

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

**RECONHECIMENTO DE PADRÕES EM GRÁFICOS DE VELA
DA BOLSA DE VALORES
ATRAVÉS DE UMA REDE NEURAL**

MARCOS SEGALLA PIZZOLATTI

2009

**MARCOS SEGALLA
PIZZOLATTI**

TCC - UFSC

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

MARCOS SEGALLA PIZZOLATTI

**RECONHECIMENTO DE PADRÕES EM GRÁFICOS DE VELA DA
BOLSA DE VALORES ATRAVÉS DE UMA REDE NEURAL**

FLORIANÓPOLIS

2009

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

MARCOS SEGALLA PIZZOLATTI

**RECONHECIMENTO DE PADRÕES EM GRÁFICOS DE VELA DA
BOLSA DE VALORES ATRAVÉS DE UMA REDE NEURAL**

Trabalho apresentado ao Curso de
Graduação em Sistemas de Informação da
Universidade Federal de Santa Catarina
como parte dos requisitos para a obtenção
do título de Sistemas de Informação.

Orientador: Professor Jovelino Falqueto.

FLORIANÓPOLIS

2009

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

MARCOS SEGALLA PIZZOLATTI

**RECONHECIMENTO DE PADRÕES EM GRÁFICOS DE VELA DA
BOLSA DE VALORES ATRAVÉS DE UMA REDE NEURAL**

Este Trabalho de Graduação foi julgado adequado para a obtenção do título de Sistemas de Informação e aprovado em sua forma final pela Comissão examinadora e pelo Curso de Sistemas de Informação da Universidade Federal de Santa Catarina.

Maria Marta Leite
Coordenadora do Curso

Renato Cislighi
Professor da Disciplina

Comissão Examinadora:

Prof. Jovelino Falqueto
Orientador

Prof. João Bosco Sobral
Co-orientador

Prof. Fernando Ostuni Gauthier
Avaliador

Prof. Mauro Roisenberg
Avaliador

Pizzolatti, Marcos Segalla, 1983-

Reconhecimento de Padrões em Gráficos de Vela da Bolsa de Valores Através de uma Rede Neural / Marcos Segalla Pizzolatti. – 2009.

70 f. : il. color. ;

Orientador: Jovelino Falqueto.

Trabalho de conclusão de curso (graduação) – Universidade Federal de Santa Catarina, Curso de Sistemas de Informação, 2009.

1. Rede Neural. 2. Bolsa de Valores. 3. Gráfico de Velas. I. Falqueto, Jovelino. II. Universidade Federal de Santa Catarina. Curso de Sistemas de Informação. III. Reconhecimento de Padrões em Gráficos de Vela da Bolsa de Valores Através de uma Rede Neural.

ABSTRACT

This project is about Artificial Neural Network and the objective is to recognize the following patterns of the Japanese Candlestick: hammers and hanging man, engulfing pattern and harami lines. The Neural Network is composed of four input nodes, three hidden nodes and two output nodes. The values (patterns) of the input layer were extracted from a table provided by the Stock Exchange of Sao Paulo, for the stock of Petrobras, from 2006 to 2008. The values of the hammer and hanging man patterns are the following: prices of opening/closing maximum/minimum, involving only a single candle. The values of the engulfing pattern and harami lines, involving two candles are the following: prices of opening/closing of the first candle and prices of opening/closing of the second candle. With these values, the Neural Network tries to perform the training, separately, for the recognition of the patterns in question.

RESUMO

Este projeto constitui-se na construção de uma Rede Neural Artificial (RNA) com o objetivo de reconhecer os padrões de reversão martelo e enforcado, padrão envolvente e linhas harami existentes no gráfico de velas japonês. A RNA é composta de quatro nós de entrada, três nós intermediários e dois nós de saída. Os valores (padrões) da camada de entrada foram extraídos de uma tabela fornecida pela Bolsa de Valores de São Paulo, para as ações da Petrobras, no intervalo de tempo de 2006 a 2008. Os valores do padrão martelo e enforcado são os seguintes: preços de abertura/fechamento e preços máximo/mínimo, envolvendo apenas uma única vela. Já os valores dos padrões padrão envolvente e linhas harami, envolvendo duas velas, são: preços de abertura/fechamento da primeira vela e preços de abertura/fechamento da segunda vela. Com estes valores, a RNA tenta realizar o treinamento, separadamente, para o reconhecimento dos padrões em questão.

LISTA DE FIGURAS

Figura 1: Configuração da Rede Neural Artificial do tipo Conexionista	14
Figura 2: Exemplo ilustrativo de velas.....	19
Figura 3: Corpo sem sombras.....	20
Figura 4: Martelo.....	21
Figura 5: Enforcado.....	22
Figura 6: Padrão envolvente.....	23
Figura 7: Harami altista em tendência de baixa	24
Figura 8: Harami baixista em tendência de alta	24
Figura 9: Cálculo das saídas e dos erros	26
Figura 10: Ajuste dos pesos	27
Figura 11: Gráfico de Velas da Bolsa de Valores	28
Figura 12: Interface do programa.....	29
Figura 13: Configuração da RNA para padrões envolvendo uma vela	33
Figura 14: Padrão 1 (Enforcado).....	34
Figura 15: Padrão 2 (Enforcado).....	34
Figura 16: Padrão 3 (Enforcado).....	35
Figura 17: Padrões 4 e 5 (Martelo e Enforcado)	35
Figura 18: Configuração da RNA para padrões envolvendo duas velas	36
Figura 19: Padrão 1 (Padrão Envolvente Altista).....	38
Figura 20: Padrão 2 (Padrão Envolvente Altista).....	38
Figura 21: Padrão 3 (Padrão Envolvente Altista).....	39
Figura 22: Padrão 4 (Padrão Envolvente Altista).....	39
Figura 23: Padrão 5 (Padrão Envolvente Baixista)	40
Figura 24: Padrões 1 e 2 (Linhas Harami Baixista e Altista)	40
Figura 25: Padrão 3 (Linhas Harami Baixista).....	41
Figura 26: Padrão 4 (Linhas Harami Altista)	41

Figura 27: Padrão 5 (Linhas Harami Altista)	42
Figura 28: Mensagem de erro – Pesos (Intermediário, Viés, Saída) e Época	45
Figura 29: Mensagem de erro – Pesos (Intermediário, Viés, Saída)	45
Figura 30: Mensagem de erro – Pesos (Intermediário, Viés) e Época	45
Figura 31: Mensagem de erro – Pesos (Intermediário, Saída) e Época.....	45
Figura 32: Mensagem de erro – Pesos (Viés, Saída) e Época	45
Figura 33: Mensagem de erro – Pesos (Intermediário, Viés).....	45
Figura 34: Mensagem de erro – Pesos (Intermediário) e Época	46
Figura 35: Mensagem de erro – Pesos (Intermediário, Saída)	46
Figura 36: Mensagem de erro – Pesos (Viés, Saída).....	46
Figura 37: Mensagem de erro – Pesos (Viés) e Época.....	46
Figura 38: Mensagem de erro – Pesos (Saída) e Época	46
Figura 39: Mensagem de erro – Pesos (Intermediário)	46
Figura 40: Mensagem de erro – Pesos (Saída).....	46
Figura 41: Mensagem de erro – Pesos (Viés).....	46
Figura 42: Mensagem de erro – Época.....	47
Figura 43: Mensagem de erro – Entradas.....	47
Figura 44: Conclusão do Treinamento do Neurônio Artificial	48
Figura 45: Teste de um padrão (Reconhecido)	48
Figura 46: Teste de um padrão (Não Reconhecido).....	49

LISTA DE TABELAS

Tabela 1: Valores (em reais) dos cinco padrões martelo e enforcado	33
Tabela 2: Valores (em reais) dos cinco padrões padrão envolvente	37
Tabela 3: Valores (em reais) dos cinco padrões linhas harami	37

SUMÁRIO

1 – INTRODUÇÃO	13
2 – OBJETIVOS.....	16
2.1 – Objetivo Geral	16
2.2 – Objetivos Específicos	16
3 – REVISÃO BIBLIOGRÁFICA.....	17
3.1 – Bolsas de Valores	17
3.1.1 – A Lógica da Bolsa de Valores	17
3.1.2 – Tipos de Ações.....	18
3.2 – Gráfico de Velas	18
3.2.1 – Indicadores de Reversão	20
3.2.1 – Padrões Martelo e Enforcado	20
3.2.2 – Padrões Envolventes	22
3.2.3 - Linhas Harami	23
3.3 – Redes Neurais Artificiais.....	25
3.3.1 – Algoritmo de Retro-propagação.....	26
4 – IMPLEMENTAÇÃO	28
4.1 – Interface e Características	29
4.2 – Funcionamento do Programa	32
4.2.1 – Padrão envolvendo uma vela (Martelo e Enforcado)	32
4.2.1.1 – Gráficos do Padrão Martelo e Enforcado	33
4.4.2 – Padrão envolvendo duas velas (Padrão Envolvente e Linhas Harami).....	36
4.4.2.1 – Gráficos do Padrão Envolvente e Linhas Harami	37
4.4.3 – Implementação para o Treinamento da Rede Neural Artificial	42
4.4.4 – Implementação para o Teste da Rede Neural Artificial	43
4.2 – Mensagens de Erro.....	43
4.5 – Figuras do Programa Após Treinamento e Teste	47
5 – CONCLUSÃO	50
6 – REFERÊNCIAS BIBLIOGRÁFICAS.....	51

ANEXO I.....	53
Algoritmo de Treinamento.....	53
Passos para Frente	54
Passos para Trás.....	55
Cálculo do Erro Global.....	57
Algoritmo de Teste	57
ANEXO II	59

1 - Introdução

Os investimentos em Bolsa de Valores tem se popularizado mundialmente como opção de investimento. Muitos investidores iniciantes, sem amplo conhecimento no mercado financeiro, ao se aventurarem no investimento de ações, acabam por perder grande parte de seus investimentos. Uma ferramenta que auxiliasse na decisão de compra/venda de ações seria de grande importância e ajuda para esta classe de investidores.

Nos dias atuais, investidores que não sabem interpretar corretamente gráficos de ações financeiras, poderiam ser facilmente auxiliados por programas implementados através de Redes Neurais Artificiais com o intuito de prever as tendências do mercado.

Investidores mais amadurecidos poderiam também se utilizar de ferramentas mais robustas implementadas por Redes Neurais Artificiais para o auxílio em suas decisões de compra/venda.

Este projeto consiste na construção de uma RNA capaz de reconhecer alguns padrões (indicadores de reversão) existentes nos gráficos de vela. Especificamente neste projeto, a RNA identifica os seguintes padrões de indicadores de reversão: martelo e enforcado, padrão envolvente e linhas harami (gravidez).

Indicadores de reversão têm como objetivo “avisar” ou dar um alerta de que uma tendência de alta ou de baixa está para perder esta tendência, revertendo a situação de compra/venda no mercado financeiro.

O gráfico de velas foi inventado pelos japoneses com intuito de aplicar na Bolsa de Valores do arroz. Utilizando uma RNA, que é um sistema inteligente capaz de processar informações de maneira similar à neurônios cerebrais, com limitações, é possível fazer um treinamento capaz de aprender os padrões existentes no gráfico japonês a fim de reconhecê-lo posteriormente, sem a intervenção do cérebro humano, e sim do computador.

O ramo Conexionista tem como objetivo a busca pela simulação de comportamentos inteligentes baseados no funcionamento de neurônios cerebrais. *É um sistema dentro da Inteligência Artificial que visa estudar a mente por uma perspectiva computacional.* (ELLIS, 2009)

Outro objetivo dentro da Inteligência Artificial tem como foco a aplicação do conhecimento obtido no modelamento da mente, explicado por Ellis, para solução de

problemas práticos, como a de encontrar padrões nos preços de ações negociadas em Bolsa de Valores, por exemplo.

Além do ramo Conexionista, implementado na prática computacional pelas Redes Neurais Artificiais, a Inteligência Artificial tem outros dois: a Inteligência Artificial Simbólica e a Inteligência Artificial Evolucionária. No ramo Simbólico os Sistemas Especialistas são os produtos mais conhecidos e no ramo Evolucionário os Algoritmos Genéticos é o que o mercado de aplicativos mais adota.

Na Figura 1 se pode observar um esboço da configuração da rede utilizada neste trabalho.

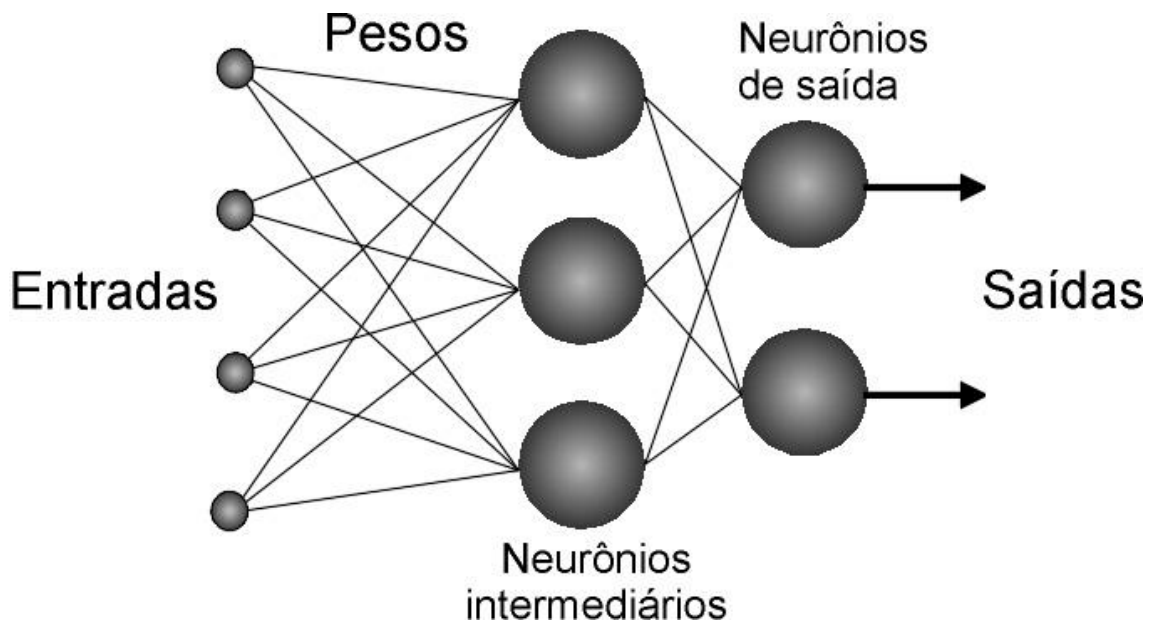


Figura 1: Configuração da Rede Neural Artificial do tipo Conexionista

Como se pode observar, a configuração da rede consta de quatro nós de entrada, três nós intermediários e dois nós de saída que correspondem a dois valores de saída.

De acordo com esta lógica, terão como dados de entrada os seguintes valores: valor de abertura, valor de fechamento, valor máximo e valor mínimo (no caso do padrão martelo e enforcado) e valores de abertura e fechamento da primeira vela e valores de abertura e fechamento da segunda vela (no caso dos padrões “padrão envolvente” e “linhas harami”). Os dois valores da camada de saída caracterizam os padrões a serem reconhecidos pela RNA.

No padrão martelo e enforcado, o valor de abertura corresponde ao preço de abertura da ação (preço de primeiro negócio efetuado com esta ação) e o valor de fechamento corresponde ao preço de fechamento da ação (preço do último negócio), ambos no decorrer de um dia. O valor máximo e mínimo corresponde aos preços máximos e mínimos atingidos no dia. A saída esperada neste padrão corresponde a dois valores no intervalo de zero a um, x e y , escolhidos para o treinamento da RNA.

Nos padrões, padrão envolvente e linhas harami, ambos os valores, de abertura e fechamento das duas velas em questão, correspondem aos preços de abertura e fechamento de cada vela atingido no dia. As saídas desses dois padrões também correspondem a dois valores, x e y , escolhidos no intervalo de zero a um, para o treinamento da RNA.

O computador utilizado é um notebook de domínio pessoal com a seguinte configuração:

- Intel(R) Core(TM) 2 Duo CPU T7100 1.80GHz
- 2GB de memória SDRAM
- HD SATA de 160GB
- Windows VISTA Ultimate
- NetBeans 6.5
- JAVA(TM) 6 Update 12

Com base nestes dados, a RNA será capaz de aprender os padrões treinados a fim de identificá-los posteriormente, após o devido treinamento. Os dados extraídos são da Bolsa de Valores das ações da Petrobras, num período de três anos (2006 - 2008).

Vale enfatizar que neste trabalho os preços extraídos dos gráficos da Bolsa de Valores se referem ao período de um dia de negociação na Bolsa de Valores, porém também existem representações onde os preços se referem a períodos diferentes, como semanas, meses, anos, etc.

2 – Objetivos

2.1 – Objetivo Geral

O objetivo geral deste projeto consiste na construção de uma RNA capaz de reconhecer indicadores de reversão nos gráficos de vela da Bolsa de Valores do Petróleo.

2.2 – Objetivos Específicos

- Estudar e avaliar a lógica dos gráficos de vela, para que através desta avaliação, se possa implementar uma RNA capaz de seguir esta lógica.
- Estudar e avaliar conceitos importantes sobre a Bolsa de Valores, para auxiliar na lógica de implementação da RNA.
- Estudar e avaliar o funcionamento de uma RNA, a fim de realizar a implementação de acordo com este tipo de rede neural.
- Construir uma Rede Neural Artificial capaz de entender a lógica por trás de alguns padrões dos gráficos de vela e conseqüentemente aprendê-los;
- Extrair dados reais da bolsa de valores e realizar treinamentos na rede neural.
- Construir uma interface capaz de interagir com o problema em questão.

3 – Revisão Bibliográfica

3.1 – Bolsas de Valores

As bolsas de valores são instituições que tem como principal objetivo administrar os mercados financeiros. Através das bolsas de valores são negociados valores mobiliários, os quais são negociados através de sistemas eletrônicos para garantir as compras e vendas desses valores. As regras de negociações estabelecem as diferenças entres os mercados financeiros. (PORTAL DO INVESTIDOR, 2009)

Segundo a Lei nº 10.303, valores mobiliários são títulos ou contratos de investimento coletivo, que geram direito de participação, de parceria ou de remuneração, inclusive resultante de prestação de serviços, cujos rendimentos advêm do esforço do empreendedor ou de terceiros. (PORTAL DO INVESTIDOR, 2009)

Pode-se dizer que proporcionar um ambiente amplamente transparente e líquido é a principal função das bolsas. Em relação às corretoras de valores que operam nas bolsas, as mesmas possuem autonomia de exercer poderes de auto-regulamentação. Os grandes investidores podem ter acesso aos sistemas de negociação, com o objetivo da efetuação de transações de compras e vendas, por meio das corretoras. (PORTAL DO INVESTIDOR, 2009)

As corretoras de valores são sociedades que servem como intermediação na compra/venda de títulos na BOVESPA (Bolsa de Valores do Estado de São Paulo).

3.1.1 – A Lógica da Bolsa de Valores

Para que cada um dos sócios de uma determinada empresa possa fazer parte dos lucros, ou prejuízos da mesma, deve-se fazer um contrato social. Desta maneira, na possibilidade de qualquer um desses sócios sair da empresa, independente do motivo, o mesmo poderá vender sua parte de ações para outro membro permanente da empresa. Porém deve-se ter em conta que, para muitas empresas, o contrato social da mesma é impactado com a saída de um de seus sócios. (LIMA, 2007)

Um conceito de “ação” teve de ser criado para a facilitação deste impacto. Esta ação terá como objetivo mostrar quem será dono de que dentro da empresa, podendo ser

negociável, o que significa que a mesma poderá ser vendida a qualquer membro da empresa pelo preço que desejar. Com isto, sem a necessidade de mudança no contrato social, a troca de membros proprietários de ações dentro de empresa se faz com maior facilidade. (LIMA, 2007)

Resumindo, o lugar onde se realizam as compras e vendas de ações de empresas é o que chamamos de bolsa de valores. Com isso, mantém registrado todos os trâmites das ações, seja de vendas ou de compras, assim como os nomes das pessoas que compram e vendem. As empresas das quais estão sendo negociadas as ações são mantidas em constante informação. Desta maneira, a empresa em que as ações foram negociadas poderá saber, através desse registro, quem será o novo dono de determinado volume de ações. (LIMA, 2007)

3.1.2 – Tipos de Ações

Os dois tipos de ações conhecidas no Brasil são as ordinárias e as preferenciais. Quando uma pessoa dona de uma ação de uma empresa tem o direito de opinar na administração da mesma, esta ação é dita ordinária. (MARQUES, 2007)

Quando seus donos têm o direito de serem os primeiros beneficiados na distribuição e dividendos ou lucros, as ações são conhecidas como preferenciais. Os donos das ações preferenciais não podem votar nas Assembléias Gerais da empresa, porém o voto será necessário caso se ultrapasse um tempo de três anos sem qualquer distribuição de dividendos das ações. (MARQUES, 2007)

Neste trabalho foi escolhida a ação do tipo preferencial (pelo motivo de conhecer investidores que possuem acesso a esses tipos de ações), a qual se refere a pequenas movimentações, sendo a mesma geralmente um pouco mais barata que as ações do tipo ordinárias.

