INFORMATION
SYSTEMS IN
MANAGEMENT

Information Systems in Management (2016) Vol. 5 (4) 508−519

USING NEURAL NETWORKS TO PREDICTION ON WARSAWS STOCK EXCHANGE

KRZYSZTOF LICHY

Institute of Information Technology, Lodz University of Technology

The paper describes an experiment consisting of the application of artificial intelligence algorithms in the processes of predicting the stock market. A special tool was developed to evaluate whether artificial neural networks can predict stock market behavior. The aim of this paper was also to test how neural networks tapping trivial and easily attainable input data perform in an environment which is both complex and difficult to predict.

Keywords: Artificial Intelligence, Neural Networks, Stock Market

1. Introduction

To answer the question of whether a neural network can, at least to a limited extent, predict broadly defined stock market behavior, we focused on predicting price increases and detecting buy-sell signals. The developed system provides an easy way of studying artificial intelligence mechanisms. The presented solution enables the generation of training and test datasets based on historical data and to create a variety of neural network structures. Thus, observations and inferences can be made about correlations between the network's structure and outcomes. The system also allows the use of various types of learning algorithms. Experiments were performed to verify whether the application can generate profitable decisions [1], [2], [7][10]. The main aim of this work is to test the capabilities of the application developed by small group of programmers.

2. The idea of application

Tests of the developed neural network (i.e., the application) were carried out with the following input data:

- network's structure;
- activation function;
- input data/number of inputs;
- output data/number of outputs;
- impact of the exchange rates of major currencies.

In accordance with the research assumptions, the application conducts analysis of historical data as a basis for the generation of training data. A variety of neural network structures can be created and trained, enabling a range of different tests. In terms of non-functional requirements, the focus is on simple and intuitive access to data and ease of use of the application. Therefore, the front end of the application is designed as a web interface, while the calculation module is a separate component, with the data stored in a database or as text files, as required. To sum up, the system consists of three elements: a web application, a data module, and a calculation module (neural network).

2.1. Generation of training data

One of the most important functionalities of the application module is the generation of training and test data. Historical listings for the Polish stock exchange are taken from the website gpwinfosfera, whereas foreign company listings can be uploaded from Yahoo Finance. The expected data format is CSV.

The application recognizes the following types of input data: opening price; closing price; the highest price of the day; the lowest price of the day; volume; difference between the highest and lowest prices of the day; difference between the closing and opening prices; increase from the previous day; increase from a specific day; candlestick.

These data can be freely added to the training and test datasets. The application allows the user to select any number of input data. Additionally, the above data types can be used to calculate an average or a price increase over a period of several days. The calculated values can be fed as an input to the neural network.

The application produces the following types of output data: value; increase; simplified increase; simple trend; trend of averages.

The first type of output in the dataset is a value determined for a date following the days for which input data were collected. The output may consist of any input data type. The same is true of the "increase" or "simplified increase," but in these cases what is calculated is a change from a prior date. "Simplified increase" differs from "increase" in that it returns 1 for an increase -1 for a decrease.

The second category of output data consists of trends. This type of output data represent changes in trends. The first variant calculates differences in the company's share prices over a given interval, while the second variant computes differences between values averaged for a number of days.
For both types of trends, the output is as follows:

$$\varphi(e) = \begin{cases} 1 & if\ in\ the\ future\ following\ a\ large\ increase\ in\ value \\ 0 & if\ in\ the\ future\ shall\ be\ negligible\ increase\ in\ value \\ 1 & if\ in\ the\ future\ there\ is\ a\ wide\ discount\ value \end{cases} \tag{1}$$

In this way, neural networks can be trained when to sell or buy stocks. The actual sensitivity of the trend functionality is determined by two parameters: the number of days over which an increase is to be calculated and the minimum percentage change for trend recognition. The number of signals is inversely proportional to the value of the percentage parameter. The number of days determines the length of the time interval to be analyzed. Short-term players will use low numbers, while medium- and long-term ones will opt for higher numbers [1], [5], [6].

2.2. The calculation module

The purpose of this module, based on the FANN library, is to create and operate neural networks. It is the brain of the whole system. Cooperation with this module requires running the appropriate program. Users communicate with the module by means of command line arguments and a standard output. The output data are separated by newlines. A detailed description of the functionalities, as well as inputs and outputs of the programs, can be found in [3].

