ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS
SBORNÍK MENDELOVY ZEMĚDĚLSKÉ A LESNICKÉ UNIVERZITY V BRNĚ

Volume LVII 31 Number 6, 2009

ADVANCED APPROACH TO NUMERICAL FORECASTING USING ARTIFICIAL NEURAL NETWORKS

M. Štencl, J. Šťastný

Received: July 9, 2009

Abstract

ŠTENCL, M., ŠŤASTNÝ, J.: *Advanced approach to numerical forecasting using artificial neural networks.* Acta univ. agric. et silvic. Mendel. Brun., 2009, LVII, No. 6, pp. 297–304

Current global market is driven by many factors, such as the information age, the time and amount of information distributed by many data channels it is practically impossible analyze all kinds of incoming information flows and transform them to data with classical methods. New requirements could be met by using other methods. Once trained on patterns artificial neural networks can be used for forecasting and they are able to work with extremely big data sets in reasonable time. The patterns used for learning process are samples of past data. This paper uses Radial Basis Functions neural network in comparison with Multi Layer Perceptron network with Back-propagation learning algorithm on prediction task. The task works with simplified numerical time series and includes forty observations with prediction for next five observations. The main topic of the article is the identification of the main differences between used neural networks architectures together with numerical forecasting. Detected differences then verify on practical comparative example.

Artificial Neural Networks, Radial basis function, Numerical Forecasting, Multi Layer Perceptron Network

The knowledge of the future creates the advantage in all kind of business. The methods traditionally used for numerical forecasting are based on precise analysis of past values. The prognosis is then built as approximation of future values using functions estimated from dependencies founded by past values analysis.

The statistical time series model is used as a traditional method for economical forecasting. (Sarle, 1994) Constructions of time series models is still very powerful method but is very thought for initial conditions and for forecast construction the time series model has to be made from input data. Especially the input conditions and time series construction necessity increase computational complexity of the task. If condition rules for inputs are joined with amount of the input data, the global complexity is even bigger and the time needed for building the time series model is often not effective.

As an alternative to time series forecasting methods, artificial intelligence based methods can be used. Generally are the methods divided into techniques based on machine learning (Mitchell, 1997) (e.g. Support Vector Machines, Decision Trees) and techniques inspired by biological processes (Mařík, Štěpánková, Lažanský; 2003) (e.g. Artificial Neural Networks, Genetic Algorithms). Most of the learning artificial intelligence methods are based on analyzing sets of patterns. Therefore we call the tasks as exploration of state space. Process of exploration is stopped after previously set number of epochs or when defined error on output is reached. The number of all neurons used in artificial neural network defines the dimension of the state's space (Mařík, Štěpánková, Lažanský; 2003).

Artificial neural networks don't need to know algorithm to reach forecast as in the statistics methods that makes the main difference (Novák, 1998), (Sarle, 1994), (Šnorek, Jiřina; 1998). The forecasting of future values with artificial neural networks is based on learned past pattern sets for defined length. The principle of artificial neural network

work is based on learning values from past periods and then approximating the future values. The accuracy of the prediction is influenced by several attributes such as topology of selected artificial neural network, learning rule, types of activation function, number of inputs, length and also structure of input time series. Globally there are two main categories of artificial neural network models – feed-forwarded networks and recurrent networks. A feed-forward network represents a function of its current input; thus, it has no internal state other than the weights themselves. A recurrent network feeds its outputs back into its own inputs. (Russel, Norvig; 2003) Russel and Norvig (2003) formulate learning as an optimization search in weight space. The definition means the reset of the weights of inputs on each input node. Artificial neural networks use two types of learning – supervised and unsupervised. When supervised learning is used a training set for output validation must be supplied. Training sets are used as inputs for the network and the computed outputs are compared with sample results. Weights of all neurons are adjusted backwards according to the output error. A specific algorithm of resetting the weights is defined by the learning algorithm used. Both genetic algorithm and back-propagation algorithm were tested as the learning algorithms. Back-propagation seems to yield better results in prediction tasks (Šťastný, Škorpil; 2007).

