ISSN 1691-5402
ISBN 978-9984-44-028-6
Environment. Technology. Resources
Proceedings of the 7th International Scientific and Practical Conference. Volume I1

RETAIL TURNOVER PREDICTION USING MODULAR ARTIFICIAL NEURAL NETWORKS

MAZUMTIRDZNIECĪBAS PIEPRASĪJUMA PROGNOZĒŠANA AR MODULĀRAJIEM NEIRONU TĪKLIEM

Aleksejs Zorins
Riga Technical University,
Faculty of Computer Science and Information Technology
1 Kalku Street, Riga LV 1658, Latvia; e-mail: Aleksejs.Zorins@ru.lv

Abstract. *The paper focuses on the retail turnover prediction with artificial neural networks. The artificial neural networks have the potential to learn complex, non-linear relationships within data. The main disadvantage is that neural networks are "black boxes", so the user cannot explain the obtained results and relationships between data. The modular neural networks allow obtaining more appropriate results by splitting the task into subtasks, thus giving the user more information in the output. In many cases an additional advantage of modular neural network is more precise prediction results, which will be shown in the experimental part of this paper.*

Keywords: *modular neural networks, artificial neural networks, time series prediction.*

Introduction

The traditional statistical methods of forecasting (linear regressions, ARIMA, GARCH etc.) are not effective in many real-world applications [1, 2, 3]. There are two main disadvantages of these methods: it is often impossible to describe real world with only few parameters or the calculation of the model requires much processing time [1, 2]. The artificial neural networks, which are based on the model of the human brain, have the potential to learn complex, non-linear relationships within data. The main disadvantage is that neural networks are "black boxes", so the user cannot explain the obtained results and relationships between data.

The modular neural networks allow obtaining more appropriate results by splitting the task into subtasks, thus giving the user more information in the output. In many cases an additional advantage of modular neural network is more precise prediction results, which will be shown in the experimental part of this paper. The next paragraph gives the theoretical background of modular neural networks, while the last paragraph will show the problem domain, prediction results of the presented method and comparison with "single" multilayer perceptron and Kohonen self-organizing map.

Materials and methods

Before we proceed to the description of the modular neural networks, let me give short introduction of two main artificial neural network architectures and learning algorithms. The multilayer perceptron (MLP) is a collection of single processing units connected one with other by the mean of weighted links and processing information from input till output as shown in Fig.1 (the number of neurons is determined by the application).

These units are combined into layers. The first layer, also called input layer, transfers the input data (input patterns) to the rest of the network without processing. The information is processed by the hidden layers and output layer. Each processing element (neuron) has many inputs and only one output. The signal is transferred by the weighted connection link. Each weight determines the importance of the link.

The first operation of an MLP is the forward propagation. An input vector is propagated from the input to the hidden layer of neurons. The output of each neuron is multiplied by the weights of the outgoing connections and then propagated further to the inputs of the next layer neurons. The neuron has the activation (squashing) function, which produces the output of a neuron. These calculations are repeated for all neurons of network until all outputs of output neurons are obtained.

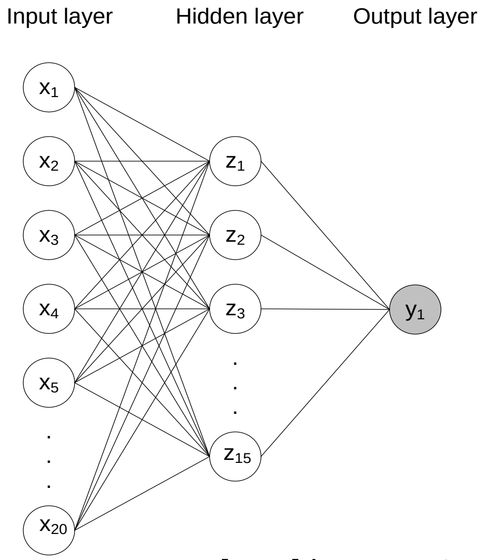

Fig.1. **MLP network architecture 20-15-1**

The most widely used training algorithm for MLP is error back propagation. In the beginning of the learning stage all weights of the network are initialized to small random values. The algorithm uses the input patterns, which consist of pairs of input vector and desired output (often called teacher response). These pairs are used to adjust the weights in the network to minimize the Mean Squared Error (MSE) which measures the difference between the real and desired output values obtained from all input patterns. After MSE is calculated, the backpropagation step computes the corrections to be applied to the weights [4].