2.3. The data module

The purpose of this module is to store application data both in text files and in a MySQL database. The essential information includes:

– Neural network structures;

- training data;
- test data;
- available financial instruments;
- tests of the trained networks;
- data generated during the training and testing of networks:
    - error as a function of the number of epochs;
    - error within the test dataset;
    - comparison of network results with test data;
    - comparison of network results with training data.

Out of the above-mentioned information, only shares, tests and related data are stored in the database alone. Other data are stored in two locations. Basic information is contained in the database, while detailed structures in text files. In the case of neural networks, a complete description of all the features and properties necessary to recreate their entire structure by the calculation module is stored in files. The test and training data are stored in a similar way.

3. The test procedure

All tests were performed on historical data, which made it possible to evaluate prediction accuracy. To facilitate research and eliminate spurious results, tests were carried out for the most liquid stocks from the WIG20 index. Of greatest interest were changes in the mean squared error over successive epochs. The next step included determination of the impact of various factors on prediction effectiveness. In this case, research was mainly based on well-trained networks, with the conclusions inferred from the test data. The objective was to verify whether the developed tool can provide accurate predictions concerning future stock market behavior. In addition, result reproducibility was assessed [9].

Table 1. Relationship between the mean squared error and the number of epochs for 4 neurons in the hidden layer

Epoch	Mean square error
1	0.18579823
50000	0.08787858
100000	0.08657525
150000	0.08592853
200000	0.08557729

Table 2. Average difference between training data value and neural network output for 4 neurons in the hidden layer

Name	Average Difference Values
Alior	0.49728983
Lotos	0.38832449
LPP	0.61485993
Tauron	0.33484827

Initially, it was investigated how different properties of neural networks affect the quality and speed of learning. The tests were conducted on 4 selected companies (Alior Bank, Tauron PE, LPP, Grupa Lotos) listed on the WIG20 to minimize spurious results.

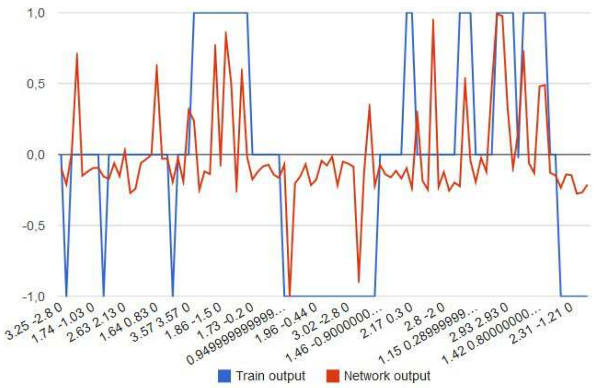

Figure 1. Training data for Alior Bank shares and neural network output for 4 neurons in the hidden layer

3.1 Evaluation of learning coefficients and algorithms for a simple network structure

Table 3. Relationship between the mean squared error, training algorithm, and learning coefficient

Name	Coefficient of Learning	Mean Square Error	Name	Coefficient of Learning	Mean Square Error
FTB	0.1	0.060799	QP	0.1	0.053251
FTB	1	0.022753	QP	1	0.026129
FTB	0.2	0.041383	QP	0.2	0.043889
FTB	0.5	0.027667	QP	0.5	0.028099
FTB	0.7	0.026295	QP	0.7	0.026413
FTI	0.05	0.115374	TRP	1	0.028963
FTI	0.1	0.112378	TRP	0.2	0.028964
FTI	0.2	0.102865	TRP	0.5	0.028965
FTI	0.5	0.085507	TRP	0.7	0.028966

Tests were conducted for a network with 20 neurons in the hidden layer. Each algorithm was tested repeatedly for different learning coefficients [4].

FTI (*FANN TRAIN INCREMENTAL*) is a backpropagation algorithm with the weights updated after each training pattern. FTB (*FANN TRAIN BATCH*) is a backpropagation algorithm with data updated only once per epoch. TRP is an alternative name for an *RPROP* algorithm, and QP stands for a *QUICKPROP* algorithm.

3.2. Evaluation of the learning algorithm for a four-layer neural network

Tests were conducted for a network with 2 hidden layers. The network structure was as follows: 3-20-8-1.

