Technique of Learning Rate Estimation for Efficient Training of MLP

Vladimir Golovko[1], Yury Savitsky[1], T. Laopoulos[3], A. Sachenko[2], L. Grandinetti[4]

[1] Brest Polytechnic Institute, Belarus, cm@brpi.belpak.brest.by
Laboratory of Artificial Neural Networks, Brest Polytechnic Institute, Moskowskaja 267, Brest, Belarus, fax:+375 162 422127
[2] Aristotle University of Thessaloniki, Greece, laopoulos@physics.auth.gr
[3] Ternopil Academy of National Economy, Ukraine, sachenko@cit.tane.ternopil.ua
[4] University of Calabria, Italy, lugran@unical.it

Abstract. A new computational technique for training of multilayer feed-forward neural networks with sigmoid activation function of the units is proposed. The proposed algorithm consists two phases. The first phase is an adaptive training step calculation, which implements the steepest descent method in the weight space. The second phase is estimation of calculated training step rate, which provide reach a state of activity of the units on the each training iteration. The simulation results are provided for the test example to demonstrate the efficiency of the proposed method, which solves the problem of training step choice in multilayer perceptrons.

1. Introduction

Multilayer perceptrons (MLP) form a wide set of feed-forward neural networks. They have a wide variety of applications in different areas: classification, control, pattern recognition, function approximation, prediction etc. Adaptability of the neural models for any application provides by training procedures. The most commonly training methods in MLP is error backpropagation (BP) algorithm [2], [3]. In spite of the fact that BP is successfully used for various kind of tasks, it have lacks such as slow convergence, non-stability of convergence and local minimum problems [4]. Many efforts have been made to development the MLP training methods using in BP variable training step size [5], [11], layer-by-layer optimization [12] and using for training the Newton method [6], [7], the Levenberg-Marquardt method [8], conjugate-gradient technique [9], [10].

In this paper a new method is developed for efficient training of MLP by combining BP, adaptive training step calculation (ATS) technique [1] and training step size estimation (SSE) method. The ATS is used to find an optimal training step, which minimizes the neural unit training error. The SSE is used to validation the training step size for guaranteeing of neural unit training adaptability. This technique along with BP allows to solve the problems of optimal learning rate choice and to advance of MLP adaptability, as demonstrated the various experiments.

In this work Section 2 will discuss the architecture of MLP. Proposed ATS technique and SSE technique is discussed in section 3. Section 4 gives the computational experiments along with discussions, confirming good performance of the proposed learning methodology. To end, Section 5 gives conclusions.

2. MLP Architecture

The basic MLP architecture is shown in Fig. 1. This layered structure has 1 input, 1 output and $L-2$ hidden layers. The model of a typical neuron in MLP shown in Fig. 2. The output of neural unit for layer l can be expressed as:

$$g(S_j^{[l]}) = \left(1 + e^{-S_j^{[l]}}\right)^{-1} \tag{1}$$

where $S_j^{[l]}$ is weighted sum of input activity of this unit, defined as:

$$S_j^{[l]} = \sum_{i=0}^{N^{[l-1]}} y_i^{[l-1]} w_{ij}^{[l]} \tag{2}$$

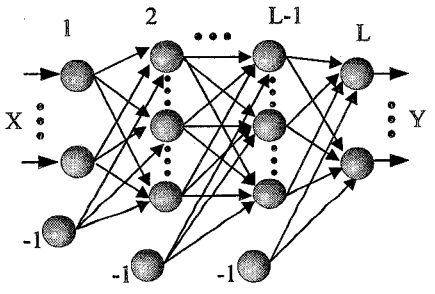

Fig. 1. The MLP architecture

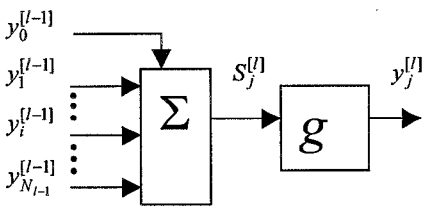

Fig. 2. Model of neural unit j for the layer l

3. Training Method

This section describes the proposed learning method as weight updation scheme. The computational scheme of the proposed ATS technique and SSE technique starts as follows.