Another type of artificial neural network is Kohonen self-organizing map, which models the human brain cortex functions [4]. The Kohonen self-organizing maps may be trained in unsupervised (in most cases) and in supervised manner. This type of network uses grid of neurons or a topological structure among the cluster units. There are *m* cluster units, arranged in a one- or two-dimensional array: the input signals are *n*-dimensional. Figure 2 shows architecture of a simple self-organizing network, which consists of input and Kohonen or clustering layer. The shadowed units are processing units. 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).

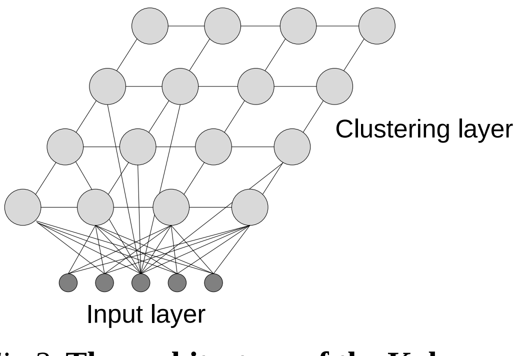

Fig.2. **The architecture of the Kohonen self-organizing map**

When a self-organizing network is used, an input vector is presented at each step. These vectors create 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 [5].

Consider the problem of charting an n-dimensional space using a one-dimensional chain of Kohonen units [4]. The units are all arranged in sequence and are numbered from 1 to m (see Fig. 3).

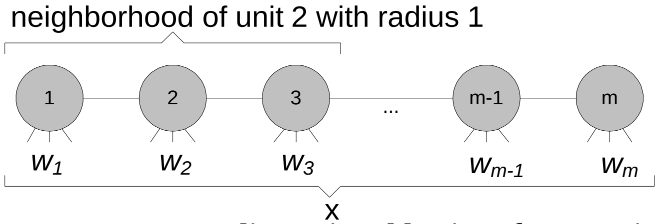

Fig.3. **A one-dimensional lattice of computing units**

The *n*-dimensional weight vectors w_1, w_2, …, 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 [4] and [7].

Theoretically, a large neural network may be trained to solve the task of any complex function approximation. The training time, however, for such function may be reasonable long, and the results of such a network cannot be totally understood and explained. Another advantage of modular approach is an improvement of generalization (ability to perform well on the test data) due to decomposition of complex function into simpler ones. The modular neural networks have been proposed to solve this problem. The main idea is a natural decomposition of a function of large complexity into simple functions and realization of each function by a separate neural network [5].

The architecture of modular neural networks is largely determined by the application. The commonly used architectures are given below [4].

Input decomposition: multiple input system may be split into subsystem in such a way that the inputs of each subsystem form a set of the inputs of the entire system (see Fig.4).

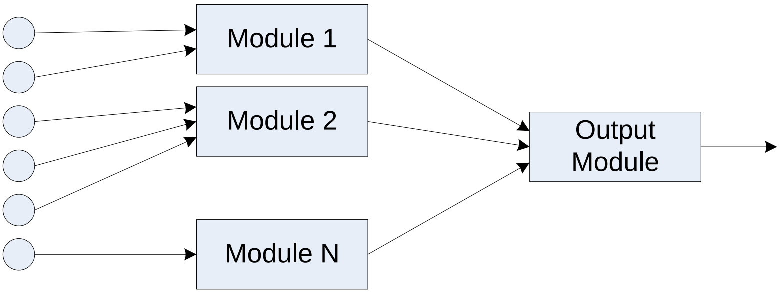

Fig.4. **Decomposition of the inputs of different modules**

Output decomposition: there is a possibility to divide a task into subtasks, each of which can be solved individually. An example of such architecture is given in Fig.5.

