Neural Networks, Vol. 3, pp. 109–118, 1990
Printed in the USA. All rights reserved.

0893-6080/90 $3.00 + .00

ORIGINAL CONTRIBUTION

Probabilistic Neural Networks

DONALD F. SPECHT

Lockheed Missiles & Space Company, Inc.

(*Received* 5 *August* 1988; *revised and accepted* 14 *June* 1989)

Abstract—*By replacing the sigmoid activation function often used in neural networks with an exponential function, a probabilistic neural network (PNN) that can compute nonlinear decision boundaries which approach the Bayes optimal is formed. Alternate activation functions having similar properties are also discussed. A four-layer neural network of the type proposed can map any input pattern to any number of classifications. The decision boundaries can be modified in real-time using new data as they become available, and can be implemented using artificial hardware "neurons" that operate entirely in parallel. Provision is also made for estimating the probability and reliability of a classification as well as making the decision. The technique offers a tremendous speed advantage for problems in which the incremental adaptation time of back propagation is a significant fraction of the total computation time. For one application, the PNN paradigm was 200,000 times faster than back-propagation.*

Keywords—Neural network, Probability density function, Parallel processor, "Neuron", Pattern recognition, Parzen window, Bayes strategy, Associative memory.

MOTIVATION

Neural networks are frequently employed to classify patterns based on learning from examples. Different neural network paradigms employ different learning rules, but all in some way determine pattern statistics from a set of training samples and then classify new patterns on the basis of these statistics.

Current methods such as back propagation (Rumelhart, McClelland, & the PDP Research Group, 1986, chap. 8) use heuristic approaches to discover the underlying class statistics. The heuristic approaches usually involve many small modifications to the system parameters that gradually improve system performance. Besides requiring long computation times for training, the incremental adaptation approach of back-propagation can be shown to be susceptible to false minima. To improve upon this approach, a classification method based on established statistical principles was sought.

It will be shown that the resulting network, while similar in structure to back-propagation and differing primarily in that the sigmoid activation function is replaced by a statistically derived one, has the unique feature that under certain easily met conditions the decision boundary implemented by the probabilistic neural network (PNN) asymptotically approaches the Bayes optimal decision surface.

To understand the basis of the PNN paradigm, it is useful to begin with a discussion of the Bayes decision strategy and nonparametric estimators of probability density functions. It will then be shown how this statistical technique maps into a feed-forward neural network structure typified by many simple processors ("neurons") that can all function in parallel.

THE BAYES STRATEGY FOR PATTERN CLASSIFICATION

An accepted norm for decision rules or strategies used to classify patterns is that they do so in a way that minimizes the "expected risk." Such strategies are called "Bayes strategies" (Mood & Graybill,

Acknowledgments: Pattern classification using eqns (1), (2), (3), and (12) of this paper was first proposed while the author was a graduate student of Professor Bernard Widrow at Stanford University in the 1960s. At that time, direct application of the technique was not practical for real-time or dedicated applications. Advances in integrated circuit technology that allow parallel computations to be addressed by custom semiconductor chips prompt reconsideration of this concept and development of the theory in terms of neural network implementations.

The current research is supported by Lockheed Missiles & Space Company, Inc., Independent Research Project RDD360 (Neural Network Technology). The author wishes to acknowledge Dr. R. C. Smithson, Manager of the Applied Physics Laboratory, for his support and encouragement, and Dr. W. A. Fisher for his helpful comments in reviewing this article.

Requests for reprints should be addressed to Dr. D. F. Specht, Lockheed Palo Alto Research Laboratory, Lockheed Missiles & Space Company, Inc., O/91-10, B/256, 3251 Hanover Street, Palo Alto, CA 94304.

1962) and can be applied to problems containing any number of categories.

Consider the two-category situation in which the state of nature θ is known to be either θ_A or θ_B. If it is desired to decide whether $\theta = \theta_A$ or $\theta = \theta_B$ based on a set of measurements represented by the p-dimensional vector $\mathbf{X}^t = [X_1 \ldots X_j \ldots X_p]$, the Bayes decision rule becomes

$$d(\mathbf{X}) = \theta_A \quad \text{if } h_A l_A f_A(\mathbf{X}) > h_B l_B f_B(\mathbf{X})$$

$$d(\mathbf{X}) = \theta_B \quad \text{if } h_A l_A f_A(\mathbf{X}) < h_B l_B f_B(\mathbf{X}) \tag{1}$$

where $f_A(\mathbf{X})$ and $f_B(\mathbf{X})$ are the probability density functions for categories A and B, respectively; l_A is the loss function associated with the decision $d(\mathbf{X}) = \theta_B$ when $\theta = \theta_A$; l_B is the loss associated with the decision $d(\mathbf{X}) = \theta_A$ when $\theta = \theta_B$ (the losses associated with correct decisions are taken to be equal to zero); h_A is the a priori probability of occurrence of patterns from category A; and $h_B = 1 - h_A$ is the a priori probability that $\theta = \theta_B$.

Thus the boundary between the region in which the Bayes decision $d(\mathbf{X}) = \theta_A$ and the region in which $d(\mathbf{X}) = \theta_B$ is given by the equation

$$f_A(\mathbf{X}) = K\, f_B(\mathbf{X}) \tag{2}$$

where

$$K = h_B l_B / h_A l_A. \tag{3}$$

In general, the two-category decision surface defined by eqn (2) can be arbitrarily complex, since there is no restriction on the densities except those conditions that all probability density functions (PDF) must satisfy, namely, that they are everywhere non-negative, that they are integrable, and that their integrals over all space equal unity. A similar decision rule can be stated for the many-category problem (Specht, 1967a).

