모의 패턴생성 프로세스를 이용한 다단신경망분류기의 성능분석

박 동 선†

요 약

본 논문에서는 클래스내부와 클래스간의 변화를 정확하게 제어할 수 있는 랜덤 프로세스 모델을 설계한다. 설계된 랜덤 프로세스를 제어하는 프로세스 내부의 파라메타들을 변화시키며, 프로세스간의 통계적인 차이와 랜덤 잡음을 변화시켜 학습을 위한 패턴들을 생성한다. 이 랜덤 프로세스 모델에서 생성된 패턴들을 이용하여 역전파알고리즘으로 학습된 다단신경망의 성능을 평가한다. 평가 실험결과는 패턴 분류문제에서 일반화된 통계적인 거리가 분류문제의 난이도에 대한 좋은 예측기가 되는 것을 보여준다. 또한 본 논문에서는 다단신경망의 성능과 베이스패턴분류기의 성능을 비교하기 위하여 베이스분류기의 이론적인 성능분석과 모의실험을 통한 평가를 하였다. 다단신경망의 분류성능이 이론적인 성능과 실험치와 매우 근사하며 그 두 성능 중간에 위치함을 발견하였다.

Performance Analysis of Multilayer Neural Net Classifiers using Simulated Pattern-Generating Processes

Dong-Sun Park†

ABSTRACT

We describe a random process model that provides sets of patterns with precisely controlled within-class variability and between-class distinctions. We used these patterns in a simulation study with the back-propagation network to characterize its performance as we varied the process-controlling parameters, the statistical differences between the processes, and the random noise on the patterns. Our results indicated that the generalized statistical difference between the processes generating the patterns provided a good predictor of the difficulty of the classification problem. Also we analyzed the performance of the Bayes classifier with the maximum-likelihood criterion and we compared the performance of the neural network to that of the Bayes classifier. We found that the performance of neural network was intermediate between that of the simulated and theoretical Bayes classifier.

1. Introduction

Classification of one-dimensional information containing signals, such as radar and sonar patterns, represent an important application of neural networks, particularly back-propagation networks[1,3]. However, effective use of this approach requires a much deeper understanding of the relationship between the characteristics of the network and those of the classifi-

† 정 회 원:전북대학교 정보통신공학과
논문접수:1996년 8월 25일, 심사완료:1996년 2월 15일

cation problem. One fundamental aspect of this relationship is the effect that the inherent variability of patterns from the same class and the distinction between patterns of different classes has on performance, specifically on the generalization capability of the network. Within class variability takes two distinct forms: unexplained or random variation in the generation of each specific pattern and additive random noise in its measurement. Using real data can provide some insight into this relationship, but controlling or even characterizing the variability in real data is difficult.

Several groups have used random variables or vectors that were sampled directly from gaussian distributions as the observed pattern in studying the relationship between the performance of neural network classifiers and the difficulty of the classification problem [4, 5, 7]. Although this approach provides some indication of the relationship between performance and problem difficulty, the results are not directly extendable to problems where the randomness in the observed pattern is more complex, such as in radar of sonar patterns. We believe that our method provides a more appropriate approach, particularly generating processes. Along these lines, Ahalt [1] simulated radar returns and studied how architecture and noise levels effected performance.

In the present paper, we describe a random process model that provides sets of patterns with precisely controlled within-class variability and between-class distinctions. The behavior of the Bayes classifier for a random process model is analyzed to obtain the theoretical performance of the Bayes classifier. We present results of a simulation study of the back-propagation network using this model and, we compare this performance to that of the Bayes classifier.

2. Random Process Model

In order to provide training and test data for our studies of neural network classifiers, we developed a general random process model for generating sets of patterns with controlled statistical properties. In this formulation, a separate random process represented each class, and the processes contained two types of variability: random variation in the generation of each specific pattern from the class and additive random noise in its measurement. Figure 1 illustrates the operation of this random process model.

