AN IMPROVEMENT TO THE NATURAL GRADIENT LEARNING ALGORITHM FOR MULTILAYER PERCEPTRONS

Michael R. Bastian, Jacob H. Gunther and Todd K. Moon

Utah State University
Department of Electrical and Computer Engineering
4120 Old Main Hill, Logan, UT 84322-4120

ABSTRACT

Natural gradient learning has been shown to avoid singularities in the parameter space of multilayer perceptrons. However, it requires a large number of additional parameters beyond ordinary backpropagation. This article describes a new approach to natural gradient learning in which the number of parameters necessary is much smaller than the natural gradient algorithm. This new method exploits the algebraic structure of the parameter space to reduce the space and time complexity of the algorithm and improve its performance.

1. INTRODUCTION

Amari and his colleagues have developed natural gradient learning for multilayer perceptrons [1, 2, 3], which instead of the steepest descent direction, uses a Quasi-Newton method [4] that exploits the Riemannian metric tensor of the underlying parameter space as the approximation to the Hessian. In the case of multilayer perceptrons, this metric tensor is the Fisher Information matrix evaluated for the current parameter. Since the Fisher Information matrix is the expected value of the Hessian matrix, it fits very nicely into a Quasi-Newton optimization framework.

However, the problem with natural gradient learning is that the Fisher Information matrix must be inverted. Also, for large networks, the algorithm becomes computationally intractable because of the large number of additional parameters in the Fisher Information matrix that is required during training. This problem can be mitigated by a new formulation of the Fisher Information matrix for multilayer perceptrons.

This article describes this new formulation. By picking an inner product for the parameter space and inducing a norm, a *Sobolev Gradient* [5] may be formulated such that the Fisher Information matrix has much smaller dimensions than in the Adaptive Natural Gradient algorithm [2] and, as a result, the learning algorithm performs better is more robust to initial values and noise.

Thanks to Anteon Corporation for funding this research.

2. A NEW PARAMETERIZATION

Let the random variable $\mathbf{X}$ be the feature map of a multilayer perceptron θ. Let the function $F(\theta, \mathbf{X})$ be the output of the perceptron when given $\mathbf{X}$ as the input. Let the random variable $\mathbf{Y}$ be the desired output of the perceptron and the random variable

$$\mathbf{Z} = \mathbf{Y} - F(\theta, \mathbf{X}) \tag{1}$$

be the output error of the perceptron. Let θ^* be the optimal perceptron such that $E[\frac{1}{2}\|\mathbf{Z}\|^2]$ is minimized.

2.1. Mapping to an Anti-Diagonal Block-Matrix

A multilayer perceptron with L layers is parameterized by L matrices $W_1, W_2, \ldots, W_L$; each matrix represents the connection weights of the layer.

Let H be a mapping of the L matrices into a block matrix such that $W_1, W_2, \ldots, W_L$ lie on the anti-diagonal of the block matrix.

$$H(W_1, W_2, \ldots, W_L) = \begin{bmatrix} 0 & \cdots & 0 & W_1 \\ 0 & \cdots & W_2 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ W_L & \cdots & 0 & 0 \end{bmatrix}. \tag{2}$$

This structure is similar in form to the weighted adjacency matrix of an L-partite graph [6].

2.2. The Inner Product of Two Perceptrons

An inner product of two perceptrons $\theta = H(W_1, W_2, \ldots, W_L)$ and $\zeta = H(V_1, V_2, \ldots, V_L)$ is

$$\langle \theta, \zeta \rangle = \sum_{i=1}^{L} \operatorname{tr}(W_i V_i^T). \tag{3}$$

2.3. The Induced Norm of a Perceptron

Similarly, the induced norm of this perceptron inner product for $\theta = H(W_1, W_2, \ldots, W_L)$ is

$$\|\theta\| = \sqrt{\sum_{i=1}^{L} \mathrm{tr}(W_i W_i^T)}. \quad (4)$$

This norm is similar to the Frobenius norm of matrices [7].

It is now possible to write the Sobolev gradient of a multilayer perceptron.

3. SOBOLEV GRADIENT

Let $S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ be n obervations of the random tuple $(\mathbf{X}, \mathbf{Y})$. To simplify presentation, assume that there are only two layers in the percptron ($L = 2$). Then the norm of the error of training example i will be

$$\phi(x_i, y_i, \theta) = \|y_i - W_2 \psi(W_1 x_i)\|^2 \quad (5)$$

where ψ is the activation function of the hidden layer.