3.2 – Gráfico de Velas

Por volta do ano 1750, os japoneses criaram um sistema gráfico de reconhecimento e avaliação de uma estrutura econômica, a de arroz, por exemplo, muito disputado na época. Este gráfico de avaliação dos movimentos diários das ações, pelo fato de serem muito parecidos com velas, foi denominado gráfico de velas. Nos anos 90,

com os estudos realizados por Steven Nison, autor do livro intitulado *Japanese Candlestick Charting Techniques*, foi quando o gráfico de velas japonês tornou-se mais popular no mundo. Um exemplo ilustrativo é apresentado na Figura 2, representando-se uma vela do gráfico com as informações de preço máximo (período onde se obteve maior preço), mínimo (período onde se obteve menor preço), abertura (preço do início) e fechamento (preço do final), com seus corpos e sombras. (NISON, 1992)

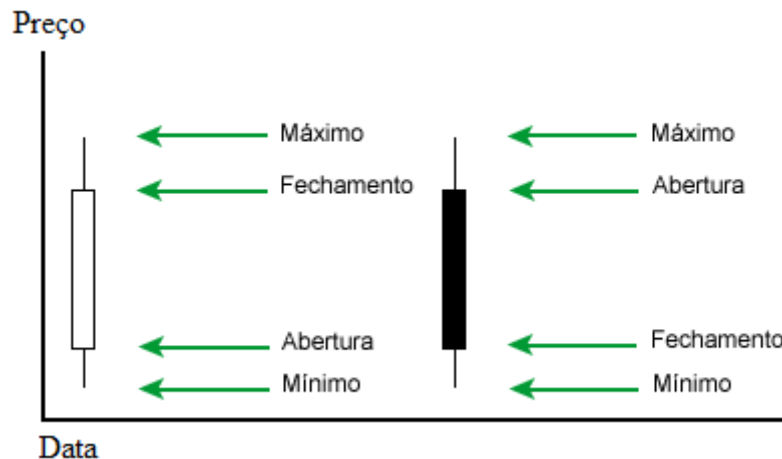


Figura 2: Exemplo ilustrativo de velas

Como podemos verificar na Figura 2, corpo e sombras são duas partes compostas por uma vela no gráfico de velas. A parte composta entre a abertura e fechamento é denominada de corpo. A cor branca do corpo da vela é devido ao fato de que o preço de abertura num determinado período de tempo foi mais baixo que o preço de fechamento. Já a cor preta refere-se a um determinado período de tempo em que o preço de abertura foi maior do que o preço de fechamento. (NELOGICA, 2009)

As sombras são segmentos de reta entre os preços de fechamento/abertura e máximo/mínimo, dependendo da cor do corpo, e nos mostram os valores mínimos e máximos alcançados pelos preços das ações. No caso da vela não possuir sombra superior/inferior, significa que o valor de abertura/fechamento é igual ao valor máximo/mínimo no período correspondente, como mostrado na Figura 3. (NELOGICA, 2009)

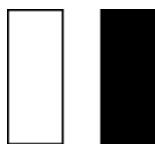


Figura 3: Corpo sem sombras

3.2.1 – Indicadores de Reversão

Os indicadores de reversão (dispositivos que alertam quando uma tendência de alta ou baixa está para se reverter) são de suma importância no que diz respeito às funções de alertar para mudanças de tendências, confirmar sinalizações dadas por outros tipos de indicadores e prever o mercado financeiro em futuras movimentações. Através dessas três funções, de alertar, confirmar e prever, é possível identificar regiões de esgotamento de tendências, alertar quando bens e direitos de uma determinada empresa (ativo) está mais ou menos valorizado, não permitir escolhas equivocadas e finalmente dar a correta visão do momento mais propício de se fazer determinado negócio financeiro. (MALHEIROS, 2005)

3.2.1 – Padrões Martelo e Enforcado

Os padrões martelo e enforcado são percebidos pelo fato da sombra inferior ser pelo menos o dobro do tamanho do corpo da vela. Não deveria existir uma sombra superior, porém se a mesma existir deverá ser mínima. A cor da sombra não tem qualquer importância nesse tipo de padrão. Sendo assim, o corpo tem que estar na parte superior da vela, formando com sua sombra o perfil aproximado de um martelo ou de uma pessoa enforcada. Na Figura 4 estão exemplificados vários martelos, tanto com corpos positivos quanto com corpos negativos. (BASTTER.COM, 2002)

Assim, tanto o martelo quanto o enforcado tem o mesmo perfil. Quando o padrão aparece após sucessivos períodos de alta é chamado enforcado (e prenuncia um possível início de tendência de baixa). Quando aparece em seguida a uma sucessão de baixas, é chamado martelo (indica uma possível reversão para se iniciar uma alta). Os dois padrões indicam então uma reversão do movimento dominante até que ocorram.

O padrão martelo é assim denominado por testar a resistência de uma tendência no mercado, vista pela famosa expressão “o mercado está martelando a base”. Assim

podemos assumir que quando o padrão aparecer numa tendência baixista do mercado significa que teremos uma formação altista, formando o padrão martelo, tal qual como mostra a Figura 4. (NISON, 1992)



Figura 4: Martelo

Fonte: (<http://www.bastter.com/BR/MERCADO/Aprendizado/Img/mart.tnlp4.gif>)

Quando aparecer no gráfico uma seqüência de velas com tendência de alta, possivelmente um perfil baixista se formará e então teremos um padrão enforcado para nos mostrar isso. O padrão enforcado é assim denominado devido ao fato de que o mercado sugere que os comprados possivelmente terão problemas, lembrando uma figura de uma pessoa enforcada que balança suas pernas. (NISON, 1992)

Segundo Fernando Braga de Matos, linha de resistência é um movimento de venda durante um determinado período, sendo a oferta maior que a procura. Testar esta resistência significa inverter a situação do mercado, vindo assim uma linha de suporte, exatamente a antítese da linha de resistência.

Exemplificando melhor os termos de comprados e vendidos, pode-se dizer que os comprados são assim denominados pelo fato dos operadores comprarem ação de uma empresa e tê-las em seu domínio. Já os vendidos são os operadores que venderam ações de uma empresa e possuem fundos para adquirirem novas ações caso o mercado esteja a favor. Os preços das ações são formados pela movimentação de compra e venda desses

operadores, assim quando os comprados são maiores que os vendidos os preços tendem a subir, já quando os vendidos estão mais fortes, os preços das ações tendem a baixar.

Na Figura 5 observa-se o aparecimento do padrão enforcado numa tendência de alta.



Figura 5: Enforcado

Fonte: (<http://www.bastter.com/BR/MERCADO/Aprendizado/Img/enf.bb.gif>)

3.2.2 – Padrões Envolventes

A combinação de duas velas indicando uma possível reversão, em especial depois de uma tendência longa de mercado, é um padrão envolvente. O padrão envolvente altista é caracterizado pela combinação de duas velas numa tendência de baixa, onde deverá existir um corpo branco capaz de envolver por completo outro corpo negro anterior, o qual deverá obviamente ser menor. Falando em mercado de ações, o padrão envolvente altista indica que os vendidos (operadores com fundos para comprarem ações caso os preços lhe convenham) perderam domínio no mercado para os comprados (operadores com ações para vender). (NISON, 1992)

Num mercado de tendência altista pode ocorrer a existência de um padrão envolvente baixista. Tal padrão é semelhante ao padrão envolvente altista, o que difere um do outro é o fato de que o padrão envolvente baixista possui uma longa vela negra

que envolve por completo uma vela branca antecedente. Ao contrário do padrão envolvente altista, o padrão envolvente baixista é uma boa indicação de que os vendidos estarão por tomar conta do mercado financeiro. (NISON, 1992)

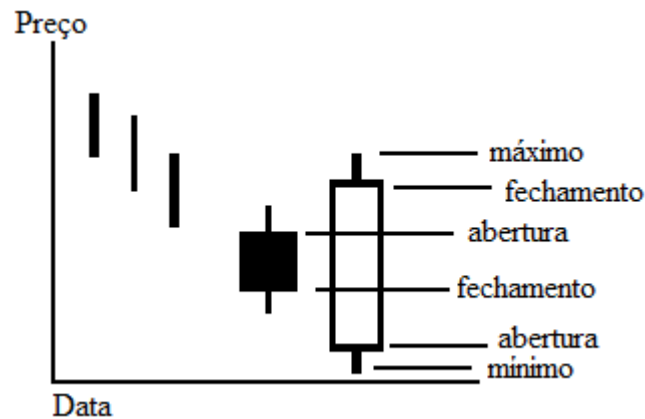


Figura 6: Padrão envolvente

3.2.3 - Linhas Harami

O padrão de linhas harami é exatamente o inverso de um padrão envolvente. No padrão de linhas harami uma vela maior engloba uma vela menor que sucede a maior. A tradução deste padrão dá-se pela palavra “gravidez”, o que podemos observar na figura desse padrão, onde a vela maior é a mãe que tem seu filho, a vela menor, na barriga. Numa tendência de baixa, o padrão linha harami terá a vela mãe com o corpo cheio (perto ou vermelho), já numa tendência altista, a vela mãe terá corpo vazio (branco). Apesar da cor da vela filha, na maioria dos casos, ser da cor inversa da vela mãe, na realidade isto não tem importância, podendo as velas filhas serem também da mesma cor da vela mãe. (BASTTER.COM, 2002)



Figura 7: Harami altista em tendência de baixa
 Fonte: (http://www.cjb.com.br/imagens/exemplo_candle_harami_altista.gif)



Figura 8: Harami baixista em tendência de alta
 Fonte: (<http://www.forumbolsa.com.br/img/candles/harami.gif>)

3.3 – Redes Neurais Artificiais

De modo semelhante a um cérebro, uma RNA é capaz de processar dados, porém com limitações. Da mesma forma que o cérebro humano possui uma complexa organização de neurônios, a RNA visa ter este mesmo princípio para poder ser capaz de processar informações. Uma RNA trabalha de modo semelhante ao que se supõe ser o funcionamento elementar do cérebro no processamento de dados. RNAs podem ser habilitadas para gravar conhecimentos de acordo com o aprendizado recebido, podendo assim realizar a disponibilização deste mesmo conhecimento para determinada tarefa a ser realizada. (ALECRIM, 2007)

A RNA Conexionista tem o intuito de descrever um processo cognitivo parecido com o de um computador, estudando a mente de uma maneira computacional. (ELLIS, 2009)

Uma rede conexionista é formada, basicamente, por unidades (numa analogia com nossos neurônios) interligadas umas às outras através de conexões (analogia com sinapses), um enfoque, portanto, fortemente apoiado na neurofisiologia humana. O processamento dessa rede é totalmente distribuído e em paralelo (daí o termo sinônimo de rede PDP, ou seja, processamento distribuído em paralelo). Modelos conexionistas, com uma arquitetura interna como a descrita acima, são construídos na forma de programas de computador. Uma vez em funcionamento, o sistema estabelece processos de regulação entre as unidades que compõem a rede (na forma de ajuste das forças/pesos de conexão), provocando, assim, a estruturação do sistema. O ‘conhecimento’ da rede está justamente nas forças/pesos definidos localmente entre as unidades (i.e., através da constante ativação ou inibição destas). Assim, surge um desempenho baseado em padrões de ativação entre unidades da rede, padrões estes que estabelecem configurações específicas a cada ativação. (ELLIS, 2009 *apud* FERREIRA JUNIOR, 2005, P.238)

Os primeiros estudos realizados na área de RNA Conexionista datam de 1943, com a publicação dos trabalhos dos americanos Warren McCulloch e Walter Pitts, que desenvolveram o primeiro neurônio composto por um modelo matemático, baseados no neurônio biológico. (BITTENCOURT, 2009)

3.3.1 – Algoritmo de Retro-propagação

O algoritmo de retropropagação tem como objetivo treinar redes neurais com várias camadas. É o algoritmo mais conhecido entre os estudiosos, o qual se utiliza de um aprendizado supervisionado através de correções dos erros obtidos no aprendizado. (MENDES, 2007; SILVA OLIVEIRA, 2007)

As duas etapas do algoritmo de retro-propagação são as seguintes:

1º - Propagação: Primeiramente são apresentados os valores (padrões) de entrada. Assim, serão calculados os pesos em relação aos valores de entrada (chegam aos nós este somatório) para que possa ser formada a camada intermediária. Após, novamente se calcula os pesos em relação aos valores da camada intermediária (chegam aos nós este somatório) para que possa ser criada finalmente a camada de saída, onde se obtém a resposta e o erro poderá ser calculado. (MENDES, 2007; SILVA OLIVEIRA, 2007)

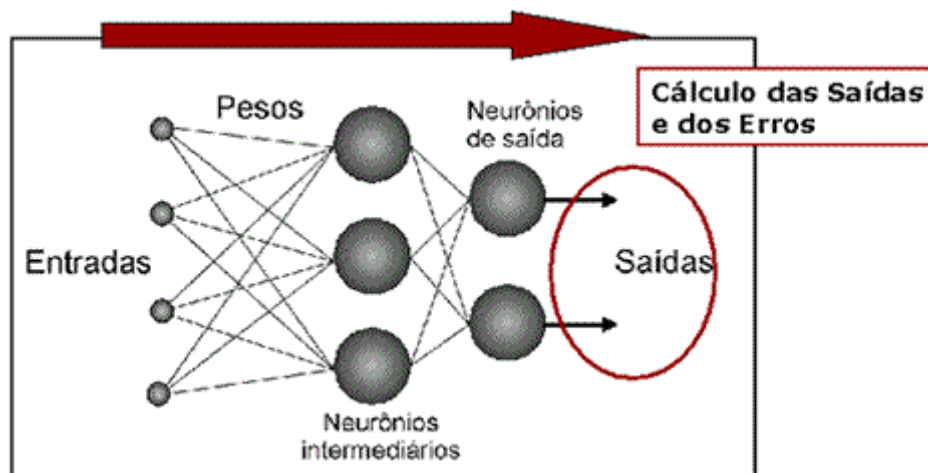


Figura 9: Cálculo das saídas e dos erros

Fonte: (<http://www.lncc.br/~labinfo/tutorialRN/imagens/esquemaFaseForward.gif>)

2º - Retro-propagação: É o caminho inverso da propagação, onde os pesos serão recalculados desde a camada de saída até a camada de entrada.

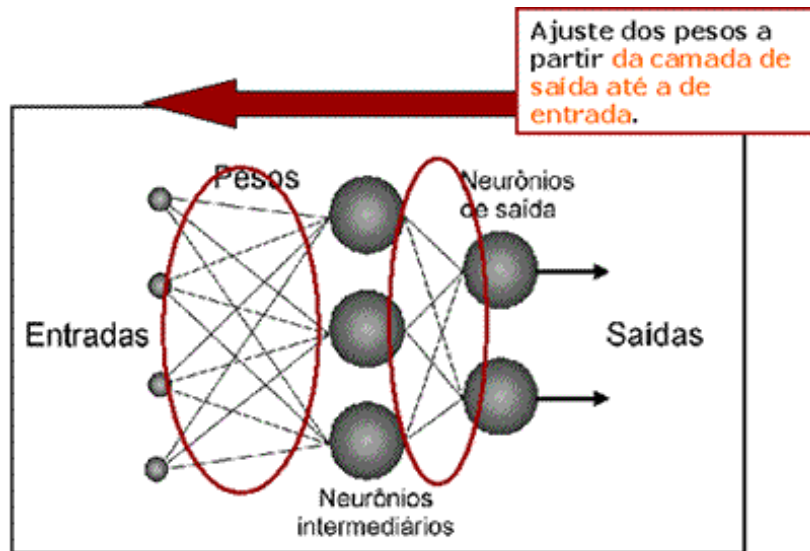


Figura 10: Ajuste dos pesos

Fonte: (<http://www.lncc.br/~labinfo/tutorialRN/imagens/esquemaFaseBackward.gif>)

Na fase de treinamento devemos apresentar os valores que desejamos para a camada de entrada da RNA e os valores que queremos como resposta na camada de saída. Assim, esses valores poderão ser comparados na camada de saída, obtendo um erro global, o qual irá influenciar na correção dos pesos na etapa de retropropagação. Deve-se ter em conta que este tipo de rede não fornece soluções ótimas, porém satisfatórias para o treinamento de neurônios de várias camadas. (MENDES, 2007; SILVA OLIVEIRA, 2007)

As RNAs possuem duas fases: treinamento e produção, sendo esta última utilizada para o fim com que foram projetadas.

4 - Implementação

Para treinar a RNA foram utilizados dados (valores de abertura, fechamento, máximo e mínimo) extraídos de um gráfico diário das ações preferenciais da Petrobras da Bolsa de Valores, ao decorrer de três anos (2006 - 2008), gráfico este retirado do site ADVFN (<http://br.advfn.com/>) e representado na Figura 11.



Figura 11: Gráfico de Velas da Bolsa de Valores
Fonte: (<http://br.advfn.com/>)

A estrutura lógica da rede contém inicialmente as seguintes etapas:

1. Gerar pesos aleatórios (intervalos reais de -1 a 1)
2. Calcular os NETs das camadas intermediárias (somatório das multiplicações dos nós de entrada em relação aos pesos)
3. Calcular as saídas das camadas intermediárias (utilizando funções de saída do tipo Sigmoidal ou Hiperbólica)
4. Calcular os NETs da camada de saída (somatório das multiplicações dos nós de saída das camadas intermediárias em relação aos pesos)
5. Calcular as saídas da camada de saída (utilizando funções de saída do tipo Sigmoidal ou Hiperbólica)
6. Calcular o vetor de erro (diferença entre o resultado esperado e o obtido)
7. Calcular o erro global
8. Caso atingido o mínimo necessário: Fim
9. Atualizar os pesos da camada de saída

10. Atualizar os pesos das camadas intermediárias
11. Voltar ao passo 2

4.1 - Interface e Características

A Interface do programa implementado está sintetizada na Figura 12. A seguir um texto descrevendo as funções sintetizadas do programa.

