

STOCK PRICE PREDICTION: KOHONEN VERSUS BACKPROPAGATION

Alexey Zorin

*Technical University of Riga
Decision Support Systems Group
1 Kalkyu Street, Riga LV-1658, phone: 371-7089530, LATVIA
E-mail: alex@ru.lv*

This paper describes the application of two different neural network types for stock price prediction. The prediction is carried out by Kohonen self-organizing maps and error backpropagation algorithm. Both experimental networks deal with price change intervals in contradiction to precise value prediction. The results are presented and its comparative analysis is performed in this paper, as well as a short discussion on both neural network architectures and learning algorithms.

1 Introduction

Financial predictions must deal with precise data, therefore in most cases stock price prediction is carried out by forecasting future point of the time series. This is one of the most popular approaches for many problem domains. However, the precise knowledge of the “exact” future value is not always necessary (forecasting cannot avoid errors and it is impossible to predict future value exactly). Many trading strategies can easily be based on the price change intervals, which can be treated as classes for neural network. In this case we face classification problem, that can be solved by means of ever Kohonen self-organizing maps or backpropagation networks. The number of classes depends on our needs – if we want to know price changes more precisely, we should use more intervals or classes.

Neural networks are very sophisticated modelling techniques, capable of modelling extremely complex functions. In particular, neural networks are non-linear. For many years linear modelling has been the commonly used technique in most modelling domains, since linear models had well-known optimisation strategies. Where the linear approximation was not valid (which was frequently the case) the models suffered accordingly. Neural networks also keep in check the curse of dimensionality problem, which helps to model non-linear functions with large numbers of variables.

The successful applications of neural networks [1], [4], [5] can also be attributed to its ease of use. Neural networks learn by example. The neural network user gathers representative data, and then invokes training algorithms to automatically learn the structure of the data. Although the user does need to have some knowledge of how to select and prepare data, how to select an appropriate neural network, and how to interpret the results, the level of user knowledge needed to successfully apply neural networks is much lower than would be the case using (for example) more traditional statistical methods.

The paper is organized as follows. Section 2 provides the discussion about Kohonen and backpropagation network architecture. Section 3 covers experimental set-up, which includes the data set description and pre-processing, as well as network architecture design for this particular problem. Section 4 shows prediction results of both networks and compares the results of implying trading strategy for each methodology. The paper concludes with comments on possible future work in the area and some conclusions.

2 Theoretical basics of neural networks

A neural network is a set of interconnected simple processing elements, or neurones. Neural networks are potentially useful for studying the complex relationships between inputs and outputs of a system. There are two neural network models investigated in this research: backpropagation networks and Kohonen self-organizing maps. There are three major steps in the neural network-based forecasting [3]: pre-processing, architecture and post-processing.

In pre-processing, information that could be used as the inputs and outputs of neural networks are collected. These data are first normalized or scale in order to reduce fluctuations and noise (however, this does not guarantee good results). In architecture, a variety of neural network models that could be used to capture the relationships between the data of inputs and outputs are built. Different models and configurations using different training, validation and forecasting data sets are used for experiments. The best model is then selected

for use in forecasting. Finally, in post-processing, different trading strategies are applied to the forecasting results to show the capability of neural network prediction.

2.1 Backpropagation networks

A multilayer feedforward network with an appropriate pattern of weights can be used to model some mapping between sets of input and output variables. Figure 1a shows an example of feedforward network architecture, with three output units and one hidden layer, which can be trained using backpropagation. The shaded nodes in Figure 1a are processing units. The arrows connecting input and hidden units and connecting hidden units and the output units represent weights.