3.1. The Directional Derivative of ϕ

The directional derivative of ϕ with respect to θ in the direction of ζ is

$$\phi'(x_i, y_i, \theta)\zeta = \lim_{t \to 0} \frac{\phi(x_i, y_i, \theta + t\zeta) - \phi(x_i, y_i, \theta)}{t}. \quad (6)$$

The Sobolev gradient of ϕ with respect to θ is the perceptron $(\nabla\phi)(x_i, y_i, \theta)$ such that for any perceptron ζ

$$\phi'(x_i, y_i, \theta)\zeta = \langle \zeta, (\nabla\phi)(x_i, y_i, \theta) \rangle. \quad (7)$$

3.2. The Sobolev Gradient of a Two-Layer Perceptron

Let $\theta = \left[\begin{smallmatrix} 0 & W_1 \\ W_2 & 0 \end{smallmatrix}\right]$ and $\zeta = \left[\begin{smallmatrix} 0 & V_1 \\ V_2 & 0 \end{smallmatrix}\right]$, then the directional derivative is

$$\begin{aligned} \phi'(x_i, y_i, \theta)\zeta &= \lim_{t \to 0} \frac{\phi(x_i, y_i, \theta + t\zeta) - \phi(x_i, y_i, \theta)}{t} \\ &= \lim_{t \to 0} \frac{1}{2t} \|y_i - (W_2 + tV_2)\Psi((W_1 + tV_1)x_i)\|^2 \\ &\quad - \lim_{t \to 0} \frac{1}{2t} \|y_i - W_2 \Psi(W_1 x_i)\|^2. \end{aligned}$$

Substituting the first order Taylor series,

$$\Psi((W_1 + tV_1)x_i) = \Psi(W_1 x_i) + t\Psi'(W_1 x_i)V_1 x_i$$

and letting $u_i = \Psi(W_1 x_i)$, $z_i = y_i - W_2 u_i$, and $b_i = [\Psi'(W_1 x_i)]^T W_2^T z_i$,

$$\begin{aligned} \phi'(x_i, y_i, \theta)\zeta &= -\langle W_2 \Psi'(W_1 x_i) V_1 x_i + V_2 u_i, z_i \rangle \\ &= \langle -W_2 \Psi'(W_1 x_i) V_1 x_i, z_i \rangle + \langle -V_2 u_i, z_i \rangle \\ &= \left\langle \begin{bmatrix} O & V_1 \\ V_2 & O \end{bmatrix}, \begin{bmatrix} O & -b_i x_i^T \\ -z_i u_i^T & O \end{bmatrix} \right\rangle. \end{aligned}$$

By arranging the product to be the inner product of two perceptrons the Sobolev gradient follows as

$$(\nabla\phi)(x_i, y_i, \theta) = \begin{bmatrix} 0 & -b_i x_i^T \\ -z_i u_i^T & 0 \end{bmatrix}, \quad (8)$$

where $b_i = [\psi'(W_1 x_i)]^T W_2^T z_i$ is the backpropagation error and $u_i = \psi(W_1 x_i)$ is the activation of the hidden units.

3.3. Perceptrons With More Than One Layer

Extending this procedure to perceptrons where $L > 2$ is done by taking the outer product of the input to a layer with its backpropagation error.

4. FISHER INFORMATION MATRIX

The Fisher Information Matrix is

$$\begin{aligned} G(\theta) &= E[(\nabla\phi)(\mathbf{X}, \mathbf{Y}, \theta)(\nabla\phi)^T(\mathbf{X}, \mathbf{Y}, \theta)] \\ &= \begin{bmatrix} E\{\|\mathbf{X}\|^2 \mathbf{B}\mathbf{B}^T\} & 0 \\ 0 & E\{\|\mathbf{U}\|^2 \mathbf{Z}\mathbf{Z}^T\} \end{bmatrix} \end{aligned} \quad (9)$$

where $\mathbf{B} = [\psi'(W_1 \mathbf{X})]^T W_2^T \mathbf{Z}$ is the backpropagation error and $\mathbf{U} = \psi(W_1 \mathbf{X})$ is the hidden unit activation. (The elements of this block-diagonal matrix are similar to the fourth-order cumulants used in Blind Source Separation [8].)

The fisher information matrix is used in the update of the perceptron weights as the Riemannian metric tensor [9].

$$\theta_{k+1} = \theta_k - \eta_k G^{-1}(\theta_k)(\nabla\phi)(\theta_k). \quad (10)$$

These can be simplified because of the diagonal block structure of $G(\theta)$.

$$W_1^{(k+1)} = W_1^{(k)} + \eta_k \tilde{G}_1^{(k)} b_k x_k^T \quad (11)$$

$$W_2^{(k+1)} = W_2^{(k)} + \eta_k \tilde{G}_2^{(k)} z_k u_k^T \quad (12)$$

where η_k is the learning rate for the connection weights.

