

REDES NEURAIS E SUA CONFIABILIDADE E DESEMPENHO EM PERÍODOS INSTÁVEIS – APLICAÇÃO A AÇÕES ORDINÁRIAS PETR3

NEURAL NETWORKS AND ITS RELIABILITY AND PERFORMANCE IN UNSTABLE PERIODS – UPPLICATION IN ORDINARY SHARES PETR3

AFONSO GENTA PALANDRI^{1*}, FILIPE ROSEIRO COGO²

1. Mestrando em Tecnologia pelo Instituto LACTEC, docente do curso de Engenharia Elétrica da Faculdade de Engenharia e Inovação Técnico Profissional – FEITEP; 2. Mestre em Ciência da Computação pela Universidade Estadual de Maringá – UEM, docente do curso de Engenharia de Controle e Automação do Centro Universitário de Maringá – CESUMAR.

* Praça Manoel Ribas, 36, Apto 181, Maringá, Paraná, Brasil. CEP 87014-120. afonsopalandri@gmail.com

Recebido em 04/11/2013. Aceito para publicação em 05/11/2013

RESUMO

Este trabalho compara e quantifica o desempenho e a confiança de previsões utilizando Redes Neurais durante períodos instáveis. Como ferramenta de trabalho foi escolhida a linguagem Python por ser de simples programação e possuir abundante material de referência. Foi escolhido como série para previsão os preços das ações PETR3, da Petrobrás, com início em 2006 e término em 2011. Esse período é particularmente interessante por englobar a crise de 2008, onde o mercado apresentou grande instabilidade e insegurança. Comparamos assim o desempenho das Redes Neurais durante períodos pré e pós crise, e verificamos se as previsões correspondem com os valores observados. No ano de 2008, os dados apresentam uma instabilidade muito grande, e a previsão teve uma diferença de R\$3,00 em comparação com o valor obtido. Vale dizer que, neste período, a dispersão dos dados é muito grande também, contribuindo ainda mais na instabilidade da previsão. Durante os outros anos, porém, as previsões se aproximaram bastante dos resultados esperados. Em 2011, por exemplo, a diferença foi de R\$0,16, o que mostra que esta técnica produz resultados significantes e relativamente confiáveis. Mas, como qualquer forma de previsão, esta técnica produz uma estimativa de aonde a série vai estar no futuro próximo. Comparando o erro e o tempo computacional necessário desta técnica, observamos erros significativamente melhores, porém uma redução drástica no tempo de treinamento devido ao algoritmo utilizado.

PALAVRAS-CHAVE: Redes neurais artificiais, bolsa de valores, séries temporais.

ABSTRACT

This paper compares and quantifies the performance and reliability of Artificial Neural Networks (ANN's) during unstable periods in the prediction of time series. The chosen time series is the price of the stock PETR3, belonging to Petrobrás, a Brazilian oil company. The difference between the predicted value and the expected value in 2008 is R\$3.00, when the market is quite unpredictable due the 2008 global crash, and the ANN prediction may be unreliable. During the other years, the difference is smaller, and in 2011 the difference is only R\$0.16, showing that this approach produces significant results with relative reliability. However, as any form of prediction, this technique only produces a rough estimate of where the series is heading in the near future. Comparing the training error of this approach with other techniques (for example, using Back propagation as learning algorithm), shows that the error is sensible smaller using RPROP, but the computational time is reduced dramatically.

KEYWORDS: Artificial neural network, stock market, time series.

1. INTRODUÇÃO

Prever uma série temporal é um problema comum em vários domínios da ciência, como por exemplo, demografia, meteorologia, energia. No entanto, prever uma série temporal financeira de forma acurada é ainda uma tarefa trabalhosa¹.

Uma série temporal consiste de observações em uma ou várias variáveis em função do tempo. Neste tipo de sistema, eventos passados influenciam eventos futuros e atrasos no comportamento são prevalentes (por exemplo, uma ação que ocorre agora só exercerá influência no sistema depois de algum tempo), portanto o tempo é uma

dimensão importante numa série temporal².

Um tipo de série temporal é o mercado de ações. Ações são títulos representativos da menor fração do capital de uma empresa. O acionista, portador do título, passa a ser coproprietário com direito a participação nos lucros³. A previsão desses preços é possível devido ao fato dos valores futuros carregarem todas as informações necessárias para efetuar a previsão, sendo necessário desenvolver métodos analíticos e computacionais para efetuar as previsões dessas séries temporais com o máximo de precisão o possível⁴.

