Tikrit Journal of Pure Science 22 (2) 2017 *ISSN: 1813 – 1662 (Print)*
E-ISSN: 2415 – 1726 (On Line)

A new learning rate based on Andrei method for training feed-forward artificial neural networks

Khalil K. Abbo[1], Hassan H. Abrahim [2], Firdos A. Abrahim[3]

[1] *Department of Mathematics, College of Computer Sciences & Mathematics, University of Mosul , Mosul , Iraq*
[2] *Department of Mathematics, College of Computer Sciences & Mathematics, University of Tikrit , Tikrit , Iraq*
[3] *Department of Mathematics, College of Education, University of Tikrit , Tikrit , Iraq*

Abstract

In this paper we developed a new method for computing learning rate for Back-propagation algorithm to train a feed-forward neural networks. Our idea is based on the approximating the inverse Hessian matrix for the error function originally suggested by Andrie. Experimental results show that the proposed method considerably improve the convergence rate of the Back-propagation algorithm for the chosen test problem.

1. Introduction

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological neurons systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections(weights), between elements, commonly neural networks are adjusted, or trained so that a particular input leads to as specific target output. The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used
in this supervised learning to train a network. Batch training of the network proceeds by making weight and bias changes based on an entire set (batch) of input vectors[6]. The batch training of the Multi-layer Feed-forward Neural network (MFFN) can be formulated as a non-linear unconstrained minimization problem [8, 9] Namely.

$$\min E(w_k),\ w_k \in R^n \qquad (1)$$

where E is the batch error measure defined as the sum of squared differences Error functions over the entire training set , defined by

$$E(w) = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{N_M}(o_{j,p}^{M} - t_{j,p})^2 \qquad (2)$$

Where $(o_{j,p}^{M} - t_{j,p})^2$ is the squared differences between the actual j-th output layer neuron for pattern P and the target output value. The scalar P is an index over input-output pairs, the general purpose of the training is to search an optimal set of connection weights in the manner that the error of the network output can be minimized.

The most popular training algorithm is the Classical Batch Back-Propagation (CBP) introduced by Rumelhart, Hinton and Williams[12]. Although the CBP algorithm is a simple learning algorithm for training Multi-layer Feed-Forward MFF networks, unfortunately it is not based on a sound theoretical basis and is very inefficient and unreliable. One iteration of the CBP algorithm can be written

$$w_{k+1} = w_k - \alpha_k g_k \qquad (3)$$

Where w_k is the vector of current weights and biases, $g_k = \nabla E(w_k)$ and α_k is the learning rate, with CBP the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate [5]. In order to overcome to the drawbacks of the CBP algorithm many gradient based training algorithms have been proposed in the literature [1, 2,5,7,13].

2. Some Modifications on CBP.

A surprising result was given by Brazilian and Brownie [3], which gives formula for the learning rate α_k and leads to super linear convergence. The main idea of Brazilia and Brownie (BB) method is to use the information in the previous iteration to decide the step size (learning rate) in the current iteration. The iteration in equation (3) is viewed as

$$w_{k+1} = w_k - D_k g_k \qquad (4)$$

Where $D_{k+1} = \alpha_k I$. In order to force matrix D_{k+1} having certain quasi-Newton(QN) property, is reasonable to require either

$$\min \left\| s_k - D_{k+1} y_k \right\|_2 \qquad (5)$$

Or

$$\min \left\| D_{k+1}^{-1} s_k - y_k \right\|_2 \qquad (6)$$

Where $s_k = w_{k+1} - w_k$ and $y_k = g_{k+1} - g_k$.

Solving equation (5) or (6) for α_k we get the following formulas

$$\alpha_k^{BB1} = \frac{s_k^T y_k}{y_k^T y_k} \qquad (7)$$

$$\alpha_k^{BB2} = \frac{s_k^T s_k}{s_k^T y_k} \qquad (8)$$

respectively. Note that we abbreviate the method defined in equation(3) with learning rate defined in equations (7) and (8) as BB1 and BB2 methods, respectively.

An alternative approach is based on the work of Plagianakos et al [11].
Following this approach, equation (3) is reformulated to the following Scheme:

$$w_{k+1} = w_k - B g_k \quad (9)$$

Where $B = diag[\lambda_1, \lambda_2, ..., \lambda_n]$ and λ_i , $i = 1, ..., n$ are eigen values for the $\nabla^2 E(w_k)$, or approximations to the Eigen-values for $\nabla^2 E(w_k)$.