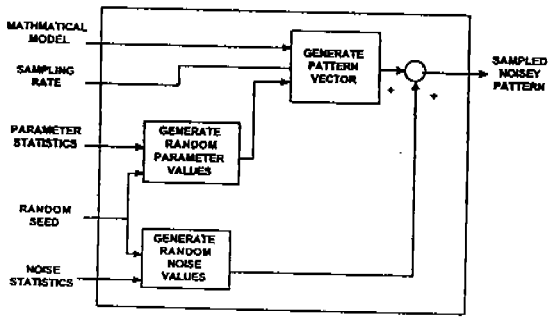
To provide the first type of variability, we assigned each process one of three mathematical expressions that contained several random parameters. Equations 1, 2, and 3 give the three expressions used in this study.

$$x(t) = X_0[u(t-T_0) - u(t-T_0-T_w)] \quad (1)$$

$$x(t) = X_0[x_1(t-T_0) - x_1(t-T_0-T_w)] \quad (2a)$$

$$x_1(t) = [1-e^{-At}\sin(2\pi Ft)]u(t) \quad (2b)$$

$$x(t) = \sum_{i=1}^{N} A_i \sin(2\pi F_i t + \theta_i) \quad (3)$$



(Fig. 1) The random process model for generating sets of patterns representative of a class. The model accepted a mathematical expression containing several parameters, the mean and variance for these parameters, the variance for the additive noise, and the sampling rate. It accessed a random generator for specific parameters and noise values for that pattern.

The first of these describes a rectangular pulse with three random parameters (i.e., amplitude, on-set time,

and pulse-width); the second describes a damped oscillatory pulse with five random parameters (i.e., amplitude, on-set time, pulse-width, exponential coefficient, and oscillatory frequency); and the third describes a pulse composed of N sinusoids with 3N random parameters (i.e., the amplitude, frequency, and phase shift for each sinusoid). For simplicity, we used N=3 and we fixed the phase shift at 0. In this study, we always used gaussian distributions for all parameters, so that the mean and variance for all model parameters characterized a process.

When using one of the three random process models to generate a representative pattern, the program selected a set of random parameters from a set of specified probability distributions. Repeatedly accessing a process produced different patterns that were represented of the underlying process. For example, accessing a rectangular-pulse process repeatedly, produced a set of patterns that had different amplitudes, on-set times, and pulse-widths where these features depended upon the mean and variance for the corresponding distribution.

To provide the second type of variability, the random process perturbed the sampled version of the pattern by adding random values to each sample value. These perturbation values were selected form a gaussian distribution with a zero mean and a variance defined by the desired signal-to-noise ratio(SNR).

In a multiple-class problem, separate processes represented each class, and at least one parameter (i.e., the distinguishing parameter) had to have a different distribution in each process. We examined classification problems with one or more distinguishing parameters. We found it convenient to quantify the statistical difference between processes by a generalized scalar distance (i.e., the Mahalanobis distance) between the parameter distributions. Equation 4 defines this distance (R) as a function of the mean vectors (m1 and m2) and the covariance matrix (C).

$$R=[(m_1-m_2)^t C^{-1}(m_1-m_2)]^{1/2} \quad (4)$$

In this study we always used independent random variables and identical variances for each distribution : under these conditions Equation 4 simplified to Equation 5 in which σ is the standard deviation used for all parameters.

$$R=[(m_1-m_2)^t (m_1-m_2)]^{1/2}/\sigma \quad (5)$$

This last equation indicates that with these constraints R equals the separation between the mean vectors normalized by the standard deviation.