The key to using eqn (2) is the ability to estimate PDFs based on training patterns. Often the a priori probabilities are known or can be estimated accurately, and the loss functions require subjective evaluation. However, if the probability densities of the patterns in the categories to be separated are unknown, and all that is given is a set of training patterns (training samples), then it is these samples which provide the only clue to the unknown underlying probability densities.

In his classic paper, Parzen (1962) showed that a class of PDF estimators asymptotically approaches the underlying parent density provided only that it is continuous.

CONSISTENCY OF THE DENSITY ESTIMATES

The accuracy of the decision boundaries depends on the accuracy with which the underlying PDFs are estimated. Parzen (1962) showed how one may construct a family of estimates of $f(X)$,

$$f_n(X) = \frac{1}{n\lambda} \sum_{i=1}^{n} \mathcal{W}\left(\frac{X - X_{Ai}}{\lambda}\right), \tag{4}$$

which is consistent at all points X at which the PDF is continuous. Let $X_{A1}, \ldots, X_{Ai}, \ldots, X_{An}$ be independent random variables identically distributed as a random variable X whose distribution function $F(X) = P[x \leqq X]$ is absolutely continuous. Parzen's conditions on the weighting function $\mathcal{W}(y)$ are

$$\sup_{-\infty<y<+\infty} |\mathcal{W}(y)| < \infty, \tag{5}$$

where sup indicates the supremum,

$$\int_{-\infty}^{+\infty} |\mathcal{W}(y)|\, dy < \infty, \tag{6}$$

$$\lim_{y \to \infty} |y\mathcal{W}(y)| = 0, \tag{7}$$

and

$$\int_{-\infty}^{+\infty} \mathcal{W}(y)\, dy = 1. \tag{8}$$

In eqn (4), $\lambda = \lambda(n)$ is chosen as a function of n such that

$$\lim_{n \to \infty} \lambda(n) = 0 \tag{9}$$

and

$$\lim_{n \to \infty} n\lambda(n) = \infty. \tag{10}$$

Parzen proved that the estimate $f_n(X)$ is consistent in quadratic mean in the sense that

$$E\, |f_n(X) - f(X)|^2 \longrightarrow 0 \qquad \text{as } n \longrightarrow \infty. \tag{11}$$

This definition of consistency, which says that the expected error gets smaller as the estimate is based on a larger data set, is particularly important since it means that the true distribution will be approached in a smooth manner.

Murthy (1965, 1966) relaxed the assumption of absolute continuity of the distribution $F(X)$, and showed that the class of estimators $f_n(X)$ still consistently estimate the density at all points of continuity of the distribution $F(X)$ where the density $f(X)$ is also continuous.

Cacoullos (1966) has also extended Parzen's results to cover the multivariate case. Theorem 4.1 in Cacoullos (1966) indicates how the Parzen results can be extended to estimates in the special case that the multivariate kernel is a product of univariate kernels. In the particular case of the Gaussian kernel, the multivariate estimates can be expressed as

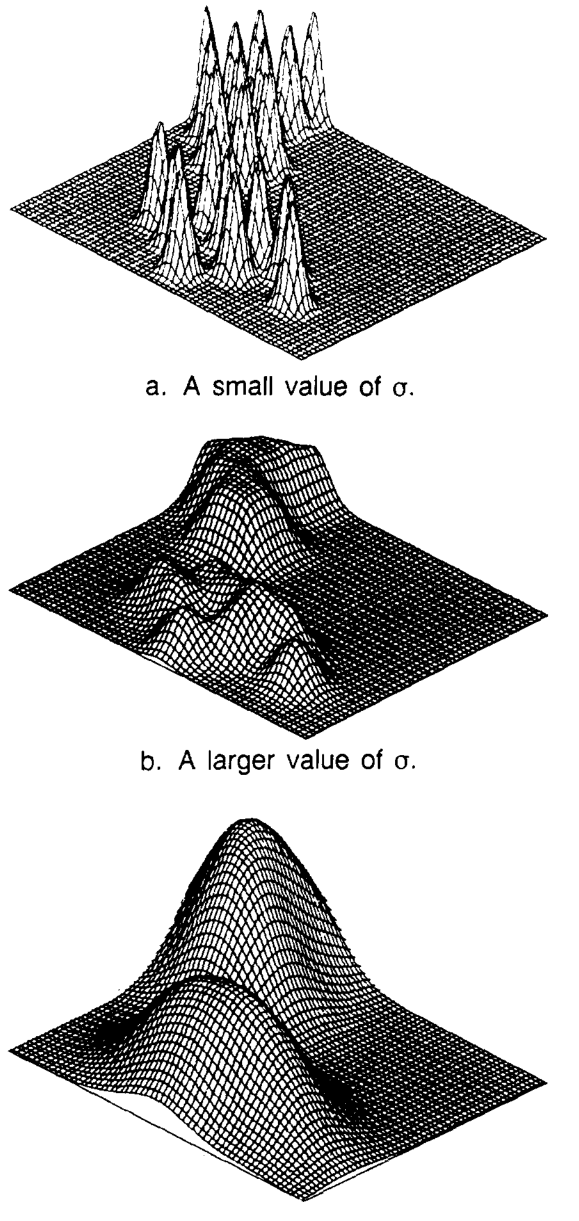
$$f_A(\mathbf{X}) = \frac{1}{(2\pi)^{p/2}\sigma^p} \frac{1}{m} \sum_{i=1}^{m} \times \exp\left[- \frac{(\mathbf{X} - \mathbf{X}_{Ai})^t (\mathbf{X} - \mathbf{X}_{Ai})}{2\sigma^2} \right] \tag{12}$$

where

i = pattern number
m = total number of training patterns
$\mathbf{X}_{Ai}$ = ith training pattern from category θ_A
σ = "smoothing parameter"
p = dimensionality of measurement space.