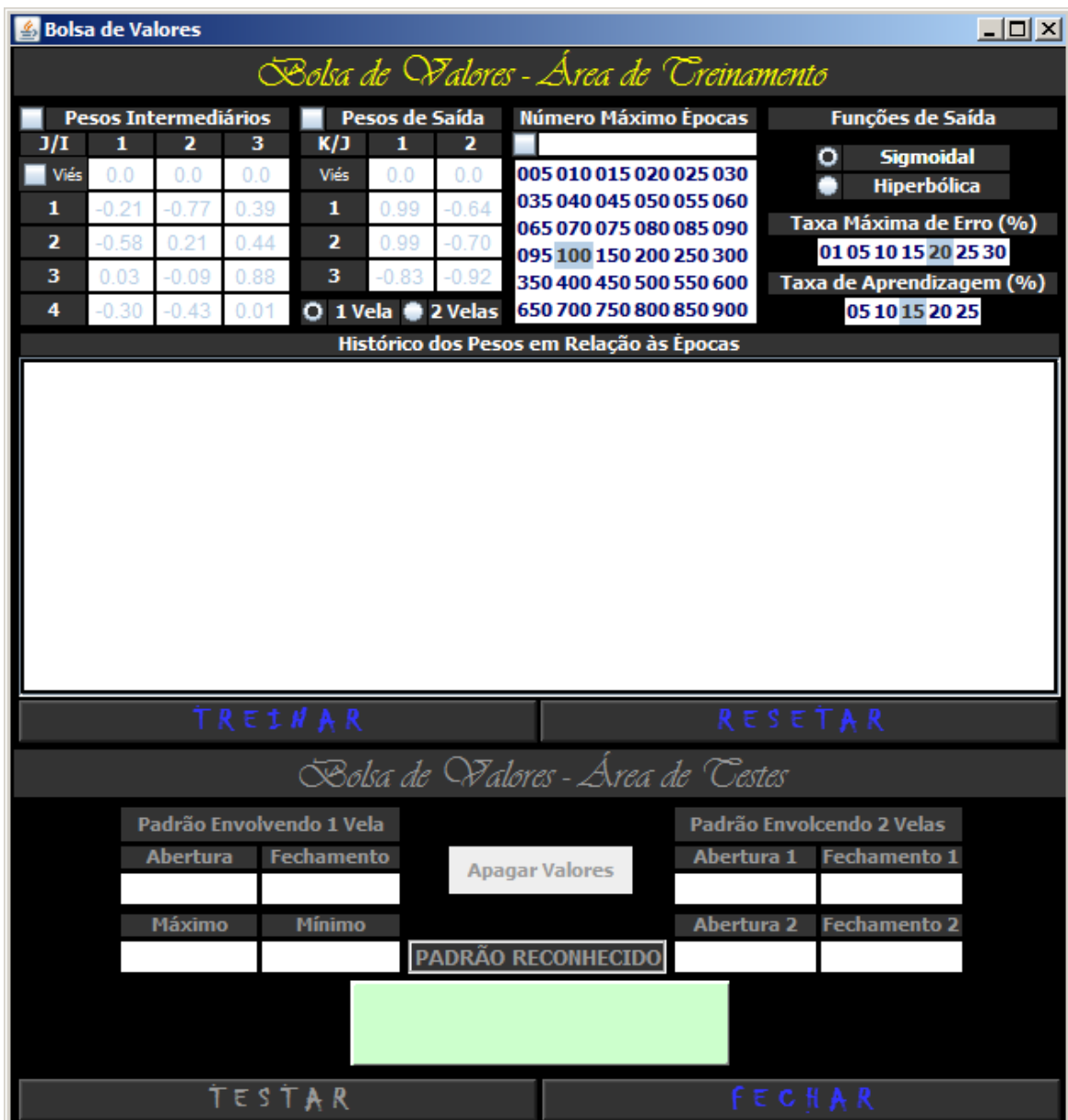


Figura 12: Interface do programa

- ❖ A parte superior (Bolsa de Valores – Área de Treinamento) do programa tem como objetivo programar e visualizar a RNA em si;
- ❖ A parte inferior (Bolsa de Valores – Área de Testes) tem como objetivo testar o programa para conferir se o mesmo foi corretamente treinado;
- ❖ Como default o programa inicia com pesos intermediários e de saída aleatórios (intervalo de -1 a 1);
- ❖ Como default os Pesos Viés são iguais a 0;
- ❖ Caso se queira treinar com os pesos Viés, basta clicar no checkBox “Viés”, e seus respectivos pesos serão gerados aleatoriamente no intervalo de -1 a +1;
- ❖ Como default a taxa máxima de erro vem setada como 20%;
- ❖ Como default a taxa de aprendizagem vem setada como 15%;
- ❖ Como default o número máximo de épocas vem setado como 100;
- ❖ Apenas será habilitada a “Área de Testes” após o devido treinamento do neurônio (após clicar no botão “TREINAR”);
- ❖ O programa permite escolher se os pesos intermediários, de saída e viés sejam números aleatórios ou se o usuário deseja escolher manualmente (caso deseje escolher manualmente, o intervalo varia de -99.99 a +99.99);
- ❖ O usuário pode escolher entre as seguintes taxas máximas de erro (%): 1, 5, 10, 15, 20, 25 e 30;
- ❖ O usuário pode escolher entre as seguintes taxas de aprendizagem (%): 5, 10, 15, 20, 25;
- ❖ Existe um quadro para se escolher o número máximo de épocas com números pré-definidos;
- ❖ Ao clicar no checkBox “Número Máximo Épocas”, é setado como default o número 1;
- ❖ Ao escolher os Pesos Intermediários, de Saída ou Viés manualmente, o programa valida, em tempo real, apenas números reais no intervalo de -99.99 a +99.99;
- ❖ Ao escolher o número máximo de épocas manualmente, o programa valida, em tempo real, apenas números naturais de até quatro dígitos (0 a 9999);
- ❖ Como default, o tipo de função de saída vem como “Sigmoidal”;
- ❖ O usuário tem a possibilidade de escolher entre dois tipos de funções de saída (Sigmoidal e Hiperbólica), através dos RadioButtons;
- ❖ Ao clicar no botão “TREINAR”, o programa irá recolher os dados de entrada (Pesos Intermediários, Pesos de Saída, Pesos Viés, Taxa Máxima de Erro, Taxa de

Aprendizagem, Época e o Tipo de Função de Saída), e irá executar o algoritmo de treinamento;

- ❖ Ao finalizar o treinamento, aparecerá no campo “Histórico dos Pesos em Relação às Épocas” o histórico dos pesos de cada época;
- ❖ Caso se acione o botão “TREINAR” e algumas das entradas não forem preenchidas, aparecerão as devidas mensagens de advertência;
- ❖ O botão RESETAR, ao ser pressionado, setará a interface da mesma maneira como quando o programa foi iniciado, atribuindo outros conjuntos de pesos aleatórios às duas camadas de nós da Rede;
- ❖ Por default, o RadioButton com padrão envolvendo uma vela vem setado como true;
- ❖ Na Área de Treinamento, o usuário pode optar em treinar o neurônio usando um padrão envolvendo uma ou duas velas, através dos RadioButtons;
- ❖ O programa valida em tempo real (apenas números reais positivos no intervalo de 0 a 99.99), ao serem pressionados, todos os campos jTextField da Área de Testes;
- ❖ Ao clicar no botão TESTAR, o programa recolhe os dados dos campos de entrada da Área de Testes, mostrando o resultado no jLabel abaixo do PADRÃO RECONHECIDO;
- ❖ Caso alguns dos campos de entrada da Área de Testes não forem preenchidos, o programa emitirá as possíveis mensagens de advertência;
- ❖ O botão FECHAR serve para encerrar o programa.

4.2 – Funcionamento do Programa

Foi implementada uma interface para interagir com o usuário. A interface já vem setada com todos os valores necessários para o treinamento da RNA, bastando o usuário pressionar o botão TREINAR. Esses valores foram previamente descritos no item Interface e Características. O usuário também tem a opção, caso queira, de preencher manualmente os valores, de acordo com os limites do mesmo.

Com os devidos dados de entrada corretamente preenchidos e demais requisitos, o usuário poderá treinar a RNA. Após seu treinamento é habilitada a área de testes para que se possa testar a RNA previamente treinada. Também é disponibilizada ao usuário a visualização do histórico dos pesos gerados durante cada época de treinamento, assim como a informação se a RNA foi devidamente treinada de acordo com a taxa máxima de erro escolhida pelo usuário.

Através da área de testes do programa, o usuário digita os valores de determina vela do gráfico de velas, a RNA então deverá ser capaz de reconhecer os padrões treinados.

4.2.1 – Padrão envolvendo uma vela (Martelo e Enforcado)

No caso de padrões envolvendo uma vela, a RNA é treinada para reconhecer apenas o indicador de reversão martelo e enforcado. Em outras palavras, para os cinco padrões que entraram na RNA ela dirá quais são martelos e enforcados e quais não são. Na implementação para reconhecimento deste padrão, foram extraídos cinco padrões do gráfico das ações da Petrobras do ano de 2006, e outros cinco padrões que não correspondem ao padrão martelo e enforcado.

Na Figura 12 se pode verificar a configuração da RNA correspondente a padrões envolvendo uma vela com os quatro valores dos nós de entrada, os três nós ocultos (intermediários) e os dois nós de saída (x e y).

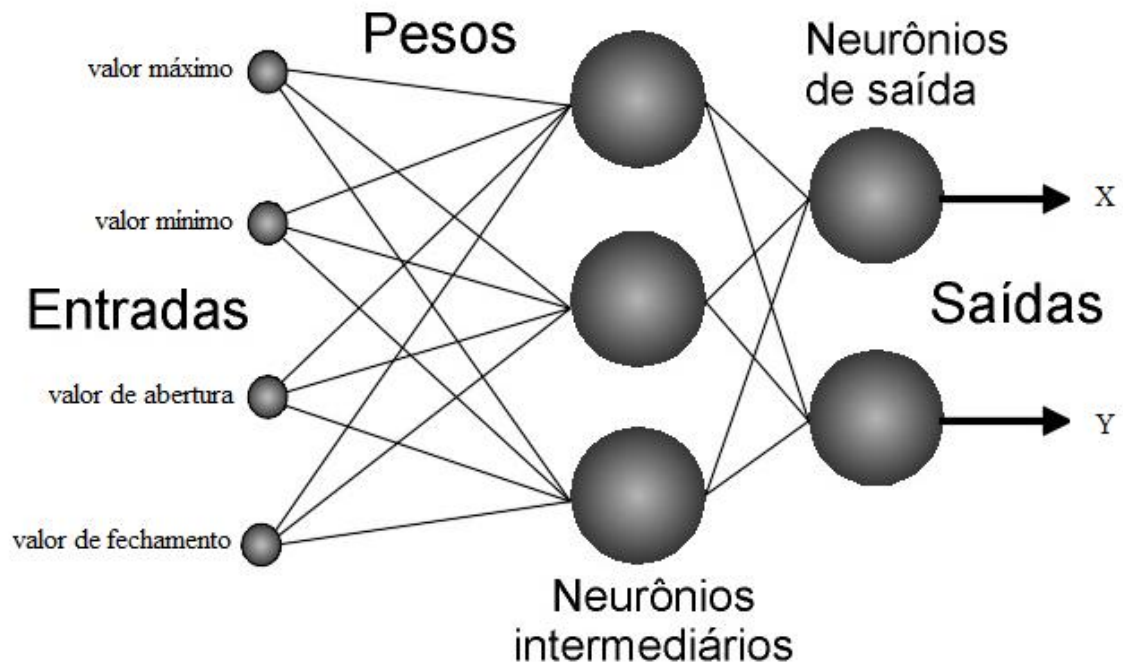


Figura 13: Configuração da RNA para padrões envolvendo uma vela

Os valores de entrada do padrão martelo e enforcado são os seguintes: valor de abertura, valor de fechamento, valor máximo e valor mínimo, exemplificados na Tabela 1.

PADRÃO	ABERTURA	MÁXIMO	MÍNIMO	FECHAMENTO
1	21,95	21,96	21,30	21,90
2	21,71	21,71	21,31	21,68
3	21,92	21,93	20,95	21,82
4	20,48	20,55	20,19	20,56
5	19,95	20,00	19,68	20,07

Tabela 1: Valores (em reais) dos cinco padrões martelo e enforcado

4.2.1.1 – Gráficos do Padrão Martelo e Enforcado

A seguir as Figuras representando os padrões martelo e enforcado cujos valores foram extraídos para implementação da RNA, juntamente com suas explicações:

Nas Figuras 14, 15 e 16 estão representados o padrão enforcado, revertendo o mercado para uma tendência de baixa. Na Figura 17 está representado o padrão

enforcado, revertendo o mercado para uma tendência de baixa e o padrão martelo, revertendo o mercado novamente para uma tendência de alta.



Figura 14: Padrão 1 (Enforcado)

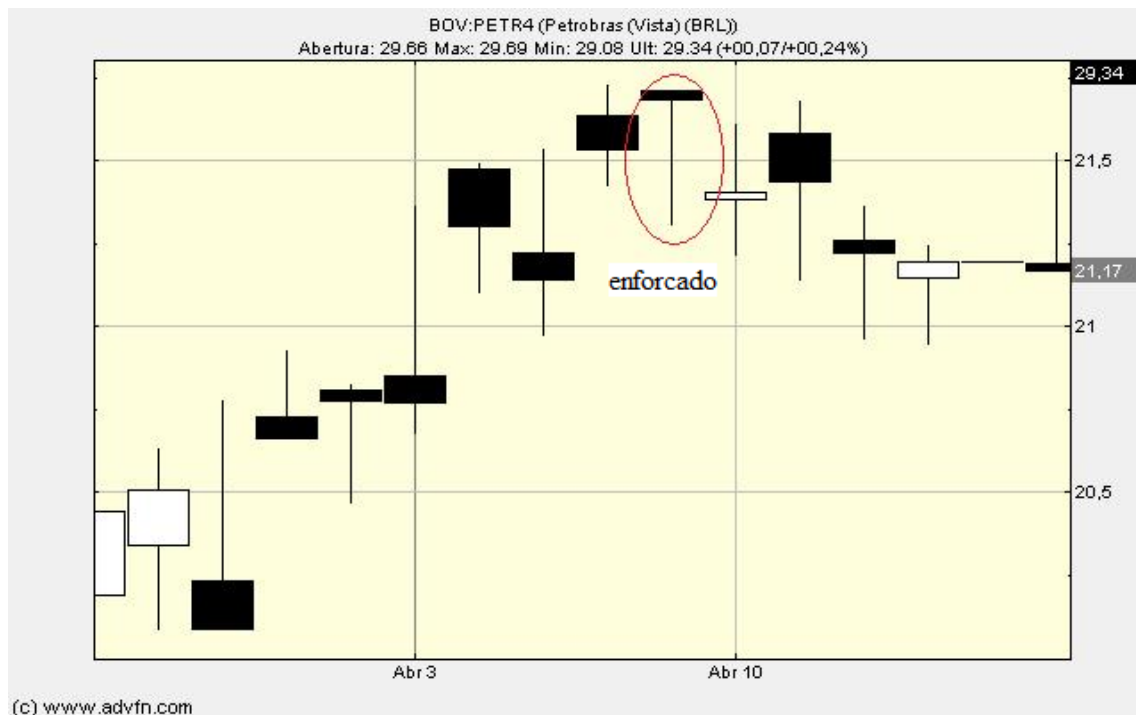


Figura 15: Padrão 2 (Enforcado)

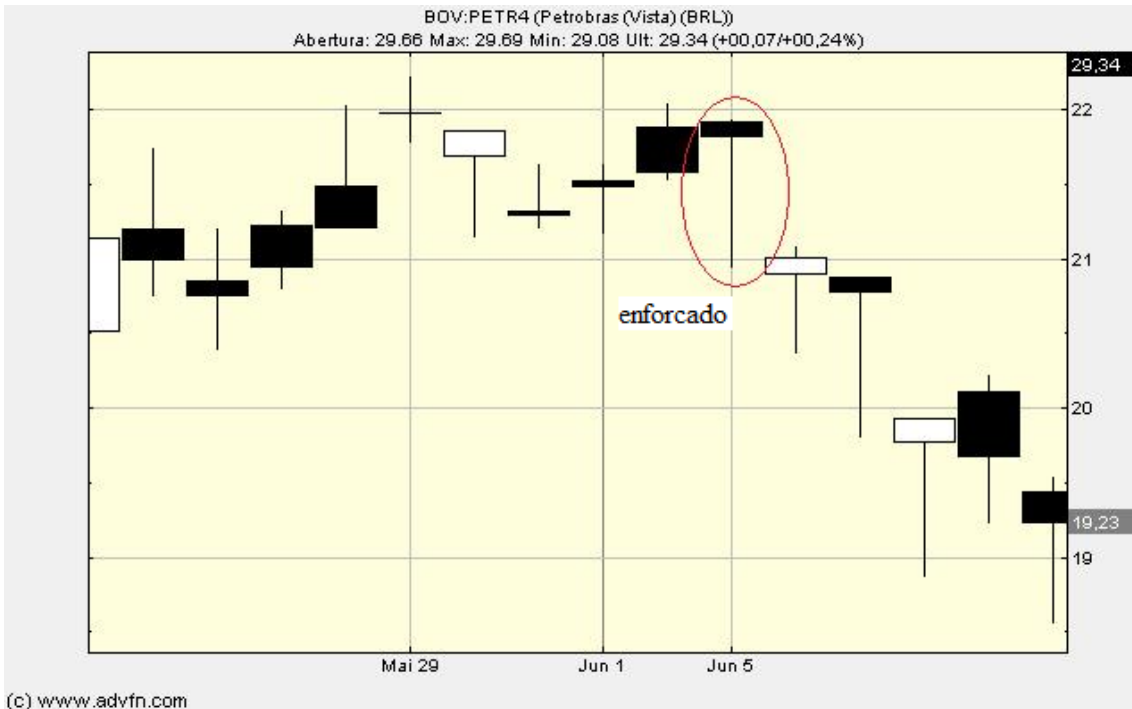


Figura 16: Padrão 3 (Enforcado)

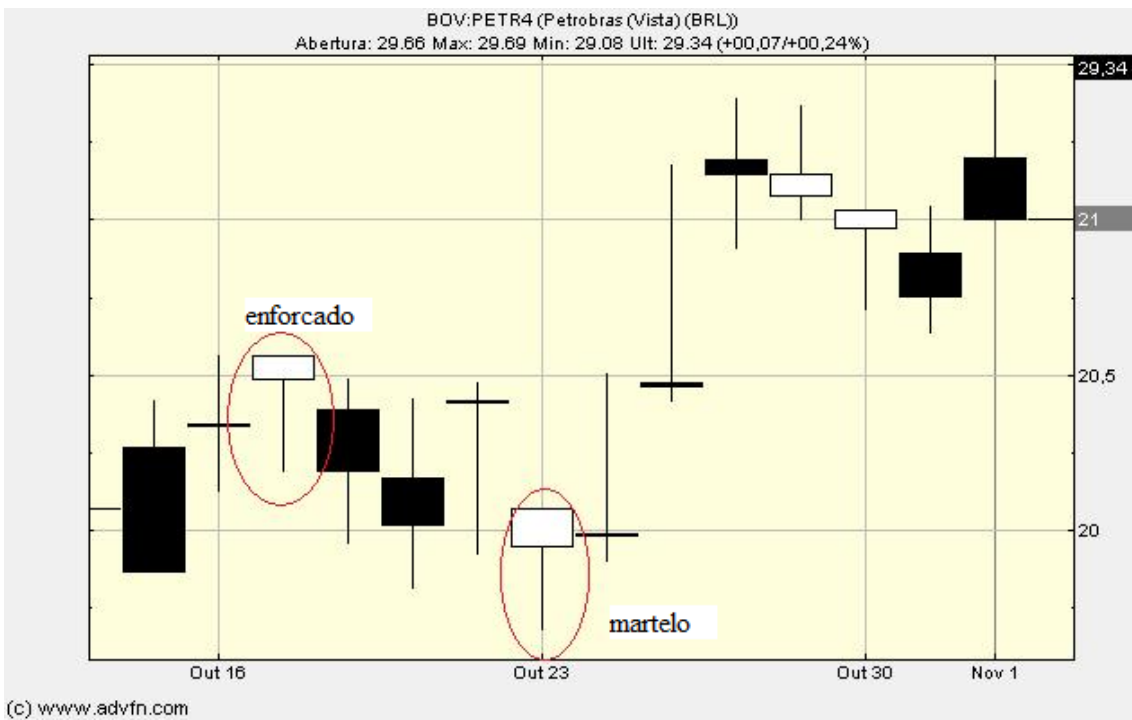


Figura 17: Padrões 4 e 5 (Martelo e Enforcado)

4.4.2 – Padrão envolvendo duas velas (Padrão Envolvente e Linhas Harami)

No caso de padrões envolvendo duas velas, a RNA é treinada para reconhecer os indicadores de reversão padrão envolvente e linhas harami. Na implementação para reconhecimento destes padrões, foram extraídos cinco padrões do tipo padrão envolvente do gráfico de ações da Petrobras entre os anos de 2006 a 2008 e cinco padrões do tipo linhas harami do gráfico de ações da Petrobras entre os anos de 2005 a 2008.

Na Figura 18 se pode verificar a configuração da RNA correspondente a padrões envolvendo duas velas com os quatro valores dos nós de entrada, os três nós ocultos (intermediários) e os dois nós de saída (x e y).

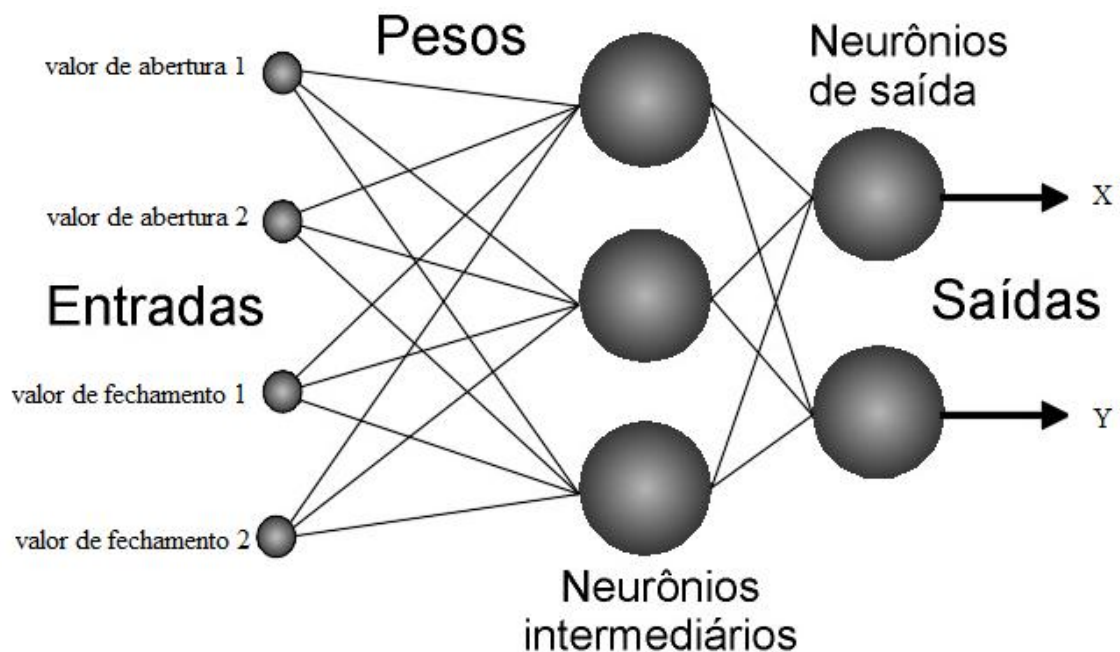


Figura 18: Configuração da RNA para padrões envolvendo duas velas

Os valores de entrada dos padrões padrão envolvente e linhas harami são os seguintes: valores de abertura e fechamento da primeira vela e valores de abertura e fechamento da segunda vela, representados na Tabela 2 e Tabela 3 respectivamente.