Existem várias ferramentas para previsão de séries temporais como, por exemplo, modelos estatísticos (suavização exponencial ou modelos auto regressivos) ou modelos computacionais (redes neurais, lógica nebulosa ou algoritmos genéticos)⁵. O mercado de ações possui um comportamento não linear e sensibilidade elevada. Portanto, a previsão desse tipo de sistema se mostra um desafio constante, tanto na área de estatística (para desenvolver modelos e métodos mais precisos) quanto na área computacional (para processar a quantidade cada vez maior de dados disponíveis).

Com características como a capacidade de aprender por meio de exemplos, generalizar a informação aprendida, auto-organização e processamento temporal, uma Rede Neural pode ser utilizada para prever os resultados de forma satisfatória⁶.

É possível abordar como uma RNA (Rede Neural Artificial) se comporta na previsão de dados turbulentos no mercado de ações. Para tal análise, foi selecionado como intervalo para testes o histórico das ações PETR3, desde o ano de 2007 ao ano de 2011. Este intervalo é particularmente interessante, pois abrange períodos de grande instabilidade, como por exemplo, a crise mundial de 2008, chamada de Grande Recessão de 2008-2012, e períodos que antecedem e sucede a crise⁷.

Mercado de ações

O mercado financeiro, ou mercado de ações, é um sistema complexo, evolucionário, não linear e dinâmico. O campo da previsão financeira é caracterizado pela intensidade, ruído, natureza desestruturada, alto grau de incerteza e relações ocultas entre os dados⁸.

O mercado de ações brasileiro, comumente chamado de BOVESPA, reúne as ações das principais empresas brasileiras de capital aberto. Uma ação é definida como a menor parcela em que se divide uma empresa, sendo que todos os dias são movimentados milhões entre investidores, seguindo a lei básica da oferta e da procura⁹.

Muitos fatores interferem nas finanças, incluindo eventos políticos, condições econômicas gerais e expectativa dos acionistas. Desta forma, prever o preço de um mercado financeiro é bem difícil. Variações, porém, nos preços não são aleatórias. Estas variações se

comportar de uma forma dinâmica e altamente não linear. A pressuposição de que os preços futuros são aleatórios vem meramente da aleatoriedade que envolve um processo não linear e ruidoso⁸.

Com a tecnologia atual, podemos acompanhar essas movimentações em tempo real, utilizando da Internet. Entre as mais diversas empresas com o capital aberto no mercado de ações, podemos citar algumas significativas, como Vale, Petrobras, Gerdau.

Na Figura 1 se observa, no intervalo de 16/11/2006 a 16/11/2011, os valores diários da ação PETR3. Se tratando de um sistema não linear e bastante sujeito a ruídos externos, podemos utilizar técnicas de representação de sistemas não lineares, como por exemplo, as redes neurais artificiais, para tentar prever o futuro deste sistema.

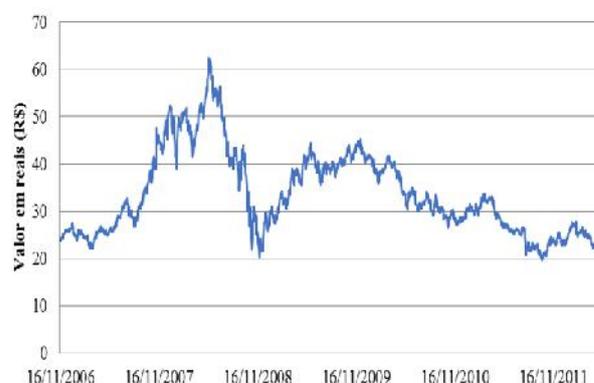


Figura 1. Cotações da ação PETR3 no período de 11/2006 a 05/2012 Redes Neurais Artificiais (Rna).

Redes Neurais Artificiais (RNA) representam uma tecnologia que nasceu de diferentes áreas de pesquisa: neurociência, matemática, estatística, física, ciências da computação e engenharia. As RNAs são empregadas hoje em dia nos mais diversos campos, como modelagem, análise de séries temporais, reconhecimento de padrões, processamento de sinal e controle, etc. A vasta aplicabilidade das RNAs deve-se a uma importante característica: a habilidade de aprender a partir dos dados da entrada, com ou sem um supervisor¹⁰.