4.1. Recursive Estimation

A recursive estimator can be used for each block of the Fisher Information Matrix and the matrix inversion lemma [10] applied to each block so that the update equations are:

$$\tilde{G}_1^{(k+1)} = (1 + \epsilon_k)\tilde{G}_1^{(k)} - \epsilon_k \|x_k\|^2 \tilde{G}_1^{(k)} b_k b_k^T \tilde{G}_1^{(k)} \quad (13)$$

$$\tilde{G}_2^{(k+1)} = (1 + \epsilon_k)\tilde{G}_2^{(k)} - \epsilon_k \|u_k\|^2 \tilde{G}_2^{(k)} z_k z_k^T \tilde{G}_1^{(k)} \quad (14)$$

where $\tilde{G}_{1,2}$ are the estimators of the inverse blocks and ϵ_k is the learning rate for the Fisher Information Matrix [2].

4.2. Reduced Complexity

If m is the number of outputs, p the number of inputs and r the number of hidden units, then the top-left block of the Fisher Information Matrix is an $r \times r$ matrix and the bottom-right block is an $m \times m$ matrix. With previous natural gradient algorithms, the Fisher Information is an $r(m+p) \times r(m+p)$ matrix. So the complexity of each step of this algorithm is $O(m^2 + r^2)$ where the Adaptive Natural Gradient method has a complexity of $O(r^2(m+p)^2)$.

5. EXPERIMENTAL RESULTS

5.1. Mackey-Glass Chaotic Time Series

The Mackey-Glass chaotic time series regression problem was used to test the new learning algorithm. The Mackey-Glass chaotic time series [3] is generated from

$$x(t+1) = 0.9x(t) + \frac{0.2x(t-17)}{1+[x(t-17)]^{10}}. \qquad (15)$$

with $x(0) = 1.2$ and $x(t) = 0$ for all $t < 0$.

The input to the neural network are the four series values at $x(t)$, $x(t-6)$, $x(t-12)$, and $x(t-18)$. The output of the neural network will be trained to output $x(t+6)$. The network was trained with 500 samples of the data generated at $t = 201, 202, \ldots, 700$. The network was then tested against 500 samples generated where $t = 5001, 5002, \ldots, 5500$.

The algorithm described in this article (BNGL) was run and compared with the results of ordinary gradient learning (OGL) and that of Adaptive Natural Gradient Learning (ANGL) [1].

5.1.1. Simulations

The simulation was done such that each algorithm was started with the same network size and initial connection weights. The training examples were selected randomly and then input to each neural network training algorithm. The networks started at the same point and shown the same data in an identical random order.

Only the initial values of the Fisher Information matrices and the learning rate values were tweaked so that each algorithm would perform the best that could be determined.

Each algorithm was run through 100 epochs of the training set.

In figure 1 it can be seen that BNGL performs very well. As was mentioned in [2], the adaptive natural gradient is very sensitive to initial conditions and can easily become unstable. However, this new method has been found by these simulations to be very stable and robust to varying initial conditions.

[1] BNGL stands for Bastian Natural Gradient Learning and because B comes after A, since it is largely based on Adaptive Natural Gradient Learning

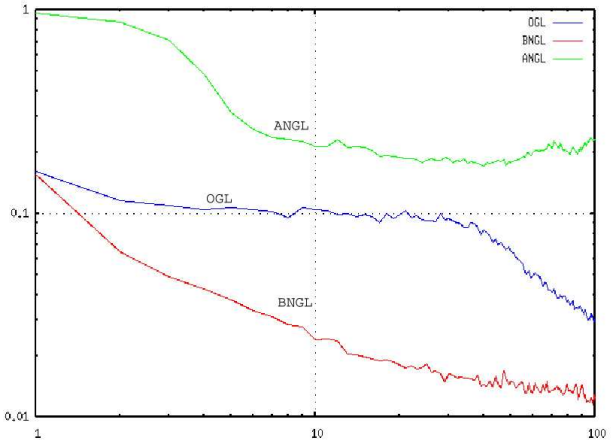

Fig. 1. Mean-squared error per training epoch

	OGL	BNGL	ANGL
Learning rate	0.1	0.005	0.0001
Momentum rate	0.1	$1/t$	$1/t$
Hidden nodes	10	10	10
Network parameters	61	61	61
Additional parameters	11	101	3721
Training data MSE	0.029	0.011	0.17
Test data MSE	0.029	0.012	0.30

Table 1. Performance Comparison of the Learning Algorithms for the Mackey-Glass Time Series

5.2. Fisher Iris Data Set

The Fisher Iris data set [11] consists of an iris species (Setosa, Versicolor, and Virginica) and four measurements of a specimen of that species: petal length, petal width, sepal length and sepal width. The perceptron is to learn to classify the species based on the measurements.