PADRÃO ENVOLVENTE	ABERTURA 1	FECHAMENTO 1	ABERTURA 2	FECHAMENTO 2
1	23,44	23,51	23,53	23,42
2	24,00	24,32	24,37	23,87
3	23,47	23,75	23,90	23,22
4	45,50	49,09	46,34	43,98
5	30,59	29,80	28,50	31,30

Tabela 2: Valores (em reais) dos cinco padrões padrão envolvente

HARAMI	ABERTURA 1	FECHAMENTO 1	ABERTURA 2	FECHAMENTO 2
1	23,25	22,63	22,75	23,15
2	23,39	23,65	23,59	23,50
3	22,65	22,18	22,40	22,61
4	23,80	25,10	24,99	24,25
5	18,89	20,52	20,17	19,95

Tabela 3: Valores (em reais) dos cinco padrões linhas harami

4.4.2.1 – Gráficos do Padrão Envolvente e Linhas Harami

Nas Figuras 19, 20, 21 e 22 estão representados o padrão envolvente altista, revertendo o mercado para uma tendência de baixa. Na Figura 23 está representado o padrão envolvente baixista, revertendo o mercado para uma tendência de alta.

Na Figura 24 está representado o padrão linhas harami baixista, revertendo o mercado para uma tendência de alta e o padrão linhas harami altista, revertendo o mercado para uma tendência de baixa. Na Figura 25 está representado o padrão linhas harami baixista, revertendo o mercado para uma tendência de alta. nas Figuras 26 e 27 estão representados o padrão linhas harami altista, revertendo o mercado para uma tendência de baixa.

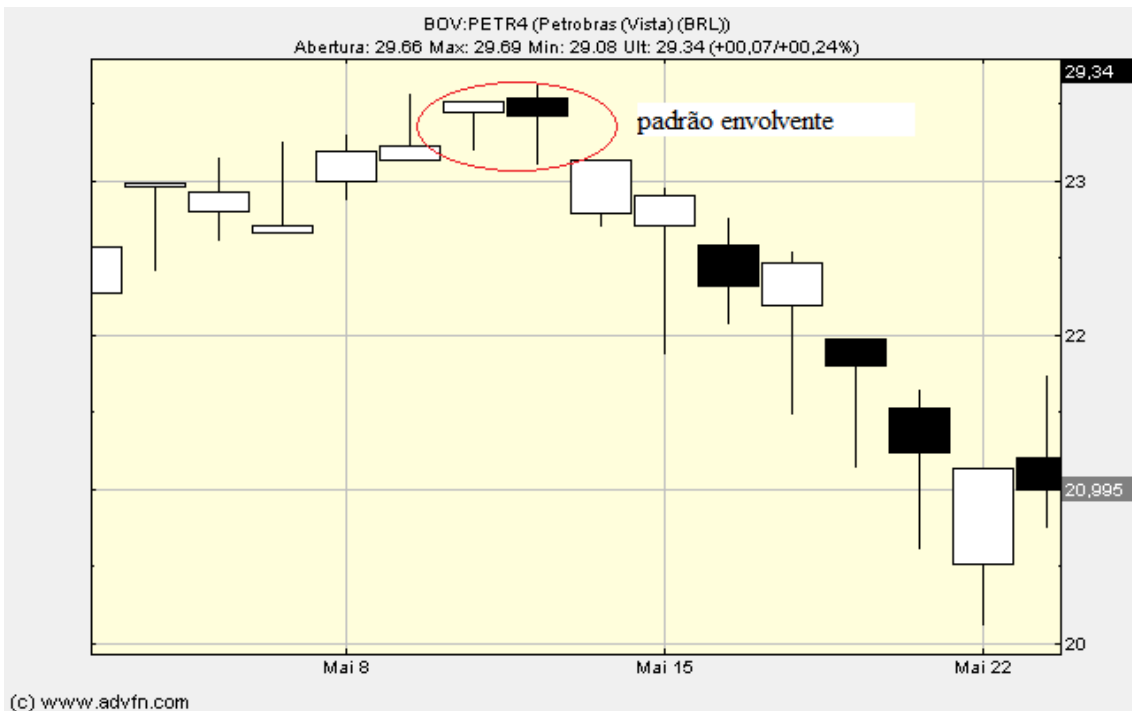


Figura 19: Padrão 1 (Padrão Envolvente Altista)

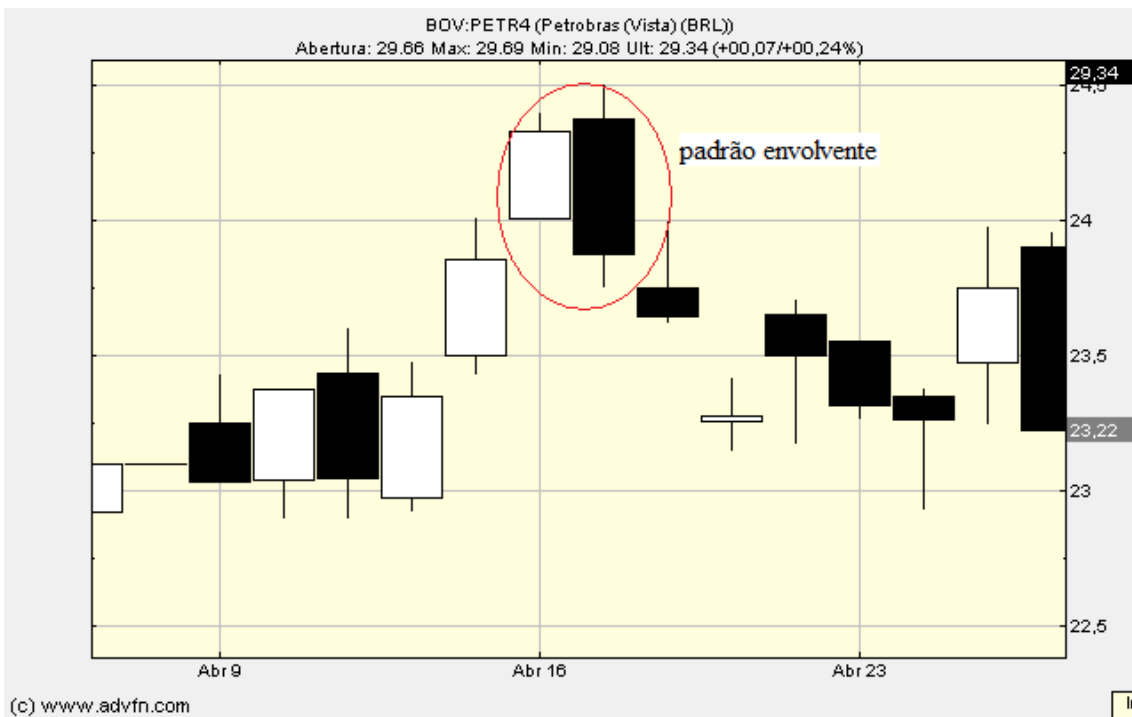


Figura 20: Padrão 2 (Padrão Envolvente Altista)

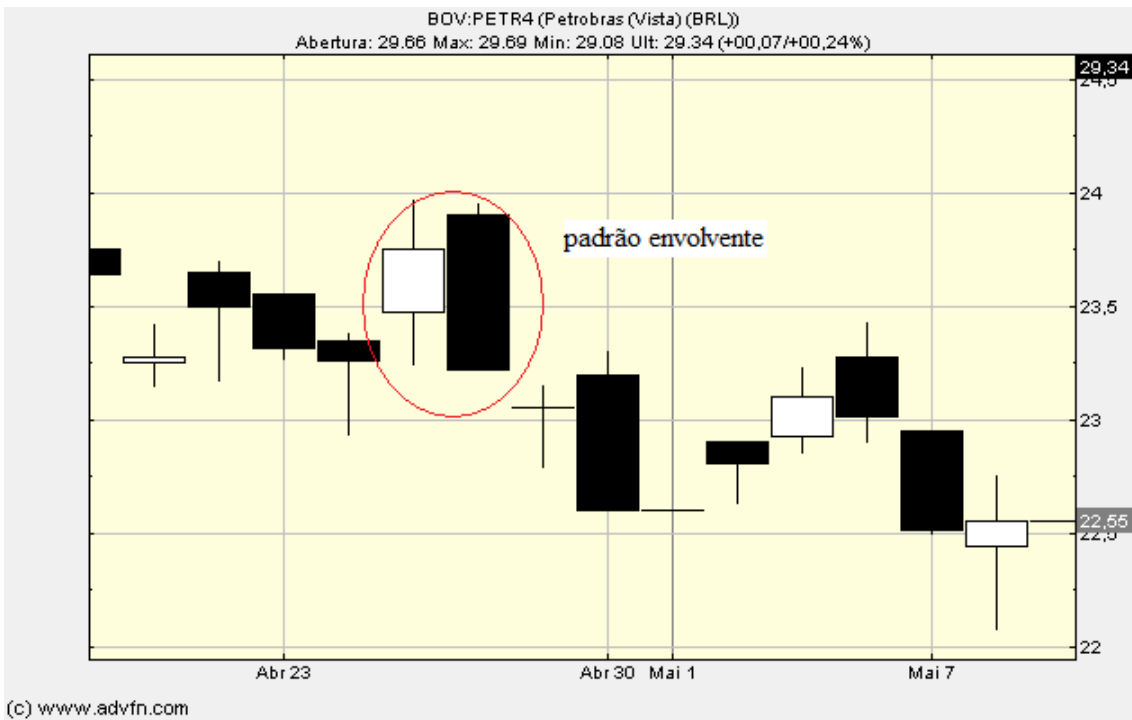


Figura 21: Padrão 3 (Padrão Envolvente Altista)

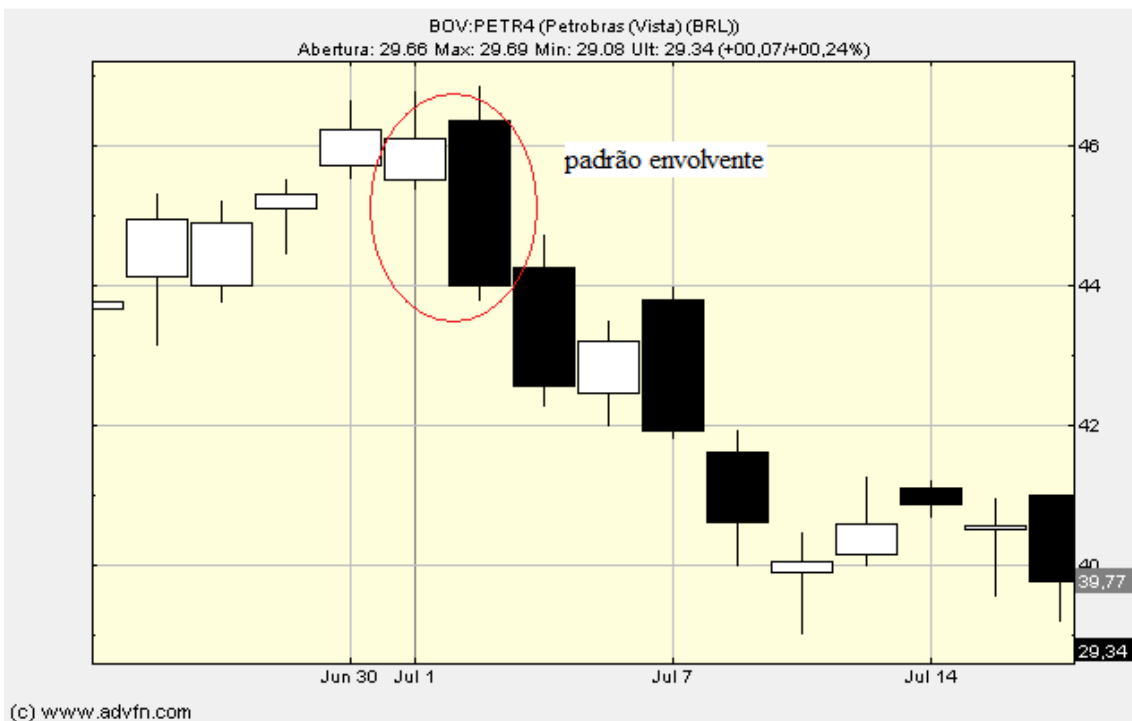


Figura 22: Padrão 4 (Padrão Envolvente Altista)

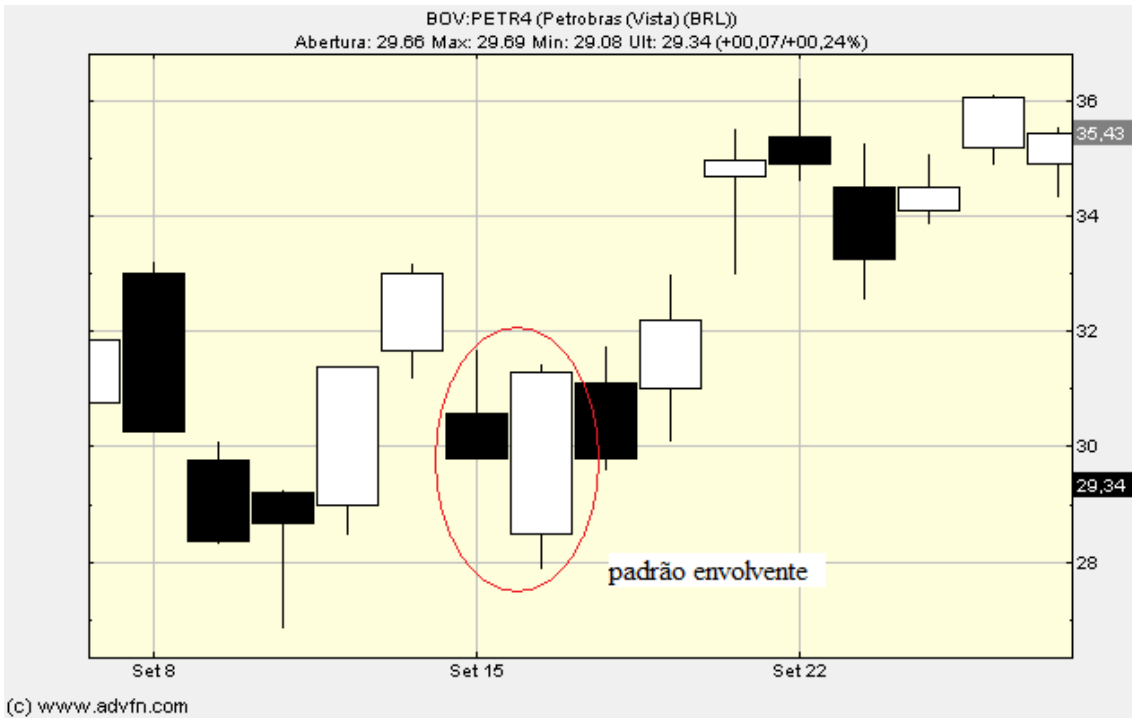


Figura 23: Padrão 5 (Padrão Envolvente Baixista)

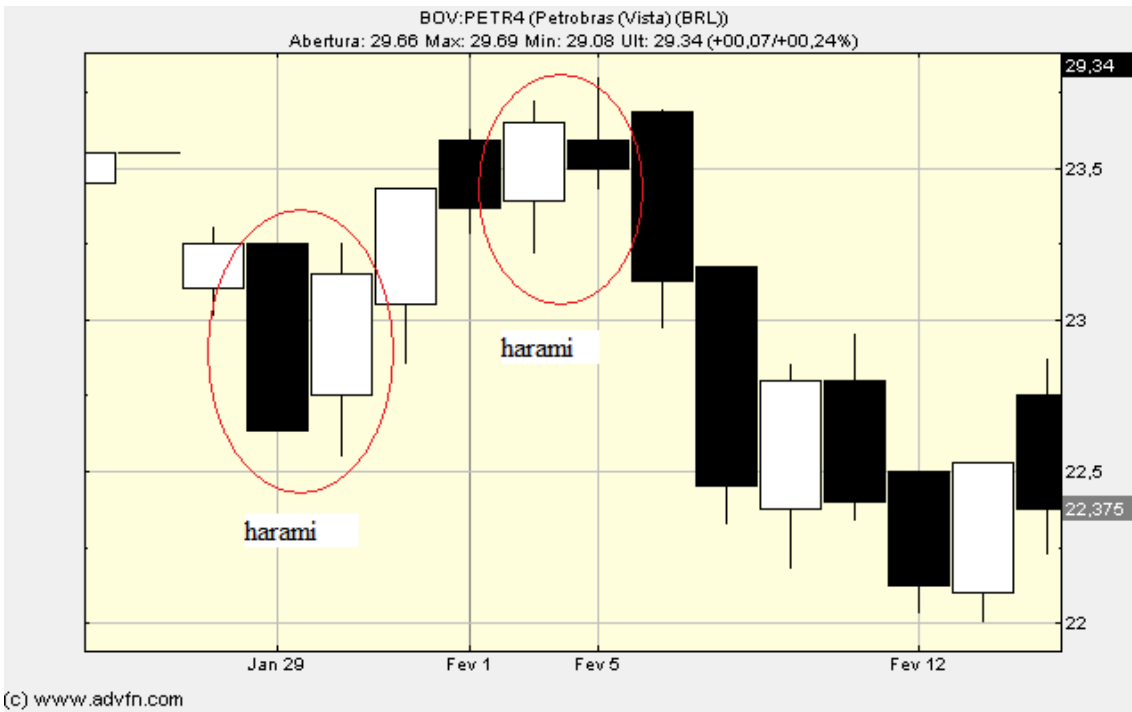


Figura 24: Padrões 1 e 2 (Linhas Harami Baixista e Altista)

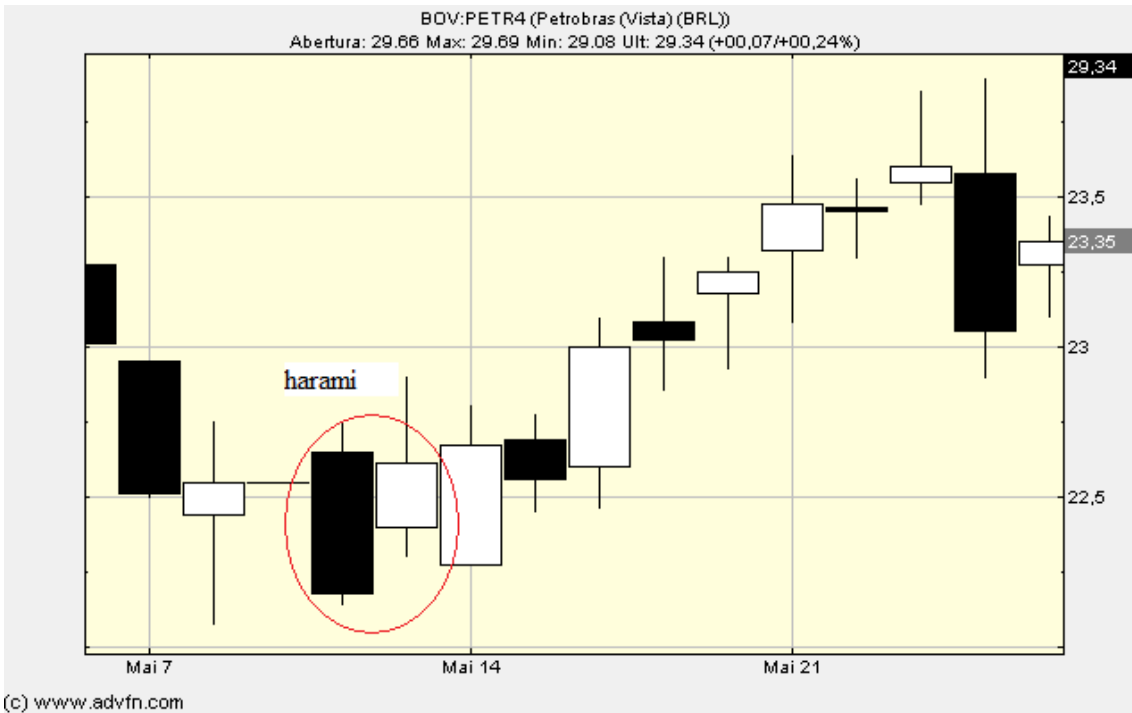


Figura 25: Padrão 3 (Linhas Harami Baixista)