Uma RNA é constituída de neurônios. Um neurônio é uma unidade de processamento de informação, fundamental para o correto funcionamento de uma rede neural. Como visto na Figura 2, um neurônio consiste de cinco elementos básicos:

- Um conjunto de sinapses ou entradas, representadas em (2) por x_{inp} , caracterizadas pelo peso

ou importância do mesmo.

- Um conjunto de pesos, representados por w_{inp} .

Estes pesos são multiplicados pela sua respectiva entrada.

- Um bloco somador, representado pelo símbolo Σ . Responsável por somar o produto de cada um dos nós ligados a ele.

- Uma função de ativação, ou função de esmagamento, representado pelo símbolo $\varphi(\cdot)$. Este se

encarrega por limitar a amplitude do resultado.

- A saída, y_k , resultante de todas as operações anteriores.

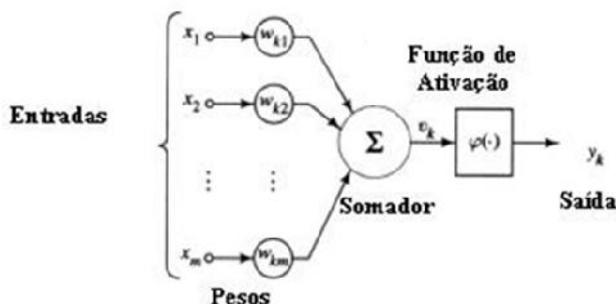


Figura 2. Estrutura Interna de um Neurônio. Adaptado de: http://www.gsigma.ufsc.br/~popov/aulas/rna/neuronio_artificial/neuronio_artificial.jpg

Um neurônio também pode ser descrito matematicamente, conforme as Equações 1 e 2:

$$u_k = \sum_{j=1}^m w_{km} * x_m \tag{1}$$

E

$$y_k = \varphi(u_k) \tag{2}$$

Onde:

- x_m é a entrada do nó m.
- w_{km} , o peso da ligação entre o nó m com o nó k.
- u_k representa a soma do produto de todos os nós

ligados ao nó k.

- E y_k , a saída do nó k, dada pela multiplicação de

u_k pela função de esmagamento.

A maneira em que os neurônios de uma rede neural são estruturados está intimamente ligada ao algoritmo de aprendizagem utilizado para treinar a rede¹⁰. Mas antes de falar sobre algoritmos de aprendizagem, falamos sobre o processo de aprendizagem como um todo. No contexto, aprendizagem é definida por:

“Aprendizagem é um processo em que parâmetros livres de uma rede neural se modificam através de um processo de estimulação do ambiente em que a rede está contida. O tipo de aprendizagem é determinado pela maneira em que os parâmetros da rede se modificam¹⁰”.

O processo de aprendizagem se dá em três etapas distintas:

1. A rede é estimulada pelo ambiente.
2. A rede modifica os pesos das conexões entre os nós (w_{km} , os ditos “parâmetros livres”), como

resultado de um estímulo.

3. A rede responde de uma nova forma ao estímulo do ambiente, devido às mudanças que ocorreram em sua estrutura interna.

Um conjunto de regras bem definidas para a solução do problema de aprendizagem é chamado de *algoritmo de aprendizagem*. Existem dois tipos de aprendizagem: supervisionada e não supervisionada.

A aprendizagem supervisionada é caracterizada pela presença de um agente supervisor. Este agente é responsável por comparar o resultado da rede com o resultado esperado. Tanto a rede quanto o agente estão sujeitos ao mesmo ambiente (por exemplo, o ambiente das ações PETR3), porém o ambiente é conhecido para o supervisor. Desta forma, o mundo deve ser totalmente observável. O supervisor sabe que, dado o conjunto de entrada (1,1,1), a resposta a este estímulo deve ser 111. E a partir da comparação entre o resultado esperado e o resultado obtido pela rede, ocorrerá o ajuste dos pesos das conexões, até que a rede seja capaz de responder aos estímulos como esperado, sem o auxílio do agente supervisor.

Já a aprendizagem não supervisionada não possui um agente supervisor. A rede é incapaz de saber se o resultado gerado está correto, ou ainda se está no caminho certo. Desta forma, aprendizagem não supervisionada é utilizada em problemas de classificação, onde a rede tem como objetivo agrupar tipos de dados similares e identificar exceções nos dados. A aprendizagem não supervisionada é utilizada em ambientes parcialmente observáveis, e nas mais diversas tarefas que envolvam classificar dados. Por exemplo, na detecção de dados incorretos ou fraudulentos.