One of the most widely used methods in time series forecasting is the classical multi-layer perceptron artificial neural network (MLP) with the back-propagation learning algorithm (Šťastný, Škorpil; 2005). Often the MLP model is also combined with statistical models in hybrid systems (Tseng, Yu, Tzeng; 2002). The MLP model is one of the basic models but often brings very good results. Regarding to the article (Šťastný, Štencl; 2008) there has been wide interest in making the comparison study of published MLP forecast with other models such as radial basis function (RBF-NN) or competitive networks.

METHODS AND RESOURCES

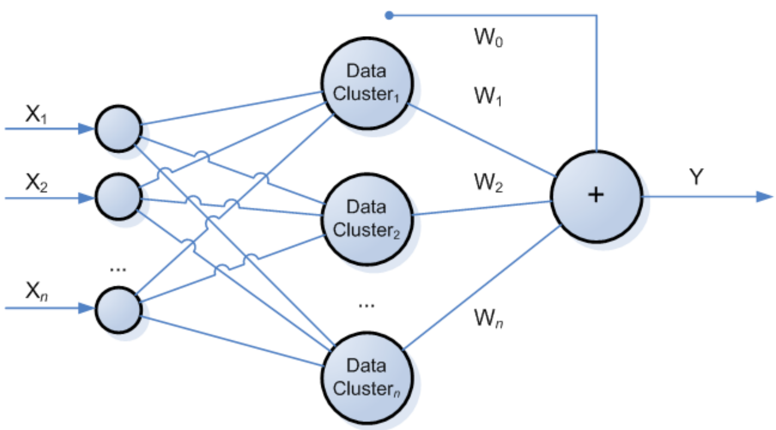
The model used for experiments in this article is the Radial Basis Function neural network. Since this article was inspired by previously published (Šťastný, Štencl; 2008) works with the same data model.

Simplified data model build for the experiments consist of one time series with total length 40 values. The values have been generated randomly and have an absolute value 3.

Radial Basis Function neural networks (RBF-NN) belong to the feed-forwarded models of neural networks. RBF-NN consists of three layers of nodes (Fig. 1). The first is the input layer that transports the input vector to each of the nodes in the hidden (second) layer. The third layer consists of on node. It sums the outputs of the hidden layer of nodes to yield the decision value (Wedding, Cios; 1996).

The RBF-NN has same topology as three layers MLP networks. The main difference is in the hidden layer. As defined in (Wedding, Cios; 1996), (Šíma, Neruda; 1996) the hidden layer of nodes each node represents a data cluster, which is centred at a particular point and has a given radius and could be named as local unit. As in (Šíma, Neruda; 1996) the local units have the relevant output localized to the point in close neighbourhood defined by its parameters. When an input vector goes on each node of hidden layer simultaneously, each node then calculates the distance from input vector to its own centre (Wedding, Cios; 1996). The MLP nodes, on the other hand, divide the input space on subspaces in that's exist big difference on output.

Radial Basis Functions are used for approximation and interpolating in numerical mathematics. The approximation process is based on some function. Usually is used the linear combination of base functions, here the radial functions. The basis function realizes the transformation of the distance value, calculated from the input vector to its own centre, to the output value of the node. The output value is then multiplied by weighting value or a constant.



1: *Simple RBF-NN Architecture*

Algorithm APC-III:

C: the number of neurons
c_j: the middle of j-th neuron
n_j: the number of samples in j-th neuron
d_{ij}: the distance between x_i and j-th neuron

```
{
    C=1; c1 ← x1; n1 = 1;
        for(i = 2; i =< P; i++)   // for every pattern from training set
{
        for(i = 1; i =< C; i++)   // for every neuron
{
    calculatedij;
    if(dij=< R0)                  // insert xi into j-th neuron
    {
        cj = (cjnj + xi)/(nj + 1);

        nj = nj + 1;
        break;
};
};
if(xi is not in any neuron)       // create new neuron
{
    C = C + 1;
    cc ← xi;
    nc= 1;
};
};
};
```

2: *Algorithm APC-III*

Problems at creation of RBF-NN consist on determination of the number of neurons in hidden layer, on determination of the middles of these neurones and on determination of the neurones width. Powerful method for determination of the number and quality of neurons of hidden layer is the algorithm APC-III (Fig. 2). This single-pass associating algorithm unlike other uses constant radial (Ripley, 1996).

