STATISTICAL ANALYSIS OF THE SINGLE-LAYER BACKPROPAGATION ALGORITHM FOR NOISY TRAINING DATA

NEIL J. BERSHAD[1], NICOLAS CUBAUD[2], and JOHN J. SHYNK[3]

1. Dept of Electrical and Computer Engineering, University of California, Irvine, CA 92717.

2. Dept of Electrical Engineering, University of Toulouse, Toulouse, France, and Visiting Scholar, University of California, Irvine, CA 92717.

3. Center for Information Processing Research, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.

ABSTRACT

The statistical learning behavior of the single-layer backpropagation algorithm was recently analyzed using a system identification formulation for noise-free training data [1,2]. Transient and steady-state results were obtained for the mean weight behavior, mean-square error (MSE), and probability of correct classification. This paper extends these results to the case of noisy training data. Three new analytical results are obtained: 1) the mean weights converge to finite values even when the bias terms are zero, 2) the MSE is bounded away from zero, and 3) the probability of correct classification does not converge to unity. However, over a wide range of signal-to-noise ratios (SNRs), the noisy training data does not have a significant effect on the perceptron stationary points relative to the weight fluctuations. Hence, one concludes that noisy training data has a relatively small effect on the ability of the perceptron to learn the model weight vector F.

I. INTRODUCTION

The backpropagation (BP) algorithm is a widely-used training procedure that adjusts the connection weights of a multilayer perceptron [3]. It is a gradient-descent method that minimizes the mean-square error (MSE) between the perceptron output signals and a set of training or desired response signals. The BP algorithm is a nonlinear procedure because of the nonlinear threshold element contained in each node, and its behavior is very complex because of the layered structure. These nonlinearities make it difficult to analyze the behavior of the connection weights and the MSE, even for a small number of nodes. However, using a system identification model of the desired response signal, the single-layer BP algorithm has been analyzed regarding its mean weight behavior, MSE, and classification performance for noise-free training data [1,2]. Since the training data and the perceptron input generally are not related deterministically, this may not be a realistic model in practice. This paper models this difference as an independent additive noise at the perceptron input, and extends the results of [1,2] to the case of noisy training data.

II. CONVERGENCE RESULTS

The system identification model of the desired response signal is shown in Fig. 1. The training sequence is generated by passing a Gaussian data vector X(n) through a linear system defined by the N-dimensional vector of weights F. The linear output of the system, $d(n) = F^T X(n)$, is then converted to a binary (± 1) signal via the signum function. The adaptive component in Fig. 1, corresponding to a single-layer perceptron, consists of a set of adaptive weights W(n) whose input is an additive noise-corrupted version of the data vector X(n), i.e., $Y(n) = X(n) + V(n)$. The noise vector V(n) is white and Gaussian with zero mean and covariance $\sigma_V^2 I$; it is also independent of X(n). The linear output $W^T(n)Y(n)$ is processed by a smooth nonlinearity to generate a signal bounded in magnitude by 1. The perceptron output is then compared to the binary training signal, and the

0-7803-2431-5/95 $4.00

weights are updated by the single-layer BP algorithm:

$$W(n+1) = W(n) + \mu e(n)\, f'[W^T(n)Y(n)]Y(n) \tag{2.1}$$

$$e(n) = \text{sgn}\,[F^T X(n)] - f[W^T(n)\, Y(n)] \tag{2.2}$$

where e(n) is the output error, f(·) is the smooth nonlinearity, f '(·) denotes its first derivative, μ is a positive step size that controls the convergence properties of the algorithm, and sgn (·) is the signum function (with sgn(0) = 1). The second term in the right-hand side of (2.1) represents an instantaneous estimate of the gradient of the MSE. The soft nonlinearity is the following error function:

$$f(x) = \sqrt{\frac{2}{\pi}}\frac{1}{\sigma}\int_0^x e^{-z^2/2\sigma^2}\,dz \tag{2.3}$$

where $\sigma^2 \geq 0$ controls the steepness of f(·). The input vector $X^T(n) = [x_1(n), x_2(n), \ldots, x_N(n)]$ consists of independent identically distributed Gaussian variates with zero means and common variance σ^2_x and covariance matrix $E[X(n)\, X^T(n)] = \sigma_x^2 I$.

The subsequent analysis closely follows the procedure used in [1,2]. Hence, only the results are presented here. The details are a straightforward application of the theory presented in [1,2].

