Comparing neural networks with other predictive models in artificial stock market

Jiří Krtek [1]

Abstract. A new way of comparing models for forecasting was created. The idea was to create a simple game in which the individual compared models would compete against each other. Therefore, inspired by the heterogeneous agent models an artificial market was created. Compared models act in the artificial market as forecasting strategies of agents who trade on the market. There are traded one risky asset paying a dividend and one risk-free asset in the artificial market. The way how agents trade (buy or sell risky asset) affects the price of risky asset, which in turn influences their expectations and therefore their subsequent decisions whether to buy or sell. Moreover, each agent can recalculate parameters of his strategy, if he is not satisfied with its performance. So the forecasting strategies and the artificial market evolve side by side. It remains only to add that the winning model is the one which earns the most money.

Keywords: Neural networks, ARMA, artificial market.

JEL classification: C23, C45, C53, G17
AMS classification: 91B26

1 Introduction

In this paper are compared ARMA models as classic delegates of linear models, neural networks as delegates of non-linear models and other simple predictive models. The forecasting models are not compared using real data as it is customary to do. Accuracy of forecasts or their standard deviations are not calculated. Instead, a new, unconventional method for comparing strategies was created. The models are compared in an artificial stock market. Traders, or agents, in the market use the aforementioned models as forecasting strategies. The best model among them is simply the one that earns the most money. It is also important to create an artificial stock market that has similar characteristics to the real market. The characteristics of the artificial market are therefore also studied and the artificial market is built to be conformable to the real one.

The structure of the market is inspired by several papers (see next section), most features are derived but several are products of own invention. The aim of this work is to create new way of comparing forecasting strategies and to compare primarily performance of linear and non-linear models.

2 Artificial stock market

The model which simulates market environment was inspired mostly by [2], [7], [6] and [8]. Two assets are traded in the market and no transaction costs. The first is a risk-free asset which is perfectly elastically supplied and has a piecewise linear rate of return r_t. The dynamics of the risk-free asset is product of own invention and is given by the following formula

$$r_{t+1} = r_t + \varepsilon_{r,t}\eta_{r,t}\Delta_r, \tag{1}$$

where $r_0 = 5 \times 10^{-5}$, $\Delta_r = 5 \times 10^{-6}$ are constants, $\varepsilon_{r,t}$ is alternatively distributed random variable ($\varepsilon_{r,t} = 1$ with probability $p_\varepsilon = 10^{-2}$, otherwise $\varepsilon_{r,t} = 0$) and $\eta_{r,t}$ is another alternatively distributed random variable ($\eta_{r,t} = 1$ with probability $p_\eta = 0.5$, otherwise $\eta_{r,t} = -1$). There is one more condition

[1]The Faculty of Mathemetics and Physics by Charles Univesiry in Prague, Department of Probability and Mathematical Statistics, Sokolovská 83, Praha 8, e-mail: krtek@karlin.mff.cuni.cz

stating that r_t is always positive, i.e. it holds that $r_t > 0$, $\forall t$. It should be noted that agents don't have to forecast the change of r_t because it is always known ahead. Thus r_t can be interpreted as short interest rate. Traders make a deposit with this safe short interest rate when they have redundant money.

The second asset is a risky stock whose price at time t is denoted by p_t. The risky stock can be divided into endlessly small pieces and pays dividend d_t at time t; the dividend is chosen (as in [6]) to follow the stationary AR process

$$d_{t+1} = \bar{d} + \rho(d_t - \bar{d}) + \varepsilon_{d,t+1}, \tag{2}$$

where $\rho = 0.95$ is chosen this way to provide a persistent process which is still stationary, $\bar{d} = 0.05$ is a constant and $\varepsilon_{d,t} \sim N(0, 10^{-2})$.

