tela de progresso do treinamento (Figura 38).

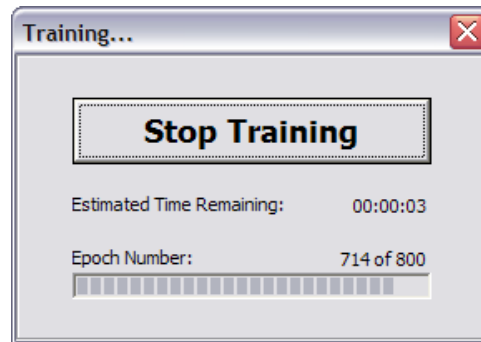
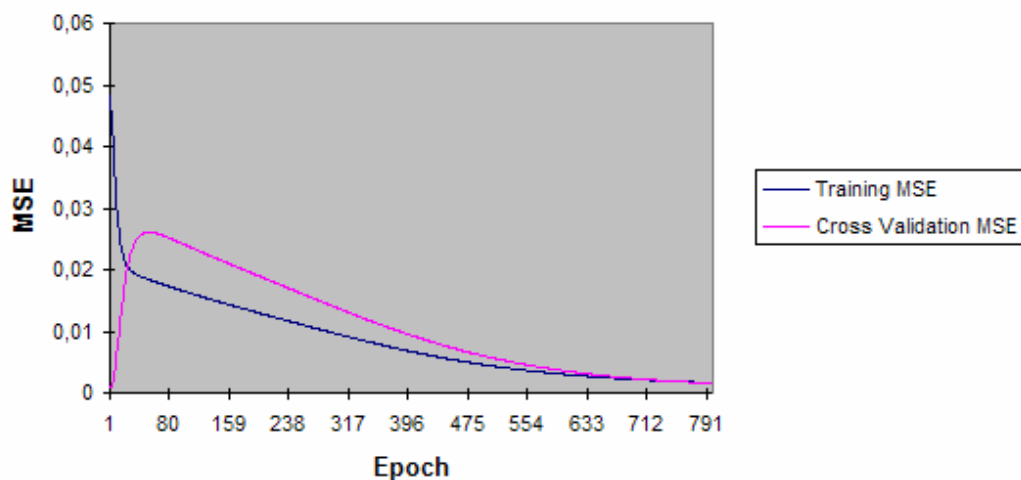


Figura 38: Progresso do treinamento no NeuroSolutions

Os resultados do treinamento e da validação são informados através de gráficos e planilhas, conforme ilustram as Figuras 39 e 40.

MSE versus Epoch



<i>Best Networks</i>	<i>Training</i>	<i>Cross Validation</i>
Epoch #	800	1
Minimum MSE	0,00163917	0,000834134
Final MSE	0,00163917	0,001549135

Figura 39: Valores dos indicadores do treinamento no NeuroSolutions

	Training MSE	Cross Validation MSE
1		
2	0,04814482	0,00083413
3	0,04700325	0,00087452
4	0,04518024	0,00101967
5	0,04299355	0,0013071
6	0,04065916	0,00175486
7	0,03832192	0,00236466
8	0,03607575	0,00312588
9	0,03397837	0,00401959
10	0,03206149	0,00502198
11	0,03033838	0,00610713
12	0,02880962	0,00724918

Figura 40: Amostra de valores de indicadores do treinamento e validação no NeuroSolutions

5.3 Testes

Os testes são executados através da funcionalidade *Test*, localizada no menu *NeuroSolutions*, *Test Network*, conforme ilustra a Figura 41.

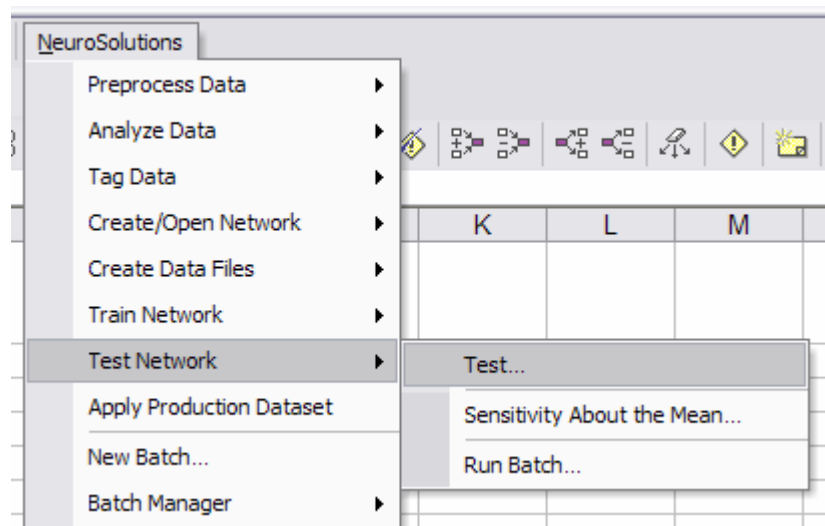


Figura 41: Funcionalidade para teste da rede no NeuroSolutions

A funcionalidade *Test* requer alguns ajustes nas configurações, conforme ilustra a Figura 42.

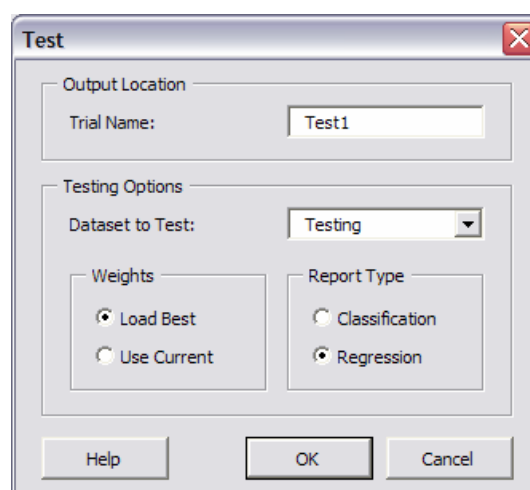


Figura 42: Configurações do teste no NeuroSolutions

Após a execução do teste, são apresentados os resultados e indicadores através de gráficos e planilhas, conforme ilustram as Figuras 43 e 44.

