PROBABALISTIC NEURAL NETWORKS FOR CLASSIFICATION, MAPPING, OR ASSOCIATIVE MEMORY

Donald F. Specht
Lockheed Palo Alto Research Laboratories
3251 Hanover St.
Palo Alto, California 94304

Abstract

It can be shown that by replacing the Sigmoid activation function often used in neural networks with an exponential function, a neural network can be formed which computes nonlinear decision boundaries. This technique yields decision surfaces which approach the Bayes optimal under certain conditions. There is a continuous control of the linearity of the decision boundaries -- from linear for small training sets to any degree of nonlinearity justified by larger training sets.

A four-layer neural network of the type proposed can map any input pattern to any number of classifications. The input variables can be either continuous or binary. Modification of the decision boundaries based on new data can be accomplished in real-time simply by defining a set of weights equal to the new training vector.

The decision boundaries can be implemented using analog "neurons" which operate entirely in parallel. The organization proposed takes into account the pin limitations of neural net chips of the future. A chip can contain any number of neurons which store training data, but the number of pins needed is limited to the dimension of the input vector plus the dimension of the output vector plus some few for power and control. Provision could easily be made for paralleling the outputs of more than one such chip.

By a change in architecture, these same components could be used as associative memories, to compute nonlinear multivariate regression surfaces, or to compute a posteriori probabilites of an event.

Motivation

To achieve the tremendous speed advantage promised by the parallel architecture of neural networks, actual hardware "neurons" will have to be manufactured in huge numbers. This can be accomplished by a) development of special semiconductor integrated circuits (very large scale integration or even wafer scale integration) [1], or b) development of optical computer components (making use of the inherently parallel properties of optics).

It is desirable to develop a standard component which can be used in a variety of applications. A component which estimates probability density functions (PDFs) can be used to form networks which can be used to map input patterns to output patterns, to classify patterns, to form associative memories, and to estimate probability density functions. The PDF estimator proposed imposes a minimum of restrictions on the form of the density. It can have many modes (or regions of activity).

The Bayes Strategy for Pattern Classification

An accepted norm for decision rules or strategies used to classify patterns is that they do so in such a way as to minimize the "expected risk." Such strategies are called "Bayes strategies" [2] and may 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 $X^t = [\ X_1 \ldots X_i \ldots X_p]$, the Bayes decision rule becomes

$$\begin{aligned} d(X) &= \theta_A \text{ if } h_A l_A f_A(X) > h_B l_B f_B(X) \\ d(X) &= \theta_B \text{ if } h_A l_A f_A(X) < h_B l_B f_B(X) \end{aligned} \qquad (1)$$

where $f_A(X)$ and $f_B(X)$ are the probability density functions for categories θ_A and θ_B respectively, l_A is the loss function associated with the decision $d(X) = \theta_B$ when $\theta=\theta_A$, l_B is the loss associated with the decision $d(X) = \theta_A$ 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 θ_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 $d(X)=\theta_A$ and the region in which $d(X)=\theta_B$ is given by the equation

$$f_A(X) = K\ f_B(X) \qquad (2)$$

where

$$K = h_B l_B \ / \ h_A l_A \ . \qquad (3)$$

Note that in general the two-category decision surface defined by (2) can be arbitrarily complex, since there is no restriction on the densities except those conditions which all probability density functions 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 (see reference [4]).

The key to using eq. 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 [3] showed that a class of PDF estimators asymptotically approach the underlying parent density provided that it is smooth and continuous. The particular estimator used in this study is:

$$f_A(X) = \frac{1}{(2\pi)^{p/2}\sigma^p} \frac{1}{m} \sum_{i=1}^{m} \exp\left[- \frac{(X - X_{ai})^t (X - X_{ai})}{2\sigma^2} \right], \qquad (4)$$

where i = pattern number,
X_{ai} = ith training pattern from category θ_A, and

σ is a "smoothing parameter".

It will be noted that $f_A(X)$ is simply the sum of small multivariate Gaussian distributions centered at each of the training samples. However the sum is not limited to being Gaussian. It can, in fact, approximate any smooth density function.

Figure 1 illustrates the effect of σ on $f_A(X)$ in the case in which the independent variable X is one-dimensional. The density is plotted from (4) for four values of σ with the same 5 training samples in each case. A value of $\sigma=0.1$ causes the estimated parent density function to have 5 distinct modes corresponding to the 5 training samples. $\sigma=0.2$ brings about a greater degree of interpolation between points, but the modes remain distinct. With $\sigma=0.5$, $f_A(X)$ has a single mode and a shape approximating that of the Gaussian distribution. The value of $\sigma=1.0$ causes some flattening of the density function, with spreading out of the tails.

