STOCHASTIC CONVERGENCE ANALYSIS OF A TWO-LAYER PERCEPTRON FOR A SYSTEM IDENTIFICATION MODEL

JEFFREY L. VAUGHN[1,2] and NEIL J. BERSHAD[2]

1. The Optical Sciences Company, POB 25309, Anaheim, CA 92825-5309.
2. Department of Electrical and Computer Engineering, University of California, Irvine, CA 92717.

ABSTRACT

This paper summarizes the results of the simulations and analysis of the learning behavior of a simple two-layer perceptron for a nonlinear system identification problem. Although it is difficult to generalize results for nonlinear systems, the analysis may improve our understanding of neural network training. Numerous sub-optimum stationary points occur for this problem and cause difficulties in the correct identification of the unknown system. The sub-optimum convergence points occur in the saturation regions of the various nonlinearities or for pathological cases. The size of the region of suboptimal convergence points may be reduced by increasing the dimensionality of the input data vector. Also, the range for the rate parameter is computed and an improvement to backpropagation is suggested.

1. INTRODUCTION

A series of papers [1-4] have studied the stochastic learning behavior of single layer perceptrons based upon a system identification model for the training data. Transient and steady-state convergence behavior of Rosenblatt's algorithm and the Backpropagation algorithm were examined using a gaussian data model. These results were extended to a partially adaptive two-layer network in [5].

This paper studies the stationary points of a fully adaptive two-layer perceptron for an appropriate nonlinear system identification model. The two-layer perceptron uses a modified form of the backpropagation (BP) algorithm [6]. Since the error surface of the BP algorithm is, in general, multimodal, the algorithm may have several local minima. All local minima occur at the stationary points of the algorithm (where the gradient of the cost function is zero). Additionally, the matrix of the derivatives (discriminant matrix) must be positive definite at the local minima.

The two-layer perceptron attempts to identify the parameters of the specific nonlinear system shown in Figure 1. The nonlinear system has the same type of structure as the specific two-layer perceptron but the parameters are fixed and not known by the perceptron. The training sequence is generated by passing a Gaussian data vector X(n) through two linear systems F_1 and F_2. The output of each linear system is $d_1(n) = F_1^T X(n)$ and $d_2(n) = F_2^T X(n)$. These outputs are each passed through clamping functions sgn(x). The outputs from these functions are weighted by h_1 and h_2, respectively, and summed. The result is passed through a second function, $g_2(x)$ that is only required to be nondecreasing. This function divides N-space into four wedge-shaped regions, each one assigned a value of $g_2(x)$. The adaptive portion of Fig. 1 (the two-layer perceptron) has adjustable weights W_1, W_2, q_1, and q_2. Note that there is no amplitude information in the training sequence. In practice, the BP algorithm requires that W_1 and W_2 be followed by differentiable nonlinearities. On the other hand, a signum function is needed in the analysis to evaluate certain integrals in closed form. The signum function is not differentiable. However, by approximating the signum with the error function, erf(x/δ), (δ small), the formal derivative of the signum function can be replaced by the derivative of the error function. This technique yields a modified BP algorithm which can be analyzed in closed form, used in simulations and preserves the essential features of BP.

2. SYSTEM IDENTIFICATION MODEL

Single and multilayer perceptrons are fundamentally nonlinear systems. The vector inputs to each layer are linearly combined, passed thru a nonlinear function (threshold) and linearly combined in the next layer, etc. The learning behavior of the neural network depends upon the particular training rule used to train the network (adapt the weights) and upon the training data. In the case of perceptron learning, the perceptron can be viewed as a nonlinear adaptive filter. Although the training rule is usually deterministic, the training data are often best modelled by random processes. Therefore, in order to study the statistical behavior of the perceptron, it is necessary to statistically define the training data. Because of the difficulty of analyzing the behavior of nonlinear stochastic

0-7803-2431-5/95 $4.00 © 1995 IEEE

systems, the right choice of model for the training data is critical. The model must reasonably represent real-world data while retaining an inherent simplicity for the analysis. This particular choice for the training data and for the system for generating the desired sequence assures that the joint statistics of d(n) and y(n) are uniquely defined by X(n). The linear operations F_1, F_2, W_1, and W_2 statistically defines d_1, d_2, z_1, and z_2 as jointly Gaussian zero mean variates. Thus, the subsequent nonlinear operations can be handled within an underlying Gaussian framework.

