

UM ESTUDO SOBRE O EFEITO DA DIMENSÃO DOS PARÂMETROS DE MODELOS ARIMA-GARCH NA PREDIÇÃO DE SÉRIES TEMPORAIS POR REDES NEURAS FEEDFORWARD

MAURI, A. OLIVEIRA

ALESSANDRA, A. MONINI

*Departamento de Administração da FEA-USP, Universidade de São Paulo
Rua Oliveira Melo, 104 CEP 04271-000 São Paulo – SP
E-mail: mauriao@usp.br*

Abstract— This work use Monte Carlo simulations to verify the ARIMA-GARCH parameters influence in feedforward artificial neural networks predictions trained with Levenberg-Marquardt algorithm. Neural networks have been widely used to time series predictions and its capacity to work with non linear models is a remarkable advantage. Nevertheless, generalized autoregressive heteroskedasticity model parameters values can influence the neural network prediction performance. The GARCH parameters values and ARIMA autoregressive terms combination also imply in neural networks performance variation. Combining ARIMA-GARCH models parameters and changing neural networks topologies we use Theil inequality coefficient to measure prediction performance and propose a rule to feedforward artificial neural networks choice.

Keywords— ARIMA-GARCH, Transfer Functions, Predictions, Monte Carlo Simulation

Resumo— Este trabalho utiliza simulações de Monte Carlo para verificar a influência de parâmetros ARIMA-GARCH nas previsões de redes neurais artificiais do tipo feedforward treinadas com algoritmo de Levenberg. Redes neurais têm sido amplamente utilizadas para previsões de séries temporais e sua capacidade de trabalhar com modelos não lineares é uma vantagem destacável. Entretanto, os valores dos parâmetros de modelos de heterocedasticidade condicional auto-regressiva generalizada podem influenciar o desempenho de previsão da rede neural. A combinação dos parâmetros GARCH e dos termos auto-regressivos ARIMA também implicam em variação de desempenho das redes neurais. Combinando os parâmetros dos modelos ARIMA-GARCH e mudando as topologias das redes neurais utilizamos o coeficiente de desigualdade de Theil para medir o desempenho de previsão e sugerir uma regra para escolha da rede neural artificial.

Palavras-chave— ARIMA-GARCH, Funções de Transferência, Previsões, Simulação de Monte Carlo.

1 Introdução

As redes neurais constituem uma ferramenta flexível amplamente utilizada para a análise de séries temporais. Sua aplicação tem sido feita nas mais variadas situações que envolvem problemas do mundo real de diversas áreas, notadamente finanças e economia. A necessidade de processamento temporal é importante em diversas aplicações, e isso tem motivado o desenvolvimento de novas técnicas e metodologias que utilizam redes neurais para esse fim. As redes neurais podem ser classificadas como uma técnica de inferência não-paramétrica não-linear multivariada, devido a possibilidade de introduzir variáveis diferentes cuja interdependência pode ser usada para prever valores futuros de uma seqüência temporal. Embora exista uma grande quantidade de técnicas que tratam de não-linearidade, a capacidade das redes neurais em lidar com este tipo de comportamento tem sido bastante explorada para realizar previsões. A característica não-paramétrica das redes neurais, ou seja, não fazer suposições sobre a forma paramétrica do rela-

cionamento funcional entre as variáveis a serem modeladas, tem sido uma vantagem adicional para sua utilização, que tem sido intensificada devido aos avanços da capacidade de processamento computacional. Do ponto de vista teórico, o processamento de sinais não-lineares [1], incorporação do tempo na rede neural [2] e o modelamento não-linear para previsão de séries temporais caóticas [3] tem levado a aplicação de redes neurais como ferramenta na tomada de decisão em finanças [4][5][6], análise de mercado [7], modelamento não-linear e previsão [8][9][10].

Mais recentemente tem havido a preocupação de comparar e relacionar a tecnologia de redes neurais com a abordagem estatística tradicional [11][12][13][14][15][16], sob a perspectiva econométrica [17], de engenharia financeira [18] e macroeconômica [19].

Neste trabalho vamos considerar séries temporais ARIMA-GARCH univariadas construídas por simulação de Monte Carlo. O objetivo dos modelos ARIMA-GARCH é a previsão de valores futuros de uma seqüência $\{y_t\}$. As previsões serão realizadas por redes neurais feedforward de três camadas, sendo

que a camada intermediária utilizará, dentre as várias possibilidades existentes, a função sigmóide. Para uma revisão sobre redes feedforward ver Fine (1999) [20] e sobre funções de ativação ver Duch e Jankowski (1997, 1999, 2000) [21][22][23]. A função sigmóide é a forma mais comum utilizada na construção de uma RNA:

$$\varphi(y) = \frac{1}{1 + e^{-gy}} \quad (1)$$

onde g é o ganho, ou parâmetro de inclinação da função sigmóide. Variando-se o parâmetro g , obtemos funções sigmóides com diferentes inclinações.