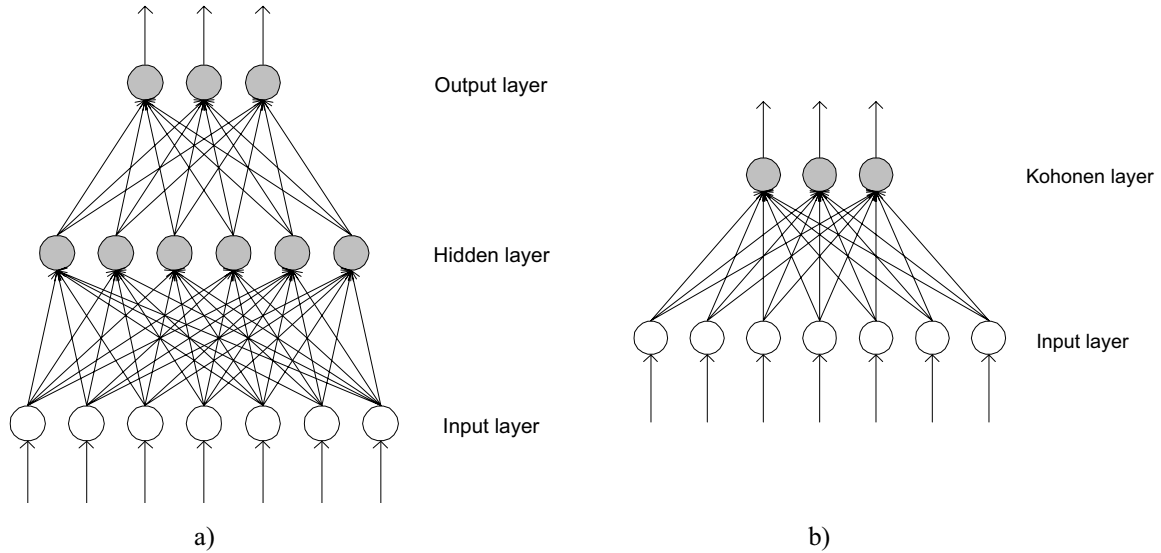


Figure 1. The architecture of two types of networks:
a) Backpropagation network, b) Kohonen self-organizing map.

The backpropagation learning algorithm [2],[6],[8] is formulated as a search in the space of the pattern of weights, W , in order to find an optimal configuration, W^* , which minimizes an error or cost function, $E(W)$. The pattern of weights will then determine how the network will respond to any arbitrary input. The error or cost function is defined by (1):

$$E = \frac{1}{2} \sum_i \sum_p (t_{ip} - o_{ip})^2 \quad (1)$$

This function compares an output value o_{ip} to a desired value t_{ip} over the set of p training vectors and i output units. The gradient descent method is used to search for the minimum of this error function through iterative updates:

$$W(k+1) = W(k) - \eta \nabla E \quad (2)$$

where η is the learning rate, and ∇E is an estimate of the gradient of E with respect to W .

The algorithm is recursive and consists of two phases: forward-propagation and backward-propagation. In the first phase, the input set of values is presented and propagated forward through the network to compute the output value for each unit. In the second phase, the total-squared error calculated in the first phase is propagated from the output units to the input units. During this process, the error signal is calculated recursively for each unit in the network and weight adjustments are determined at each level. These two phases are executed in each iteration of the backpropagation algorithm until the error function converges.

2.2 Kohonen self-organizing maps

In this section we consider self-organizing networks. The main difference between them and conventional models is that the correct output cannot be defined a priori, and therefore a numerical measure of the magnitude of the mapping error cannot be used [6]. However, the learning process leads to the determination of well-defined network parameters for a given application.

The self-organizing networks assume a topological structure among the cluster units. This property is observed in the brain, but is not found in other artificial neural networks [2]. There are m cluster units, arranged

in a one- r two-dimensional array: the input signals are n -dimensional. Figure 1b shows architecture of a simple self-organizing network, which consists of input and Kohonen or clustering layer. The shadowed units in the Figure 1b are processing units. This simplified network may cluster the data into three classes, but in the real problem domains one clustering unit for each class is not enough, therefore we should understand each Kohonen layer neurone in the Figure as a number of units (cluster of neurones).

When a self-organizing network is used, an input vector is presented at each step. These vectors constitute the “environment” of the network. Each new input produces an adaptation of the parameters. If such modifications are correctly controlled, the network can build a kind of internal representation of the environment.

Consider the problem of charting an n -dimensional space using a one-dimensional chain of Kohonen units [6]. The units are all arranged in sequence and are numbered from 1 to m (see Figure 2).

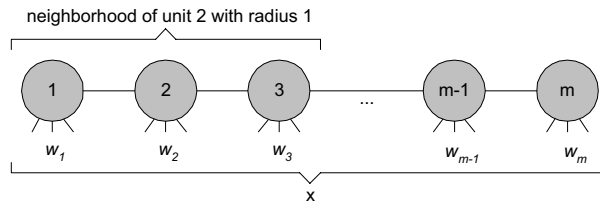


Figure 2. A one-dimensional lattice of computing units.

The n -dimensional weight vectors w_1, w_2, \dots, w_m are used for the computation. The objective of the charting process is that each unit learns to specialize on different regions of input space. When an input from such a region is fed into the network, the corresponding unit should compute the maximum excitation. Kohonen’s learning algorithm is used to guarantee that this effect is achieved.