Para o problema da previsão das ações PETR3, por se tratar de um ambiente totalmente observável, foi utilizado aprendizagem supervisionada. Dentre os algoritmos de aprendizagem supervisionada, citamos dois tipos, o algoritmo *Backpropagation* e o algoritmo *Resilient Propagation*.

O algoritmo *Backpropagation* (BPROP) é o algoritmo mais comumente utilizado para aprendizagem supervisionada com redes MLP. A ideia básica deste

algoritmo é aplicar repetidamente a Equação 3 para calcular a influência de cada peso na rede a respeito de uma função erro arbitrária E ¹¹.

$$\frac{\partial E}{\partial w_{km}} = \frac{\partial E}{\partial s_k} \frac{\partial s_k}{\partial net_k} \frac{\partial net_k}{\partial w_{km}} \quad (3)$$

Onde w_{km} é o peso da conexões entre o nó k e o nó m; s_k é o resultado e net_k é a soma das entradas multiplicadas pelos respectivos pesos do nó k. Uma vez que a derivada parcial para cada peso é conhecida, o objetivo de minimizar a função erro é alcançado ao executar uma gradiente descendente, demonstrado na Equação 4¹¹.

$$w_{km}(t + 1) = w_{km}(t) - \epsilon \frac{\partial E}{\partial w_{km}} \quad (4)$$

Onde ϵ é a taxa de aprendizagem. A escolha da taxa de aprendizagem tem um efeito importante no tempo necessário para alcançar a convergência. Se for muito pequeno, muitas épocas serão necessárias para alcançar uma solução aceitável. Ao contrário, uma taxa de aprendizagem grande vai possivelmente levar a oscilação do processo de treinamento, impedindo que o erro diminua depois de certo valor¹¹.

O algoritmo *Resilient Propagation* (RPROP) executa uma adaptação direta na etapa de mudança dos pesos baseado em informações de um gradiente local. Para alcançar isso, foi introduzido para cada peso um valor de atualização individual, Δ_{km} , que determina o tamanho

da mudança do peso. Esta mudança de valores adaptativa evolui durante o processo de treinamento baseado em uma visão local da função erro E , de acordo

com a seguinte regra de aprendizagem:

$$\Delta_{km}^{(t)} = \begin{cases} \eta^+ * \Delta_{km}^{(t-1)}, & \text{se } \frac{\partial E}{\partial w_{km}}^{(t-1)} * \frac{\partial E}{\partial w_{km}}^{(t)} > 0 \\ \eta^- * \Delta_{km}^{(t-1)}, & \text{se } \frac{\partial E}{\partial w_{km}}^{(t-1)} * \frac{\partial E}{\partial w_{km}}^{(t)} < 0 \\ \Delta_{km}^{(t-1)}, & \text{senão} \end{cases} \quad (5)$$

$$w_{km}^{(t+1)} = w_{km}^{(t)} + \Delta w_{km}^{(t)} \quad (6)$$

Verbalizando, esta regra funciona da seguinte forma: toda vez que a derivada parcial do peso correspondente

w_{km} muda de sinal indica que a última mudança foi muito grande e que o algoritmo pulou sobre um mínimo local, e seu valor de atualização diminui por um fator η^- . Se a derivada mantém seu sinal, o valor de atualização é aumentado levemente de forma a acelerar a convergência em regiões rasas¹¹.

Uma vez que o valor de atualização foi adaptado para cada peso, o tamanho da mudança de peso executa uma regra simples: se a derivada é positiva (isto é, o erro está aumentando), subtrai-se do peso seu valor de atualização (como visto em 5). Se a derivada é negativa, adiciona-se ao peso seu valor de atualização (regra 6)¹¹.

$$\Delta w_{km}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, & \text{se } \frac{\partial E}{\partial w_{km}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)}, & \text{se } \frac{\partial E}{\partial w_{km}}^{(t)} < 0 \\ 0, & \text{senão} \end{cases} \quad (7)$$

$$w_{km}^{(t+1)} = w_{km}^{(t)} + \Delta w_{km}^{(t)} \quad (8)$$

Porém há uma exceção: se a derivada parcial mudar de sinal (por exemplo, o passo anterior foi muito grande e o mínimo foi perdido), inverte-se o $\Delta w_{ij}^{(t)}$, como

mostrado na Equação 9.