Para escolha de outros tipos de funções de ativação, ver Chen et al. (2001).

Através de simulações com números aleatórios, vamos analisar o comportamento da função de ativação sigmóide [33] e da sua primeira derivada quando os parâmetros relacionados a estacionariedade, erro quadrado e volatilidade condicional dos modelos ARIMA-GARCH aumentam individualmente. Um número aleatório pode ser obtido de formas diferentes e assim recebendo uma classificação específica tal como número pseudo-aleatório e quasi-aleatório [24][25][26]. Nesse trabalho as simulações são realizadas usando números pseudo-aleatórios. Com respeito ao algoritmo de minimização utilizado destacamos que atualmente estão disponíveis vários algoritmos de otimização, tal como: backpropagation, steepest-descent, backpropagation com momentum, Levenberg-Marquardt e Gauss-Newton. Todos estes algoritmos realizam a tarefa de minimizar o erro quadrado médio, neste trabalho utilizaremos o algoritmo de Levenberg-Marquardt. As previsões serão realizadas um passo a frente, por isso teremos apenas um neurônio na camada de saída da rede neural [27][28][29][30].

2 Processo GARCH

O modelo de volatilidade de séries temporais mais importante para estimar variância condicional é o modelo de processos de heterocedasticidade condicional auto-regressiva generalizada – *generalized autoregressive conditional heteroskedasticity* – GARCH [34][35][36][37]. Este modelo é um dos mais utilizados para a modelagem da volatilidade de séries temporais, ou seja, ele é utilizado quando a volatilidade da série não é constante [31][32][38]. Esta técnica é capaz de capturar a influência de muitos fatos sobre a série, notadamente heterocedasticidade, aglomerados de volatilidade e excesso de curtose.

Seja $\{v_t\}$ uma seqüência de variáveis aleatórias i.i.d. tal que $v_t \sim N(0,1)$. Sendo que $\{\varepsilon_t\}$ é chamada heterocedasticidade condicional auto-regressiva generalizada ou processo GARCH(p,q) se

$$\varepsilon_t = v_t \sigma_t, t \in Z \quad (2)$$

sendo que (σ_t) é um processo não negativo tal que

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \dots + \alpha_p \varepsilon_{t-p}^2 + \beta_1 \sigma_{t-1}^2 + \dots + \beta_q \sigma_{t-q}^2 \quad (3)$$

$$\alpha_0 > 0, \alpha_i \geq 0 \quad i=1, \dots, p \quad \beta_i \geq 0 \quad i=1, \dots, q \quad (4)$$

As condições sobre os parâmetros asseguram forte positividade da variância condicional (3) [39].

Se a equação (3) for escrita em termos do operador defasagem L temos que

$$\sigma_t^2 = \alpha_0 + \alpha(L) \varepsilon_{t-1}^2 + \beta(L) \sigma_t^2 \quad (5)$$

sendo que

$$\alpha(L) = \alpha_1 L + \alpha_2 L^2 + \dots + \alpha_p L^p \quad (6)$$

$$\beta(L) = \beta_1 L + \beta_2 L^2 + \dots + \beta_q L^q \quad (7)$$

Se as raízes da equação característica, ou seja,

$$1 - \beta_1 x - \beta_2 x^2 - \dots - \beta_q x^q = 0$$

situam-se fora do círculo unitário e o processo $\{\varepsilon_t\}$ é estacionário, então podemos escrever (3) como

$$\sigma_t^2 = \frac{\alpha_0}{1 - \beta(1)} + \frac{\alpha(L)}{1 - \beta(L)} \varepsilon_t^2 = \alpha_0^* + \sum_{i=1}^{\infty} \delta_i \varepsilon_{t-i}^2 \quad (8)$$

sendo que $\alpha_0^* = \frac{\alpha_0}{1 - \beta(1)}$, e δ_i são os coeficientes de

B^i na expansão de $\alpha(L)[1 - \beta(L)]^{-1}$. Da equação

(2) segue que o processo GARCH(1,1)

$$\alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \quad (9)$$

é estacionário se o processo $\{\sigma_t^2\}$ é estacionário.