A recursion for the mean weights can be obtained by averaging (2.1) in two steps, first by conditioning on W(n), and then by averaging over the randomness in W(n) using a small μ approximation, resulting in

$$\overline{W}(n+1) = \overline{W}(n) + \mu\frac{2\sqrt{\alpha}}{\pi} \times \left[\frac{\left\{F - \frac{F^T\,\overline{W}(n)\,\overline{W}(n)}{\alpha a^2 + P(n)}\right\}}{\sqrt{F^TF}\left[\alpha a^2 + P(n)\left(1-\rho_{12}^2\right)\right]^{\frac{1}{2}}} - \frac{a\overline{W}(n)}{\left[\alpha a^2 + P(n)\right]\left[\alpha a^2 + 2P(n)\right]^{\frac{1}{2}}}\right] \tag{2.4}$$

where $P(n) = \overline{W}^T(n)\overline{W}(n)$, $E[W(n)] = \overline{W}(n)$, and $\alpha = \frac{\sigma_x^2}{\sigma_x^2+\sigma_v^2}$, $a = \frac{\sigma}{\sigma_x}$, $\rho_{12} = \frac{\sqrt{\alpha}F^T\overline{W}(n)}{\sqrt{F^TF\,P(n)}}$.

Inserting $\overline{W}(n) = \frac{k(n)F}{\sqrt{F^TF}}$ into (2.4) yields a scalar nonlinear difference equation for k(n):

$$k(n+1) = k(n) + \frac{2\mu a\sqrt{\alpha}}{\pi\left[\alpha a^2 + k^2(n)\right]} \times \left[\frac{\alpha a}{\sqrt{\alpha a^2 + (1-\alpha)\,k^2(n)}} - \frac{k(n)}{\sqrt{\alpha a^2 + 2\,k^2(n)}}\right] \tag{2.5}$$

with steady-state solution ($k_{ss} = \lim_{n\to\infty} k(n)$)

$$k_{ss} = \frac{a\alpha}{\sqrt{1-\alpha}}. \tag{2.6}$$

The conditional MSE is

$$E\left[e^2(n)|W\right] = 1 + \frac{2}{\pi}\sin^{-1}\left[\frac{(\sigma_x^2+\sigma_v^2)W^TW}{(\sigma_x^2+\sigma_v^2)W^TW + \sigma^2}\right] - \frac{4}{\pi}\sin^{-1}\left[\frac{\sigma_x F^T W(n)}{\sqrt{F^TF}\sqrt{(\sigma_x^2+\sigma_v^2)W^TW + \sigma^2}}\right], \tag{2.7}$$

which, after replacing W by $\overline{W}(n) = \frac{k(n)F}{\sqrt{F^TF}}$, yields

$$E\left[e^2(n)|\overline{W}(n)\right] = 1 + \frac{2}{\pi}\sin^{-1}\left[\frac{k^2(n)}{k^2(n)+\alpha a^2}\right] - \frac{4}{\pi}\sin^{-1}\left[\frac{\sqrt{\alpha}k(n)}{\sqrt{k^2(n)+\alpha a^2}}\right]. \tag{2.8}$$

When k(n) converges to k_{ss},

$$\lim_{n\to\infty} E\left[e^2(n)|\overline{W}(n)\right] = 1 - \frac{2}{\pi}\sin^{-1}\alpha, \tag{2.9}$$

which can be verified as the minumum of the MSE surface. Note that as α approaches 1 (noise-free training), (2.5)-(2.9) reduce to the results in [1,2] as expected (i.e., (2.6) approaches infinity and (2.9) approaches zero).

The probability of correct pattern classification P_c can be evaluated in terms of k(n) and the trace of the covariance matrix of the weight fluctuations, yielding

$$P_c = \frac{1}{2} \times \left[1 + \frac{2}{\pi} \tan^{-1} \left\{ \frac{\sqrt{\alpha}\, k(n)}{\sqrt{(1-\alpha)k^2(n) + \mathrm{tr}[K_{\varepsilon\varepsilon}(n)]}} \right\} \right] \tag{2.10}$$

where tr [·] denotes the trace of a matrix. The scalar term $\mathrm{tr}[K_{\varepsilon\varepsilon}(n)]$ satisfies a deterministic recursion given in [4]. Again note that (2.10) reduces to [2-(3.15)] for $\alpha = 1$. For small μ, $\mathrm{tr}[K_{\varepsilon\varepsilon}(n)]$ is proportional to μ in steady-state. Thus,

$$\lim_{\mu \to 0} P_c = \frac{1}{2} \left[1 + \frac{2}{\pi} \tan^{-1} \left\{ \frac{\sqrt{\alpha}}{\sqrt{(1-\alpha)}} \right\} \right]$$

$$= \frac{1}{2} \left[1 + \frac{2}{\pi} \tan^{-1} \left\{ \frac{\sigma_X}{\sigma_V} \right\} \right] \quad . \tag{2.11}$$

Even if the number of training samples increases without bound, P_c can never exceed (2.11). Thus, although the perceptron precisely learns the correct hyperplane F, it will not make error-free decisions because the perceptron input is a noise-corrupted version of X(n).