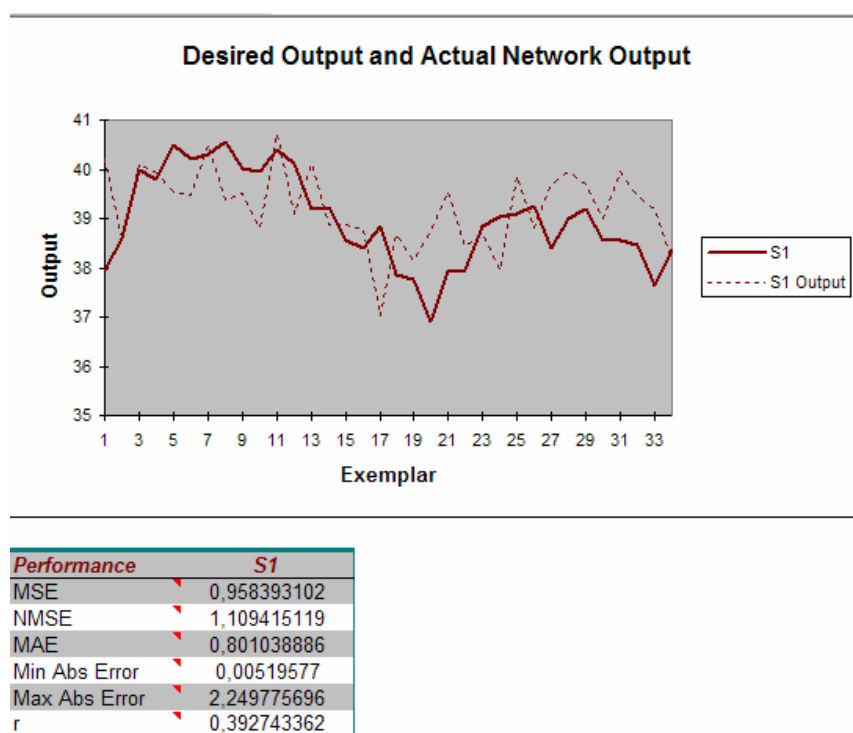


Figura 43: Comparativo e Indicadores do teste no NeuroSolutions

S1	S1 Output
37,95	40,1997757
38,59	38,5951958
40	40,1122322
39,8	39,924263
40,49	39,5287018
40,2	39,4844475
40,3	40,4913521
40,55	39,3747635
40,03	39,5106621
39,95	38,7977066
40,4	40,7036247
40,11	39,0924568
39,2	40,124321

Figura 44: Amostra de valores após o teste no NeuroSolutions

5.4 Resultados Obtidos

Nesta seção, são apresentados os resultados obtidos com as ferramentas Previsão de Ativos (Capítulo 4) e *NeuroSolutions* para Excel 4.2, quando aplicada a base de dados *Trade*.

5.4.1 Resultados Obtidos com a Ferramenta Previsão de Ativos

Foram utilizados os três métodos distintos de treinamento, com o objetivo de encontrar valores para os parâmetros que, quando combinados, realizassem a melhor previsão. Neste contexto, foram realizados vários experimentos preliminares, com o objetivo de se alcançar uma melhor configuração de rede. Os parâmetros que tiveram seus valores alterados durante os testes são: o número de neurônios da camada oculta, a taxa de aprendizagem, a taxa de momentum e o número de ciclos.

5.4.1.1 Método Trein 1

O método *Trein 1* obteve seus melhores resultados com os seguintes parâmetros configurados: 8 neurônios na camada oculta, função de ativação

sigmóide, 600 ciclos, taxa de aprendizagem igual a 0,4 e taxa de momentum igual a 0,4. Foram efetuados 10 treinamentos com a rede configurada com estes parâmetros e seus resultados estão relacionados na Tabela 5. Foram utilizadas as medidas de erro MSE e MAPE para avaliar os resultados das previsões obtidas nos testes.

Treinamentos	MSE	MAPE (%)
1	0,675028	1,61
2	0,714685	1,70
3	0,742759	1,73
4	0,769356	1,73
5	0,716871	1,70
6	0,785412	1,80
7	0,765679	1,76
8	0,741847	1,73
9	0,811312	1,83
10	0,703138	1,67
Média	0,742609	1,73

Tabela 5: Experimentos realizados com a MLP no método *Trein 1*

Entre os dez treinamentos executados com o método *Trein 1* e a configuração informada, o treinamento 1 apresentou o melhor resultado com MSE de 0,675028 e MAPE de 1,61%. O pior resultado foi alcançado no treinamento 9 com MSE igual a 0,811312 e MAPE de 1,83%. Os valores médios obtidos nos dez treinamentos foram: MSE igual a 0,742609 e MAPE de 1,73%, conforme a Tabela 5.

O Gráfico 4 apresenta uma comparação entre os valores reais e os valores previstos com o melhor treinamento. O Gráfico 5 ilustra a comparação entre valores previstos com o melhor e o pior treinamento.

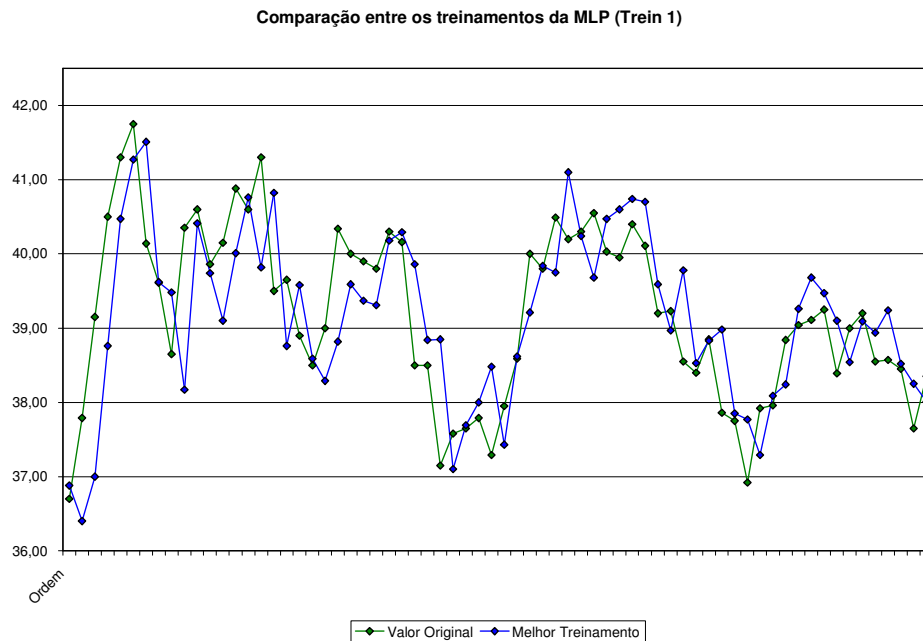


Gráfico 4: Comparação entre os valores originais e os previstos no melhor treinamento (Trein 1).

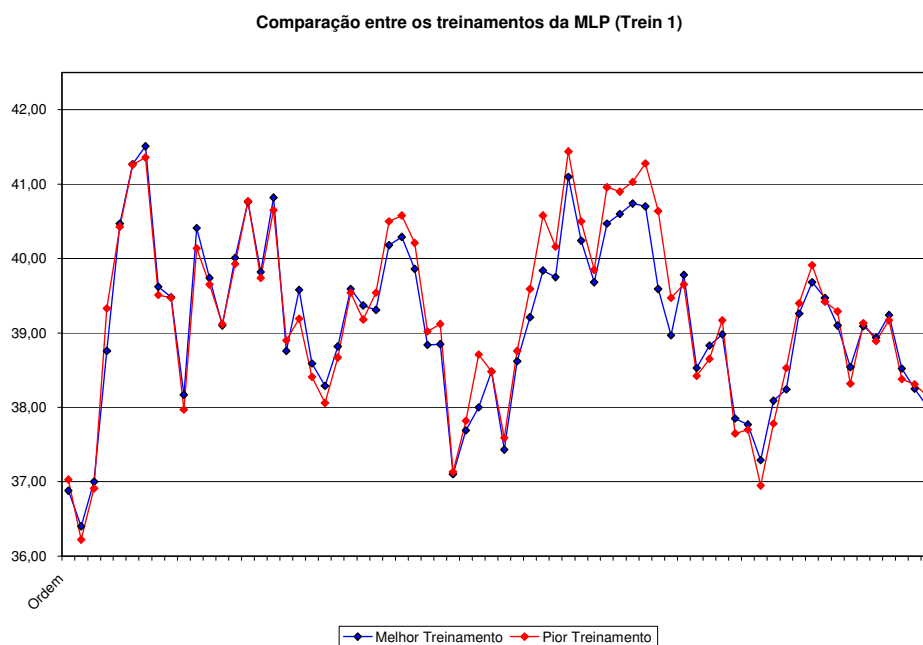


Gráfico 5: Comparação entre os valores previstos no melhor e pior treinamento (Trein 1).