The learning process consists on precept of given network to answer correctly to engage training set. As the hidden layer was in this network represented so-called areas and the middles of the areas are fast given to it, the learning process oversimplifies only to setting of scales and thresholds of the output layer. Gradient method and Last Mean Square (LMS) method were tested for learning of neuron network.

The gradient method uses relations derived for outgoing layer for algorithm Back-propagation (BP). On difference from the method BP this method optimizes only scales and thresholds of outgoing layer.

Method Least Mean Square (LMS) tries to find optimal scaled vector for general middle quadratic error of the network. Normal division gives this scaled vector:

$$w = (H^T H)^{-1} H^T y, \tag{1}$$

where w is scale vector, H is suggestion matrix and y is vector of outgoing values. This method in contrast to of others ones uses like transient function of outgoing neurons layer in lieu of sigmoid the linear function (Ripley, 1996), (Sarle, 1994).

The input-output pair forms the training set. RBF network learning is divided into three stages (Šíma, Neruda; 1996). The stages works with training set on input defined as

$$\{(x^{(t)}, d^{(t)})\ t = 1, \ldots, k\}, \tag{2}$$

where
$x^{(t)} \in R^n$ are the inputs,
$d^{(t)} \in R^n$ are the wanted results.

In the learning stages network
1. estimates centres of c_j with $x^{(t)}$,
2. estimates widths b_j,
3. determines the weights w_{sj} of input neurones $(x^{(t)}, y^{(t)})$.

In the first stage, centres c_j are determined for each RBF unit. The centres c_j are representing by the weights between the input and the hidden layer. For example the algorithms for cluster analysis are used. To speed up this stage, non-adaptive methods can also be used such as uniform or random distribution of RBF neuron centres over the input space.

Second stage setups other values of RBF neurons. The setup values of RBF units (b_j) determine the wideness of the area around estimated centres of c_j.

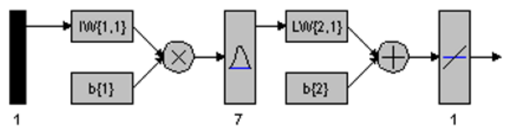
The objective of the third stage of learning is to determine the weights of input neurons, for example the least square method or gradient algorithms can be used.

In global view, the RBF-NN learning includes in the first stage unsupervised learning. In the second stage RBF units are setup. Typically used function for this setup is the Gaussian Radial Basis Function (Šíma, Neruda; 1996) defined as on (3).

$$\varphi(x) = e^{-\left(\frac{\|x-c\|}{b}\right)^2}. \qquad (3)$$

RBF unit determines the important output values in radial zone with centre in ***c***. The ***b*** represents width of φ and determines the size of the radial zone. The setup parameters of RBF units determine the wideness of the *controlled* area and affect the generalization capability of the network. If the parameters are smaller means lower generalization capability and on the other hand for wider area the units lost their local mean.

In the last stage, the supervised learning is used. The last stage setups the weights w_{sj}. The setup is made by mineralization process of typical error function. (Šíma, Neruda; 1996)



3: *Matlab RBF-NN topology*

After the learning process is the RBF-NN ready to approximate training sets and also provide good results for answer outside the training set. Different techniques for regularization are discussed for long time. As example Bishop (1991) works with same RBF units as training patterns. This technique brings uniform resolution and wideness of Gaussian function, if the input data have same time of generation.