$$\Delta w_{km}^{(t)} = -\Delta w_{km}^{(t)}, \quad \text{se } \left(\frac{\partial E}{\partial w_{km}}^{(t-1)} * \frac{\partial E}{\partial w_{km}}^{(t)} \right) < 0 \quad (9)$$

O algoritmo RPROP, em comparação ao BPROP, não somente é mais rápido, mas também é mais simples para ser implementado. Em alguns casos, o RPROP chega a ser quatro vezes mais rápido que o BPROP. Sua simplicidade é sua característica mais marcante, pois para um treinamento baseado no algoritmo BPROP ser considerado aceitável, são necessários exaustivos testes com sua taxa de aprendizagem, pois não há forma de determinar analiticamente um valor de ϵ ótimo¹¹.

Existem três tipos principais de topologias de redes, as quais se referem às ligações que são realizadas entre os neurônios:

1. *Feedforward* com uma única camada, ou *Single-Layer Feedforward*, ou ainda conhecida como *Perceptrons*. Embora seja chamada de camada única, possui duas camadas (camada de entrada e camada de

saída), porém como não há processamento na camada de entrada, somente contabiliza a camada de saída¹⁰.

2. *Feedforward* com múltiplas camadas, ou *Multilayer Feedforward*, ou ainda *Multilayer Perceptron*, representada abaixo na figura (3). São conhecidas por conter camadas internas, representadas na figura pela camada *Hidden*, que por sua vez possuem neurônios ocultos. A função desses neurônios ocultos é intervir entre a entrada e a saída de alguma maneira útil. Ao adicionar camadas ocultas, a rede é capaz de extrair estatísticas de maior ordem, e capaz de representar sistemas não lineares. Essa característica é particularmente valiosa quando o tamanho da camada de entrada é grande¹⁰.

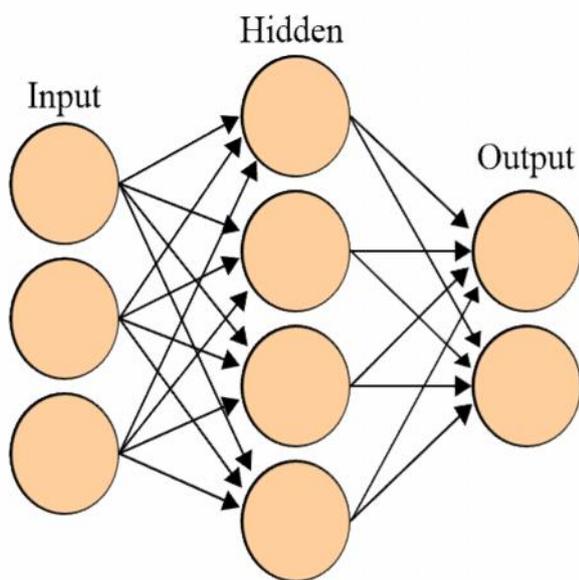


Figura 3. Exemplo de uma rede MLP (3,4,2). Fonte: http://upload.wikimedia.org/wikipedia/commons/e/4/Artificial_neural_network.svg

3. Redes recorrentes, ou *Recurrent networks*. São redes que possuem pelo menos um loop de *realimentação*. Por exemplo, uma rede recorrente pode consistir de uma única camada de neurônios onde cada neurônio de entrada é alimentado pelo sinal de saída da própria rede.

O objetivo deste trabalho foi observar e comparar a confiabilidade e o desempenho, por meio de métricas como: o tempo de treinamento no caso do desempenho, o erro de treinamento e a diferença entre a previsão e o valor observado de uma RNA na previsão dos valores das ações PETR3, durante o período entre 2007 e 2011.

Foi necessário efetuar um estudo piloto, para delimitar o espectro de testes. No estudo piloto, foi testado mais de 200 redes diferentes, modificando topologia, função de esmagamento e algoritmo de aprendizagem, em busca de uma configuração aceitável.

2. MATERIAL E MÉTODOS

Para definir os parâmetros de configuração da Rede Neural foi feito um estudo piloto cujo objetivo foi comparar diferentes configurações para o mesmo problema. Neste estudo, foram comparados diversos componentes que compõem uma RNA, totalizando 256 redes distintas. Os componentes testados foram:

- A quantidade de camadas ocultas na RNA.
- O número de nós ocultos em cada camada oculta.
- A quantidade de nós de entrada.
- A função de ativação.
- O algoritmo de aprendizagem.
- A normalização ou não dos dados no intervalo [0,1].