5.4.1.2 Método Trein 2

O método *Trein 2* obteve seus melhores resultados com os seguintes parâmetros configurados: 8 neurônios na camada oculta, função de ativação sigmóide, 600 ciclos, taxa de aprendizagem igual a 0,4 e taxa de momentum igual a 0,4. Foram efetuados 10 treinamentos com a rede configurada com estes parâmetros e seus resultados estão relacionados na Tabela 6. Foram utilizadas as medidas de erro MSE e MAPE para avaliar os resultados das previsões obtidas nos testes.

Treinamentos	MSE	MAPE (%)
1	0,921528	2,00
2	0,901399	1,98
3	0,853446	1,93
4	0,875541	1,95
5	0,824026	1,90
6	0,869706	1,94
7	0,852887	1,92
8	0,860281	1,94
9	0,927116	2,01
10	0,737679	1,78
Média	0,862361	1,94

Tabela 6: Experimentos realizados com a MLP no método *Trein 2*

Entre os dez treinamentos executados com o método *Trein 2* e a configuração informada, o treinamento 10 apresentou o melhor resultado com MSE de 0,737679 e MAPE de 1,78%. O pior resultado foi alcançado no treinamento 9 com MSE igual a 0,927116 e MAPE de 2,01%. Os valores médios obtidos nos dez treinamentos foram: MSE igual a 0,862361 e MAPE de 1,94%, conforme a Tabela 6.

O Gráfico 6 apresenta uma comparação entre os valores reais e os valores previstos com o melhor treinamento. O Gráfico 7 ilustra a comparação entre valores previstos com o melhor e o pior treinamento.

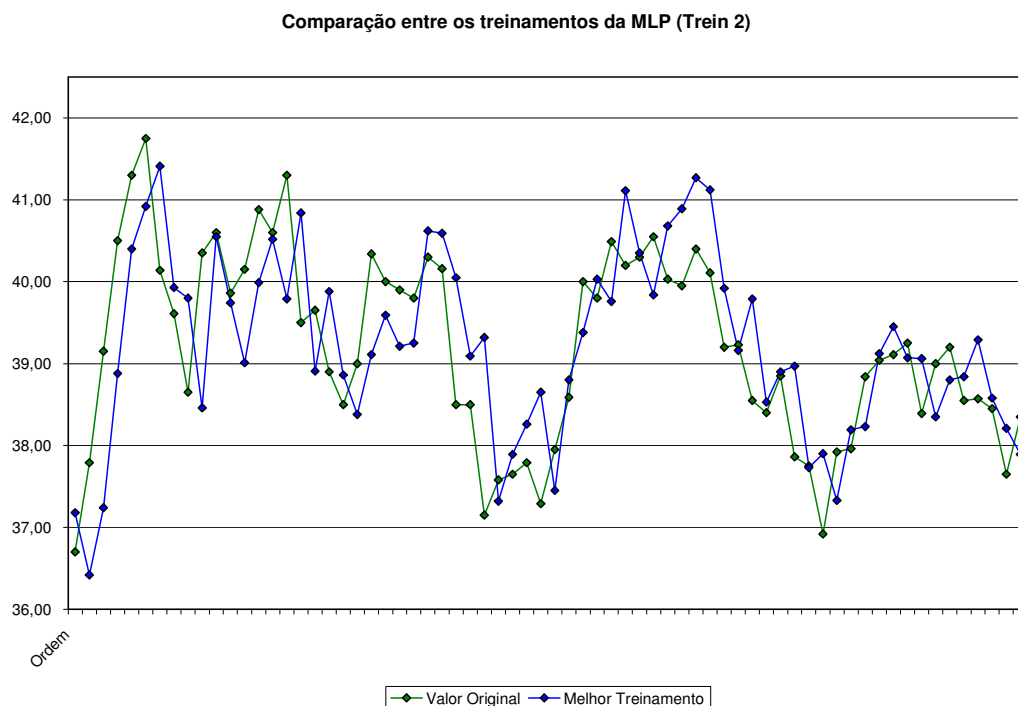


Gráfico 6: Comparação entre os valores originais e os previstos no melhor treinamento (Trein 2).

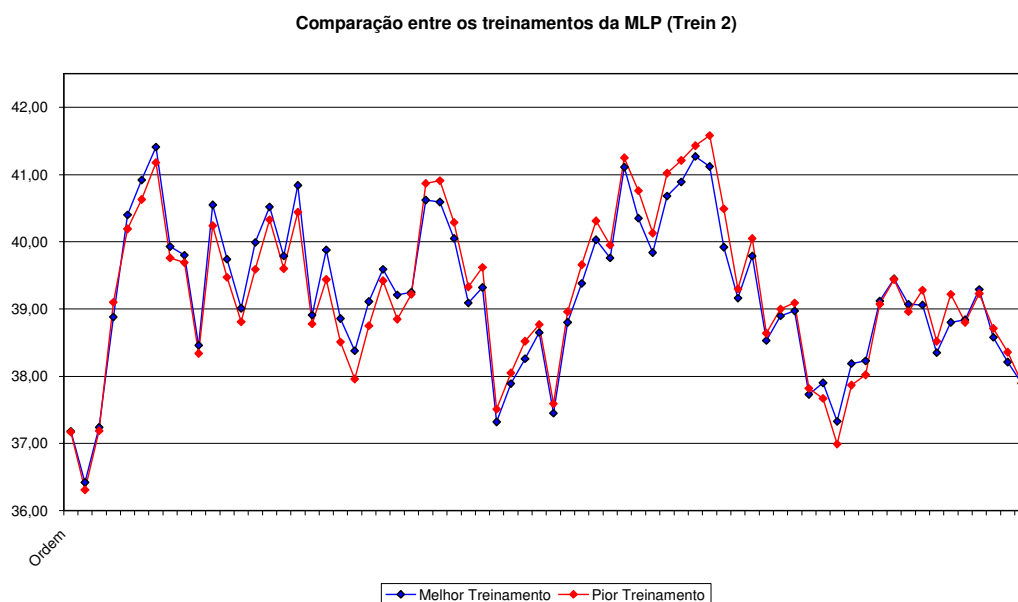


Gráfico 7: Comparação entre os valores previstos no melhor e pior treinamento (Trein 2).