RESULTS

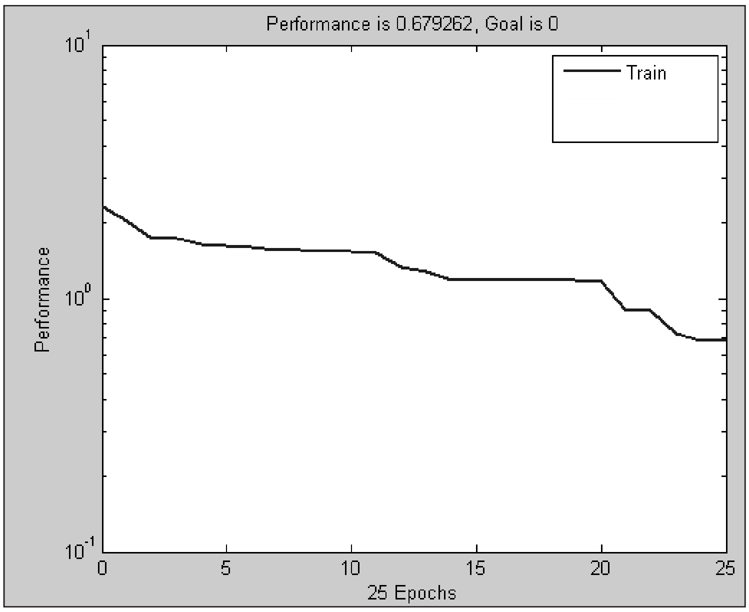
The experiment consists of comparison tasks of the RBF-NN implementation in Matlab R2007a and MLP-NN used in (Šťastný, Štencl; 2008). The task is based on previously discussed model of time series. The forecast is made for five future periods. Figure 3 describes the standard topology of Matlab RBF-NN implantation algorithm.

The learning of the network was much faster then in the MLP network with Back-propagation algorithm. But obtained values are not as exact as in the case of MLP network. Precisely in the case, Matlab implementation is not allowing better configuration of the network. Figure 4 shows the performance progress for the Matlab implementation of RBF-NN. The performance value for the MLP-NN was 0.23124. The performance value for the Matlab was 0.6973.

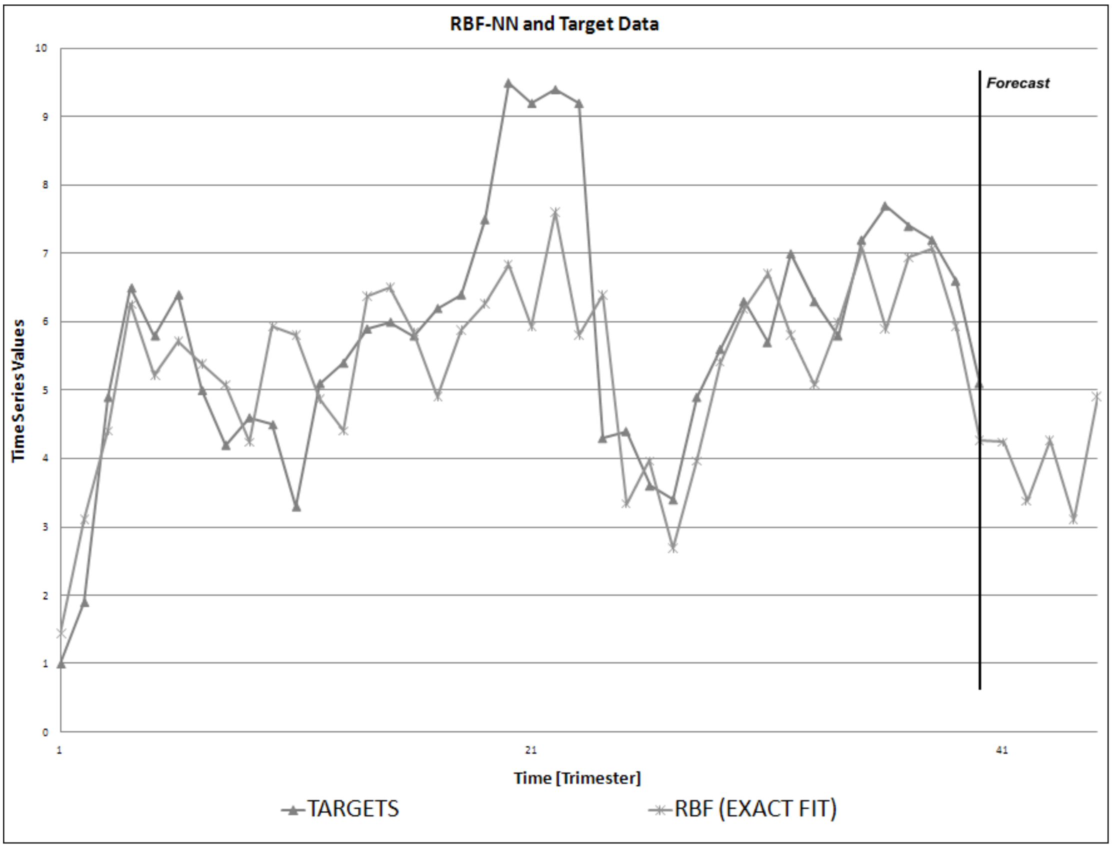
On the Fig. 6 is shown the approximation of the input time series by the MLP network with Back-propagation learning algorithm and the approximation made by Matlab implementation of RBF-NN. The network used is exact fit. Globally the RBF-NN is used for approximation.

