

UNIVERSIDADE DO SAGRADO CORAÇÃO

RAONI GHIOTTO

**APLICAÇÃO DE REDES NEURAIS ARTIFICIAIS
NA PRECIFICAÇÃO DE OPÇÕES**

BAURU

2008

UNIVERSIDADE DO SAGRADO CORAÇÃO

RAONI GHIOTTO

**APLICAÇÃO DE REDES NEURAIS ARTIFICIAIS NA
PRECIFICAÇÃO DE OPÇÕES**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Naturais como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob orientação da Prof^a. Ms. Patricia Bellin Ribeiro.

BAURU

2008

G424a

Ghiotto, Raoni

Aplicação de redes neurais artificiais na precificação de opções / Raoni Ghioto – 2008.
74f.

Orientadora: Profa. Ms. Patricia Bellin Ribeiro.

Trabalho de Conclusão de Curso (Bacharel em Ciências da Computação) - Universidade Sagrado Coração - Bauru - SP.

1. Redes neurais artificiais. 2. Rede neural. 3. Opções de compra. 4. Opções. 5. Precificação de opções. 6. Treinamento. 7. Validação. 8. Topologia. I. Ribeiro, Patricia Bellin. II. Título

DEDICATÓRIA

Dedico este trabalho à minha família, Renato (pai), Eunice (mãe), Luna (irmã) e Augusto (irmão).

AGRADECIMENTOS

Agradeço aos meus familiares Vó Mime, Rita, Karine, Tia Nenas e ao Robertinho da Bahia que sempre me receberam bem em suas casas.

À minha orientadora Patricia por estar comigo nesse trabalho desde o começo e pela dedicação em torná-lo viável.

Ao professor Ronaldo por ter aceitado ser meu orientador no primeiro semestre.

Ao professor Kelton por ser gente boa.

Ao coordenador do curso de ciência da computação, o professor Elvio, por sempre estar disponível para tirar minhas dúvidas referentes ao curso.

À minha amiga e irmã Camila pela força dada.

Ao André Pilastrri por ter sido companheiro de TCC durante a maior parte desse trabalho.

Aos amigos da faculdade: Julio, Thiago Leite, Gigliotti, Tucks, Futuro, Pascal, Silas, Glauber, Smallville, Fernando, Lucas, Pietro e também aos que não mencionei, mas que estudaram comigo.

Aos amigos de Presidente Alves: Ricardo, Leandrão, Ariovaldo, Huck, Dani, Thaty e Verena.

RESUMO

As redes neurais artificiais são utilizadas para resolver diversos problemas em várias áreas, sendo: a robótica, o processamento de imagens e sinais, o mercado financeiro, etc. O modelo de rede neural utilizado nesse trabalho foi o *Multi – layer perceptron*, por ser um dos mais usados por pesquisadores para solucionar problemas relacionados com o mercado de capitais. A rede neural artificial foi utilizada para se precificar as opções de compra da Vale. Os dados utilizados como atributos de entrada da rede neural foram obtidos da Bm&fbovespa (2008) e o método de validação utilizado foi o *cross validation*. A escolha da melhor topologia de rede neural foi feita com base na maior taxa de acertos da rede, obtidos pelo método de validação. Após a precificação, foram construídos gráficos comparando os preços das opções reais, observadas no mercado de opções, com os fornecidos pela rede neural. Através dos gráficos foi constatado que a rede neural conseguiu identificar se a tendência dos preços das opções é de alta ou de baixa. Portanto, com os resultados obtidos, a rede neural mostrou-se uma boa ferramenta de precificação de opções, sugerindo que os profissionais do mercado financeiro a utilizem em suas análises de mercado e que desenvolvam mais pesquisas com as redes neurais artificiais, a fim de obter melhores resultados.

Palavras-chave: Redes neurais artificiais, rede neural, opções de compra, opções, precificação de opções, treinamento, validação, topologia.

ABSTRACT

The artificial neural networks are used to solve various problems in various areas, including: the robotics, processing of images and signs, the financial market and etc. The model of neural network used in this study was the Multi - Layer Perceptron, as one of the most used by researchers to solve problems related to the capital market. The artificial neural network was used to price the call options of the Vale. The data used as input attributes of the neural network were obtained from the Bm&fbovespa (2008) and the method used was the validation of cross validation. The choice of the best neural network topology was based on the rate of hit obtained by the method of validation. After pricing, graphics were constructed by comparing the prices of real options, observed in the options market, with those provided by the neural network. Through graphic was found that the neural network was able to identify the trend of prices of the options is high or low. So, with the results, the neural network proved to be a good tool for pricing options, suggesting that the financial market practitioners to use in their analysis of the market and to develop more research with the artificial neural networks in order to obtain better results.

Key words: *artificial neural networks, neural network, calls, options, pricing of options, training, validation, topology.*

LISTA DE ILUSTRAÇÕES

Figura 2.1 - Exemplo de negociação das ações da Vale.....	16
Figura 2.2 - Exemplo de negociação de opções.	18
Figura 3.1 - Representação em diagrama em blocos do sistema nervoso	22
Figura 3.2 - Neurônio biológico.	23
Figura 3.3 - Sinapse.	24
Figura 3.4 - Neurônio artificial projetado por McCulloch e Pitts.	25
Figura 3.5 - Função de ativação degrau.	26
Figura 3.6 - Função de ativação logística.....	26
Figura 3.7 - Função de ativação linear.	27
Figura 3.8 - Função de ativação gaussiana.	27
Figura 3.9 - Rede feedforward de uma única camada.	28
Figura 3.10 - Rede feedforward de duas camadas.....	29
Figura 3.11 - Rede com recorrência entre saídas e camadas intermediárias.	29
Figura 3.12 - Rede com recorrência auto-associativa.	30
Figura 3.13 - Diagrama em blocos do aprendizado supervisionado.	32
Figura 3.14 - Aprendizado por reforço.	33
Figura 3.15 - Aprendizado não supervisionado.	34
Figura 4.1 - Dois tipos de problemas, linearmente e não - linearmente separáveis.	38
Figura 4.2 - Grafo arquitetural de um perceptron de múltiplas camadas com duas camadas ocultas.....	40
Figura 4.3 - Fluxo de processamento do algoritmo back –propagation.	41
Figura 5.1 - Local do site da BM&FBOVESPA onde foram coletados os dados.....	42
Figura 5.2 - Arquitetura da rede neural utilizada nesse trabalho.....	45
Figura 5.3 - Interface simplificada da rede MLP.....	47
Figura 5.4 - Estrutura dos cálculos realizados por uma rede MLP.....	50
Figura 5.5 - Dez arquivos de treinamento e validação	52

LISTA DE TABELAS

Tabela 1.1 – Trabalhos que aplicaram redes neurais artificiais ao mercado de opções	12
Tabela 2.1 – Estrutura do código da opção de compra em função da data de exercício.	17
Tabela 5.1 – Conjunto com todos os dados, incluindo os campos vazios.	44
Tabela 5.2 – Conjunto com os dados organizados sem os campos vazios.....	44
Tabela 6.1 – Topologias de redes neurais utilizadas nesse trabalho	53
Tabela 6.2 – Detalhes da topologia com a melhor taxa de acerto.	54
Tabela 6.3 – Novas taxas de acertos da melhor topologia.....	58
Tabela 6.4 – Novas taxas de acertos para as outras topologias..	58

LISTA DE GRÁFICOS

Gráfico 6.1 – Comparativo de preços entre a cotação real e os obtidos pela rede neural para a VALEG70.....	55
Gráfico 6.2 – Comparativo de preços entre a cotação real e os obtidos pela rede neural para a VALEG76.....	56
Gráfico 6.3 – Comparativo de preços entre a cotação real e os obtidos pela rede neural para a VALEH74.....	56
Gráfico 6.4 – Comparativo de preços entre a cotação real e os obtidos pela rede neural para a VALEH82.....	57

SUMÁRIO

1	Introdução.....	11
2	Bolsa de valores.....	14
2.1	Ações.....	15
2.2	Mercado de Opções.....	16
2.2.1	Tipos de opções.....	19
2.2.2	Variáveis que influenciam no preço da opção (prêmio).....	19
3	Redes Neurais Artificiais.....	20
3.1	Histórico.....	21
3.2	Analogia entre o cérebro humano e RNAs.....	22
3.3	Neurônios Biológicos.....	23
3.4	Neurônios artificiais.....	24
3.5	Função de ativação.....	25
3.6	Arquiteturas de Redes Neurais Artificiais e Aprendizagem.....	28
3.7	Aprendizagem.....	31
3.7.1	Aprendizado supervisionado.....	32
3.7.1.1	Correção de erros.....	33
3.7.1.2	Aprendizado por reforço.....	33
3.7.2	Aprendizado não-supervisionado.....	34
3.7.2.1	Aprendizado Hebbiano.....	35
3.7.2.2	Aprendizado por competição.....	36
4	Modelos de Redes Neurais Artificiais.....	37
4.1	Perceptron de uma única camada.....	37
4.2	Redes Adaline (Adaptive Linear Element).....	38
4.3	Perceptron de múltiplas camadas (MLP – Multi – layer perceptron).....	39
4.3.1	Algoritmo de <i>back-propagation</i>	41
5	Materiais e Métodos.....	42
5.1	Base de dados.....	42
5.2	Rede Neural Artificial.....	45
5.3	Normalizações dos dados.....	46
5.4	Softwares utilizados para treinamento e validação.....	46
5.5	Validação cruzada.....	51
6	Resultados.....	53
7	Conclusão.....	60
	Referências bibliográficas.....	62
	Bibliografia consultada.....	65
	Apêndice A – Código fonte comentado do programa implementado em Java para realizar a validação.....	66
	Apêndice B – Tabela com os dados da validação da melhor topologia de rede neural.....	70

1 Introdução

O comportamento das opções (derivativo de ações da bolsa de valores) obedece a padrões complexos devido ao grande número de parâmetros relevantes, sendo a maioria deles de natureza aleatória.

Um dos modelos mais utilizados para precificar as opções foi criado por Black e Scholes (B&S) (1973), que através de variáveis como: preço da ação, prazo de vencimento da opção, volatilidade do ativo e etc., pode-se obter uma estimativa do preço teórico da opção.

O modelo de Black e Scholes (1973) foi inicialmente aplicado em opções europeias de compra e de venda e depois passaram a ser amplamente utilizadas. No entanto não se mostrou completamente eficaz na prática levando a diferenças significativas entre o valor da opção calculado por esse modelo e o valor realmente verificado no mercado.

“Uma premissa básica do B&S é que a volatilidade do preço das ações é constante. Porém, Adler *et al.* (1999) e La-nari (2000) constataram que esse princípio não é observado no mercado brasileiro, resultando em desvios empíricos do método de Black e Scholes”. (ADLER *et al.*, 1999; LA-NARI, 2000 apud FREITAS e SOUZA, 2002, p. 63)

Uma alternativa a formula de precificação de opções proposto por Black e Scholes (1973) é a utilização de redes neurais artificiais (RNAs), que de acordo com estudos realizados tem se apresentado como solução mais apropriada. A rede neural artificial (RNA) consegue precificar as opções através de um processo de aprendizagem, ou seja, a rede irá aprender com os dados recebidos pela camada de entrada e com isso precificará as opções.

Para Braga *et al.* (2007) as principais tarefas que as RNAs podem executar são: classificação, categorização, aproximação, previsão e otimização. Através dessas tarefas as RNAs podem ser aplicadas no setor financeiro, no setor elétrico, na engenharia, na bioinformática, etc.

Fernandes (2000) em seu trabalho utilizou uma RNA para precificar as opções da Telebrás. Ele concluiu que a RNA na precificação de opções, com liquidez, deve ser uma ferramenta alternativa para os investidores e que necessita de mais estudos.

Freitas e Souza (2002) também utilizam em seus trabalhos uma RNA para precificar as opções da Telebrás e compará-las com o modelo de Black e Scholes (1973) e o valor real

observado no mercado de opções. Concluíram que o modelo de RNAs foi superior ao modelo de Black e Scholes (1973).

GRADOJEVIC et al. (2007) precificaram as opções de compra do índice S&P500 (índice calculado pela Standard e Poors sobre 500 ações negociadas nos EUA (Bm&fbovespa (2008)) e concluíram que a precificação de opções com redes neurais possui um futuro promissor.

A Tabela 1.1 exibe algumas aplicações das RNAs no mercado de opções:

TABELA 1.1 – Trabalhos que aplicaram redes neurais artificiais ao mercado de opções.

Autoria	Descrição
Hutchinson <i>et al.</i> (1994)	Utilizaram redes neurais na precificação e no <i>hedge</i> de opções sobre futuro de índice S&P 500, entre 1987 e 1991.
Lajbcygier <i>et al.</i> (1996)	Aplicando-se ao mercado australiano de derivativos, compararam o modelo de Hutchinson <i>et al.</i> (1994) com um de quatro parâmetros de entrada: S/X , T , r e α .
Garcia e Gençay (2000)	Propuseram variação do modelo de Hutchinson <i>et al.</i> (1994), dividindo a função de precificação em duas partes: uma controlada pela razão S/X e outra, por uma função do tempo até o vencimento.
Malliaris e Salchenberger (1996)	Desenvolveram um modelo de redes neurais <i>back-propagation</i> para estimar a volatilidade futura, aplicando-o ao mercado de opções sobre índice S&P 100.
Yao <i>et al.</i> (2000)	Construíram modelos de redes neurais <i>back-propagation</i> para precificar opções sobre futuro de índice Nikkei 225, em negociação no Singapore Exchange-Derivatives Trading (SGX-DT).
QI <i>et al.</i> (1996)	Por meio de rede MLP, precificaram opções de compra sobre índice S&P 500, negociadas entre dezembro de 1994 e janeiro de 1995.
Ormoneit (1999)	Utilizou modelo de rede neural MLP na precificação e no <i>hedge</i> de opções de compra sobre DAX, índice alemão de ações.
Burgess e Refenes (1999)	Empregaram redes neurais para estimar a volatilidade implícita de opções sobre o índice Ibex35, que contém as 35 ações mais líquidas negociadas na bolsa de valores espanhola por meio do sistema Cats.

Fonte: FREITAS, 2002, p. 64

Dentre as vantagens deste modelo, Braga et al. (2007, p. 3) cita que “a solução de problemas por meio de RNAs é bastante atrativa, já que a forma como estes são representados internamente pela rede e o paralelismo natural inerente à arquitetura das RNAs criam a possibilidade de um desempenho superior ao dos modelos convencionais”. Outra importante vantagem sobre as RNAs é o poder de generalização, pois de acordo com Haykin (2001, p.28) “a generalização se refere ao fato de a rede neural produzir saídas adequadas para entradas que não estavam presentes durante o treinamento (aprendizagem)”.

Entretanto como desvantagem é possível apresentar o fato de que existe uma necessidade de uma base de dados substancial a fim de obter uma rede bem treinada. Ressaltando que essa não é apropriada para derivativos pouco negociados ou recentemente criados.

O objetivo desse trabalho é utilizar o modelo das redes neurais artificiais para estudar a precificação de opções, analisando o seu comportamento em função do tempo e realizando comparativo com as cotações reais de mercado. Com isso será verificado se a rede neural artificial é funcional no contexto do mercado de capitais. Espera-se com esse trabalho desenvolver uma ferramenta que contribua para a previsão do comportamento das opções que é de grande importância para quem atua no mercado de capitais.

Além deste capítulo o trabalho está composto de mais seis capítulos conforme descrição sumária abaixo:

Capítulo 2: Bolsa de valores.

Capítulo 3: Redes Neurais Artificiais.

Capítulo 4: Modelos de Redes Neurais Artificiais.

Capítulo 5: Materiais e Métodos.

Capítulo 6: Resultados.

Capítulo 7: Conclusão.

2 Bolsa de valores

Segundo o Portal do Investidor (2008) “as bolsas de valores são instituições administradoras de mercados”. Nas bolsas de valores ficam localizados os centros de negociação de valores mobiliários em que as compras e vendas podem ser negociadas eletronicamente. No Brasil a principal bolsa de valores é a Bovespa e o órgão responsável por fiscalizar as bolsas de valores é a Comissão de Valores Mobiliários (CVM).

Ainda segundo o Portal do Investidor “a principal função de uma bolsa de valores é proporcionar um ambiente transparente e líquido, adequado à realização de negócios com valores mobiliários”. Para que os investidores realizem operações de compra e venda de valores é necessário ser cliente de uma corretora de valores.

Os valores mobiliários negociados na bolsa estão especificados na informação da CVM abaixo:

O mercado de valores mobiliários brasileiro negocia, predominantemente, ações, debêntures e quotas de fundos de investimento. Entretanto, existem vários outros tipos de valores mobiliários. O art. 2º da Lei nº 6.385, de 07.12.76, com alterações feitas pela Lei nº 10.303, de 31.10.01, define como valores mobiliários:

I. as ações, debêntures e bônus de subscrição; II. os cupons, direitos, recibos de subscrição e certificados de desdobramento relativos aos valores mobiliários; III. os certificados de depósito de valores mobiliários; IV. as cédulas de debêntures; V. as cotas de fundos de investimento em valores mobiliários ou de clubes de investimento em quaisquer ativos; VI. as notas comerciais; VII. os contratos futuros, de opções e outros derivativos, cujos ativos subjacentes sejam valores mobiliários; VIII. outros contratos derivativos, independentemente dos ativos subjacentes; (Portal do Investidor, acesso em 14 jun. 2008)

Dentre esses, serão utilizadas nesse trabalho as ações e os contratos de opções.

2.1 Ações

De acordo com a Bm&fbovespa (2008) “ações são títulos nominativos negociáveis que representam para quem as possui, uma fração do capital social de uma empresa. Ação é um pedacinho de uma empresa”.

O Portal do investidor (2008) informa que para uma empresa ter suas ações negociadas na bolsa é preciso que ela seja aberta ou pública, ou seja, que o público em geral detenha suas ações. A empresa deverá também obedecer às normas e regras estabelecidas pela bolsa.