Table 4. Relationship between the mean squared error and the training algorithm for a four-layer network

FTB	0.005206
FTI	0.084318
QP	0.003559
TRP	0.014549

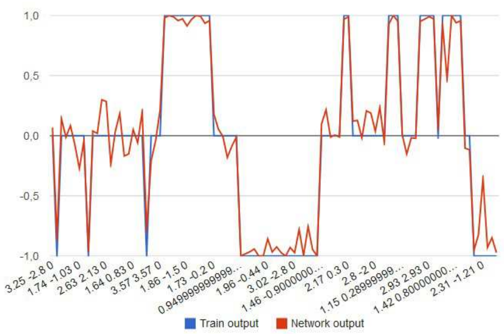

Figure 2. Training data for Alior Bank shares and results for the best-trained network

4. Analysis of confidence

The objective of this section was to examine the profits or losses that the decisions of the neural networks would imply if their capacity were trusted to a given degree. Tests were performed on randomly selected neural networks as well as tests from previous analysis. The sale coefficient stands for the decline in price at the output interpreted as a sell signal, while the buy coefficient stands for the rise in price at the output interpreted as a buy signal. While computing profits, the costs of transactions and intermediaries were not taken into account. Obviously, they should be considered in real-life investing [4][8][11][12].

Table 5. Examples of profits

Name	Profit
Alior	2,7%
Tauron	–2,2%
Lotos	–1,6%

5. Effects of tests

5.1. Effect of network structure on learning process

The objective was to select the best network structure for predicting stock market processes. In the first variant, it is perfectly clear that deviations were huge and errors high, but the neural network's analysis of the input data was still going in the right direction, as can be seen on figure 1. The neural network was supposed to indicate when to buy or sell stocks. The greater the number of neurons in the hidden layer, the more consistent the results with the training data. With 20 neurons in the hidden layer, the chart became increasingly similar to the original. At 25 neurons, learning efficiency dropped, which is quite surprising. It was only after another layer had been added that the results improved. Still, the neural network results did not overlap with the training data in the chart. It may be expected that learning efficiency is affected by other factors, examined in the next section.

As stated above, factors influencing the learning process other than network structure should also be examined. Therefore, differences in the mean squared errors for various learning factors and algorithms were investigated. The test confirmed the theoretical assumption that the higher the learning coefficient, the faster the learning process, and thus the smaller the mean squared error. However, the use of an excessively high value is bound to impair the training process.

Although it could be assumed that the training algorithms would affect only the speed of learning and at 200,000 epochs no sizable benefits would be perceptible, to our surprise it turned out that those algorithms had a relatively large impact on training effectiveness. The traditional backpropagation algorithm was found to be the most difficult to train. This may be due to the fact that weights were adjusted following every single piece of training data. By using the RPROP algorithm, even an eight-fold performance gain was achieved. However, the best solutions were the backpropagation algorithm with weights calculated after every epoch and QUICKPROP, which led to a sixteen-fold rate increase. These tests allowed the specification of the most efficient network training methods.

5.2. Selection of input data

This section was designed to answer one of the crucial questions posed in this paper: what datasets can unambiguously identify share price increases or decreases? While it is difficult to infer from the tests whether the investigated data conclusively determine the results, it was shown how changes in input data improve the learning process and reduce errors.

It can be immediately noticed that differences between the closing and opening prices and between minimum and maximum daily prices were forecast quite correctly. This to some extent vindicates basic technical analysis involving candlestick charts. An additional parameter which improves the network's learning is the number of days, i.e., the number of times the data from previous days are to be replicated. The effect of the above parameter is clearly manifested by the first and sixth training datasets. Adding the same input data, but taken from three days, resulted in a nine-fold reduction of the error.

On the other hand, the introduction of the volume parameter led to unexpected results. While volume is considered one of the crucial components of technical analysis, it greatly impaired the quality of neural network training. This may be due to the fact that this parameter is too ambiguous and confusing. It might be useful if a larger amount of input data were used. Another unsuccessful solution was for data to be largely based on averages and increases. This can be explained by the fact that averages are slightly delayed, i.e., changes in certain parameters are not immediately evident. Only major changes in data affect averages, but by the time that happens it may already be too late to react. This problem can also arise from misapplication of averages.