DISCUSSION

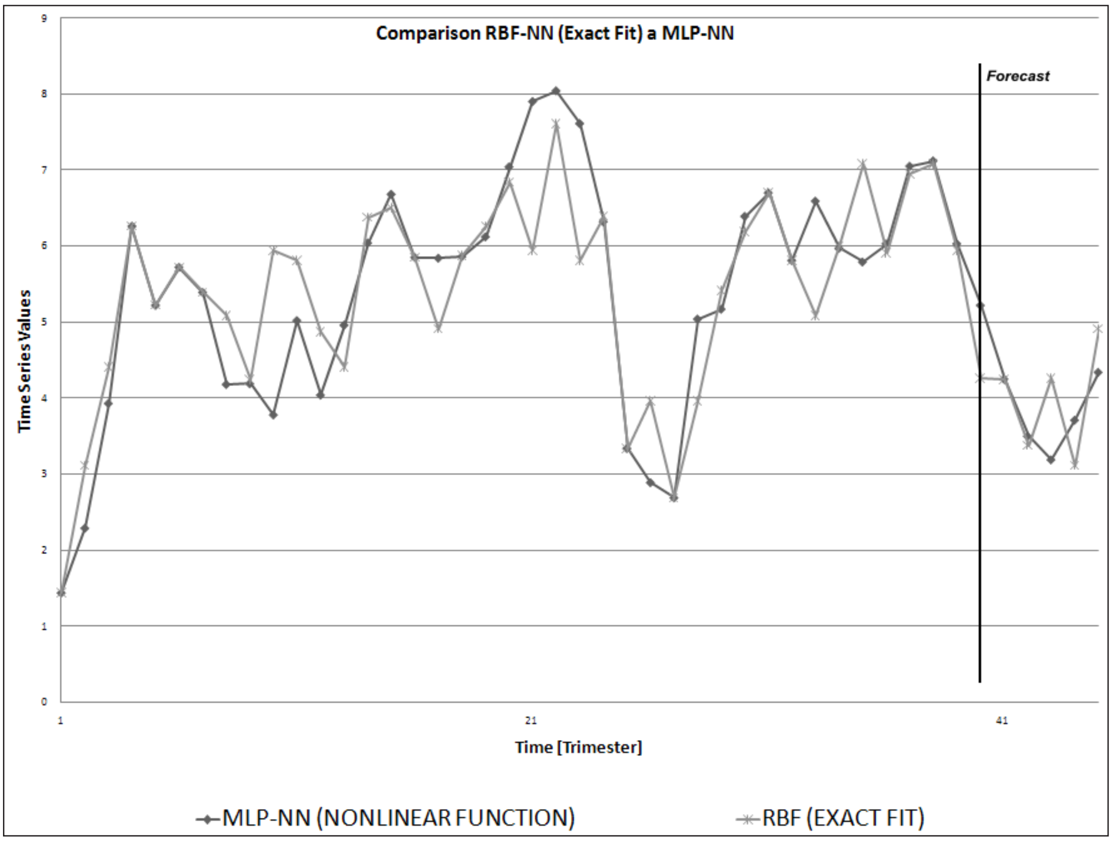
The differences between obtained results could by caused by few facts. First of all is that the RBF-NN have different activations function (ex. Fig. 5, 6). An MLP network uses one from four activation functions – Linear, Threshold, Standard sigmoid, Hyperbolical tangents (Novák, 1998), (Šíma, Neruda; 1996). An RBF-NN uses the Gaussian or Multi-quad-