An initial point is chosen in the weight space at random. The new point is adaptive training step definition and it validation for guaranteeing of neuron training activity after weight updation phase. The next point is weight updation by using BP method.

3.1. BP Technique

The most popular training algorithm for MLP is BP and is described in brief with the following notations.

$y_j^{[l]}$ Output of the j th unit in layer l.

$w_{ij}^{[l]}$ Weight connecting i th unit in layer $l-1$ to j th unit in layer l.

x^p Input p th training sample.

d^p Desired p th training sample.

L Number of layers.

N^l Number of units in layer l.

P Number of training patterns.

t Number of current training iteration.

In this notations $w_{0j}^{[l]}$ represents weight connecting i th unit in the bias layer to j th unit in layer l and $y_0^{[l-1]} = -1$.

BP implements a gradient search technique to find the network weights, that minimizes the squared error function given below:

$$E(t) = \sum_{p=1}^{P} E^p(t) = \sum_{p=1}^{P} \sum_{k=1}^{N^{[L]}} \left(y_k^{[L]}(t) - d_k^{[L]}\right)^2 \tag{3}$$

The weights of MLP are updated iteratively according to following rule:

$$w_{ij}^{[l]}(t+1) = w_{ij}^{[l]}(t) - \alpha \frac{\partial E^p}{\partial w_{ij}^{[l]}} \tag{4}$$

Here $\frac{\partial E^p}{\partial w_{ij}^{[l]}}$ is gradient error defined as

$$\frac{\partial E^p}{\partial w_{ij}^{[l]}} = \frac{\partial E^p}{\partial y_j^{[l]}} \frac{\partial y_j^{p,[l]}}{\partial S_j^{p,[l]}} \frac{\partial S_j^{p,[l]}}{\partial w_{ij}^{[l]}}, \quad \frac{\partial E^p}{\partial y_j^{[l]}} = \gamma_j^{p,[l]}, \quad \frac{\partial y_j^{p,[l]}}{\partial S_j^{[l]}} = g'(S_j^{p,[l]}), \quad \frac{\partial S_j^{p,[l]}}{\partial w_{ij}^{[l]}} = y_i^{p,[l-1]}, \tag{5}$$

$\alpha > 0$ is a constant, called training step.

Before training the weights are initialize by small random values. The BP training rule (4) is repeated for all training samples until then will achieved acceptable squared error (3).

3.2. Using the ATS Technique in BP

In BP there is a problem of choice of an optimal training step [5], [11]. For choice of adaptive step it is possible to use a method of steepest descent [1]. According to it, the training step $\alpha^{p,[l]}(t)$ for layer l is selected by minimizing a square error $E^p(t)$ for training sample p as given below:

$$\alpha^{p,[l]}(t) = \min E^p(y_j^{p,[L]}(t+1)), j = 1, N^{[L]} \tag{6}$$

The expression for ATS calculation, considered in [7], is follows:

$$\alpha^{p,[l]}(t) = \frac{\sum_{j=1}^{N^{[l]}} (\gamma_j^{p,[l]})^2 g'(S_j^{p,[l]})}{g'(0) \cdot (1 + \sum_{i=1}^{N^{[l-1]}} (y_i^{p,[l-1]})^2) \sum_{j=1}^{N^{[l]}} \gamma_j^{p,[l]^2} (g'(S_j^{p,[l]}))^2} \tag{7}$$

In expression (7) was used decomposition of activation function g under the Tailor series and limitation by first two members. Therefore this is approximate method of ATS definition for nonlinear activation functions. For MLP efficient

training using ATS it is necessary to limit the ATS size [1]. The limitation size is defined empirically. Therefore there is a problem to definition of acceptable bounds of the training step.

Let's consider the next problem connected with incorrect choice of the training step size.