Para a Bm&fbovespa (2008) as ações podem ser:

- Ordinárias, que dão o direito a voto, àqueles que as possuem, nas assembleias deliberativas da companhia;
- Preferenciais, que não dão o direito a voto, mas tem preferência na distribuição de resultados da empresa ou no reembolso do capital em caso de liquidação da companhia.

Nesse trabalho serão analisadas as ações da empresa Vale PN, pelo fato de ser uma das mais líquidas da Bovespa, ou seja, com grande volume de negócios diariamente.

A Figura 2.1 apresenta um exemplo real de negociação das ações da Vale. Nela identificam-se, no lado esquerdo do painel principal, as ofertas de compra e do lado direito as ofertas de vendas. As corretoras de valores são as responsáveis pelas ofertas de compra ou venda de seus clientes. Quando um cliente dá uma ordem de compra ou venda, será o nome de sua corretora que aparecerá como compradora ou vendedora das ações. No canto direito abaixo da Figura 2.1 têm-se os negócios que foram realizados.

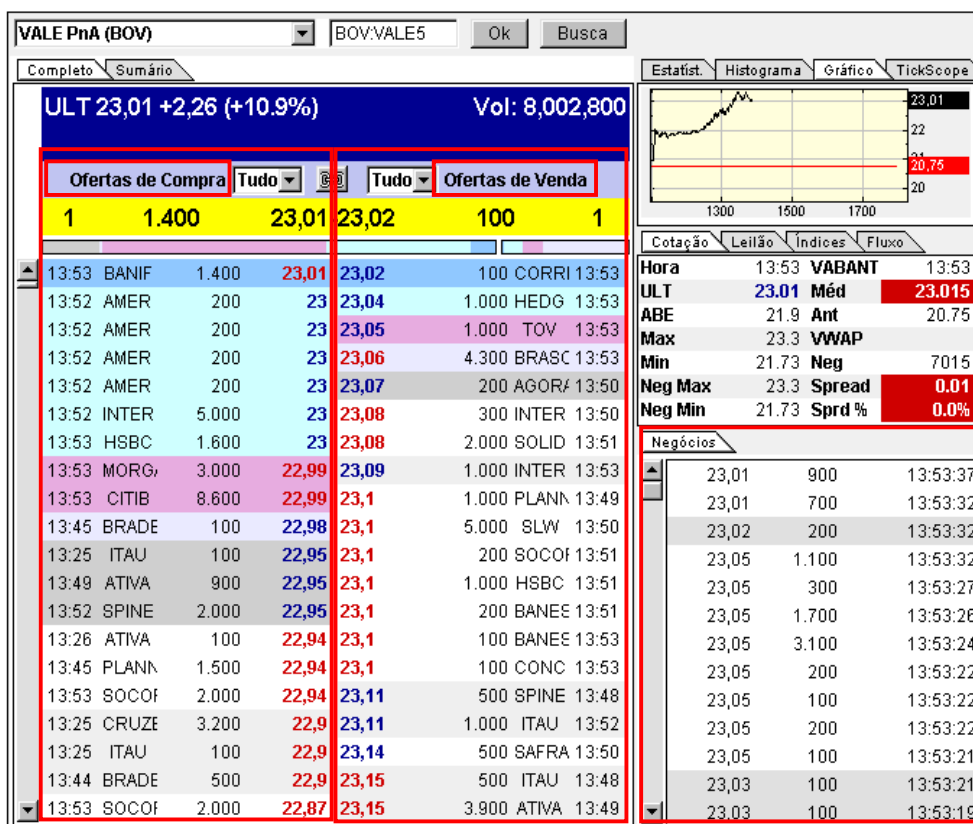


Figura 2.1 – Exemplo de negociação das ações da Vale.

Fonte: Disponível em <<http://br.advfn.com/>>. Acesso em 23 nov. 2008

2.2 Mercado de Opções

O mercado de opções para a Bovespa (2000) “é o mercado em que são negociados direitos de compra ou venda de um lote de ações, com preços e prazos de exercício preestabelecidos”.

Conforme Issa (2007) as opções são instrumentos derivativos, ou seja, o preço de uma opção e suas características de negociação está vinculado ao ativo que originou essa opção. Por exemplo: uma opção da Vale PN está ligada às ações da Vale PN. O mesmo autor diz que “uma opção de um ativo adjacente será o direito de comprar o ativo (opção de compra) ou o direito de vender o ativo (opção de venda) a determinado preço e dentro de determinado período de tempo no futuro”.

Pela denominação das opções na Bovespa, o código de uma opção é composto por um número e uma letra. O número indica o preço de exercício e a letra identifica se é uma opção de compra ou de venda e também o mês de vencimento (BOVESPA, 2000).

A Tabela 2.1 exhibe a estrutura do código da opção de compra de acordo com a data de exercício (vencimento). Esse tipo de opção será pesquisada nesse trabalho.

TABELA 2.1 – Estrutura do código da opção de compra em função da data de exercício

Opção de compra	Vencimento
A	Janeiro
B	Fevereiro
C	Março
D	Abril
E	Maiο
F	Junho
G	Julho
H	Agosto
I	Setembro
J	Outubro
K	Novembro
L	Dezembro

Fonte: Bovespa, 2000, p. 6

Exemplo:

VALEF52 – Opção de compra sobre a Vale PN, com vencimento em junho e preço de exercício a R\$ 52,00 por ação.

VALEG48 - Opção de compra sobre a Vale PN, com vencimento em julho e preço de exercício a R\$ 48,00 por ação.

O preço de exercício é o valor que o proprietário da opção pagará pelas respectivas ações adjacentes (ações ligadas às opções) até a data de exercício, caso ele deseje exercer esse direito (ISSA, 2007).

Exemplo: a opção de compra da VALEF52 dá o direito de comprar as ações da VALE PN a R\$ 52,00, independente do valor real da ação, até o mês de junho.

Data de exercício é o dia em que a opção deixará de existir, que no Brasil é sempre na terceira segunda – feira do mês.

A Figura 2.2 mostra um exemplo real de negociação da opção de compra da Vale. O código VALEL26 indica que é uma opção com preço de exercício igual a R\$ 26,00 e vencimento em dezembro de 2008.

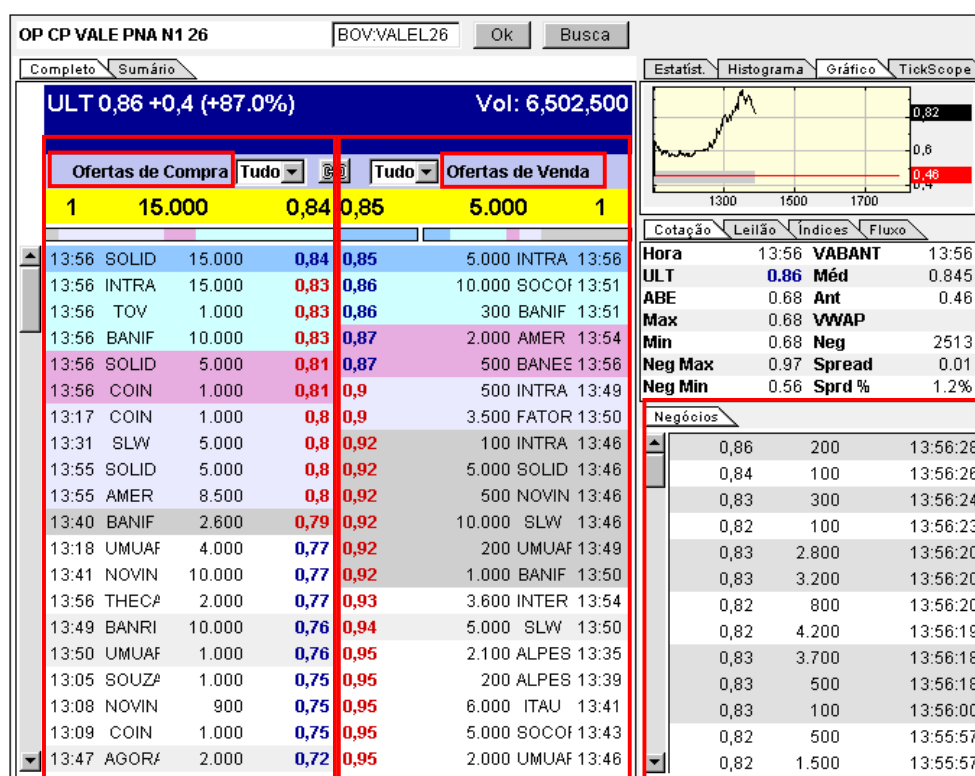


Figura 2.2 – Exemplo de negociação de opções.

Fonte: Disponível em <<http://br.advfn.com/>>. Acesso em 23 nov. 2008

Nesse trabalho será utilizada a opção de compra da Vale por ser uma das mais líquidas da bolsa, ou seja, com grande número de negócios.

2.2.1 Tipos de opções

No mercado de opções encontra-se dois tipos de opções:

- Opções americanas: são essas as opções negociadas no Brasil. O titular pode exercer o seu direito de comprar as ações – objeto (ações relacionadas àquelas opções que o titular possui) a qualquer momento até a data de exercício.
- Opções européias: a diferença é que o titular só poderá exercer o direito de comprar a ação adjacente (ação cuja opção foi derivada) na data do exercício.

Issa (2007) relata também que a opção pode ser classificada de acordo com o preço da ação adjacente em:

- Opção de compra dentro do valor: conhecida também como opção de compra *in the money* (ITM), o preço da ação – objeto é maior que o preço de exercício da opção.
- Opção de compra no valor ou *at the money* (ATM): o preço da ação – objeto é igual ao preço de exercício da opção.
- Opção de compra *out of the money* (OTM): é a opção que está fora do valor, ou seja, o preço da ação – objeto está abaixo do preço de exercício da opção.

2.2.2 Variáveis que influenciam no preço da opção (prêmio)

O prêmio de uma opção nada mais é do que o valor pago quando se compra uma opção. Esse valor sofre influência das seguintes variáveis: preço do ativo adjacente, preço de exercício da opção, tempo que resta até a data de exercício, alguns fatores externos como as taxas de juros e a volatilidade da ação – objeto.

Dentre as variáveis é importante destacar que a volatilidade da ação – objeto (taxa de variação no preço da ação) é a única que não se tem conhecimento, pois não tem como saber o preço da ação no próximo dia, ou seja, no futuro. Para resolver esse problema, utiliza-se a volatilidade histórica, que é a taxa de variação da ação – objeto ocorrida no passado (Issa, 2007).

3 Redes Neurais Artificiais

Na literatura encontram-se várias definições de Redes Neurais artificiais (RNAs), como por exemplo:

As redes neurais artificiais consistem em um método para solucionar problemas de inteligência artificial, construindo um sistema que tenha circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. São técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento, enquanto que o cérebro de um mamífero pode ter muitos bilhões de neurônios (TATIBANA e KAETSU, 2008).

Ou:

RNAs são sistemas paralelos distribuídos compostos por unidades de processamento simples (neurônios artificiais) que calculam determinadas funções matemáticas (normalmente não-lineares). Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos essas conexões estão associadas a pesos, os quais armazenam o conhecimento adquirido pelo modelo e servem para ponderar a entrada recebida por cada neurônio da rede. (BRAGA et al., 2007, p. 3).

3.1 Histórico

Em 1943 McCulloch e Pitts criaram o primeiro modelo de um neurônio artificial. Warren McCulloch, psiquiatra e neurofisiologista, se dedicou a pesquisar a representação de um evento no sistema nervoso. Walter Pitts foi um excelente matemático que se uniu a McCulloch em 1942.

Sobre o modelo pode – se dizer que:

McCulloch e Pitts descrevem um cálculo lógico das redes neurais que unificava os estudos de neurofisiologia e da lógica matemática. Eles assumiam que o seu modelo formal de um neurônio seguia uma lei “tudo ou nada”. Com um número suficiente dessas unidades simples e com conexões sinápticas ajustadas apropriadamente e operando de forma síncrona, McCulloch e Pitts mostraram que uma rede assim constituída realizaria, a princípio, a computação de qualquer função computável (HAYKIN, 2001, p. 63).

Segundo Haykin (2001) o segundo trabalho mais significativo em redes neurais foi publicado por Hebb em 1949. Ele publicou o livro *The Organization of Behavior*. Nesse livro Hebb demonstra a formulação de uma regra de aprendizagem para a modificação sináptica, que é feita através da variação dos pesos de entrada do neurônio.

Conforme Braga et al. (2007) em 1958 Frank Rosenblatt publicou em seu trabalho (*The Perceptron: a probabilistic model for information storage and organization in the brain*) um novo modelo para o problema de reconhecimento de padrões com a utilização de redes neurais. Esse modelo é o perceptron, que é um método de aprendizagem supervisionada. Em 1969 Minsky e Papert publicaram um livro demonstrando que existem limites de cálculo sobre o modelo do perceptron de única camada. Com essa publicação, a abordagem conexionista ficou adormecida nos anos 70. John Hopfield em 1982 publicou um artigo que fez com que as pesquisas sobre redes neurais artificiais retornassem e também abriu espaço para os físicos pesquisarem nessa área. Ainda conforme Braga et. al. “o grande feito de Hopfield foi, sem dúvida, mostrar a relação entre redes recorrentes auto-associativas e sistemas físicos, o que também abriu espaço para a utilização de teorias correntes da física para estudar tais modelos”.

Em 1986, foi publicado um livro em dois volumes de Rumelhart e McClelland: *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*. Segundo Haykin (2001, p. 68) “este livro exerceu uma grande influência na utilização da aprendizagem por *back-*

propagation, que emergiu como o algoritmo de aprendizagem mais popular para o treinamento de perceptrons de múltiplas camadas”.

Por fim temos que:

Talvez mais do que qualquer outra publicação, o artigo de 1982 de Hopfield e o livro em dois volumes de 1986 de Rumelhart e McClelland foram as publicações mais influentes, responsáveis pelo ressurgimento do interesse em redes neurais nos anos 1980. As redes neurais certamente trilharam um longo caminho desde os dias iniciais de McCulloch e Pitts. De fato, elas se estabeleceram como um tema interdisciplinar com raízes profundas em neurociências, psicologia, matemática, ciências físicas e engenharia. É desnecessário dizer que elas estão aqui para ficar e que continuarão a crescer em teoria, projeto e aplicações. (HAYKIN, 2001, p. 69).

3.2 Analogia entre o cérebro humano e RNAs

O sistema nervoso humano pode ser dividido em três estágios (Haykin, 2001), o diagrama em blocos da Figura 3.1 mostra de forma clara esses três estágios. Nela a rede neural está no centro representando o cérebro humano, ela recebe as informações, processa as e toma as decisões apropriadas. Na figura existem dois conjuntos de setas. Um dos conjuntos de setas que vai da esquerda para a direita indica a transmissão para frente das informações. O outro que vai da direita para a esquerda indica a presença de realimentação no sistema. Os receptores recebem estímulos do corpo humano ou do meio externo e transforma – os em impulsos elétricos que são transmitidos para a rede neural. Os atuadores são responsáveis por gerar respostas de saída, isso de acordo com os impulsos elétricos recebidos da rede neural.

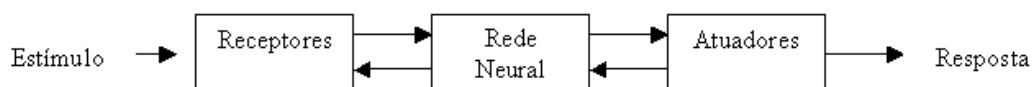


Figura 3.1 – Representação em diagrama em blocos do sistema nervoso.
Fonte: HAYKIN, 2001, P.32

3.3 Neurônios Biológicos

De acordo com Braga et al. (2007, p. 5) “o cérebro humano contém em torno de 10^{11} neurônios, sua célula fundamental”. Os neurônios se comunicam continuamente e em paralelo. A base dos estudos sobre RNAs é composta pela estrutura individual, a topologia das conexões e do comportamento de um neurônio biológico.

Na Figura 3.2 encontra - se: a soma que é o corpo celular do neurônio e é também onde está localizado o núcleo do neurônio, os dendritos que realizam a comunicação com os outros neurônios e conduzem os impulsos nervosos até a soma e o axônio que é o mais longo e tem a função de transmitir as informações, impulsos nervosos, para o dendrito de outro neurônio conectado a ele.

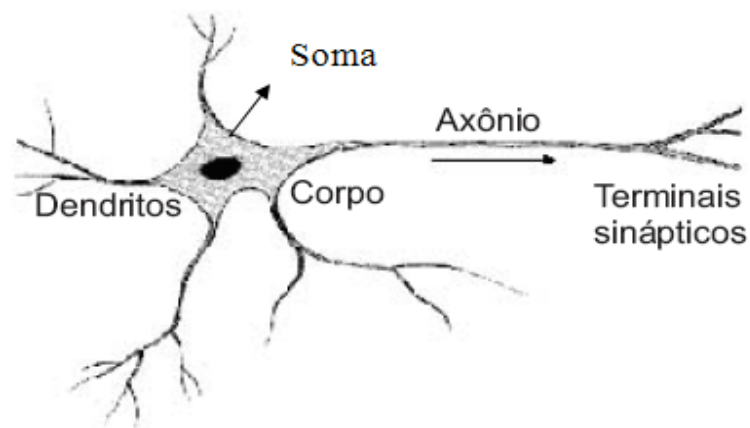


Figura 3.2 – Neurônio biológico.

Fonte: Disponível em: <http://www.scielo.br/scielo.php?pid=S0100-19652006000100003&script=sci_arttext>. Acesso em 15 jun. 2008

A conexão entre o axônio de um neurônio e o dendrito de outro é chamado de sinapse. As sinapses são responsáveis pelo controle de fluxo da informação, impulsos nervosos, entre os neurônios.

A Figura 3.3 mostra a sinapse do neurônio biológico.