There are 8 groups of traders. Most of the traders predicts the future changes of price and future dividend itself. So they don't predict the future price of the risky asset itself, but a change from the last known value because it is assumed (in accordance with the real market) that the price process is not stationary. All traders forecast the future price (or price change) and dividend using their lagged values. As in [2] and many other heterogeneous-agent models, it is assumed that all traders maximize their expected utility function. It is also assumed that all traders have the same constant absolute risk aversion utility function U, with the same risk aversion parameter $\lambda = 1$. Let $W_{i,t}$ denote the wealth of trader i at time t. Furthermore, let $h_{i,t}$ be the number of agent i's shares at time t, then trader's goal is to maximize the expected utility at time $t+1$ given information up to time t over his number of shares, i.e.,

$$\max_{h_i,t} E[U(W_{i,t+1})|I_t] = E_{i,t}[U(W_{i,t+1})], \tag{3}$$

where I_t denotes the information set available at time t. Under the assumption of exponential CARA utility function and the Gaussian distribution of forecasts, the optimal number of shares at time $t+1$ is given by the following ratio

$$h^*_{i,t+1} = \frac{E_{i,t}[p_{t+1} + d_{t+1}] - (1+r)p_t}{\lambda \sigma^2_{i,t}}, \tag{4}$$

where $\sigma^2_{i,t}$ is the conditional variance of $p_{t+1} + d_{t+1}$ given I_t. Traders differ only in their forecasting strategies, i.e., the way they calculate $E_{i,t}[p_{t+1} + d_{t+1}]$. The conditional variance of $p_{t+1} + d_{t+1}$, i.e., the term $\sigma^2_{i,t}$ is estimated in the same way by all traders. Following [2], the estimate of conditional variance is given by

$$\sigma^2_{i,t} = (1-\theta)\sigma^2_{t-1|n} + \theta(p_t + d_t - E_{i,t-1}[p_t + d_t])^2, \tag{5}$$

where $\theta = 0.01333$, $n = 10$

$$\sigma^2_{t|n} = \frac{\sum_{j=0}^{n-1}[P_{t-j} - \bar{P}_{t|n}]^2}{n-1} \tag{6}$$

with

$$\bar{P}_{t|n} = \frac{\sum_{j=0}^{n-1} P_{t-j}}{n}. \tag{7}$$

2.1 Price evolution

The model of price evolution is inspired by [2] and [8]. According to the previous section, $h^*_{i,t}$ will be used for the desired number of risky shares at time t, while $h_{i,t}$ will denote the actual number of shares held. Let $b_{i,t}$, be the number of shares that trader i would like to buy and let $a_{i,t}$ be the number of shares which he would like to sell, i.e.,

$$b_{i,t} = \begin{cases} h^*_{i,t} - h_{i,t-1}, & h^*_{i,t} \geq h_{i,t-1}, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

$$a_{i,t} = \begin{cases} h_{i,t-1} - h^*_{i,t}, & h^*_{i,t} < h_{i,t-1}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Let N be the total number of traders. If $B_t = \sum_{i=1}^{N} b_{i,t}$ denotes the overall demand for the risky asset and $A_t = \sum_{i=1}^{N} a_{i,t}$ the overall supply of the risky asset, then the number of shares held by trader i

at time t is dependent on the number of shares he/she wanted to buy/sell. Traders' offers and demands are first sorted in descending order, subsequently the offers, respectively demands are satisfied as long as there are some demands, respectively offers. So the bigger volume of an offer/demand the bigger probability that it will be satisfied.

The price adjustment is based on the excess demand $B_t - A_t$, it is inspired by (but not completely adopted from) [2] and is given by

$$p_{t+1} = p_t \cdot [1 + \beta(B_t - A_t) \cdot \min(A_t, B_t) \,/\, Total_{shares}] + \eta_{p,t+1}\varepsilon_{p,t+1}, \tag{10}$$

where $\eta_{p,t}$ is alternatively distributed random variable, $\eta_{p,t} = 1$ with probability $p_p = 2 \times 10^{-2}$ and $\eta_{p,t} = 0$ otherwise, and $\varepsilon_{p,t}$ is the noise (i.i.d. random variables from $N(0, 4)$) added in order to represent other traders potentially present in the market, and β is some function of excess demand. Following [2], β was set to

$$\beta(B_t - A_t) = \begin{cases} \tanh(\beta_1(B_t - A_t)), & B_t \geq A_t, \\ \tanh(\beta_2(B_t - A_t)), & B_t < A_t. \end{cases} \tag{11}$$

The parameters β_1 and β_2 were set to 10^{-4} and 2×10^{-4} respectively. It is clear that the price does not change – except the possible noise – when there is no demand for risky asset or no supply of it which is in accordance with the real market.