This section explored the basic parameters of neural networks and their learning processes. It can be concluded that although the network structure, learning parameters, and training algorithms are very important, the main problem with applying artificial neural networks to stock exchange analysis is the selection of input and output data. The most difficult task is to find a set of data that would unambiguously determine whether the share prices are going to rise or decline. The present study successfully identified data which enable high quality and effective network training. The subsequent section of the paper examines whether such training ensures accurate prediction of future stock behavior.

5.3. Effect of neural network training on prediction quality

The purpose of this section was to examine whether the quality of network training affects the prediction of stock market behavior and whether a focus on the learning process would improve prediction accuracy. Analysis of results shows that indeed the better a given network has been trained, the smaller the error for the test set. However, it is hardly reassuring that the smaller the error for the training data, the larger the error for the test dataset. It is immediately clear from results that the mean squared error increases for data concerning the future. However, initially the error is small enough to offer an opportunity for accurate predictions for several days ahead.

The result show that the network returns a correct trend for the coming days, but in an unreliable and haphazard manner. Probably accurate predictions are possible, since the results overlap increasingly. This case could actually generate profits in a real investment setting.

5.4. Currencies

The objective of this section was to determine whether the main currencies of the world market included into the training and test data improve prediction and the learning process. The results are not reassuring as the introduction of currencies into the training dataset only aggravated confusion. Apparently, currency exchange rates are not always correlated with share prices and often change for a variety of reasons which have no effect on the actual prices of the tested shares. In general, this factor led to disappointing results, but some share prices were predicted extremely accurately. The network reflected the actual data with only a small error. However, due to considerable confusion, it is hard to determine whether exchange rates are helpful in estimating future buy-sell signals. It should be borne in mind that the test generated few positive results; in most cases the quality of both prediction accuracy and of the learning process decreased. Not knowing the future, it is impossible to decide whether the introduction of the exchange rates of the dollar, euro, or franc enable more accurate forecasts. Therefore, the application of currencies is not conducive to better prediction outcomes for stock market behavior [4].

5.5. Analysis of confidence

The objective of this section was to verify how much one can trust signals from neural networks and determine whether a profit can be made using the developed tool. Generally speaking, it was shown that the future can indeed be predicted and that neural networks can lead to decent profits of up to 8%. Notably, well-trained networks brought higher profits, while those with inferior training tended to bring losses. In addition, one can see that the best results were achieved by choosing coefficients with a focus on low purchase prices. The opposite is true in the case of focusing on predicted sale prices and fast purchases. This gives rise to more transactions, but they are far less profitable. The network buys almost anything and sells it surprisingly quickly. In addition, a minor imbalance due to lower quality of the learning process or an ambiguous set of inputs may cause wrong predictions and entail losses. Therefore, it can be concluded that only predicted increases can be trusted, and sell signals should be immediately acted upon. Although partly confirmed by this study, it is not completely true. As it was shown the buy-sell signals often appeared in excess, confusing the predictive power of the neural networks. Although the initially forecast directions of change were correct, numerous inflection points occurred in the case of both upward or downward trends. Hence, a large number of transactions. Moreover, it is very likely that the networks would have provided better results if they had been trained to reject momentary fluctuations. This study showed that prediction is possible, but it should be wisely used and that at this stage it should be treated as confirmation rather than a guide for real-life stock exchange transactions.

5.6. Aim achieved

This section of the study was completed quite successfully. It was shown that there are cases in which an application can provide a relatively accurate approximation of future prices, and, most importantly, generate profits. It was also demonstrated that it is worthwhile to focus on high-quality network training because well-trained networks provide better prediction results than those trained less effectively. The inclusion of the relevant currencies in the dataset caused come confusion and failed to improve the results; it even made them worse. However, thanks to the introduction of an additional parameter some networks can learn with fewer errors and provide more precise predictions. In general, there are many indications that prediction is possible, but there is also a certain dose of residual disbelief. The question also arises as to whether or not the generated predictions are spurious. There are still too many buy-sell signals in the charts. The analysis demonstrated the extent to which one can trust the networks. Indeed, while predictions about stock market behavior can be made, they should be treated with caution.

6. General conclusions

The primary objective of this study was to verify whether artificial intelligence mechanisms are able to successfully predict stock market behavior. A tool was developed and tests performed to determine whether neural networks are viable candidates for forecasting future stock exchange listings.