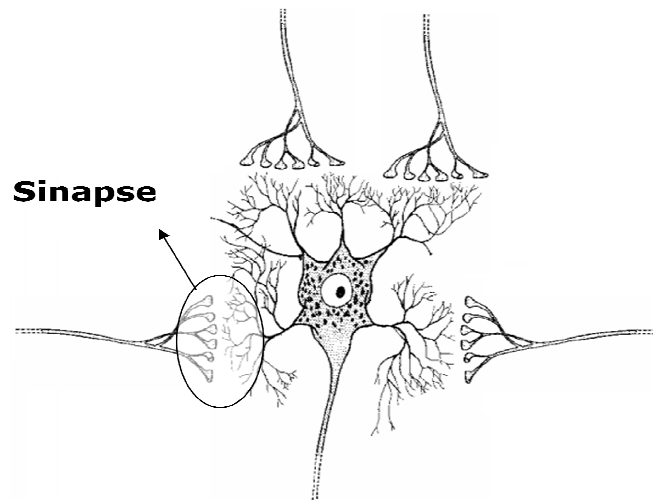


Figura 3.3 – Sinapse.

3.4 Neurônios artificiais

Segundo Braga et al. (2007) as redes neurais artificiais tentam reproduzir as funções das redes biológicas, isso através da implementação do seu comportamento funcional e da sua dinâmica. Em um neurônio artificial os dendritos correspondem às entradas de dados, as sinapses estão relacionadas aos pesos (utilizado junto com os dados de entrada, indicam se o neurônio artificial irá disparar) do neurônio artificial, os axônios fornecem os dados de saída que servirão de entrada para outro neurônio artificial, se houver neurônios conectados.

A Figura 3.4 mostra um neurônio artificial projetado por McCulloch e Pitts (1943), sendo que essa estrutura permanece até hoje, nele os dados de entrada estão representados pela letra x . A letra w indica os pesos que representam as sinapses em um neurônio biológico. O símbolo \sum representa a soma ponderada dos dados de entradas. A função de ativação é representada por $f(a)$, essa função é responsável por ativar ou não a saída, pois dependerá da soma ponderada dos dados de entrada. A letra y simboliza os dados de saída, que resultam da função de ativação.

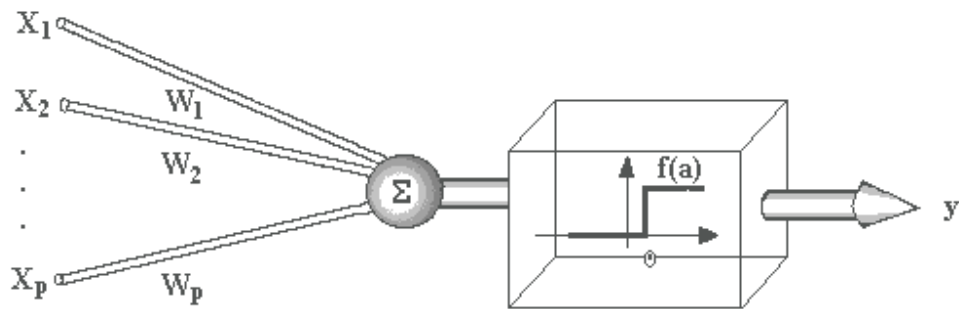


Figura 3.4 - Neurônio artificial projetado por McCulloch e Pitts.

Fonte: Disponível em: <<http://www.din.uem.br/ia/neurais/#indice>>. Acesso em 10 jun. 2008

3.5 Função de ativação

A função de ativação é responsável por ativar ou não a saída, pois dependerá da soma ponderada dos dados de entrada. Caso a saída seja ativa, o resultado da função de ativação será a informação de saída.

Conforme Braga et al. (2007) nas figuras abaixo são mostrados os gráficos das funções de ativação mais comuns.

A Figura 3.5 representa a função degrau para $\theta = 3$, descrita pela fórmula:

$$f(u) = \begin{cases} 1 & \text{se } \sum_{i=1}^n x_i w_i \geq \theta \\ 0 & \text{se } \sum_{i=1}^n x_i w_i < \theta \end{cases}$$

onde $\sum_{i=1}^n x_i w_i$ e θ é o limiar (*threshold*) da função de ativação.

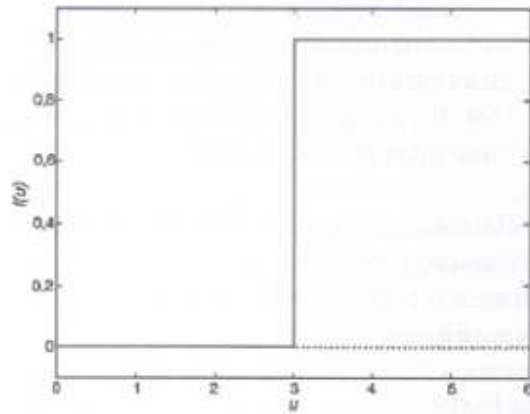


Figura 3.5 – Função de ativação degrau.
Fonte: BRAGA et al., 2007, p. 10

Na Figura 3.6 tem-se uma função de ativação logística, cuja equação é:

$$f(u) = \frac{1}{1 + e^{-\beta u}}$$

onde β é a inclinação da função.

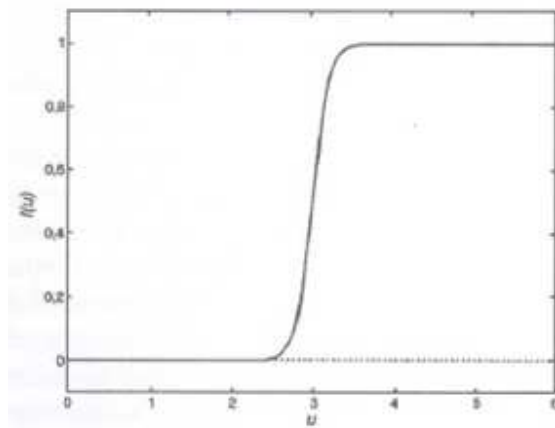


Figura 3.6 – Função de ativação logística
Fonte: BRAGA et al., 2007, p. 10

Na Figura 3.7, está mostrada a função de ativação linear dada pela equação da reta:

$$f(u) = u$$

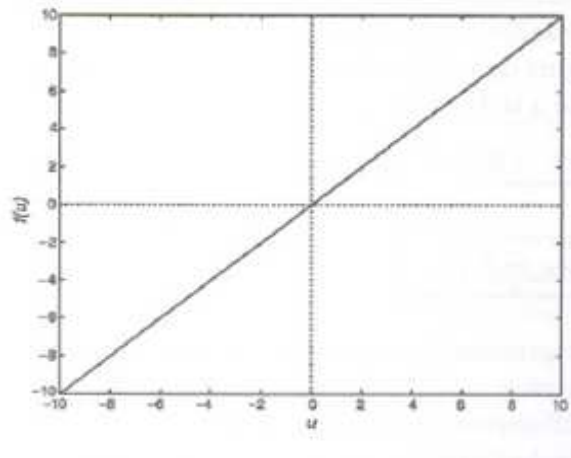


Figura 3.7 – Função de ativação linear.
 Fonte: BRAGA et al., 2007, p. 10

A Figura 3.8 mostra a função de ativação gaussiana que é utilizada em redes neurais artificiais do tipo *Radial Basis Functions* (RBF), cuja equação é:

$$f(u) = e^{\left[\frac{-(u - \mu)^2}{r^2} \right]}$$

onde μ é o centro (ponto médio) e r é o raio de abertura da função.

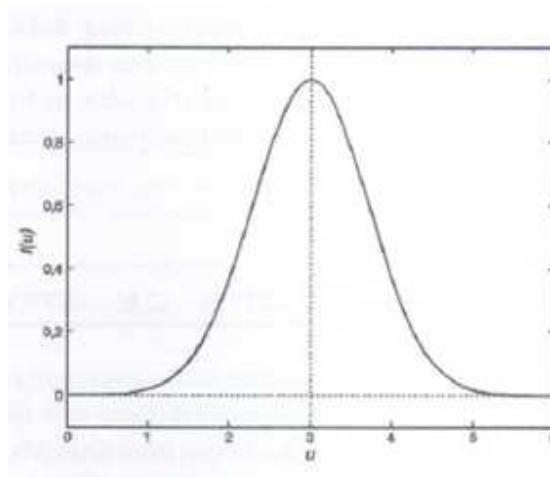


Figura 3.8 – Função de ativação gaussiana.
 Fonte: BRAGA et al., 2007, p. 10

3.6 Arquiteturas de Redes Neurais Artificiais e Aprendizagem

Um único neurônio artificial não é capaz de resolver problemas complexos, mas um conjunto de neurônios conectados em camadas é capaz de resolver estes problemas complexos. Em consequência disso, existem vários tipos de estruturas de RNAs.

A Figura 3.9 apresenta uma rede neural artificial com uma única camada, alimentada para frente (*feedforward*)”.

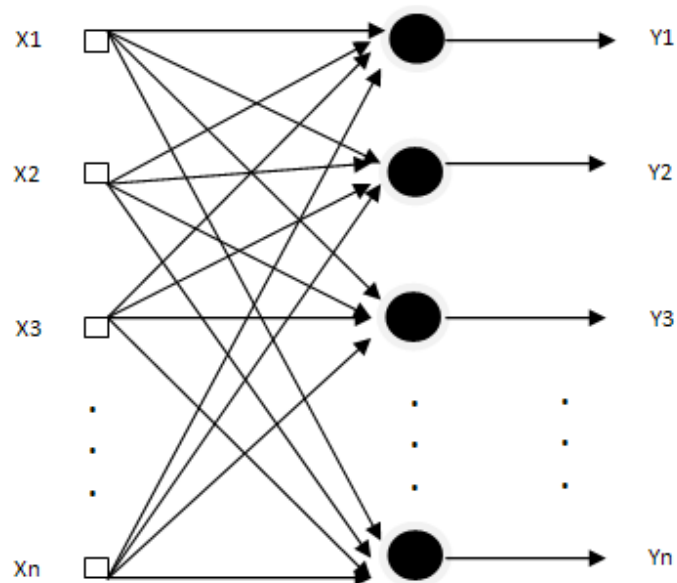


Figura 3.9 – Rede feedforward de uma única camada
Fonte: BRAGA et al., 2007, p. 11

A Figura 3.10 possui uma camada a mais em relação ao da Figura 3.9 oferecendo maior capacidade computacional.

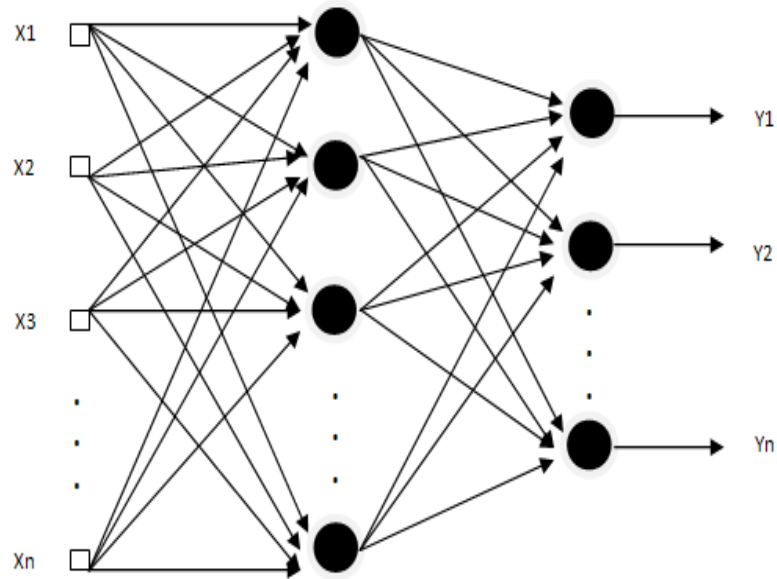


Figura 3.10 - Rede feedforward de duas camadas
 Fonte: BRAGA et al., 2007, p. 11

Na rede neural da Figura 3.11 a saída dependerá da entrada e também do seu valor atual, “essa estrutura de RNA é utilizada na resolução de problemas que envolvam processamento temporal, como em previsão de eventos futuros”.

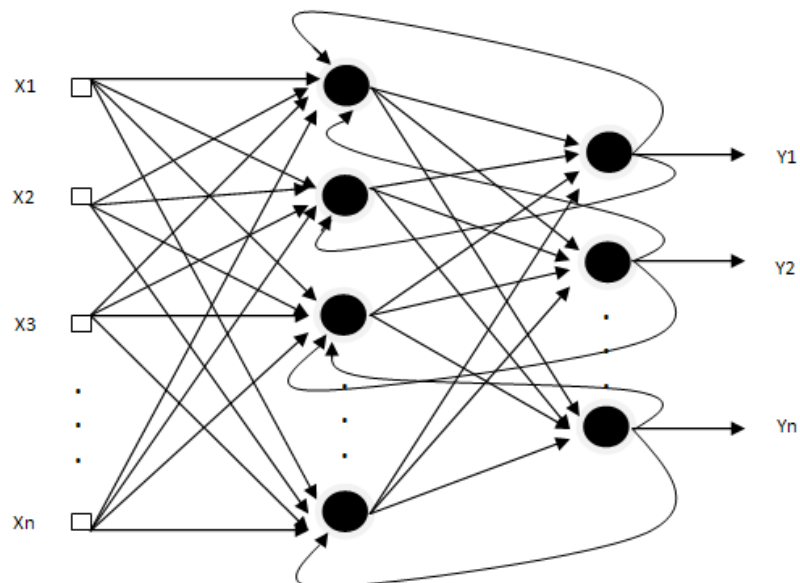


Figura 3.11 – Rede com recorrência entre saídas e camadas intermediárias
 Fonte: BRAGA et al., 2007, p. 11

Por fim a arquitetura de rede neural da Figura 3.12 “possui um único nível de neurônios, em que a saída de cada um deles está conectada às entradas de todos os outros”.

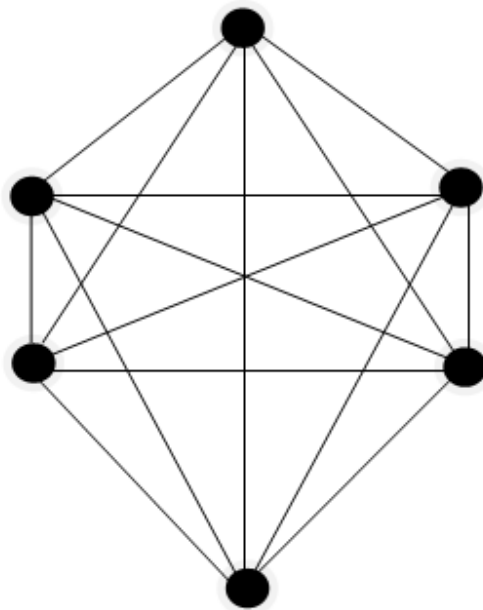


Figura 3.12 – Rede com recorrência auto-associativa
Fonte: BRAGA et al., 2007, p. 11

3.7 Aprendizagem

Aprendizagem é uma das mais importantes propriedades de uma rede neural artificial. Ela tem a função de ajustar os pesos das sinapses para que a rede forneça uma saída adequada. A cada processo de aprendizagem a rede neural adquire mais conhecimento e fica mais bem adaptada.

Uma definição de aprendizagem é a de que:

Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre (HAYKIN, 2001, p.75).

Ou:

Aprendizado é o processo pelo qual os parâmetros livres de uma rede neural são ajustados por meio de uma forma continuada de estímulo pelo ambiente externo, sendo o tipo específico de aprendizado definido pela maneira particular como ocorrem os ajustes dos parâmetros livres (BRAGA et al., 2007, p.12).

O aprendizado de uma rede neural artificial é feito pela implementação de um algoritmo de aprendizagem. A diferença entre esses algoritmos está na forma em que os pesos das sinapses são calculados. Braga et al. (2007, p. 13) cita que “há vários algoritmos diferentes para treinamento de redes neurais, podendo os mesmos serem agrupados em dois paradigmas principais: aprendizado supervisionado e aprendizado não-supervisionado”.

3.7.1 Aprendizado supervisionado

Este paradigma de aprendizagem funciona da seguinte forma: a rede recebe os dados do ambiente externo e obtém uma resposta como saída. Essa resposta é comparada com a resposta desejada, fornecida pelo professor. Através desta comparação, gera-se um sinal de erro que servirá para que alterações nos pesos das sinapses da rede neural sejam realizadas, a fim de que a rede forneça uma resposta o mais próximo da resposta desejada.

A Figura 3.13 exhibe: um professor que tem a função de supervisionar a rede e fornecer a resposta desejada. O sistema de aprendizagem recebe os dados do meio externo e fornece a resposta real, o sinal de erro que segundo Haykin (2001, p. 88) “[...] é definido como a diferença entre a resposta desejada e a resposta real da rede”.

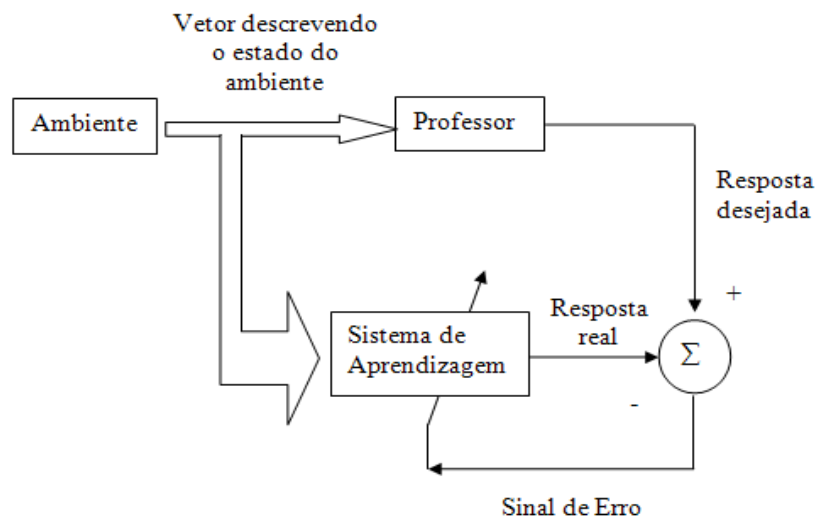


Figura 3.13 – Diagrama em blocos do aprendizado supervisionado.
Fonte: HAYKIN, 2001, p. 88

3.7.1.1 Correção de erros

É um algoritmo de aprendizado baseado no paradigma de aprendizagem supervisionado. O objetivo desse algoritmo é diminuir o sinal de erro, que é a comparação da resposta real com a resposta desejada. Para Haykin (2001) o sinal de erro aciona um mecanismo de controle, que tem a finalidade de ajustar os pesos sinápticos do neurônio. Esses ajustes são projetados para aproximar passo a passo o sinal de saída da resposta desejada.

