Advanced Neural Network Learning Applied To Breast Cancer Detection

Zarita Zainuddin
School of Mathematical Sciences
Universiti Sains Malaysia, 11800 Minden, Malaysia
zarita@cs.usm.my

Computer Aided Diagnostics (CAD) in the form of neural networks (utilized for pattern recognition and classification) offer significant potential to provide an accurate and early automated diagnostic technology. This paper considers neural network training to detect breast cancer by analyzing the fine needle aspirates (FNA) of a breast mass. For enhanced learning, three gradient-based multi layer perceptron (MLP) training methods originated from optimization theory, namely, steepest-descent gradient search, conjugate-gradient and Levenberg Marquardt are considered. In addition, two newly proposed methods, the Dynamic Momentum Factor and Dynamic Learning Rate are discussed. The results presented show that significant improvements in convergence performance can be obtained through the integration of these acceleration methods whilst preserving the generalization capability of the networks.

INTRODUCTION

Breast Cancer is second only to lung cancer as a tumor-related cause of death in women. More that 180,000 new cases are reported annually in the US alone. Furthermore, the American Cancer Society estimates that at least 25% of these deaths could be prevented if all women in the appropriate age groups were regularly screened.

Although there exists reasonable agreement on the criteria for benign/malignant diagnoses using fine needle aspirate (FNA) and mammogram data, the application of these criteria are often quite subjective. Additionally, proper evaluation of FNA and mammogram sensor data is a time consuming task for the physician. Intra-and-inter-observer disagreement and/or inconsistencies in the FNA and mammogram interpretation further exacerbate the problem.

Consequently, C*omputer Aided Diagnostics* (CAD) in the form of neural networks (utilized for pattern recognition and classification) offer significant potential to provide an accurate and early automated diagnostic technology. This automated technology may well be useful in further assisting with other problems resulting from physical fatigue, poor mammogram image quality, inconsistent FNA discriminator numerical assignments, as well as other possible sensor interpretation problems.

Some practical results of CAD of breast cancer sensor data using neural networks are expected to be:

- Operational software which will aid the physician in making the diagnosis, quite possibly in real time, and once formulated and tested, they are always consistent, not prone to human fatigue or bias.
- Providing diagnostic assistance for the intra-and-inter-observability problems by ultimately minimizing the subjective component of the diagnostic process
- Providing an initial detection and/or classification process in the absence of a qualified physician
- Providing possible (and probably currently unknown) relationships between sensor environment discriminators and a correct diagnosis.

The efficient supervised training of neural networks (NNs) is a subject of considerable ongoing research and numerous algorithms have been proposed to this end. The backpropagation algorithm (BPA) [1] is one of the most common supervised training methods. It uses the gradient or steepest descent method to reduce

the error function where the weights are adjusted by the algorithm so as to make the error decreases along a descent direction. In doing so, the two parameters, learning rate (LR) and momentum factor (MF) are used to control the size of weight adjustment along the descent direction and for dampening oscillations of the iterations. In the conventional backpropagation algorithm (BPA), these two parameters are empirically chosen. In general, the MF should be less than unity to stabilize the BPA. When error oscillations happen, an MF close to unity is needed to smooth the error oscillations. As for the selection of the LR, it is more arbitrary due to the fact that the error surface usually consists of many flat and steep regions and behaves quite differently from application to application. A large LR is helpful to acceleration of the learning when the weight search crosses a plateau. Nevertheless, in the meanwhile, it increases the possibility that the weight search jumps over steep regions and moves into undesirable regions. When this happens, failure of the backpropagation learning may be caused. Therefore, an efficient BPA should be capable of dynamically varying its LR and MF in accordance with the regions the weight adjustment lies in. Research into the dynamic change of the LR and MF parameters has been carried out extensively by numerous authors including Becker & leCun [2], Battiti [3] and Yu et al. [4].

In this contribution, the performance of the Dynamic Momentum Factor (DMF) [5] and Dynamic Learning Rate (DLR) [6, 7] algorithms are evaluated and compared against the conventional BP and three other gradient based optimization methods - the steepest descent, conjugate gradient and Levenberg Marquardt methods [8] on the breast cancer detection problem using continuous-valued training data. This is accomplished by training a 30-6-4-2 MLP consisting of 30 input nodes, 6 first hidden layer nodes, 4 second hidden layer nodes and 2 output nodes. After being trained, the networks are tested on generalization capabilities on a testing set consisting of images outside the training set. The capabilities of the networks should at least be similar to the conventional BP trained network although lesser function evaluations are necessary to converge.