Equation (4) can be used directly with the decision rule expressed by (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 (4). First, the entire training set must be stored and used during testing; and second, the amount of computation necessary to classify an unknown point is proportional to the size of the training set. At the time when this approach was first proposed and used for pattern recognition [4-7], both of these considerations severely limited the direct use of (4) in real-time or dedicated applications. Approximations had to be used instead. Computer memory has since become sufficiently dense and inexpensive that storage of the training set is no longer an impediment, but computation time with a serial computer still is a constraint. Now with large scale neural networks with massively parallel computing capability on the horizon, the second impediment to the direct use of (4) will soon be lifted.

The Probabalistic Neural Network

There is a striking similarity between a parallel analog network which can be used to classify patterns using nonparametric estimators of a PDF and feed-forward neural networks used with other training algorithms. Figure 2 shows a "neural network" organization for classification of input patterns **X** into 2 categories.

In Figure 2 the input units are merely distribution units which supply the same voltage values to all of the pattern units. The pattern units (shown in more detail in Fig. 3) each form a dot product of the input pattern vector **X** with a weight vector $\mathbf{W}_i$, $Z_i = \mathbf{X}\cdot\mathbf{W}_i$, and then perform a nonlinear operation on Z_i before outputting its activation level to the summation unit. Instead of the Sigmoid Activation Function commonly used for back-propagation [8], the nonlinear operation used here is $\exp[(Z_i - 1)/\sigma^2]$. Assuming that both X and W_i are normalized to unit length, this is equivalent to using

$$\exp[-(W_i - X)^t(W_i - X) / 2\sigma^2] \quad .$$

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

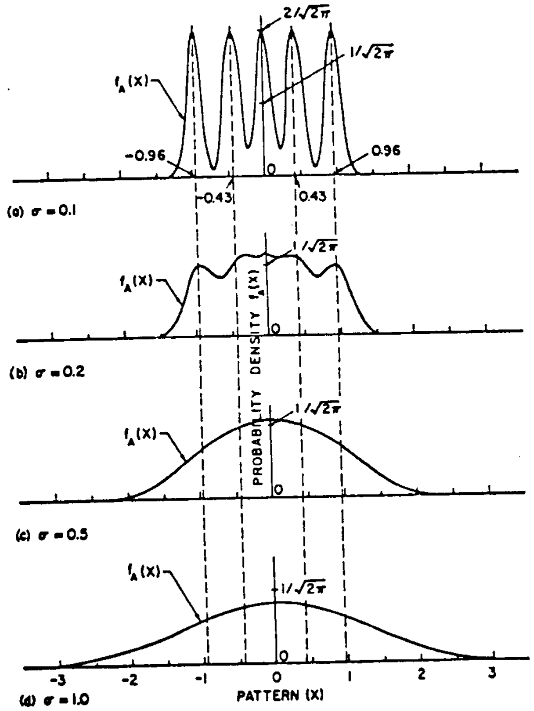
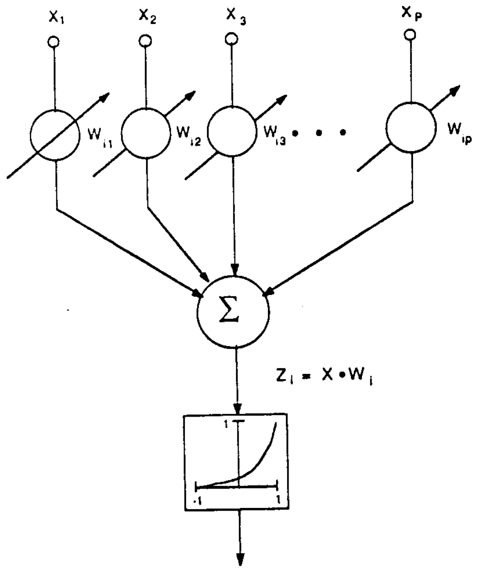


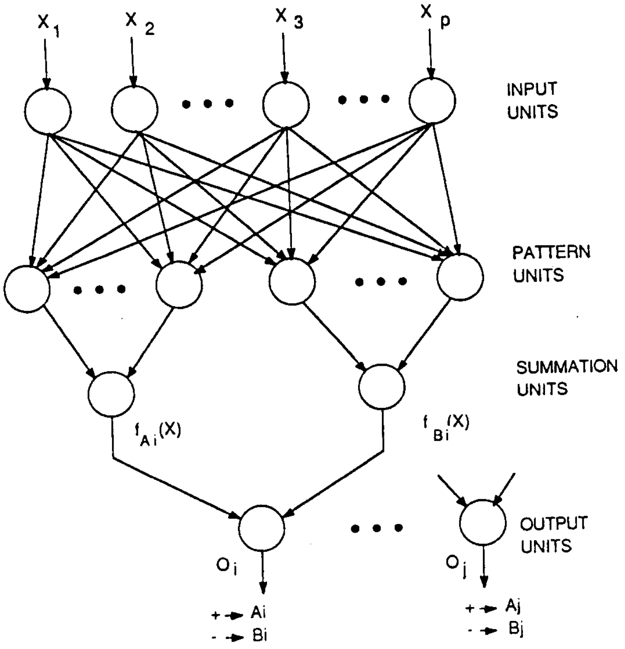
Fig. 1 The smoothing effect of σ on an estimated PDF from 5 samples.