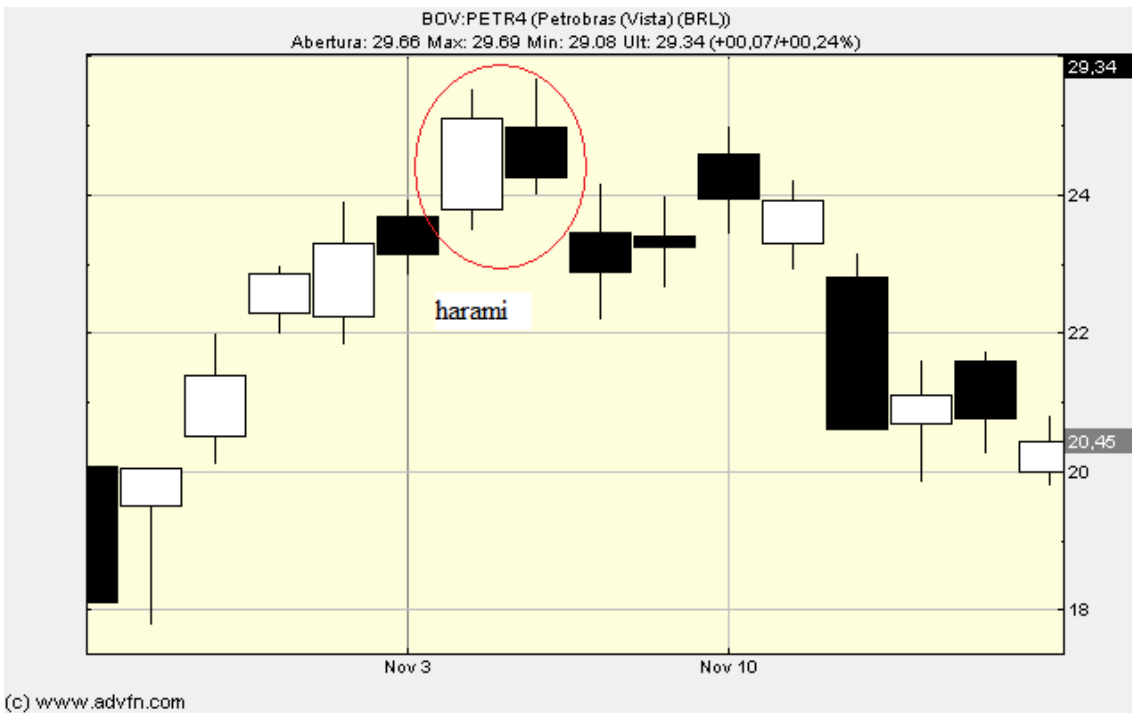


Figura 26: Padrão 4 (Linhas Harami Altista)

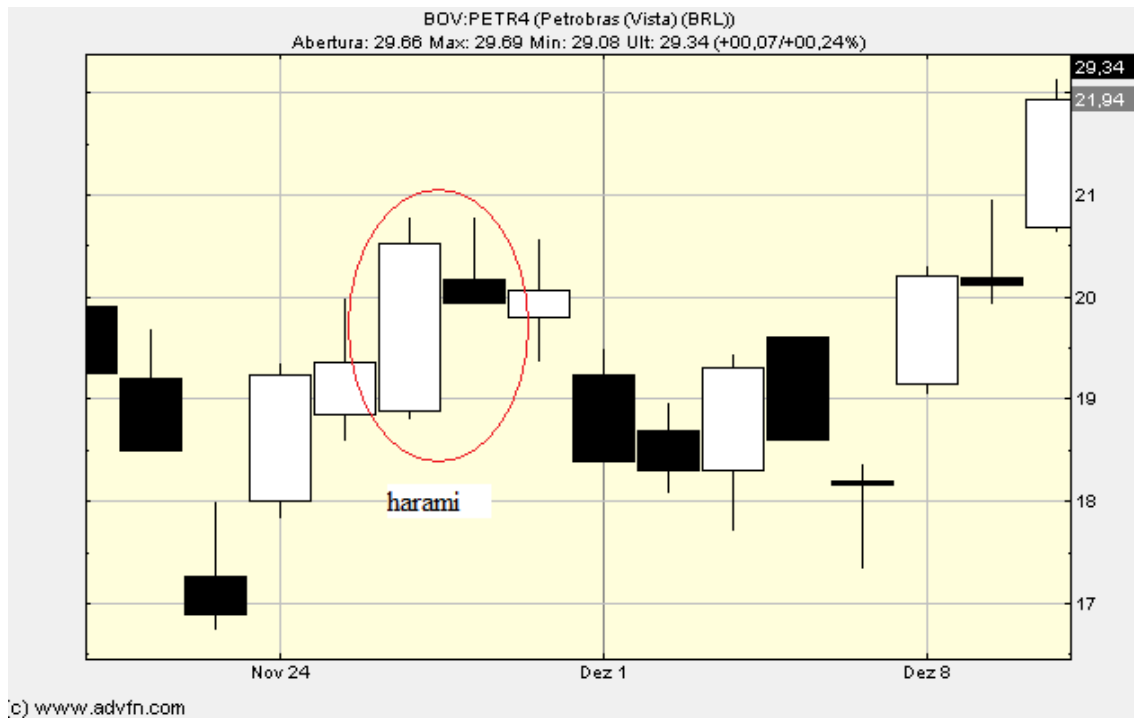


Figura 27: Padrão 5 (Linhas Harami Altista)

4.4.3 – Implementação para o Treinamento da Rede Neural Artificial

Para a implementação do treinamento da RNA, foi utilizado o algoritmo de retro-propagação. Este algoritmo considera o uso de quatro nós de entrada, três nós intermediários e dois nós de saída. Há opção de inclusão do nó de viés, conforme discutido no item 4.1.

Ao final de cada época, a RNA compara a taxa de erro global com a taxa máxima de erro escolhida pelo usuário. Caso a taxa de erro máxima escolhida pelo usuário seja igual ou menor à taxa de erro global, o treinamento é concluído. O treinamento é concluído com sucesso quando a taxa de erro máxima é menor ou igual à taxa de erro global, antes de completar o máximo (ou exatamente igual) número de épocas escolhidas pelo usuário. Caso contrário, a RNA é treinada, porém sem sucesso.

O programa implementado tem as funções principais disponibilizadas no anexo. São elas: parte de teste da RNA, onde o usuário vai colocar os valores correspondentes ao padrão que se deseja realizar o teste. Se a RNA reconhecer o padrão em questão, irá imprimir no programa o tipo de padrão reconhecido. Caso não reconheça, irá informar que este padrão não foi reconhecido. Parte de treinamento, onde a RNA vai pegar todos os valores correspondentes ao treinamento na interface. Com isto, geram-se os

históricos dos pesos em relação a cada época de treinamento e é disponibilizado ao usuário se a RNA foi treinada com sucesso, ou não, de acordo com a porcentagem de erro máxima escolhida pelo usuário.

4.4.4 – Implementação para o Teste da Rede Neural Artificial

Para testar o programa, é utilizada apenas a primeira parte do algoritmo de retro-propagação, os “passos para frente” do algoritmo, o qual consiste em calcular os valores dos NETs e das camadas de saída das camadas intermediárias, e calcular os NETs e os valores de saída das camadas de saída. Após a finalização do algoritmo, serão comparadas as taxas de erro dos valores colocados para testar a RNA.

Caso estas taxas de erro estejam dentro do reconhecimento de um padrão treinado, então a RNA irá reconhecer o padrão, caso contrário, o padrão não será reconhecido. O reconhecimento dos padrões depende da taxa máxima de erro conseguida no final de um treinamento.

4.2 – Mensagens de Erro

A seguir as figuras representando todas as mensagens de erros existentes no programa implementado e suas descrições:

- ❖ Figura 28: esta figura representa quando o usuário optou por preencher manualmente os pesos intermediários, viés, de saída e a época, sendo que não os preencheu por completo.
- ❖ Figura 29: esta figura representa quando o usuário optou por preencher manualmente os pesos intermediários, viés, de saída, sendo que não os preencheu por completo.
- ❖ Figura 30: esta figura representa quando o usuário optou por preencher manualmente os pesos intermediários, viés e a época, sendo que não os preencheu por completo.
- ❖ Figura 31: esta figura representa quando o usuário optou por preencher manualmente os pesos intermediários e de saída e a época, sendo que não os preencheu por completo.

- ❖ Figura 32: esta figura representa quando o usuário optou por preencher manualmente os pesos viés, de saída e a época, sendo que não os preencheu por completo.
- ❖ Figura 33: esta figura representa quando o usuário optou por preencher manualmente os pesos intermediários e viés, sendo que não os preencheu por completo.
- ❖ Figura 34: esta figura representa quando o usuário optou por preencher manualmente os pesos intermediários e a época, sendo que não os preencheu por completo.
- ❖ Figura 35: esta figura representa quando o usuário optou por preencher manualmente os pesos intermediários e de saída, sendo que não os preencheu por completo.
- ❖ Figura 36: esta figura representa quando o usuário optou por preencher manualmente os pesos viés e de saída, sendo que não os preencheu por completo.
- ❖ Figura 37: esta figura representa quando o usuário optou por preencher manualmente os pesos viés e a época, sendo que não os preencheu por completo.
- ❖ Figura 38: esta figura representa quando o usuário optou por preencher manualmente os pesos de saída e a época, sendo que não os preencheu por completo.
- ❖ Figura 39: esta figura representa quando o usuário optou por preencher manualmente os pesos intermediários, sendo que não os preencheu por completo.
- ❖ Figura 40: esta figura representa quando o usuário optou por preencher manualmente os pesos de saída, sendo que não os preencheu por completo.
- ❖ Figura 41: esta figura representa quando o usuário optou por preencher manualmente os pesos viés, sendo que não os preencheu por completo.
- ❖ Figura 42: esta figura representa quando o usuário optou por preencher manualmente a época, sendo que não a preencheu por completo.
- ❖ Figura 43: referente à área de testes, esta figura representa quando o usuário não preencheu por completo todas as quatro entradas requeridas para o teste da rede.

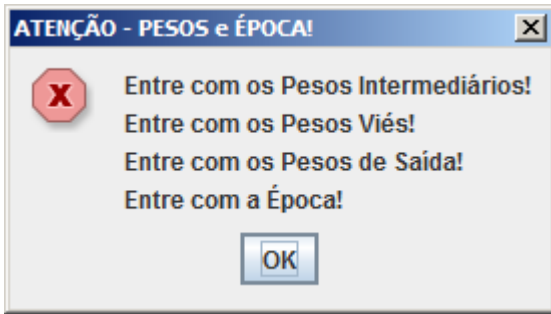


Figura 28: Mensagem de erro – Pesos (Intermediário, Viés, Saída) e Época

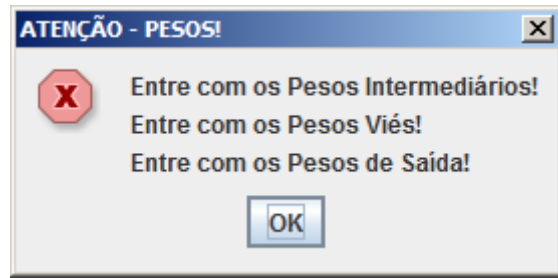


Figura 29: Mensagem de erro – Pesos (Intermediário, Viés, Saída)

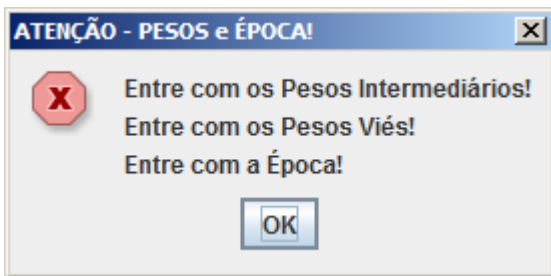


Figura 30: Mensagem de erro – Pesos (Intermediário, Viés) e Época

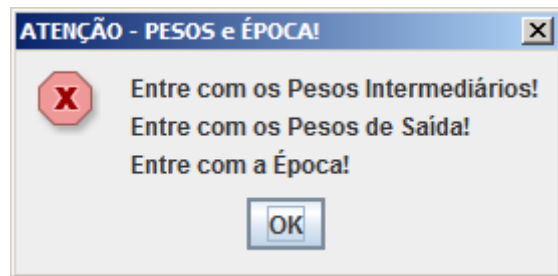


Figura 31: Mensagem de erro – Pesos (Intermediário, Saída) e Época

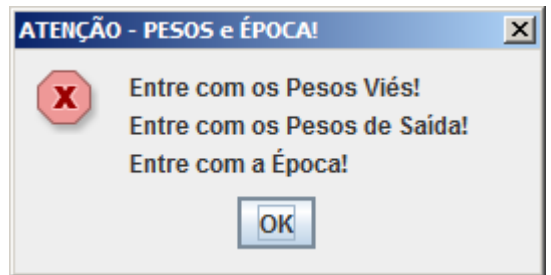


Figura 32: Mensagem de erro – Pesos (Viés, Saída) e Época

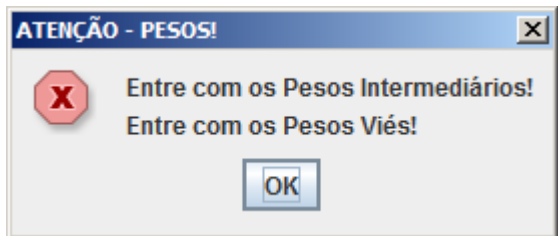


Figura 33: Mensagem de erro – Pesos (Intermediário, Viés)

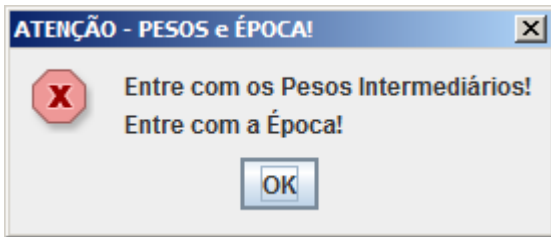


Figura 34: Mensagem de erro – Pesos (Intermediário) e Época

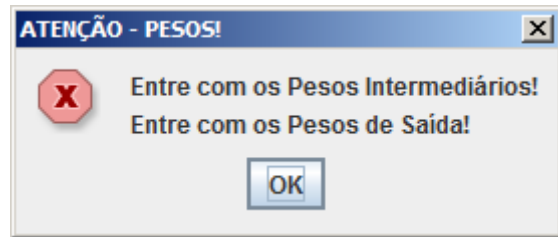


Figura 35: Mensagem de erro – Pesos (Intermediário, Saída)

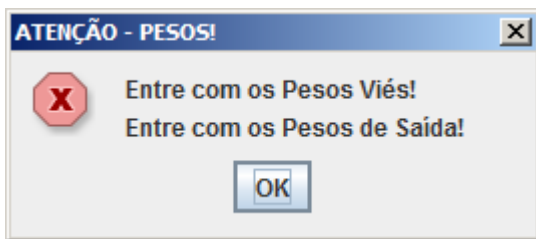


Figura 36: Mensagem de erro – Pesos (Viés, Saída)

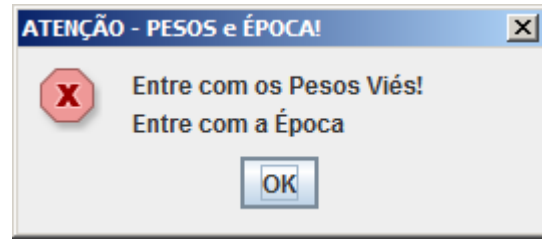


Figura 37: Mensagem de erro – Pesos (Viés) e Época

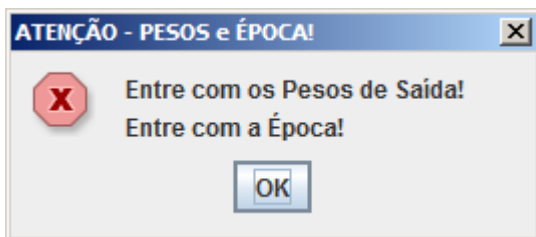


Figura 38: Mensagem de erro – Pesos (Saída) e Época

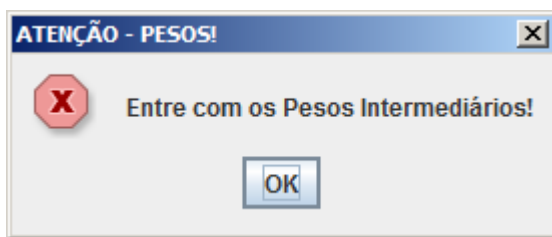


Figura 39: Mensagem de erro – Pesos (Intermediário)

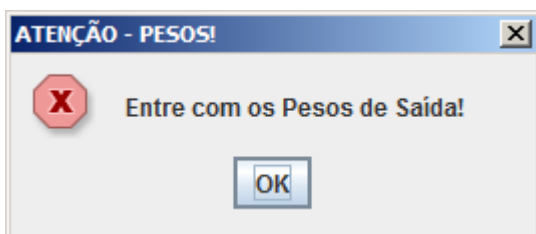


Figura 40: Mensagem de erro – Pesos (Saída)

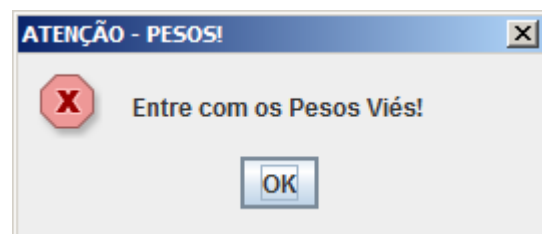


Figura 41: Mensagem de erro – Pesos (Viés)

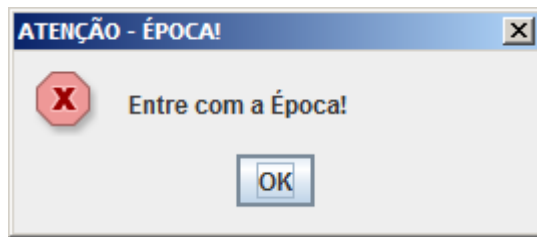


Figura 42: Mensagem de erro – Época

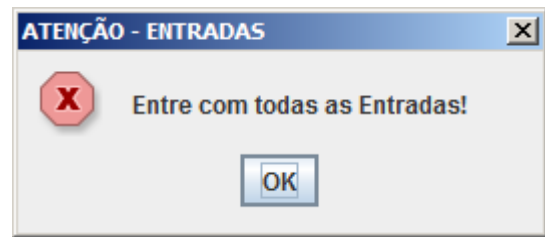


Figura 43: Mensagem de erro – Entradas

4.5 – Figuras do Programa Após Treinamento e Teste

A seguir as Figuras representando o treinamento completo da RNA com suas respectivas descrições:

- ❖ Figura 44: visualização da área de treinamento do programa onde observa-se todos os valores corretamente setados para o treinamento da RNA de padrões envolvendo duas velas e o histórico das épocas em relação aos pesos, mostrando um treinamento sem sucesso para o erro máximo de 20% escolhido para o treinamento.
- ❖ Figura 45: visualização da área de testes do programa onde observa-se os quatro valores de entrada preenchidos e reconhecidos corretamente pela RNA como sendo um padrão envolvente, mesmo com um erro máximo maior que 20%.
- ❖ Figura 46: visualização da área de testes do programa onde observa-se os quatro valores de entrada preenchidos e reconhecidos corretamente pela RNA como sendo um padrão não reconhecido, mesmo com um erro máximo maior que 20%.