3. Performance Analysis of The Bayes Classifier

We compared the performance of the neural network classifier to that of the Bayes classifier using both simulation and analytical techniques. In the Bayes classifier, each class had a decision function, and the classifier assigned an observed pattern to the class with the maximum decision function. Equation 6 gives the discrete form of this decision function for class i assuming equal class probabilities, a binary cost matrix with zero diagonal terms, and a gaussian distribution on the model parameters [9].

$$h_i=2\sum_k z_i(k)\,x(k)-\sum_k z_i(k)^2 \quad (6)$$

In this equation $x(k)$ is the observed pattern and $z(k)$ is the prototype pattern for class i. We obtained prototype patterns using mean values for all model parameters. The first term on the right-hand side of Equation 6 is twice the cross correlation between the input pattern and the prototype; the second term is the power in the prototype. Conceptually the power of the prototype can be treated as a threshold so that the classification actually depends on the first term in Equation 6. In conventional signal detection formulations cross correlation is accomplished by matched filters [9].

Equation 7 gives the continuous form of Equation

6.

$$h_i = 2\int_0^T z_i(t)x(t)\,dt - \int_0^T z_i(t)^2\,dt \tag{7}$$

Under certain circumstances, Equation 7 can be used to derive an analytical expression for the expected value of the error rate for the Bayes classifier. We develop equations that describe the expected error rates for the Bayes classifier for the rectangular-pulse model with amplitude as the distinguishing parameter, and we list analogous results for this model with onset time and pulse-width as the distinguishing parameters. In this development, $\tilde{A}_i$, $\tilde{T}_0$, and $\tilde{T}_W$ represent mean values of the parameter distributions, and σ_{A_i}, σ_{T_0} and σ_{T_w} represent the corresponding standard deviations.

With amplitude as the distinguished parameter, Equation 8 defines the prototype pattern.

$$z_i(t) = \tilde{A}_i[u(t-\tilde{T}_0) - u(t-\tilde{T}_0-\tilde{T}_w)] \tag{8}$$

Substituting Equation 8 into Equation 7 and subtracting h_1 from h_2 leads to Equation 9 in which A_i, T_0, and T_w are the parameters for the pattern being classified and j is either 1 or 2.

$$h_1 - h_2 = (\tilde{A}_1 - \tilde{A}_2)\tilde{T}_w\, G_j \tag{9}$$

$$G_j = 2\alpha A_j - (\tilde{A}_1 + \tilde{A}_2) \tag{9a}$$

$$\alpha = \frac{1}{\tilde{T}_w}\int_{\tilde{T}_0}^{\tilde{T}_0+\tilde{T}_w}[u(t-T_0) - u(t-T_0-T_w)]\,dt \tag{9b}$$

The sign of $(h_1 - h_2)$, shown in Equation 9a, and hence the sign of G_j, shown in Equation 9b, defines the class assignment.

Since G_j is a linear function of the gaussian random variable A_j, G_j also has a gaussian distribution. Its mean and standard deviation are defined in Equation 10.

$$\tilde{G}_j = E[2\alpha A_j - (\tilde{A}_1 + \tilde{A}_2)] = 2E[\alpha]\,\tilde{A}_j - (\tilde{A}_1 + \tilde{A}_2) \tag{10a}$$

$$\sigma_{G_j} = E[(G_j - E[G_j])^2]^{1/2} = 2E[\alpha]\sigma_A \tag{10b}$$

An expression for $E[\alpha]$ is obtained by rewriting Equation 9b using min and max operations as shown in Equation 11.

$$\alpha = \frac{1}{\tilde{T}_w}[\min\{(\tilde{T}_0 + \tilde{T}_w), (T_0 + T_w)\} - \max\{\tilde{T}_0, T_0\}] \tag{11}$$

Considering all possible combinations for the min and max operations leads to four equally likely simplified expressions for Equation 11. Equation 12 results when the expected value of Equation 11 is taken using these four simplified expression.