Note that $f_A(\mathbf{X})$ is simply the sum of small multivariate Gaussian distributions centered at each training sample. However, the sum is not limited to being Gaussian. It can, in fact, approximate any smooth density function.

Figure 1 illustrates the effect of different values for the smoothing parameter σ on $f_A(\mathbf{X})$ for the case in which the independent variable $\mathbf{X}$ is two-dimensional. The density is plotted from eqn (12) for three values of σ with the same training samples in each case. A small value of σ causes the estimated parent density function to have distinct modes corresponding to the locations of the training samples. A larger value of σ, as indicated in Figure 1b, produces a greater degree of interpolation between points. Here, values of $\mathbf{X}$ that are close to the training samples are estimated to have about the same probability of occurrence as the given samples. An even larger value of σ, as indicated in Figure 1c, produces a greater degree of interpolation. A very large value of σ would cause the estimated density to be Gaussian regardless of the true underlying distribution. Selection of the proper amount of smoothing will be discussed in the section "Limiting Conditions as $\sigma \rightarrow 0$ and as $\sigma \rightarrow \infty$."



a. A small value of σ.

b. A larger value of σ.

c. An even larger value of σ.

FIGURE 1. The smoothing effect of different values of σ on a PDF estimated from samples. From *Computer-Oriented Approaches to Pattern Recognition* (pp. 100–101) by W. S. Meisel, 1972, Orlando, FL: Academic Press. Copyright 1972 by Academic Press. Reprinted by permission.

Equation (12) can be used directly with the decision rule expressed by eqn (1). Computer programs have been written to perform pattern-recognition tasks using these equations, and excellent results have been obtained on practical problems. However, two limitations are inherent in the use of eqn (12): (a) the entire training set must be stored and used during testing, and (b) the amount of computation necessary to classify an unknown point is proportional to the size of the training set. When this approach was first proposed and used for pattern recognition (Meisel, 1972, chap. 6; Specht, 1967a 1967b), both considerations severely limited the direct use of eqn (12) in real-time or dedicated applications. Approximations had to be used instead. Computer memory has since become dense and inexpensive enough so that storing the training set is no longer an impediment, but computation time with a serial computer still is a constraint. With large-scale neural networks with massively parallel computing capability on the horizon, the second impediment to the direct use of eqn (12) will soon be lifted.

THE PROBABILISTIC NEURAL NETWORK

There is a striking similarity between parallel analog networks that classify patterns using nonparametric estimators of a PDF and feed-forward neural networks used with other training algorithms (Specht, 1988). Figure 2 shows a neural network organization for classification of input patterns $\mathbf{X}$ into two categories.

In Figure 2, the input units are merely distribution units that supply the same input values to all of the pattern units. Each pattern unit (shown in more detail in Figure 3) forms a dot product of the input pattern vector $\mathbf{X}$ with a weight vector $\mathbf{W}_i$, $Z_i = \mathbf{X} \cdot \mathbf{W}_i$, and then performs a nonlinear operation on Z_i before outputting its activation level to the summa-

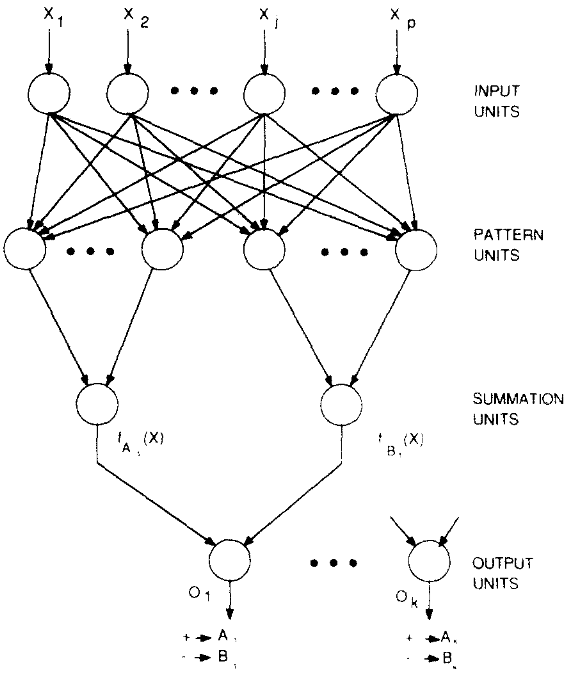


FIGURE 2. Organization for classification of patterns into categories. From "Probabilistic Neural Networks for Classification, Mapping, or Associative Memory" by D. F. Specht, 1988, *Proceedings, IEEE International Conference on Neural Networks*, 1, p. 528. Copyright 1988 by IEEE. Reprinted by permission.

tion unit. Instead of the sigmoid activation function commonly used for back-propagation (Rumelhart et al., 1986), the nonlinear operation used here is $\exp[(Z_i - 1)/\sigma^2]$. Assuming that both $\mathbf{X}$ and $\mathbf{W}_i$ are normalized to unit length, this is equivalent to using

$$\exp[-(\mathbf{W}_i - \mathbf{X})^t(\mathbf{W}_i - \mathbf{X})/2\sigma^2]$$

which is the same form as eqn (12). Thus, the dot product, which is accomplished naturally in the interconnections, is followed by the neuron activation function (the exponentiation).