During training phase all neural units of MLP are divided on subset of neural units with small training efficiency and on subset of neural units with large training efficiency. The training efficiency for the neural unit j on the layer l is defined by size of activation function derivative $g'(S_j^{p,[l]})$. For sigmoid nonlinearity it is defined as

$$g'(S_j^{p,[l]}) = \frac{\partial y_j^{p,[l]}}{\partial S_j^{p,[l]}} = g(S_j^{p,[l]})(1 - g(S_j^{p,[l]})) = y_j^{p,[l]}(1 - y_j^{p,[l]}) \tag{8}$$

When training step size is too large, it provides in the most cases to form output activity closely to 0 or to 1 for the next training iterations. In this cases there is reduction of training adaptability of this units, as $g'(S_j^{p,[l]}) \to 0, \Rightarrow \frac{\partial E^p}{\partial w_{ij}^{[l]}} \to 0$ (see Fig. 3). As result there is reduction of the training efficiency and reduction adaptability of MLP.

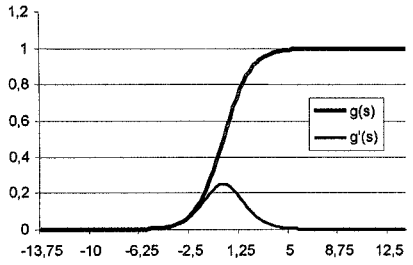

Fig. 3. The sigmoid activation function and it derivative

For solving of described above problems consider below the developed in this paper the technique of ATS validation.

3.3. SSE Technique

As criterion for definition of acceptable bounds of the training step in this work is used minimum acceptable of activation function derivative rate ε. During the training it is necessary to provide the next inequality:

$$g'(S_j^{p,[l]}(t+1)) > \varepsilon \tag{9}$$

Solve this inequality for sigmoid nonlinearity according $\alpha^{p,[l]}$:

$$y_j^{p,[l]}(t+1)(1 - y_j^{p,[l]}(t+1)) > \varepsilon \tag{10}$$

where

$$y_j^{p,[l]}(t+1) = g\left(S_j^{p,[l]}(t+1)\right)$$

Along the expression (4) we can get:

$$\frac{1}{2} - \frac{1}{2}\sqrt{1-4\varepsilon} < y_j^{p,[l]}(t+1) < \frac{1}{2} + \frac{1}{2}\sqrt{1-4\varepsilon}\,. \tag{11}$$

As sigmoid function is increasing and monotonic, it is possible to transformation this inequality to the next form:

$$g^{[-1]}\left(\frac{1}{2} - \frac{1}{2}\sqrt{1-4\varepsilon}\right) < S_j^{p,[l]}(t+1) < g^{[-1]}\left(\frac{1}{2} + \frac{1}{2}\sqrt{1-4\varepsilon}\right), \tag{12}$$

where $g^{[-1]}$ is inverse function, defined as:

$$g^{[-1]}(y) = \ln\frac{y}{1-y}$$

Then the expression (12) can be presented as follows:

$$\ln\frac{1-\sqrt{1-4\varepsilon}}{1+\sqrt{1-4\varepsilon}} < S_j^{p,[l]}(t+1) < \ln\frac{1+\sqrt{1-4\varepsilon}}{1-\sqrt{1-4\varepsilon}} \tag{13}$$

In this expression the $S_j^{p,[l]}$ is calculated as

$$S_j^{p,[l]}(t+1)=\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l]}\left(w_{ij}^{[l]}(t)-\alpha^{p,[l]}(t)\frac{\partial E^p}{\partial w_{ij}^{[l]}}\right)=S_j^{p,[l]}(t)-\alpha^{p,[l]}(t)\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l]}\frac{\partial E^p}{\partial w_{ij}^{[l]}} \tag{14}$$

Final solving for inequality (9) is follows:

$$\begin{cases} \dfrac{S_j^{p,[l]}(t)-\ln\dfrac{1-\sqrt{1-4\varepsilon}}{1+\sqrt{1-4\varepsilon}}}{\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\dfrac{\partial E^p}{\partial w_{ij}^{[l]}}}>\alpha^{p,[l]}(t)>\dfrac{S_j^{p,[l]}(t)-\ln\dfrac{1+\sqrt{1-4\varepsilon}}{1-\sqrt{1-4\varepsilon}}}{\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\dfrac{\partial E^p}{\partial w_{ij}^{[l]}}}, if \sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\frac{\partial E^p}{\partial w_{ij}^{[l]}}>0 \\ \dfrac{S_j^{p,[l]}(t)-\ln\dfrac{1-\sqrt{1-4\varepsilon}}{1+\sqrt{1-4\varepsilon}}}{\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\dfrac{\partial E^p}{\partial w_{ij}^{[l]}}}<\alpha^{p,[l]}(t)<\dfrac{S_j^{p,[l]}(t)-\ln\dfrac{1+\sqrt{1-4\varepsilon}}{1-\sqrt{1-4\varepsilon}}}{\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\dfrac{\partial E^p}{\partial w_{ij}^{[l]}}}, if \sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\frac{\partial E^p}{\partial w_{ij}^{[l]}}<0 \end{cases} \tag{15}$$