A well known difficulty to this approach is that the computation of the Eigen values or estimating them is not a simple task, hence the schema defined in equation (9) is not practical .

3- Development Method

In the following we suggest another procedure for computing a scalar approximation of the Hessian of the function E at $w_k \in R^n$ which can be used to get the step-size along the negative gradient. Let us consider the initial point w_0 where $E(w_0)$ and $g_0 = \nabla E(w_0)$ can immediately be computed. Using the backtracking procedure (initialized with $\alpha_0 = 1$) we can compute the step-length

$$\alpha_k = \arg\min_{\alpha > 0} E(w_k - \alpha g_k)$$

So, the first step is computed using the backtracking along the negative gradient. Now, at point $w_{k+1} = w_k - \alpha_k g_k$, $k = 0,1,...$, from Taylor series we have

$$E_{k+1} = E_k - \alpha_k g_k^T g_k + \frac{1}{2}\alpha_k^2 g_k^T \nabla^2 f(z) g_k , \quad (10)$$

Where z is on the line segment connecting w_k and w_{k+1}. Having in view the local character of the searching procedure and that the distance between w_k and w_{k+1} is enough small we can choose $z = w_{k+1}$ and consider α_{k+1} as a scalar approximation of the $\nabla^2 E(w_{k+1})$, where $\alpha_{k+1} \in R$. This is an anticipative view point, in which a scalar approximation to the Hessian at point w_{k+1} is computed using only the local information from two successive points: w_k and w_{k+1}. There for, we can write see [4]:

$$\alpha_{k+1} = \frac{2}{g_k^T g_k \alpha_k^2}\left[E_{k+1} - E_k + \alpha_k g_k^T g_k\right] \quad (11)$$

Observe that at w_{k+1}, $k = 0,1,...$ we know $E(w_{k+1})$, g_{k+1} and approximation of $\nabla^2 E(w_{k+1})$ as $\alpha_{k+1} I$ Now, in order to compute the next estimation $w_{k+2} = w_{k+1} - \alpha_{k+1} g_{k+1}$ we must consider a procedure to step length α_{k+1}. For this let us consider the function:

$$\phi_{k+1}(\alpha) = E(w_{k+1}) - \alpha g_k^T g_k + \frac{1}{2}\alpha^2 g_k^T g_k \quad (12)$$

Observe that $\phi_{k+1}(0) = E(w_{k+1})$, $\phi_{k+1}(\frac{2}{\alpha_{k+1}}) = E(w_{k+1})$ and $\phi'(0) = -g_k^T g_k < 0$. Therefore, $\phi_{k+1}(\alpha)$ is a convex function for all $\alpha \geq 0$. To have a minimum for $\phi_{k+1}(\alpha)$ we must have $\alpha_{k+1} > 0$. Considering for moment $\alpha_{k+1} > 0$, then from $\phi'_{k+1}(\alpha)$ we get

$$\bar{t} = \frac{1}{\alpha_{k+1}} \quad (13)$$

as the minimum point of $\phi_{k+1}(\alpha)$, observe that

$$\phi_{k+1}(\bar{t}) = E_{k+1} - \frac{1}{2\alpha_{k+1}}\|g_k\|^2 \quad (14)$$

Showing that, if $\alpha_{k+1} > 0$, then at every iteration the value of function E_{k+1} is reduced[4]. On the other hand, if happen that $\alpha_{k+1} < 0$, we may define the following correction on the value of α_{k+1}

$$\eta_k = \frac{2}{g_k^T g_k}\left[E_{k+1} - E_k + \alpha_k g_k^T g_k + \delta\right], \quad \delta > 0 \quad (15)$$

$$\alpha_{k+1} = \frac{2}{g_k^T g_k (\alpha_k + \eta_k)^2}\left[E_{k+1} - E_k + (\alpha_k + \eta_k) g_k^T g_k\right] \quad (16)$$

To avoid the above problem (equations 15 and 16), we suggest the following formula to compute the learning rate at each epoch

$$\alpha_1 = \frac{5\alpha_0\|g_1\|^2}{|E_1| + \alpha_0^2\|g_1\|^2} \qquad k = 1$$

$$\alpha_{k+1} = \frac{5\alpha_k\|g_k\|^2}{|E_{k+1} - E_k| + \alpha_k^2\|g_k\|^2} \qquad k \geq 1 \quad .. (17)$$

with the use of Backtraking strategy to achieve the Wolfe conditions.