The first part of the study explored how the various technical properties of neural networks influenced their learning processes, which proved to be the key to accurate prediction of short-term stock market behavior. Good training enabled neural networks to predict the future. The following factors proved significant at the stage of learning:

- structure of the network – the greater the number of layers and neurons, the better the results;
- learning algorithm – can increase the rate of the learning process and eliminate stalling at local minima.

- Selecting the appropriate functions of prediction activation and the right combination of data values enables networks to learn to clearly recognize situations and make buy or sell decisions. The tests performed for this paper led to the following conclusions:
  - Intraday stock price data, such as closing, opening, minimum, and maximum prices, are useful.
  - Although volume is commonly believed a crucial factor, it confuses neural networks. Perhaps it should be operationalized in a different form, e.g., logarithmic.
  - Data applied in basic technical analysis involving candlestick charts are helpful.
  - The same values should be used over several days for the network to recognize pattern formation.
  - The theory of moving average crossovers does not work well for neural network predictions.
  - Value increases may be used as an additional element, but alone they cannot identify a price increase or decrease.
  - Currencies generally do not facilitate the process of network learning or predicting share prices.

In summary, input data are the most important factor for accurate prediction of stock market behavior. The selection of an appropriate set improves learning, which, as it has been proven, leads to better predictions.

Prior to conducting forecasts, it would be useful to check the past results for the examined company and select the factors for which a given network brings the best learning and forecasting results. Each company is different, attracts different investors and depends on different determinants. Therefore, before taking a risk, it is recommended to examine which factors facilitate effective network operation and what parameters impact price changes of the shares. There is no golden mean in stock market speculation, which carries an inherent element of risk. Even the most seasoned investor has the right to be wrong. Therefore, one should not demand AI mechanisms to be infallible, either.

Despite the fact that the developed networks were relatively simple and the training data straightforward, fairly accurate prediction results were achieved. This is promising and gives grounds for further reflection and research in this field, especially in terms of experimenting with different sets of inputs. In addition, the output of neural networks could also be subjected to scrutiny, which was beyond the scope of this work, as those data may hold a large potential for improving the learning and prediction processes. Yet another issue worth examining are correlations between different shares or commodities. It would also be useful to evaluate how common opinions about companies affect their performance in the stock exchange, as the stock market is primarily governed by crowd psychology stimulated by rumors, subjective opinions and media news. Therefore, the influence of these factors on the learning process should also be explored.

In conclusion, artificial intelligence mechanisms stand an excellent chance of succeeding on the trading floor, but further tests must be carried out to develop a good and reliable investment system. There is no doubt that such solutions should be considered for predicting stock market behavior [13].

REFERENCES

[1] Bronw C. (2012) *Technical Analysis for the Trading Professional,* Mc Graw Hill

[2] Elder A. (2012) *Zawód inwestor giełdowy - Psychologia rynków. Taktyka inwestycyjna. Zarządzanie portfelem*, Wolters Kluwer

[3] Fast Artificial Neural Network Library (FANN) http://leenissen.dk

[4] Łędzewicz M (2014). *Mechanizmy sztucznej inteligencji w predykcji zachowań giełdowych*, Politechnika Łódzka

[5] Wanjawa B. A (2014) *Neural Network Model for Predicting Stock Market Prices,* LAP LAMBERT Academic Publishing

[6] Witkowska D. (2002) *Sztuczne sieci neuronowe i metody statystyczne: wybrane zagadnienia finansowe,* C.H. Beck Łódź

[7] Witkowska D. (1990) *Sztuczne sieci neuronowe w analizach ekonomicznych*, C.H. Beck Łódź

[8] Zirilli J.S.(1996) *Financial Prediction Using Neural Networks,* Wiley

[9] Azoff M.E. (1994) *Neural Network Time Series: Forecasting of Financial Markets* Wiley

[10] *http://aitech.pl/sphinx/pakiet-sphinx/* Aitech artificial intelligence laboratory

[11] Gately E. (1995) *Neural Networks for Financial Forecasting* Wiley

[12] Trippi R.R.(1992) *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance,* Probus Pub Co

[13] Pring M.J. (2015) *Martin Pring's Introduction to Technical* Analysis, Mc Graw Hill Education