$$g(Z_i) = \exp[(Z_i - 1)/\sigma^2]$$

Fig. 3 The Pattern Unit

Figure 2 Organization for Classification of Patterns into Categories.



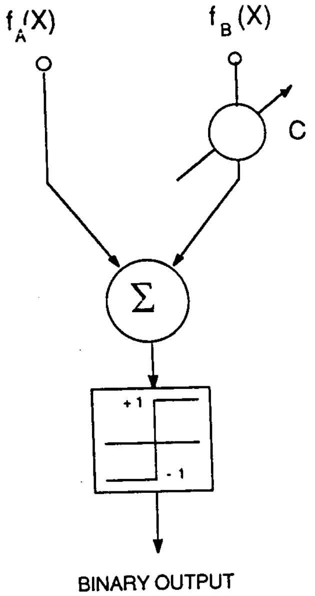


Figure 4 An Output Unit

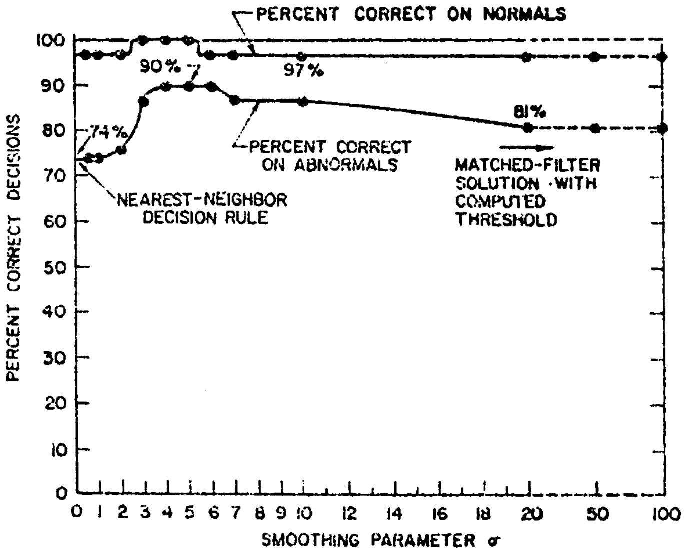


Fig. 5 Percentage of testing samples classified correctly versus smoothing parameter σ.

The decision units are 2-input neurons, as shown in Figure 4, which produce a binary output. They have only a single variable weight, C_i , where

$$C_i = - \frac{h_{Bi} l_{Bi}}{h_{Ai} l_{Ai}} \cdot \frac{n_{Ai}}{n_{Bi}} \qquad (5)$$

and

n_{Ai} = number of training patterns from category A_i ,
n_{Bi} = number of training patterns from category B_i .

Note that C_i 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 taken in proportion to their a priori probabilities, $C_i = - l_{Bi} / l_{Ai}$. 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_i may simplify to -1 (an inverter).

Training of the network is accomplished by setting each X pattern in the training set equal to the W_i weight vector in one of the pattern units, and then connecting the pattern unit's output to the appropriate summation unit. A separate neuron is required for every training pattern. As is indicated in Fig. 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.

Consistency of the Density Estimates

The accuracy of the decision boundaries is dependent on the accuracy with which the underlying PDFs are estimated. Parzen [3] and Murthy [9] have shown how one may construct a family of estimates of f(X) which include the estimator of (4) and which are consistent (asymptotically approach identity with the PDF) at all points X at which the density function is continuous, providing $\sigma=\sigma(n)$ is chosen as a function of n such that

$$\lim_{n\to\infty} \sigma(n) = 0 \quad \text{and} \quad \lim_{n\to\infty} n\sigma(n) = \infty . \qquad (6)$$

Several other functions (Parzen windows) could also be used which have these same properties and would result in decision surfaces which are also asymptotically Bayes optimal. For some of these the only difference in the network 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.

Limiting Conditions as $\sigma\to0$ and as $\sigma\to\infty$

It has been shown [4] that the decision boundary defined by equation (2) varies continuously from a hyperplane when $\sigma = \infty$ to a very nonlinear boundary representing the nearest neighbor classifier when $\sigma\to0$. The nearest neighbor decision rule has been investigated in detail by Cover and Hart [10]. Hecht-Nielsen [11] has proposed a neural network which implements this decision rule.