3.7.1.2 Aprendizado por reforço

De acordo com Braga et al. (2007, p.15) “o aprendizado por reforço é muitas vezes considerado na literatura como um caso particular de aprendizado supervisionado”, pois nesse tipo de aprendizado as ações executadas pela rede, que são consideradas boas, são maximizadas, ou seja, reforça – se essas ações. Já em um aprendizado supervisionado os pesos são alterados em função de um critério adotado, por exemplo, o sinal de erro.

A Figura 3.14 demonstra de forma simples o funcionamento do aprendizado por reforço, em que o crítico exerce função semelhante ao do supervisor no aprendizado supervisionado.

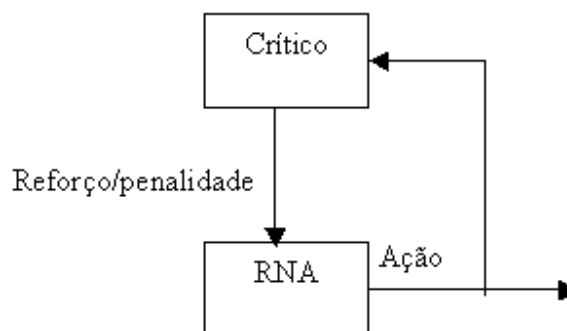


Figura 3.14 – Aprendizado por reforço.
Fonte: BRAGA et al., 2007, p.16

3.7.2 Aprendizado não-supervisionado

Nesse paradigma de aprendizado não existe um professor supervisionando a rede. Conforme Braga et al. (2007, p. 16) diz: “neste esquema de treinamento somente os padrões de entrada estão disponíveis para a rede, ao contrario do aprendizado supervisionado, cujo conjunto de treinamento possui pares de entrada e saída”. Portanto o aprendizado não supervisionado é aplicado para resolver problemas em que só se tem os dados de entrada.

Na Figura 3.15 encontra-se um esquema do aprendizado não supervisionado, onde a rede neural artificial recebe os dados do meio externo gerando assim uma resposta.

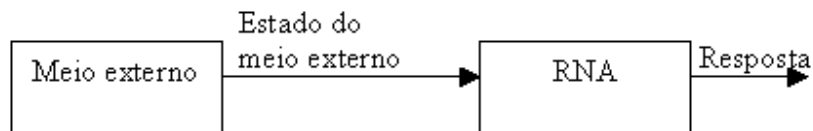


Figura 3.15 - Aprendizado não supervisionado
Fonte: Braga et al. (2007) p.17

3.7.2.1 Aprendizado Hebbiano

Haykin (2001, p. 80) comenta que “o postulado de aprendizado de Hebb é a mais antiga e mais famosa de todas as regras de aprendizagem”.

Um trecho do livro de Hebb (*The Organization of Behavior*) está descrito abaixo:

Quando um axônio da célula A está perto o suficiente para excitar uma célula B e participa do seu disparo repetida ou persistentemente, então algum processo de crescimento ou modificação metabólica acontece em uma das células ou em ambas, de tal forma que a eficiência de A como uma das células que dispara B é aumentada (HEBB, 1949, apud HAYKIN, 2001, p. 80).

Hebb descreve essa aprendizagem a nível celular, ou seja, neurobiologicamente. Desse conceito foi criada uma regra que está dividida em duas partes:

1. Se dois neurônios em ambos os lados de uma sinapse (conexão) são ativadas simultaneamente (i.e., sincronamente), então a força daquela sinapse é seletivamente aumentada.
2. Se dois neurônios em ambos os lados de uma sinapse são ativados assincronamente, então aquela sinapse é seletivamente enfraquecida ou eliminada (STENT, 1973; CHANGEUX e DANCHIN, 1976, apud HAYKIN, 2001, p.80).

Em Braga et al. (2007, p. 17) encontra-se a seguinte definição para o aprendizado Hebbiano: “[...] a regra de aprendizado de Hebb propõe que o peso de uma conexão sináptica deve ser ajustado se houver sincronismo entre os “níveis de atividade” da entrada e saída”. Ocorrerá um fortalecimento da sinapse quando dois neurônios forem ativados sincronicamente, sendo que eles estejam posicionados em lados diferentes dessa sinapse.

3.7.2.2 Aprendizado por competição

Segundo Haykin (2001) ao contrário do aprendizado hebbiano em que vários neurônios de saída podem estar ativos sincronicamente, na aprendizagem por competição apenas um neurônio de saída será ativo. Para se tornar ativo os neurônios de saída competem entre si. Os neurônios competem entre si.

De acordo com Braga et al. (2007, p. 17) a idéia de aprendizado por competição é, “[...] dado um padrão de entrada, fazer com que as unidades de saída disputem entre si para serem ativadas”. Conseqüentemente a unidade de saída vencedora será ativada e terá seus pesos alterados.

4 Modelos de Redes Neurais Artificiais

Existem diversos modelos de redes neurais que utilizam o aprendizado supervisionado e não-supervisionado. Nesse capítulo serão apresentados alguns modelos de aprendizado supervisionado com ênfase no modelo de rede neural *Multi – layer perceptron* (MLP), pois foi esse o utilizado nesse trabalho.

4.1 Perceptron de uma única camada

Rosenblatt (1958) propôs o perceptron como o primeiro modelo para a aprendizagem supervisionada, que para Haykin (2001, p. 143) “o perceptron é a forma mais simples de uma rede neural usada para a classificação de padrões ditos linearmente separáveis (i.e., padrões que se encontram em lados opostos de um hiperplano)”. Ele consiste de apenas um único neurônio e através do treinamento seus pesos sinápticos serão ajustados.

De acordo com Bishop (1995) a rede neural perceptron de uma única camada foi utilizada para classificar problemas, no qual as entradas eram habitualmente imagens binárias de caracteres ou formas simples. O perceptron de uma única camada possui uma pequena capacidade computacional.

Baseados nos conceitos de Haykin (2001) têm-se que o perceptron é uma rede neural artificial capaz de resolver problemas linearmente separáveis e incapaz de solucionar problemas não - linearmente separáveis.

A Figura 4.1 apresenta esses dois tipos de problemas. A fronteira de decisão classifica os padrões em duas classes. Sendo a primeira da esquerda um problema linearmente separável e a da direita não – linearmente separável. O responsável pelo deslocamento da fronteira de decisão em relação à origem é o bias.

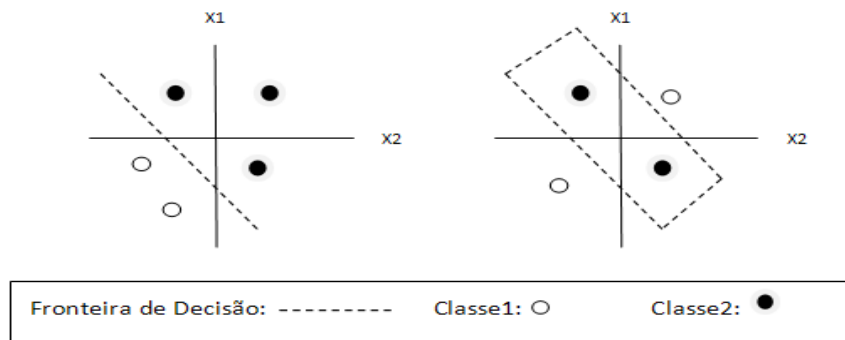


Figura 4.1 – Dois tipos de problemas, linearmente e não - linearmente separáveis.
Fonte: Haykin (2001).

4.2 Redes Adaline (*Adaptive Linear Element*)

É uma rede com uma única camada, portanto com baixa capacidade computacional, ela é semelhante ao perceptron de uma camada.

Segundo Braga et al. (2007, p. 39) “ambos os modelos são baseados em elementos de processamento que executam operações sobre a soma ponderada de suas entradas”. A rede Adaline resolve somente problemas com características lineares, pois utiliza a função de ativação linear. O modelo Adaline é utilizado para a aproximação de funções, diferentemente do Perceptron que é usado na classificação de padrões.

Ainda o mesmo autor diz que o algoritmo de treinamento desse modelo neural, conhecido como Regra Delta, foi o precursor do algoritmo de treinamento da rede neural Perceptron de múltiplas camadas, cujo nome é *Backpropagation*. A Regra Delta realiza os ajustes dos pesos sinápticos usando a minimização do erro quadrático entre a resposta desejada e a saída da rede neural.

4.3 Perceptron de múltiplas camadas (MLP – *Multi – layer perceptron*)

Nesse trabalho será utilizado o MLP *feedforward*, que conforme Haykin (2001, p. 183) “[...] a rede consiste de um conjunto de unidades sensoriais (nós de fonte) que constituem a camada de entrada, uma ou mais camadas ocultas de nós computacionais e uma camada de saída de nós computacionais”. Nesse tipo de rede os dados de entrada se propagam para frente (*feedforward*) através da rede neural, camada por camada.

Diferentemente do perceptron de uma única camada, o perceptron com múltiplas camadas possui maior capacidade computacional. Como exemplo tem-se o problema do OU Exclusivo (XOR) em que o primeiro não consegue resolver, mas com o segundo é possível encontrar a solução.

Na literatura encontra-se que:

Os perceptrons de múltiplas camadas têm sido aplicados com sucesso para resolver diversos problemas difíceis, através do seu treinamento de forma supervisionada com um algoritmo muito popular conhecido como algoritmo de retropropagação de erro (*error back-propagation*). Este algoritmo é baseado na regra de aprendizagem por correção de erro (HAYKIN, 2001, p. 183).

A Figura 4.2 exibe a arquitetura de um perceptron de múltiplas camadas contendo duas camadas ocultas, nela o fluxo dos dados vão da esquerda para a direita, camada por camada.

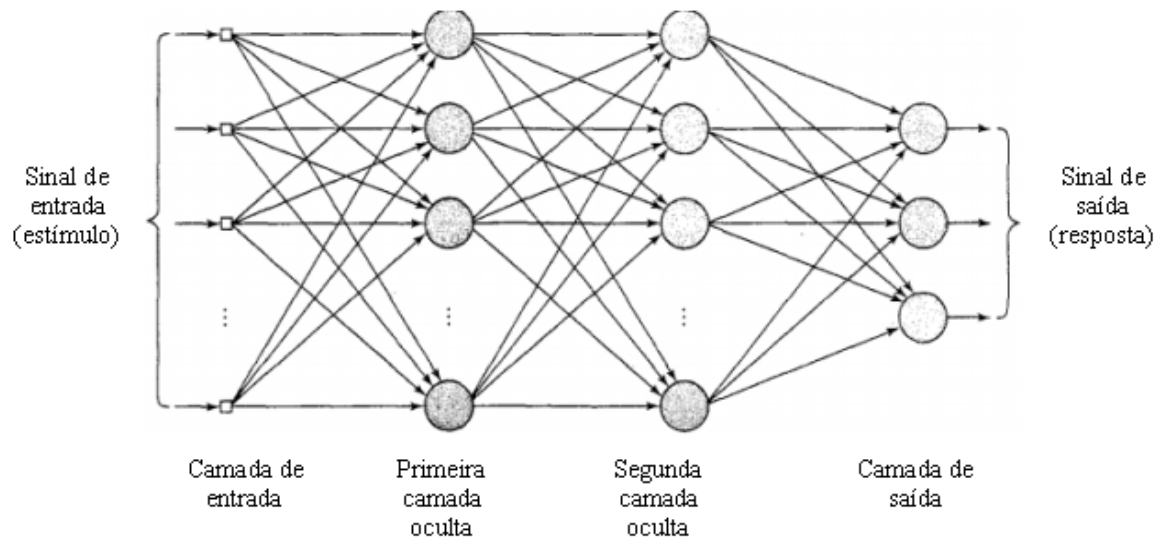


Figura 4.2 - Grafo arquitetural de um perceptron de múltiplas camadas com duas camadas ocultas.
Fonte: Haykin, 2001, p.186

Em relação ao número de camadas tem-se que:

[...] a grande maioria dos problemas práticos raramente necessita de mais de uma camada intermediária. A utilização de duas camadas intermediárias somente é necessária quando a função a ser aproximada apresenta descontinuidades. Assim, a utilização de um número maior de camadas do que o necessário pode levar a problemas de convergência da rede, já que as camadas intermediárias trabalham com estimativas de erro, as quais, quando em maior quantidade, podem dificultar a convergência global da rede (BRAGA et al., 2007, p.70).

Ainda o mesmo autor relata que para a rede neural obter um bom desempenho é necessário definir a quantidade de neurônios para cada camada. Quanto mais neurônios, maior será a complexidade da rede neural. Para determinar a quantidade de neurônios de uma rede, é preciso realizar testes, por exemplo: uma rede com dez neurônios, após o treinamento, obteve 50% de acertos. Já uma rede com 30 neurônios obteve 70% de acertos. Portanto, a rede neural com 30 neurônios conseguiu a melhor taxa de acertos. Em alguns problemas, poucos neurônios em uma rede neural são capazes de solucioná-los.

A seguir será descrito o algoritmo de treinamento utilizado pela rede neural MLP.

4.3.1 Algoritmo de *back-propagation*

O algoritmo mais popular de treinamento da rede MLP é o *back-propagation* que, faz ajustes nos pesos da rede por meio de um mecanismo de correção de erros. O treinamento da rede ocorre em duas fases, cada fase percorre a rede em um sentido (BRAGA et al., 2007).

A Figura 4.3 exibe as duas fases do treinamento da rede pelo algoritmo de *back-propagation*. A primeira é a fase *forward* que é utilizada para determinar a saída da rede de acordo com os dados de entrada. A segunda fase é a *backward* que tem a função de atualizar os pesos das conexões em relação com a saída fornecida e a saída desejada.

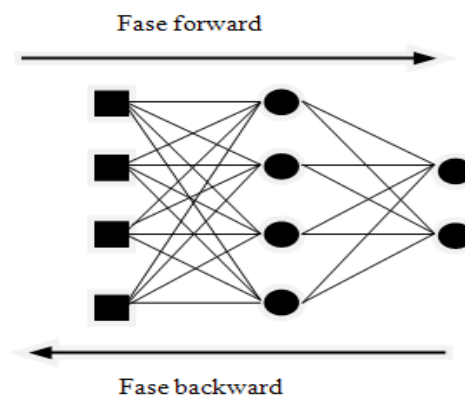


Figura 4.3 - Fluxo de processamento do algoritmo back-propagation.
Fonte: Braga et al. (2007,p.74)

Para Fausett (1993) o treinamento da rede MLP pelo *back-propagation* envolve três estágios: o *feedforward* como entrada padrão de treinamento, o cálculo e *back-propagation* do erro associado, e os ajustes dos pesos.

Já para Haykin (2001) a aprendizagem por *error back-propagation* consiste de dois passos: um passo para frente, *forward*, e um passo para trás, *backward*. No *forward* os pesos da rede permanecem inalterados, eles somente serão atualizados durante o *backward*. O sinal de erro gerado pela diferença entre a saída desejada e a saída real, propaga para trás através da rede, originando assim o nome de *error back-propagation*.

5 Materiais e Métodos

Nesse capítulo está descrito os materiais e métodos utilizados para desenvolver esse trabalho.

5.1 Base de dados

Os dados utilizados para realizar esse trabalho foram retirados do site da BM&FBOVESPA S.A. (Bolsa de Valores, Mercadorias e Futuros), sendo que essa empresa foi criada em 2008 com a fusão entre a Bolsa de Mercadorias & Futuros (BM&F) e a Bolsa de Valores de São Paulo (BOVESPA) (BM&FBOVESPA, 2008).

Para ter acesso gratuito aos arquivos da Série histórica de cotações da BOVESPA, é necessário estar cadastrado no site. Esse cadastro é feito no site da BM&FBOVESPA.

A Figura 5.1 mostra o local do site da BM&FBOVESPA onde foram retirados os dados.



Figura 5.1 – Local do site da BM&FBOVESPA onde foram coletados os dados.

Fonte: Disponível em: <www.bovespa.com.br/Principal.asp>. Acesso em 20 out. 2008

A opção de compra escolhida para se precificar através de uma rede neural artificial foi a da Vale, por suas ações serem uma das mais negociadas na Bolsa de Valores.

Para treinar a rede foram utilizados as variáveis do modelo Black e Scholes como: o preço da ação (Vale), a taxa de juros, o prazo para vencimento da opção, a volatilidade do preço das ações da Vale e o preço de exercício da opção de compra.

- Preço da ação: Foi utilizado o último preço das ações da Vale no mesmo dia que a da opção analisada.
- Taxa de juros: Utilizou – se a taxa de juros cotada no dia em que foi avaliada a opção.
- Prazo para vencimento da opção: Tempo em dias úteis até o vencimento da opção.
- Volatilidade: Foi calculada pelo desvio padrão do valor diário de fechamento das ações da Vale em um período de 60 dias úteis anteriores à primeira cotação utilizada pela rede neural.
- Preço de exercício: Preço de exercício das séries de opções avaliadas.
- Séries históricas dos preços das ações da Vale, no período de outubro de 2006 a agosto de 2007.
- Série histórica da cotação de preço das opções de compra da Vale das seguintes séries: JAN 50-56-60, FEV 56-60-66, MAR 56-62-68, ABR 62-68-72, MAI 68-72-78, JUN 68-72-78, JUL 70-76-80 e AGO 74-78-82.
- Série histórica da taxa de juros ao ano (depósito interbancário - DI).

Nas séries de opções analisadas foram desconsiderados os preços de mercado referente à primeira cotação, devido à existência de dados faltosos. Durante o período de cotação da opção, em alguns casos, a opção não era negociada, ou seja, nesse dia ninguém comprou ou vendeu essa opção. Sendo assim, os dados referentes a esse dia eram eliminados, tais como: o preço da ação, a taxa de juros, a volatilidade da ação, o tempo e o preço de exercício da opção.