5.4.1.3 Método Trein 3

O método *Trein 3* obteve seus melhores resultados com os seguintes parâmetros configurados: 21 neurônios na camada de entrada, 7 neurônios na camada oculta, função de ativação sigmóide, 300 ciclos, taxa de aprendizagem igual a 0,4 e taxa de momentum igual a 0,4. Foram efetuados 10 treinamentos com a rede configurada com estes parâmetros e seus resultados estão relacionados na Tabela 7. Foram utilizadas as medidas de erro MSE e MAPE para avaliar os resultados das previsões obtidas nos testes.

Treinamentos	MSE	MAPE (%)
1	0,727338	1,75
2	0,697859	1,74
3	0,715097	1,73
4	0,716644	1,75
5	0,717463	1,74
6	0,704666	1,74
7	0,716515	1,74
8	0,709787	1,73
9	0,698484	1,74
10	0,711603	1,74
Média	0,711546	1,74

Tabela 7: Experimentos realizados com a MLP no método *Trein 3*

Entre os dez treinamentos executados com o método *Trein 3* e a configuração informada, o treinamento 2 apresentou o melhor resultado com MSE de 0,697859 e MAPE de 1,74%. O pior resultado foi alcançado no treinamento 1 com MSE igual a 0,727338 e MAPE de 1,75%. Os valores médios obtidos nos dez treinamentos foram: MSE igual a 0,711546 e MAPE de 1,74%, conforme a Tabela 7.

O Gráfico 8 apresenta uma comparação entre os valores reais e os valores previstos com o melhor treinamento. O Gráfico 9 ilustra a comparação entre valores previstos com o melhor e o pior treinamento.

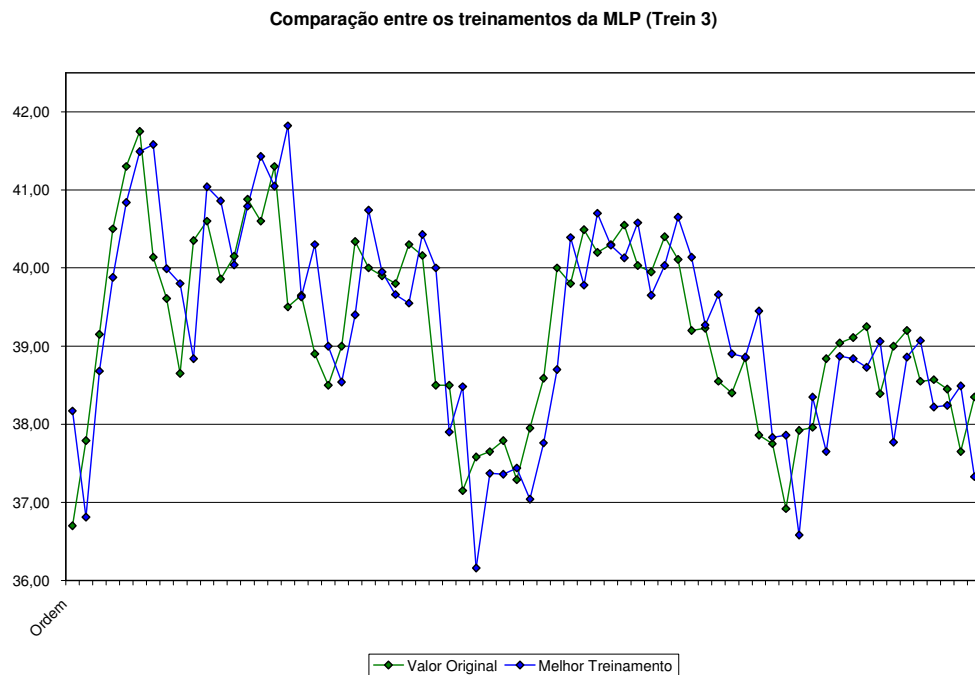


Gráfico 8: Comparação entre os valores originais e os previstos no melhor treinamento (Trein 3).

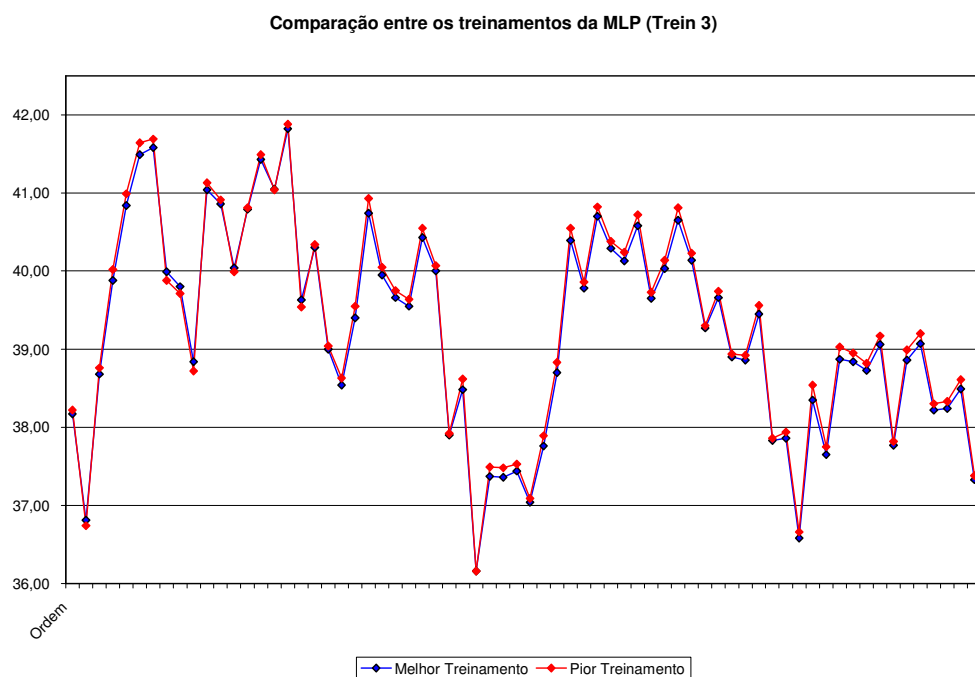


Gráfico 9: Comparação entre os valores previstos no melhor e no pior treinamento (Trein 3).

5.4.2 Resultados Obtidos com a Ferramenta NeuroSolutions

Foram utilizados os mesmos três métodos de treinamento, conforme a ferramenta Previsão de Ativos, com o objetivo de comparar os resultados obtidos entre ambas.

5.4.2.1 Método Trein 1

O método *Trein 1* obteve seus resultados com os seguintes parâmetros configurados: 8 neurônios na camada oculta, função de ativação sigmóide, 600 ciclos, tamanho do passo (*step size*) igual a 0,4 e taxa de momentum igual a 0,4. Foram efetuados 10 treinamentos com a rede configurada com estes parâmetros e seus resultados estão relacionados na Tabela 8. Foram utilizadas as medidas de erro MSE e MAPE para avaliar os resultados das previsões obtidas nos testes.