DESCRIPTION OF THE DATA SET

The data set used was obtained from the University of Wisconsin Breast Cancer problem [9]. Features were computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. Ten real-valued features were computed for each cell nucleus: (a) radius (b) texture (c) perimeter (d) area (e) smoothness (f) compactness (g) concavity (h) concave points (i) symmetry (j) fractal dimension.

The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features, from which one attribute, malignant or benign, must be detected.

TRAINING METHODS

In Evans [10], the reasons for the slow convergence of the backpropagation have been discussed. To date, many techniques have been proposed to deal with the inherent problems of backpropagation. These techniques can be divided roughly into two main categories; those referred to as global techniques that use global knowledge of the state of the entire network, such as the direction of the overall weight update vector. Most of these techniques have their roots in the well-explained domain of optimization theory. The simplest is a first-order method that uses the steepest-descent (SD) direction [1]. An alternative is the conjugate gradient (CG) method, which modifies the SD direction by conjugating it with the previously used direction [11]. Finally, the Levenberg – Marquardt (LM) method is a second – order method that approximates the second derivative using the first-order gradient [12].

In contrast, local adaptation strategies are based on weight specific information only, such as the temporal behavior of the partial derivative of the current weight. Two local adaptive learning rules are presented here, namely, the Dynamic Momentum Factor (DMF) [5] and Dynamic Learning Rate (DLR) [6].

Steepest Descent Method

The first method proposed by Rumelhart and McClelland [1] for training NNs is the SD method. The value of the weight update is calculated as follows:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\mathbf{p}(n) \quad (1)$$

$$\mathbf{p}(n) = -\frac{\partial E(n)}{\partial w(n)} + \alpha(n)\mathbf{p}(n-1) \quad (2)$$

where n is the iteration count, η is the step width (learrning rate), μ is the momentum factor, and $\mathbf{p}(n)$ is the step direction taken in the nth iteration step.

Conjugate Gradient Method

In optimization theory, the CG method has been known since Fletcher and Reeves [13]. Leonard and Kramer [11] introduced the original Fletcher – Reeves algorithm in the field of NN research. The method is, in some way, an extension to SD, introducing a formula for determining the momentum factor α in Eq. (2):

$$\alpha(n) = \frac{g^T(n+1)g(n+1)}{g^T(n)g(n)} \tag{3}$$

Levenberg Marquardt Method

The third commonly used minimization method is LM. It is directly applicable only when the error measure is a sum of squared errors:

$$E = \sum_{N,C}(y-d)^2 \tag{4}$$

Then, an approximate Hessian $\mathbf{H}$ can be written as

$$\frac{\partial^2 E(n)}{\partial w(n)^2} \approx \mathbf{H} = \left(\frac{\partial E}{\partial w}\right)^T \frac{\partial E}{\partial w} + \mu\mathbf{I} \tag{5}$$

where $\mathbf{I}$ is the identity matrix of dimension v and μ is a small scalar; this increment makes the matrix $\mathbf{H}$ to be invertible. The calculation of the weight update is then based on Eq. (1) and Eq. (2), using a square matrix $\mathbf{R}$, instead of the scalar η to premultiply the search – direction vector $\mathbf{p}$ with:

$$\mathbf{R} = \mathbf{H}^{-1} \tag{6}$$

and setting the momentum α in Eq. (2) to zero.

Dynamic Momentum

A momentum factor update rule, the basis of the Dynamic Momentum Factor (DMF) algorithm, which dynamically adapts the momentum factor with respect to the iteration number is given below. A complete analysis regarding momentum factor and a derivation of this rule has been presented in Evans and Zainuddin [5].