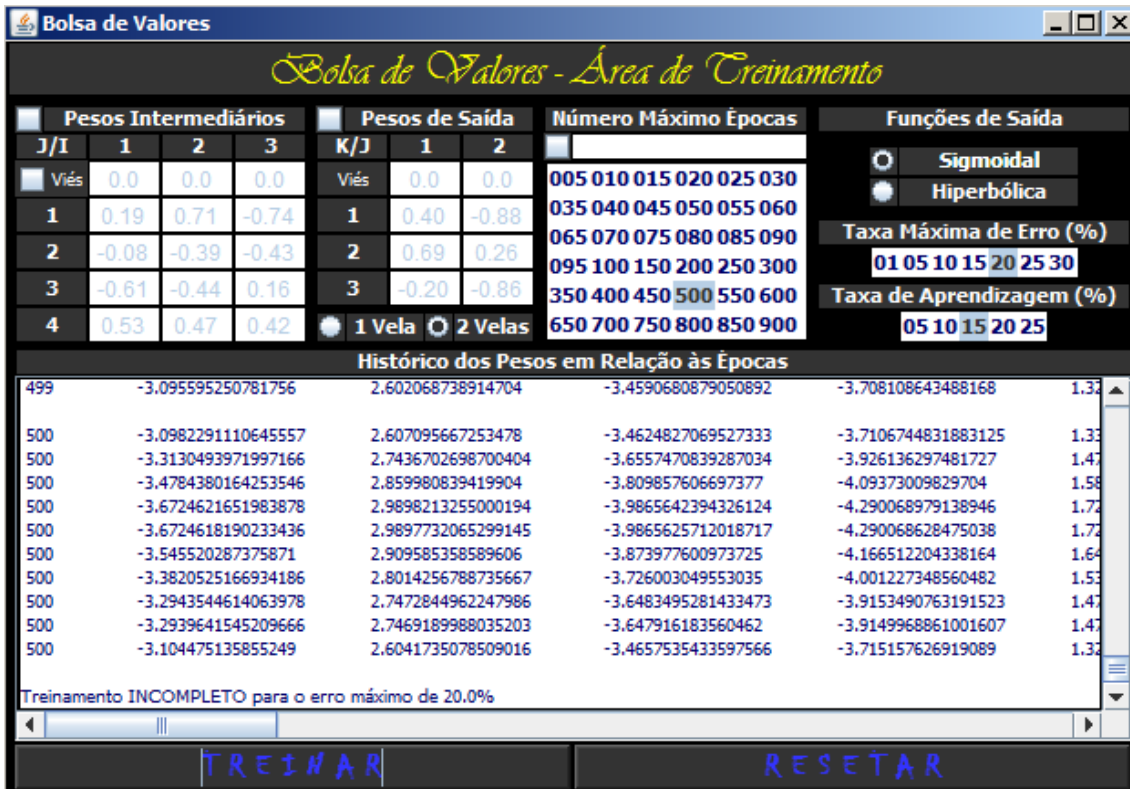


Figura 44: Conclusão do Treinamento do Neurônio Artificial



Figura 45: Teste de um padrão (Reconhecido)

Bolsa de Valores - Área de Testes

Padrão Envolvendo 1 Vela		Apagar Valores	Padrão Envolvendo 2 Velas	
Abertura	Fechamento		Abertura 1	Fechamento 1
			23	24
Máximo	Mínimo	PADRÃO RECONHECIDO	Abertura 2	Fechamento 2
			23	24

NÃO RECONHECIDO!

TESTAR FECHAR

Figura 46: Teste de um padrão (Não Reconhecido)

5 – Conclusão

Foi verificado que ao treinar o padrão martelo e enforcado, a RNA não atingiu a capacidade de baixar, em curto espaço de tempo (poucas épocas), a taxa máxima de erro para pelo menos 30%.

Já quando é treinado um padrão envolvendo duas velas, o padrão envolvente e linhas harami, a RNA consegue ter pelo menos 50% de taxa máxima de erro ao final de um treinamento de até 500 épocas. Porém ainda não é considerada uma boa taxa máxima de erro para reconhecimento de um padrão, tendo esta que ser de pelo menos 30%.

Com a mesma configuração da RNA adotada para o reconhecimento dos padrões de indicadores de reversão propostos nesse projeto, a RNA foi testada para treinar padrões envolvendo carros, motos e caminhões, com os seguintes valores de entrada: cilindros, cilindradas, peso e velocidade máxima. Os valores destas entradas são bem distantes entre si, e ao treinar a RNA com estes valores, se consegue um treinamento para uma taxa máxima de erro de 20%, com sucesso.

Considerando os fatos em questão, uma hipótese de não haver um total sucesso no treinamento da RNA para o correto reconhecimento dos padrões de indicadores de reversão propostos por esse projeto, é o fato desses valores serem muito aproximados entre si. Porém devemos levar em conta de que uma RNA possui suas limitações e dificilmente haverá 100% de acerto.

As dificuldades encontradas nesse projeto foram justamente o fato da proximidade dos valores de entradas dos padrões treinados na RNA, dificultando o sucesso no final de um treinamento. Foram realizados testes para a solução deste problema, como a mudança na configuração da RNA para duas camadas ocultas, uma única camada de saída, como também converter os valores de entrada para uma potência de 10, porém todas as tentativas sem sucesso.

Para trabalhos futuros, poderá ser avaliado com mais detalhes essa proximidade dos valores que impedem um melhor sucesso no treinamento da RNA como também tornar a RNA um pouco mais flexível, podendo o usuário escolher também a configuração da RNA, escolhendo a quantidade de camadas de entradas, camadas ocultas e de saída desejadas.

6 – Referências Bibliográficas

1. ALECRIM, Emerson. **Redes Neurais Artificiais**. Publicado em 09/05/2004. Documento em HTML disponível em <<http://www.infowester.com/redesneurais.php>>. Acesso em: 28 de Junho de 2007.
2. BASTTER.COM. **Aprendizado CandleStick: Martelos e Enforcados**. 2002. Documento em HTML disponível em: http://www.bastter.com/BR/MERCADO/Aprendizado/AnaliseTecnica/CandleSticks/martelos_e_enforcados.aspx. Acesso em: 19 de Abril de 2009.
3. BITTENCOURT, Guilherme. **Inteligência Computacional – Conexionismo**. Artigo HTML disponível em: <http://www.das.ufsc.br/gia/softcomp/node6.html>. Acesso em: 21 de Abril de 2009.
4. ELLIS, Nick. **Modelo Conexionista**. Documento digital em formato PDF disponível em <<http://www.veramenezes.com/conexionismo.pdf>>. Acesso em: 01 de março de 2009.
5. LIMA, Gerson. **A Lógica da Bolsa de Valores**. Documento em HTML disponível em <http://www.desempregozero.org.br/artigos/a_logica_da_bolsa_de_valores.php>. Acesso em: 28 de Junho de 2007.
6. MALHEIROS, Rivadavila S. **Análise Técnica: Fundamentos e Prática**. Realização: CMA Educacional, 2005.
7. MARQUES, Camila. **O que são ações, e como funciona a Bolsa de Valores**. Documento em HTML disponível em <http://www.imesexplica.com.br/1203bolsa_abre.asp>. Acesso em: 28 de Junho de 2007.
8. MENDES, Daniela Quintela. SILVA OLIVEIRA, Márcio Ferreira. **Tutorial de Redes Neurais**. Documento em HTML disponível em <http://www.lncc.br/~labinfo/tutorialRN/frm4_backpropagation.htm>. Acesso em: 28 de Junho de 2007.
9. NELOGICA. **Introdução a Análise Técnica**. Documento em HTML disponível em <<http://www.nelogica.com.br/tutoriais/introtec/introtec02.php>>. Acesso em: 01 de março de 2009.
10. NISON, Steve. **Curso de Análise Técnica Avançada Candlestick**. Realização: CMA Educacional. Resenha BM&F. 1992.
11. PORTAL DO INVESTIDOR, **O que é Bolsa de valores**. Documento em HTML disponível em <<http://www.portaldoinvestidor.gov.br/Acad%C3%AAmico/EntendendooMerca>>

dodeValoresMobili%C3%A1rios/Oque%C3%A9BolsadeValores/tabid/92/Default.aspx>. Acesso em: 01 de março de 2009.

ANEXO I

Algoritmo de Treinamento

Esta parte do algoritmo implementa o treinamento da RNA através do algoritmo de retro-propagação. Este algoritmo recebe os valores dos padrões a serem treinados (para padrões envolvendo uma vela: preços máximo/mínimo e preços abertura/fechamento; para padrões envolvendo duas velas: preços abertura/fechamento da primeira vela e preços abertura/fechamento da segunda vela) bem como os valores de saída x e y de cada padrão.

```
public String treinarNeuronio(String viesUmH,String viesDoisH,String viesTresH,String viesUmO,String
viesDoisO,String pesoW11H,
String pesoW21H,String pesoW31H,String pesoW12H,String pesoW22H,String pesoW32H,String
pesoW13H,
String pesoW23H,String pesoW33H,String pesoW14H,String pesoW24H,String pesoW34H,String
pesoW11O,
String pesoW21O,String pesoW12O,String pesoW22O,String pesoW13O,String pesoW23O,int epoca,
double maximoErro, double txAprend) throws Exception {
try {
if (padraoVela == 0)
this.criarMatriz1Vela();
else
this.criarMatriz2Vela();

this.setPesosDouble(viesUmH,viesDoisH,viesTresH,viesUmO,viesDoisO,pesoW11H,pesoW21H,pesoW31H,pesoW12H
,pesoW22H,pesoW32H,pesoW13H,
pesoW23H,pesoW33H,pesoW14H,pesoW24H,pesoW34H,pesoW11O,pesoW21O,pesoW12O,pesoW22O,pesoW13O,
pesoW23O);
maxErro = maximoErro/100;
N = txAprend/100;
valor = ""; //Setando este valore para VAZIO, não haverá criação de LOG
valor = this.algoritmoRetropropagacao(epoca);
} catch (Exception e) {
System.out.println("CtrNeuronio - treinarNeuronio(): "+e);
}
return valor;
}
private String algoritmoRetropropagacao(int epoca) throws Exception {
int contPadrao, contEpoca = 0;
erroGlobal = maxErro;
double x1=0.0,x2=0.0,x3=0.0,x4=0.0,somaQuadradosErros;
this.setValor(contEpoca);
try {
contEpoca = 1;
while (contEpoca <= epoca) {
contPadrao = 0;
somaQuadradosErros = 0.0;
while(contPadrao < tamanho) {
if (padraoVela == 0) {
x1 = objCtrNumerico.normalizaEntrada(matrizBolsa1Vela[contPadrao][0],19.85,21.95); //ABERTURA
x2 = objCtrNumerico.normalizaEntrada(matrizBolsa1Vela[contPadrao][1],19.69,21.90);
//FECHAMENTO
x3 = objCtrNumerico.normalizaEntrada(matrizBolsa1Vela[contPadrao][2],20.00,22.02); //MÁXIMO
x4 = objCtrNumerico.normalizaEntrada(matrizBolsa1Vela[contPadrao][3],19.68,21.64); //MÍNIMO
} else {
x1 = objCtrNumerico.normalizaEntrada(matrizBolsa2Vela[contPadrao][0],22.65,45.50); //ABERTURA 1
x2 = objCtrNumerico.normalizaEntrada(matrizBolsa2Vela[contPadrao][1],22.18,46.09);
//FECHAMENTO 1
x3 = objCtrNumerico.normalizaEntrada(matrizBolsa2Vela[contPadrao][2],22.40,46.34); //ABERTURA 2
```

```

                x4      =      objCtrNumerico.normalizaEntrada(matrizBolsa2Vela[contPadrao][3],22.61,43.98);
//FECHAMENTO 2
    }
    this.passosFrente(x1, x2, x3, x4);
    this.passosTras(contPadrao);
    somaQuadradosErros += Math.pow(E1,2) + Math.pow(E1,2);
    this.setValor(contEpoca);
    contPadrao++;
}
this.calculaErroGlobal(somaQuadradosErros);
System.out.println("Treinamento: erroGlobal: "+erroGlobal+" maxErro: "+maxErro+" ÉPOCA: "+contEpoca);
valor += "\n";
if (erroGlobal < maxErro) {
    contEpoca = epoca + 1;
    valor += "Treinamento COMPLETO para o erro máximo de "+maxErro*100+"%";
}
else
    contEpoca++;

if (contEpoca == epoca+1 && erroGlobal > maxErro)
    valor += "Treinamento INCOMPLETO para o erro máximo de "+maxErro*100+"%";
}
} catch (Exception e) {
    System.out.println("CtrNeuronio - algoritimoRetropropagacao(): "+e);
}
}
return valor;
}

```

Passos para Frente

Implementa a primeira parte do algoritmo de retro-propagação.

```

private void passosFrente(double x1, double x2, double x3, double x4) throws Exception {
    try {
        netH1 = this.getValorNetH1(x1, x2, x3, x4);
        netH2 = this.getValorNetH2(x1, x2, x3, x4);
        netH3 = this.getValorNetH3(x1, x2, x3, x4);
        this.calculaSaidaH();
        netO1 = this.getValorNetO1();
        netO2 = this.getValorNetO2();
        this.calculaSaidaO();
    } catch (Exception e) {
        System.out.println("CtrNeuronio - passosFrente(): "+e);
    }
}

private double getValorNetH1(double x1, double x2, double x3, double x4) throws Exception {
    double net = 0.0;
    try {
        net = (pesVumH*1 + pesW11H*x1 + pesW12H*x2 + pesW13H*x3 + pesW14H*x4);
    } catch (Exception e) {
        System.out.println("CtrNeuronio - getValorNetH1(): "+e);
    }
    return net;
}

private double getValorNetH2(double x1, double x2, double x3, double x4) throws Exception {
    double net = 0.0;
    try {
        net = (pesVdoisH*1 + pesW21H*x1 + pesW22H*x2 + pesW23H*x3 + pesW24H*x4);
    } catch (Exception e) {
        System.out.println("CtrNeuronio - getValorNetH2(): "+e);
    }
    return net;
}

```

```

private double getValorNetH3(double x1, double x2, double x3, double x4) throws Exception {
    double net = 0.0;
    try {
        net = (pesVtresH*1 + pesW31H*x1 + pesW32H*x2 + pesW33H*x3 + pesW34H*x4);
    } catch (Exception e) {
        System.out.println("CtrNeuronio - getValorNetH3(): "+e);
    }
    return net;
}

private double getValorNetO1() throws Exception {
    double net = 0.0;
    try {
        net = (pesVumO*1 + pesW11O*H1 + pesW12O*H2 + pesW13O*H3);
    } catch (Exception e) {
        System.out.println("CtrNeuronio - getValorNetO1(): "+e);
    }
    return net;
}

private double getValorNetO2() throws Exception {
    double net = 0.0;
    try {
        net = (pesVdoisO*1 + pesW21O*H1 + pesW22O*H2 + pesW23O*H3);
    } catch (Exception e) {
        System.out.println("CtrNeuronio - getValorNetO2(): "+e);
    }
    return net;
}

```

Passos para Trás

Implementa a segunda parte do algoritmo de retro-propagação.

```

private void passosTras(int contPadrao) throws Exception {
    try {
        this.calculaErro(contPadrao);
        this.calculaDeltaO();
        this.calculaDeltaH();
        this.atualizaPesosO();
        this.atualizaPesosH(contPadrao);
    } catch (Exception e) {
        System.out.println("CtrNeuronio - passosTras(): "+e);
    }
}

private void calculaSaidaH() throws Exception {
    try {
        if (funcaoSaida == 0) {
            H1 = objCtrNumerico.funcaoSigmoidal(netH1);
            H2 = objCtrNumerico.funcaoSigmoidal(netH2);
            H3 = objCtrNumerico.funcaoSigmoidal(netH3);
        } else {
            H1 = objCtrNumerico.funcaoHiperbolica(netH1);
            H2 = objCtrNumerico.funcaoHiperbolica(netH2);
            H3 = objCtrNumerico.funcaoHiperbolica(netH3);
        }
    } catch (Exception e) {
        System.out.println("CtrNeuronio - calculaSaidaH(): "+e);
    }
}

private void calculaSaidaO() throws Exception {
    try {
        if (funcaoSaida == 0) {

```

```

        O1 = objCtrNumerico.funcaoSigmoidal(netO1);
        O2 = objCtrNumerico.funcaoSigmoidal(netO2);
    } else {
        O1 = objCtrNumerico.funcaoHiperbolica(netO1);
        O2 = objCtrNumerico.funcaoHiperbolica(netO2);
    }
} catch (Exception e) {
    System.out.println("CtrNeuronio - calculaSaidaO(): "+e);
}
}

private void calculaErro(int indice) throws Exception {
    try {
        if (padraoVela == 0) {
            E1 = matrizBolsa1Vela[indice][4] - O1;
            E2 = matrizBolsa1Vela[indice][5] - O2;
        } else {
            E1 = matrizBolsa2Vela[indice][4] - O1;
            E2 = matrizBolsa2Vela[indice][5] - O2;
        }

    } catch (Exception e) {
        System.out.println("CtrNeuronio - calculaErro(): "+e);
    }
}

private void calculaDeltaO() throws Exception {
    try {
        if (funcaoSaida == 0) {
            deltaO1 = E1*objCtrNumerico.funcaoDerivadaSigmoidal(netO1);
            deltaO2 = E2*objCtrNumerico.funcaoDerivadaSigmoidal(netO2);
        } else {
            deltaO1 = E1*objCtrNumerico.funcaoDerivadaHiperbolica(netO1);
            deltaO2 = E2*objCtrNumerico.funcaoDerivadaHiperbolica(netO2);
        }
    } catch (Exception e) {
        System.out.println("CtrNeuronio - calculaDeltaO(): "+e);
    }
}

private void calculaDeltaH() throws Exception {
    try {
        if (funcaoSaida == 0) {
            deltaH1 = objCtrNumerico.funcaoDerivadaSigmoidal(netH1) * (deltaO1*pesW11O + deltaO2*pesW21O);
            deltaH2 = objCtrNumerico.funcaoDerivadaSigmoidal(netH2) * (deltaO1*pesW12O + deltaO2*pesW22O);
            deltaH3 = objCtrNumerico.funcaoDerivadaSigmoidal(netH3) * (deltaO1*pesW13O + deltaO2*pesW23O);
        } else {
            deltaH1 = objCtrNumerico.funcaoDerivadaHiperbolica(netH1) * (deltaO1*pesW11O + deltaO2*pesW21O);
            deltaH2 = objCtrNumerico.funcaoDerivadaHiperbolica(netH2) * (deltaO1*pesW12O + deltaO2*pesW22O);
            deltaH3 = objCtrNumerico.funcaoDerivadaHiperbolica(netH3) * (deltaO1*pesW13O + deltaO2*pesW23O);
        }
    } catch (Exception e) {
        System.out.println("CtrNeuronio - calculaDeltaH(): "+e);
    }
}

private void atualizaPesosO() throws Exception {
    try {
        if (vies == 0) {
            pesVumO = pesVumO + N * deltaO1 * 1;
            pesVdoisO = pesVdoisO + N * deltaO2 * 1;
        }
        pesW11O = pesW11O + N * deltaO1 * H1;
        pesW21O = pesW21O + N * deltaO2 * H1;
        pesW12O = pesW12O + N * deltaO1 * H2;
        pesW22O = pesW22O + N * deltaO2 * H2;
        pesW13O = pesW13O + N * deltaO1 * H3;
        pesW23O = pesW23O + N * deltaO2 * H3;
    } catch (Exception e) {
        System.out.println("CtrNeuronio - atualizaPesosO(): "+e);
    }
}

private void atualizaPesosH(int indice) throws Exception {
    try {
        if (vies == 0) {

```



```

    pesVumH = pesVumH + N * deltaH1 * 1;
    pesVdoisH = pesVdoisH + N * deltaH2 * 1;
    pesVtresH = pesVtresH + N * deltaH3 * 1;
}
if (padraoVela == 0) {
    pesW11H = pesW11H + N * deltaH1 * matrizBolsa1Vela[indice][0];
    pesW21H = pesW21H + N * deltaH2 * matrizBolsa1Vela[indice][0];
    pesW31H = pesW31H + N * deltaH3 * matrizBolsa1Vela[indice][0];
    pesW12H = pesW12H + N * deltaH1 * matrizBolsa1Vela[indice][1];
    pesW22H = pesW22H + N * deltaH2 * matrizBolsa1Vela[indice][1];
    pesW32H = pesW32H + N * deltaH3 * matrizBolsa1Vela[indice][1];
    pesW13H = pesW13H + N * deltaH1 * matrizBolsa1Vela[indice][2];
    pesW23H = pesW23H + N * deltaH2 * matrizBolsa1Vela[indice][2];
    pesW33H = pesW33H + N * deltaH3 * matrizBolsa1Vela[indice][2];
    pesW14H = pesW14H + N * deltaH1 * matrizBolsa1Vela[indice][3];
    pesW24H = pesW24H + N * deltaH2 * matrizBolsa1Vela[indice][3];
    pesW34H = pesW34H + N * deltaH3 * matrizBolsa1Vela[indice][3];
} else {
    pesW11H = pesW11H + N * deltaH1 * matrizBolsa2Vela[indice][0];
    pesW21H = pesW21H + N * deltaH2 * matrizBolsa2Vela[indice][0];
    pesW31H = pesW31H + N * deltaH3 * matrizBolsa2Vela[indice][0];
    pesW12H = pesW12H + N * deltaH1 * matrizBolsa2Vela[indice][1];
    pesW22H = pesW22H + N * deltaH2 * matrizBolsa2Vela[indice][1];
    pesW32H = pesW32H + N * deltaH3 * matrizBolsa2Vela[indice][1];
    pesW13H = pesW13H + N * deltaH1 * matrizBolsa2Vela[indice][2];
    pesW23H = pesW23H + N * deltaH2 * matrizBolsa2Vela[indice][2];
    pesW33H = pesW33H + N * deltaH3 * matrizBolsa2Vela[indice][2];
    pesW14H = pesW14H + N * deltaH1 * matrizBolsa2Vela[indice][3];
    pesW24H = pesW24H + N * deltaH2 * matrizBolsa2Vela[indice][3];
    pesW34H = pesW34H + N * deltaH3 * matrizBolsa2Vela[indice][3];
}
}
} catch (Exception e) {
    System.out.println("CtrNeuronio - atualizaPesosH(): "+e);
}
}
}