2.1. Gaussian Model for Training Data

The input vector $X^T(n) = [x_1(n), x_2(n), \ldots, x_N(n)]$ consists of independent, identically distributed (i.i.d.) Gaussian variates with zero means and common variance σ_x^2. The covariance matrix for X(n) is modeled as $E[X(n)\,X^T(n)] = \sigma_x^2\, I$. For simplicity and without loss of generality, set $\sigma_x{}^2 = 1$ (i.e. any scaling can be absorbed in F_1 and F_2). The perceptron is trained as follows. A vector X(n) is randomly selected. Two intermediary training signals with values ±1 are generated according to $\mathrm{sgn}[d_i(n)]$, i = 1, 2, where $d_i(n) = F_i^T X(n)$. The final training sequence is given by

$$d(n) = g_2\,[h_1\,\mathrm{sgn}(d_1)+h_2\,\mathrm{sgn}(d_2)] \qquad (2.1)$$

where

$$\begin{aligned} \mathrm{sgn}(x) &= 1, \quad x > 0 \\ &= \mathrm{sgn}(0), \quad x = 0 \\ &= -1, \quad x < 0 \end{aligned} \qquad (2.2)$$

and where sgn(0) is arbitrary but is usually chosen to be zero or unity.

2.2. Perceptron Outputs

The perceptron input is also X(n). The first layer of the perceptron is similar to the first layer of the nonlinear system. Two intermediate signals with values lying in the [-1,1] region are generated according to $g_1[W_i^T X(n)]$, i = 1, 2. The input to the second-layer nonlinearity, z(n), is given by

$$z(n) = q_1\, g_1[W_1^T X(n)] + q_2\, g_1[W_2^T X(n)] \qquad (2.3)$$

Finally, the output of the two-layer neural net, y(n), is given by

$$y(n) = g_2\{z(n)\}\ . \qquad (2.4)$$

The error signal (which drives the learning algorithm) is

$$e(n) = d(n) - y(n) \qquad (2.5)$$

For ease of performing certain expectations, it will be assumed that $g_1(x) = \mathrm{sgn}(x)$, but has the derivative that is associated with $g_1(x) = \mathrm{erf}(x/\delta)$, for some small value δ.

2.3. Modified Back-Propagation Training Algorithm

There are (2N + 2) adjustable weights in the perceptron. If $g_2(x)$ is a nondecreasing function, minimization of E{relative entropy} leads to the update equations given in [6]:

$$q_i(n+1) = q_i(n) + \mu_q\, e(n)\ g_1[W_i^T(n)\, X(n)]\ , \qquad i = 1, 2. \qquad (2.6)$$

$$W_i(n+1) = W_i(n) + \mu e(n)\, q_i(n)\ \ g_1{}'\{W_i{}^T(n)\, X(n)\} X(n) \qquad i = 1, 2. \qquad (2.7)$$

where μ_q and μ are the step-sizes. The first equation is essentially the delta rule for the outer layer.

Eqs. (2.6) and (2.7) are four coupled non-linear stochastic difference equations which describe the behavior of the back-propagation algorithm. The stationary points of the algorithm are determined by applying the orthogonality principle to each of the update terms. This yields four coupled nonlinear deterministic equations for q_1, q_2, W_1 and W_2.

3. RESULTS

3.1. Sub-optimum Local Minima

The stationary points of (2.6) and (2.7) are defined by the orthogonality principle,

$$E\{-e(n)\, g_1[W_i^T(n)\, X(n)]\} = 0 \qquad i = 1, 2. \qquad (3.1)$$

$$E\{-q_1\ e(n)\, g_1{}'\{W_i{}^T(n)\, X(n)\}\, X(n)\} = 0, \qquad i = 1, 2. \qquad (3.2)$$

The discriminant matrix is constructed from the derivatives of (3.1) and (3.2), and is positive definite at a local minimum. The expectations in (3.1) and (3.2) are evaluated in [5] and [7] using methods developed in [8]. The results are summarized here: Stationary points can occur under any of the following conditions;

a) If the outer layer weights are zero, (3.2) is solved and only solutions to (3.1) are needed. These conditions are guaranteed to produce a discriminant matrix with both positive and negative eigenvalues.

b) If one of the outer layer weights is zero, the weight vector associated with the other outer layer weight must be an average of the directions of F_1, F_2. This also guarantees that the discriminant matrix will have both positive and negative eigenvalues. These conditions are saddle points in the cost function.

c) If the inner layer weight vectors lie in the F_1, F_2-plane and lie between F_1 and F_2, then, with careful choice of q_1 and q_2, stationary points can be found. These stationary points are local maxima in all directions orthogonal to the F_1, F_2-plane. When the norms of W_1 and W_2 are small, these stationary points are local minima for directions in the F_1, F_2-plane. When the norms of W_1 and W_2 are large, they become local maxima in these directions. Our simulations and analysis indicate that the norms of W_1 and W_2 are monotonically increasing with time. These stationary points eventually become local maxima except for the pathological case when $q_1 = q_2$ and $W_1 = W_2$.