A configuração apresentada abaixo foi a que apresentou a melhor relação entre tempo de treinamento e erro de previsão, no problema em questão:

- [5] Nós de Entrada
- [5] Camadas Ocultas
- [10] Nós Ocultos em cada camada.

Com relação à função de ativação e ao algoritmo de aprendizagem, foram escolhidas as seguintes opções:

- Função de ativação Sigmóide.
- Algoritmo de aprendizagem *Resilient Propagation*.

A escolha do algoritmo de aprendizagem *Resilient Propagation* (RPROP) foi feita a partir dos resultados obtidos na comparação entre o *Backpropagation* (BPROP) e o *Resilient Propagation*. O tempo de treinamento do RPROP foi aproximadamente 10 vezes mais rápido que o BPROP, obtendo ainda resultados melhores com relação à previsão dos valores estimados.

Durante o estudo piloto não foi possível determinar se é benéfico ou não normalizar os dados antes de treinar a RNA. Por exemplo, ao utilizar um conjunto de dados pequeno (que no caso deste trabalho foi de um mês de valores da BOVESPA, 01/11/2011 a 30/11/2011) foram obtidos resultados melhores com a abordagem de não normalização. Dessa forma, a título de comparação, foi testada esta variável no problema apresentado.

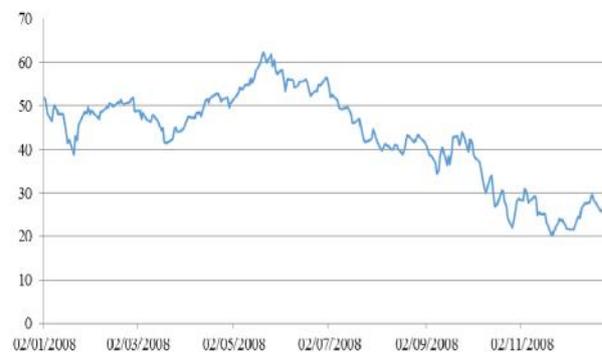


Figura 4. Cotações no ano de 2008.

Uma observação importante é que, durante o estudo, algumas redes apresentaram um tempo de treinamento

bastante curto (em torno de 10 segundos de treinamento), não condizente com o tamanho do conjunto de dados. Isso ocorre, pois durante o treinamento, caso o erro não melhore por um número pré-determinado de épocas, o algoritmo de treinamento entende que chegou ao melhor resultado e termina o treinamento. Depois de observado tal comportamento, foi escolhido arbitrariamente o valor de 1000 épocas, em que, se nessas 1000 épocas o algoritmo não observou nenhuma melhora, ele finaliza o treinamento. Caso haja uma melhora, ele continua por mais 1000 épocas até que essa regra seja completada.

Para o conjunto de treinamento foram escolhidas as ações PETR3 de 2007 a 2011. Foi escolhido um intervalo anual, pois assim podemos observar como a configuração de RNA escolhida se comporta em diferentes períodos.

No conjunto de dados escolhido, temos alguns períodos mais críticos e outros mais calmos, como por exemplo, o ano de 2008, mostrado na Figura 4, primeiro ano da Grande Recessão de 2008-2012⁷. Os valores das ações, que em Junho de 2008 estavam perto de R\$60,00, caíram para menos de R\$30,00 em menos de seis meses. O desvio padrão deste conjunto de dados é de aproximadamente R\$10,51.



Figura 5. Cotações no ano de 2011.

Já no ano de 2011 (Figura 5) a variação no preço é bem menor, com o desvio padrão dos dados cotando aproximadamente R\$3,79. Sendo assim, pode-se esperar uma maior confiabilidade nas previsões geradas por uma RNA neste intervalo de testes.

Os resultados obtidos para os dados normalizados podem ser verificados a partir da observação da Tabela 1.

Tabela 1. Resultado para os dados normalizados.

Período	Tempo de treinamento (s)	Erro de aprendizagem	Valor Previsto (R\$)	Valor Esperado (R\$)
2011	85,533	0,0196	23,16	23,00
2010	81,669	0,0379	29,03	30,45
2009	104,505	0,0401	41,01	41,56
2008	72,630	0,0315	26,24	29,4
2007	108,531	0,0125	47,95	48,01

Como citado anteriormente foram utilizados 5 nós de

entrada e o conjunto de treinamento total de um ano. A primeira coluna da Tabela 1 apresenta o período correspondente. A segunda coluna apresenta o tempo de treinamento que a rede consumiu para alcançar a convergência. A terceira coluna diz respeito ao erro observado durante a aprendizagem. A quarta coluna é o valor previsto pela rede. A quinta coluna é o valor esperado ou valor que a rede deveria prever.