A Kohonen unit computes the Euclidian distance (the dot product metric can also be used) between an input x and its weight vector w . In the Kohonen one-dimensional network, the neighbourhood of radius 1 of a unit at the k -th position consists of the units at the positions $k-1$ and $k+1$. Units at both ends of the chain have asymmetrical neighbourhoods. Kohonen learning uses a neighbourhood function ϕ , whose value $\phi(i, k)$ represents the strength of the coupling between unit i and unit k during the training process. The complete description of Kohonen learning algorithm can be found in [2] and [6].

3 Experimental set-up

The data set consists of the Ventspils Nafta stock prices on daily up-dates for the period from December 1, 2000 till December 28, 2001. The total number of data points is 272 values. The description statistics of the data set is as follows: mean = 0.678; standard deviation = 0.061; maximum = 0.85, minimum = 0.58. Figure 3 shows the graphical representation of the data set.

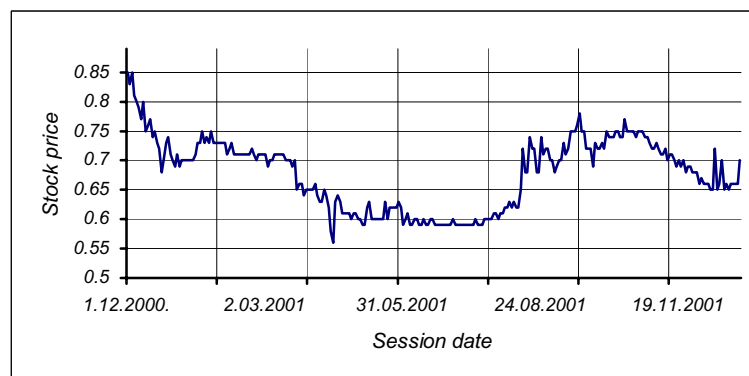


Figure 3. The data set: Ventspils Nafta stock prices on daily up-dates (December 1, 2000 – December 28, 2001).

Analysing the data set it has been found, that the most often changes of daily stock price are happening in the range from 1 to 4% (without taking to account the sign of the change), therefore we can use 7 classes or intervals: $(-\infty; -4.5\%]$, $(-4.5\%; -2.5\%]$, $(-2.5\%; -0.5\%]$, $(-0.5\%; 0.5\%]$, $(0.5\%; 2.5\%]$, $(2.5\%; 4.5\%]$, $(4.5\%; \infty)$. This number of classes is large enough to implement trading strategy on predicted results.

The input layer of the network is largely determined by application as for backpropagation, as well as for Kohonen network. In this particular case it is possible to use two similar approaches. The first one is standard windowing method, described in [4], while the second is almost identical. The difference is, that in the first case we use the set of last closing prices of the stock as input vector, but in the second one, the window is a set of the opening, the closing, the highest and the lowest price of the three or more trading days [8]. We have tried both of these approaches, and the better results are possible with the first one.

The architecture of the best performing backpropagation network is quite simple (20-10-7): it has 20 input neurons, 10 hidden neurons and 7 output neurons. The number of input units is crucial for the experiments. In this case the better results are possible with the look-ahead horizon of 4 weeks or 20 trading days. The number of classes determines the output layer size. An interesting estimate of the hidden layer size was proposed in [7]:

$$B = \frac{\phi(P - A) - C}{A + C + 1} \quad (3)$$

where B is the number of hidden layer neurons; P is the total number of input patterns; A is the input layer size; C is the output layer size and ϕ is the percentage of unique patterns that the net will be required to learn (accuracy).

The main purpose of Kohonen networks is to split data into number of clusters or classes that is defined before. In our case we already have classes, and each class is assigned to a pre-determined interval. Therefore we cannot use standard Kohonen learning algorithm without any corrections. The architecture of the Kohonen self-organizing map is shown in Figure 4.

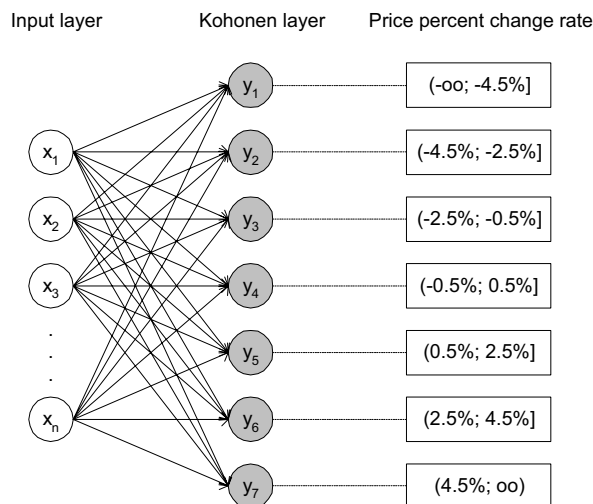


Figure 4. The Kohonen network architecture for price prediction.