Algorithm(FISBP).

Step1. Initialization: Select $w_1 \in R^n$, $\varepsilon > 0$ and $0 < \rho \leq \sigma < 1$. Compute $E(w_1)$ and $g_1 = \nabla E(w_1)$. Consider $d_1 = -g_1$ and set $k = 1$.

Step2.Test for continuation of iterations. IF $\|g_k\| < \varepsilon$,set $w^* = w_k$ and $E^* = E_k$, then stop. Else go to Step 3.

Step3. Learning rate computation. Compute α_k from (17) and test for the Wolfe line Search conditions and update the variables $w_{k+1} = w_k - \alpha_k g_k$. Compute E_{k+1}, g_{k+1}, $s_k = w_{k+1} - w_k$ and $y_k = g_{k+1} - g_k$.

Step4. Set k=k+1 and go to Step 2.

4. Experiments and Results:

A computer simulation has been developed to study the performance of the following algorithms.

1- GD: classical back-propagation algorithm.
2- GDA: Adaptive back-propagation algorithm taken from Matlab-Toolbox.
3- F1SBP: New suggested training algorithm.

The simulations have been carried out using MATLAB(7.6) the performance of the MSBP has been evaluated and compared with batch versions of the above algorithm. The algorithms were tested using the initial weights, initialized by the Nguyen – widrow method [10] and received the same sequence of input patterns . The weights of network are updated only after the entire set of patterns to be learned has been presented . For each of the test problems, a table summarizing the performance of the algorithms for simulations that reached solution is presented . The reported parameters are min the minimum number of epochs for 50 simulation , mean the mean value of epochs for 50 simulation, Max the maximum number of epochs for 50 simulation, Tav the average of total time for 50 simulation and Succ, the succeeded simulations out of (50) trails within error function evaluations limit. If an algorithm fails to converge within the above limit considered that it fails to train the FFNN, but its epochs are not included in the statical analysis of the algorithm, one gradient and one error function evaluations are necessary at each epoch.

1- Problem (XOR Problem)

The first problem we have been encountered with is the XOR Boolean function problem, which is considered as a classical problem for the FFNN training . The XOR function maps two binary inputs to a single binary output. As it is well known this function is not linearly separable. The network architectures for this binary classification problem consists of one hidden layer with 3 neurons and an output layer of one neuron. The termination criterion is set to $\varepsilon_2 \le 0.002$ within the limit of 1000 epochs, and table(1) summarizes the result of all algorithms i.e for 50 simulations the minimum epochs for each algorithm are listed in the first column (Min), the maximum epochs for each algorithm are listed in the second column, third column contains (Mean) the mean value of epochs and (Tav) is the average of time for 50 simulations and last columns contain the percentage of succeeds of the algorithms in 50 simulations.

Table (1): Results of simulations for the XOR function

algorithms	min	max	mean	Tav	succ
GD	230	2000	652.78	8.83028	92%
GDA	50	84	66.5	1.13614	100%
F1SBP	3	26	8.9	0.5456	100%

2- Function Approximation Problem.

The second problem we have considered is the approximation of continuous function, $f(x) = \cos(\pi x) + 0.1 * rand(\sin x)$

Where $x = -1:0.05:1$. This problem maps one real input to a single real output. The selected architecture of the FFNN is one neuron in input lager, ten neuron in hidden layer and one neuron in output neuron, with sigmoid function in hidden neuron's and a linear function in output neuron. The error goal has been let to 0.001 and the maximum epochs to 1000. The results of the simulations presented in table (2) and figure(2)

Table(2): Results of simulations for the Function Approximation Problem

algorithms	min	max	mean	Tav	succ
BP	--	--	--	--	--
GDA	191	795	391.94	5.60024	100%
F1SBP	107	662	285.54	8.29016	100%

3- SPECT Heart Problem.