A Tabela 5.1 mostra os dados antes de serem excluídos, descritos pelos seguintes itens: data; Vale5 que é o código da ação da Vale e apresenta os preços de fechamento dessa ação; Volatilidade da ação em um período de 60 dias (Volatilidade 60 D); Taxa de juros ao ano (TX Juros (aa)); VALEH74, é o código da opção de compra da Vale com vencimento em agosto de

2007 e preço de exercício igual a R\$ 74,00, nesse item tem-se os preços de fechamento dessa opção; Tempo em dias úteis até o vencimento da opção (Tempo D);

TABELA 5.1 – Conjunto com todos os dados, incluindo os campos vazios.

DATA	VALE5	VOLATILIDADE 60 D	TX JUROS (aa)	VALEH74	TEMPO D
22/5/2007	72,55	4,960703485	12,35	5	62
23/5/2007	71,5	4,990885376	12,35		61
24/5/2007	69,39	5,006481051	12,34	2,9	60
25/5/2007	71,2	4,912366914	12,36	3,5	59
28/5/2007	71,65	4,877255862	12,37	3,78	58
29/5/2007	70,9	4,811705393	12,38		57
30/5/2007	72,61	4,674303752	12,38	3,89	56
31/5/2007	73,43	4,44793614	12,38		55
1/6/2007	75,2	4,378226829	12,39	5	54
4/6/2007	75,51	4,323539419	12,39		53
5/6/2007	75,36	4,278933612	12,4		52
6/6/2007	73,51	4,250722089	12,38	4,42	51
8/6/2007	72,7	4,191263391	11,88	3,64	50

A Tabela 5.2 apresenta os dados organizados, ou seja, sem que haja dados faltosos.

TABELA 5.2 – Conjunto com os dados organizados sem os campos vazios.

DATA	VALE5	VOLATILIDADE 60 D	TX JUROS (aa)	VALEH74	TEMPO D
24/5/2007	69,39	5,006481051	12,34	2,9	60
25/5/2007	71,2	4,912366914	12,36	3,5	59
28/5/2007	71,65	4,877255862	12,37	3,78	58
30/5/2007	72,61	4,674303752	12,38	3,89	56
1/6/2007	75,2	4,378226829	12,39	5	54
6/6/2007	73,51	4,250722089	12,38	4,42	51
8/6/2007	72,7	4,191263391	11,88	3,64	50

5.2 Rede Neural Artificial

A rede neural artificial utilizada nesse trabalho foi a *Multi-Layer Perceptron* (MLP) *feedforward* com cinco dados para a camada de entrada, cinco neurônios na camada intermediária e um neurônio na camada de saída. Essa rede foi escolhida, pois ela permite a solução de problemas mais complexos e também por ser uma rede amplamente utilizada em problemas relacionados com o mercado de capitais. Como exemplo tem-se os trabalhos de Fernandes (2000) e Freitas e Souza (2002) que a utilizaram para precificar opções.

Foram testadas e comparadas diferentes topologias, que se diferenciaram por número de neurônios da camada intermediária, número de ciclos e taxa de aprendizagem.

A função logística foi utilizada no treinamento da rede. Como saída, a rede deve fornecer valores entre zero e um, devido à normalização dos dados que foi entre esse intervalo, sendo esse o motivo da escolha da função logística.

Na Figura 5.2 é apresentada a arquitetura da rede neural escolhida, pois dentre todas as arquiteturas analisadas nesse trabalho, foi a que ofereceu o melhor resultado na validação cruzada.

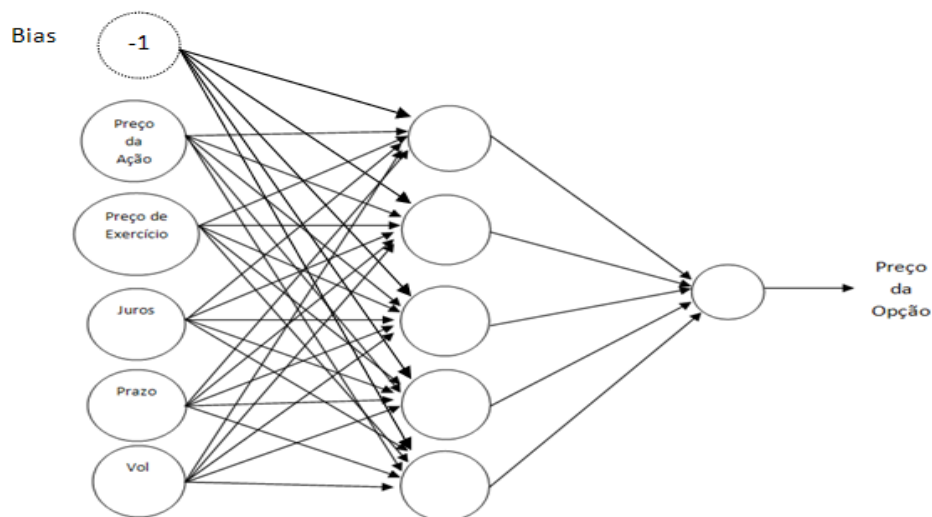


Figura 5.2 – Arquitetura da rede neural utilizada nesse trabalho.

Segundo Tatibana e Kaetsu (2008) “a camada de entrada possui um neurônio especial chamado de bias e serve para aumentar os graus de liberdade, permitindo uma melhor adaptação, por parte da rede neural, ao conhecimento a ela fornecido.” O bias é o limiar de ativação neural, ou seja, é a fronteira entre duas áreas de dados.

5.3 Normalizações dos dados

Como foi utilizada a função de ativação logística, os dados foram normalizados entre zero e um. Para realizar essa normalização, dividi-se o maior valor encontrado em um conjunto de dados por todos os dados desse conjunto. Por exemplo:

- Conjunto de dados {1,5,8,3};
- O maior valor encontrado é o número oito;
- Um dividido por oito é igual a 0,125;
- Cinco dividido por oito é igual a 0,625;
- Oito dividido por oito é igual a 1;
- Três dividido por oito é igual a 0,375;
- Conjunto dos dados normalizados entre zero e um {0,125; 0,625; 1; 0,375};

5.4 Softwares utilizados para treinamento e validação.

Para a realização do treinamento utilizou – se a rede MLP, implementada por Ribeiro et al. (2008).

A Figura 5.3 apresenta a interface simplificada.

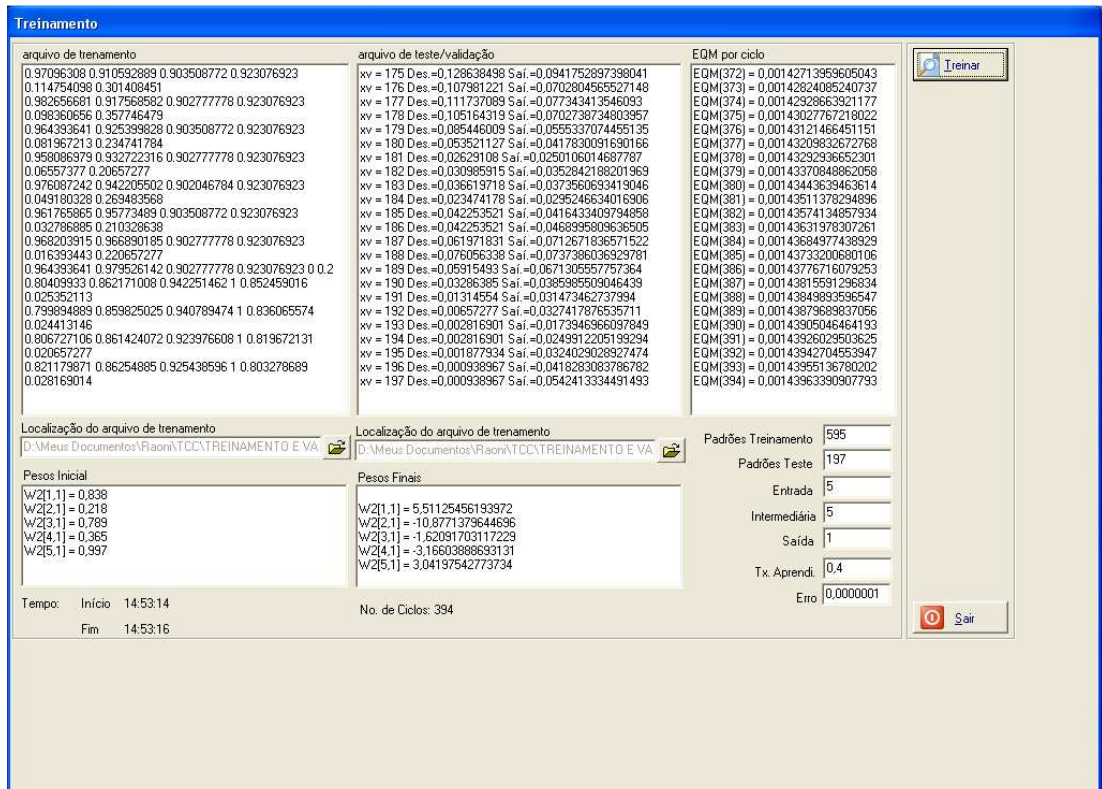


Figura 5.3 – Interface simplificada da rede MLP.
Fonte: Ribeiro et al., 2008.

Esse programa é composto pelas seguintes características:

- Arquivo de treinamento: nesse campo estão os dados de entrada da rede neural e também a saída desejada, que é o preço da opção observada no mercado de opções.
- Arquivo de validação: mostra a saída desejada e a saída da rede neural.
- Erro quadrático médio (EQM) por ciclo: a cada ciclo do treinamento é calculado o EQM.
- Pesos iniciais: são pesos utilizados para o início do treinamento da rede, esses pesos são escolhidos pelo programa aleatoriamente.
- Pesos finais: são pesos que a rede nos fornece no fim do treinamento, esse peso é utilizado para realizar a validação da rede e também para se precificar a

opção, ou seja, o objetivo do treinamento da rede é encontrar o peso final adequado, que nesse trabalho serviu para precificar opções.

- Padrões de treinamento e teste: indica ao programa o número de padrões de entrada que será utilizado pela camada de entrada. Na Figura 5.3 temos como exemplo 595 padrões de entrada para o treinamento e 197 padrões de entrada para o teste.
- Campos entrada, intermediária e saída: fornecem ao programa a arquitetura da rede neural que será utilizada no treinamento.
- Taxa de aprendizagem: fornece ao programa o valor da taxa escolhida para realizar o treinamento.
- Erro: indica ao programa a taxa de erro que se espera que a rede alcance no final do treinamento.
- Tempo: mostra-nos a hora de início e fim do treinamento.
- Número de ciclo: informa-nos quantos ciclos foram utilizados pelo treinamento.

Para realizar a validação das redes neurais artificiais utilizadas nesse trabalho, foi implementado um programa, utilizando a linguagem de programação Java. O algoritmo está descrito abaixo:

1. Ler o arquivo de validação e peso final.
2. Realizar o somatório do resultado da multiplicação dos dados de entrada com os seus respectivos pesos, para cada neurônio da camada intermediária.
3. Calcular a função logística para cada neurônio da camada intermediária.
4. Armazenar os resultados.
5. Realizar o somatório do resultado da multiplicação dos dados dos neurônios da camada intermediária com os seus respectivos pesos, para o único neurônio da camada de saída.
6. Calcular a função logística para o neurônio da camada saída.
7. Armazenar os resultados.

8. Multiplicar as respostas desejadas pelo número utilizado na normalização desse conjunto de dados e depois armazenar os resultados.
9. Multiplicar as respostas da rede (resultados da função logística do neurônio da camada de saída) pelo número utilizado na normalização do conjunto de dados da resposta desejada e depois armazenar os resultados.
10. Dividir os resultados do item nove pelos do item dez e armazenar as respostas.
11. Armazenar um se a resposta do item dez for maior ou igual a 0,85 e menor ou igual a 1,15, caso contrário armazenar o valor zero.
12. Gravar os dados armazenados dos itens oito, nove, dez, onze e doze em um arquivo texto.

O código fonte comentado está disponível no apêndice A.

A Figura 5.4 apresenta os cálculos que são feitos durante a validação da rede para um padrão de entrada. Nele o somatório é feito pela soma do resultado da multiplicação entre o dado de entrada e seu respectivo peso. Após isso, é calculada a função logística com o resultado do somatório. O resultado da função logística é valor do neurônio da camada intermediária. Isso é feito para todos os neurônios da camada intermediária. Em seguida, é feito o somatório dos valores dos neurônios da camada intermediária com seus pesos. Por fim, aplica-se a função logística no resultado da somatória para encontrar a resposta da rede, que nesse trabalho é a precificação da opção.

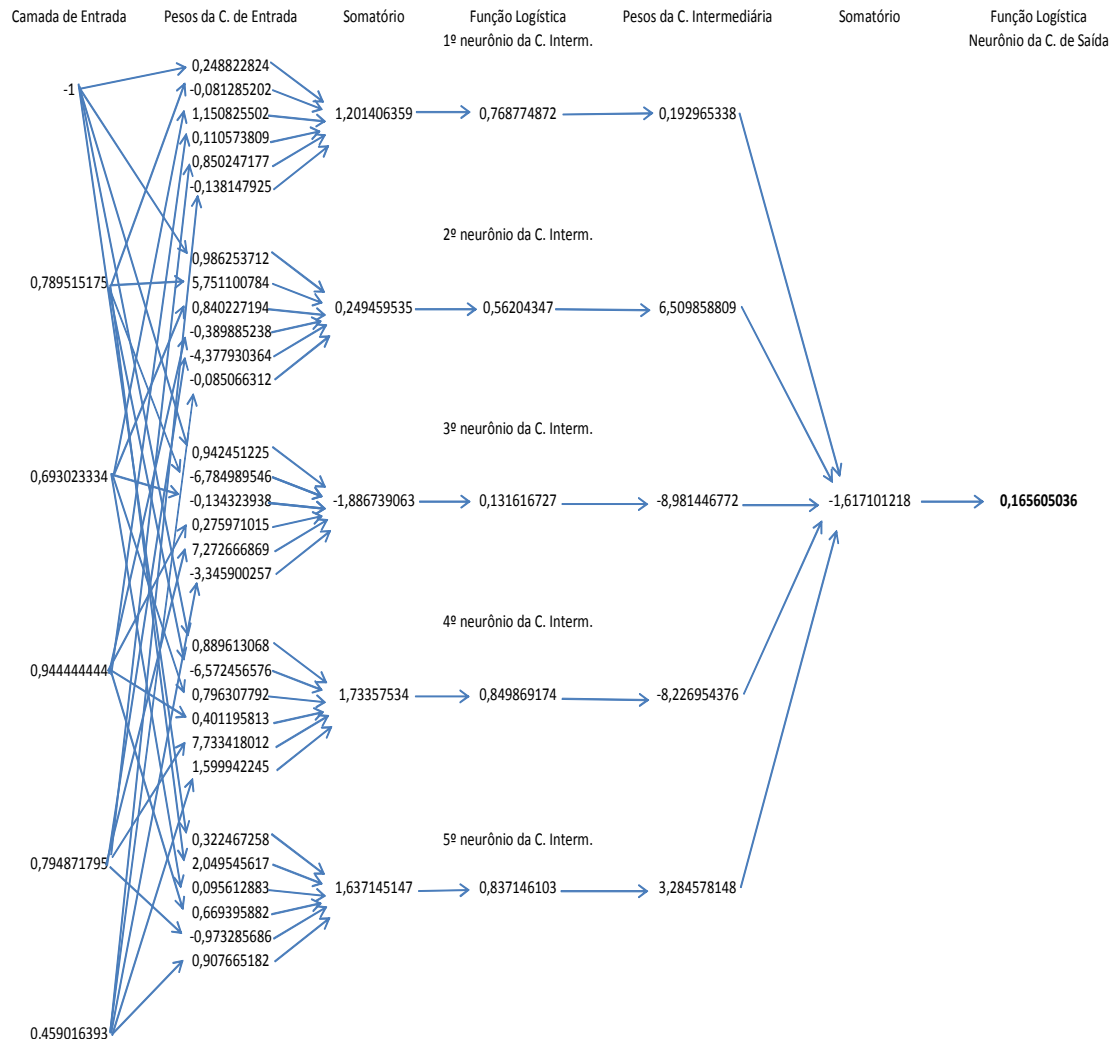


Figura 5.4 – Estrutura dos cálculos realizados por uma rede MLP.

Os dados da Figura 5.4 estão normalizados, exceto os pesos, sendo necessário realizar a desnormalização para obter o preço da opção fornecido pela rede neural. Nesse trabalho a desnormalização foi feita multiplicando por 10.65 o valor do neurônio da camada de saída, esse número (10.65) foi utilizado para normalizar os preços das opções de compra da Vale observados no mercado de opções, por esse motivo o mesmo foi usado na desnormalização.

Foi utilizado o Excel do pacote Microsoft Office para organizar os dados da validação e das topologias das redes utilizadas no treinamento. É importante destacar que foi feito a somatória dos dados do item 11 do algoritmo acima e com isso soube – se a porcentagem de acertos de cada rede.

Com essa ferramenta foi possível selecionar as cotações das ações da Vale e das opções de compra utilizadas nesse trabalho, pois esses dados vieram misturados com as cotações de ações de outras empresas, quando extraídas do site da BM&FBOVESPA.

Os gráficos de comparação entre os preços reais de mercado e os da rede neural também foram feitos com o auxílio do Excel.

5.5 Validação cruzada

Para Haykin (2001, p.240), “o uso da validação cruzada é atrativo quando temos que projetar uma rede neural grande cujo objetivo seja uma boa generalização”. O mesmo autor ainda diz que desejamos que a rede neural fique bem-treinada para aprender o máximo possível sobre o passado e com isso generalizar no futuro.

Com a validação cruzada é possível identificar o “melhor” modelo de rede neural.