Os resultados para o conjunto de dados não normalizados podem ser observados a partir da Tabela 2.

Tabela 2. Resultado para os dados não normalizados.

Período	Tempo de treinamento (s)	Erro de aprendizagem	Valor Previsto (R\$)	Valor Esperado (R\$)
2011	298,730	1,3783	31,69	23,00
2010	196,692	2,4141	30,66	30,45
2009	281,475	3,5181	39,60	41,56
2008	293,263	9,015	41,70	29,4
2007	121,350	3,6119	39,13	48,01

Assim sendo, podemos observar que para a configuração escolhida foi obtido resultados melhores quando se normaliza o conjunto de dados antes do treinamento. Ao normalizar os dados, ajustamos a escala de cada valor para que os valores fiquem no intervalo [0,1]. Este ajuste é necessário pois caso o conjunto de dados possua valores discrepantes (normalmente possui), estes valores irão influenciar o treinamento da rede de forma errônea.

Outra diferença significativa entre a normalização ou não dos dados foi o tempo de treinamento. Seguindo as regras descritas anteriormente, as redes cujas entradas não foram normalizadas demoraram pelo menos o dobro para terminarem o treino, com resultados longe de satisfatórios.

Se a sua função de ativação está no intervalo [0,1], (por exemplo, função sigmoide), então você deve ter a certeza que o seu conjunto de treino esteja neste intervalo¹².

3. RESULTADOS E DISCUSSÃO

Seguindo a metodologia descrita acima foi possível observar que, para as ações PETR3, é possível utilizar RNA's com significativa confiança. Mas em uma aplicação real a utilização de RNAs serve somente para fornecer subsídios para o processo decisório, o qual deve ser realizado pelo usuário da aplicação. A RNA deve ser vista como mais uma ferramenta e não como a verdade absoluta. O valor obtido pela previsão deve ser visto como uma indicação do comportamento da ação, muitas vezes não importando o valor em si.

Foi observado que, em todos os intervalos, a previsão foi perto do esperado menos em 2008. Embora o resultado obtido pela rede em 2008 esteja relativamente

longe do resultado esperado, a variação neste período é muito grande. No intervalo de uma semana até o dia 07/01/2009 (dia em que a cotação foi de R\$29,40), os valores foram de R\$25,89 a R\$29,40, chegando até a R\$31,02, como apresentado na Figura 6.

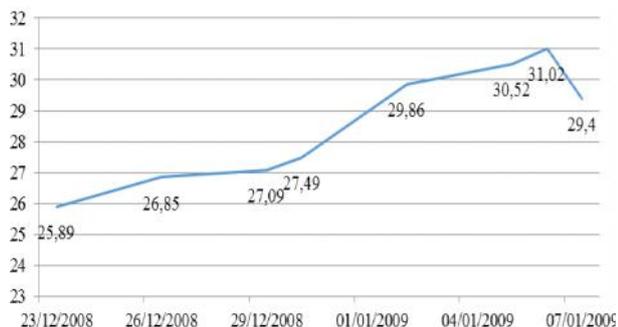


Figura 6. Cotações de 23/12/2008 a 07/01/2009.

Embora o resultado no período crítico (2008) não foi tão perto quanto nos períodos mais amenos, o resultado ainda pode ser considerado satisfatório, porém não ótimo. A previsão para este período (R\$26,24) só vai ser observada no fim de 2011.

Porém em outros períodos, como por exemplo, em 2007 (ano que precedeu a grande crise, caracterizada por uma valorização atípica) ou em 2011 (primeiro ano de relativa estabilidade no mercado de ações) a rede obteve resultados bastante satisfatórios. Em 2009 e 2010, o mercado apresentou um erro similar à rede treinada com o conjunto de 2008, porém devido ao desvio padrão dos dados nestes períodos ser menor, o resultado foi mais perto do esperado.