The summation units simply sum the inputs from the pattern units that correspond to the category from which the training pattern was selected.

The output, or decision, units are two-input neurons as shown in Figure 4. These units produce binary outputs. They have only a single variable weight, C_k,

$$C_k = -\frac{h_{B_k} l_{B_k}}{h_{A_k} l_{A_k}} \cdot \frac{n_{A_k}}{n_{B_k}} \tag{13}$$

where

n_{A_k} = number of training patterns from category A_k
n_{B_k} = number of training patterns from category B_k

Note that C_k is the ratio of a priori probabilities, divided by the ratio of samples and multiplied by the ratio of losses. In any problem in which the numbers of training samples from categories A and B are ob-

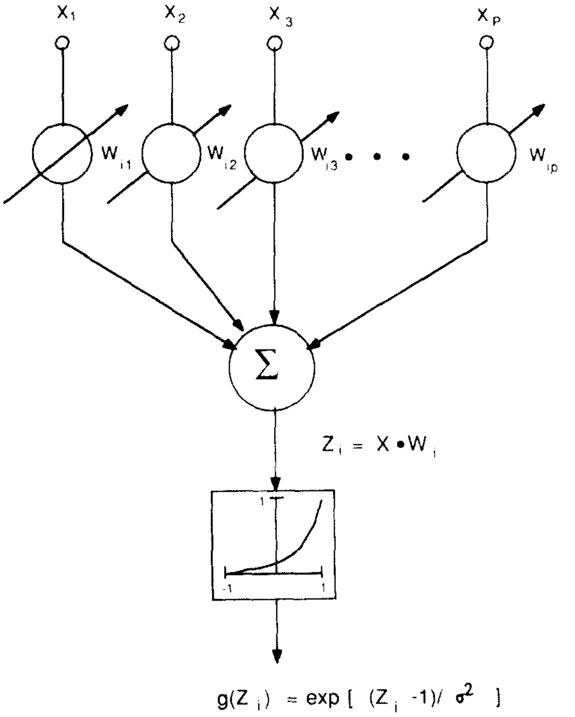


FIGURE 3. The pattern unit. From "Probabilistic Neural Networks for Classification, Mapping, or Associative Memory" by D. F. Specht, 1988, *Proceedings, IEEE International Conference on Neural Networks*, 1, p. 528. Copyright 1988 by IEEE. Reprinted by permission.

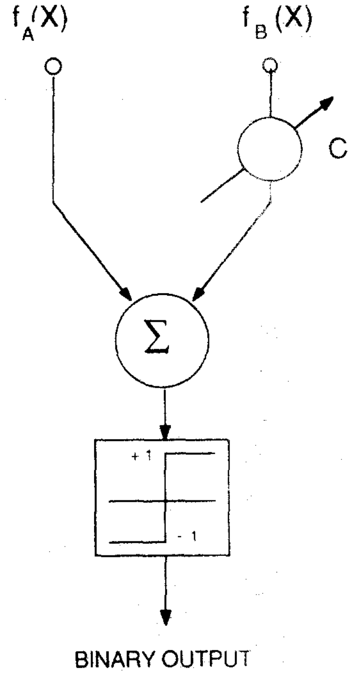


FIGURE 4. An output unit. From "Probabilistic Neural Networks for Classification, Mapping, or Associative Memory" by D. F. Specht, 1988, *Proceedings, IEEE International Conference on Neural Networks*, 1, p. 528. Copyright 1988 by IEEE. Reprinted by permission.

tained in proportion to their a priori probabilities, $C_k = -I_{B_k}/I_{A_k}$. This final ratio cannot be determined from the statistics of the training samples, but only from the significance of the decision. If there is no particular reason for biasing the decision, C_k may simplify to -1 (an inverter).

The network is trained by setting the $\mathbf{W}_i$ weight vector in one of the pattern units equal to each of the $\mathbf{X}$ patterns in the training set and then connecting the pattern unit's output to the appropriate summation unit. A separate neuron (pattern unit) is required for every training pattern. As indicated in Figure 2, the same pattern units can be grouped by different summation units to provide additional pairs of categories and additional bits of information in the output vector.

ALTERNATE ACTIVATION FUNCTIONS

Although eqn (12) has been used in all the experimental work so far, it is not the only consistent estimator that could be used. Alternate estimators suggested by Cacoullos (1966) and Parzen (1962) are given in Table 1, where

$$f_A(\mathbf{X}) = \frac{1}{n\lambda^p} K_p \sum_{i=1}^{n} \mathcal{W}(y) \tag{14}$$

$$y = \frac{1}{\lambda} \sqrt{\sum_{j=1}^{p} (\mathbf{X}_j - \mathbf{X}_{A_{ij}})^2} \tag{15}$$

and K_p is a constant such that

$$\int K_p \mathcal{W}(y)\, dy = 1. \tag{16}$$

$Z_i = \mathbf{X} \cdot \mathbf{W}_i$ as before.