Para realizar a validação cruzada seguem-se os seguintes passos:

1. Do conjunto de dados selecionados, pegou – se aleatoriamente 75% para realizar o treinamento e 25% para a validação da rede.
2. O passo anterior foi utilizado para criar dez arquivos de treinamento e dez arquivos de validação.

Os dez arquivos de treinamento foram utilizados para treinar as topologias de rede candidatas. Por exemplo: são feitos com uma topologia de rede dez treinamentos, um treinamento para cada arquivo com os 75% dos dados.

Os dez arquivos de validação foram utilizados pelo programa em Java para obtermos a porcentagem de acerto de cada treinamento.

A figura 5.5 mostra os dez arquivos de treinamento e validação.

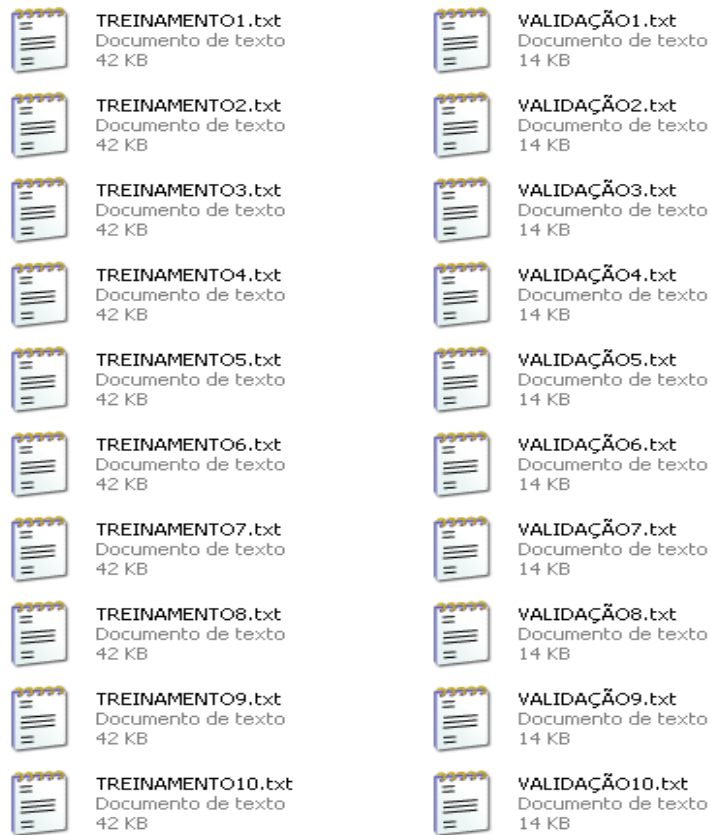


Figura 5.5 – Dez arquivos de treinamento e validação.

Os dados do treinamento 1 são diferentes dos dados da validação 1, os dados do treinamento 2 são diferentes dos dados da validação 2 e assim por diante.

6 Resultados

Foram feitas diversas análises para encontrar a topologia de rede neural *Multi Layer Perceptron* (MLP) mais adequada para se precificar opções. Em todas as topologias, a taxa de erro que indica o mínimo aceitável de erro que a rede pode oferecer no final do treinamento, foi fixada em 0.0000001, pois, foi verificado em testes preliminares que erros com taxas maiores não apresentavam bons resultado. O momentum, que tem o objetivo de controlar a velocidade de treinamento da rede, foi fixado em 0.1.

A Tabela 6.1 mostra seis topologias utilizadas nesse trabalho, cujos itens estão descritos abaixo:

- Entrada: mostra a quantidade de variáveis de entrada que a camada de entrada da rede terá.
- Camada Intermediária: é a quantidade de neurônios que a camada intermediária da rede terá.
- Saída: nela temos o número de neurônios da camada de saída.
- Taxa de Aprendizagem: é a taxa de aprendizado utilizada para treinar a rede.
- Acertos: taxa de acertos de cada topologia.

Nela é possível verificar que a melhor topologia de rede neural foi a que obteve 60% de acerto. Dentro dessa topologia foram feitos dez treinamentos e validações.

TABELA 6.1 – Topologias de redes neurais utilizadas nesse trabalho.

ENTRADA	C. INTERM.	SAÍDA	TX. APREND.	ERRO	MOMENTUM	ACERTOS
5	3	1	0,5	0,0000001	0.1	59,50%
5	8	1	0,8	0,0000001	0.1	43,33%
5	5	1	0,2	0,0000001	0.1	45,66%
5	5	1	0,9	0,0000001	0.1	48,18%
5	5	1	0,4	0,0000001	0.1	60,00%
5	5	1	0,1	0,0000001	0.1	58,79%

A Tabela 6.2 detalha a topologia com a melhor taxa de acerto.

TABELA 6.2 – Detalhes da topologia com a melhor taxa de acerto.

ENTRADA: 5; C. INTERM.: 5; SAÍDA: 1; TX. APREND.: 0,4						
Nº CICLOS	INÍCIO	FIM	ARQ. TREIN.	ARQ. VALID.	ACERTOS 0,85/1,15	Total
231	00:30:15	00:30:16	1	1	57,07%	60,00%
1190	01:07:06	01:07:10	2	2	50,51%	
931	17:25:42	17:25:45	3	3	57,07%	
987	17:27:54	17:27:58	4	4	60,61%	
980	02:32:30	02:32:33	5	5	66,67%	
1338	17:32:06	17:32:11	6	6	63,64%	
1011	17:33:58	17:34:01	7	7	60,10%	
1879	17:36:33	17:36:40	8	8	60,10%	
1713	17:40:10	17:40:16	9	9	61,11%	
1597	17:43:31	17:43:36	10	10	63,13%	

Nela tem-se:

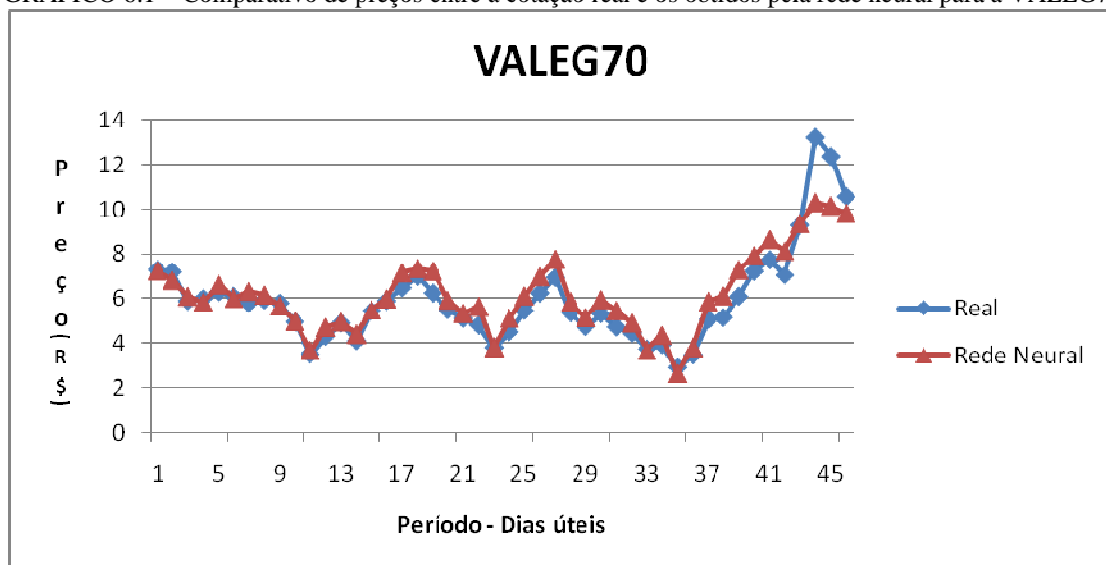
- Número de ciclos: é a quantidade de ciclos que foram utilizados para treinar a rede. Esse valor varia a cada novo treinamento.
- Início: mostra a hora de início do treinamento da rede neural.
- Fim: indica a hora final do treinamento.
- Arquivo de Treinamento: mostra qual arquivo de treinamento está sendo usado para treinar a rede.
- Arquivo de Validação: exhibe qual arquivo de validação está sendo usado para validar a rede.
- Acertos 0,85/1,15: apresenta a porcentagem de acerto de determinado treinamento, em uma faixa entre 0,85 e 1,15, sendo que o valor 1 indica que a saída da rede é idêntica à resposta desejada. Essa faixa de valor é referente ao resultado da divisão entre a resposta desejada (preço real da opção) e a resposta da rede neural.
- Total: foi feita a somatória das porcentagens de acertos para cada topologia e depois dividida por dez. Isso foi feito para obter a taxa geral de acerto da topologia analisada.

Nota-se que existe um treinamento com 66,67% de acertos, sendo a melhor taxa de acerto dos dez treinamentos e validações, portanto esse treinamento foi o escolhido para se precificar as opções de compra da Vale e criar os gráficos comparativos entre o preço real (preço observado no mercado de opções) da opção e o preço fornecido pela rede neural.

Para fazer a precificação utilizou – se os pesos finais da topologia com 66,67%. Esses pesos junto com os dados de entrada foram inseridos no programa em Java descrito no capítulo sobre materiais e métodos.

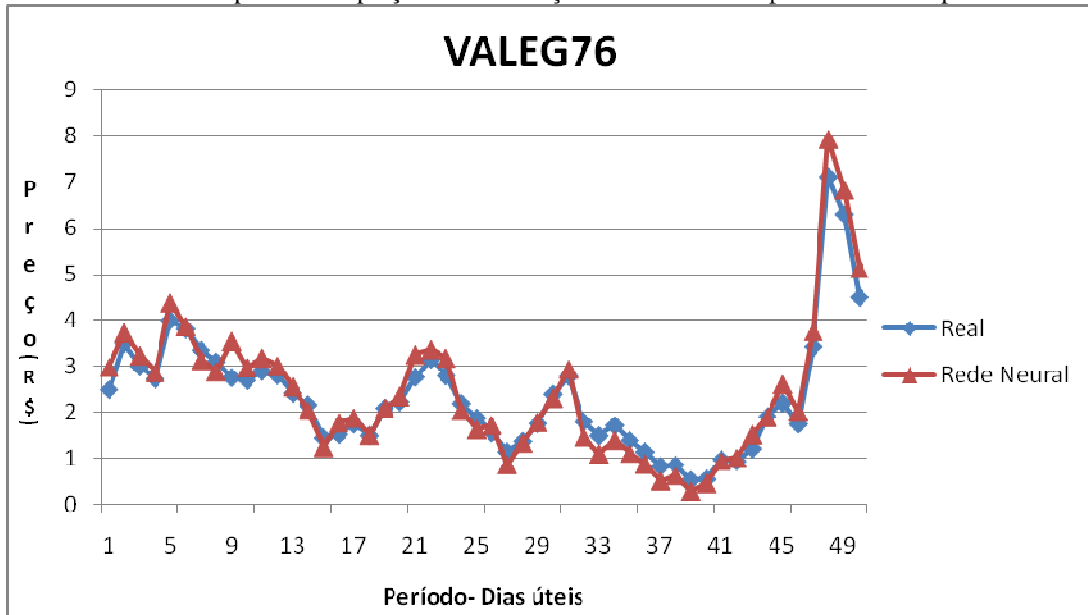
O Gráfico 6.1 mostra o comparativo entre a cotação real da VALEG70 e o preço fornecido pela rede neural. VALEG70, são as opções de compra da Vale com preço de exercício igual a R\$ 70,00 e com vencimento no mês de julho de 2007. Nele verifica-se que os preços obtidos pela rede neural se igualam quase que totalmente ao preço real. Exceto no período entre 43 e 46 dias, que o preço real é maior que o da rede neural, mas no último dia eles ficam bem próximos.

GRÁFICO 6.1 – Comparativo de preços entre a cotação real e os obtidos pela rede neural para a VALEG70.



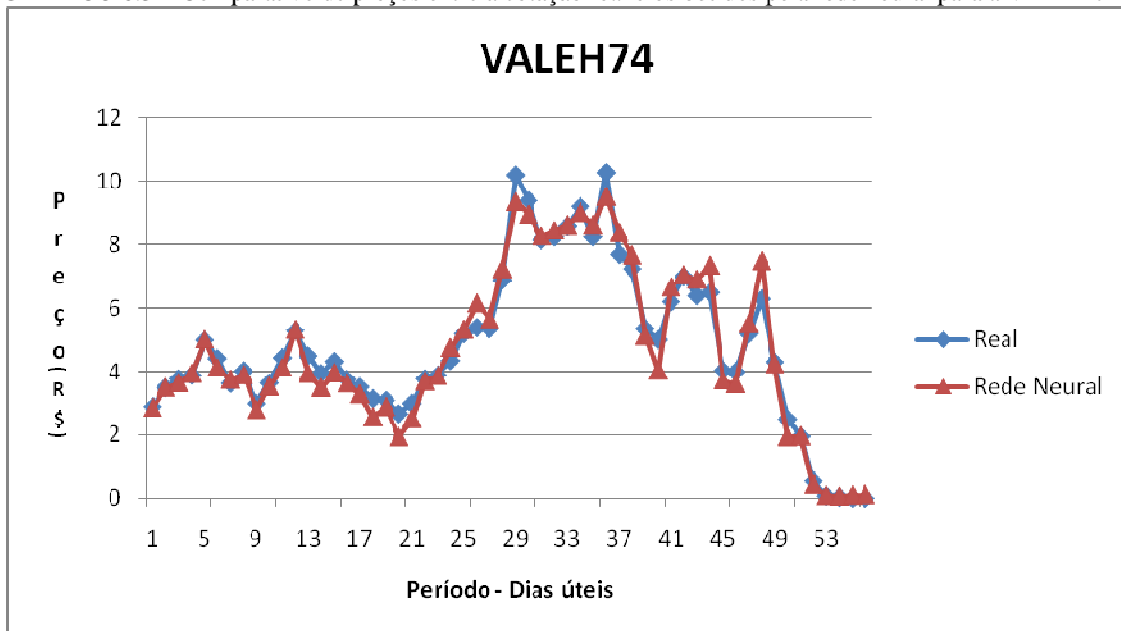
No Gráfico 6.2 nota-se a grande semelhança entre os preços de mercado e os da rede neural. Nesse gráfico foram analisadas as opções de compra da série com vencimento no mês de julho de 2007 e preço de exercício igual a R\$ 76,00. Observa – se também que a rede neural conseguiu identificar a tendência da opção avaliada, se é de alta ou de baixa.

GRÁFICO 6.2 – Comparativo de preços entre a cotação real e os obtidos pela rede neural para a VALEG76.



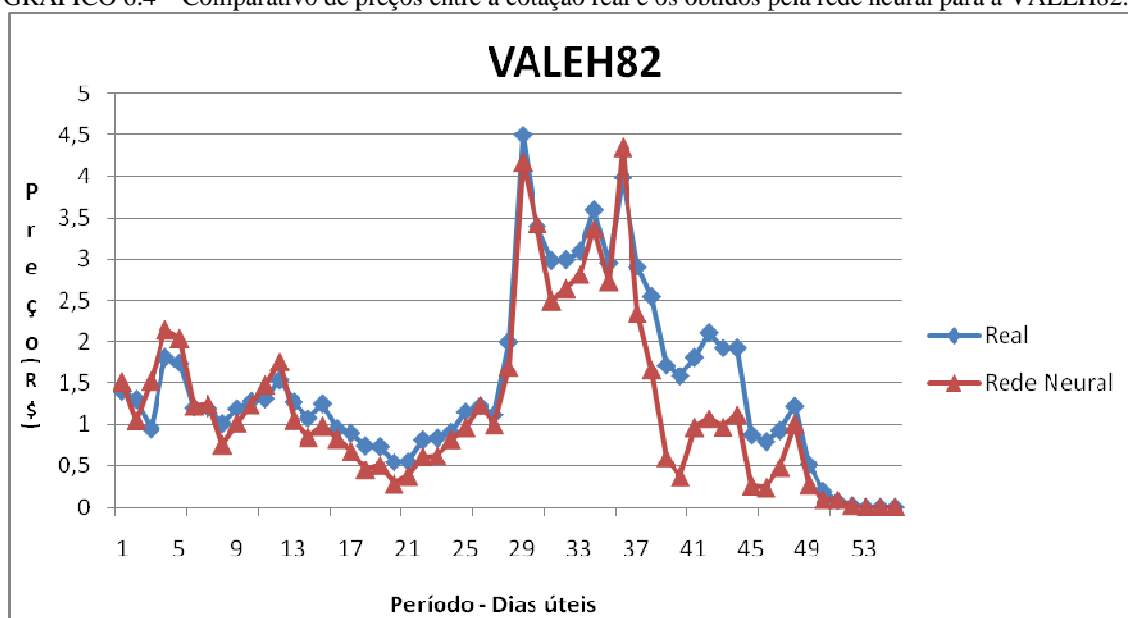
A série com vencimento em agosto de 2007 e preço de exercício igual à R\$74,00 foi estudado no gráfico 1.4. Nele é possível verificar a proximidade do preço real com o da rede neural em praticamente todo o período da série.

GRÁFICO 6.3 – Comparativo de preços entre a cotação real e os obtidos pela rede neural para a VALEH74.



O Gráfico 6.4 mostra os preços de mercado da série com vencimento em agosto de 2007 e preço de exercício igual a R\$ 82,00. O preço disponibilizado pela rede neural, nesse gráfico, foi o que mesmo se aproximou do preço real de mercado, em comparação com os outros avaliados anteriormente. Sendo que após o dia 29 houve grande disparidade entre os preços, acentuado entre os dias 37 e 49, após o dia 39 observa-se novamente a semelhança entre os preços. Mesmo assim é possível identificar através dos preços da rede neural a tendência da opção, ou seja, se o preço da opção está caindo ou subindo.

GRÁFICO 6.4 – Comparativo de preços entre a cotação real e os obtidos pela rede neural para a VALEH82.



Diante dos gráficos estudados, é visível que o acerto da rede neural foi maior, devido à grande similaridade entre os preços de mercado e o da rede neural. Sendo possível aumentarmos a faixa de acerto que está entre 0,85 e 1,15 para um intervalo de 0,75 e 1,25.