Treinamentos	MSE	MAPE (%)
1	3,142558	4,08
2	3,389500	4,27
3	2,773158	3,81
4	4,036682	4,79
5	2,995038	3,95
6	3,489148	4,37
7	2,488480	3,51
8	3,611731	4,46
9	3,352500	4,25
10	2,908371	3,89
Média	3,218717	4,14

Tabela 8: Experimentos realizados no NeuroSolutions com a MLP no método *Trein 1*

Entre os dez treinamentos executados com o método *Trein 1* e a configuração informada, o treinamento 7 apresentou o melhor resultado com MSE de 2,488480 e MAPE de 3,51%. O pior resultado foi alcançado no treinamento 4 com MSE igual a 4,036682 e MAPE de 4,79%. Os valores médios obtidos nos dez treinamentos foram: MSE igual a 3,218717 e MAPE de 4,14% (Tabela 8).

O Gráfico 10 apresenta uma comparação entre os valores reais e os valores previstos com o melhor treinamento. O Gráfico 11 ilustra a comparação entre valores previstos com o melhor e o pior treinamento.

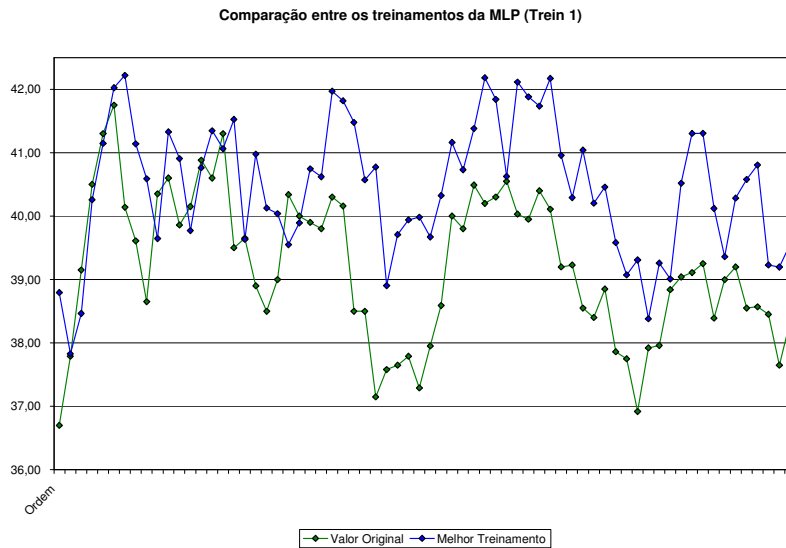


Gráfico 10: Comparação entre os valores originais e os previstos no melhor treinamento do NeuroSolutions (*Trein 1*).

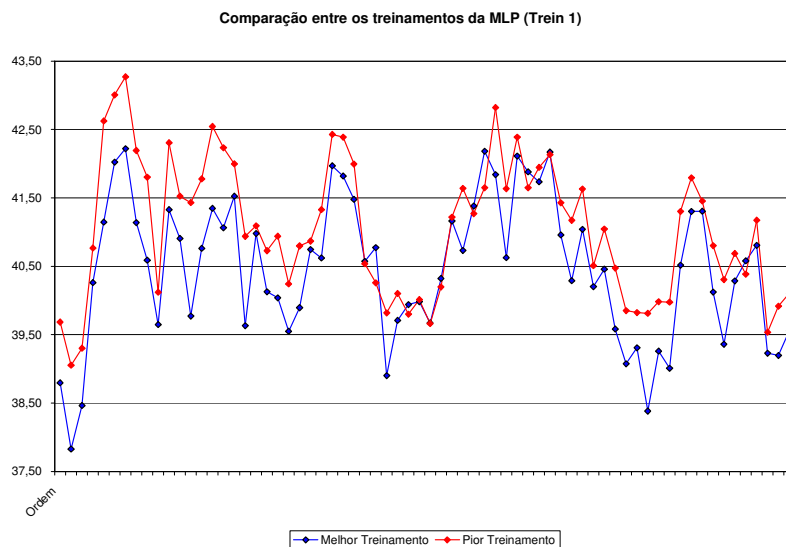


Gráfico 11: Comparação entre os valores previstos no melhor e pior treinamento do NeuroSolutions (*Trein 1*).

5.4.2.2 Método Trein 2

O método *Trein 2* obteve seus resultados com os seguintes parâmetros configurados: 8 neurônios na camada oculta, função de ativação sigmóide, 600 ciclos, tamanho do passo (*step size*) igual a 0,4 e taxa de momentum igual a 0,4. Foram efetuados 10 treinamentos com a rede configurada com estes parâmetros e seus resultados estão relacionados na Tabela 9. Foram utilizadas as medidas de erro MSE e MAPE para avaliar os resultados das previsões obtidas nos testes.

Treinamentos	MSE	MAPE (%)
1	3,714387	4,36
2	2,851998	3,76
3	4,162746	4,62
4	2,875297	3,88
5	3,861102	4,63
6	2,841289	3,84
7	3,926506	4,66
8	4,102054	4,73
9	2,989519	3,91
10	6,547921	6,12
Média	3,787282	4,45

Tabela 9: Experimentos realizados no NeuroSolutions com a MLP no método *Trein 2*

Entre os dez treinamentos executados com o método *Trein 2* e a configuração informada, o treinamento 6 apresentou o melhor resultado com MSE de 2,841289 e MAPE de 3,84%. O pior resultado foi alcançado no treinamento 4 com MSE igual a 6,547921 e MAPE de 6,12%. Os valores médios obtidos nos dez treinamentos foram: MSE igual a 3,787282 e MAPE de 4,45%, conforme a Tabela 9.

O Gráfico 12 apresenta uma comparação entre os valores reais e os valores previstos com o melhor treinamento. O Gráfico 13 ilustra a comparação entre valores previstos com o melhor e o pior treinamento.

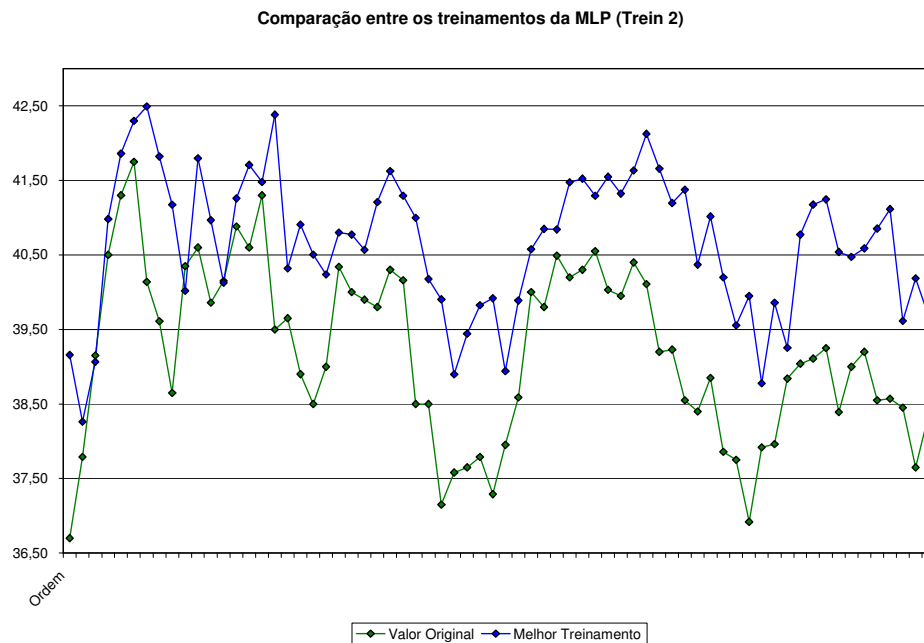


Gráfico 12: Comparação entre os valores originais e os previstos no melhor treinamento do NeuroSolutions (*Trein 2*).