Embora muitas variantes dos processos GARCH tenham sido sugeridas, Ding, Granger e Engle (1993) [40] demonstraram com simulações de Monte Carlo que o modelo original GARCH proposto por Bollerslev (1986) [34] é capaz de produzir o padrão de autocorrelação que aparece em dados financeiros. Normalmente, os processos GARCH servem como benchmark para as avaliações de séries temporais com presença de volatilidade condicional [41]. Sendo que o AR(1)-GARCH(1,1)

$$y_t = \phi_0 + \phi_1 y_{t-1} + v_t \sqrt{\alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2} \quad (10)$$

é o modelo mais comumente utilizado.

Existem muitos trabalhos escritos sobre o desempenho de previsão dos modelos GARCH [42][43][44] e tradicionalmente algumas medidas são mais utilizadas para comparar o ajuste e a precisão da previsão de modelos e técnicas alternativas: o RMSE (raiz do erro quadrado médio), MAE (erro absoluto médio), MAPE (erro percentual absoluto médio), CORR (co-

eficiente de correlação de Pearson), Sign (proporção de vezes em que o sinal é corretamente previsto) e o TIC (coeficiente de desigualdade de Theil). Para avaliar o desempenho de previsão da RNA para diversas combinações de parâmetros de modelos ARIMA-GARCH utilizaremos o TIC dado por:

$$TIC = \frac{\sqrt{\sum_{t=T+1}^{T+h} \frac{(\hat{y}_t - y_t)^2}{h}}}{\sqrt{\sum_{t=T+1}^{T+h} \frac{\hat{y}_t^2}{h} + \sum_{t=T+1}^{T+h} \frac{y_t^2}{h}}} \quad (11)$$

O TIC varia entre zero e um, sendo que zero indica ajuste perfeito.

3 Resultados de previsões de séries AR-GARCH para diferentes topologias de RNA

Nesta seção vamos realizar a previsão de um passo a frente de processos AR[1]-GARCH[1,1] com diferentes combinações de parâmetros, usando cinco topologias de RNA feedforward[49][50][51][52][53]. Cada um dos parâmetros do modelo AR(1)-GARCH(1,1) (ϕ_1 , α_1 , β_1), individualmente, irá variar nos intervalos: [0,1;0,2], [0,3;0,4], [0,5;0,6] e [0,7;0,8]. Desta forma, teremos as 64 combinações de parâmetros apresentadas na Figura-1. Para cada combinação serão geradas via Monte Carlo 100 séries com tamanho 1000. As séries são divididas em duas partes, uma para treinamento constituindo de 900 valores e os 100 valores restantes serão utilizados para realizar a previsão. Esta estratégia de segmentação da série temporal é comum na análise de séries temporais, a primeira parte, contendo um maior número de observações, é usada para estimar o modelo e as observações finais são utilizadas para medir a capacidade de previsão do modelo estimado[54][55].

Os valores observados e ajustados serão comparados pelo coeficiente de desigualdade de Theil (TIC) dado pela expressão (11). Este procedimento será realizado para redes neurais com as seguintes topologias 10-1-1, 10-3-1, 10-5-1, 10-7-1 e 10-9-1, função de ativação do tipo sigmóide e algoritmo de treinamento de Levenberg-Marquardt [45][46][47][48]. Os dados na maioria dos casos necessitam ser normalizados, a exceção é para dados que em certo sentido já estão normalizados, tal como dados binários ou dados que compõem a mesma série temporal e são apresentados no mesmo formato [56].

Neste trabalho não realizaremos a normalização dos valores das séries temporais. Os valores de entrada da RNA são os mesmos obtidos pela simulação.

Primeiro caso: previsões obtidas pela RNA 10-1-1.

A Figura-2 apresenta o Box-plot e gráfico de linha dos coeficientes TIC para os 100 valores previstos pela RNA considerando α_1 e β_1 variando no intervalo [0,1;0,2] e ϕ_1 variando nos intervalos [0,1;0,2], [0,3;0,4], [0,5;0,6] e [0,7;0,8].

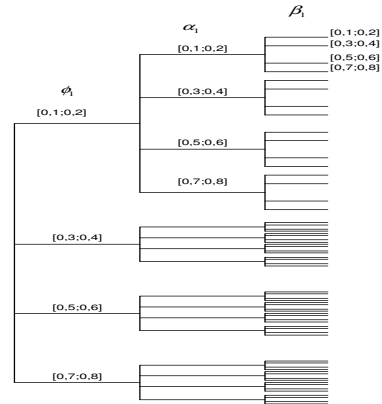


Figura-1 Árvore de combinação de parâmetros para simulação das séries AR[1]-GARCH[1,1].

A medida que aumentamos o valor do parâmetro ϕ_1 que está diretamente relacionado a estacionariedade da série, o coeficiente de desigualdade de Theil diminui. Ou seja, mantendo os parâmetros α_1 e β_1 variando no intervalo [0,1;0,2], a previsão realizada pela RNA um passo à frente torna-se cada vez melhor quando o parâmetro ϕ_1 da série aumenta.