$$E[\alpha] = \frac{1}{4\tilde{T}_w}(E[(\tilde{T}_0 + \tilde{T}_w) - T_0] + E[(T_0 + T_w) - \tilde{T}_0] \tag{12a}$$

$$+ E[(T_0 + T_w) - T_0] + E[(\tilde{T}_0 + \tilde{T}_w) - \tilde{T}_0])$$

$$= \frac{1}{4\tilde{T}_w}((\tilde{T}_w - 0.675\sigma_{T_0}) + (\tilde{T}_w - 0.675(\sigma_{T_0} + \sigma_{T_w}))$$

$$+ (\tilde{T}_w - 0.675\sigma_{T_w}) + \tilde{T}_w) \tag{12b}$$

$$= 1 - \frac{0.3375(\sigma_{T_0} + \sigma_{T_w})}{\tilde{T}_w} \tag{12c}$$

Substituting Equation 12c into Equation 10 produces expressions for the mean value and the standard deviation of G_j in term of them mean values and standard deviations of the original parameters. These expressions are shown the Equation 13.

$$\tilde{G}_j = 2\left(1 - \frac{0.3375(\sigma_{T_0} + \sigma_{T_w})}{\tilde{T}_w}\right)\tilde{A}_j - (\tilde{A}_1 + \tilde{A}_2) \tag{13a}$$

$$\sigma_{G_j} = 2\left(1 - \frac{0.3375(\sigma_{T_0} + \sigma_{T_w})}{\tilde{T}_w}\right)\sigma_A \tag{13b}$$

For any combination of parameter distribution values, $\tilde{G}_1$, $\tilde{G}_2$, and σ_G can be computed. Since G_1 and G_2 are gaussian random variables, standard bayesian analysis can be used to compute the probability of the two types of errors(i.e., class 1 assigned to class 2 and vice versa), which is equivalently the expected error rate. Equation 14 gives an expression for the expected error rate.

$$E[ERROR\ RATE]=\frac{1}{2}\left(1+N(\tilde{G}_1,\ \sigma_{G_1})-N(\tilde{G}_2,\ \sigma_{G_2})\right) \tag{14}$$

In this Equation, $N(.)$ is the normal distribution function. The resulting expression is given in Equation 14; Equation 13 provides expressions for the mean value and the standard deviation required in this equation with amplitude as the distinguishing parameter. Park gives an analogous development for the rectangular-pulse model with on-set time and pulse-width as the distinguishing parameters [6].

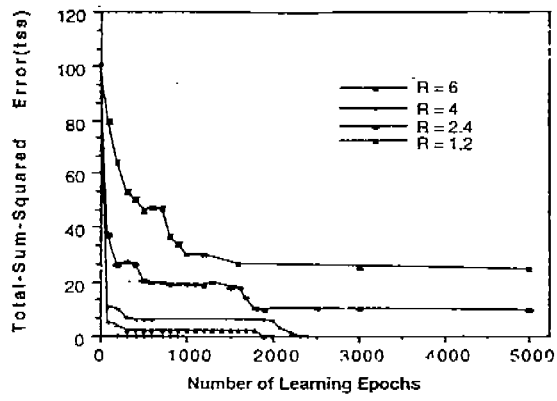
4. Neural Net Classifiers for Random Process Models

In a series of simulation studies using three different pattern models, we measured the performance of neural network classifiers as we varied the Mahalanobis distance between the random processes generating the patterns. For convenience, we kept the variance constant and manipulated the separation between the mean vectors to control the Mahalanobis distance. In one series we did alter the variance, but we found identical results, and so for most studies we simply varied R. In these studies we used the random processes to generate two independent sets of patterns, one for training the network and one for testing it. In most simulations, we used a SNR of 15; however, in several studies we varied the SNR in order to characterize its effect on performance.

We completed studies using two classes with the rectangular-pulse model, the damped-oscillatory-pulse model, and the sum-of-sinusoids model using various combinations of distinguishing parameters. In all cases, the network had 32 input units, four hidden units, and two output units. We used the standard back-propagation training algorithm with weights being updated after presenting the entire set of training patterns [8]. We used a learning rates between 0.01 and 0.05 and a momentum coefficient of 0.9. We terminated training when the total-summed-square-error reached 0.04 or after 20,000 training epochs or when the change in the total-summed-squared-error over 2000 epochs was less than 0.01. After training we characterized performance using the error rate with the test data.