2.2 Forecasting strategies

Traders want to forecast future change of the price and future dividends. As was previously stated, this work compares neural networks with more traditional models used for prediction. Traders therefore use neural network, ARMA or another model as their forecasting strategy. This is the only thing that differentiates one trader from another. Each trader uses the same forecasting strategy for both future price (or log-return, i.e. change of logarithm of the price) prediction and future dividend prediction.

The first group of agents uses AR models as their forecasting strategy for a prediction of the log-return and the dividend. Parameters of the AR models are estimated by conditional maximum likelihood (see [4] for details). Traders of the second type forecast the same future values with the aid of ARMA models estimated by conditional maximum likelihood again. Traders of the third and fourth type uses a feed-forward neural network with 3 and 4 layers respectively to forecast the same quantities. The Levenberg–Marquardt algorithm and backpropagation are used for searching for optimal parameters, see Section 2.3 for details. The fifth group of traders utilizes the Elman recurrent networks for forecasting the log-returns and dividends again. The Levenberg–Marquardt algorithm is used for searching for optimal parameters. For all NN models, the activation function of hidden neurons is tangent hyperbolic and the activation function of output neuron is linearity. For more information about feed-forward neural networks and Elman's networks see [5], [1], [9] or [3].

There are three more types of forecasting strategy. The first of them is strategy which employs an average of last realized values to predict the future price itself and future dividend. The second is so-called Naive forecasting strategy which can be viewed as special case of the previous. It simply predicts the future value by the last known. This forecasting strategy can be viewed as the basic for comparison with other more advanced strategies. Traders with the last strategy make random predictions of future price changes and future dividends.

Although some tables presented in Subsection 3.2 contain also results for traders utilizing the average and random forecasting strategy, these traders fulfill special roles in the artificial market. The purpose of random traders is to bring liquidity in the market. Traders employing average serve as an antipole to all the trend-chasers in the market, i.e. traders forecasting with AR or ARMA models or any kind of neural network, so they essentially fulfill the role of fundamentalists. They should force the price to return back. As these two types of traders should somehow stabilize the market (average) or should support trading (random) the number of traders with these two strategies is bigger than the number of other traders. Specifically, there is 280 random traders and 40 traders using averages, while each other group of traders has 4 only traders. Some tables in Subsection 3.2 don't contain results for the strategies fulfilling special roles because the main goal of this work is to compare NNs and ARMA models. Moreover in some cases it don't make sense to present results for these strategies.

2.3 Learning of traders

The learning is partially own invention and partially taken from [2]. A part of trader learning from [2] is adopted and part is left out. Every trader knows his wealth as well as the wealth of all other agents in this artificial market. After the given period of time $k = 400$, each trader counts the difference between his/her present wealth and wealth at time $t - k$, i.e., $W_{i,t} - W_{i,t-k}$. This indicates how much money the trader has earned or lost. Everyone consequently gets a rank $R_{i,t}$ according his difference of wealth. The quantity

$$r_{i,t} = \frac{R_{i,t}}{N} \tag{12}$$

is then the probability that agent i recounts parameters of his forecasting strategy due to *pressure of society*. The smaller his rank the smaller the probability that he learns new parameters of his strategy (because he is satisfied with the current parameters). Each trader can also recounts his strategy on the basis of growth rate of his wealth over the previous period. Therefore, even if trader i does not recount his strategy because of *pressure of society*, there is still chance that he will do it for a different reason. Specifically, because he is not satisfied with himself. Let

$$\chi_{i,t}^{k} = \frac{W_{i,t} - W_{i,t-k}}{|W_{i,t-k}|} \tag{13}$$

be trader i's growth rate of wealth over the previous period. It in fact measures how effective trader i's strategy was. Then

$$s_{i,t} = \frac{1}{1 + \exp\{\chi_{i,t}^{k}\}} \tag{14}$$

is the probability that trader i will recount his strategy because of its low efficiency. If trader i recounts parameters of his strategy at the end of the tth day, he will learn on the data of length $m = 300$ (common to all traders).