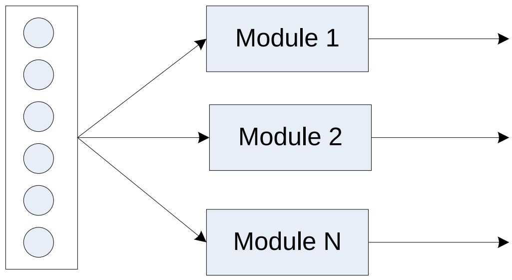

Fig.5. **Decomposition of the outputs of different modules**

Hierarchical decomposition: a multi-input multi-output system often can be decomposed into simple module hierarchical structure. In this case the outputs of the lower levels act as the inputs of the higher levels. Fug.6 and Fig.7 shows simple pipelining and more complicated hierarchical structure examples.

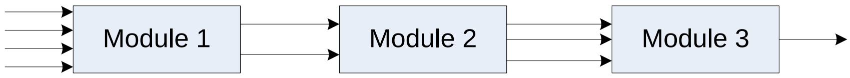

Fig.6. **A pipelining architecture**

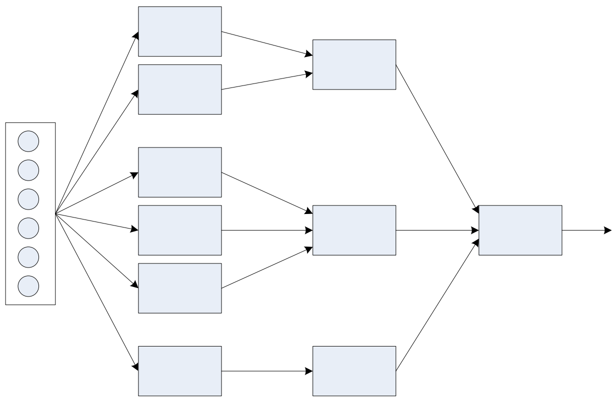

Fig.7. **Complex hierarchical decomposition**

The next part of the paper presents an implementation of modular neural network approach for retail turnover prediction task.

Results

The data set consists of three UK retail turnover time series (see Fig. 8, 9 and 10).