Figure 2 shows learning curves for the rectangular-pulse model with amplitude as the distinguishing parameter. This figure shows that with a large R ($R=4$ or 6, i.e., widely separated classes) the error decreased rapidly to zero, but, as R decreased, the network required more and more training epochs. For $R=2.4$ or less, the learning curves plateaued at a nonzero value, indicating that the network was unable to learn all the patterns in the training set because of the closeness of the two processes. We do have some preliminary evidence that using more complicated networks reduced these final errors.

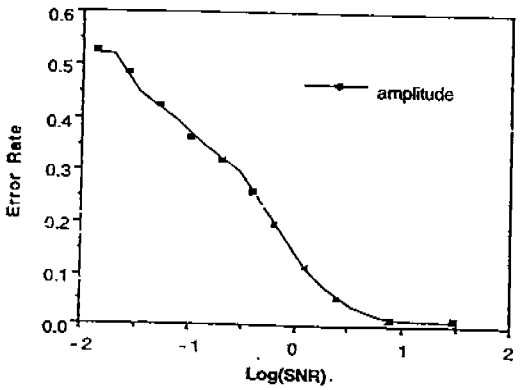
Figure 3 shows error rate as a function of the log of the SNR for the two-class problem with the rectangular-pulse model using amplitude as the distinguishing parameter.



(Fig. 2) Learning curves for the rectangular-pulse model with amplitude as the distinguishing parameter.

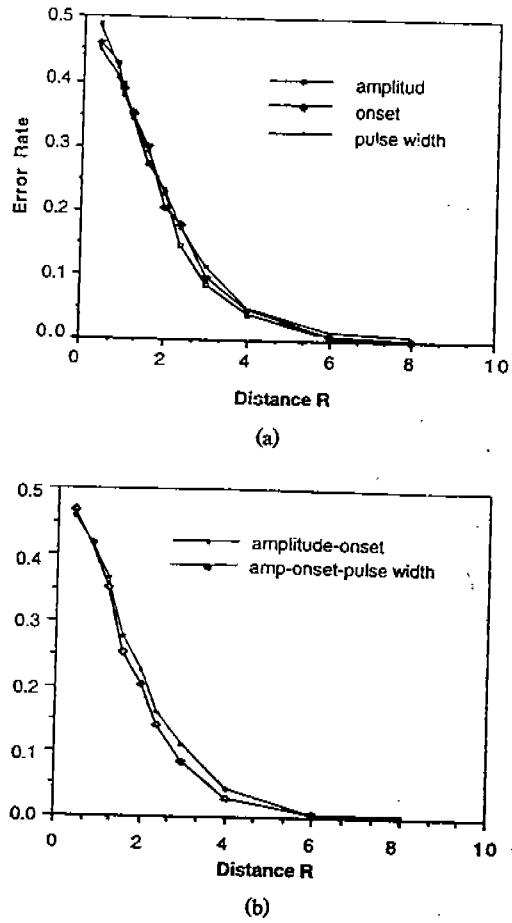
When the log of the SNR was −1.71 or lower (i.e., a SNR of 0.0195 or lower) the error rate approached 0.5 (i.e., the error rate for totally random classification).

At the other extreme, the error rate approached zero, and when the log of the SNR was above 1.49 (i.e., a SNR of 31.2) the error rate equaled 0.0065 or less. In between, the error rate decreased in approximately a linear fashion with a value near 0.1 when the log of the SNR equaled 0 (i.e., a SNR of 1). It shows that the neural network classifier is very tolerant to the gaussian noise.



(Fig. 3) Error rate as a function of the log of the SNR of the patterns for the two-class problem with the rectangular-pulse model. For low SNRs the error rate approached 0.5, the value obtained with random classification. The relationship was approximately linear for log(SNR) between −1.71 and 0.893, which corresponds to SNR range of 0.0195 to 7.91 on a linear scale.

