STUDIA UNIV. "BABEŞ–BOLYAI", INFORMATICA, Volume **XLIII**, Number 1, 1998

RBF NEURAL NETWORKS TRAINED WITH COMBINED EVOLUTIONARY AND BACKPROPAGATION METHOD

P. ANDRÁS

Abstract. RBF neural networks with multi-dimensional internal input space are used in this paper. A generalized backpropagation training method combined with evolutionary selection is proposed. The training method adapts simultaneously the weights and the internal parameters of the networks. The evolutionary selection is working with natural coding and directed crossover operator. These can speed up the evolutionary process. The proposed networks and training methodology are used for an approximation task.

1. Introduction

The RBF neural networks are widely used in applications for control, classification, financial analysis and prediction, and others ([5], [6], etc.). They are working in most of the cases with three layers (input, hidden and output). The training of these networks is done generally with simple weight adjusting or with a kind of adapted backpropagation ([2],[4]).

The applications of RBF networks have to solve the optimal selection of the centers of the hidden neurons and the adaptation of the weights of the network. These problems can be serious in some cases. Evolutionary methods ([3]) can help to solve these problems ([1], [7], [8], [9]).

A combined backpropagation and evolutionary selection method is proposed in this paper for the training of RBF neural networks. The proposed evolutionary method uses natural coding and a proper combination rule what allows that functionally close components of a network are exchanged together, and for components with similar functionality. This property makes possible the avoidance of many degenerate descendants what can slow the evolutionary process.

1991 *Mathematics Subject Classification.* 65D15, 65Y10, 68T05, 68Q22.

1991 *CR Categories and Descriptors.* I.2.6 [**Artificial Intelligence**]: Learning – *Concept learning, Connectionism and neural nets*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search – *Backtracking, Heuristic methods*; F.1.1 [**Computation by Abstract Devices**]: Models of Computation; F.1.2 [**Computation by Abstract Devices**]: Modes of Computation .

2. RBF neural networks

The RBF networks, used for this paper, were three layer networks. The weights of the neurons in the hidden layer, were vectors instead of scalars. The transfer function of the hidden layer neurons was:

$$y = e^{-\frac{\sum_{j=1}^{m}(w^{jT}x-c_j)^2}{r^2}}$$

where $c = (c_1, c_2, \ldots c_m)$ is the center of the hidden neuron (the center of the radial function of the neuron), $w^j,\ j = 1, m$ are weights associated to each component of the center, and x is the n-dimensional input vector. We can name these RBF networks as RBF networks with multi-dimensional internal input space because of the m-dimensional centers and m-dimensional weight vectors. These networks are common generalization of the RBF networks with transfer functions like

$$f_1(x) = e^{-\frac{||x-c||^2}{r^2}}$$
$$f_2(x) = e^{-\frac{(w^Tx-c)^2}{r^2}}$$

For the training of the networks we propose the generalized backpropagation training, what adapts all the weights and the internal parameters of the neurons. The upgrade formulas can be determined using the gradient descent method for the squared error function

$$E = (d-y)^2$$

where y is the actual output and d is the desired output.

So we obtain the upgrade formulas:

$$\Delta w_i^{out} = \alpha \cdot (y-d) \cdot e^{-\frac{\sum_{j=1}^{m}(w_{hidden_i}^{jT}x-c_j^i)^2}{r_i^2}};$$

$$\Delta c_j^i = \beta \cdot (y-d) \cdot \frac{2\left(w_{hidden_i}^{jT}x-c_j^i\right)}{r_i^2} \cdot e^{-\frac{\sum_{j=1}^{m}(w_{hidden_i}^{jT}x-c_j^i)^2}{r_i^2}} \cdot w_i^{out};$$

$$\Delta r_i = \gamma \cdot (y-d) \cdot \frac{4\sum_{j=1}^{m}\left(w_{hidden_i}^{jT}x-c_j^i\right)^2}{r_i^3} \cdot e^{-\frac{\sum_{j=1}^{m}(w_{hidden_i}^{jT}x-c_j^i)^2}{r_i^2}} \cdot w_i^{out};$$

$$\Delta w_k^{j,hidden_i} = \eta \cdot (y-d) \cdot \frac{2\left(w_{hidden_i}^{jT}x-c_j^i\right)}{r_i^2} \cdot e^{-\frac{\sum_{j=1}^{m}(w_{hidden_i}^{jT}x-c_j^i)^2}{r_i^2}} \cdot w_i^{out} \cdot x_k;$$

where α, β, γ, η are learning constants ($|\alpha|, |\beta|, |\gamma|, |\eta| \leq 1$).

The advantage of the generalized backpropagation is that it adapts the internal parameters of the hidden neurons too. So the center selection problem is partly solved by the backpropagation training.

3. Evolutionary selection of neural networks

The evolutionary selection of the neural networks can provide neural network individuals with very good performance, faster than only the training of them ([9]).