```

Cálculo do Erro Global

Cálculo do erro global para que possa ser comparado com a taxa máxima de erro escolhida pelo usuário.

```

private void calculaErroGlobal(double somaQuadradosErros) throws Exception {
    try {
        erroGlobal = 0.5*(somaQuadradosErros);
    } catch (Exception e) {
        System.out.println("CtrNeuronio - calculaErroGlobal(): "+e);
    }
}
}

```

Algoritmo de Teste

Esta parte do algoritmo implementa os testes a serem realizados após o treinamento da RNA. O algoritmo de teste recebe os quatro valores de entrada (para padrões de uma vela: preços máximo/mínimo e preços abertura/fechamento; para padrões envolvendo duas velas: preços de abertura/fechamento da primeira vela e preços abertura/fechamento da segunda vela).

```

public void testarBolsaValores(String ent1,String ent2,String ent3,String ent4)
    throws Exception {
    try {
        this.setEntradasDouble(ent1,ent2,ent3,ent4);
        if (padraoVela == 0) {
            abe1 = objCtrNumerico.normalizaEntrada(abe1,19.85,21.95); //ABERTURA
            fech1 = objCtrNumerico.normalizaEntrada(fech1,19.69,21.90); //FECHAMENTO
            max = objCtrNumerico.normalizaEntrada(max,20.00,22.02); //MÁXIMO
            min = objCtrNumerico.normalizaEntrada(min,19.68,21.64); //MÍNIMO
            this.passosFrente(abe1, fech1, max, min);

            if (((yUmMarEnf - maxErro <= O1) && (yUmMarEnf + maxErro >= O1)) &&
                ((yDoisMarEnf - maxErro <= O2) && (yDoisMarEnf + maxErro >= O2))) {
                Interface.jLabelResultado.setText("MARTELO-ENFORCADO!");
            } else if (((yUmNada - maxErro <= O1) && (yUmNada + maxErro >= O1)) &&
                ((yDoisNada - maxErro <= O2) && (yDoisNada + maxErro >= O2))) {
                Interface.jLabelResultado.setText("NÃO RECONHECIDO!");
            } else
                Interface.jLabelResultado.setText("NÃO RECONHECIDO!");

        } else {
            abe1 = objCtrNumerico.normalizaEntrada(abe1,22.65,45.50); //ABERTURA 1
            fech1 = objCtrNumerico.normalizaEntrada(fech1,22.18,46.09); //FECHAMENTO 1
            abe2 = objCtrNumerico.normalizaEntrada(abe2,22.40,46.34); //ABERTURA 2
            fech2 = objCtrNumerico.normalizaEntrada(fech2,22.61,43.98); //FECHAMENTO 2
            this.passosFrente(abe1, fech1, abe2, fech2);

            if (((yUmPadEnv - maxErro <= O1) && (yUmPadEnv + maxErro >= O1)) &&
                ((yDoisPadEnv - maxErro <= O2) && (yDoisPadEnv + maxErro >= O2))) {
                Interface.jLabelResultado.setText("PADRÃO ENVOLVENTE!");
            } else if (((yUmHarami - maxErro <= O1) && (yUmHarami + maxErro >= O1)) &&
                ((yDoisHarami - maxErro <= O2) && (yDoisHarami + maxErro >= O2))) {
                Interface.jLabelResultado.setText("HARAMI!");
            } else
                Interface.jLabelResultado.setText("NÃO RECONHECIDO!");
        }
        System.out.println("Saida O1: "+O1);
        System.out.println("Saida O2: "+O2);

    } catch (Exception e) {
        System.out.println("ControleNeuronio - testarBolsaValores(): "+e);
    }
}

```

ANEXO II

Reconhecimento de Padrões em Gráficos de Vela da Bolsa de Valores Através de uma Rede Neural

Marcos Segalla Pizzolatti¹

Sistemas de Informação – Universidade Federal de Santa Catarina (UFSC)
Campus Reitor João David Ferreira Lima - Bairro Trindade - Florianópolis - Santa Catarina - Brasil - CEP 88040-970

***Abstract.** This project is about Artificial Neural Network and the objective is to recognize the following patterns of the Japanese Candlestick: hammers and hanging man, engulfing pattern and harami lines. The Neural Network is composed of four input nodes, three hidden nodes and two output nodes. The values (patterns) of the input layer were extracted from a table provided by the Stock Exchange of Sao Paulo, for the stock of Petrobras, from 2006 to 2008. The values of the hammer and hanging man patterns are the following: prices of opening/closing maximum/minimum, involving only a single candle. The values of the engulfing pattern and harami lines, involving two candles are the following: prices of opening/closing of the first candle and prices of opening/closing of the second candle. With these values, the Neural Network tries to perform the training, separately, for the recognition of the patterns in question.*

***Resumo.** Este projeto constitui-se na construção de uma Rede Neural Artificial (RNA) com o objetivo de reconhecer os padrões de reversão martelo e enforcado, padrão envolvente e linhas harami existentes no gráfico de velas japonês. A RNA é composta de quatro nós de entrada, três nós intermediários e dois nós de saída. Os valores (padrões) da camada de entrada foram extraídos de uma tabela fornecida pela Bolsa de Valores de São Paulo, para as ações da Petrobras, no intervalo de tempo de 2006 a 2008. Os valores do padrão martelo e enforcado são os seguintes: preços de abertura/fechamento e preços máximo/mínimo, envolvendo apenas uma única vela. Já os valores dos padrões padrão envolvente e linhas harami, envolvendo duas velas, são: preços de abertura/fechamento da primeira vela e preços de abertura/fechamento da segunda vela. Com estes valores, a RNA tenta realizar o treinamento, separadamente, para o reconhecimento dos padrões em questão.*

Palavras-Chave: Redes Neurais, Inteligência Artificial, Conexão, RNA, Bolsa de Valores, Gráfico de Velas.

1. Introdução

Os investimentos em Bolsa de Valores tem se popularizado mundialmente como opção de investimento. Muitos investidores iniciantes, sem amplo conhecimento no mercado financeiro, ao se aventurarem no investimento de ações, acabam por perder grande parte de seus investimentos. Uma ferramenta que auxiliasse na decisão de compra/venda de ações seria de grande importância e ajuda para esta classe de investidores.

Nos dias atuais, investidores que não sabem interpretar corretamente gráficos de ações financeiras, poderiam ser facilmente auxiliados por programas implementados através de Redes Neurais Artificiais com o intuito de prever as tendências do mercado.

Investidores mais amadurecidos poderiam também se utilizar de ferramentas mais robustas implementadas por Redes Neurais Artificiais para o auxílio em suas decisões de compra/venda.

Este projeto consiste na construção de uma RNA capaz de reconhecer alguns padrões (indicadores de reversão) existentes nos gráficos de vela. Especificamente neste projeto, a RNA identifica os seguintes padrões de identificadores de reversão: martelo e enforcado, padrão envolvente e linhas harami (gravidez).

Indicadores de reversão têm como objetivo “avisar” ou dar um alerta de que uma tendência de alta ou de baixa está para perder esta tendência, revertendo a situação de compra/venda no mercado financeiro.

O gráfico de velas foi inventado pelos japoneses com intuito de aplicar na Bolsa de Valores do arroz. Utilizando uma RNA, que é um sistema inteligente capaz de processar informações de maneira similar à neurônios cerebrais, com limitações, é possível fazer um treinamento capaz de

aprender os padrões existentes no gráfico japonês a fim de reconhecê-lo posteriormente, sem a intervenção do cérebro humano, e sim do computador.

O ramo Conexionista tem como objetivo a busca pela simulação de comportamentos inteligentes baseados no funcionamento de neurônios cerebrais. *É um sistema dentro da Inteligência Artificial que visa estudar a mente por uma perspectiva computacional.* (ELLIS, 2009)

Outro objetivo dentro da Inteligência Artificial tem como foco a aplicação do conhecimento obtido no modelamento da mente, explicado por Ellis, para solução de problemas práticos, como a de encontrar padrões nos preços de ações negociadas em Bolsa de Valores, por exemplo.

Além do ramo Conexionista, implementado na prática computacional pelas Redes Neurais Artificiais, a Inteligência Artificial tem outros dois: a Inteligência Artificial Simbólica e a Inteligência Artificial Evolucionária. No ramo Simbólico os Sistemas Especialistas são os produtos mais conhecidos e no ramo Evolucionário os Algoritmos Genéticos é o que o mercado de aplicativos mais adota.

Na Figura 1 se pode observar um esboço da configuração da rede utilizada neste trabalho.

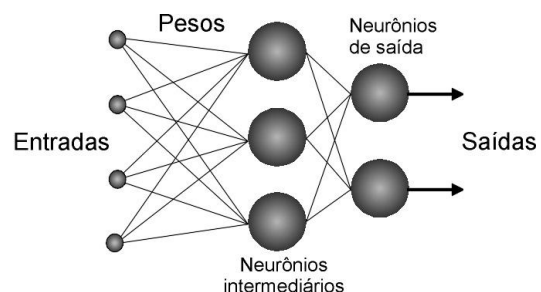


Figura 47: Configuração da Rede Neural Artificial do tipo Conexionista

Como se pode observar, a configuração da rede consta de quatro

nós de entrada, três nós intermediários e dois nós de saída que correspondem a dois valores de saída.

De acordo com esta lógica, terão como dados de entrada os seguintes valores: valor de abertura, valor de fechamento, valor máximo e valor mínimo (no caso do padrão martelo e enforcado) e valores de abertura e fechamento da primeira vela e valores de abertura e fechamento da segunda vela (no caso dos padrões “padrão envolvente” e “linhas harami”). Os dois valores da camada de saída caracterizam os padrões a serem reconhecidos pela RNA.

No padrão martelo e enforcado, o valor de abertura corresponde ao preço de abertura da ação (preço de primeiro negócio efetuado com esta ação) e o valor de fechamento corresponde ao preço de fechamento da ação (preço do último negócio), ambos no decorrer de um dia. O valor máximo e mínimo corresponde aos preços máximos e mínimos atingidos no dia. A saída esperada neste padrão corresponde a dois valores no intervalo de zero a um, x e y, escolhidos para o treinamento da RNA.

Nos padrões, padrão envolvente e linhas harami, ambos os valores, de abertura e fechamento das duas velas em questão, correspondem aos preços de abertura e fechamento de cada vela atingido no dia. As saídas desses dois padrões também correspondem a dois valores, x e y, escolhidos no intervalo de zero a um, para o treinamento da RNA.

O computador utilizado é um notebook de domínio pessoal com a seguinte configuração:

- Intel(R) Core(TM) 2 Duo CPU T7100 1.80GHz
- 2GB de memória SDRAM
- HD SATA de 160GB
- Windows VISTA Ultimate
- NetBeans 6.5

- JAVA(TM) 6 Update 12

Com base nestes dados, a RNA será capaz de aprender os padrões treinados a fim de identificá-los posteriormente, após o devido treinamento. Os dados extraídos são da Bolsa de Valores das ações da Petrobras, num período de três anos (2006 - 2008).

Vale enfatizar que neste trabalho os preços extraídos dos gráficos da Bolsa de Valores se referem ao período de um dia de negociação na Bolsa de Valores, porém também existem representações onde os preços se referem a períodos diferentes, como semanas, meses, anos, etc.

2. Objetivos

2.1. *Objetivo Geral*

O objetivo geral deste projeto consiste na construção de uma RNA capaz de reconhecer indicadores de reversão nos gráficos de vela da Bolsa de Valores do Petróleo.

2.2. *Objetivos Específicos*

- Estudar e avaliar a lógica dos gráficos de vela, para que através desta avaliação, se possa implementar uma RNA capaz de seguir esta lógica.
- Estudar e avaliar conceitos importantes sobre a Bolsa de Valores, para auxiliar na lógica de implementação da RNA.
- Estudar e avaliar o funcionamento de uma RNA, a fim de realizar a implementação de acordo com este tipo de rede neural.
- Construir uma Rede Neural Artificial capaz de entender a lógica por trás de alguns padrões

- dos gráficos de vela e conseqüentemente aprendê-los;
- Extrair dados reais da bolsa de valores e realizar treinamentos na rede neural.
- Construir uma interface capaz de interagir com o problema em questão.

3. Implementação

Para treinar a RNA foram utilizados dados (valores de abertura, fechamento, máximo e mínimo) extraídos de um gráfico diário das ações preferenciais da Petrobras da Bolsa de Valores, ao decorrer de três anos (2006 - 2008), gráfico este retirado do site ADVFN (<http://br.advfn.com/>) e representado na Figura 11.



Figura 2: Gráfico de Velas da Bolsa de Valores

Fonte: (<http://br.advfn.com/>)

A estrutura lógica da rede contém inicialmente as seguintes etapas:

12. Gerar pesos aleatórios (intervalos reais de -1 a 1)
13. Calcular os NETs das camadas intermediárias (somatório das multiplicações dos nós de entrada em relação aos pesos)
14. Calcular as saídas das camadas intermediárias (utilizando funções de saída do tipo Sigmoidal ou Hiperbólica)

15. Calcular os NETs da camada de saída (somatório das multiplicações dos nós de saída das camadas intermediárias em relação aos pesos)
16. Calcular as saídas da camada de saída (utilizando funções de saída do tipo Sigmoidal ou Hiperbólica)
17. Calcular o vetor de erro (diferença entre o resultado esperado e o obtido)
18. Calcular o erro global
19. Caso atingido o mínimo necessário: Fim
20. Atualizar os pesos da camada de saída
21. Atualizar os pesos das camadas intermediárias
22. Voltar ao passo 2

3.1. Interface e Características

A Interface do programa implementado está sintetizada na Figura 12. A seguir um texto descrevendo as funções sintetizadas do programa.

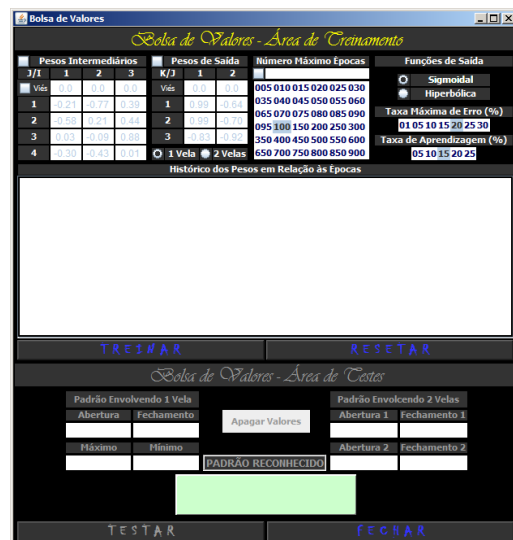


Figura 3: Interface do programa

- ❖ A parte superior (Bolsa de Valores – Área de Treinamento) do programa tem como objetivo programar e visualizar a RNA em si;

- ❖ A parte inferior (Bolsa de Valores – Área de Testes) tem como objetivo testar o programa para conferir se o mesmo foi corretamente treinado;
- ❖ Como default o programa inicia com pesos intermediários e de saída aleatórios (intervalo de -1 a 1);
- ❖ Como default os Pesos Viés são iguais a 0;
- ❖ Caso se queira treinar com os pesos Viés, basta clicar no checkBox “Viés”, e seus respectivos pesos serão gerados aleatoriamente no intervalo de -1 a +1;
- ❖ Como default a taxa máxima de erro vem setada como 20%;
- ❖ Como default a taxa de aprendizagem vem setada como 15%;
- ❖ Como default o número máximo de épocas vem setado como 100;
- ❖ Apenas será habilitada a “Área de Testes” após o devido treinamento do neurônio (após clicar no botão “TREINAR”);
- ❖ O programa permite escolher se os pesos intermediários, de saída e viés sejam números aleatórios ou se o usuário deseja escolher manualmente (caso deseje escolher manualmente, o intervalo varia de -99.99 a +99.99);
- ❖ O usuário pode escolher entre as seguintes taxas máximas de erro (%): 1, 5, 10, 15, 20, 25 e 30;
- ❖ O usuário pode escolher entre as seguintes taxas de aprendizagem (%): 5, 10, 15, 20, 25;
- ❖ Existe um quadro para se escolher o número máximo de épocas com números pré-definidos;
- ❖ Ao clicar no checkBox “Número Máximo Épocas”, é setado como default o número 1;
- ❖ Ao escolher os Pesos Intermediários, de Saída ou Viés manualmente, o programa valida, em tempo real, apenas números reais no intervalo de -99.99 a +99.99;
- ❖ Ao escolher o número máximo de épocas manualmente, o programa valida, em tempo real, apenas números naturais de até quatro dígitos (0 a 9999);
- ❖ Como default, o tipo de função de saída vem como “Sigmoidal”;
- ❖ O usuário tem a possibilidade de escolher entre dois tipos de funções de saída (Sigmoidal e Hiperbólica), através dos RadioButtons;
- ❖ Ao clicar no botão “TREINAR”, o programa irá recolher os dados de entrada (Pesos Intermediários, Pesos de Saída, Pesos Viés, Taxa Máxima de Erro, Taxa de Aprendizagem, Época e o Tipo de Função de Saída), e irá executar o algoritmo de treinamento;
- ❖ Ao finalizar o treinamento, aparecerá no campo “Histórico dos Pesos em Relação às Épocas” o histórico dos pesos de cada época;
- ❖ Caso se acione o botão “TREINAR” e algumas das entradas não forem preenchidas, aparecerão as devidas mensagens de advertência;
- ❖ O botão RESETAR, ao ser pressionado, setará a interface da mesma maneira como quando o programa foi iniciado, atribuindo outros conjuntos de pesos aleatórios às duas camadas de nós da Rede;
- ❖ Por default, o RadioButton com padrão envolvendo uma vela vem setado como true;
- ❖ Na Área de Treinamento, o usuário pode optar em treinar o neurônio usando um padrão envolvendo uma ou duas velas, através dos RadioButtons;
- ❖ O programa valida em tempo real (apenas números reais positivos no intervalo de 0 a 99.99), ao serem pressionados, todos os campos jTextField da Área de Testes;
- ❖ Ao clicar no botão TESTAR, o programa recolhe os dados dos campos de entrada da Área de Testes, mostrando o resultado no

jLabel abaixo do PADRÃO RECONHECIDO;

- ❖ Caso alguns dos campos de entrada da Área de Testes não forem preenchidos, o programa emitirá as possíveis mensagens de advertência;
- ❖ O botão FECHAR serve para encerrar o programa.

3.2. Funcionamento do Programa

Foi implementada uma interface para interagir com o usuário. A interface já vem setada com todos os valores necessários para o treinamento da RNA, bastando o usuário pressionar o botão TREINAR. Esses valores foram previamente descritos no item Interface e Características. O usuário também tem a opção, caso queira, de preencher manualmente os valores, de acordo com os limites do mesmo.

Com os devidos dados de entrada corretamente preenchidos e demais requisitos, o usuário poderá treinar a RNA. Após seu treinamento é habilitada a área de testes para que se possa testar a RNA previamente treinada. Também é disponibilizada ao usuário a visualização do histórico dos pesos gerados durante cada época de treinamento, assim como a informação se a RNA foi devidamente treinada de acordo com a taxa máxima de erro escolhida pelo usuário.

Através da área de testes do programa, o usuário digita os valores de determina vela do gráfico de velas, a RNA então deverá ser capaz de reconhecer os padrões treinados.

3.2.1. Padrão envolvendo uma vela (Martelo e Enforcado)

No caso de padrões envolvendo uma vela, a RNA é treinada para reconhecer apenas o indicador de reversão martelo e enforcado. Em outras palavras, para os

cinco padrões que entraram na RNA ela dirá quais são martelos e enforcados e quais não são. Na implementação para reconhecimento deste padrão, foram extraídos cinco padrões do gráfico das ações da Petrobras do ano de 2006, e outros cinco padrões que não correspondem ao padrão martelo e enforcado.

Na Figura 12 se pode verificar a configuração da RNA correspondente a padrões envolvendo uma vela com os quatro valores dos nós de entrada, os três nós ocultos (intermediários) e os dois nós de saída (x e y).

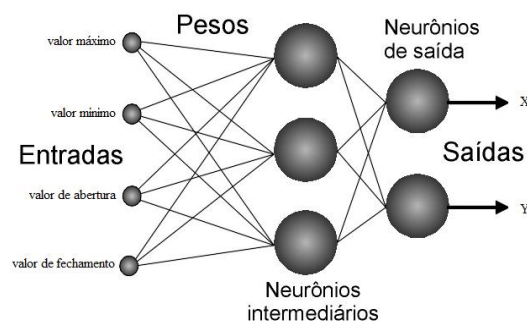


Figura 4: Configuração da RNA para padrões envolvendo uma vela

Os valores de entrada do padrão martelo e enforcado são os seguintes: valor de abertura, valor de fechamento, valor máximo e valor mínimo, exemplificados na Tabela 1.