III. COMPUTER SIMULATIONS

Monte Carlo (MC) simulations of (2.1) and (2.2) have yielded results in excellent agreement with the theory [4]. The inputs X(n) and V(n) were independent vectors, jointly Gaussian with zero means and covariance matrices σ^2_X I and σ^2_V I, respectively. The perceptron had two adaptive weights (N = 2) and the underlying weight vector was $F = [-1,1]^T$. The weights were initialized to zero, the step size was μ = 0.005, $\sigma^2 = \sigma^2_X = 1$, and the SNR was varied over 0, 10, 20, and ∞ dB. The weight trajectories were averaged over 100 independent computer runs.

IV. THEORETICAL PREDICTIONS

Figures 2-4 display computer evaluations of (2.5), (2.8) and (2.10) for 5000 learning samples and the parameters used in the previous MC simulations. Figure 2 shows that the scale factor k(n) has converged only for SNR = 0 dB. The converged value agrees with that given by (2.6), i.e., $k_{ss} = 0.707$. Figure 3 shows that the MSE has converged for SNR = 0 and nearly converged for SNR = 10 dB; (2.9) yields 0.667 and 0.274, respectively, for the asymptotes. Figure 4 shows that P_c has converged for SNR = 0, 10, and 20 dB; (2.10) yields 0.75, 0.90, and 0.968, respectively, for the asymptotes. Figure 4 indicates that, although the classification performance of the perceptron is heavily dependent on the SNR, the classification performance relative to (2.11) is not. The perceptron nearly achieves the classification performance given by (2.10) after three or four hundred samples and roughly independent of the SNR.

V. CONCLUSIONS

A statistical analysis of the convergence behavior of the single-layer backpropagation algorithm for noisy Gaussian training data has been presented. The analysis is based upon a nonlinear system identification model of the desired response signal which is capable of generating an arbitrary hyperplane decision boundary. It is demonstrated that, contrary to the noise-free case [1], the weights converge to finite values. The algorithm, on average, quickly learns the correct hyperplane associated with the system identification model but, because of the noisy training data, the MSE is bounded away from zero and the probability of correct classification does not converge to unity (unlike the noise-free cse presented in [2]). However, the noisy training data does not have a significant effect on the perceptron mean weights relative to their fluctuations. Hence, one may conclude that noisy training data has a relatively small effect on the ability of the perceptron to learn the model weight vector F. This behavior is probably due to the time-averaging properties of the algorithm during the learning phase.

REFERENCES

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

[2] N. J. Bershad, J. J. Shynk, and P. L. Feintuch, "Statistical Analysis of the Single-Layer Backpropagation Algorithm: Part II--MSE

and Classification Performance," *IEEE Trans. on Signal Processing*, Vol. 41, pp. 583-591, Feb. 1993.
3] B. Widrow and M. A. Lehr, "30 Years of Adaptive Neural Networks:Perceptron, Madaline and Backpropagation," *Proc. IEEE*, Vol. 78, pp. 1415-1442, Sept. 1990.
[4] N. J. Bershad, N. Cubaud, and J. J. Shynk, "Stochastic Convergence Analysis of the Single-Layer Backpropagation Algorithm for Noisy Training Data," submitted to *IEEE Trans. on Signal Processing*, Nov.1994.

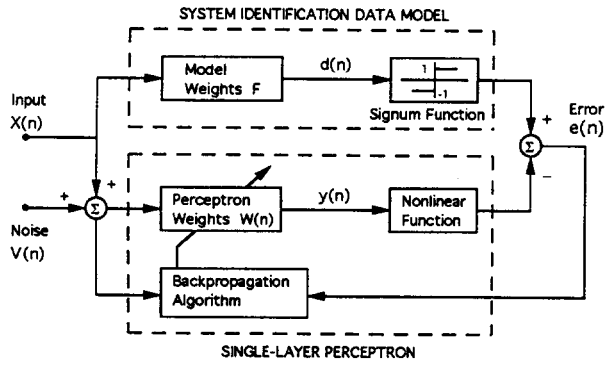

Fig. 1 - Single layer perceptron with desired response model and noisy training data.