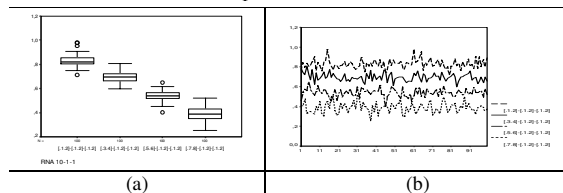


Figura-2 (a) Box-plot do TIC para a RNA 10-1-1 considerando aumento gradativo de ϕ_1 . (b) TIC de acordo com o aumento de ϕ_1 para a RNA 10-1-1.

Se considerarmos que o parâmetro ϕ_1 continue variando no intervalo [0,1;0,2] e que os parâmetros α_1 e β_1 variem, individualmente, dentro das várias possibilidades do primeiro ramo da árvore de combinações da Figura-1, podemos ver na Figura-3a que a maior concentração do TIC para cada combinação dos parâmetros oscilou em torno de um valor próximo de 0,8, indicando que as previsões não foram boas. Na Figura-3b apresentamos os box-plot relacionados aos coeficientes TIC originados pelas previsões em que o parâmetro ϕ_1 varia no intervalo [0,3;0,4] e α_1 e β_1 assumem, individualmente, valores de acordo com as combinações possíveis pré-definidas. Notamos que os valores do TIC associados a estas previsões concentraram-se em torno de 0,7,

ou seja, foram menores quando comparados aos obtidos na Figura-3a. Nos dois casos seguintes, mostrados na Figura-3c e 4d em que o parâmetro ϕ_1 varia nos intervalos de [0,5;0,6] e de [0,7;0,8], respectivamente, temos que o TIC das previsões continuou em sua tendência de declínio. Esse comportamento leva à conclusão que para uma rede neural com dez neurônios na camada de entrada, um neurônio na camada intermediária, função de ativação sigmóide e um neurônio na camada de saída, a RNA realiza melhores previsões quando a série tem parâmetro ϕ_1 entre [0,7;0,8], exceto para as seguintes combinações de parâmetros: ϕ_1 entre [0,7;0,8], α_1 entre [0,3;0,4] e β_1 entre [0,7;0,8]; ϕ_1 entre [0,7;0,8], α_1 entre [0,5;0,6] e β_1 entre [0,5;0,6]; ϕ_1 entre [0,7;0,8], α_1 entre [0,5;0,6] e β_1 entre [0,7;0,8]; ϕ_1 entre [0,7;0,8], α_1 entre [0,7;0,8] e β_1 entre [0,5;0,6]; ϕ_1 entre [0,7;0,8], α_1 entre [0,7;0,8] e β_1 entre [0,7;0,8].

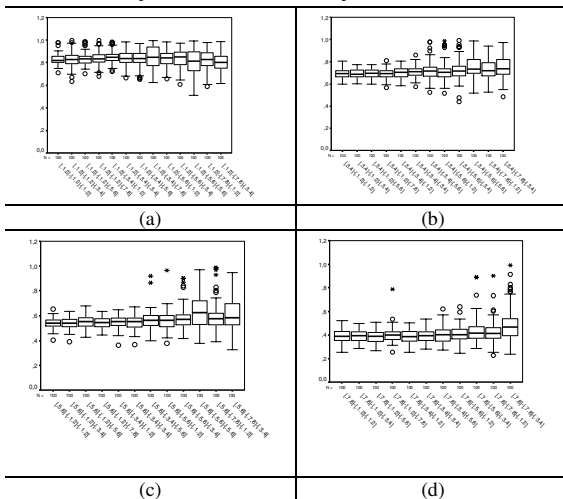


Figura-3 (a) Box-plot do TIC para a RNA 10-1-1 considerando ϕ_1 variando no intervalo [0,1;0,2] e várias combinações de α_1 e β_1 , (b) ϕ_1 varia no intervalo [0,3;0,4], (c) ϕ_1 varia no intervalo [0,5;0,6], (d) ϕ_1 varia no intervalo [0,7;0,8].

É importante ressaltar que, nas simulações realizadas, em séries em que o parâmetro relacionado a volatilidade podia assumir valores no intervalo [0,7;0,8] se o parâmetro ϕ_1 ou α_1 também estivessem nesse intervalo ou no intervalo [0,5;0,6] a variância apresentava persistência auto-regressiva acentuada, e nestes casos não é adequada a utilização desta topologia de RNA para realizar previsão.