Let's consider in detail application of the given expression for SSE. In (14) the expression $\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\frac{\partial E^p}{\partial w_{ij}^{[l]}}$ specifies the direction of gradient search for $S_j^{p,[l]}$ during training. If $\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\frac{\partial E^p}{\partial w_{ij}^{[l]}}<0$ this implyies that $S_j^{p,[l]}(t)<S_j^{p,[l]}(t+1)$ and correspondingly that $y_j^{p,[l]}(t)<y_j^{p,[l]}(t+1)$ for unit j of layer l. In this case it is necessary to use the next limitation for $\alpha^{p,[l]}(t)$:

$$\alpha^{p,[l]}(t)<\frac{S_j^{p,[l]}(t)-\ln\dfrac{1+\sqrt{1-4\varepsilon}}{1-\sqrt{1-4\varepsilon}}}{\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\dfrac{\partial E^p}{\partial w_{ij}^{[l]}}} \tag{16}$$

In other case, if $\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\frac{\partial E^p}{\partial w_{ij}^{[l]}}>0$ then after next training iteration $S_j^{p,[l]}(t)>S_j^{p,[l]}(t+1)$ and correspondingly $y_j^{p,[l]}(t)>y_j^{p,[l]}(t+1)$. In this case it is necessary to use the next limitation for $\alpha^{p,[l]}(t)$:

$$\alpha^{p,[l]}(t)<\frac{S_j^{p,[l]}(t)-\ln\dfrac{1-\sqrt{1-4\varepsilon}}{1+\sqrt{1-4\varepsilon}}}{\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\dfrac{\partial E^p}{\partial w_{ij}^{[l]}}} \tag{17}$$

As $\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]}\frac{\partial E^p}{\partial w_{ij}^{[l]}}=\gamma_j^{p,[l]}g'(S_j^{p,[l]})\sum_{i=0}^{N^{[l-1]}}(y_i^{p,[l-1]})^2$ and $g'(S_j^{p,[l]})>0$, then for definition of gradient search direction in expressions (16), (17) it is necessary to analyze only $\gamma_j^{p,[l]}$.

So, the SSE method includes the phase of analyzing of unit error $\gamma_j^{p,[l]}$ and phase of application of expression (16) or (17) for ATS validation.

The next section illustrates the technique of the SSE application in computational algorithm.

3.4. Computational Scheme

In this section the proposed learning algorithm is expressed as computational steps as given below.

1) *Initialize MLP by small random weights.*
2) *Choice the limitation of the sigmoid derivative* ε.
3) *BP*

3.1) Choose the limitation of square error, at which gradient descent will come to a halt.
3.2) Repeat
3.2.1) Feed Forward
3.2.2) Compute Gradient
3.2.3) ATS
3.2.4) ESS: adjust the training step size for each unit j on layer l using the next rules:

$$\alpha^{p,[l]}(t) < \begin{cases} \dfrac{S_j^{p,[l]}(t) - \ln\dfrac{1+\sqrt{1-4\varepsilon}}{1-\sqrt{1-4\varepsilon}}}{\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]} \dfrac{\partial E^p}{\partial w_{ij}^{[l]}}}, if\ \gamma_j^{p,[l]} < 0 \\ \dfrac{S_j^{p,[l]}(t) - \ln\dfrac{1-\sqrt{1-4\varepsilon}}{1+\sqrt{1-4\varepsilon}}}{\sum_{i=0}^{N^{[l-1]}} y_i^{p,[l-1]} \dfrac{\partial E^p}{\partial w_{ij}^{[l]}}}, if\ \gamma_j^{p,[l]} > 0 \end{cases}$$

3.3) Update weights using adjusted training step.