d) The remaining stationary points display some interesting properties and possibly have some important implications for more complicated networks. These stationary points have the following properties:

i) The vectors F_1, F_2, W_1, and W_2 form a three dimensional subspace of the input data space. If F_1 and F_2 are unit vectors, and $F = F_1+F_2$, and $W_1^TF > 0$ and $W_2^TF > 0$, then the vectors F_1, F_2, W_1, and W_2 intersect a half sphere.

ii) The intersection points form the vertices of a convex quadralateral on the sphere with F_1 and F_2 defining opposite vertices.

iii) These vectors form a stationary point for some set of positive outer layer weights. If $W_1^TF < 0$ or $W_2^TF < 0$ hold, then the associated outer layer weight is negative.

These stationary points can be the source of sub-optimum local minima. Two cases were studied for the output function $g_2(x)$. If $g_2(x) = \alpha x+\beta$, then the discriminant matrix always has negative eigenvalues in directions orthogonal to the three dimensional subspace. Additionally, as the norms of W_1 and W_2 increase, the directions in the subspace will have local maxima at the stationary point. However, if $g_2(x) = \tanh(\alpha x+\beta)$, then the discriminant matrix can have all positive eigenvalues for stationary points. This occurs when $\pm (q_1+q_2)$ lies well into the saturation region of $g_2(x)$ but $\pm (q_1 - q_2)$ lies in the linear region of $g_2(x)$. This can occur for $q_1 = q_2$ as small as $q_1=1.3811/\alpha$. If $\pm (q_1+q_2)$ lies in a somewhat saturated region, the eigenvalues can be positive for directions in the three dimensional subspace. In this case, if the input data vector X has only three elements, adding a fourth element would change a local minima to a saddle point. This has implications for larger networks. Kolmolgorov's mapping theorem [9] suggests that a two layer perceptron network needs at least twice as many perceptrons in the input layer as there are elements in the input data vector. The potential problem of getting stuck at a suboptimal local minima can be minimized. The input data vector can be augmented with enough random inputs to make the number of elements in the input data vector equal to twice the number of input perceptrons. This augmentation has the added potential benefit of supplying the data for a new measure of convergence.

3.2. Rate Parameter

The bounds on the outer layer rate parameter can be computed for our system. The bounds are given by $0<\mu_q<|\Delta \varepsilon/\Delta q|^{-1} \lambda_{max}^{-1}$. λ_{max} $(0 < \lambda_{max} < 2)$ is the largest eigenvalue of the correlation matrix for the outer layer inputs, $\mathrm{sgn}(W_1^TX)$ and $\mathrm{sgn}(W_2^TX)$. The value of $|\Delta \varepsilon/\Delta q|$ is twice the average derivative of $g_2(x)$ over the update interval. $|\Delta \varepsilon/\Delta q|$ is bounded by $g_2'(x)$ at the current point plus the maximum of $g_2'(x)$. This latter term is bounded by twice the maximum of $g_2'(x)$. This first bound provides a changing μ_q which yields better performance than a fixed μ_q. The inner layer will converge with any rate parameter, but the best results are obtained when the inner layer rate parameter is about the same size as δ.

3.3. Improvement to BP

The simulations showed an interesting problem that probably exists in all BP algorithms which was greatly accentuated by the discontinous input nonlinearity in our model. The output error is caused by errors in both the inner and outer layer weights. When the parameters are near convergence, the output error caused by the inner layer weight vector misdirection does not decrease. This output error only occurs less frequently. However, when the error occurs, the BP algorithm uses the large error to correct both the inner layer weight vector direction (a correct response) and the outer layer weights (an incorrect response). The outer layer should not be corrected when the inner layer is corrected. The inner layer derivative in (2.7) can be viewed as a switch for turning on the inner layer update. Good results have been observed when using the inner layer derivative to turn off the outer layer update.

4. CONCLUSIONS

It is difficult to generalize the behavior of non-linear systems using results based on simpler systems. Hence, the results presented here may not generalize to more complex systems. It is also doubtful that this analysis can be generalized to significantly more complex systems. However, the analysis provides a clear understanding of a simple system in a field where most insight has been developed by experience rather than analysis.

REFERENCES

1. J. J. Shynk and S. Roy, "Convergence Properties and Stationary Points of a Perceptron Learning Algorithm," *Proc. of IEEE,* Vol. 78, pp. 1599-1604, Oct. 1990.

2. J. J. Shynk and N. J. Bershad, "Steady-State Analysis of a Single-Layer Perceptron Based on a System Identification Model with Bias Terms," *IEEE Trans. on Circuits and Systems,* Vol. 38, pp. 1030-1042, Sept. 1991.

