A NEURAL NET FOR PREDICTION PROBLEMS

Vladimir A. Golovko, Jury V. Savitsky, Vitaly B. Gladishchuk
Brest Polytechnical Institute
Moskovskaja St., 267
224017, Brest,
Republic of Belarus
Phone: (0162) 42-10-81
E-Mail: cm@brpi.belpak.brest.by

Abstract.

In this paper are presents and investigates the various modifications of gradient descent method for Back-Propagation algorithm. Opportunity on reduction of neural network training time and increasing of training algorithm stability are considered. The particular expressions for choosing of optimum steps of training neurons with sigmoid and linear functions of output activity are received. Results of experiments on predicting of various mathematical series and functions by using neural networks are presented. The analysis of speed training neural network for offered training algorithm demonstrates its advantage before classical Back-Propagation algorithm.

Keywords: Neural Net, Back Propagation, Training Speed, Prediction

1. Introduction

At present time are activised scientific investigations in the neural nets area. This scientific direction is priority for artificial intelligence problems. Neural networks allow to model reflection processes of biological organisms in different aspects. After training neural nets possesses ability to generalise and to predict results. This ability allow to use neural net for predicting different situations. Example of neural net for predicting time series is presented in this work. In this case it is used neural net with feed-forward connections. Choice of network structure and elaboration of effective training algorithms are main construction problems of this networks. In most cases the Back Propagation algorithm take as a principle of the multilevel neural nets training [1-3]. It is based on the gradient descent method in the weights area. Main deficiencies of this method are large training time and constant training speed. The constant training speed value makes gradient method unstable, if speed so large, but small training speed leads to very large training time of neural net. Therefore this paper views the modifications of gradient descent algorithm for predicting neural net. It basic is method of steepest descent and conjugate gradients. At result from behind dynamic adaptation of training step increases the training speed of multilevel neural net, and Back Propagation algorithm is more stable. It was confirmed by modelling results of prediction net for time series.

2. Base results

Consider the training algorithms for multilevel feed-forward neural nets. As mentioned above, constant training speed makes uneffective the gradient descent method at the Back Propagation algorithm. According to this algorithm connection weights from neural element i to neural element j are calculated as:

$$Wij(t+1) = Wij(t) - \alpha \cdot \nabla E(Wij(t)) \tag{1}$$

In this Formula $\nabla E(Wij(t) = \partial E / \partial Wij(t)$ - gradient of fınction E, α - the training speed, E - error fimction, it's defined as

$$E = \frac{1}{2}\sum_{j} (Yj - Dj)^2 \tag{2}$$

where Yj - the real output value of neural element, Dj - the corresponding training exemplar.

Define kj as neural element threshold. For adaptation the neural element threshold necessary to make all thresholds with null values and instead it to add the supplementary connection with weights equal kj. Then neural element will be modified in the training process as:

$$kj(t+1) = kj(t) - \alpha \cdot \nabla E(kj(t)) \tag{3}$$

For stability and for increasing work speed of the Back Propagation algorithm necessary to make adaptive the neural net training speed. Method of steepest descent is suitable for this purpose. In accordance with this method:

$$Wij(t+1) = Wij(t) - \alpha \cdot \nabla E(Wij(t)), \tag{4}$$

$$kj(t+1) = kj(t) - \alpha \cdot \nabla E(kj(t)), \tag{5}$$

$$\text{where} \quad \alpha(t) = \min\left[E(Wij(t+1), kj(t+1))\right], \tag{6}$$