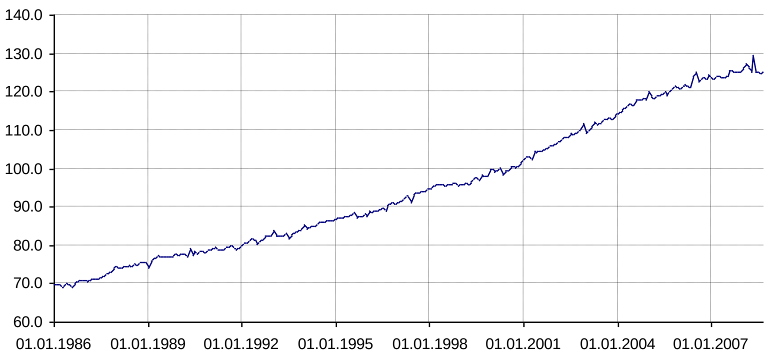

Fig. 8. **Food stores turnover in the UK (Jan. 1986 – Sept. 2008)**

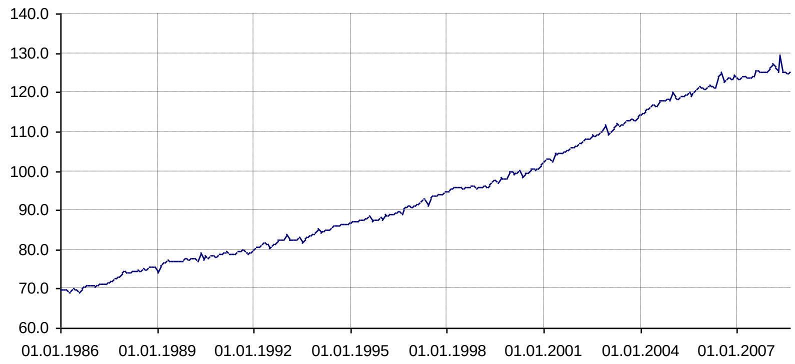

Fig. 9. **Non-specialized stores turnover in the UK (Jan. 1986 – Sept. 2008)**

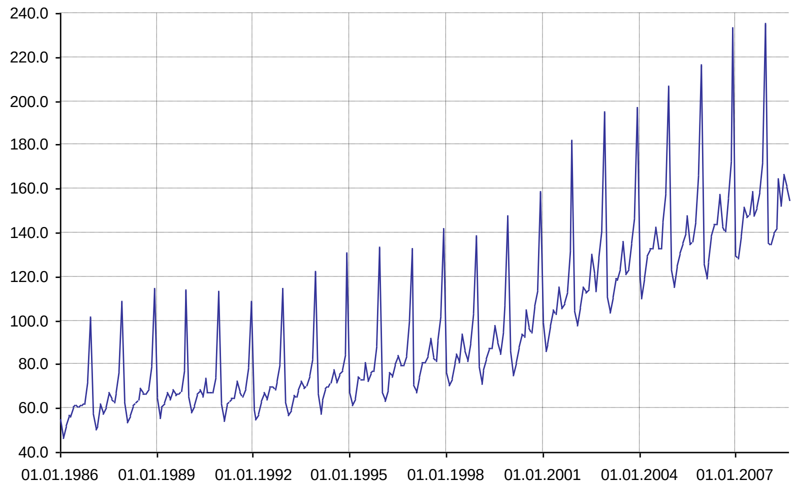

Fig. 10. **Textiles turnover in the UK (Jan. 1986 – Sept. 2008)**

Each data set size is 272 data points, which is big enough for experiments. The first two time series has a rising trend during almost all period, but the last time series has highly expressed seasonal component.

The main idea behind the decomposition of these data is the use of interval value prediction. Most often a person needs precise or exact values as a prediction result, however, in many cases such a prediction has quite big error and can be replaced with the interval value prediction, when we obtain value (retail turnover) change intervals. Varying the length of such intervals, one may obtain required forecasting precision. In this case we have classification problem.

The interval value prediction is based on the partitioning of time series into number of intervals, which may represent factor changes in absolute values or percentage. The simplest case allows using standard windowing method for input value preparation; however, better results are possible with time series transformation methods. See [8 and 9] for complete description.

Fig. 11 shows an example of modular neural network used in experiments.

Modular layer consists of seven individual networks ($y_1, \ldots, y_7$), each of which is responsible for training on different class data (each class denotes value change interval, as shown in Fig.11). As a result we have modular network with seven modules, which are trained individually, but in the test phase, when we input test data (unknown for the network), all modules work together.

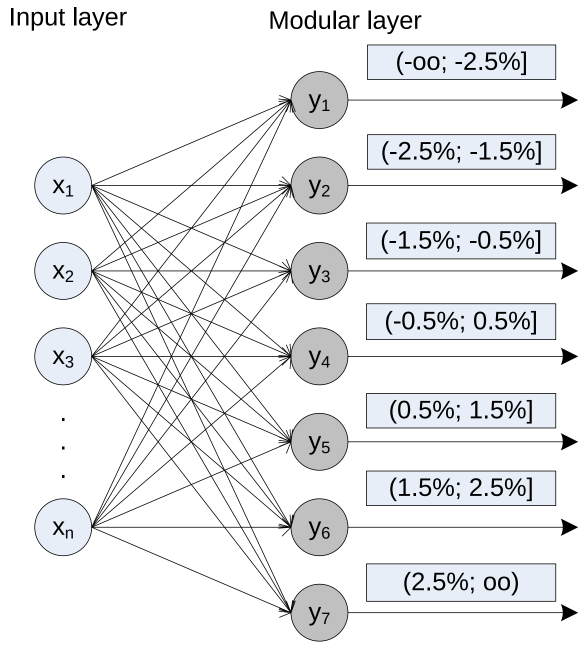

Fig. 11. **Modular neural network for interval value prediction**

In the input layer ($x_1, \ldots, x_n$) we have description of the intervals (segments) of time series (see Table 1).