Podemos verificar, na Figura-4, que aumentando gradativamente o valor do parâmetro α_1 as previsões de um passo à frente da RNA 10-1-1 não tem melhora significativamente como ocorreu quando aumen-

tamos os valores de ϕ_1 . Os valores do TIC obtidos ficam oscilando em torno de 0,8.

Quando alteramos o parâmetro ligado a volatilidade, β_1 , e mantemos ϕ_1 e α_1 variando simultaneamente no intervalo [0,1;0,2], ver Figura-5, verificamos que as previsões de acordo com o TIC não tiveram melhora, pois seus valores oscilaram em torno de 0,8.

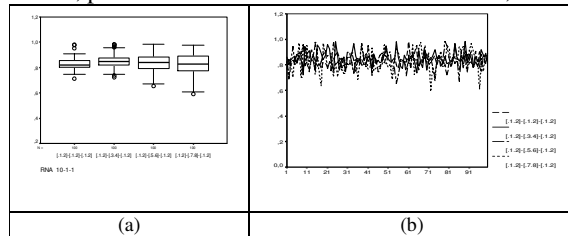


Figura-4 (a) Box-plot do TIC considerando aumento gradativo de α_1 , (b) TIC com o aumento de α_1 .

Segundo caso: previsões obtidos pelas RNA 10-3-1, 10-5-1, 10-7-1 e 10-9-1. Neste segundo caso vamos apresentar os resultados obtidos pelas redes neurais com três, cinco, sete e nove neurônios na camada intermediária. Na Figura-6 apresentamos o box-plot do coeficiente TIC para as topologias RNA 10-3-1, 10-5-1, 10-7-1 e 10-9-1 com os parâmetros variando nos intervalos [0,1;0,2], [0,3;0,4], [0,5;0,6] e [0,7;0,8]. Pela figura podemos identificar quatro agrupamentos em cada uma dessas topologias.

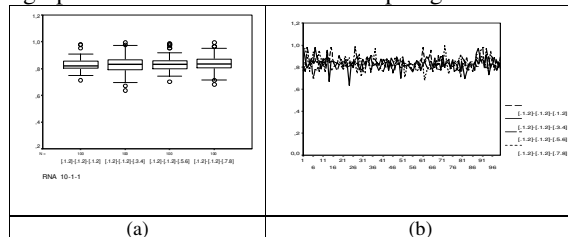


Figura-5 (a) Box-plot do TIC para a RNA 10-1-1 considerando aumento gradativo da volatilidade β_1 , (b) variação do TIC de acordo com o aumento de β_1 para a RNA 10-1-1.

A Figura-6a corresponde aos resultados de ajuste de previsão obtidos pela RNA 10-3-1 seguindo as combinações de parâmetros ϕ_1 , α_1 e β_1 . O primeiro agrupamento de box-plot da Figura-7a corresponde ao primeiro ramo mostrado na árvore de combinações de parâmetros da Figura-1 onde temos ϕ_1 variando em [0,1;0,2] com as várias combinações de α_1 e β_1 , no segundo temos que ϕ_1 varia em [0,3;0,4], no terceiro agrupamento ϕ_1 está variando entre [0,5;0,6] e no quarto agrupamento ϕ_1 varia em [0,7;0,8]. Novamente aqui é preciso dizer que foram removidas as combinações de parâmetros que implicaram em séries com efeito auto-regressivo da volatilidade muito acentuado. Podemos ver que para as

quatro topologias de rede o mesmo efeito de melhora de ajuste foi obtido com o aumento do parâmetro ϕ_1 .

Dos resultados obtidos vamos analisar se a rede com maior número de neurônios na camada intermediária mostrou desempenho de previsão melhor do que a rede com um neurônio nessa camada. Inicialmente, vamos analisar qual foi o comportamento de previsão para o caso em que o parâmetro ϕ_1 ligado a estacionariedade torna-se cada vez maior, a Figura-7 apresenta as distribuições dos coeficientes de desigualdade de Theil dessa situação para a rede 10-9-1.

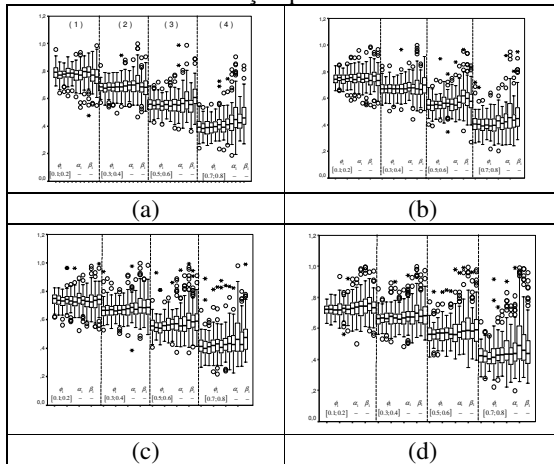


Figura-6 (a) Box-plot do TIC para a RNA 10-3-1, (b) Box-plot do TIC para a RNA 10-5-1, (c) Box-plot do TIC para a RNA 10-7-1, (d) Box-plot do TIC para a RNA 10-9-1.

