Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals of DAAAM International

ARTIFICIAL NEURAL NETWORKS WITH RADIAL BASIS FUNCTION IN PREDICTION BENCHMARK

SAMEK, D[avid]

Abstract: *The paper continues the research presented in the last DAAAM symposium (Samek, 2009), where the four types of artificial neural networks were tested in CATS prediction benchmark and the results were compared and discussed. This contribution is focused on artificial neural networks with radial basis function in the hidden layer. The special attention is paid not only to the prediction accuracy, but also to the computational demands of predictor.*
Key words: *artificial, neural, network, prediction, benchmark*

1. INTRODUCTION

The prediction task is one of the oldest problems that mankind has been trying to resolve. Perhaps the best known is weather forecasting; nevertheless there are many and many other applications where the prediction and predictors are necessary. Heat producers need to know the approximate consumption of the heat for nearest week, energy producers and distributors work with expected consumption rates, money brokers cannot live without sophisticated software that predicts exchange rates, etc. There are many methods how to predict, one of them is utilization of artificial neural networks (ANNs). However, because there are many different predictive techniques, it is necessary to compare and validate prediction results of various methods. One established approach is the CATS (Competition on Artificial Time Series) benchmark (Lendasse et al., 2004; Samek, 2009).

This paper continues the research presented in (Samek, 2009) where the four types of ANNs were tested in the CATS benchmark:

- Multilayered Feed-Forward Neural Network with hyperbolic tangent sigmoid transfer function in the hidden layer and linear transfer function in the output layer (*mffnntp*)
- Multilayered Feed-Forward Neural Network with hyperbolic tangent sigmoid transfer function in the both layers (*mffnntt*)
- Adaptive Linear Network (*adaline*)
- Elman Neural Network (*enn*)

Despite, the tested networks showed interesting predictive performance, the results were not optimal yet. Thus, the research aimed to fast (in terms of low computational demands) and precise predictors had to continue. This contribution focuses on the prediction using artificial neural networks with radial basis transfer function.

2. ARTIFICIAL NEURAL NETWORKS WITH RADIAL BASIS FUNCTION

Artificial neural networks with radial basis function (RBF) have typically two layers. As can be seen from figure 1, the hidden layer consists of radial basis transfer function, while the output layer uses linear transfer function (Suykens et al., 1996).

The radial basis function in the hidden layer is a function that normalizes radial distance between input vector **u** and the vectors formed from the rows of weight matrix $\mathbf{W}_1$. The bias vector **b** decides the range of influence of the particular RBF unit around its centre defined in the matrix $\mathbf{W}_1$. General mathematical description of RBF networks is as follows (Nelles, 2001):

(1)

, (2)

where **y** is the output vector of the network, $\mathbf{x}_1$ stands for the output vector of the hidden layer and S_i is transfer function of the i-th layer.

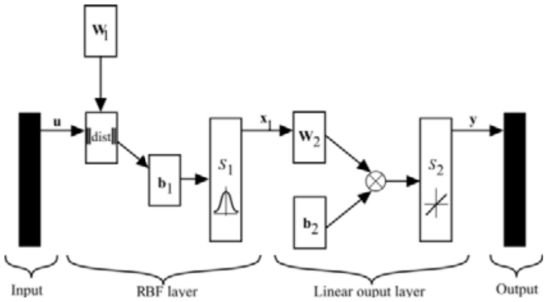

Fig. 1. Schema of radial basis function neural network

RBF networks are popular for their fast training and adaptation. However, these advantages bring some drawbacks. The main disadvantage of RBF network is high memory requirement, because in the classic approach the number of neurons in the hidden layer is equal to the number of training data. In the RBF networks the weights $\mathbf{W}_1$ and biases $\mathbf{b}_1$ of the hidden layer are determined directly from the data. No training is involved. The weights $\mathbf{W}_2$ and biases $\mathbf{b}_2$ of the output layer are determined by supervised learning (Yegnanarayana, 1999). RBF networks following this approach are further denoted as *rbf*. Nevertheless, there was developed improved design method that uses suboptimal solution of the function approximation using fewer RBF neurons in the hidden layer (Demuth et al., 2010), where the training algorithm iteratively adds a RBF neuron to the hidden layer until the training error reaches the desired goal. Such RBF networks will be in the following text symbolized as *rbfu*.

3. SIMULATIONS

Correspondingly to the previous research (Samek, 2009); there were used 5 values of the signal for the prediction. Furthermore, it was necessary to define a parameter that has a key influence to the prediction quality. By many experiments it was chosen the spread parameter, which defines the smoothness of the approximation function. The tested values of spread parameter were as follows: spread=[0.1 0.5 1 5 10 50 100 500 1000 5000]. There were trained and tested one hundred samples of *rbf* for each setting of the spread parameter. The number of neurons in hidden layer was 4875 because of the size of training data. The average values of the results are presented in the figure 2.