Momentum Constant Update Rule

Let $\Delta\alpha_{ji}(n,0)$ denote the positive adjustment applied at iteration n to the momentum constant at iteration 0, α_{ji} (0,0). We define $\Delta\alpha_{ji}(n,0)$ as

$$\Delta\alpha_{ji}(n,0) = \gamma_a^b + \alpha_{ji}(0,0) \tag{7}$$

for all $n \in [a,b]$ where $0 \le \gamma_a^b \le 1-\alpha_{ji}(0,0)$ and $\gamma_a^b > \gamma_c^d$ for a>c and b>d.

The constraint $0 \le \gamma_a^b \le 1-\alpha_{ji}(0,0)$ is imposed to ensure that $0<\alpha_{ji}(n)\le 1$ as the momentum constant α has to be in the range $0 \le |\alpha| \le 1$ to ensure convergence of the learning algorithm. The initial value of α, $\alpha_{ji}(0,0)$ can be chosen to be any small value in the interval [0,1]. Note that without loss of generality, we define $\Delta\alpha_{ji}(n,0)$ as a positive adjustment. If α is negative, then we will consider a negative adjustment but it is unlikely that a negative α would be used in practice.

The iteration number domain is partitioned into n intervals and a suitable value for the momentum constant is assigned for each respective interval. As n gets large, the momentum constant is incremented gradually making sure that $\alpha_{ji}(n)$ is less that or equals to 1. This method only requires n comparisons, where n is the number of iterations and no storage requirement is demanded at all.

Dynamic Adaptation of the Learning Rate

In Evans et al [10], we see that the convergence rate is crucially dependent on the optimal choice of the learning rate parameter. It is necessary to find a method that allows the parameters to be adjusted in the course of the learning procedure.

Presently, there exists many acceleration methods to overcome the slow convergence problem. There are methods that exploit the information contained in the second derivative of the cost function while others do not use higher-order derivatives [14, 15].

Below we present the learning rate update rule, which forms the basis of the Dynamic Learning Rate (DLR) method which dynamically adapts the learning rate parameter with respect to the magnitude of the partial derivative of the error surface with respect to the current weight, $w_{ji}(n)$, $\partial 1(n)/\partial w_{ji}(n)$. A derivation of the rule can be found in Zainuddin and Evans [6].

Learning Rate Update Rule

Let $\Delta\eta_{ji}(n)$ denote the adjustment applied at iteration n to the learning rate parameter at iteration 0, $\eta_{ji}(0)$. We define $\Delta\eta_{ji}(n)$ as

$$\Delta\eta_{ji}(n) = \lambda_{\delta b}^{\delta a} + \eta_{ji}(0) \tag{8}$$

for all $\delta = \left|\frac{\partial \xi(n)}{\partial w_{ji}(n)}\right| \in (\delta a, \delta b)$ and $\lambda_{\delta b}^{\delta a} < \lambda_{\delta d}^{\delta c}$ for $\delta a > \delta c$ and $\delta b > \delta d$.

In this learning rate adaptation method, the partial derivative domain is partitioned into n intervals (not necessarily of equal size) and a suitable value for the learning rate parameter is assigned for each respective interval.

If $\left|\frac{\partial \xi(n)}{\partial w_{ji}(n)}\right| \in (\delta a, \delta b)$ where δa and δb are small positive values, then $\lambda_{\delta b}^{\delta a}$ is large. On the other hand, if δa and δb are moderate, then $\lambda_{\delta b}^{\delta a}$ is moderate and if δa and δb are large, then $\lambda_{\delta b}^{\delta a}$ is small. The values of $\lambda_{\delta b}^{\delta a}$ are problem dependent. The values of the learning rate for each interval are assigned at the beginning of the learning procedure and they are kept fixed for the whole training process. The learning rate for each connection weight is adapted by determining which interval the gradient belongs to.

SIMULATIONS ON THE BREAST CANCER DETECTION PROBLEM

All the data set inputs have been scaled to the range –1 to 1 for the experiment. A 30-6-4-2 multi layer perceptron was used where the output nodes correspond to the 2 classification classes. The training set consists of 100 vector pairs while the testing set consists of 50 vector pairs. We have chosen the batch mode weight updating because results by other researchers [16], [17] suggest that in tasks where generalization is important, the pattern mode should be avoided, despite their faster training times. The weights and threshold values were initialized to values drawn at random with a uniform distribution between –1 and 1. The learning process was terminated when the sum of the square of the error reached $1*10^{-3}$.