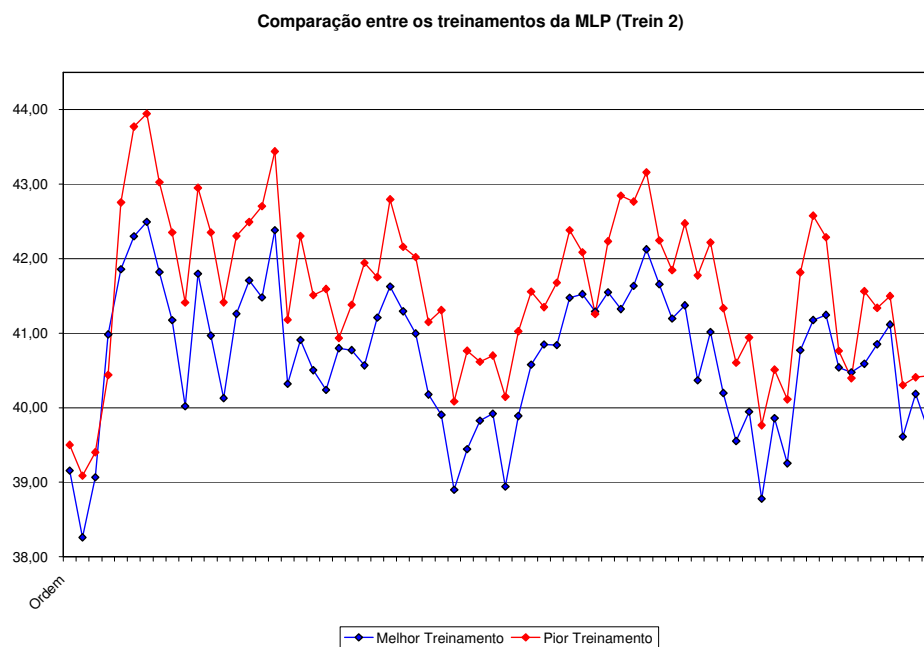


Gráfico 13: Comparação entre os valores previstos no melhor e pior treinamento do NeuroSolutions (*Trein 2*).

5.4.2.3 Método *Trein 3*

O método *Trein 3* obteve seus resultados com os seguintes parâmetros configurados: 21 neurônios na camada de entrada, 7 neurônios na camada oculta, função de ativação sigmóide, 300 ciclos, tamanho do passo (*step size*) igual a 0,4 e taxa de momentum igual a 0,4. Foram efetuados 10 treinamentos com a rede configurada com estes parâmetros e seus resultados estão relacionados na Tabela 10. Foram utilizadas as medidas de erro MSE e MAPE para avaliar os resultados das previsões obtidas nos testes.

Treinamentos	MSE	MAPE (%)
1	2,628571	3,63
2	2,820001	3,82
3	2,882379	3,87
4	2,508422	3,50
5	2,789374	3,70
6	3,068602	3,93
7	2,961740	3,89
8	2,677445	3,67
9	3,146266	4,03
10	2,927536	3,85
Média	2,841034	3,79

Tabela 10: Experimentos realizados no NeuroSolutions com a MLP no método *Trein 3*

Entre os dez treinamentos executados com o método *Trein 3* e a configuração informada, o treinamento 4 apresentou o melhor resultado com MSE de 2,508422 e MAPE de 3,50%. O pior resultado foi alcançado no treinamento 9 com MSE igual a 3,146266 e MAPE de 4,03%. Os valores médios obtidos nos dez treinamentos foram: MSE igual a 2,841034 e MAPE de 3,79%, conforme a Tabela 10.

O Gráfico 14 apresenta uma comparação entre os valores reais e os valores previstos com o melhor treinamento. O Gráfico 15 ilustra a comparação entre valores previstos com o melhor e o pior treinamento.

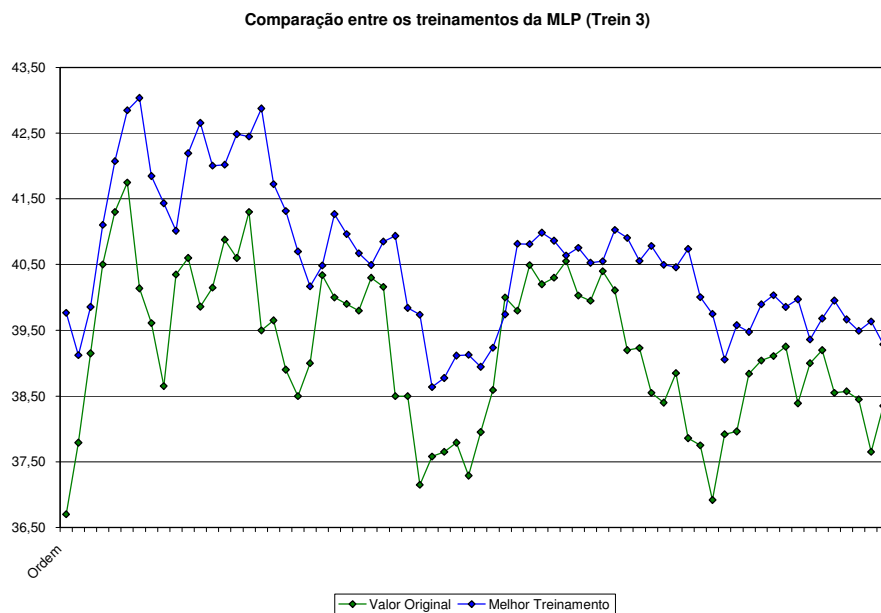


Gráfico 14: Comparação entre os valores originais e os previstos no melhor treinamento do NeuroSolutions (*Trein 3*).