In many cases in the evolutionary selection of neural networks abstract coding and blind crossover and mutation operator are used ([1], [7], [8], [9]). Here a natural coding and a directed crossover operator is proposed.

The neural networks are coded as strings of hidden neurons. The mutation and crossover operators will operate over the genes formed by a whole hidden neuron. Practically the coding of a hidden neuron is realized by two arrays, the first one is an $m \times n$ array coding the input weights of the neuron for each center component, the second one is an $m+2$ dimensional vector, what contains the radius and the output weight of the neuron in the first two positions, and the m components of the center in the rest.

The mutation will modify with small values the weights and internal parameters of the selected neuron. The directed crossover realizes the crossover between two networks under the direction of a region of the internal input space. This means that a random m-dimensional region is selected in the internal input space, and all of the neurons are exchanged, which have their centers situated in the selected region.

Algorithmically this crossover means:

Step 1. Two networks, a and b, are selected for crossover (n_a, n_b being the number of neurons in the networks).

Step 2. The values of the vector c^{cross} and r^{cross} are selected randomly.

Step 3. List_exch_a:= $\emptyset$; List_exch_b:= $\emptyset$.

Step 4a. $k := 1$;

Step 5a. If $||c_a^k - c^{cross}|| < r^{cross}$ then neuron k form network a is put in the list List_exch_a.

Step 6a. $k := k + 1$, if $k \leq n_a$ then go to Step 5a.

Step 4b. $k := 1$;

Step 5b. If $||c_b^k - c^{cross}|| < r^{cross}$ then neuron k form network b is put in the list List_exch_b.

Step 6b. $k := k + 1$, if $k \leq n_b$ then go to Step 5b.

Step 7. The neurons in List_exch_a are deleted from the network a and are added to the network b, the neurons in List_exch_b are deleted from the network b and are added to the network a.

Step 8. Stop.

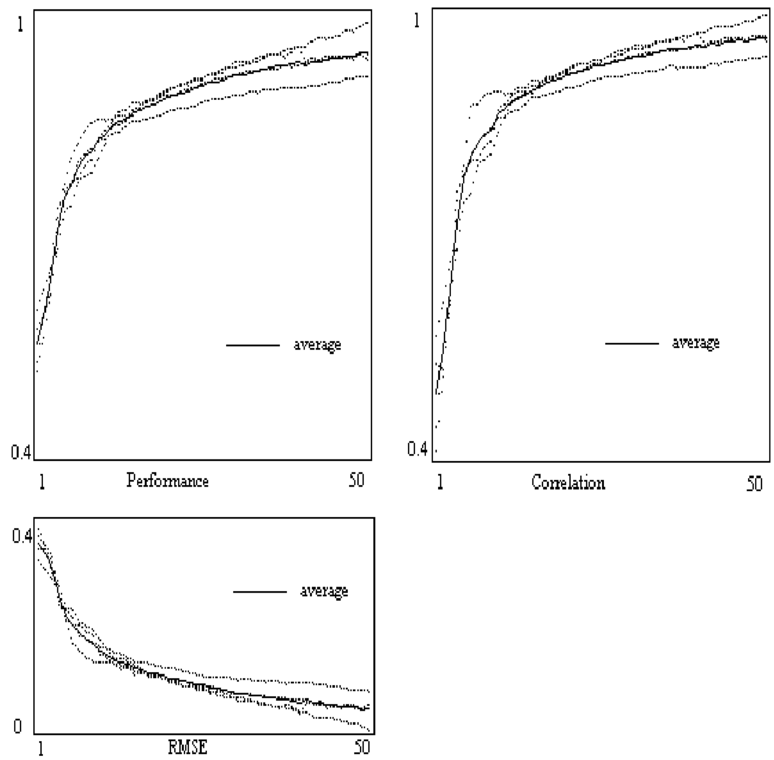

FIGURE 1. Results of the experiments

The proposed method has the advantage that uses a natural coding and a crossover what allows that functionally close components of a network are exchanged together, and for components with similar functionality. This property makes possible the avoidance of many degenerate descendants what can slow the evolutionary process.