Verificamos que a melhora do ajuste de previsão continua acontecendo quando aumentamos ϕ_1 da série, mantendo os termos de erro quadrado e volatilidade abaixo de 0,2. Comparando os resultados obtidos pela RNA 10-9-1, ver Figura-7, com aqueles da RNA 10-1-1, ver Figura-2, notamos que com ϕ_1 abaixo de 0,4 a rede com nove neurônios na camada intermediária apresentou menores TIC.

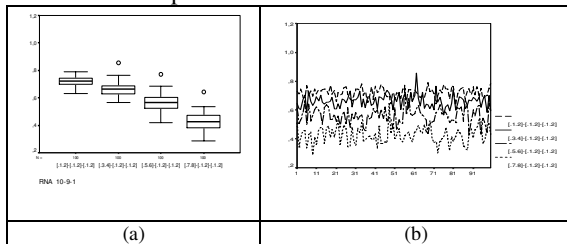


Figura-7 (a) Box-plot do TIC para a RNA 10-9-1 considerando aumento gradativo de ϕ_1 , (b) variação do TIC de acordo com o aumento de ϕ_1 para a RNA 10-9-1.

Na Tabela-1 temos os resultados dos percentis de 25, 50 e 75 % obtidos pela RNA 10-9-1 e pela RNA 10-1-1. Verificamos que com ϕ_1 acima de 0,5 a RNA 10-1-1 teve desempenho de previsão melhor que a RNA 10-9-1.

	ϕ_1 [0,1;0,2]	ϕ_1 [0,3;0,4]	ϕ_1 [0,5;0,6]	ϕ_1 [0,7;0,8]	
	α_1 [0,1;0,2]	α_1 [0,1;0,2]	α_1 [0,1;0,2]	α_1 [0,1;0,2]	
	β_1 [0,1;0,2]	β_1 [0,1;0,2]	β_1 [0,1;0,2]	β_1 [0,1;0,2]	
RNA 10-1-1					
Percentil	25	,8019	,6643	,5175	,3513
	50	,8202	,6938	,5385	,3892
	75	,8546	,7218	,5670	,4302
RNA 10-9-1					
Percentil	25	,6977	,6290	,5232	,3819
	50	,7224	,6632	,5642	,4257
	75	,7454	,6879	,6055	,4702

Tabela-1 Percentis do TIC obtidos na RNA 10-1-1 e 10-9-1.

4 Conclusão

Neste trabalho, mostramos através de simulações que os parâmetros de séries temporais que apresentam heterocedasticidade condicional auto-regressiva generalizada influenciam no desempenho da previsão de redes neurais feedforward.

Verificamos que a medida que o parâmetro da estacionariedade aumenta, mantendo os outros dois parâmetros fixos, as previsões da RNA melhoram, independente da topologia. No caso da RNA com maior número de neurônios na camada intermediária 10-9-1 verificou-se uma leve melhora de previsão quando o parâmetro relacionando a estacionariedade varia entre 0,1 e 0,8 mantendo-se os demais parâmetros inferiores a 0,2. Todas as topologias apresentaram melhores previsões para séries em que o parâmetro ligado a estacionariedade estava dentro do maior intervalo de simulação [0,7;0,8]. Nosso estudo concentrou-se no uso de RNA do tipo feedforward de três camadas treinada com algoritmo de Levenberg-Marquardt. Para essa configuração específica concluímos que é importante realizar a estimação do modelo AR-GARCH antes da seleção da topologia da rede neural, pois os valores dos parâmetros influenciam diretamente nos resultados das previsões.

Referências

- [1] LAPEDES, A. e FARBER, R. Nonlinear Signal Processing Using Neural Networks, Prediction and System Modelling. Los Alamos Report LA-UR-87-2662, Los Alamos National Laboratory, 1987.
- [2] ELMAN, J. L. Finding structure in time (CRL Technical Report 8801). La Jolla: University of California, San Diego, Center for Research in Language, 1988.
- [3] CASDAGLI, M. Nonlinear Prediction of chaotic time series. Physica D, vol. 35, pp. 335-356, 1989.
- [4] HAWLEY, D. D., JOHNSON, J. D. e RAINA, D. Artificial Neural Systems: A New Tool for Financial Decision-Making. Finan Anal J, 1990.
- [5] REFENES, A. N., BURGESS, A. N., BENTZ Y., Neural Networks in Financial Engineering: A Study in Methodology, IEEE Transactions on Neural Networks, Vol. 8, No. 6, 1997.
- [6] TAYLOR, J. W. A Quantile Regression Neural Network Approach to Estimating the Conditional Density of Multiperiod Returns, Said Business School, University of Oxford, Journal of Forecasting, Vol. 19, pp. 299-311, 2000.