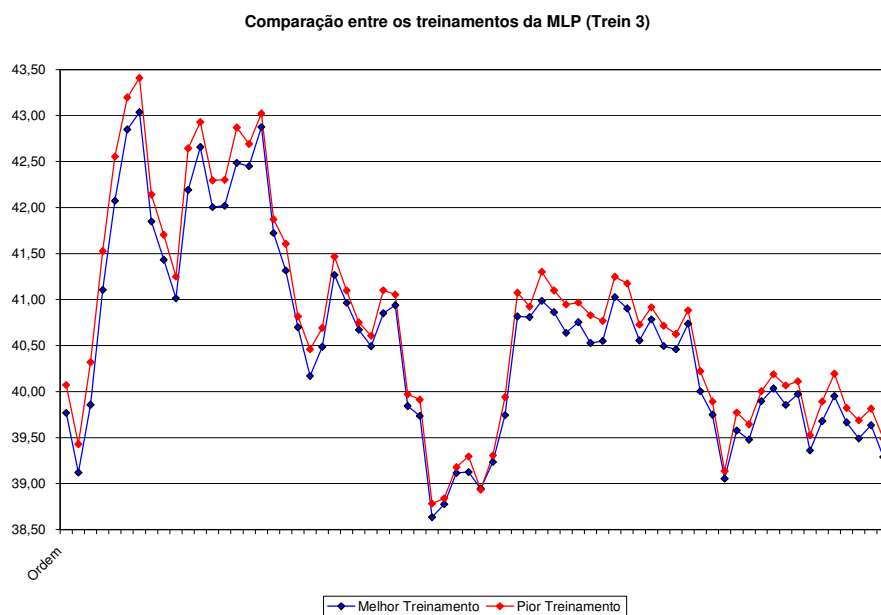


Gráfico 15: Comparação entre os valores previstos no melhor e pior treinamento do NeuroSolutions (*Trein 3*).

5.4.3 Comparação entre os Resultados

5.4.3.1 **Resumo dos Resultados com a Ferramenta Previsão de Ativos**

Dentre os métodos de treinamento *Trein 1*, *Trein 2* e *Trein 3*, o primeiro deles apresentou os melhores resultados, conseguindo valores de MSE e MAPE, respectivamente, iguais a 0,675028 e 1,61% com totais médios obtidos nos dez treinamentos de 0,742609 e 1,73%. Os piores resultados foram obtidos com o método *Trein 2* que conseguiu obter os seguintes valores: MSE igual a 0,737679 e MAPE de 1,78% e totais médios de 0,862361 e 1,94%. O método *Trein 3* teve um posicionamento intermediário obtendo valores de MSE e MAPE iguais a 0,697859 e 1,74% respectivamente, porém, na média obteve resultados melhores que o método *Trein 1* com MSE de 0,711516 e MAPE de 1,74%.

5.4.3.2 **Resumo dos Resultados com a Ferramenta NeuroSolutions**

Apresentou seus melhores resultados através do método *Trein 1*, com os valores de MSE e MAPE, respectivamente, iguais a 2,488480 e 3,51% e média de 3,218717 e 4,14%. Já o método *Trein 3* não apresentou os melhores resultados individuais com MSE e MAPE, respectivamente, iguais a 2,508422 e 3,50%, porém os totais médios foram melhores que os obtidos pelo método *Trein 1*, com MSE e MAPE iguais a 2,841034 e 3,79%. O método que apresentou os piores resultados foi o *Trein 2* com MSE e MAPE, respectivamente, iguais a 2,841289 e 3,84%, valores médios iguais a 3,787282 e 4,45%.

5.4.3.3 **Os Melhores Resultados**

Foram obtidos com a ferramenta Previsão de Ativos, utilizando o método *Trein 1*, que apresentou MSE de 0,675028 e MAPE de 1,61%, enquanto o *NeuroSolutions* atingiu seus melhores índices com valores de MSE e MAPE respectivamente iguais a 2,488480 e 3,51%, também com o método *Trein 1*.

Capítulo 6 - Conclusão

Neste capítulo, serão ressaltados os resultados alcançados, a relevância e as contribuições deste trabalho de pesquisa e ainda serão indicadas as atividades futuras que poderão ser realizadas como continuação deste trabalho.

6.1 Conclusões do Trabalho

Através da investigação experimental realizada pode-se concluir que:

- A rede MLP pode reproduzir a série temporal da TNLP4 com precisão significativa. Os resultados obtidos com a ferramenta desenvolvida neste trabalho demonstram que a rede foi capaz de captar as tendências dos preços, conseguindo, na maioria das vezes, aproximar a estimativa do valor real. Isto confirma a idéia de que as redes neurais podem facilitar o trabalho dos investidores de ações, proporcionando grandes possibilidades de obtenção de ganhos financeiros.
- Para investigar o comportamento das redes neurais em aplicações práticas, faz-se necessária uma intensa investigação experimental. Muitos experimentos preliminares são realizados para se obter a arquitetura que apresenta os melhores resultados. Esse é um trabalho intenso de busca que deve ser feito com diversas combinações possíveis entre os parâmetros de configuração da rede. A inclusão de unidades ocultas em número superior ao necessário pode levar à "memorização" da seqüência de preços empregados no treinamento e, portanto, prejudicar a realização de previsões dos preços futuros. Por outro lado, uma rede neural artificial com estrutura mais simples que a necessária pode não alcançar aproximação satisfatória mesmo para as cotações apresentadas durante o processo de ajuste dos

pesos.

- Embora ambas as ferramentas tenham apresentados índices baixos de erro, conseguiu-se melhores resultados com a ferramenta Previsão de Ativos.

6.2 Contribuições do Trabalho

A grande contribuição deste trabalho foi a demonstração da aplicabilidade das Redes Neurais na previsão de ações da BOVESPA. Os resultados dos experimentos confirmaram a capacidade dos modelos neurais em manipular informações decorrentes de ambientes extremamente dinâmicos, como a série da TNLP4. Portanto os resultados alcançados comprovam a relevância desta técnica como um apoio para a tomada de decisão em investimentos no Mercado Financeiro.

Uma outra contribuição é a disponibilização da base de dados *Trade* de forma gratuita para que outros pesquisadores possam utilizá-la em seus estudos.

Finalmente, outra contribuições deste trabalho é a disponibilização do componente *TMVBMultiLayerPerceptron* de forma gratuita, para que outros pesquisadores possam utilizá-lo em seus trabalhos, incrementá-lo com novas implementações e conhecer, na prática, o funcionamento de seus algoritmos.

6.3 Trabalhos Futuros

São muitas as possibilidades de trabalhos futuros a serem realizadas como continuidade deste trabalho de pesquisa. Alguns deles são relacionados a seguir:

- Implementação de uma ferramenta que colete informações do mercado financeiro, com modo de operação *on-line*, para que o treinamento possa ser contínuo e traga a possibilidade de resposta imediata da rede e o acompanhamento otimizado da dinâmica do

mercado.

- Realizar previsões para períodos superiores a um único dia.
- Implementar e analisar outros modelos de redes neurais como GRNN, RBF, NARX (*Nonlinear Autoregressive with Exogenous Input*), NARMAX (*Nonlinear Autoregressive Moving Avarage with Exogenous Input*), etc.
- Implementar e analisar técnicas de Computação Evolucionária como algoritmos genéticos no aprendizado das redes ou na otimização do número de neurônios da camada oculta.
- Analisar a viabilidade de sistemas híbridos com Inteligência Artificial Simbólica e Lógica *Fuzzy* em aplicações de previsão.