Except aforementioned learning, the traders using the forecasting strategies based on feed-forward neural networks can recount the parameters of their strategies each time after the new log-return and the new dividend amount is known. In that time, they already know the correct log-return and dividend they forecasted before so they can update the parameters of their forecasting strategies simply by backpropagation with momentum.

3 Simulation

Fourteen simulations with the same setups were performed. Traders always learn at the beginning of the simulation on simulated data of length 150. The dividend process was generated according to (2). For price, its mean was set to 1000 and the log-returns were taken from GARCH(1, 1) process with Gaussian innovations. Traders estimated parameters of their strategies on these simulated data. After this initial learning the traders were given initial amount of money (800) and shares (0.15). The last values of simulated dividend and price served as initial values to the market simulation. Traders were let to trade for 200 time steps. After this period all agents recalculate (learn) the parameters of their forecasting strategies, the realized price series (200 time steps) were discarded and the same initial amount of money and shares were given to all of them. The same procedure was applied again, i.e. 200 time steps of trading, recalculation of parameters, price series discarded and initial money and shares allocation. Then the competition (described in this paper) having aforementioned 1170 time steps started. The previous two procedures were performed in order that agents learn the parameters of their strategies on data arisen from their trading.

3.1 Verifying stylized facts

To get more realistic results, the constructed market should have similar characteristics as the real markets have. The characteristics of real markets (also called stylized facts) are for example heavy tail distribution of returns and their volatility clustering or that prices follow random walk. As can be seen from the numbers presented below, some stylized facts hold in the artificial market and some not. The results of testing are presented in the Table 1. The first tested stylized fact (column *adf*) was that the price series have a unit root. This hypothesis was tested by augmented Dickey–Fuller test. The Kwiatkowski–Phillips–Schmidt–Shin test (column *kpss*) was used to test the second stylized fact – the null about level stationarity of log-returns. The hypothesis that the log-returns are uncorrelated in time – third stylized fact (column *B–P*) – was tested by Box–Pierce test. The last stylized fact tested was that log-returns have distribution with fat tails. It was tested whether the log-returns have Gaussian distribution and also their kurtosis was calculated. The goal of this analysis was to show that the hypothesis about Gaussian distribution of log-returns in the artificial market is rejected. The test used for this purpose was Jarque–Bera test, the column *J–B*. The kurtosis indicates fat tails when it is higher than 3 (the kurtosis of Gaussian distribution) which is the reason why it was calculated, column *kurt*. The last row of the Table 1, named *summary*, contains proportion of null rejections on 5% confidence level for first four columns. For the last column (*kurt*) it contains the proportion of excess kurtosis, i.e. the case when the calculated kurtosis is higher then 3.

simulation no.	adf	kpss	B–P	J–B	kurt
1	0.65	0.10	~ 0	~ 0	5.0
2	0.95	0.06	~ 0	~ 0	4.7
3	0.87	0.04	~ 0	~ 0	3.9
4	0.69	0.09	~ 0	~ 0	4.6
5	0.58	0.10	~ 0	~ 0	4.2
6	0.77	0.10	~ 0	~ 0	3.8
7	0.93	0.09	~ 0	~ 0	4.4
8	0.48	0.03	~ 0	~ 0	3.6
9	0.39	0.01	~ 0	~ 0	4.9
10	0.92	0.10	~ 0	~ 0	4.4
11	0.70	0.10	~ 0	~ 0	5.6
12	0.53	0.10	~ 0	~ 0	3.6
13	0.95	0.01	~ 0	~ 0	3.9
14	0.80	0.10	~ 0	~ 0	3.7
summary	0 %	28.6 %	100 %	100 %	100 %

Table 1: Results of stylized facts testing.

3.2 Comparing strategies

When comparing agents, the criterion of their success was the amount of money they earned. Specifically, it was the ratio of last and initial wealth of the particular traders. The results of the 14 simulations are presented in the Table 2 below. The very interesting phenomenon is that the traders utilizing Elman networks won every simulation with only one exception (simulation no. 11) where the traders with ARMA models as their forecasting strategy won. The best trader in all simulations was utilizing Elman networks. So the 11th simulation was won by traders using ARMA models in average, however they all were beaten by one of the traders forecasting with Elman networks. Except the presented results other more common statistics were also studied, e.g. root mean square error and sign test of price predictions. Traders with Elman networks as forecasting strategy had the lowest RMSE in most of simulations. Regarding the sign test, the Elman netowrks were the best strategy in the direction estimation in ten out of the 14 cases. The results are also interesting in the following feature. Although the FF1L and FF2L strategies had much better results with respect to the sign test than AR models, the results of all these three strategies are pretty the same for the ratio of final and initial wealth (see Table 2). This might seem quite strange, but it is exactly the reason why the competition, i.e. the market simulation, was performed.