- [7] FISHMAN, M. B., BARR, D. S. e LOICK, W. J. Using Neural Nets in Market Analysis. Technical Analysis of Stocks and Commodities, vol. 9, no. 4, 1991.
- [8] CASDAGLI, M. e EUBANK, S. Nonlinear Modeling and Forecasting, Santa Fe Institute Proceedings vol. XII, Addison-Wesley, 1992.
- [9] AZZOF, E. M. Reducing Error in Neural Network Time Series Forecasting. Neural Computing and Applications, vol. 1, pp. 240-247, 1993.
- [10] CLEMENTS, M. P. e HENDRY, D. F. Forecasting Non-stationary Economic Time Series. MIT Press: Cambridge, MA, 1999.
- [11] CHENG, B. E TITTERINGTON, D. M., Neural Networks: A Preview from Statistical Perspective. Statistical Science, 1994.
- [12] RIPLEY, B. D., Statistical Aspects of Neural Networks. Chapman e Hall, 1993.
- [13] RIPLEY, B. D., Neural Networks and Related Methods of Classification, Journal of the Royal Statistical Society, Series B, 56, 3, pp. 409-456, 1994.
- [14] RIPLEY, B. D., Pattern Recognition and Neural Networks. Cambridge Press, 1996.
- [15] HWANG, J.T.G. e DING, A. A. Prediction intervals for artificial neural networks. Journal of the American Statistical Association 92: 109-125, 1997.
- [16] MEDEIROS M., TERÄSVIRTA, T. e RECH, G. Building Neural Network Models for Time Series: A Statistical Approach, Journal of Forecasting 25, pp. 49-75, 2006.
- [17] KUAN, C. M., e WHITE, H. Artificial Neural Networks: An Econometric Perspective. *Econometric Reviews* 13, pp. 1-91, 1994.
- [18] ABU-MOSTAFA, Y. S., ATIYA, A. F., MAGDON-ISMAIL, M, e WHITE, H. Introduction to the special issue on neural networks in financial engineering. IEEE Transactions on Neural Networks 12: 653-655, 2001.
- [19] TERÄSVIRTA, T., VAN DIJK, D. e MEDEIROS M. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: a re-examination. International Journal of Forecasting, 2005.
- [20] FINE, T. L. Feedforward Neural Network Methodology. Springer: New York, 1999.
- [21] DUCH, W. e JANKOWSKI, N. New neural transfer functions. *Journal of Applied Mathematics and Computer Science*, 7(3):639-658, 1997.
- [22] DUCH, W. e JANKOWSKI, N. Survey of neural transfer functions. *Neural Computing Surveys*, 2:163-212, 1999.
- [23] DUCH, W. e JANKOWSKI, N. Taxonomy of neural transfer functions. In Shun-Ichi Amari, C. Lee Giles, Marco Gori, and Vincenzo Piuri, editors, *International Joint Conference on Neural Networks*, volume III, pages 477-484. Como, Italy & Los Alamitos, California, July 2000. Computer Society and IEEE.
- [24] ROSS, S. M. Simulation, academic press, 2006.
- [25] KNUTH, D. E. Seminumerical Algorithms, Vol. 2 of The Art of Computer Programming, second edn, Addison-Wesley, 1981.
- [26] NIEDERREITER, H. Random Number Generation and Quasi-Monte Carlo Methods. CBMS-NSF Regional Conference Series in Applied Mathematics 63. SIAM, 1992.
- [27] BISHOP, C. M. Neural Networks for Pattern Recognition. Oxford University Press, 1995.
- [28] HERTZ, J., KROGH, A. E PALMER, R. G., Introduction to the Theory of Neural Computation, Addison-Wesley, 1991.
- [29] WILLIAMS, R. J. e ZIPSER, D. Gradient based learning algorithms for recurrent networks and their complexity, 1995.
- [30] ROJAS, R., Neural Networks. A Systematic Introduction. Springer-Verlag, 1996.
- [31] REFENES, A. N. Currency Exchange Rate Prediction and Neural Network Design Strategies. Neural Computing and Applications, vol.1, pp. 46-58, 1993.
- [32] ORMONETT, D. e NEUNEIER, R. Experiments in predicting the German stock index DAX with density estimating neural networks. Computational Intelligence in Financial Engineering, 1996.
- [33] CHEN, X., RACINE J., SWANSON, N. R., Semiparametric ARX neural-network models with an application to forecasting inflation. IEEE Transactions on Neural Networks 12: 674-683, 2001.