When $\mathbf{X}$ and $\mathbf{W}_i$ are both normalized to unit length, Z_i ranges from -1 to $+1$, and the activation function is of the form shown in Table 1. Note that here all of the estimators can be expressed as a dot product feeding into an activation function because all involve $y = 1/\lambda \sqrt{2 - 2\mathbf{X} \cdot \mathbf{X}_{A_i}}$. Non-dot product forms will be discussed later.

All the Parzen windows shown in Table 1, in conjunction with the Bayes decision rule of eqn (1), would result in decision surfaces that are asymptotically Bayes optimal. The only difference in the corresponding neural networks would be the form of the nonlinear activation function in the pattern unit. This leads one to suspect that the exact form of the activation function is not critical to the usefulness of the network. The common elements in all the networks are that: the activation function takes its maximum value at $Z_i = 1$ or maximum similarity between the input pattern $\mathbf{X}$ and the pattern stored in the pattern unit; the activation function decreases as the pattern becomes less similar; and the entire curve should be compressed towards the $Z_i = 1$ line as the number of training patterns, n, is increased.

LIMITING CONDITIONS AS $\sigma \to 0$ AND AS $\sigma \to \infty$

It has been shown (Specht, 1967a) that the decision boundary defined by eqn (2) varies continuously from a hyperplane when $\sigma \to \infty$ to a very nonlinear boundary representing the nearest neighbor classifier when $\sigma \to 0$. The nearest neighbor decision rule has been investigated in detail by Cover and Hart (1967).

In general, neither limiting case provides optimal separation of the two distributions. A degree of averaging of nearest neighbors, dictated by the density of training samples, provides better generalization than basing the decision on a single nearest neighbor. The network proposed is similar in effect to the k-nearest neighbor classifier.

Specht (1966) contains an involved discussion of how one should choose a value of the smoothing parameter, σ, as a function of the dimension of the problem, p, and the number of training patterns, n. However, it has been found that in practical problems it is not difficult to find a good value of σ, and that the misclassification rate does not change dramatically with small changes in σ.

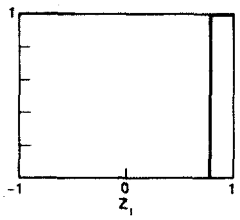
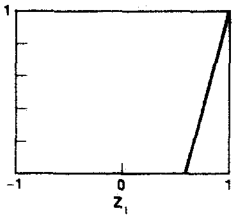
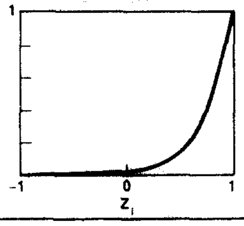
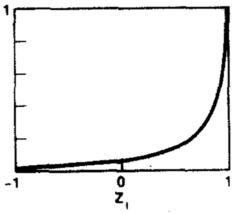
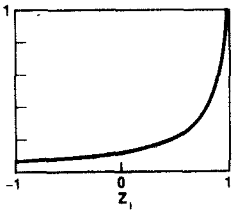
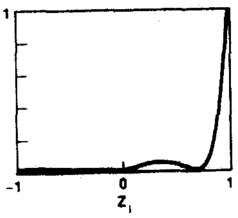
Specht (1967b) describes an experiment in which electrocardiograms were classified as normal or abnormal using the two-category classification of eqns (1) and (12). In that case, 249 patterns were available for training and 63 independent cases were available for testing. Each pattern was described by a 46-dimensional pattern vector (but not normalized to unit length). Figure 5 shows the percentage of testing samples classified correctly versus the value of the smoothing parameter, σ. Several important conclusions are obvious. Peak diagnostic accuracy can be obtained with any σ between 4 and 6; the peak of the curve is sufficiently broad that finding a good value of σ experimentally is not difficult. Furthermore, any σ in the range from 3 to 10 yields results only slightly poorer than those for the best value. It turned out that all values of σ from 0 to ∞ gave results that were significantly better than those of cardiologists on the same testing set.

The only parameter to be tweaked in the proposed system is the smoothing parameter, σ. Because it controls the scale factor of the exponential activation function, its value should be the same for every pattern unit.

AN ASSOCIATIVE MEMORY

In the human thinking process, knowledge accumulated for one purpose is often used in different ways for different purposes. Similarly, in this situa-

TABLE 1
Parzen Weighting Functions and Their Equivalent Neural Network Activation Functions

W (y)	ACTIVATION FUNCTION
$1, y \leq 1$ $0, y \geq 1$	
$1 - y, \; y \leq 1$ $0, \quad y \geq 1$	
$e^{-1/2\, y^2}$	
$e^{-\lvert y \rvert}$	
$\frac{1}{1+y^2}$	
$\left(\frac{\sin(y/2)}{y/2}\right)^2$	

tion, if the decision category were known, but not all the input variables, then the known input variables could be impressed on the network for the correct category and the unknown input variables could be varied to maximize the output of the network. These values represent those most likely to be associated with the known inputs. If only one parameter were unknown, then the most probable value of that parameter could be found by ramping though all possible values of the parameter and choosing the one that maximized the PDF. If several parameters are unknown, this method may be impractical. In this case, one might be satisfied with finding the closest mode of the PDF. This goal could be achieved using the method of steepest ascent.