3. N. J. Bershad, J. J. Shynk, and P. L. Feintuch, "Statistical Analysis of the Single-Layer Backpropagation Algorithm, Pt. I: Mean Weight Behavior," *IEEE Trans. on Signal Processing,* Vol. SP-41, pp. 573-582, Feb. 1993.

4. N. J. Bershad, J J. Shynk, and P. L. Feintuch, "Statistical Analysis of the Single-Layer Backpropagation Algorithm, Pt. II: MSE and Classification Performance," *IEEE Trans. on Signal Processing,* Vol. SP-41, pp. 583-591, Feb. 1993.

5. N. J. Bershad, J. J. Shynk, J. L. Vaughn, and C. F. N. Cowan, "Stochastic Convergence Analysis of a Partially Adaptive Two-Layer Perceptron using a System Identification Model," accepted *Signal Processing,* Nov. 1994.

6. J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computation,* Addison-Wesley, Reading MA, 1991.

7. Jeffrey L. Vaughn, "Stochastic Convergence Analysis of a Two-Layer Perceptron for a System Identification Model," Ph.D. Dissertation, ECE Dept., UC Irvine, Irvine CA (in preparation).

8. S. Nabeya, "Absolute anad Incomplete Moments of the Multivariate Normal Distibution," *Biometrika,* Vol. 48, Nos. 1 and 2, pp. 77-84, 1961.

9. R. Hecht-Nielsen, "Neurocomputing," Addison-Wesley, New York, 1990.

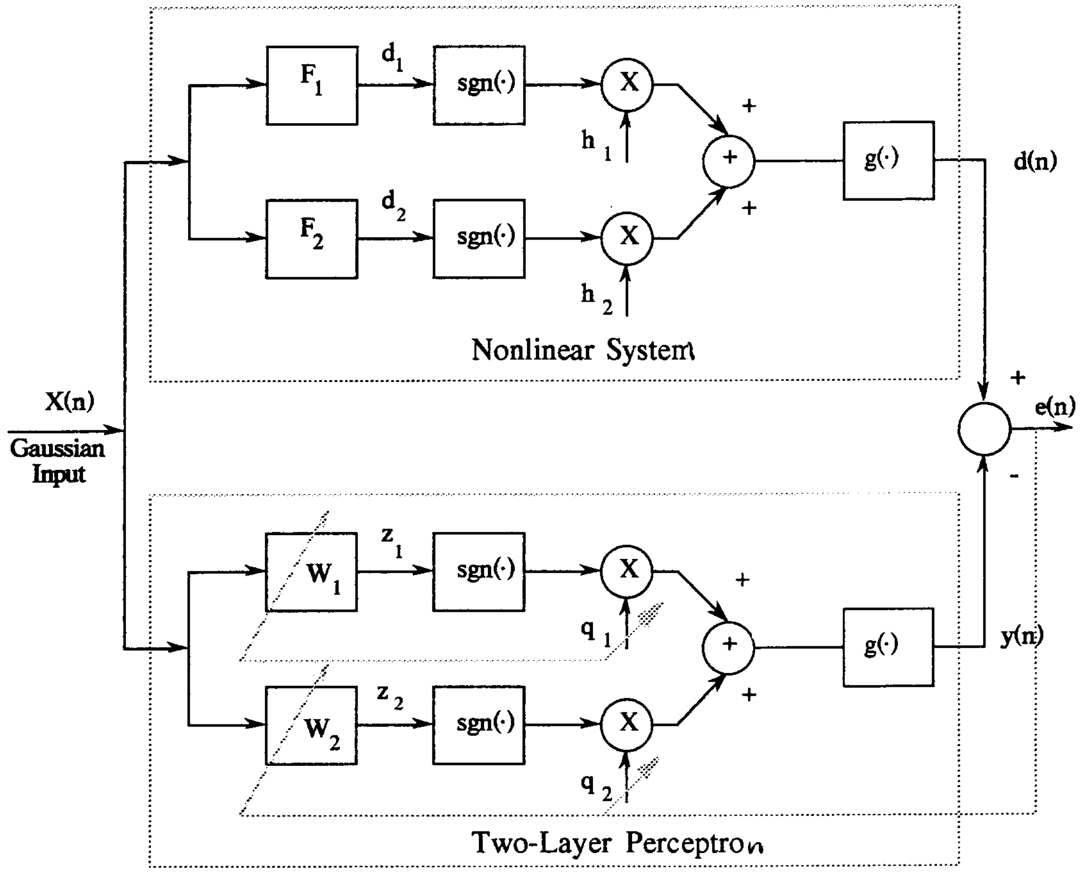

Figure 1. Nonlinear System Identification Using a Two-Layer Perceptron