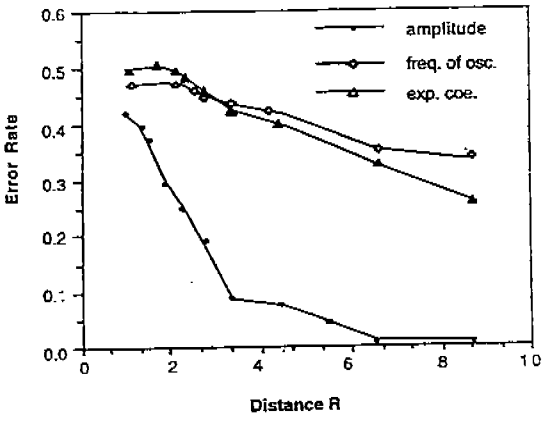
Figure 4 shows error rate as a function of R for the rectangular-pulse for various combinations of distinguishing features in a two-class problem. All cases demonstrated a nearly identical relationship. For small R (0.4 was the smallest that we investigated), the error rate approached 0.5, a value consistent with a totally random classification. As R increased, the error rate fell rapidly to approximately 0.15 at $R=2$, where the effective separation between the two processes equaled two standard deviation. It became and 0.0 at $R=8$.



(Fig. 4) Error rates as a function of the generalized distance R for the two-class problem with the rectangular-pulse model using (a) one distinguishing parameter and (b) using combinations of distinguishing parameters. All curves showed an identical relationships. For small distance, the error rate approached 0.5, a rate obtained by random classification. It then decreased rapidly as R increased, reaching a value near 0.15 with R =2, where the effective separation between the two processes equaled two standard deviation.

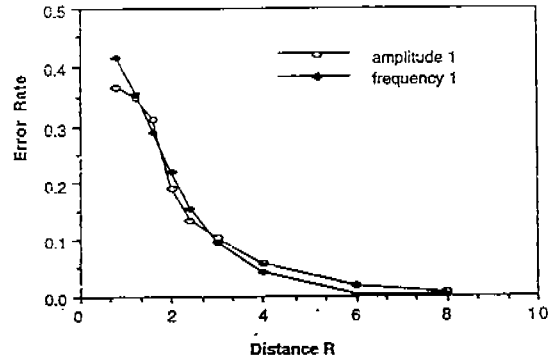
Figure 5 shows the error rate for the two-class problem with various combinations of distinguishing parameters for the damped-oscillatory-pulse model. In general, these data exhibit the same relationship seen with the rectangular-pulse model. However, they

decreased more slowly than with the rectangular-pulse model; for example, the error rate at $R=2$ was nominally 0.25, which compares to a value of 0.15 for the rectangular-pulse model. Moreover, with the exponential coefficient or the frequency of oscillation as the single distinguishing parameter, the error rate decreased much more slowly than with the other distinguishing parameters in this model. Although we used a sampling frequency well above the oscillation frequency(i.e., well above the Nyquist frequency), we suspected that the poor performance with these parameters resulted from undersampling the pattern. We did find that increasing the sampling rate reduced the error rate substantially, and we believe that detecting differences in the low amplitude, high frequency fluctuations and the rapid exponential rise on the relatively high amplitude pulse requires a sampling rate well beyond the Nyquist frequency. The performance of this random process model using more than one parameter can be found in Park [6].

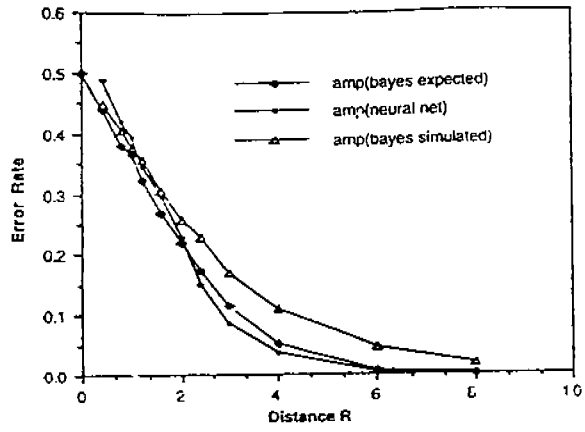


(Fig. 5) Error rates as a function of the generalized distance R for the two-class problem with the damped oscillatory pulse model using one distinguishing parameter.