Referências

- [Azevedo, 2000] AZEVEDO, F. M.; BRASIL, L. M.; OLIVEIRA, R. L. *Redes Neurais com Aplicações em Controle e em Sistemas Especialistas*. Visual Books, 2000.
- [Bosaipo, 2001] BOSAIPO, C. R. *Aplicação das Redes Neurais na Previsão do Comportamento de Mercados Financeiros*. Artigo – COPPE da UFRJ, 2001.
- [Bressan, 2002] BRESSAN, A. A.; LIMA, J. E. *Modelos de Previsão de Preços Aplicados aos Contratos Futuros de Boi Gordo na BM&F*. Artigo – FACE / Universidade Federal de Minas Gerais, Departamento de Ciências Administrativas; Universidade Federal Viçosa, Departamento de Economia Rural, 2002.
- [Castro, 2003] CASTRO JR., F. F. *Previsão de Insolvência de Empresas Brasileiras Usando Análise Discriminate, Regressão Logística e Redes Neurais*. Dissertação de Mestrado – Universidade de São Paulo, Faculdade de Economia, Administração e Contabilidade, 2003.
- [Cottrell, 1985] COTTRELL, G.W. *A Connectionist Approach to Word Sense Disambiguation*. Tese de Doutorado – Rochester: University of Rochester, 1985.
- [Dazzi, 1999] DAZZI, R. S. *Sistemas Especialistas Conexionistas: Implementação por Redes Diretas e Bidirecionais*. Dissertação de Mestrado – UFSC, 1999.
- [Fahlman, 1990] FAHLMAN, S. e LEBIERE, C. *The Cascade-Correlation Learning Architecture, Advances in Neural Information Processing Systems II*. Morgan Kauffman, 1990.

- [Fernandes, 1995] FERNANDES, L. L. *Utilização de Redes Neurais na Análise e Previsão de Séries Temporais*. Dissertação de Mestrado – CGPCC da UFRGS, 1995.
- [Haykin, 2001] HAYKIN, S. *Redes Neurais: Princípios e Práticas*, Bookman, 2 ed., 2001.
- [Khalsa, 1997] KHALSA, D. S.; STAUTH, C. *Longevidade do Cérebro*. Objetiva, 7 ed., 1997.
- [Lawrence, 1997] LAWRENCE, R. *Using Neural Networks to Forecast Stock Market Prices*. Artigo – University of Manitoba, Department of Computer Science, 1997.
- [Martineli, 1999] MARTINELI, E. *Extração de Conhecimento de Redes Neurais Artificiais*. Dissertação de Mestrado – USP São Carlos, Instituto de Ciências Matemáticas e de Computação, 1999.
- [Medeiros, 2003] MEDEIROS, L. F. *Redes Neurais em Delphi*, Visual Books, 2003.
- [Mello, 2004] MELLO, M.T. *Aplicação de Redes Neurais Artificiais no Processo de Precificação de Ações*. Trabalho de Conclusão de Curso – Universidade Federal de Pelotas, Instituto de Física e Matemática, 2004.
- [Modro, 2000] MODRO, N. R. *Sistema Inteligente de Monitoramento e Gerenciamento Financeiro para Micro e Pequenas Empresas*. Dissertação de Mestrado – Universidade Federal de Santa Catarina, Engenharia de Produção, 2000.
- [Mueller, 1996] MUELLER, A. *Uma Aplicação de Redes Neurais Artificiais na Previsão do Mercado Acionário*. Dissertação de Mestrado – UFSC, 1996.

- [Oliveira, 2001] OLIVEIRA, E. *Previsão da Série Temporal do Índice da Bolsa de Valores de São Paulo Usando Redes Neurais e Estatística*. Dissertação de Mestrado – Universidade Federal de Pernambuco, Centro de Informática, 2001.
- [Passari, 2003] PASSARI, A. F. *Exploração de Dados Atomizados para Previsão de Vendas no Varejo Utilizando Redes Neurais*. Dissertação de Mestrado – Universidade de São Paulo, Faculdade de Economia, Administração e Contabilidade, Departamento de Administração, 2003.
- [Prechelt, 1994] PRECHELT, L. *Proben1 – A Set of Neural Network Benchmark Problems and Benchmarking Rules*. Universität Karlsruhe, Fakultät für Informatik, 1994.
- [Ribeiro, 1999] RIBEIRO, J. N. *Modelos de Redes Neurais Construtivas para Classificação e Verificação de Padrões*. Dissertação de Mestrado – Universidade Federal de Pernambuco, Centro de Informática, 1999.
- [Rivals, 1997] RIVALS, I.; PERSONNAZ, L. *Construction of confidence intervals for neural networks based on least squares estimation*. Artigo – Laboratoire d'Électronique, École Supérieure de Physique et de Chimie Industrielles, Paris, France, 1997.
- [Rumelhart, 1986] RUMELHART, D.; HINTON, G.; WILLIAMS, R. *Learning Internal Representations by Error Propagation*. In : Rumelhart & McClelland: *Parallel Distributed Processing - Explorations in the Microstructure of Cognition - Vol.1: Foundations*. Cambridge: MIT Press, 1986.

- [Seng, 1998] SENG, T. L.; KHALID, M.; YUSOF, R.; OMATU, S. *Adaptative Neuro-Fuzzy Control System by RBF and GRNN Neural Networks*. Artigo – Universiti Teknologi Malaysia, Center for Artificial Intelligent and Robotics, Malaysia, 1998.
- [Setiono, 2000] SETIONO, R.; THONG, J. Y. L. *An Approach to Generate Rules from Neural Networks for Regression Problems*. Artigo – National University of Singapore, School of Computing, SINGAPORE; Hong Kong University of Science and Technology, School of Business and Management, Kowloon, HONG KONG, 2001.
- [Simpson, 1990] SIMPSON, P. *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*. New York: Pergamon Press Inc., 1990.
- [Vieira, 2000] VIEIRA, R. S.; THOMÉ, A. C. G. *Avaliação de Redes Neurais Aplicadas à Previsão de Índices de Mercados de Ações*. Artigo – Universidade Federal do Rio de Janeiro, 2000.
- [Yonenaga, 2000] YONENAGA, W. H.; FIGUEIREDO, R. S. *Previsão do Preço da Soja Utilizando Redes Neurais*. Artigo – Universidade Federal de São Carlos, 2000.
- [Yunos, 1995] YUNOS, Z.M.; ZAID, S.T.; JAMALUDDIN, R.; SHAMSUDDIN, S.M.; SALLEHUDDIN, R.; Alwi, R. *Development Of Intelligent System For Predicting Stock Market Returns*. Artigo – Faculty of Computer Science & Information System University Technology of Malaysia, K.B. 791 81310 Skudai, Johor, Malaysia, 1995.

- [Zaneti, 1998] ZANETI, L. A.; ALMEIDA, F. C. *Exploração do Uso de Redes Neurais na Previsão do Comportamento de Ativos Financeiros*. Artigo – USP, Faculdade de Economia Administração e Contabilidade, 1998.