So, implementation proposed computational algorithm provides the saving of training adaptability for MLP units during training phase.

The next section gives the simulation results and demonstrates the computational effort of the BP algorithm with combination ATS technique and SSE technique.

4. Simulation Results and Discussion

To assess the performance of the proposed learning technique experiments were conducted on standard problems of parity. Here the output of the MLP is required to be '1' if the input pattern contains an odd number of '1' and '0' otherwise. In this problem the most similar patterns which differ by a single bit require different answer. For simulation a three-layer MLP of size 4-4-1 is considered. The training set contains 16 samples. Ten different series of experiments are considered for various training conditions and the average results is provided in the Table 1. The simulation results demonstrate the efficiency of SSE technique for ATS validation.

TABLE 1
Table showing the simulation results for various training conditions: 1) using constant training step; 2) using ATS technique with various constant limitations; 3) combining the ATS technique and SSE technique. NIT - number of training iterations; MSE - mean square error.

NIT	MSE	Type of training step	Size of the training step	ATS validation	ε
2010	0.0241	Constant	0.099	-	-
2350	0.0067	Constant	0.50	-	-
2140	0.0941	Constant	0.99	-	-
2500	0.091	ATS	-	<0.50	-
1990	0.078	ATS	-	<0.99	-
2090	0.0096	ATS	-	<1.99	-
2050	0.267	ATS	-	<2,50	-
2100	0.0052	ATS	-	SSE	0.099
2250	0.0051	ATS	-	SSE	0.05
2310	0.0292	ATS	-	SSE	0.0099

5. Conclusions

In this paper a new algorithm based on combining the adaptive training step technique and training step size estimation technique in backpropagation is proposed. The SSE is used to validation the training step size for guaranteeing of neural unit training adaptability. This technique along with BP allows to solve the problems of optimal learning rate choice and to advance of MLP adaptability. The testing example demonstrates efficiency of proposed method of the training step size estimation.

Acknowledgments

This paper is supported by INTAS program "INTAS OPEN 97-0606". The authors express gratitude to the European Union for financial support.

References

[1] Vladimir Golovko, Yury Savitsky, "New Approach of the recurrent Neural Network Training", Proc. of the Int. Conf. on Neural Networks and Artificial Intelligence ICNNAI'99, 12-15 October 1999, Brest, Belarus, - pp. 32-35

[2] G. E. Hinton, "How neural networks learn from experience," *Sci. Amer.*, pp. 145-151, Sept. 1992.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by backpropagating errors," *Nature,* vol. 323, pp. 533-536, 1986.

[4] R. Beale and T. Jackson, *Neural Computing: An Introduction.* Bristol, U.K.: Inst. Phys., 1992.

[5] X.-H. Yu and G.-A. Chen, "Efficient backpropagation learning using optimal learning rate and momentum." *Neural Networks,* vol. 10, no. 3, pp. 517-527, 1997.

[6] R. Battiti, "First- and second-order methods for learning: Between steepest descent and Newton methods," *Neural Comput.,* vol. 4, pp. 141-166, 1992.

[7] S. Osowski, P. Bojarczak, and M. Stodolski, "Fast second-order learning algorithm for feedforward multilayer neural networks and its applications," *Neural Networks,* vol. 9, no. 9, pp. 1583-1596, 1996.

[8] T. H. Martin and B. M. Mohammad, "Training feedforward network with Marquardt algorithm," *IEEE Trans. Neural Networks,* vol. 5, pp. 959-963, Nov. 1996.

[9] E. M. Johansson, F. U. Dowla, and D. M. Goodman, "Backpropagation learning for multilayer feedforward neural networks using the conjugate gradient method," *Int. J. Neural Systems,* vol. 2, no. 4, pp. 291-302, 1992.

[10] M. S. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks,* vol. 6, no. 4, pp. 525-534, 1993.

[11] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis, "Effective backpropagation training with variable stepsize," *Neural Networks,* vol. 10, no. 1, pp. 69-82, 1997.

[12] G.-J. Wang and C.-C. Chen, "A fast multilayer neural-network training algorithm based on the layer-by-layer optimizing procedures," *IEEE Trans. Neural Networks,* vol. 7, pp. 768-775, May 1996.