Estimation and Classification by Sigmoids based on Mutual Information

Yoram Baram*

August 31, 1994

Abstract

An estimate of the probability density function of a random vector is obtained by maximizing the mutual information between the input and the output of a feedforward network of sigmoidal units with respect to the input weights. Classification problems can be solved by selecting the class associated with the maximal estimated density. Newton's method, applied to an estimated density, yields a recursive maximum likelihood estimator, consisting of a single internal layer of sigmoids, for a random variable or a random sequence. Applications to the diamond classification and to the prediction of a sun-spot process are demonstrated.

*Y. Baram is with the Department of Computer Science, Technion, Israel Institute of Technology, Haifa 32000, Israel. He is also associated with the NASA Ames Research Center, Moffett Field, CA 94035.
This work was supported in part by the Director's Discretionary Fund, NASA Ames Research Center and in part by the Fund for the Promotion of Research at the Technion.

1 Introduction

Neural networks are being applied to a wide variaty of pattern recognition and signal processing problems. Statistical and Information theoretic methods are playing an increasing role in the design and analysis of such networks. The representation of probability density functions by neural networks (e.g., [1] - [6]) has been of particulat interest. Employing well established statistical performance criteria in neural network design leads not only to the development of new tools for problems that have been traditionally solved by linear regression methods, but to a more profound understanding and a more efficient application of neural networks.

In this work we first employ the maximum mutual information criterion in deriving the parameters of a feedforward sigmoidal network which produces an estimate of the probability density function (pdf) of a random vector. This criterion has been used recently in developing learning ([8, 9]) and feature selection [10] methods. The estimated pdf obtained for each class can be used as a comparative measure in solving classification problem. Then we derive a recursive maximum likelihood estimator for a random variable, given a random vector. This estimator employs the parameters calculated by the pdf estimator, and can be used in an adaptive mode. Application in the prediction of random sequences is immediate. Employing a particular sigmoidal nonlinearity ($\tanh(Wx+t)$) produces explicit expressions for the parameters of the resulting algorithms. Applications to real classification and process prediction problems are described.

2 Mutual Information and PDF Estimation by Sigmoids

Let $x \in R^n$ and $y \in R^n$ be random vectors, having probability density functions $p_X(x)$ and $p_Y(y)$, respectively. The *mutual information* between x and y is defined as [12]

$$I(x,y) = h(x) + h(y) - h(x,y)$$

where

$$h(x) = -E_x\{\log p_X(x)\}$$

with $E_x\{\cdot\}$ denoting expectation with respect to $p_X(x)$, is the entropy of x. Put in the form

$$I(x,y) = h(x) + h(x \mid y)$$

the mutual information between x and y is known to be the "information about x contained in y" (symmetriacally, of course, it is the "information about y contained in x")[12]. If y is to be used in making inferences about x, it is desirable to maximize the mutual information between them.

Let the i'th component of the vector y be

$$y_i = \frac{1}{\det W} G_i(u_i) \tag{1}$$

where $W \in R^2$ is a real nonsingular matrix, $G_i(\cdot)$ is a monotone increasing, continuous bounded function, a "sigmoid", and

$$u_i = W_i^T x + t_i \tag{2}$$

where W_i is the ith row of W and t_i is a scalar "threshold". In vector form

$$y = \frac{1}{\det W} G(u) \tag{3}$$

where $y = [y_1, \ldots, y_n]^T$, $u = [u_1, \ldots, u_n]^T$, and $G(u) = [G_1(u_1), \cdots, G_n(u_n)]^T$.

The probability density function of y satisfies [11]

$$p_Y(y) = \frac{p_X(x)}{\mid \det J(x) \mid} \Big|_{x=G^{-1}(y)} \tag{4}$$

where $\det J(x)$ is the determinant of the Jacobian of $y = G(x)$, whose i,j'th component is

$$J_{i,j} = \frac{\partial y_i(x)}{\partial x_j}$$

Clearly, $J(x)$ is a square matrix, and, for sigmoidal $G_i(x)$, the vector $x = G^{-1}(y)$, whose components are $G_i^{-1}(x)$, is well defined. It follows that

$$I(x,y) = h(y) = h(x) + E\{\log \mid \det J(x) \mid\}$$

Since $E\{\log p_X(x)\}$ does not depend on W or t, the maximum mutual information criterion becomes

$$\max_{W,t} E\{\log \mid \det J(x) \mid\} \tag{5}$$

How does the maximum mutual information criterion apply to the estimation of the pdf of a random vector? First note that