A more general approach to forming an associative memory is to avoid distinguishing between inputs and outputs. By concatenating the **X** vector and the

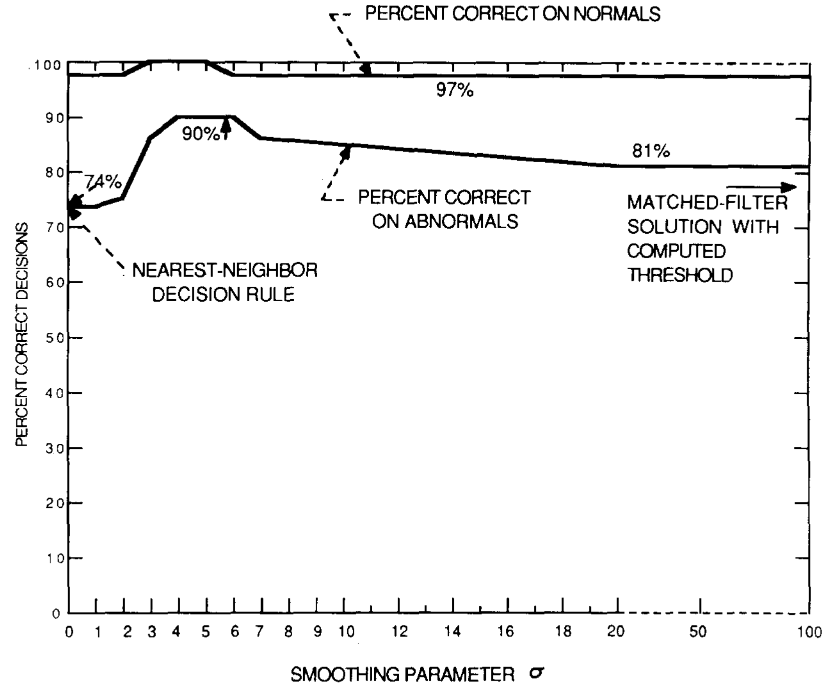


FIGURE 5. Percentage of testing samples classified correctly versus smoothing parameter σ. From "Vectorcardiographic Diagnosis Using the Polynomial Discriminant Method of Pattern Recognition" by D. F. Specht, 1967, *IEEE Transactions on Bio-Medical Engineering*, 14, 94. Copyright 1967 by IEEE. Reprinted by permission.

output vector into one longer measurement vector $\mathbf{X}'$, a single probabilistic network can be used to find the global PDF, $f(\mathbf{X}')$. This PDF may have many modes clustered at various locations on the hypersphere. To use this network as an associative memory, one impresses on the inputs of the network those parameters that are known, and allows the other parameters to relax to whatever combination maximizes $f(\mathbf{X}')$, which occurs at the nearest mode.

SPEED ADVANTAGE RELATIVE TO BACK PROPAGATION

One of the principle advantages of the PNN paradigm is that it is very much faster than the well-known back propagation paradigm (Rumelhart, 1986, chap. 8) for problems in which the incremental adaptation time of back propagation is a significant fraction of the total computation time. In a hull-to-emitter correlation problem supplied by the Naval Ocean Systems Center (NOSC), the PNN accurately identified hulls from difficult, nonlinear boundary, multiregion, and overlapping emitter report parameter data sets.

Marchette and Priebe (1987) provide a description of the problem and the results of classification using back-propagation and conventional techniques. Maloney (1988) describes the results of using PNN on the same data base.

The data set consisted of 113 emitter reports of three continuous input parameters each. The output layer consisted of six binary outputs indicating six possible hull classifications. This data set was small, but as in many practical problems, more data were either not available or expensive to obtain. To make the most use of the data available, both groups decided to hold out one report, train a network on the other 112 reports, and use the trained network to classify the holdout pattern. This process was repeated with each of the 113 reports in turn. Marchette and Priebe (1987) estimated that to perform the experiment as planned would take in excess of 3 weeks of continuous running time on a Digital Equipment Corp. VAX 8650. Because they didn't have that much VAX time available, they reduced the number of hidden units until the computation could be performed over the weekend. Maloney (1988), on the other hand, used a version of PNN on an IBM PC/AT (8 MHz) and ran all 113 networks in 9 seconds (most of which was spent writing results on the screen). Not taking into account the I/O overhead or the higher speed of the VAX, this amounts to a speed improvement of 200,000 to 1!

Classification accuracy was roughly comparable. Back-propagation produced 82% accuracy whereas PNN produced 85% accuracy (the data distributions overlap such that 90% is the best accuracy that NOSC ever achieved using a carefully crafted special purpose classifier). It is assumed that back propagation would have achieved about the same accuracy as PNN if allowed to run 3 weeks. By breaking the problem into subproblems classified by separate

PNN networks, Maloney reported increasing the PNN classification accuracy to 89%.

The author has since run PNN on the same database using a PC/AT 386 with a 20 MHz clock. By reducing the displayed output to a summary of the classification results of the 113 networks, the time required was 0.7 seconds to replicate the original 85% accuracy. Compared with back-propagation running over the weekend which resulted in 82% accuracy, this result again represents a speed improvement of 200,000 to 1 with slightly superior accuracy.