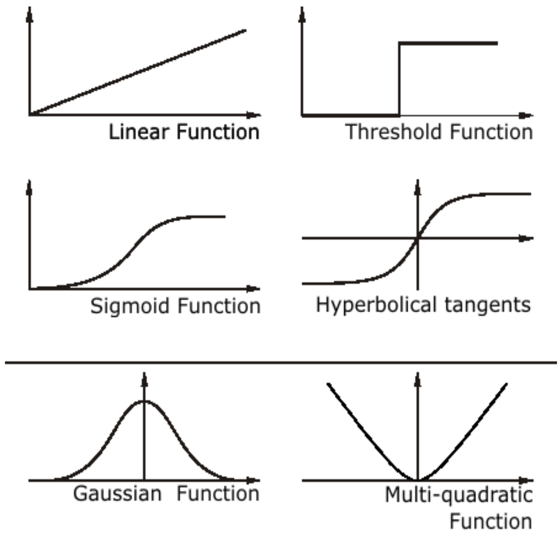
4: *Matlab learning output*



5: *Comparison of trained RBF-NN (Exact Fit) on Target Data*



6: *Comparison of forecast made by RBF-NN (Exact Fit) and MLP-NN* ratic functions. Regarding to activation functions, both approximates input values. The difference is made with the extrapolation of input data, where better results for this precise data model bring MLP-NN. The progress of the functions allows better adaptability for the sigmoid functions used in MLP network. Next factor effecting obtained results is also implemented algorithm of RBF-NN in Matlab. The experiments use the standard Matlab RBF-NN. The standard implementation was not optimized because of future work. The next steps are planed to the same comparison model (MLP-NN with Back-propagation learning algorithm and RBF-NN) with bigger model of time series where especially could be notified the strongest merit of RBF-NN – fast learning. The RBF-NN was learned in noticeably shorter time than the MLP-NN learning process on goal value of error function. The error function represents the precision of learned neural network. The next research and testing includes experiments with implementation of RBF-NN with the learning algorithm APC-III presented previously in this article. The conclusion of the article includes the fact that the forecast for the simplified time series model used in this article was better by using the MLP-NN. Notifying the RBF-NN could have better results for more complexes model or at least same in shorter time regarding to the speed of learning process.



7: *NN Activation functions*

SUMMARY

The paper presents Radial Basis Functions neural networks (RBF-NN) in comparison with Multi Layer Perceptron neural network (MLP-NN) on previously published results of time series based prediction. The article specifies the main differences between RBF-NN and MLP-NN in both theoretical and practical points of view. Asset of the paper is present in definitions of the algorithms describing RBF-NN with specified learning process and presentation of the learning procedure of the network. The need of building own RBF-NN for time series forecasting is the results of the practical experiment presented in the article. On the other hand the experiment confirmed the powerful learning capability of the RBF-NN.

RBF-NN are very powerful methods for approximation task. For forecasting it is necessary to develop own implementation focusing on time series model used. In practical experiment in this article had better results MLP-NN. RBF-NN reached similar error function in shorter time, but with worse forecast.

SOUHRN

Pokročilé metody numerických predikcí s využitím neuronových sítí

Článek představuje neuronové sítě typu RBF (sítě radiálních bazických funkcí) v porovnání s vícevrstvými neuronovými sítěmi (MLP) s učícím algoritmem zpětného šíření (Back-propagation). Srovnání je podloženo praktickou úlohou spojenou s odhadem budoucího vývoje zjednodušeného modelu časové řady publikovaného dříve. Výchozí je srovnání RBF a MLP sítí v obou rovinách – teoretické i praktické. Přínosem článku je také přesné vymezení odlišných přístupů k učení obou sítí. Praktická úloha ukázala nutnost vlastní implementace RBF sítě pro její využití na predikce různých modelů časových řad. Jako výhodu plynoucí z provedeného experimentu je nutné zmínit rychlost učení sítě typu RBF.