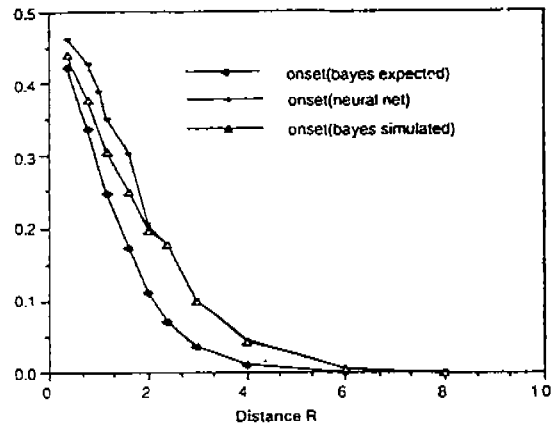
Figure 6 shows the error rate for the two-class problem with various combinations of distinguishing



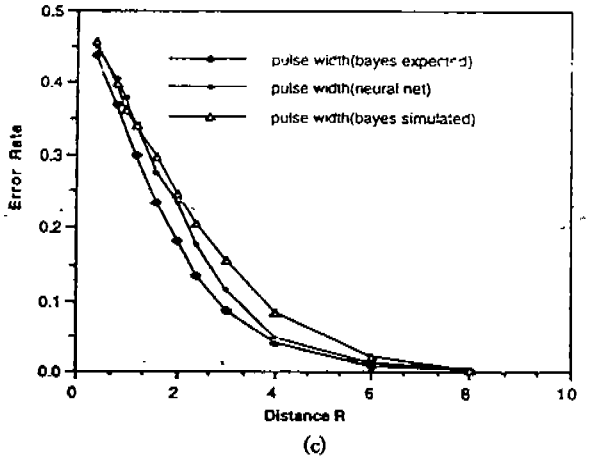
(Fig. 6) Error rates as a function of the generalized distance R for the two-class problem with the sum-of-sinusoids pulse for one and two distinguishing parameters. These curves were similar to those seen with the rectangular-pulse (Fig. 4).



(a)



(b)



(c)

(Fig. 7) Comparison of error rates of neural net and Bayes classifiers for the two-class problem with the rectangular-pulse model using (a) amplitude, (b) starting time, and (c) pulse-width as the distinguishing parameter. In general the error rate for the ANN classifier fell in between the expected and observed error rates for the Bayes classifier.

parameters for the sum-of-sinusoids pulse model. The curves in this figure were very similar to those for the rectangular pulse model (Fig. 4).

In a simulation study we used Equation 6 to evaluate the performance of the Bayes classifier using the rectangular-pulse test data with amplitude, on-set time, and pulse-width as single distinguishing parameters. Figure 7 shows these results, which we designate as the simulated Bayes performance. This figure also shows the performance curves for the corresponding neural network classifier along with those obtained from Equations 13 and 14 with amplitude as the distinguishing parameter and from the corresponding equations in Park [6] with on-set time and pulse-width as the distinguishing parameters, which we designate as the theoretical Bayes performance.

5. Conclusion

This paper describes a methodology for studying the relationship between the performance of neural network classifiers and the difficulty of the classification problem. The method used patterns that have two independently controlled sources of variability. The first source provided random variation in the underlying shape of the pattern; while the second provided random variation in its measurement.

With each of our models we found that error rates appeared to depend on the Mahalanobis distance between the parameter distributions, regardless of which model parameters were used to create this distance. Since the Mahalanobis distance reflects the overall statistical difference, this finding suggested that in real pattern classification problems the statistical differences between the processes generating the patterns may determine the performances regardless of how this difference is exhibited.