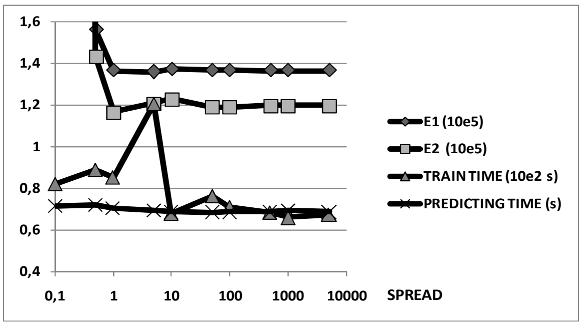

Fig. 2. Results of *rbf*

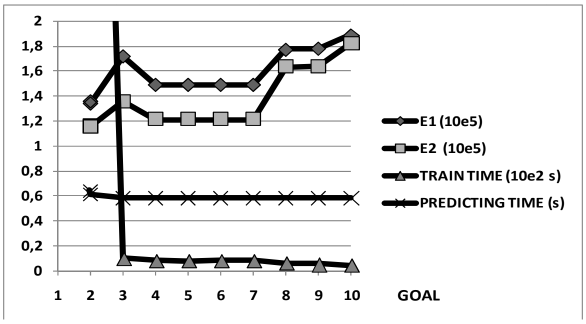

Fig. 3. Results of *rbfu*

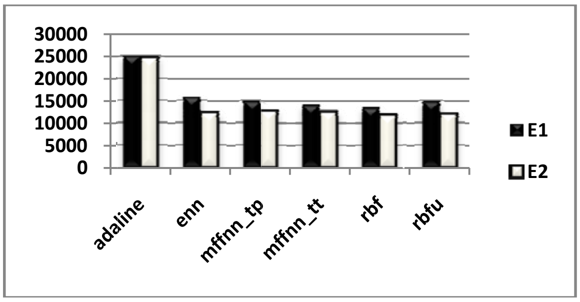

Fig. 4. Comparison of the prediction quality criterions E_1 an E_2

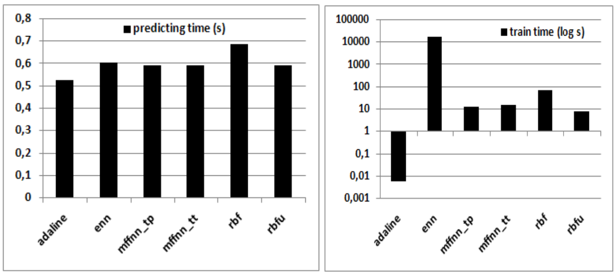

Fig. 5. Comparison of the predicting time and train time

As is depicted in the figure, both criterions E_1 an E_2, that describes the prediction error (Lendasse et al., 2004; Samek, 2009), are out of range for spread = 0.1. However, for the spread higher than 1 both criterion falls down and remain almost same. The training time and predicting time goes down with increasing spread. The sharp peak in the spread = 5 is probably only computational error. From the shown data it could be suggested usage of the spread parameter in the interval <50; 5000>.

In the case of improved design method *rbfu*, the parameter goal was chosen as the driving factor for benchmarking. After many experiments it was decided to test the following values of goal=[1.98 2 3 4 5 6 7 8 9 10]. One hundred of *rbfu* samples were tested for each tested value of the goal parameter. Results (average values) of the simulations are depicted in the figure 3.

As can be seen from the figure, the train time for low values of goal is extremely long. Though, the prediction error described by E_1 an E_2 is the best. Thus, it is necessary to find a compromise between computational demands (short train and predicting time) and prediction accuracy (small E_1 an E_2). From the fig. 3 we can conclude that goal should be set from the interval <4; 7>. Remarkable are also memory requirements because this nets use only about 6 neurons in the hidden layer.

4. RESULTS COMPARISON

In this chapter the results of *rbf* and *rbfu* are compared to performance of *adaline*, *mffnntp*, *mffnntt* and *enn*. The values of criterions E_1 an E_2 are depicted in the figure 4. The computational demands of the predictors (expressed by computation time of the prediction and the time of the predictor training) are illustrated in the figure 5.

From the point of view of prediction quality criterions (Samek, 2009), the best results are provided by *rbf* network, while the second best is multilayered feed-forward neural network with hyperbolic tangent sigmoid transfer function in both layers (*mffnntt*). On the other hand, the *rbf* has the longest predicting time and the average train times are also markedly high. Nevertheless, the improved version of RBF networks *rbfu* performed slightly higher criterions E_1 an E_2, but the training time and prediction time is one of the shortest.

5. CONCLUSION

It can be concluded that artificial neural networks with radial basis function offer interesting way how to model (and predict) time series or nonlinear system output. The prediction accuracy is very good, but the computational requirements are higher. Thus, the given task was not fully fulfilled and further research is necessary.

In applications, where the computation time plays an important role, the improved design *rbfu* should be considered. The weakest prediction accuracy was performed by *adaline*. On the other hand, this network has extremely short training and prediction. Therefore, it might be useful in time-critical applications where the prediction accuracy is not fundamental.

6. ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the grant MSM 7088352102. This support is gratefully acknowledged.

7. REFERENCES

Demuth, H.; Beale, M. & Hagan, M. (2010). Neural Network Toolbox 6: User Guide, *Available from:* http://www.mathworks.com/access/helpdesk/help/pdf_doc/nnet/nnet.pdf *Accessed:* 2010-05-20

Lendasse, A.; Oja, E. & Simula, O. (2004). Time Series Prediction Competition: The CATS Benchmark. *Proc. of IEEE Int. Joint Conf. on Neural Networks*, pp. 1615-1620, ISBN 0-7803-8359-1, Budapest, Jul 2004, IEEE, Budapest

Nelles, O. (2001). *Nonlinear system identification*, Springer, ISBN 3-540-67369-5, Berlin

Samek, D. (2009). Prediction Benchmark of Artificial Neural Networks. *Annals of DAAAM for 2009 & P. of the 20th Int. DAAAM Sym.*, Katalinic, B. (Ed.), pp. 621-622, ISBN 978-3-901509-70-4, Vienna, Nov 2009, DAAAM Int., Vienna

Suykens, J. A. K.; Vandewalle, J. P. L. & De Moor, B. L. R. (1996). *Artificial Neural Net. for Modeling and Control of Non-lin. Sys.*, Kluwer AP, ISBN 0-7923-9678-2, Dordrecht

Yegnanarayana, B. (1999). *Artificial Neural Networks*, Prentice-Hall of India, ISBN 81-203-1253-8, New Delhi