The value of the learning rate was η = 2 and the momentum factor was chosen to be α = 0.9 in the first simulation (BP batch) while the DMF method was used for the second simulation. The partition of the iteration number domain and the α values for each interval is shown is Table 1. For the DLR method, the partition of the gradient domain and their respective η values chosen for each interval are shown in Table 2. Subsequently, the DLR, CG, LM and SD methods were employed. Table 3 shows the results of the simulations discussed above, which are the average of 10 trials. It can be observed that the DMF and DLR methods improved the convergence profoundly. A speedup of 97.08 % was obtained for the DMF method while a speedup of up to 97.27 % was obtained for the DLR method. The SD method gave 6 instances of no convergence. Although the CG method is able to provide a good convergence rate (99.19 %), it nevetheless, requires much more complexity and computation

TABLE 1. The chosen values of the momentum factor $\alpha(n)$ for the breast cancer detection problem using the Momentum Factor Update Rule.

Iteration number (n)	Value of $\alpha(n)$
$1 \le n < 100$	0.5
$100 \le n < 200$	0.6
$200 \le n < 300$	0.7
$300 \le n < 400$	0.8
$400 \le n < 500$	0.9
$500 \le n$	0.95

TABLE 2. The chosen values of η for the breast cancer detection problem using the Learning Rate Update Rule.

Gradient $\lvert\partial\xi/\partial w\rvert$	η
$10^{-2} \le \delta$	5
$5x10^{-3} \le \delta < 10^{-2}$	10
$10^{-3} \le \delta \le 5x10^{-3}$	15
$5x10^{-4} \le \delta < 10^{-3}$	20
$10^{-4} \le \delta \le 5x10^{-4}$	30
$10^{-5} \le \delta < 10^{-4}$	40
$10^{-6} \le \delta < 10^{-5}$	80
$\delta < 10^{-6}$	160

TABLE 3:The simulation results for the classification of iris plant using Batch Mode BP, DMF, DLR methods, Conjugate Gradient ,Steepest Descent and Levenberg – Marquardt methods.

per iteration than the other methods. The LM gave the best performance (99.90%). However, it involves a large number of computations and demands a huge storage requirement since it must store the approximate Hessian matrix. As for the DLR method, it was found that the η values change considerably during the learning process, providing the best progress in the reduction of the error function.

GENERALIZATION CAPABILITY OF THE MLP

Generalization is the ability of the network to respond to inputs it has not seen before and a network is said to generalize well when the output of the network is correct for input patterns that are never used in training the network. After being trained with the batch BP, DMF, DLR, CG, LM and SD methods respectively, the generalization capability of the MLPs on new vector pairs was tested.

The testing set for the breast cancer detection problem consists of 50 vector pairs. It should be emphasized here that these vector pairs were never used in training the network.

TABLE 4. Recognition rates of input patterns in the testing set.

Algorithm	Recognition Rate (%)
Batch BP	96
Dynamic MF	96
Dynamic LR	96
Conjugate Gradient	96
Levenberg-Marquardt	96
Steepest Descent	96

Table 4 shows the recognition rates of the MLP for input patterns in the testing sets. As can be seen, the DMF and DLR methods had similar generalization capability with the batch BP although lesser function evaluations were necessary to converge. It is important to note here that both the DMF and DLR methods demonstrated similar generalization capabilities when compared to the CG ,LM and SD methods.

The MLPs identified and categorized perfectly the input patterns on which they were trained. This is expected since the sum of squared errors for the Breast Cancer Detection problem, valued at $1*10^{-3}$ is a very small number. The recognition rate is as high as 96 % and similar errors occurred for the networks trained with the 5 different algorithms where these input patterns do not have many features in common with the input patterns used in the training set.

CONCLUSION

The acceleration methods namely, Dynamic Momentum Factor (DMF) and Dynamic Learning Rate (DLR), have proven to be very effective and superior in terms of convergence when tested and compared with the Batch BP on the Breast Cancer Detection problem. A speed up of up to 97.08 % and 97.27 % was obtained for the DMF and DLR methods respectively.

The Dynamic Momentum Factor method assigns an optimal value to the momentum factor for each individual weight at each iteration and this greatly enhanced the convergence rate. The main advantage of the DMF method is that the momentum factor is allowed to vary with time in the course of the learning. This in effect, stabilizes the network at the beginning of the learning process and accelerates the learning when the network is stable.