4. Results

We used a population of RBF networks. The population was initialized in each case with 50 networks, each having 10 hidden neurons. The parameters of the networks were initialized with random values from $[-1, 1]$ for the weights and $[0, 2]$ for the radius and center components of the hidden neurons. The input space of the networks was set to be 3-dimensional, and practically it was the region $[-\pi, \pi]^3$. The goal of the networks was to approximate the function

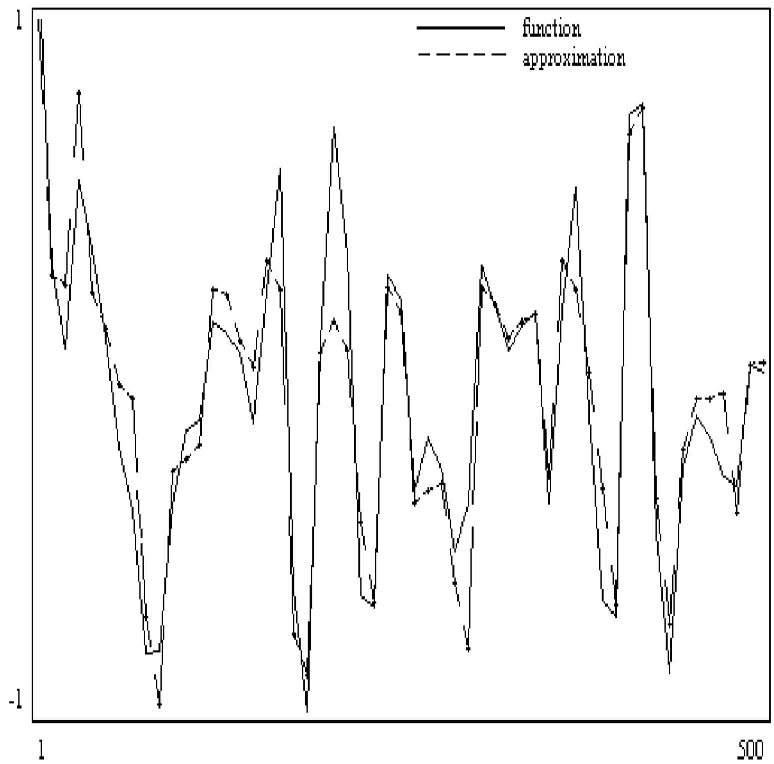

FIGURE 2. Approximation of the target function

$$f(x_1, x_2, x_3) = \frac{\sin x_1 + \sin x_2 + \sin x_3}{3}$$

The networks were trained in each step with 100 random training pairs. The new generation of the networks was created after each training, using the proposed coding and crossover mechanism.

The performance of the networks was determined using the test set composed by the vectors $x^t = (x_1^t, x_2^t, x_3^t)$, with

$$x_k^t = \pi \sin\left(\frac{t\pi}{100} + \frac{(1 + (t\ div\ 15))\, k}{\sqrt{43}}\right)$$

for $t = 1, 500$, and $k = 1, 3$. The correlation between the target data and the network output and the RMSE of the network output was measured. The performance of the networks was calculated as a combination of these measures.

We made four experiments. The results of these are presented in the Figure 1. For each experiment and each step we took into account the best performing network from the network population.

The approximation result is shown in the Figure 2.

5. Conclusions

RBF neural networks with multi-dimensional internal input space were proposed in this paper. The generalized backpropagation combined with evolutionary selection was proposed for the training of them.

The proposed backpropagation training realizes the simultaneous training of the weights and internal parameters of the network, leading to a parallel solution of the weight and center selection problem.

The performance of the backpropagation training is enhanced by the combination with the evolutionary selection. The proposed evolutionary method uses natural coding and directed crossover, what prevents the generation of degenerated descendants, and so, avoids the slowing down of the evolutionary process.

References

[1] S.A. Billings and G.L. Zheng: Radial Basis Function Network Configuration Using Genetic Algorithms, in Neural Networks, vol. 8. no. 6. p. 877-890, 1995

[2] A. Cichocki, R. Ubenhauen: Neural Networks for Optimization and Signal Processing, John Wiley & Sons, Chichester, 1993.

[3] D.E. Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Reading, MA, 1989.

[4] S. Haykin: Neural Networks. A Comprehensive Foundation, IEEE Press, 1994.

[5] V.R. Vemuri, R.D. Rogers (eds.): Artificial Neural Networks. Forecasting Time Series, IEEE Computer Society Press, Los Alamitos, CA, 1994.

[6] D. A. White, D. A. Sofge (eds.): Handbook of Intelligent Control. Neural, Fuzzy and Adaptive Approaches, Van Nostrand Reinhold, New York, NY, 1992.

[7] B.A. Whitehead, T.D. Choate: Cooperative-Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction in IEEE Transactions on Neuronal Networks, vol. 7. no. 4., p. 869-880, 1996.

[8] D.Whitley, T. Hanson: Optimizing neural networks using faster more accurate genetic search, in J.D. Schaffer (ed.): Proceedings of the Third International Conference on GA, p.391-396., Morgan Kaufman, San Mateo, CA, 1989.

[9] X.Yao: A review of evolutionary artificial neural networks, International Journal of Intelligent Systems vol.8., no.4., p.539-567., 1993.

Babeş-Bolyai University, Faculty of Mathematics and Informatics,
RO 3400 Cluj-Napoca, str. Kogălniceanu 1, Romania
E-mail address: peter@civitas.org.soroscj.ro