Sítě typu RBF ukázaly svou vysokou účinnost a výkonnost pro aproximační úlohy. Pro jejich využití na odhad budoucího vývoje modelů časových řad je však nutné provést vlastní implementaci. Vlastní experiment prokázal lepší výsledky u MLP sítě. Současně sítě typu RBF však dosáhly podobné chyby ve znatelně kratší době, čímž potvrdily svou účinnost, avšak nedosáhly stejně kvalitní predikce.

neuronové sítě, sítě typu RBF, číselné předpovědi, vícevrstvá síť

MSM 6215648904/03 Development of relationships in the business sphere as connected with changes in the life style of purchasing behaviour of the Czech population and in the business environment in the course of processes of integration and globalization (Research design of Mendel University of Agriculture and Forestry in Brno)
MSM 0021630529 Intelligent Systems in Automation (Research design of Brno University of Technology)
No 102/07/1503 Advanced Optimisation of Communications Systems Design by Means of Neural Networks. The Grant Agency of the Czech Republic (GACR)

REFERENCES

BISHOP, C. M., 1991: *Improving the generalization properties of radial basis function neural networks*. Neural Computation 3, pp. 579–588.

BORSE, A. G., 2007: Introduction of the Radial Basis Function (RBF) networks. *ANU College of Engineering and Computer Science*. [cit. 2009-06-24]. Cited from http://axiom.anu.edu.au/~daa/courses/GSAC6017/rbf.pdf

MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kol., 2003: *Umělá inteligence*. 1. vydání. Sv. 1. Praha: Academia. ISBN 80-200-0496-3.

MITCHELL, T. M., 1997: Machine Learning. 1. vydání. McGraw-Hill. ISBN 0-07-042807-7.

NOVÁK, M. a kol., 1998: *Umělé neuronové sítě. Teorie a aplikace.* C. H. Beck, Praha.

RIPLEY, B. D., 1996: *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge (UK).

RUSSEL, S. J., NORVIG, P., 2003: *Artificial Intelligence – A Modern Approach*. 2nd Edition. Person Education, Inc., New Jersey. pp. 736–741, ISBN 0-13-080302-2.

SARLE, W. S., 1994: *Neural Networks and Statistical Models*. Proceedings of the Nineteenth Annual SAS Users Group International Conference, Cary, NC: SAS Institute, pp 1538–1550.

ŠÍMA, J., NERUDA, R., 1996: *Teoretické otázky neuronových sítí*. 1. vydání. Praha: MATFYZPRESS, MFF UK. ISBN 80-85863-18-9.

ŠNOREK, M., JIŘINA, M., 1998: *Neuronové sítě a počítače*. ČVUT, Praha.

ŠŤASTNÝ, J., ŠKORPIL, V., 2005: *Neural Networks Learning Methods Comparison*. International Journal WSEAS Transactions on Circuits and Systems, Issue 4, Volume 4, pp. 325–330, ISSN 1109-2734.

ŠŤASTNÝ, J., ŠKORPIL, V., 2007: *Genetic Algorithm and Neural Network*. WSEAS Applied Informatics & Communications, pp. 347–351, ISBN 978-960-8457-96-6, ISSN 1790-5117.

ŠŤASTNÝ, J., ŠTENCL, M., 2008: *Nové metody predikce numerických dat*. In: Aktuálne manažérske a marketingové trendy v teórii a praxi. 1. vyd. Žilina: EDIS. s. 28–33. ISBN 978-80-8070-966-2.

TSENG, F. M., YU, H. C., TZENG, G. H., 2002: *Combining neural network model with seasonal time series ARIMA model*. Technological Forecasting and Social Change, 69, 71–87.

WEDDING D. K., CIOS K. J., 1996: *Time series forecasting by combining RBF networks, certainty factors, and the Box-Jenkins model*. Neurocomputing, vol. 10. pp. 149–168. DOI:10.1016/0925-2312(95)00021-6.

Address

Ing. Michael Štencl, doc. RNDr. Ing. Jiří Šťastný, CSc., Ústav Informatiky, Mendelova zemědělská a lesnická univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, e-mail: michael.stencl@mendelu.cz, stastny@fme.vutbr.cz.