The number of input units depends on the window size we are using (in this case there are 20 units). The number of clusters in the Kohonen layer is already determined (seven clusters of neurons). In the Figure 4 there is only scheme of the networks, and each unit in the Kohonen layer means a cluster of neurons (25 neurons with rectangular grid topology and initial radius 2). The training process can be considered as a supervised learning, because of the fact that each of the Kohonen clusters is trained individually. This is done by freezing all other neuron clusters while the inputs with the desired response identical to the given cluster are processed. We can say that this network is trained in seven stages accordingly to the number of classes. After the learning procedure is completed, all clusters are “de-frozen” (all neurons may be active), and the network works in a common manner.

4 Prediction results

There are many trading strategies, which allows interpreting prediction results in backpropagation and in Kohonen networking cases. The backpropagation network prediction results can be used, for example, in the most simple strategy – if the predicted price belongs to an increasing interval, then buy; if the price is going down, then sell. However, the better strategy is to use the committee of networks with different time horizon, and the strategy may be to buy, when the majority of networks classify the input pattern as a rising case; to sell, when networks predict price decrease; or to do nothing (untradable day), when the committee have no unambiguous answer.

An interpretation of the Kohonen network results may be as follows. If one Kohonen layer cluster is noticeably higher than all the rest, then we have a clear signal. If two or more neuron clusters are significantly active, then additional interpretation is required. A good approach to analysing multiple signals is to ignore signals when they are not occurring in contiguous neurones (clusters). If, for example, neurons 2 and 6 are active, then it is obvious that the network is not able to classify the pattern. A simple strategy would be to buy when neurons in the positive ranges are active, sell when neurons in the negative ranges are active and stand aside when neutral or conflicting activation occur. Table 1 shows the backpropagation and the Kohonen network prediction results on the training and on the test sets.

Table 1. Backpropagation and Kohonen network's performance on the training and the test sets.

	Backpropagation	Kohonen
Training Set:		
Correctly classified cases	84%	91%
Test Set:		
Correctly classified cases	65%	77%
Undetermined cases	-	11%

The correctly classified cases are given in percents from all test patterns. There are no undetermined cases for the backpropagation network, and for the Kohonen network the total percentage of such cases is 11%. It would be interesting to estimate the approximate returns of these strategies, but, unfortunately, the author does not have information about the costs per deal and other possible costs.

5 Conclusions

The results from this study indicate that a neural network is able to capture the relationships in stock price changes over time and make respectable predictions. The backpropagation network is like a "standard" in many problem domains, especially, in forecasting, but Kohonen self-organizing maps may be used more efficiently in this particular case. The main advantage of Kohonen network here is that we can easily determine the untradable days thus reducing the risk of losses.

Finally, future research will be connected with further study of possibilities to implement Kohonen networks in the area of forecasting. Another direction of experiments is counterpropagation networks, which are a hybrid model of backpropagation and Kohonen networks. This kind of networks can also be used not only for interval, but also for the single price value prediction.

References

- [1] Baestaens D. E., Van den Bergh W. M. (1995). Tracking the Amsterdam Stock Index Using Neural Networks. *Neural Networks in Capital Markets, Vol. 5*, P. 149-161.
- [2] Fausett L. (1994). Fundamentals of Neural Networks. Architectures, algorithms and applications, *Prentice Hall, Inc.*, P. 169-187.
- [3] Hean-Lee Poh, Jingtao Yao, Teo Jasic. (1998). Neural Networks for the Analysis and Forecasting of Advertising and Promotion Impact, *International Journal of Intelligent Systems in Accounting, Finance & Management, Vol.7*, P. 253-268.
- [4] Refenes A. N., Azema-Barac M., Chen L., Karoussos S. A. (1993). Currency Exchange Rate Prediction and Neural Network Design Strategies, *Springer-Verlag, London Limited*, P. 46-58.
- [5] Refenes A. N., Zaprani A., Francis G. (1994). Stock Performance Modelling Using Neural Networks. *Neural Networks, Vol. 7. No 2*. P. 357 – 388.
- [6] Rojas R. (1996). Neural networks. A systematic approach. *Springer, Berlin*, P. 389-410.
- [7] Zirilli J. S. (1997). Financial Predictions using Neural Networks. *International Thomson Computer Press, London*, 135 p.
- [8] Zurada J. M. (1992). Introduction to Artificial Neural Systems, *West Publishing Company, St. Paul*, 684 p.