Utilizando de RNAs também, agora com o algoritmo de aprendizagem *Backpropagation*, conseguiu um erro de 0,0164 e tempo de treinamento de 5000 a 10000 épocas no melhor dos casos, enquanto neste trabalho, utilizando do algoritmo *Resilient Propagation*, foi obtido um erro de 0,0125 e 3000 a 5000 épocas. O trabalho de de Faria foi baseado nas ações PETR4, porém como as ações PETR3 e PETR4 seguem os mesmos índices, foi feita esta comparação.

No trabalho de Koskela *et al.* (1996)¹³ foi comparado como as redes neurais *Multilayer Perceptron*, FIR e Elman se comportam em diferentes cenários. Em um destes cenários foi efetuada a previsão baseada em séries geradas numericamente, cenário que se assemelha ao apresentado durante este trabalho. O erro obtido neste cenário, utilizando as abordagens MLP, Elman e FIR foram 0,0152, 0,0261 e 0,0305, respectivamente. Novamente, o erro obtido, no melhor dos casos, pela abordagem utilizando do algoritmo *Resilient Propagation* foi de 0,0125.

Outro ponto interessante é que as RNA's não são capazes de prever mudanças decorrentes de fatores exteriores (por exemplo, a crise imobiliária de 2008).

Mas com dados suficientes, ela é capaz de prever se as ações continuarão caindo ou subindo.

No problema descrito (e genericamente falando, qualquer problema em que o conjunto de dados seja suficientemente grande), deve-se normalizar antes de alimentar a rede, de acordo com o intervalo de sua função de ativação.

4. CONCLUSÃO.

Tendo em vista estas informações podemos concluir que uma RNA é capaz de gerar dados confiáveis, desde que o conjunto de treinamento seja suficientemente grande e que seja feito um pré-processamento correto antes de treiná-la.

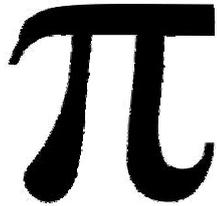
No entanto, como em qualquer técnica preditiva em que valores são o produto desejado, e no caso de mercado de ações, valores de tamanha magnitude, as RNA's devem ser encaradas como uma ferramenta, um auxiliador, e não uma entidade capaz de tomar decisões por conta própria.

REFERÊNCIAS

- [1] Lendasse A, Bodt E, Wertz ., & Verleysen M. Non-linear financial time series forecasting - Application to the Bel 20 stock market index. 2000.
- [2] Woldridge JM. Introductory Econometrics: A Modern Approach. South-Western College. 2008.
- [3] Marques FC, Gomes RM. Análise de Séries Temporais Aplicadas ao Mercado Financeiro com o uso de Algoritmos Genéticos e Lógica Nebulosa. Belo Horizonte: CEFET-MG. 2008.
- [4] Cavalcante F Mercado de Capitais: o que é, como funciona. Rio de Janeiro: Campus / Elsevier. 2005.
- [5] Ribeiro CV, Goldschmidt RR, Choren R. Métodos para Previsão de Séries Temporais e suas Tendências de Desenvolvimento. 2009.
- [6] Braga A, Ludermir BT, & Carvalho AC. Redes Neurais Artificiais: Teoria e aplicações. Rio de Janeiro: LTC. 2000.
- [7] Wessel D. Did 'Great Recession' Live Up to the Name? The Wall Street Journal. 2010.
- [8] Huang W, Nakamori Y, Wang SY. Forecasting stock market movement direction with support vector machine. Elsevier. 2004.
- [9] de Faria EL, Albuquerque MP, González Alfonso JL, Albuquerque MP, Cavalcante JT. Previsão do Mercado de Ações Brasileiro utilizando Redes Neurais Artificiais. Rio de Janeiro. 2008.
- [10] Haykin S. Neural Networks - A Comprehensive Foundation. Pearson Education. 1999.
- [11] Riedmiller M, Braun H. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. University of Karlsruhe. 1993.
- [12] Sarle WS. (11 de 10 de 2002). comp.ai.neural-nets FAQ, Part 2 of 7: Learning. Acesso em 01 de 10 de 2012, disponível em:

<http://www.faqs.org/faqs/ai-faq/neural-nets/part2/>.

- [13]Koskela T, Lehtokangas M, Saarinen J, Kaski K. Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks. Tampere University of Technology, Tampere, Finland. 1996.

A large, bold, black Greek letter pi symbol (π) is centered on the page. The symbol is rendered in a classic serif font, with a thick, slightly curved top bar and a vertical stem that tapers slightly towards the bottom.