In this study, we found similar performance of neural network and Bayes classifiers; more specifically we found that the performance of the neural network was intermediate between that of the theoretical and the simulated Bayes classifiers. We believe that the differences between the theoretical and the simulated Bayes classifier may be attributed to nongaussian distributions for the sampled values of the patterns. Other groups have reported similar findings. For example, Huang and Lippmann [4] reported that with a single random variable as input the Bayes and neural network classifiers had similar error rates with gaussian distributions and with several nongaussian distributions; however, with nongaussian distributions with long tails the neural network had lower error rates. In a terrain classification application, Decatur [2] had lower error rates with the neural network than with the Bayes classifier, and he attributed this difference to nongaussian distributions. Weiss reported similar findings with four sets of real data, including Fisher's iris data [10]. Kohonen and associates [5] used a multidimensional random vector as input. With two elements they found identical performance with the Bayes and neural network classifiers. How-

ever, as the number of elements in the vector increased, the error rate for the neural network became increasing higher than that for the Bayes classifier. Peterson and Hartman [7] performed similar studies, but they did not find the difference in the performance observed by Kohonen's group. They attributed this difference to the use of binary encoded inputs, Manhatan learning, and full feedforward networks.

In summary, this report describes a methodology for studying how the difficulty of the pattern classification problem effects the performance of neural network classifiers. Simulation studies using this method suggest that generalized statistical difference between the processes generating the patterns provided a good predictor of the difficulty of the classification problem. In turn, this suggest that in experimental studies where the models are unknown, some measure of the overall statistical difference between the patterns(e.g., the trace of the pattern covariance matrix) may provide a predictor of problem difficulty.

REFERENCES

[1] S.C. Ahalt, F. D. Garber, I Jouny and A.K. Krishnammurthy, "Performance of Synthetic Neural Network Classification of Noisy Radar Signals," in D. Touretzky(ed.), *Advances in Neural Information Processing Systems*, Morgan Kaufmann Publisher, pp. 281-288, 1989.

[2] S.E. Decauter, "Application if Neural Networks to Terrain Classification," *Proceedings of International Joint Conference on Neural Networks*, vol. 1, pp. 283-288, 1989.

[3] R.P. Gorman and T.J. Sejinowski, "Learned Classification of Sonar Targets Using a Massively Parallel Network," *IEEE Transactions on Acoustic, Speech, and Signal Processing*. vol. 36, pp. 1135-1140, 1988.

[4] W.Y Huang and R.L Lippmann, "Comparison Between Neural Net and Conventional Benchmarking Studies," *Proceedings of the IEEE First International Conference on Neural Networks*, vol. 4, pp. 485-493, 1987.

[5] T.Kohonen, G.Barna, and R.Chrisley, "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies," *Proceedings of the IEEE Second International Conference on Neural Networks*. Vol 1, pp. 61-68, 1988.

[6] D.S. Park "Relationship between the Performance of Multilayer Neural Networks Pattern Classifiers and the Statistics of pattern Generating Processes," Ph. D. Dissertation, Department of Electrical and Computer Engineering, University of Missouri-Columbia, 1990.

[7] C. Peterson and E. Hartman, "Explorations of the Mean Field Theory of Learning Algorithm," *Neural Networks*, vol. 2, pp. 475-492, 1989.

[8] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing*, vol. 1, MIT Press, 1986.

[9] M.D. Srinath and P.K. Rajasekaran, *An Introduction to Statistical Signal Processing With Applications*, John Wiley and Sons, 1979.

[10] S.M. Weiss and I. Kapouleas, "An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods," *Proceedings of the Eleventh International Conference on Artificial Intelligence*, vol. 1, pp. 781-787, 1989.

박 동 선

1979년 고려대학교 전자공학과(학사)

1984년 미주리대 전기 및 컴퓨터공학과(석사)

1990년 미주리대 전기 및 컴퓨터공학과(박사)

1991년 3월~현재 전북대학교 정보통신공학과 조교수

관심분야:패턴인식, 인공지능, 멀티미디어 통신, 알고리즘