As for the Dynamic Learning Rate method, it was found that the η values changed considerably during the learning process, providing the best progress in the reduction of the error function. The conjugate gradient method has a much faster convergence rate than the other methods since it uses second order information to calculate the new direction, hence it entails more complexity and computation. The Levenberg – Marquardt method gave the best performance but it is suitable only for moderate numbers of network parameters since it involves a large number of computations and requires a huge storage requirement.

In terms of generalization capability, both the DMF and DLR showed similar generalization capability to the batch BP although lesser function evaluations are necessary to converge. In other words, the capability of the networks to recognize input patterns outside the training set is not impaired by the employment of these acceleration methods. Hence, these algorithms are promising in practical applications where generalization is important.

REFERENCES

[1] Rumelhart D.E. and McClelland J.L., eds., 1986, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Cambridge, MA: MIT Press.

[2] Becker, S. & Le Cun, Y., 1988, Improving the convergence of back-propagation learning with second order methods. Technical Report CRG-TR-88-5, University of Toronto, Toronto, Canada.

[3] Battiti R., 1992, First and second order methods for learning: between steepest descent and Newton's method. Neural Computation, 4, 141-166.

[4] Yu, X.H., Chen, G.A. & Cheng, S.X., 1995, Dynamic learning rate optimization of the backpropagation algorithm. IEEE Transaction on Neural Networks, 6(3), 669-677.

[5] Evans D.J & Zainuddin Z., 1997, Acceleration of the backpropagation through dynamic adaptation of the momentum. Neural, Parallel & Scientific Computations, 5(3), 297-308. (see also Internal Report

No. 1028, PARC, Loughborough University of Tech., U.K. 1996).
[6] Zainuddin Z. & Evans D.J., 1997, Acceleration of the Back Propagation through dynamic adaptation of the learning rate, International Journal of Computer Mathematics, 334, 1-17. (see also Internal Report No. 1029, PARC, Loughborough University of Tech., U.K. 1996).
[7] Zainuddin, Z. & Evans, D.J., Human Face Recognition using Accelerated Multilayer Perceptrons, Int. J. of Computer Mathematics, 80 (2/3), 2003.
[8] Press, W.H., Flannery, B.P., Teukolsky, S.A., & Vetterling, W.T., 1986., "Numerical recipes : the art of scientific computing". Cambridge : Cambridge University Press.
[9] W. H. Wolberg, W. N. Street, and O. L. Mangasarian, 1995, Wisconsin Diagnostic Breast Cancer, University of Wisconsin, Computer Science Department:
http://www.cs.wisc.edu/~olvi/uwmp/mpml.html
[10] Evans, D.J. Ahmad Fadzil M.H. & Zainuddin, Z., 1997, Accelerating backpropagation in human face recognition, Proc. IEEE Int. Conf. on Neural Networks, IEEE Press, 1347-1352
[11] Leonard, J., & Kramer, M.A., 1990, Improvement to the back-propagation algorithm for training neural networks. Computers and Chemical Engineering, 14(3), 337-341.
[12] Chen, S., Billings, S.A., & Grant. P.M., 1990, Non-linear system identification using neural networks. International Journal of Control, 51(6), 1191-1215.
[13] Fletcher, R., & Reeves, C.M., 1964, Function minimization by conjugate gradients, Computer Journal, 7, 149-154.
[14] Van Der Smagt, P.P. & Krose, B.J.A., 1991, Acceleration of the backpropagation through dynamic adaptation of the learning rate, International Conf. on Artificial Neural Networks, p. 351-356, Espoo, Finland: Elsevier Science Publishers.
[15] Salomon R., 1996, Accelerating backpropagation through dynamic self-adaptation. Neural Networks, 9 (4), 589-602.
[16] Alpsan, D., Towsey, M., Ozdamar, O., Tsoi, A. & Ghista, D.N., 1994, Determining hearing threshold from brain stem evoked potentials. In IEEE Engineering in Medicine and Biology, 465-471.
[17] Cohn, D. & Tesauro, G., 1994, How tight are the Vapnik-Chervonenkis Bounds? Neural Computation, 4, 249-269.