QUASINEWTONIAN ACCELERATION OF TRAINING: A SIMPLE TEST

Serge Ye. Gilev, Alexander N. Gorban

Krasnoyarsk Computing Center RAS, Krasnoyarsk State Technical University
660036, Krasnoyarsk-36, Computing Center RAS, Russian Federation
E-mail: gorban@cc.krascience.rssi.ru

Donald C. Wunsch II
Applied Computational Intelligence Laboratory
Department of Electrical Engineering, Texas Tech University
Lubbock, TX 79409-3102
E-mail: Dwunsch@aol.com

Three supervisor methods of training neural networks have been compared in a test. The essence of all methods compared is to minimize step-by-step the function of estimate H and differ from one another by the direction along which a step of this minimization is made:
1) to optimize in random direction;
2) to optimize in direction of H antigradient (steepest descent);
3) to correct this direction by the quasinewtonian single-step method (BFGS formula).
The comparison was made on two standard problems:
1) to recognize the direction of cyclic shift of 0-1 sequence;
2) to recognize the symmetry of 0-1 sequence.

1. Algorithms

The steepest descent direction - the antigradient of the estimate H is traditionally considered not very convenient for optimization. All textbooks on optimization cite examples of rather simple functions which the steepest descent method fails to minimize.

In the cases when the matrix of second derivatives is positive definite the direction considered the best is the Newtonian direction

$$D_2^{-1}\mathrm{grad}H \tag{1}$$

where D_2 is the matrix of H second derivatives.

Quadratic forms are minimized by (1) in one step. This formula is, however difficult to realize by many reasons.

1) **Time**. Searching for all second derivatives and inversion of D_2 takes too much computational costs.

2) **Memory**. To solve problems of high dimension it takes N^2 elements of D_2 - which is too many.

3) And if D_2 **is not positive definite**?

Contrivances to cope with these difficulties galore. Actually all literature on smooth optimization methods is dedicated to this issue.

Idea one. Not to differentiate H twice and not to invert the matrix D_2, but to use the iteration procedure that in many steps makes possible to accumulate “something like” D_2^{-1}.

Idea two. To store in the memory not the matrix but results of its effect on some vectors only.

The idea of quasinewtonian methods with limited memory is to search for the correction to the deepest descent direction as an effect of the small rank matrix. The matrix is not kept in the memory and its effect on the vectors is constructed by the scalar products for several specially selected vectors.

A simple and fairly efficient method is based on BFGS formula [1,2] and makes use of the results of the previous step. To describe it denote: s_k is the direction of descent at the k-th step; h_k is the value of the k-th step (the k-th step is the shift by $h_k s_k$; g_k is the gradient of the estimate function at the initial point of the k-th step; $y_k = g_{k+1}$-g_k is the change of the gradient as a result of the k-th step. The final point of the k-th step is, of course, the initial point for the $k+1$-th.

BFGS formula for the descent direction at the $k+1$-th step is:

$$s_{k+1} = -g_{k+1} + \frac{(s_k, g_{k+1}) y_k + (y_k, g_{k+1}) s_k}{(y_k, s_k)} - h_k s_k \frac{(s_k, g_{k+1})}{(y_k, s_k)} - \frac{(y_k, y_k)(s_k, g_{k+1})}{(y_k, s_k)^2} s_k \qquad (2)$$

After a certain sequence of k steps from time to time it is expedient to get back to the steepest descent - to restart. It is also used when a recurrent s_{k+1} is not a very appropriate direction of the descent and moving along it results either in a too small step or does not bring any improvement at all.

BFGS formula was several times proposed and used to train the neural networks [3-5].

2. Tests

The problem of training neural networks is considered as a problem of minimizing the mean estimate by the training set. The methods compared were:

A) searching in random directions (Rnd);

B) steepest descent (SD);

C) one of quasinewtonian methods (BFGS).

The methods were compared by the "training speed" criterion i.e. by the number of operating cycles of the neural network required to train a network to recognize all examples in a training sampling. (One operating cycle of a neural network is a discrete time unit in which all neurons exchange their signals). One-dimensional optimization in the direction of the step was done by parabolic search.

Comparison was made for two problems of binary classification:

1) of recognition of the direction of cyclic shift of a finite binary sequence
(with elements 0, 1);

2) of recognition of symmetry or non-symmetry of a finite binary sequence.

The direction of the shift was recognized for sequences of 8 elements. The initial sequence and a sequence shifted one position left or right was applied to the network input (total of 16 input signal). The textbook volume was 504 examples.

In the problem of recognizing the symmetry the sequence length was assumed to be 10, yielding the textbook volume of 1024 examples, of which only 32 belonged to the class of symmetrical sequences. In this connection in the last problem the estimate of symmetrical examples was taken with the weight 31.

Use was made of full connected neural networks with the neuron activation function $\varphi = x/(c+|x|)$, where c is the steepness ("characteristic") of the sigmoid function. The value of c used in experiment was equal to 0.1.

Minimization for the total estimate of all textbook examples was made. The training was ceased when all examples from the training sampling were recognized correctly.

To compare the methods 100 neural networks have been generated for each of the test problems (to test the training of the symmetry recognition problem by the Rnd method use was made of the first 20 neural networks only). The initial values of "synaptic weights" of the neural networks were taken randomly with uniform distribution over the interval of (-0.01; 0.01).

Table 1. Comparison of training methods. Problem of recognizing the shift direction.

Method	AV	σ	Min	Max
Rnd	7713	2499	3680	16259
SD	2132	935	839	6199
BFGS	296	102	191	1028

Table 2. Comparison of training methods. Problem of recognizing the sequence symmetry.

Method	AV	σ	Min	Max
Rnd	33114	12310	18190	65150
SD	2820	1053	1180	7115
BFGS	731	259	388	1831

The training times in both tables are given in thousands of operating cycles of the neural networks. Column AV is mean training time; column σ is mean quadratic deviation; columns Min and Max are minimum and maximum times, respectively.

The results obtained make possible to the following conclusions:

1) the Rnd method can be competitive provided with the hardware implementation of the neural network the cost of memorizing the signal is approximately 50 times more than the cost of signal exchange;

2) the acceleration the quasinewtonian method yields is 5-10 times more compared to the steepest descent method.

REFERENCES

1. Shanno D. *Conjugate Gradient Methods with Inexact Searches*. Mathematics of Operations Research, 1978. V. 3, No.3.

2. Gill Ph.E., Murrey W., Wright M. *Practical Optimization*. Academic Press, 1981.

3. Gorban A.N. *Traning Neural Networks*. Moscow, USSR-USA JV "ParaGraph" 1990. – 160 pp. (Russian) (English Translation: AMSE Transaction, Scientific Siberian, A, 1993, Vol. 6. Neurocomputing. Tassin (France): AMSE Press, pp.1-134).

4. Saad E.W., Prochorov D.V., Wunsh II D.C. *Adwanced Neural Network Traning Methods for Low False Alarm Stock Trend Predicion.* Proc. of the International Conference on Neural Networks (ICNN'96), June 3-6, 1996, Washington D.C., IEEE, 1996, V.4, pp. 2021-2026.

5. Gilev S.E., Gorban A.N. *On Completeness of the Class of Functions Computable by Neural Networks*, Proc. of the World Congress on Neural Networks (WCNN'96), Sept. 15-18, 1996, San Diego, CA, Lawrence Erlbaum Associates, 1996, pp. 984-991.