This dataset contains data instances derived from cardiac single proton Emission Computed Tomography (SPECT) images from the University of Colorado. This is also a binary classification task, where patients heart images are classified as normal is abnormal. The class distribution has 55 instances of the abnormal class 20.6% and 212 instances of the normal class (79.4%)'From them there have been selected 80 instances for the training process and the remainder 187 for testing the neural networks generalization capability. The network architecture for this medical classification problem constitute of 1 hidden layer with 6 neurons and an output layer of 2 neurons. The termination criterion is set to $E_{rr} \le 0.1$ within the limit of 1000 epochs.

Table(3): Results of simulations for the SPECT Heart Problem

algorithms	min	max	mean	Tav	succ
GD	505	1827	989.9	13.14658	100%
GDA	70	525	190.82	2.71288	100%
F1SBP	25	87	55.52	2.1963	100%

5. Conclusions

In this paper we proposed a new formula for computing learning rate in the back-propagation algorithm for training feed-forward multi-layer neural networks. Based on our numerical experiments, we concluded that our proposed method outperforms classical Back-propagation and adaptive Back-propagation training algorithms and has a potential to significantly enhance the computational efficiency and robustness of training process.

References

[1] Abbo K. and Hind M.(2012) 'Improving the learning rate of the Back - propagation Algorithm Aitken process'. Iraqi Journal of the statistical sciences, accepted (to appear)
[2] Abbo K. and Zena T.(2012) 'Minimization algorithm for training feed-forward neural Networks'. J. of Education and Sci. (to appear).
[3] Brazilia J and Brownie M.(1988)'Tow point step-size gradient methods'. SIMA. Journal of Numerical Analysis, 8.
[4] Andrei, N,(2005) A New Gradient Descent Method with Anticipative Scalar Approximation of Hessian for Unconstrained Optimization, Scrieri Matematice1, Romania.
[5] Gong L., Liu G., Li Y. and Yuan F. (2012) 'Training Feed- forward Neural Networks Using the gradient descent method with optimal Step size, J. of Computational Information Systems 8:4.
[6] Hertz J., Krogh A .and Palmer R .(1991) 'Introduction to the theory of Neural computation'. Addison-Wesley, Reading , MA .
[7] Jacobs R (1988) 'Increased rates of convergence through learning rate adaptation' .Neural Networks , vol. 1, no.4.
[8] Kostopoulos A. Sotiropoulos D. and Grapsa T. (2004). "A new efficient learning rate for Perry's spectral conjugate gradient Training method",1st International Conference. From Scientific Computing to Computational Engineering'. 1st IC-SCCE. Greece.
[9] Livieris I. and Pintelas R. (2011). "An advanced conjugate gradient training algorithm based on a modified secant equation", Technical Report NO.TR11-03. University of Patras Department of Mathematics, Patras, Greece.
[10] Nguyen D. and Widrow B. (1990). "Improving the learning speed of 2-layer neural network by choosing initial values of the adaptive weights", Biological Cybernetics, 59:
[11] Plagianakos V., Magoulas G., and Vrahatis M. (2002) ' Determing non-monotone strategies for effective training of multi-layer perceptrons'. IEEE Transactions on Neural Networks, 13(6).
[12] Rumelhart D., Hinton G. and Williams R (1986) 'Learning representations by back-propagation errors' Nature,32
[13] Sotirpoulos D., Kotsiopoulos A and Grapsa T.(2004) 'training neural networks using two point step-size gradient methods'. International conference of numerical Analysis andApplied Mathematics. Patras, Greece.

معدل تعليم جديد باعتماد على طريقة Andrei في تدريب الشبكات العصبية الاصطناعية ذوات التغذية الامامية

خليل خضر عبو[1] ، حسن حسين ابراهيم[2] ، فردوس علي ابراهيم[3]

[1]قسم الرياضيات ، كلية علوم الحاسوب والرياضيات ، جامعة الموصل ، الموصل ، العراق
[2]قسم الرياضيات ، كلية علوم الحاسوب والرياضيات ، جامعة تكريت ، تكريت ، العراق
[3]قسم الرياضيات ، كلية التربية ، جامعة تكريت ، تكريت ، العراق

الملخص

تم في هذا البحث تطوير خوارزمية جديدة لحساب عامل التعليم للشبكات العصبية ذوات التغذية الامامية. تعتمد الفكرة على تطوير طريقة Andrei وذلك باستخدام تقريب جديد لمصفوفة هيسي لدالة الخطأ. وقد بين النتائج العددية ان الطريقة المقترحة تتقارب الى الحل بشكل اسرع من الطرق التي ثم مقارنتها معها.