A Tabela 6.3 mostra as novas taxas de acerto da melhor topologia, a com 60% de acerto.

TABELA 6.3 – Novas taxas de acertos da melhor topologia.

ENTRADA: 5; C. INTERM.: 5; SAÍDA: 1; TX. APREND.: 0,4								
Nº CICLOS	INÍCIO	FIM	ARQ. TREIN.	ARQ. VALID.	ACERTOS 0,85/1,15	Total	ACERTOS 0,75/1,25	Total
231	00:30:15	00:30:16	1	1	57,07%	60,00%	70,20%	75,40%
1190	01:07:06	01:07:10	2	2	50,51%		71,21%	
931	17:25:42	17:25:45	3	3	57,07%		74,75%	
987	17:27:54	17:27:58	4	4	60,61%		78,28%	
980	02:32:30	02:32:33	5	5	66,67%		80,81%	
1338	17:32:06	17:32:11	6	6	63,64%		74,24%	
1011	17:33:58	17:34:01	7	7	60,10%		72,73%	
1879	17:36:33	17:36:40	8	8	60,10%		77,27%	
1713	17:40:10	17:40:16	9	9	61,11%		77,27%	
1597	17:43:31	17:43:36	10	10	63,13%		77,27%	

Na tabela acima encontramos uma taxa de acerto total igual a 75,40% para o intervalo entre 0,75 e 1,25. Com 15,40% a mais que as taxas cujo valor estava limitado no intervalo entre 0,85 e 1,15. Sendo que com a antiga limitação tínhamos 66,67% para o melhor treinamento e agora, temos 80,81% de acerto. Houve uma melhora significativa nas taxas de acertos após o aumento da limitação.

O mesmo foi verificado nas outras topologias.

A tabela 6.4 mostra as novas taxas de acertos para as demais topologias.

TABELA 6.4 – Novas taxas de acertos para as outras topologias.

ENTRADA	C. INTERM.	SAÍDA	TX. APREND.	ERRO	MOMENTUM	ACERTOS 0,85/1,15	ACERTOS 0,75/1,25
5	3	1	0,5	0,0000001	0.1	59,50%	75,20%
5	8	1	0,8	0,0000001	0.1	43,33%	57,48%
5	5	1	0,2	0,0000001	0.1	45,66%	62,53%
5	5	1	0,9	0,0000001	0.1	48,18%	64,19%
5	5	1	0,4	0,0000001	0.1	60,00%	75,40%
5	5	1	0,1	0,0000001	0.1	58,79%	73,23%

Em todas as topologias da tabela 2.1 houve aumento da porcentagem de acerto. A primeira topologia merece destaque, devido ao acerto (75,20%) estar bem próximo da “melhor” topologia (75,4) e também por possuir somente três neurônios na camada intermediária. A terceira topologia foi a que teve uma maior variação na taxa de acerto, passando de 45,66% para 62,53% variando 16,87%.

Fernandes (2000) utilizou em seu trabalho uma rede neural MLP com cinco atributos de entrada, duas camadas intermediárias com cinco neurônios em cada camada e um neurônio na camada de saída.

Em outro trabalho, os autores Freitas e Souza (2002) utilizaram vários modelos de redes neurais MLP para precificar opções, sendo a que obteve melhor resultado é a com quatro neurônios na camada intermediária e um neurônio na camada de saída.

Diferentemente das topologias utilizadas por FERNADES 2000 e FREITAS e SOUZA 2002, nesse trabalho utilizou - se cinco variáveis de entrada, cinco neurônios na camada intermediária e um neurônio na camada de saída na topologia com melhor resultado de acertos para se precificar as opções, obtendo uma boa taxa de acerto.

7 Conclusão

Este trabalho apresentou uma comparação entre o preço da opção de compra da Vale observado no mercado e o obtido pela rede neural.

A partir dos resultados ficou constatado que a precificação de opções através da rede neural *Multi Layer Perceptron* (MLP) é viável, pois diante dos gráficos observados nos resultados, ficou visível a semelhança entre a cotação obtida pela rede neural e a cotação real. Demonstrando que a rede neural conseguiu identificar se a tendência é de alta ou de baixa das cotações das opções, sendo uma boa ferramenta para identificar possíveis compras ou vendas. Essa ferramenta deve ser usada em conjunto com outras formas de análise, para que o investidor tenha uma maior segurança em decidir comprar ou vender as opções de compra da Vale. Através dos gráficos é possível um investidor fazer uma análise técnica, cujo conceito está descrito abaixo:

A análise técnica enfatiza que o mais importante é entender o comportamento do mercado, registrado no gráfico de preços, que, por sua vez, é consequência de todas as informações realmente relevantes, não importando quais são essas. Portanto, para a Análise Técnica, não há necessidade de se avaliar outras informações, além do gráfico de preços (MATSURA, 2006, p. XVI).

É importante destacar que dentre as topologias analisadas a que obteve o segundo melhor resultado foi a com três neurônios na camada intermediária, a diferença entre essa topologia e a melhor foi mínima, mostrando que é possível conseguir bons resultados com poucos neurônios.

Novos estudos poderiam utilizar outros modelos de redes neurais para se precificar opções e compará-los com os resultados da rede MLP. É possível também acrescentar novos atributos de entrada como, por exemplo: as cotações do índice Bovespa, as cotações do índice S&P500 (bolsa de valores dos EUA), volume de negociações, etc. Outra sugestão é aumentar a quantidade de padrões de entrada para um período maior que um ano e também precificar as opções de outras empresas que compõem o índice Bovespa. Diante disso, há um amplo potencial de estudo nessa área e que deve ser explorada.

Portanto, com os resultados obtidos, a rede neural mostrou-se uma boa alternativa para se precificar as opções de compra da Vale. É esperado que os profissionais do mercado financeiro aumentem o interesse em realizar novas pesquisas com as redes neurais artificiais, assim surgirão modelos de rede cada vez mais aperfeiçoados. Auxiliando-os nas tomadas de decisões de compra ou venda de opções.

Referências bibliográficas

- BISHOP, C. M. *Neural Networks for Pattern Recognition*. New York: Oxford University Press Inc., 1995.
- BM&FBOVESPA a nova bolsa. Apresenta informações sobre os mercados da Bolsa de Valores de São Paulo. Disponível em: < <http://www.bovespa.com.br/Principal.asp> >. Acesso em: 20 nov. 2008.
- BLACK, F.; SCHOLES, M. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* 81, pp. 637–659, May–June 1973.
- BOVESPA. *Como atuar no mercado de opções*. Folheto explicativo, 2000. Disponível em: <<http://www.bovespa.com.br>> Acesso em: 10 jun. 2008.
- BRAGA, A. de P.; CARVALHO, A. C. P. de L. F.; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações*. 2. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2007.
- BURGESS, A. N.; REFENES, N. Modelling Non-Linear Moving Average Processes Using Neural Networks with Error Feedback: An Application to Implied Volatility Forecasting. *Signal Processing*, v.74, pp. 89–99, 1999.
- FAUSETT, L. *Fundamentals of neural networks: Architectures, Algorithms And Applications*. USA: Prentice Hall, 1993.
- FERNANDES, M. A. *Precificação e hedge dinâmico de opções de Telebrás utilizando redes neurais*. 2000. 86 f. Monografia (Pós-Graduação) – Escola de Administração, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- FREITAS, S. O.; SOUZA, A. A. Utilização de redes neurais na precificação de opções. *Resenha BM&F*, n. 150, p. 63-73, jul/ago. 2002. Disponível em: <<http://www.bmf.com.br>>. Acesso em: 08 mai. 2008.
- GARCIA, R.; GENÇAY, R. Pricing and Hedging Derivative Securities with Neural Networks and Homogeneity Hint. *Journal of Econometrics*, v. 94, pp. 93–115, 2000.
- GRADOJEVIC, N.; GENÇAY, R.; KUKOLJ, D. *Option Pricing with Modular Neural Networks*. 2007. Disponível em: < http://papers.ssrn.com/sol3/papers.cfm?abstract_id=959544>. Acesso em: 20 nov. 2008.
- HAYKIN, S. *Redes Neurais: princípios e prática*. Tradução de Paulo Martins Engel. 2 ed. Porto Alegre: Bookman, 2001.

HEBB, D. O. *The Organization of Behavior: A neuropsychological theory*. New York: Wiley, 1949.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective properties. *Proceedings of the National Academy of Sciences, USA*, vol. 79, p. 2554-2558, 1982.

HULL, J. C. *Options, Futures and Other Derivatives*. Upper Saddle River. Prentice Hall, Inc., 1997.

HUTCHINSON, J. M.; LO, A.W.; POGGIO, T. A Nonparametric Approach To Pricing and Hedging Derivative Securities Via Learning Networks. *Journal of Finance*, vol. 49, n. 3, pp. 851–889, July 1994.

ISSA, M. *Investindo em opções: como aumentar seu capital operando com segurança*. 2. ed. Rio de Janeiro: Elsevier, 2007.

LAJBCYGIER, P.; BOEK, C.; PALANISWAMI, M.; FLITMAN, A. Neural Network Pricing of All Ordinaries SPI Options on Futures. In: REFENES et al. *Neural Networks in Financial Engineering. Proceedings of 3rd International Conference On neural Networks in the Capital Markets* (Oct 1995), pp. 64–77. London: World Scientific, 1996.

MALLIARIS, M.; SALCHENBERGER, L. Using Neural Networks to Forecast The S&P 100 Implied Volatility. *Neurocomputing*, v. 10, pp. 183–195, 1996.

MATSURA, E. *Comprar ou vender? Como investir na Bolsa utilizando análise gráfica*. 4 ed. São Paulo: Saraiva, 2006.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115-133, 1943.

ORMONEIT, D. A Regularization Approach to Continuous Learning with an Application to Financial Derivatives Pricing. *Neural Networks*, v. 12, pp. 1405–1412, 1999.

PORTAL do investidor. Apresenta conceitos sobre a bolsa de valores. Disponível em: <<http://www.portaldoinvestidor.gov.br>>. Acesso em: 14 jun. 2008.

QI, M.; MADDALA, G.S. Option Pricing Using Artificial Neural Networks: The Case of S&P 500 Index Call Options. In: REFENES et al. *Neural Networks in Financial Engineering, Proceedings of 3rd International Conference On neural Networks in the Capital Markets* (Oct 1995), pp. 78–91. London: World Scientific, 1996.

RIBEIRO, P. B.; SCHIABEL, Homero, PATROCINIO, A. C., ROMERO, R. A. F. *Comparativo no Desempenho de duas RNAs com tipos diferentes de Aprendizagem para a Classificação de Nódulos Mamários por Contorno* In: Congresso Brasileiro de Engenharia Biomédica CBEB, 2008, Salvador.

ROSENBLATT, F. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, vol. 65, p. 386-408, 1958.

RUMELHART, D. E.; MACCLELLAND, J. L. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*. Cambridge, MA: MIT Press, 1986.

TATIBANA, C. Y.; KAETSU, D. Y. Homepage de Redes Neurais. Disponível em: <<http://www.din.uem.br/ia/neurais>>. Acesso em: 15 nov. 2008.

YAO, J.; LI, Y.; TAN, C. L. Option Price Forecasting Using Neural Networks. *Omega*, v. 28, pp. 455–466, 2000.

Bibliografia consultada

DEITEL, H. M.; DEITEL, P. J. *Java: como programar*. Tradução de Edson Furmankiewicz. Revisão técnica Fábio Lucchini. 6 ed. São Paulo: Pearson Pentice Hall, 2005.

Apêndice A – Código fonte comentado do programa implementado em Java para realizar a validação.

```
//O import é utilizado para importar as bibliotecas que serão usadas pelo programa.
import java.util.Scanner;
import java.io.*;
import javax.swing.JOptionPane;

public class Validar {
    public static void main(String[] args) {
        // Vetor onde serão armazenados os dados do arquivo de validação.
        String[] vetVal= new String[1188];
        // Matriz que será usada para armazenar os dados da validação.
        String [][] matVal = new String[198][7];
        // Matriz que conterà os dados de saída.
        String [][] matSaida = new String[198][6];
        // Variável que servirá para realizar os cálculos da função logística.
        double dFunclog = 0;

        int a=0, x =0, camInt = 0, iNumpesos = 0; //variáveis de apoio.
        // Variável do tipo string que serve para armazenar o número
        //de neurônios da camada intermediária.
        String sInterm;

        try { //Serve para tratar exceção.

            for (int b = 1; b<=6;b++) // Realiza um looping de seis voltas.
            {
                //Mostrará uma caixa de diálogo, onde o usuário deverá digitar um valor.
                sInterm = JOptionPane.showInputDialog("Digite o número de neurônios da camada intermediária");
                //Converte para integer a string capturada pela caixa de diálogo.
                camInt = Integer.parseInt(sInterm);
                // Calcula a quantidade de pesos da rede de acordo com o número
                //de neurônios da camada intermediária. O número seis é por
                //existir seis atributos de entrada.
                iNumpesos = 6 * camInt + camInt;
                // Cria um vetor para armazenar os pesos.
                String[] vetPeso= new String[iNumpesos];
                //Cria uma matriz para guardar os dados referentes aos
                //neurônios da camada intermediária.
                String [][] matCamInter = new String[198][camInt];

                for (int c = 1; c<=10;c++) // Realiza um loop de 10 voltas.
                {

                    //Especifica o arquivo que será manipulado, no caso um arquivo de validação.
                    File fileval = new File("c:\\Validacao\\VALIDAÇÃO"+c+".txt");
                    //Destina-se a leitura do arquivo.
                    FileInputStream in0 = new FileInputStream(fileval);
                    //É utilizado para percorrer o arquivo.
                    Scanner scannerValidacao = new Scanner(in0);
                    //Especifica um arquivo que será manipulado, no caso um arquivo
                    //com os pesos da rede neural.
                    File filepeso = new File("c:\\Dados\\PESO"+b+". "+c+".txt");
                    //Destina-se a leitura do arquivo.
                    FileInputStream in1 = new FileInputStream(filepeso);
                    //É utilizado para percorrer o arquivo, nesse caso do arquivo Peso.
                    Scanner scannerPeso1 = new Scanner(in1);
                }
            }
        }
    }
}
```

```

//Realiza um loop até que não haja mais dados no arquivo de validação.
while (scannerValidacao.hasNext()) {
    //Armazena um conjunto de dados que estejam separados por espaços em branco.
    String readLine = scannerValidacao.next();
    //Armazena os dados do arquivo de validação no vetor.
    vetVal[a]= readLine;
    // É usada para indicar a posição em que será armazenado o dado no vetor.
    a++;
}
a = 0;
//Realiza um loop até que não haja mais dados no arquivo Peso.
while (scannerPeso1.hasNext()) {
    //Armazena um conjunto de dados que estejam separados por espaços em branco.
    String readLine = scannerPeso1.next();
    //Armazena os dados do arquivo Peso no vetor.
    vetPeso[a]= readLine;
    // É usada para indicar a posição em que será armazenado o dado no vetor.
    a++;
}
a=0;
x=0;
// Esse for serve para passar os valores do vetor de validação para a matriz de validação.
// Ele realiza um loop de 197 voltas, isso porque o arquivo de
//validação possui 197 padrões de entrada.
for (int i =0;i<198; i++)
{
    for (int j = 0; j<7;j++)
    {
        // Acrescenta -1, que é o valor do bias, na primeira coluna.
        if (j == 0) {matVal[i][j] = "-1"; }
        else
        {
            //Armazena os valores do vetor validação para a matriz validação.
            matVal[i][j] = vetVal[x];
            x++;
        }
        // Insere na primeira coluna da matriz de saída as
        //respostas desejadas da rede neural.
        if (j == 5) matSaida[i][0] = vetVal[x];
    }
}

int z = 0;
// Abaixo serão feitos os cálculos para obter os dados da camada intermediária.
for (int p = 0; p< 198;p++)
{
    // O número de neurônios da camada intermediária corresponde
    // ao número de voltas desse for.
    for (int i =0;i<camInt; i++)
    {
        // A variável da função logística recebe o valor zero.
        dFunclog = 0;
        z = i;
    }
}

```

```

for (int j = 0; j<6;j++)// Realiza um loop de 6 voltas.
{
    //Soma o valor da variável da função logística com o resultado da
    //multiplicação do peso com o valor de entrada.
    dFunclog += Double.valueOf(vetPeso[z]) * Double.valueOf(matVal[p][j]);
    z +=camInt;// Auxiliar de posição do vetor Peso.
}
// Realiza o cálculo da função logística.
dFunclog = 1/(1+ Math.exp(-1*dFunclog));
//Armazena o valor da variável da função logística na
//matriz camada intermediária.
matCamInter[p][i] = String.valueOf(dFunclog);
}
}
// Na seqüência serão realizados os cálculos para obter a resposta da rede neural.
for (int p = 0; p< 198;p++)
{
    dFunclog = 0;
    z = camInt * 6;
    for (int i = 0; i<camInt;i++)
    {
        //Soma o valor da variável da função logística com o resultado da
        //multiplicação do peso com o valor do neurônio da camada intermediária.
        dFunclog += Double.valueOf(vetPeso[z]) * Double.valueOf(matCamInter[p][i]) ;
        z++;// Auxiliar de posição do vetor Peso.
    }
    // Realiza o cálculo da função logística que nesse
    //caso é a resposta da rede, ou seja, o preço da opção obtida pela rede neural.
    dFunclog = 1/(1+ Math.exp(-1*dFunclog));
    // Armazena o valor da variável da função logística na matriz de saída.
    matSaida[p][2] = String.valueOf(dFunclog);
}
}
for (int d = 0 ;d<198;d++)
{
    //Armazena na coluna 1 da matriz saída o resultado da multiplicação da
    //resposta desejada, que está alocada na coluna 0 da matriz saída, com o valor 10.65.
    matSaida[d][1] = String.valueOf(Double.valueOf(matSaida[d][0]) * 10.65);
    //Armazena na coluna 3 da matriz saída o resultado da multiplicação entre
    //a resposta da rede neural, que está alocada na coluna 2 da matriz saída, e o valor 10.65.
    matSaida[d][3] = String.valueOf(Double.valueOf(matSaida[d][2]) * 10.65);
    //Armazena na coluna 4 da matriz saída o resultado da divisão entre
    //os dados da coluna 1(resposta desejada) com os dados da coluna 3(resposta da rede).
    matSaida[d][4] = String.valueOf(Double.valueOf(matSaida[d][1])/
        Double.valueOf(matSaida[d][3]));
    //Armazena na quinta coluna da matriz saída o número 1 se o valor da coluna 4 da
    //matriz saída for maior ou igual a 0.85 e menor ou igual a 1.15.
    if ((Double.valueOf(matSaida[d][4])<= 1.15) & (Double.valueOf(matSaida[d][4])>= 0.85))
    {
        matSaida[d][5] = "1";
    }
    // Caso contrário armazena o valor 0 na quinta coluna da matriz saída.
    else matSaida[d][5] = "0";
}
}

```

```

// Abaixo será gravado no arquivo txt os dados da matriz saída.
File arquivo;
//Especifica o arquivo que será manipulado(uma representação abstrata do arquivo).
arquivo = new File("c:\\SAÍDA\\SAÍDA"+b+"."+c+".txt");
//Serve para escrever no arquivo.
FileOutputStream fos = new FileOutputStream(arquivo);
// Variável do tipo string que recebeu espaço em branco.
String espaco = "    ";
// Variável do tipo string que recebeu um separador de linha.
String pulalinha = System.getProperty ("line.separator");
for (int d = 0 ;d<198;d++)
{
    for (int f = 0 ;f<6;f++)
    {
        // Executa se o resto da divisão de f por 6 for igual a 0.
        if (f%6 == 0)
        {
            //Pula uma linha no arquivo txt.
            fos.write(pulalinha.getBytes());
        }
        // Escreve no arquivo txt o valor da matriz saída.
        fos.write(matSaida[d][f].getBytes());
        // Coloca no arquivo txt um espaço em branco.
        fos.write(espaco.getBytes());
    }
}

fos.close();// Fecha o arquivo.

}
}
}

catch (Exception ee) {
    ee.printStackTrace();
}

}
}

```

Apêndice B – Tabela com os dados da validação da melhor topologia de rede neural.