PNN NOT LIMITED TO MAKING DECISIONS

The outputs $f_A(\mathbf{X})$ and $f_B(\mathbf{X})$ can also be used to estimate a posteriori probabilities or for other purposes beyond the binary decisions of the output units. The most important use we have found is to estimate the a posteriori probability that $\mathbf{X}$ belongs to category A, $P[A|\mathbf{X}]$. If categories A and B are mutually exclusive and if $h_A + h_B = 1$, we have from the Bayes theorem

$$P[A|\mathbf{X}] = \frac{h_A f_A(\mathbf{X})}{h_A f_A(\mathbf{X}) + h_B f_B(\mathbf{X})} \tag{17}$$

Also, the maximum of $f_A(\mathbf{X})$ and $f_B(\mathbf{X})$ is a measure of the density of training samples in the vicinity of $\mathbf{X}$, and can be used to indicate the reliability of the binary decision.

PROBABILISTIC NEURAL NETWORKS USING ALTERNATE ESTIMATORS OF $f(\mathbf{X})$

The earlier discussion dealt only with multivariate estimators that reduced to a dot product form. Further application of Cacoullos (1966), Theorem 4.1, to other univariate kernels suggested by Parzen (1962) yields the following multivariate estimators (which are products of univariate kernels):

$$f_A(\mathbf{X}) = \frac{1}{n(2\lambda)^p}\sum_{i=1}^{n} 1, \qquad \text{when all } |X_j - X_{Aij}| \le \lambda \tag{18}$$

$$f_A(\mathbf{X}) = \frac{1}{n\lambda^p}\sum_{i=1}^{n}\prod_{j=1}^{p}\left[1 - \frac{|X_j - X_{Aij}|}{\lambda}\right], \qquad \text{when all } |X_j - X_{Aij}| \le \lambda \tag{19}$$

$$\begin{aligned} f_A(\mathbf{X}) &= \frac{1}{n(2\pi)^{p/2}\lambda^p}\sum_{i=1}^{n}\prod_{j=1}^{p} e^{-\frac{1}{2}\frac{(X_j - X_{Aij})^2}{\lambda^2}} \\ &= \frac{1}{n(2\pi)^{p/2}\lambda^p}\sum_{i=1}^{n}\exp\left[\frac{-\sum_{j=1}^{p}(X_j - X_{Aij})^2}{2\lambda^2}\right] \end{aligned} \tag{20}$$

$$\begin{aligned} f_A(\mathbf{X}) &= \frac{1}{n(2\lambda)^p}\sum_{i=1}^{n}\prod_{j=1}^{p} e^{-|X_j - X_{Aij}|/\lambda} \\ &= \frac{1}{n(2\lambda)^p}\sum_{i=1}^{n}\exp\left[-\frac{1}{\lambda}\sum_{j=1}^{p}|X_j - X_{Aij}|\right] \end{aligned} \tag{21}$$

$$f_A(\mathbf{X}) = \frac{1}{n(\pi\lambda)^p}\sum_{i=1}^{n}\prod_{j=1}^{p}\left[1 + \frac{(X_j - X_{Aij})^2}{\lambda^2}\right]^{-1} \tag{22}$$

$$f_A(\mathbf{X}) = \frac{1}{n(2\pi\lambda)^p}\sum_{i=1}^{n}\prod_{j=1}^{p}\left[\frac{\sin\dfrac{(X_j - X_{Aij})}{2\lambda}}{\dfrac{X_j - X_{Aij}}{2\lambda}}\right]^2 . \tag{23}$$

Equation (20) is simply an alternate form of the dot product estimator of eqn (12). The forms that do not reduce to a dot product would require an alternate network structure. They all can be implemented computationally as is.

It has not been proven that any of these estimators is the best and should always be used. Since all the estimators converge to the correct underlying distribution, the choice can be made on the basis of computational simplicity or similarity to computational models of biological neural networks. Of these, eqn (21) (in conjunction with eqns (1) through (3)) is particularly attractive from the point of view of computational simplicity.

When the measurement vector $\mathbf{X}$ is restricted to binary measurements, eqn (21) reduces to finding the Hamming distance between the input vector and a stored vector followed by use of the exponential activation function.

One final and very useful variation now suggests itself. If the input variables are expressed in binary ($+1$ or -1) form, all input vectors automatically have the same length, $\sqrt{p}$, and do not have to be normalized. These patterns can again be used with the network of Figures 2 through 4. In this case, the range of Z_i is $+p$ to $-p$. This change can be accommodated by a small change in the activation function $g(Z_i) = \exp[(Z_i - p)/p\,\sigma^2]$.

The variations in the shape of the activation function indicated in Table 1 still are allowed without relinquishing the basic attribute of the network of asymptotic Bayes optimality.

Even when the input measurements are inherently continuous, it may be desirable to convert them to a binary representation because certain technologies that might be used for massively parallel hardware lend themselves to computation of Hamming distances. Continuous measurements can be expressed in binary form by a coding scheme sometimes called the "thermometer code," in which each feature is represented by an n bit binary code that is a series of $+1$'s followed by a series of -1's (Widrow et al., 1963). The value of the feature is represented by the sum of the +1's. This seemingly inefficient code has the following advantages:

1. The absolute value of the difference between the feature value of a stored training vector and the feature value of the pattern to be classified can be measured by the Hamming distance.
2. The entire sum over p features as required in eqn (21) can be handled by one large Hamming distance calculation over a feature vector that is n times p bits long.

DISCUSSION

Operationally, the most important advantage of the probabilistic neural network is that training is easy and instantaneous. It can be used in real-time because as soon as one pattern representing each category has been observed, the network can begin to generalize to new patterns. As additional patterns are observed and stored into the network, the generalization will improve and the decision boundary can become more complex.