In order to predict fimctions or time series can be used both linear and sigmoid activation fimctions. For linear activation fimction:

$$Yj = M \cdot Xj,$$

$$Xj = \sum_{i} Yj \cdot Wij(t+1) - kj(t+1),$$

where Yi - output value of the neural element i. Then from (6) training speed will be:

$$\alpha(t) = \frac{1}{M^2 \cdot (1 + \sum_{i} Yi^2)} \tag{7}$$

It is easy to show that multilayer neural net with feed-forward connections for liner activation fimction of the neural element is equivalent to a two-layer neural net and for that neural net Back-Propagation algorithm is simple. Only weights and thresholds of the one layer must have set. For linear activation function of the neural elements:

$$\nabla E(Wij(t)) = \gamma_j \cdot M \cdot Yi,$$
$$\nabla E(kj(t)) = -M \cdot \gamma_j,$$

then training rule of the neural net is next:

$$Wij(t+1) = Wij(t) - \alpha(t) \cdot \gamma_j \cdot M \cdot Yi,$$
$$kj(t+1) = kj(t) + \alpha(t) \cdot M \cdot \gamma_j,$$

$$\text{where } \gamma_j = Yj - Dj.$$

Now receive expressions for sigmoid activation function of the neural elements. In this case output value of the neural element j defined as:

$$Yj = \frac{1}{1 + e^{-Xj}}$$

$$\text{where } Xj = (\sum_i Yi \cdot Wij) - Tj.$$

Using the Taylor series of Yi and (6) we receive next expression:

$$\alpha(t) = \frac{4 \cdot \sum_j \gamma_j^2 \cdot Y_j \cdot (1 - Y_j)}{\sum_j \gamma_j^2 \cdot Y_j^2 \cdot (1 - Y_j)^2 \cdot (1 + \sum_i Y_i^2)}$$

where $\gamma_j = Y_j - D_j$ - for output layer and $\gamma_j = \sum_k \gamma_k Y_k (1 - Y_k) W_{jk}$ - for hidden layers, k - number elements in the next layer concerning element j. For sigmoid activation function of the neural elements every neural net layer has its training speed $\alpha(t)$. If neural net use different activation function for every layer then training step will be elected using different expressions for every layer. Receiving in this section expressions for choosing training step allow to raise effect of the Back Propagation algorithm.

3. Experimental results

Consider obtaining algorithms (see section about) and gradient method for comparison. Consider example of the simple neural net for predicting mathematical functions and series. Let we have next function: $Y = 0.5X^2 + 1$ and we now its first six values:

X	0	1	2	3	4	5
Y	1.0	1.5	3.0	5.5	9.0	13.5

It is necessary to predict the rest of values of this function. For neural net training and for predict will be used sliding mean value method with window length |W|=3 .Every next training vector will be get by displacement to the right. Proceed from the above-stated neural net will be consist from three input neural element and one output element. Conventional gradient method with constant training step was used for training this network. Training time with error:

$$E = \frac{1}{2}\sum_{L}\sum_{j}(Y_j^L - D_j^L) = 0.00001$$

was equally 20 minutes.

For using steepest descent method and training speed by expression (7) training time with total error 10^{-10} was equally 6 seconds. After training this network predict any value above-stated function. Also we have analogous experiment with function $Y= SIN(X)$. Value X was change with step 6 degrees. Neural net for this function consist from 10 input neural elements and 1 output neural element. There are 20 training vectors were used for training process. For conventional gradient method the training time with total error $E = 8.28 \times 10^{-4}$ was equally 2 minutes. For steepest descent method this time with total error $E = 10^{-6}$ was equally 25 seconds. Neural net predict any value this function as well as in above case. This neural nets was simulated using conventional computer (IBM PC AT).

4. Conclusion

In this paper it is proposed and studied different modifications of the gradient descent method of the Back Propagation algorithm. We exceed concrete expressions of the training speed for sigmoid and for linear activation function of the neural element. The training process of the proposed algorithms contain both weight modifications of the neural elements and threshold modifications of the neural elements. Simulating results of the neural nets demonstrate advantage before conventional algorithms.

References

[1] Utans Joachim, Moody John. Selecting Neural Network Architectures via the Prediction Risk: Application to Bond Rating Prediction. /Proceedings of the First International Conference on Wall Street. IEEE Computer Society Press, Los Alamitos, CA, 1991.

[2] Hertz John A. Introduction to the Theory of the Neural Computation. /John A.Hertz, 1991, Addison-Wesley Publishing Company, 115-133.

[3] Chan, L.W. and Fallside, F. An adaptive training algorithm for backpropagation networks,Computer Speech and Language, Vol. 2, 1987, 205-218.