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.

Reference [12] 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.

Reference [5] describes an experiment in which electrocardiograms were classified as normal or abnormal using the 2-category classification of equations (1) and (4). There were, in this case, 249 patterns available for training and 63 independent cases available for testing. Each pattern was described by a 46-dimensional pattern vector (but not normalized to unity length). Figure 5 shows the percentage of testing samples classified correctly versus value of the smoothing parameter, σ. Several important conclusions are immediately 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 at all difficult. Furthermore, any σ in the range from 3 to 10 yields results only slightly poorer than those for the best value, and all values of σ from 0 to ∞ give results which are significantly better than those to be expected from classification by chance.

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 situation, if the decision category, but not all of the input variables were known, 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 which maximized the PDF. If several parameters are unknown, this may be impractical. In this case, one might be satisfied with finding the closest mode of the PDF. This could be done by the method of steepest ascent.

A more-general approach to forming an associative memory is to avoid making a distinction between inputs and outputs. By concatenating the **X** vector and the **Y** vector into one longer measurement vector **Z**, a single probabalistic network can be used to find the global PDF, f(**Z**). 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 which are known, and allows the rest of the parameters to relax to whatever combination maximizes f(**Z**), which occurs at the nearest mode.

Discussion

The most obvious advantage of this network is that training is trivial 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 net, the generalization will improve and the decision boundary can get more complex.

Other characteristics of this network are: 1) The shape of the decision surfaces can be made as complex as necessary, or as simple as desired, by proper choice of the smoothing parameter σ. 2) The decision surfaces can approach Bayes-optimal. 3) It tolerates erroneous samples. 4) It works for sparse samples. 5) It is possible to make σ smaller as n gets larger without retraining. 6) For time-varying statistics , old patterns can be overwritten with new patterns.

A 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 back to the inputs. For systems involving thousands of neurons, and if the number is too large to fit into a single 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 2 such partial sums per output bit.

The probabalistic neural network proposed here, with variations, can be used for mapping, classification, associative memory, or the direct estimation of a posteriori probabilities.

Acknowledgements and Historical Perspective

Pattern classification using the equations in 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 complex computations to be addressed by a custom chip prompt the reconsideration of this concept.

The current research is being supported by Lockheed Missiles & Space Co., Independent Research Project RDD 360, under the direction of Dr. R. C. Smithson.

References

[1] Mead, Carver, "Silicon Models of Neural Computation," Vol. I, IEEE First International Conference on Neural Networks, San Diego, CA, June, 1987, pp. 93-106.

[2] Mood, A. M. and Graybill, F. A., Introduction to the Theory of Statistics. New York: Macmillan, 1962.

[3] Parzen, E., "On estimation of a probability density function and mode," Ann. Math. Stat., Vol. 33, pp. 1065-1076, Sept. 1962.

[4] Specht, D. F., "Generation of Polynomial Discriminant Functions for Pattern Recognition," IEEE Trans. on Electronic Computers, EC-16, pp. 308-319.

[5] Specht, D. F., "Vectorcardiographic diagnosis using the polynomial discriminant method of pattern recognition," IEEE Trans. on Bio-Medical Engineering, BME-14, pp. 90-95, Apr. 1967.

[6] Huynen, J. R., Bjorn, T., and Specht, D. F., "Advanced Radar Target Discrimination Technique for Real-Time Application," Lockheed Missiles and Space Co., LMSC-D051707, Jan. 1969.

[7] Bjorn, T. and Specht, D. F., "Discrimination Between Re-entry Bodies in the Presence of Clutter Using the Polynomial Discriminant Method of Pattern Recognition," (U), Lockheed Rept. LMSC-B039970, Dec. 1967, (S).

[8] Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, Parallel Distributed Processing, Volume 1: Foundations, The MIT Press, Cambridge, Mass. and London, England, 1986.

[9] Murthy, V. K., "Estimation of probability density," Ann. Math. Stat., Vol. 36, pp. 1027-1031, June 1965.

[10] Cover, T. M., and Hart, P. E., "Nearest neighbor pattern classification," IEEE Trans. on Information Theory, IT-13, pp.21-27, Jan 1967.

[11] Hecht-Nielsen, R., "Nearest matched filter classification of spatiotemporal patterns," Applied Optics, Vol. 26, No. 10, May 1987.

[12] Specht, D. F., "Generation of Polynomial Discriminant Functions for Pattern Recognition," Stanford Electronics Labs. Rept. SU-SEL-66-029, May 1966. [available as Defense Documentation Center Rept. AD 487 537].