Other advantages of the PNN are: (a) The shape of the decision surfaces can be made as complex as necessary, or as simple as desired, by choosing the appropriate value of the smoothing parameter σ; (b) The decision surfaces can approach Bayes optimal; (c) Erroneous samples are tolerated; (d) Sparse samples are adequate for network performance; (e) σ can be made smaller as n gets larger without retraining; (f) For time-varying statistics, old patterns can be overwritten with new patterns.

Another practical advantage of the proposed network is that, unlike many networks, it operates completely in parallel without a need for feedback from the individual neurons to the inputs. For systems involving thousands of neurons (too many to fit into a single semiconductor chip), such feedback paths would quickly exceed the number of pins available on a chip. However, with the proposed network, any number of chips could be connected in parallel to the same inputs if only the partial sums from the summation units are run off-chip. There would be only two such partial sums per output bit.

It has been shown that the exact form of the activation function is not critical to the effectiveness of the network. This fact will be important in the design of analog or hybrid neural networks in which the activation function is implemented with analog components.

The probabilistic neural network proposed here can, with variations, be used for mapping, classification, associative memory, or to directly estimate a posteriori probabilities.

REFERENCES

Cacoullos, T. (1966). Estimation of a multivariate density. *Annals of the Institute of Statistical Mathematics (Tokyo)*, **18**(2), 179–189.

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, **IT-13**, 21–27.

Maloney, P. S. (1988, October). *An application of probabilistic neural networks to a hull-to-emitter correlation problem.* Paper presented at the 6th Annual Intelligence Community AI Symposium, Washington, DC.

Marchette, D., & Priebe, C. (1987). An application of neural networks to a data fusion problem. *Proceedings, 1987 Tri-Service Data Fusion Symposium* **1**, 230–235.

Meisel, W. S. (1972). *Computer-oriented approaches to pattern recognition.* New York: Academic Press.

Mood, A. M., & Graybill, F. A. (1962). *Introduction to the theory of statistics.* New York: Macmillan.

Murthy, V. K. (1965). Estimation of probability density. *Annals of Mathematical Statistics*, **36**, 1027–1031.

Murthy, V. K. (1966). Nonparametric estimation of multivariate densities with applications. In P. R. Krishnaiah (Ed.), *Multivariate analysis* (pp. 43–58). New York: Academic Press.

Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, **33**, 1065–1076.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (1986). *Parallel distributed processing, Volume 1: Foundations.* Cambridge, MA: The MIT Press.

Specht, D. F. (1967a). Generation of polynomial discriminant functions for pattern recognition. *IEEE Transactions on Electronic Computers*, **EC-16**, 308–319.

Specht, D. F. (1967b). Vectorcardiographic diagnosis using the polynomial discriminant method of pattern recognition. *IEEE Transactions on Bio-Medical Engineering*, **BME-14**, 90–95.

Specht, D. F. (1966). *Generation of polynomial discriminant functions for pattern recognition.* Ph.D. dissertation, Stanford University. Also available as report SU-SEL-66-029. Stanford Electronics Laboratories.

Specht, D. F. (1988). Probabilistic neural networks for classification mapping, or associative memory. *Proceedings, IEEE International Conference on Neural Networks*, **1**, 525–532.

Widrow, B., Groner, G. F., Hu, M. J. C., Smith, F. W., Specht, D. F., & Talbert, L. R. (1963). Practical applications for adaptive data-processing systems. *1963 WESCON convention record*, 11.4.

NOMENCLATURE

C_k	weight of output unit for decision number k
$d(\mathbf{X})$	decision on pattern $\mathbf{X}$
$f(\mathbf{X})$	probability density function (PDF) of the random vector $\mathbf{X}$
$f_R(\mathbf{X})$	probability density function estimated from a set of samples taken from category R, where $R = A$ or B
$f_{R_k}(\mathbf{X})$	probability density function estimated from a set of samples taken from category R_k, where $R = A$ or B and k = decision number
h_R	a priori probability of a sample belonging to category R
k	decision number (used for multiple output bits)
K	the ratio $h_B l_B / h_A l_A$
l_R	loss associated with the decision $d(\mathbf{X}) \neq \theta_R$ when θ_R is the Rth state of nature
l_{R_k}	loss associated with the decision $d(\mathbf{X}) \neq \theta_{R_k}$ when θ_{R_k} is the R_kth state of nature

m	number of training patterns
n_R	number of training patterns from category R
n_{R_k}	number of training patterns from category R_k
$P[R\|\mathbf{X}]$	probability of R given $\mathbf{X}$
p	dimension of the pattern vectors
R	category (A or B)
$\mathbf{W}$	weight vector (p-dimensional)
$\mathbf{X}$	pattern vector (p-dimensional)
X_j	jth component of $\mathbf{X}$
$\mathbf{X}'$	transpose of $\mathbf{X}$, $\mathbf{X}' = [X_1 \ldots X_j \ldots X_p]$
$\mathbf{X}_{Ri}$	ith training pattern vector from category R
X_{Rij}	jth component of $\mathbf{X}_{Ri}$
$\mathcal{W}(y)$	weighting factor
Z_i	dot product of $\mathbf{X}$ with weight vector $\mathbf{W}_i$
θ	state of nature
θ_R	Rth state of nature; the Rth category (for the two-category case, $R = A$ or B)
λ	a parameter of the weighting function $\mathcal{W}(y)$
σ	"smoothing parameter" of the probability density function estimator