- [34] BOLLERSLEV, T., Generalized Autoregressive Conditional Heteroskedasticity, Journal of Econometrics, 31, pp. 307-327, 1986.
- [35] BOLLERSLEV, T., CHOU, R. Y., KRONER, K. F., ARCH Modeling in Finance: A Review of the Theory and Empirical Evidence. Journal of Econometrics 52, pp. 5-59, 1992.
- [36] ENGLE, R. F. Autoregressive conditional heteroskedasticity with estimates of the variance of U. K. inflation, *Econometrica* 50, pp. 987-1008, 1982.
- [37] MEISSNER, G., e KAWANO, N. Capturing the volatility smile of options on high-tech stocks-a combined GARCH-Neural Network approach. *Journal of Economics and Finance* 25(3), pp. 276-293, 2001.
- [38] GONZALES, M. F., e BURGESS, N. Modeling Market Volatilities: The Neural Network Perspective. *European Journal of Finance* 3: pp. 137-157, 1997.
- [39] POSEDEL, P., Properties and Estimation of GARCH(1,1) Model, Metodoloski zvezki, Vol. 2, No.2, 243-257, 2005.
- [40] DING, Z., GRANGER, C.W. J. e ENGLE, R. F., A long memory property of stock market returns and a new model, Journal of Empirical Finance, 1, 83-106, 1993.
- [41] QI, M., e ZHANG, G. P. An investigation of model selection criteria for neural network time series forecasting. *European Journal of Operational Research* 132(3) (August), pp. 666-680, 2001.
- [42] CHAKRABORTY, K. Forecasting the behaviour of Multivariate Time Series Using Neural Networks. Neural Networks, vol. 5, pp. 961-970, 1992.
- [43] BARUCCI, E. e RENO, R. On measuring volatility and the GARCH forecasting performance, Article provided by Elsevier in its journal Journal of International Financial Markets, Institutions and Money. Vol.12, 2002.
- [44] ANDERS, W. Garch forecasting performance under different distribution assumptions, Journal of Forecasting, 2006.
- [45] RUMELHART, D. E., HINTON, G. E. e WILLIAMS, R. J., "Learning internal representations by error propagation", In Parallel Distributed Processing, vol 1, pp. 318-362. Cambridge, MA: MIT Press, 1986a.
- [46] RUMELHART, D. E., HINTON, G. E. e WILLIAMS, R. J., "Learning representations of back-propagating errors", Nature, vol. 323, pp. 533-536, 1986b.
- [47] RUMELHART, D. E. e McCLELLAND, J. L. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, 1986.
- [48] WERBOS, P. J. "Back-propagation: Past and future", Proceeding of International Conference on Neural Networks, San Diego, CA, 1, pp. 343-354, 1988.
- [49] ANDERSEN, T. J. e WILAMOWSKI, B.M. "A. Modified Regression Algorithm for Fast One Layer Neural Network Training", World Congress of Neural Networks, vol. 1, pp. 687-690, Washington DC, USA, 1995.
- [50] BATTITI, R., "First- and second-order methods for learning: between steepest descent and Newton's method", Neural Computation, vol. 4, no. 2, pp. 141-166, 1992.
- [51] CHARALAMBOUS, C., "Conjugate gradient algorithm for efficient training of artificial neural networks", IEEE Proceedings, vol. 139, no. 3, pp. 301-310, 1992.
- [52] HAGAN, M. T. e MENHAJ, M., "Training feedforward networks with the Marquardt algorithm", IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, 1994.
- [53] SHAH, S. e PALMIERI, F., "MEKA - A fast, local algorithm for training feedforward neural networks", Proceedings of International Joint Conference on Neural Networks, San Diego, CA, 3, 41-46, 1990.

- [54] SURATGAR, A. A., TAVAKOLI, M. B. e HOSEINABADI, A. Modified Levenberg-Marquardt Method for Neural Networks Training, Transactions on Engineering, Computing and Technology, 2005.
- [55] WILAMOWSKI, B. M., IPLIKCI, S., KAYNAK, O. e EFE, M. O. An Algorithm for Fast Convergence in Training Neural Networks, IEEE, 2001.
- [56] RECH, G. Forecasting with artificial neural network models. Working Paper Series in Economics and Finance 491, Stockholm School of Economics, 2002.