$$-I(x,y) = E\{\log p_X(x)\} - E\{\log \mid \det J(x) \mid\} = D(p_X(x), \mid \det J(x) \mid)$$

where $D(p_X(x), \mid \det J(x) \mid)$ is the *divergence* between $p_X(x)$ and $\mid \det(J(x)) \mid$. Furthermore, since $y \in [0,1]^n$, we have $h(y) \leq 0$, as the maximal entropy of any density on $[0,1]^n$ is not greater than 0, the entropy of a uniform density [12]. It follows that $D(p_X(x), \mid \det J(x) \mid) \geq 0$, hence, maximizing the mutual information between x and y is equivalent to minimizing the divergence between $p_X(x)$ and $\det(J(x))$. Noting that

$$\frac{\partial y_i(x)}{\partial x_j} = \sum_{k=1}^{n} \frac{\partial y_i(x)}{\partial u_k} \frac{\partial u_k(x)}{\partial x_j}$$

it follows that

$$\det(J) = \prod_{i=1}^{n} g(u_i) \tag{6}$$

where

$$g(u_i) = \partial G(u_i)/\partial u_i \tag{7}$$

Since $G(u_i)$ is monotone increasing in u_i, it follows that $g(u_i) > 0$, hence

$$|\det(J)| = \det(J)$$

The proposed pdf estimate is then

$$\hat{p}_X(x) = \det J(x) \tag{8}$$

3 Parameter Adaptation Algorithm

To find the optimal parameters, we need to maximize

$$S = E\{\log \det J(x)\} = \sum_{i=1}^{n} E\{\log g(W_i^T x + t_i)\}$$

The gradients of the latter with respect to W and $t = (t_1, \ldots, t_n)^T$ are found to be

$$\nabla_W S = E\{B(u)x^T\} \tag{9}$$

and

$$\nabla_t S = E\{B(u)\} \tag{10}$$

where

$$B(u) = [b(u_1), \ldots, b(u_n)]^T$$

with

$$b(u_i) = \frac{\partial}{\partial u_i} \log g(u_i) = \frac{\partial g(u_i)/\partial u_i}{g(u_i)}$$

We use, in particular, the function

$$G_i(u_i) = 0.5[1 + \tanh(u_i)]$$

for which

$$g(u_i) = \frac{0.5}{\cosh^2(u_i)}$$

hence

$$b(u_i) = -2\tanh(u_i)$$

Iterative algorithms of the form

$$W(k+1) = W(k) + \mu(k)\nabla S(k) \tag{11}$$

where $\mu(k)$ is a step size control parameter and $\nabla S(k)$ is an empirical version of ∇S, can be used in searching for the optimal parameters. One possibility is replacing the expectations in (9) and (10) by empirical averages over the input samples. Another is replacing them by the samples themselves. For instance, $E\{B[W(k)x + t]x^T\}$ would be replaced by $\frac{1}{M}\sum_{i=1}^{M} B[W(k)x^{(i)} + t]x^{(i)^T}$, where $x^{(i)}$ is the i'th training input vector, or simply by $B[W(k)x^{(k)} + t]x^{(k)^T}$. Using the inverse of the Hessian of S with respect to the parameters as the step size control parameter is likely to speed up the convergence rate of the algorithm [13], although its computation may require considerable time for high dimensional inputs. The Hessian in our case is a four dimensional tensor. Updating the columns of W one at a time, the Hessian for the m'th column at the k'th iteration is a matrix whose i,j'th element is

$$\left[\nabla^2 S(k)^{(m)}\right]_{i,j} = E\left\{\frac{x_m x_j}{\cosh^2(u_i(k))}\right\} \approx \frac{1}{M}\sum_{i=1}^{M} \frac{x_m^{(i)} x_j^{(i)}}{\cosh^2(u_i(k))} \tag{12}$$

An immediate application of the pdf estimate is in classification problems. In training, the pdf corresponding to each of the classes is learnt by a different set of parameters. In operation, the class corresponding to the largest pdf is selected. Another application is in estimating random variables and random sequences. This is discussed next.

4 Estimating Random Variables and Sequences

Let x be a random variable, having a probability density function (pdf) $p_X(x)$, let y be a vector of n random variables having a joint pdf $p_Y(y)$, and let the joint pdf of x and Y be denoted $p_{X,Y}(x,y)$. The maximum likelihood estimate of x given y is obtained by maximizing the corresponding conditional pdf

$$p(x \mid y) = \frac{p_{X,Y}(x,y)}{p_Y(y)}$$

with respect to x, which is the same as maximizing $p_{X,Y}(x,y)$ with respect to x.