4 Conclusions

It was created an artificial market in order to compare forecasting strategies in it. The purpose of it was to show a new way of comparing strategies. Except one stylized fact, the requirement of unpredictable

sim. no.	AR	ARMA	Avg	Elman	FF1L	FF2L	Naive	Rnd
1	1.035	1.107	0.973	1.110	1.061	1.043	1.077	1.068
2	1.030	1.117	0.965	1.161	1.052	1.067	1.081	1.066
3	1.051	1.146	0.983	1.178	1.024	1.031	1.044	1.058
4	1.016	1.088	0.943	1.104	1.032	1.051	1.052	1.042
5	1.057	1.183	0.977	1.189	1.030	1.048	1.044	1.059
6	1.053	1.110	0.974	1.202	1.056	1.035	1.038	1.055
7	1.031	1.143	0.955	1.177	1.017	1.032	1.055	1.050
8	1.037	1.120	0.958	1.164	1.045	1.049	1.049	1.048
9	1.032	1.093	0.960	1.187	1.044	1.047	1.056	1.057
10	1.060	1.121	0.993	1.191	1.091	1.081	1.087	1.084
11	1.038	1.142	0.970	1.102	1.039	1.044	1.074	1.066
12	1.044	1.130	0.970	1.186	1.036	1.020	1.038	1.050
13	1.040	1.103	0.954	1.186	1.045	1.039	1.041	1.043
14	1.043	1.105	0.973	1.231	1.021	1.034	1.035	1.047
average	1.040	1.122	0.968	1.169	1.042	1.044	1.055	1.057

Table 2: The average ratio of final and initial wealth for particular groups of traders. AR = AR models, ARMA = ARMA models, Avg = moving average, Elman = Elman networks, FF1L = feed-forward NN with 1 hidden layer, FF2L = feed-forward NN with 2 hidden layers, Naive = naive strategy, Rnd = random forecast

(or at least uncorrelated) log-returns, the artificial market seems to have similar characteristics as the real one. The goal of this paper was to compare the NNs and ARMA models. The Elman networks completely dominate in the presented contest "the winner is the one who earns the most money" as well as in the customary statistics (linked to the outputs of the market simulation) such as RMSE and sign test.

Acknowledgements

I would like to thank my supervisor Miloslav Vošvrda for helpful comments. The work was supported by grant of GA ČR P402/12/G097. The Bernard Bolzano Endowment Fund is also greatly acknowledged.

References

[1] Boden, M.: *A Guide to Recurrent Neural Networks and Backpropagation.* In the Dallas project, 2002.

[2] Chen, S., and Yeh, C.: *Genetic Programming in Agent-Based Artificial Markets: Simulations and Analysis.*

[3] Dorffner, G.: Neural Networks for Time Series Processing, *Neural Network World* **6** (1996), 447–468.

[4] Hamilton, J. D: *Time Series Analysis.* Princeton University Press, 1994.

[5] Haykin, S.: *Neural Networks: A Comprehensive Foundation.* Prentice Hall, 1999.

[6] LeBaron, B.: *Building the Santa Fe Artificial Stock Market.* Brandeis University, 2002.

[7] Matassini, L., and Franci, F.: On Financial Markets Trading, *Physica A: Statistical Mechanics and its Applications* **3–4** (2001), 526–542.

[8] Palmer, R. G., and Arthur, W. B., and Holland, J. H., and LeBaron, B., and Taylor, P.: Artificial Economic Life: A Simple Model of a Stockmarket, *Physica D* **75** (1994), 264–274.

[9] Šíma, J., and Neruda, R.: *Teoretické Otázky Neuronových Sítí.* Matfyzpress, Praha, 1996.