Figure 2 shows some erratic behavior as the algorithm progresses. This occurs as the algorithm avoids singularities, which causes the resulting perceptron to wrongly classify some patterns. The algorithm weaves back and forth, but on a logarithmic plot it seems to become unstable. However, by running the algorithm in excess of 1,000,000 learning cycles this behavior disappears.

5.2.1. Simulations

The data set consists of 150 examples including 50 of each species. The data was whitened by removing the mean of each measurement and then, using the Cholesky decomposition [10], dividing the data set by the square root of the covariance matrix. The data set was split into a training set with 30 examples of each species (90 total) and a test

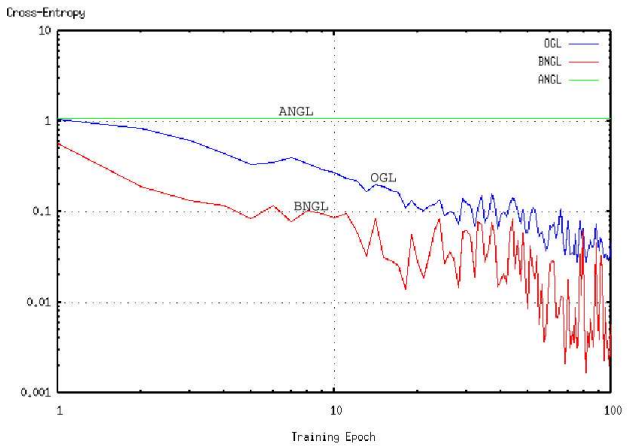

Fig. 2. Cross-entropy of training epochs for Iris problem

	OGL	BNGL	ANGL
Learning rate	0.02	0.005	0.001
Momentum rate	0.0	$1/t$	$1/t$
Hidden nodes	4	4	4
Cross-Entropy	0.028	0.0016	1.096
Errors	9	4	60

Table 2. Performance Comparison of the Learning Algorithms for the Iris Problem

set with 20 examples of each species (60 total). OGL and BNGL were each run through 100 learning cycles.

6. CONCLUSION

By employing the Sobolev gradient, a new learning algorithm was devised that performs very well and requires fewer parameters than the Adaptive Natural Gradient Algorithm. The algorithm, as described in this article, is easy to implement and requires no matrix inversions.

7. REFERENCES

[1] Shunichi Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[2] Shunichi Amari, Hyeyoung Park, and Kenji Fukumizu, "Adaptive method of realizing natural gradient learning for multilayer perceptrons," *Neural Computation*, vol. 12, no. 6, pp. 1399–1409, 2000.

[3] H. Park, S. Amari, and K. Fukumizu, "Adaptive natural gradient learning algorithms for various stochastic models," *Neural Networks*, vol. 13, no. 7, pp. 755–764, 2000.

[4] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, Springer Series in Operations Research. Springer, 1999.

[5] J. W. Neuberger, *Sobolev Gradients and Differential Equations*, Number 1670 in Lecture Notes in Mathematics. Springer, 1997.

[6] Richard A. Brualdi and Herbert J. Ryser, *Combinatorial Matrix Theory*, Number 39 in Encyclopedia of Mathematics and Its Applications. Cambridge University Press, 1991.

[7] Roger A. Horn and Charles R. Johnson, *Matrix Analysis*, Cambridge University Press, 1990.

[8] J. F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. on Signal Processing*, vol. 44, pp. 3017–3030, 1996.

[9] Todd K. Moon and Jacob H. Gunther, "Contravariant adaptation on structured matrix spaces," *Signal Processing*, vol. 82, pp. 1389–1410, 2002.

[10] Todd K. Moon and Wynn C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, 1999.

[11] R. A. Fisher, "The use of multiple measurements in axonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[12] C. R. Rao, "Information and accuracy attainable in the estimation of statistical parameters," *Bulletin of the Calcutta Mathematical Society*, vol. 37, pp. 81–91, 1945.

[13] Louis L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*, Pearson Education, 2002.

A. EFFICIENT ESTIMATION OF PARAMETERS

An estimation algorithm is said to be *efficient* [10, 12, 13] if for all k:

$$G(\theta_k)(\theta_k - \theta^*) = (\nabla\phi)(\theta_k).$$

Solving for, θ^*,

$$\theta^* = \theta_k - G^{-1}(\theta_k)(\nabla\phi)(\theta_k)$$

from this fixed point formula, a recursive update formula is derived:

$$\theta_{k+1} = \theta_k - \eta_k G^{-1}(\theta_k)(\nabla\phi)(\theta_k).$$

Hence, $G^{-1}(\theta)$ is the Riemannian metric at θ [9].