ENTRADA: 5; C. INTERM.: 5; SAÍDA: 1; TX. APREND.: 0,4						
DES. NORM.	DES.*10,65	SAÍDA NORM.	SAÍDA*10,65	DES/SAIDA	0,85 =<X<=1,15	0,75<=X<=1,25
0,157746479	1,68	0,165605036	1,76	0,952546389	1	1
0,023474178	0,25	0,040300616	0,43	0,582476899	0	0
0,073239437	0,78	0,0909343	0,97	0,805410469	0	1
0,281690141	3,00	0,282975122	3,01	0,995459031	1	1
0,028169014	0,30	0,025362618	0,27	1,110650868	1	1
0,225352113	2,40	0,189639974	2,02	1,188315458	0	1
0,244131455	2,60	0,256254526	2,73	0,952691291	1	1
0,001877934	0,02	0,008726274	0,09	0,215204572	0	0
0,244131455	2,60	0,228937963	2,44	1,066365104	1	1
0,356807512	3,80	0,344419333	3,67	1,035968304	1	1
0,291079812	3,10	0,305255245	3,25	0,953562033	1	1
0,105164319	1,12	0,084790757	0,90	1,240280456	0	1
0,085446009	0,91	0,0608334	0,65	1,404590398	0	0
0,000938967	0,01	0,007228336	0,08	0,129900847	0	0
0,605633803	6,45	0,571418958	6,09	1,059876985	1	1
0,356807512	3,80	0,405950885	4,32	0,878942565	1	1
0,292018779	3,11	0,289645415	3,08	1,008194031	1	1
0,071361502	0,76	0,047668398	0,51	1,497040077	0	0
0,319248826	3,40	0,335375035	3,57	0,951915894	1	1
0,014084507	0,15	0,015331417	0,16	0,918669647	1	1
0,137089202	1,46	0,150149536	1,60	0,913017817	1	1
0,016901408	0,18	0,018385536	0,20	0,919277419	1	1
0,01314554	0,14	0,017761975	0,19	0,740094508	0	0
0,323943662	3,45	0,403930183	4,30	0,801979342	0	1
0,262910798	2,80	0,253419383	2,70	1,037453392	1	1
0,094835681	1,01	0,088568637	0,94	1,070759181	1	1
0,539906103	5,75	0,563088362	6,00	0,958830158	1	1
0,093896714	1,00	0,093415514	0,99	1,005151181	1	1
0,110798122	1,18	0,134773347	1,44	0,822107075	0	1
0,033802817	0,36	0,035178582	0,37	0,960891967	1	1
0,892018779	9,50	0,806003371	8,58	1,106718422	1	1
0,497652582	5,30	0,513112379	5,46	0,969870543	1	1
0,159624413	1,70	0,16345275	1,74	0,976578325	1	1
0,159624413	1,70	0,165612013	1,76	0,963845616	1	1
0,117370892	1,25	0,123991619	1,32	0,946603432	1	1
0,187793427	2,00	0,167117486	1,78	1,123720993	1	1

ENTRADA: 5; C. INTERM.: 5; SAÍDA: 1; TX. APREND.: 0,4						
DES. NORM.	DES.*10,65	SAÍDA NORM.	SAÍDA*10,65	DES/SAIDA	0,85 =<X<=1,15	0,75<=X<=1,25
0,353051643	3,76	0,381350709	4,06	0,925792543	1	1
0,309859155	3,30	0,353502271	3,76	0,876540777	1	1
0,485446009	5,17	0,503642685	5,36	0,963869869	1	1
0,002816901	0,03	0,024415991	0,26	0,115371152	0	0
0,037558685	0,40	0,032288095	0,34	1,163236311	0	1
0,272300469	2,90	0,26537166	2,83	1,026109829	1	1
0,088262911	0,94	0,094653192	1,01	0,932487425	1	1
0,523943662	5,58	0,566099401	6,03	0,925532973	1	1
0,098591549	1,05	0,127110523	1,35	0,775636406	0	1
0,595305164	6,34	0,603356605	6,43	0,986655585	1	1
0,281690141	3,00	0,270185945	2,88	1,042578811	1	1
0,529577465	5,64	0,54706551	5,83	0,968032996	1	1
0,011267606	0,12	0,020401868	0,22	0,552283055	0	0
0,624413146	6,65	0,621942359	6,62	1,003972695	1	1
0,022535211	0,24	0,024583031	0,26	0,916697809	1	1
0,20657277	2,20	0,189233596	2,02	1,091628411	1	1
0,225352113	2,40	0,243564239	2,59	0,925226603	1	1
0,14084507	1,50	0,112554826	1,20	1,251346342	0	0
0,001877934	0,02	0,02169876	0,23	0,086545684	0	0
0,38028169	4,05	0,408196713	4,35	0,931613798	1	1
0,00657277	0,07	0,019679934	0,21	0,333983331	0	0
0,175586854	1,87	0,144000394	1,53	1,219349818	0	1
0,309859155	3,30	0,323660168	3,45	0,957359558	1	1
0,061032864	0,65	0,054467287	0,58	1,120541665	1	1
0,535211268	5,70	0,600047189	6,39	0,891948629	1	1
0,139906103	1,49	0,15355618	1,64	0,911106953	1	1
0,41971831	4,47	0,479616639	5,11	0,87511207	1	1
0,184037559	1,96	0,19036211	2,03	0,96677621	1	1
0,015023474	0,16	0,02523519	0,27	0,595338256	0	0
0,076056338	0,81	0,076569708	0,82	0,993295384	1	1
0,045070423	0,48	0,061710566	0,66	0,730351797	0	0
0,01314554	0,14	0,0186125	0,20	0,706274812	0	0
0,150234742	1,60	0,129320525	1,38	1,161723884	0	1
0,030046948	0,32	0,022610345	0,24	1,328902687	0	0
0,037558685	0,40	0,039398639	0,42	0,953299051	1	1
0,333333333	3,55	0,33470451	3,56	0,995903321	1	1
0,401877934	4,28	0,402168843	4,28	0,999276649	1	1
0,638497653	6,80	0,704535493	7,50	0,906267548	1	1
0,046948357	0,50	0,060586833	0,65	0,774893734	0	1

ENTRADA: 5; C. INTERM.: 5; SAÍDA: 1; TX. APREND.: 0,4						
DES. NORM.	DES.*10,65	SAÍDA NORM.	SAÍDA*10,65	DES/SAIDA	0,85 =<X<=1,15	0,75<=X<=1,25
0,101408451	1,08	0,13164923	1,40	0,770292776	0	1
0,033802817	0,36	0,030129422	0,32	1,121920519	1	1
0,084507042	0,90	0,078053869	0,83	1,082675892	1	1
0,058215962	0,62	0,049996917	0,53	1,164391035	0	1
0,028169014	0,30	0,03548182	0,38	0,793899924	0	1
0,126760563	1,35	0,110530874	1,18	1,146833983	1	1
0,129577465	1,38	0,157093822	1,67	0,824841254	0	1
0,291079812	3,10	0,284339781	3,03	1,023704145	1	1
0,2	2,13	0,189242194	2,02	1,05684676	1	1
0,15399061	1,64	0,173052727	1,84	0,889847924	1	1
0,101408451	1,08	0,10811336	1,15	0,937982603	1	1
0,042253521	0,45	0,04780028	0,51	0,883959701	1	1
0,000938967	0,01	0,013905655	0,15	0,06752411	0	0
0,516431925	5,50	0,523510043	5,58	0,9864795	1	1
0,553051643	5,89	0,602943673	6,42	0,917252585	1	1
0,145539906	1,55	0,144921133	1,54	1,004269719	1	1
0,071361502	0,76	0,087388441	0,93	0,81660116	0	1
0,248826291	2,65	0,257501729	2,74	0,966309207	1	1
0,326760563	3,48	0,363137815	3,87	0,899825217	1	1
0,128638498	1,37	0,128614739	1,37	1,000184732	1	1
0,178403756	1,90	0,138647544	1,48	1,286743	0	0
0,384976526	4,10	0,38628029	4,11	0,996624825	1	1
0,017840376	0,19	0,014357017	0,15	1,242624172	0	1
0,464788732	4,95	0,498166089	5,31	0,932999541	1	1
0,178403756	1,90	0,194766296	2,07	0,915988853	1	1
0,135211268	1,44	0,159448075	1,70	0,847995615	0	1
0,033802817	0,36	0,019043165	0,20	1,775063014	0	0
0,300469484	3,20	0,279808064	2,98	1,073841402	1	1
0,0657277	0,70	0,064199279	0,68	1,023807443	1	1
0,253521127	2,70	0,284661336	3,03	0,890606116	1	1
0,017840376	0,19	0,011398406	0,12	1,56516406	0	0
0,076995305	0,82	0,103340012	1,10	0,745067698	0	0
0,01314554	0,14	0,02188108	0,23	0,600771978	0	0
0,110798122	1,18	0,116579308	1,24	0,950409842	1	1
0,061032864	0,65	0,075467795	0,80	0,808727274	0	1
0,003755869	0,04	0,011857141	0,13	0,316760097	0	0
0,15399061	1,64	0,119640287	1,27	1,287113342	0	0
0,214084507	2,28	0,181358038	1,93	1,180452265	0	1
0,835680751	8,90	0,759369643	8,09	1,100492704	1	1

ENTRADA: 5; C. INTERM.: 5; SAÍDA: 1; TX. APREND.: 0,4						
DES. NORM.	DES.*10,65	SAÍDA NORM.	SAÍDA*10,65	DES/SAIDA	0,85 =<X<=1,15	0,75<=X<=1,25
0,022535211	0,24	0,025066943	0,27	0,899001151	1	1
0,015962441	0,17	0,019850528	0,21	0,804131794	0	1
0,579342723	6,17	0,638394882	6,80	0,907499009	1	1
0,137089202	1,46	0,131384317	1,40	1,043421358	1	1
0,055399061	0,59	0,038216352	0,41	1,449616673	0	0
0,169014085	1,80	0,165656572	1,76	1,020267913	1	1
0,002816901	0,03	0,014374615	0,15	0,195963577	0	0
0,098591549	1,05	0,117905623	1,26	0,83619039	0	1
0,009389671	0,10	0,010410373	0,11	0,901953379	1	1
0,382159624	4,07	0,44853467	4,78	0,852018025	1	1
0,563380282	6,00	0,521389439	5,55	1,080536428	1	1
0,422535211	4,50	0,479193142	5,10	0,881763893	1	1
0,442253521	4,71	0,445981373	4,75	0,991641238	1	1
0,125821596	1,34	0,113506643	1,21	1,108495437	1	1
0,370892019	3,95	0,403744252	4,30	0,918631082	1	1
0,269483568	2,87	0,26542053	2,83	1,015307926	1	1
0,183098592	1,95	0,187194201	1,99	0,978121068	1	1
0,371830986	3,96	0,422257259	4,50	0,880579264	1	1
0,482629108	5,14	0,540803308	5,76	0,892430021	1	1
0,099530516	1,06	0,110581454	1,18	0,900065176	1	1
0,216901408	2,31	0,21665859	2,31	1,00112074	1	1
0,18028169	1,92	0,156668918	1,67	1,150717655	0	1
0,653521127	6,96	0,690568133	7,35	0,946352859	1	1
0,066666667	0,71	0,07264695	0,77	0,917680192	1	1
0,555868545	5,92	0,523828712	5,58	1,061164713	1	1
0,056338028	0,60	0,05467586	0,58	1,030400406	1	1
0,054460094	0,58	0,063184506	0,67	0,86192165	1	1
0,000938967	0,01	0,004175785	0,04	0,224859989	0	0
0,143661972	1,53	0,113840773	1,21	1,261955341	0	0
0,010328638	0,11	0,004661583	0,05	2,215693426	0	0
0,088262911	0,94	0,11022507	1,17	0,800751688	0	1
0,083568075	0,89	0,084568048	0,90	0,988175525	1	1
0,41314554	4,40	0,380480454	4,05	1,085852207	1	1
0,008450704	0,09	0,011637867	0,12	0,726138582	0	0
0,020657277	0,22	0,020707091	0,22	0,997594354	1	1
0,228169014	2,43	0,201437205	2,15	1,13270542	1	1
0,58028169	6,18	0,570411006	6,07	1,017304512	1	1
0,183098592	1,95	0,195661744	2,08	0,935791476	1	1
0,111737089	1,19	0,097901661	1,04	1,141319644	1	1
0,291079812	3,10	0,313558758	3,34	0,928310259	1	1
0,002816901	0,03	0,009153632	0,10	0,307735887	0	0

ENTRADA: 5; C. INTERM.: 5; SAÍDA: 1; TX. APREND.: 0,4						
DES. NORM.	DES.*10,65	SAÍDA NORM.	SAÍDA*10,65	DES/SAIDA	0,85 =<X<=1,15	0,75<=X<=1,25
0,281690141	3,00	0,299608311	3,19	0,940194682	1	1
0,369014085	3,93	0,385107376	4,10	0,958210899	1	1
0,224413146	2,39	0,195646837	2,08	1,147031811	1	1
0,0657277	0,70	0,065269571	0,70	1,007019023	1	1
0,014084507	0,15	0,014056728	0,15	1,001976183	1	1
0,633802817	6,75	0,63428183	6,76	0,999244795	1	1
0,402816901	4,29	0,420210847	4,48	0,958606623	1	1
0,057276995	0,61	0,049977196	0,53	1,146062601	1	1
0,82629108	8,80	0,812761448	8,66	1,016646499	1	1
0,238497653	2,54	0,222213697	2,37	1,073280612	1	1
0,41314554	4,40	0,436570085	4,65	0,946344135	1	1
0,107981221	1,15	0,099807478	1,06	1,081895095	1	1
0,211267606	2,25	0,213567608	2,27	0,989230568	1	1
0,027230047	0,29	0,04048227	0,43	0,672641307	0	0
0,480751174	5,12	0,523034426	5,57	0,919157804	1	1
0,093896714	1,00	0,09424453	1,00	0,996309429	1	1
0,047887324	0,51	0,066373732	0,71	0,721480061	0	0
0,002816901	0,03	0,020682784	0,22	0,136195443	0	0
0,061971831	0,66	0,080038299	0,85	0,774277214	0	1
0,084507042	0,90	0,091260269	0,97	0,926000364	1	1
0,788732394	8,40	0,77910931	8,30	1,012351391	1	1
0,016901408	0,18	0,025422308	0,27	0,664825873	0	0
0,007511737	0,08	0,023876597	0,25	0,31460668	0	0
0,610328638	6,50	0,648446795	6,91	0,941216216	1	1
0,713615023	7,60	0,728058322	7,75	0,980161893	1	1
0,089201878	0,95	0,097950092	1,04	0,910687029	1	1
0,002816901	0,03	0,007932004	0,08	0,355131051	0	0
0,092957746	0,99	0,077751238	0,83	1,195578988	0	1
0,135211268	1,44	0,136657262	1,46	0,989418828	1	1
0,148356808	1,58	0,163340768	1,74	0,908265647	1	1
0,515492958	5,49	0,535513336	5,70	0,962614604	1	1
0,077934272	0,83	0,107386547	1,14	0,725735895	0	0
0,497652582	5,30	0,572296284	6,09	0,869571577	1	1
0,033802817	0,36	0,028065225	0,30	1,204437772	0	1
0,150234742	1,60	0,197940911	2,11	0,758987828	0	1
0,118309859	1,26	0,129969637	1,38	0,910288447	1	1
0,004694836	0,05	0,00590269	0,06	0,795372247	0	1
0,863849765	9,20	0,885207099	9,43	0,975873065	1	1
0,41314554	4,40	0,461688636	4,92	0,894857504	1	1
0,035680751	0,38	0,037814568	0,40	0,94357155	1	1
0,464788732	4,95	0,470637281	5,01	0,987573128	1	1
0,19342723	2,06	0,205723552	2,19	0,940228905	1	1
0,023474178	0,25	0,040078632	0,43	0,585703072	0	0
Total de acertos					132	160
Porc. de acertos					66,67%	80,81%