PADRÃO	ABERTURA	MÁXIMO	MÍNIMO	FECHAMENTO
1	21,95	21,96	21,30	21,90
2	21,71	21,71	21,31	21,68
3	21,92	21,93	20,95	21,82
4	20,48	20,55	20,19	20,56
5	19,95	20,00	19,68	20,07

Tabela 4: Valores (em reais) dos cinco padrões martelo e enforcado

3.2.1.1. Gráficos do Padrão Martelo e Enforcado

A seguir as Figuras representando os padrões martelo e enforcado cujos valores foram extraídos para implementação da RNA, juntamente com suas explicações:

Nas Figuras 14, 15 e 16 estão representados o padrão enforcado, revertendo o mercado para uma tendência de baixa. Na Figura 17 está representado o padrão enforcado, revertendo o mercado para uma tendência de baixa e o padrão martelo, revertendo o mercado novamente para uma tendência de alta.

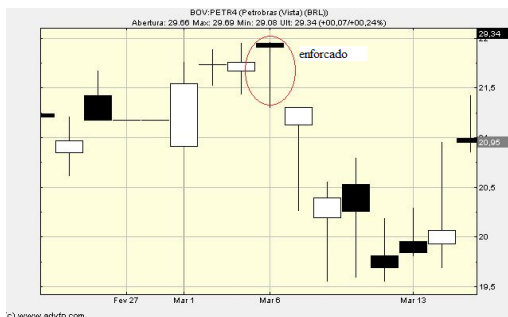


Figura 5: Padrão 1 (Enforcado)

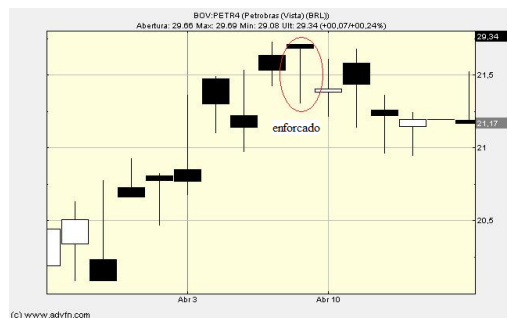


Figura 6: Padrão 2 (Enforcado)

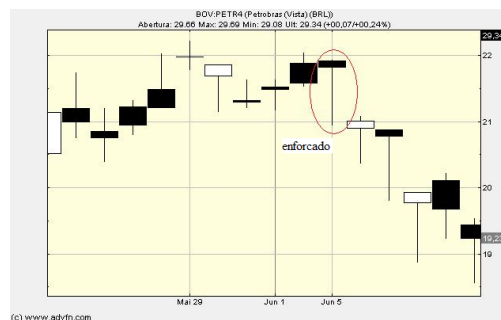


Figura 7: Padrão 3 (Enforcado)

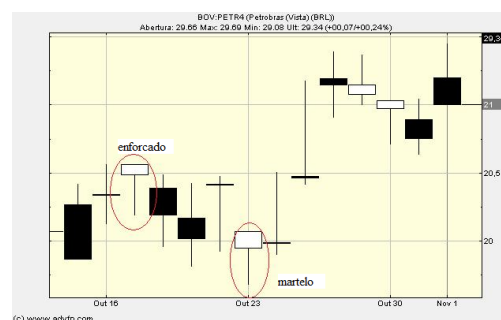


Figura 8: Padrões 4 e 5 (Martelo e Enforcado)

3.2.2. Padrão envolvendo duas velas (Padrão Envoltivo e Linhas Harami)

No caso de padrões envolvendo duas velas, a RNA é treinada para reconhecer os indicadores de reversão padrão envoltivo e linhas harami. Na implementação para reconhecimento destes padrões, foram extraídos cinco padrões do tipo padrão envoltivo do gráfico de ações da Petrobras entre os anos de 2006 a 2008 e cinco padrões do tipo linhas harami do gráfico de ações da Petrobras entre os anos de 2005 a 2008.

Na Figura 18 se pode verificar a configuração da RNA correspondente a padrões envolvendo duas velas com os quatro valores dos nós de entrada, os três nós ocultos (intermediários) e os dois nós de saída (x e y).

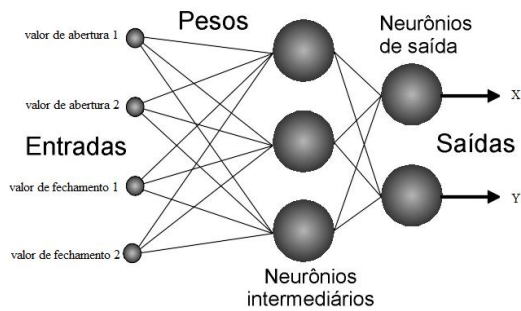


Figura 9: Configuração da RNA para padrões envolvendo duas velas

Os valores de entrada dos padrões padrão envolvente e linhas harami são os seguintes: valores de abertura e fechamento da primeira vela e valores de abertura e fechamento da segunda vela, representados na Tabela 2 e Tabela 3 respectivamente.

PADRÃO ENVOLVENTE	ABERTURA 1	FECHAMENTO 1	ABERTURA 2	FECHAMENTO 2
1	23,44	23,51	23,53	23,42
2	24,00	24,32	24,37	23,87
3	23,47	23,75	23,90	23,22
4	45,50	49,09	46,34	43,98
5	30,59	29,80	28,50	31,30

Tabela 5: Valores (em reais) dos cinco padrões padrão envolvente

HARAMI	ABERTURA 1	FECHAMENTO 1	ABERTURA 2	FECHAMENTO 2
1	23,25	22,63	22,75	23,15
2	23,39	23,65	23,59	23,50
3	22,65	22,18	22,40	22,61
4	23,80	25,10	24,99	24,25
5	18,89	20,52	20,17	19,95

Tabela 6: Valores (em reais) dos cinco padrões linhas harami

3.2.2.1. Gráficos do Padrão Envolvente e Linhas Harami

Nas Figuras 19, 20, 21 e 22 estão representados o padrão envolvente altista, revertendo o mercado para uma tendência de baixa. Na Figura 23 está representado o padrão envolvente baixista, revertendo o mercado para uma tendência de alta.

Na Figura 24 está representado o padrão linhas harami baixista, revertendo o mercado para uma tendência de alta e o padrão linhas harami altista, revertendo o mercado para uma tendência de baixa. Na Figura 25 está representado o padrão linhas harami baixista, revertendo o mercado para uma tendência de alta. nas Figuras 26 e 27 estão representados o padrão linhas harami altista, revertendo o mercado para uma tendência de baixa.

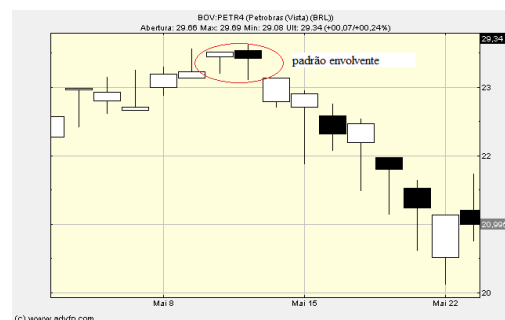


Figura 48: Padrão 1 (Padrão Envolvente Altista)

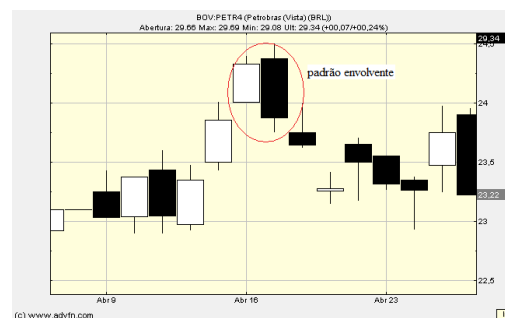


Figura 11: Padrão 2 (Padrão Envolvente Altista)

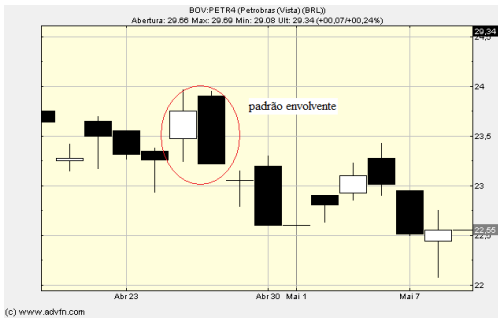


Figura 12: Padrão 3 (Padrão Envolvente Altista)

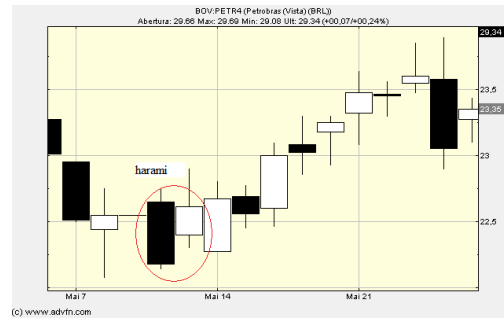


Figura 16: Padrão 3 (Linhas Harami Baixista)

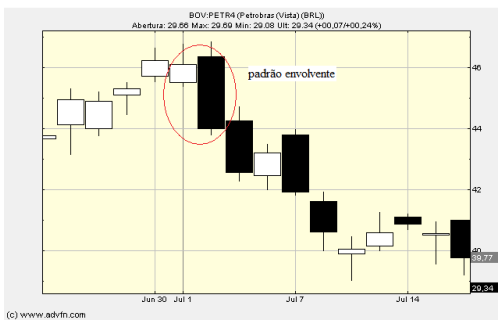


Figura 13: Padrão 4 (Padrão Envolvente Altista)

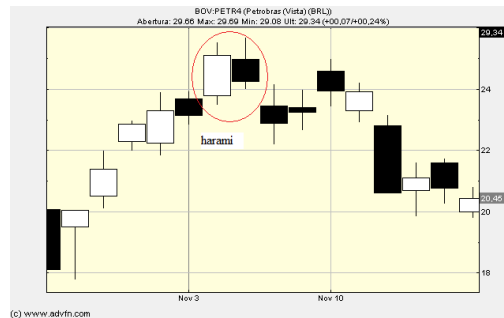


Figura 17: Padrão 4 (Linhas Harami Altista)

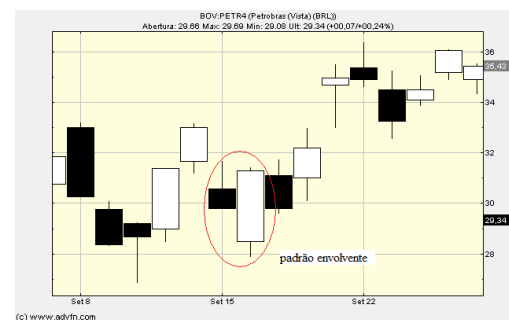


Figura 14: Padrão 5 (Padrão Envolvente Baixista)

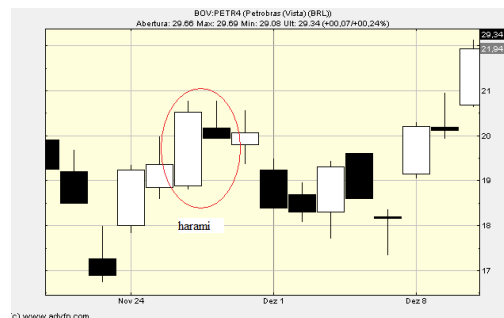


Figura 18: Padrão 5 (Linhas Harami Altista)

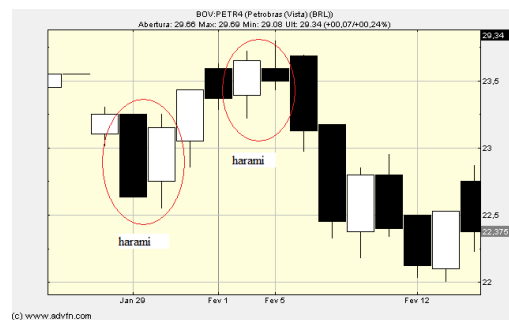


Figura 15: Padrões 1 e 2 (Linhas Harami Baixista e Altista)

3.2.3. Implementação para o Treinamento da Rede Neural Artificial

Para a implementação do treinamento da RNA, foi utilizado o algoritmo de retro-propagação. Este algoritmo considera o uso de quatro nós de entrada, três nós intermediários e dois nós de saída. Há opção de inclusão do nó de viés, conforme discutido no item 4.1.

Ao final de cada época, a RNA compara a taxa de erro global com a taxa máxima de erro escolhida pelo usuário. Caso a taxa de erro máxima escolhida pelo usuário seja igual ou menor à taxa de erro global, o treinamento é concluído. O treinamento é concluído com sucesso quando a taxa de erro máxima é menor ou igual à taxa de erro global, antes de completar o máximo (ou exatamente igual) número de épocas escolhidas pelo usuário. Caso contrário, a RNA é treinada, porém sem sucesso.

O programa implementado tem as funções principais disponibilizadas no anexo. São elas: parte de teste da RNA, onde o usuário vai colocar os valores correspondentes ao padrão que se deseja realizar o teste. Se a RNA reconhecer o padrão em questão, irá imprimir no programa o tipo de padrão reconhecido. Caso não reconheça, irá informar que este padrão não foi reconhecido. Parte de treinamento, onde a RNA vai pegar todos os valores correspondentes ao treinamento na interface. Com isto, geram-se os históricos dos pesos em relação a cada época de treinamento e é disponibilizado ao usuário se a RNA foi treinada com sucesso, ou não, de acordo com a porcentagem de erro máxima escolhida pelo usuário.

3.2.4. Implementação para o Teste da Rede Neural Artificial

Para testar o programa, é utilizada apenas a primeira parte do algoritmo de retro-propagação, os “passos para frente” do algoritmo, o qual consiste em calcular os valores dos NETs e das camadas de saída das camadas intermediárias, e calcular os NETs e os valores de saída das camadas de saída. Após a finalização do algoritmo, serão comparadas as taxas de erro dos valores colocados para testar a RNA.

Caso estas taxas de erro estejam dentro do reconhecimento de um padrão treinado, então a RNA irá reconhecer o padrão, caso contrário, o padrão não será reconhecido. O reconhecimento dos padrões depende da taxa máxima de erro conseguida no final de um treinamento.

3.3. Figuras do Programa Após Treinamento e Teste

A seguir as Figuras representando o treinamento completo da RNA com suas respectivas descrições:

- ❖ Figura 44: visualização da área de treinamento do programa onde observa-se todos os valores corretamente setados para o treinamento da RNA de padrões envolvendo duas velas e o histórico das épocas em relação aos pesos, mostrando um treinamento sem sucesso para o erro máximo de 20% escolhido para o treinamento.
- ❖ Figura 45: visualização da área de testes do programa onde observa-se os quatro valores de entrada preenchidos e reconhecidos corretamente pela RNA como sendo um padrão envolvente, mesmo com um erro máximo maior que 20%.
- ❖ Figura 46: visualização da área de testes do programa onde observa-se os quatro valores de entrada preenchidos e reconhecidos corretamente pela RNA como sendo um padrão não reconhecido, mesmo com um erro máximo maior que 20%.

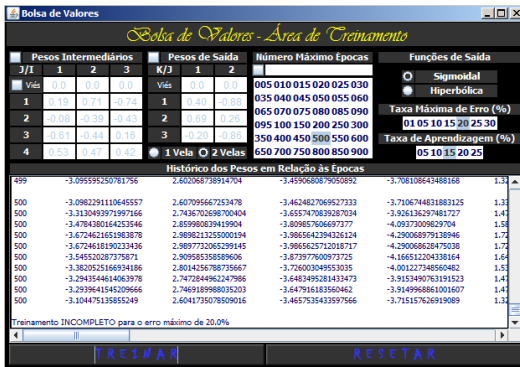


Figura 19: Conclusão do Treinamento do Neurônio Artificial



Figura 20: Teste de um padrão (Reconhecido)



Figura 21: Teste de um padrão (Não Reconhecido)

4. Conclusão

Foi verificado que ao treinar o padrão martelo e enforcado, a RNA não atingiu a capacidade de baixar, em curto espaço de tempo (poucas épocas), a taxa máxima de erro para pelo menos 30%.

Já quando é treinado um padrão envolvendo duas velas, o padrão envolvente e linhas harami, a RNA consegue ter pelo menos 50% de taxa máxima de erro ao final de um treinamento de até 500 épocas. Porém ainda não é considerada uma boa taxa máxima de erro para reconhecimento de

um padrão, tendo esta que ser de pelo menos 30%.

Com a mesma configuração da RNA adotada para o reconhecimento dos padrões de indicadores de reversão propostos nesse projeto, a RNA foi testada para treinar padrões envolvendo carros, motos e caminhões, com os seguintes valores de entrada: cilindros, cilindradas, peso e velocidade máxima. Os valores destas entradas são bem distantes entre si, e ao treinar a RNA com estes valores, se consegue um treinamento para uma taxa máxima de erro de 20%, com sucesso.

Considerando os fatos em questão, uma hipótese de não haver um total sucesso no treinamento da RNA para o correto reconhecimento dos padrões de indicadores de reversão propostos por esse projeto, é o fato desses valores serem muito aproximados entre si. Porém devemos levar em conta de que uma RNA possui suas limitações e dificilmente haverá 100% de acerto.

As dificuldades encontradas nesse projeto foram justamente o fato da proximidade dos valores de entradas dos padrões treinados na RNA, dificultando o sucesso no final de um treinamento. Foram realizados testes para a solução deste problema, como a mudança na configuração da RNA para duas camadas ocultas, uma única camada de saída, como também converter os valores de entrada para uma potência de 10, porém todas as tentativas sem sucesso.

Para trabalhos futuros, poderá ser avaliado com mais detalhes essa proximidade dos valores que impedem um melhor sucesso no treinamento da RNA como também tornar a RNA um pouco mais flexível, podendo o usuário escolher também a configuração da RNA, escolhendo a quantidade de camadas de entradas, camadas ocultas e de saída desejadas.

5. Referências

12. ALECRIM, Emerson. **Redes Neurais Artificiais**. Publicado em 09/05/2004. Documento em HTML disponível em <<http://www.infowester.com/redesneurais.php>>. Acesso em: 28 de Junho de 2007.
13. BASTTER.COM. **Aprendizado CandleStick: Martelos e Enforcados**. 2002. Documento em HTML disponível em: http://www.bastter.com/BR/MERCADO/Aprendizado/AnaliseTecnica/CandleSticks/martelos_e_enforcados.aspx. Acesso em: 19 de Abril de 2009.
14. BITTENCOURT, Guilherme. **Inteligência Computacional – Conexionismo**. Artigo HTML disponível em: <http://www.das.ufsc.br/gia/softcomp/node6.html>. Acesso em: 21 de Abril de 2009.
15. ELLIS, Nick. **Modelo Conexionista**. Documento digital em formato PDF disponível em <<http://www.veramenezes.com/conexionismo.pdf>>. Acesso em: 01 de março de 2009.
16. LIMA, Gerson. **A Lógica da Bolsa de Valores**. Documento em HTML disponível em <http://www.desempregozero.org.br/artigos/a_logica_da_bolsa_de_valores.php>. Acesso em: 28 de Junho de 2007.
17. MALHEIROS, Rivadavila S. **Análise Técnica: Fundamentos e Prática**. Realização: CMA Educacional, 2005.
18. MARQUES, Camila. **O que são ações, e como funciona a Bolsa de Valores**. Documento em HTML disponível em <http://www.imesexplica.com.br/1203bolsa_abre.asp>. Acesso em: 28 de Junho de 2007.
19. MENDES, Daniela Quintela. SILVA OLIVEIRA, Márcio Ferreira. **Tutorial de Redes Neurais**. Documento em HTML disponível em <http://www.lncc.br/~labinfo/tutorialRN/frm4_backpropagation.htm>. Acesso em: 28 de Junho de 2007.
20. NELOGICA. **Introdução a Análise Técnica**. Documento em HTML disponível em <<http://www.nelogica.com.br/tutoriais/introtec/introtec02.php>>. Acesso em: 01 de março de 2009.
21. NISON, Steve. **Curso de Análise Técnica Avançada Candlestick**. Realização: CMA Educacional. Resenha BM&F. 1992.
22. PORTAL DO INVESTIDOR, **O que é Bolsa de valores**. Documento em HTML disponível em <<http://www.portaldoinvestidor.gov.br/Acad%C3%AAmico/EntendooMercadodeValoresMobil%C3%A1rios/Oque%C3%A9BolsadeValores/tabid/92/Default.aspx>>. Acesso em: 01 de março de 2009.