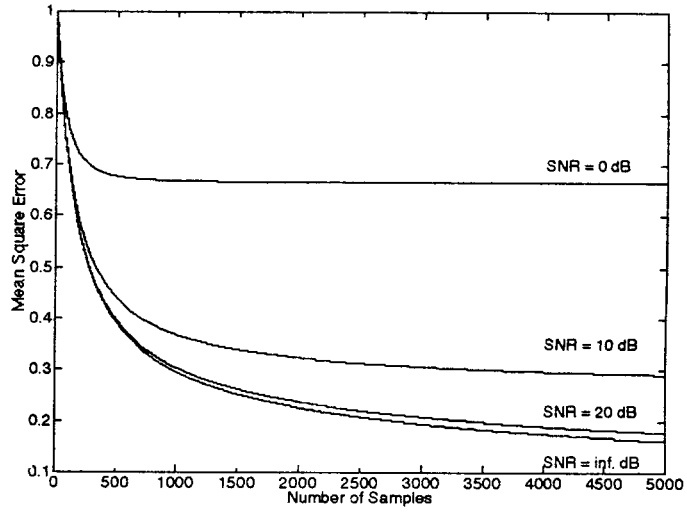

Fig. 3 - Mean-square-error trajectories for SNR = 0, 10, 20, inf. dB ($\mu = 0.005$, $\sigma^2 = \sigma_x^2 = 1$)

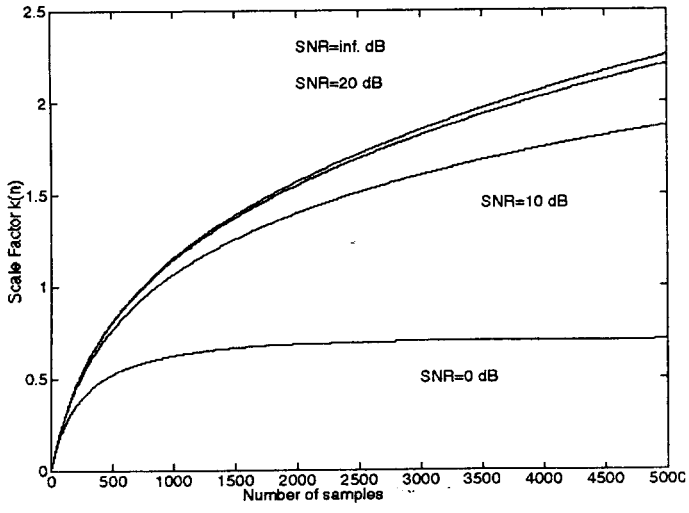

Fig. 2 - Scale factor k(n) trajectories for SNR = 0, 10, 20, inf. dB ($\mu = 0.005$, $\sigma^2 = \sigma_x^2 = 1$).

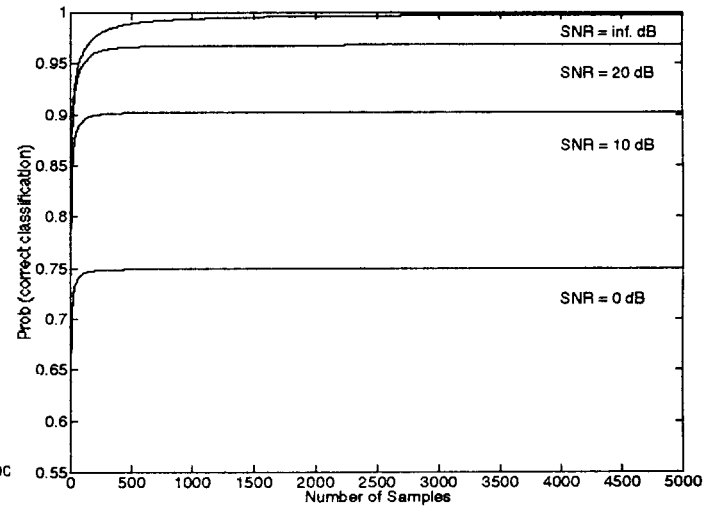

Fig. 4 - Prob (Correct Classification) trajectories for SNR = 0, 10, 20, inf. dB ($\mu = 0.005$, $\sigma^2 = \sigma_x^2 = 1$).