Table 1.

Description of neural network inputs

Identifier label	***Description***
AC	Linear regression coefficient *a* for each approximated segment
BC	Linear regression coefficient *b* for each approximated segment
LS	Length of a segment (in data points)
SS	Starting point of a segment
ES	End point of a segment
CL	Class identifier

To achieve this effect we split our data set into seven subsets for each value change interval (class). For instance, all segments belonging to (-0.5%; 0.5%] turnover change interval, will be input for module y_4 and so on. In the test phase we input the unknown data to each module and compare the outputs. The strongest signal will be the winner, allowing identifying the class, which denotes the future behavior of the time series.

Table 2 shows the experimental results for different learning algorithms. The standard Kohonen network architecture cannot be applied for supervised learning, therefore it is not included.

Table 2.

Classification results for different training algorithms

Learning algorithm	***Correctly classified cases (in percents)***	
	Training set	***Test set***
Simple backpropagation MLP	89	72
Modular backpropagation MLP	95	91
Modular Kohonen self-organizing map	96	93

Table 2 shows average classification performance on all three data sets with 20 experiments on each time series.

Discussion

As we can see from the Table 2, the best results are obtained by modular Kohonen self-organizing map (96% of correctly classified cases in the training phase and 93% of correctly classified cases in the test phase). The average performance in the training set is quite good also for simple MLP; however, the poor results on the test set show weak generalization power of such approach. On the other side, both modular MLP and Kohonen self-organizing maps are almost equally good on the training and the test sets (the difference is only 1-2% of correctly classified cases).

Acknowledgment

The author would like to acknowledge Professor Arkady Borisov for his scientific guidance and support.

Summary

Modular neural networks allow decomposing complex problems into subtasks, which can be solved individually. As a result one may obtain better precision, less learning time and more appropriate explanation of the results. The experiments on the retail turnover data sets show good prediction performance of modular approach in comparison with single neural networks.
Further research results will show another modular decomposition implementation approaches for the prediction tasks.

Bibliography

1. Baestaens D.E., Van den Bergh W.M. Tracking the Amsterdam Stock Index Using Neural Networks. Neural Networks in Capital Markets, Vol. 5, 1995. P. 149-161.
2. Poh H.L., Yao J., Jasics T. Neural Networks for the Analysis and Forecasting of Advertising and Promotion Impact, International Journal of Intelligent Systems in Accounting, Finance & Management, Vol.7, 1998. P. 253-268.
3. Refenes A.N., Zapranis A., Francis G. Stock Performance Modelling Using Neural Networks, Neural Networks, Vol. 7. No 2. 1994. P. 357-388.
4. Rojas R. Neural networks. A systematic approach, Springer, Berlin, 1996.
5. Konar A. Computational intelligence: principles, techniques and applications. Springer-Verlag, London, 2005.
6. Palit A.K. Computational intelligence in time series forecasting: theory and engineering applications. – London, Springer-Verlag, 2005.
7. Fausett L. Fundamentals of Neural Networks. Architectures, algorithms and applications, Prentice Hall, 1994.
8. Zorin A. Data preprocessing methods for interval based neural network prediction, Proceedings of the International Conference “Environment. Technology. Resources”, Rezekne, Latvia, June 20-22, 2007. P. 211-219.
9. Zorin A. Forecasting with neural networks: exact and interval value prediction, Proceedings of the International Conference “Simulation and Optimisation in Business and Industry”, Tallinn, Estonia, May 17-20, 2006. P. 249-254.