In real problems, $p_{X,Y}(x,y)$ is not available and must be estimated from the data. Defining

$$\tilde{x} = (y^T, x)^T$$

as the vector obtained by concatenating y and x, the proposed pdf estimate of $\tilde{x}$ is

$$\hat{p}_{\tilde{X}}(\tilde{x}) = \prod_{i=1}^{N} g(\tilde{W}_i \tilde{x} + t_i)$$

where $\tilde{W}_i$ is the i'th row of the weights matrix obtained in estimating $p_{\tilde{X}}(\tilde{x})$ by the algorithm described in the previous section.

Maximization of $\hat{p}_{X,Y}(x,y)$ is equivalent to maximization of $\log p_{X,Y}(x,y)$, which is, in turn, equivalent to the maximization of

$$f(\tilde{x}) = \sum_{i=1}^{n} \log g(\tilde{W}_i \tilde{x} + t_i)$$

Newton's iterative optimization algorithm for maximizing $f(\tilde{x})$ with respect to the estimated variable x is [13]

$$x(k) = x(k-1) + \left[\frac{\partial^2 f(\tilde{x})}{\partial x^2}\right]^{-1} \frac{\partial f(\tilde{x})}{\partial x}\Big|_{x=x(k-1)} \tag{13}$$

In our case

$$\frac{\partial f(\tilde{x})}{\partial x} = \sum_{i=1}^{n} \frac{\partial}{\partial x} \log g(\tilde{W}_i \tilde{x} + t_i)$$

and

$$\frac{\partial^2 f(\tilde{x})}{\partial x^2} = \sum_{i=1}^{n} \frac{\partial^2}{\partial x^2} \log g(\tilde{W}_i \tilde{x} + t_i)$$

We have

$$g(\tilde{W}_i \tilde{x} + t) = \frac{0.5}{\cosh^2(\tilde{W}_i \tilde{x} + t_i)}$$

It follows that

$$\frac{\partial}{\partial x} \log g(\tilde{W}_i \tilde{x} + t_i) = -\tilde{W}_{i,n} \tanh(\tilde{W}_i \tilde{x} + t)$$

and

$$\frac{\partial^2}{\partial x^2} \log g(\tilde{W}_i \tilde{x} + t) = -\tilde{W}_{i,n}^2$$

Hence the iterative algorithm (4) becomes

$$x(k) = x(k-1) + \mu(k-1) \tilde{W}^{(n)^T}(k-1) z(k-1)$$

where

$$\mu(k-1) = \left[\sum_{i=1}^{n} \tilde{W}_{i,n}^2(k-1)\right]^{-1}$$

$\tilde{W}^{(n)}(k)$ is the last column of the weights matrix $\tilde{W}(k)$ and $z(k-1)$ is the vector whose i'th component is

$$z_i(k-1) = \tanh\left(u_i(k-1)\right)$$

where $u_i(k) = \tilde{W}(k)\tilde{x}(k) + t(k)$.

The problem of predicting the value of a random sequence $x_1, x_2, \ldots\ldots$, at instance n given N previous values is naturally addressed by the proposed method. Simply define $y = (x_{n-N}, \ldots, x_{n-1})^T$ and $x = x_n$ and apply the algorithm described above.

References

[1] H. W. White, "Learning in Artificial Neural Networks: A Statistical Perspective," Neural Computation, Vol. 1, pp. 425–464, 1989.

[2] N. Tishby, E. Levine, S. Solla, "Consistent Inference of Probabilities in Layered Neural Networks: Prediction and Generalization," Proceedings of the Joint Conference on Neural Networks, Washington D. C., Vol. 2, pp. 403–409, 1989.

[3] S. P. Luttrell, "The Use of Bayesian and Entropic Methods in Neural Network Theory," in J. Skilling (ed.), Maximum Entropy and Bayesian Methods, pp. 363–370, Kluwer Academic Publishers, Boston, 1989.

[4] E. Yair and A. Gersho, "The Boltzmann Perceptron Network: A Soft Classifier," Vol. 3, pp. 203–221, 1990.

[5] H. G. C. Tavern, "A neural Network Approach to Statistical Pattern Recognition by 'Semiparametric' Estimation of Probability Density Functions," IEEE Trans. on Neural Networks, Vol. 2, No. 3, pp. 366–378, May 1991.

[6] M. D. Richard and R. P. Lipmann, "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities," Neural Computation, Vol. 3, pp. 461–483, 1991.

[7] H. Shioler and U. Hartman, "Mapping Neural Networks Derived from the Parzen Window Estimator," Neural Networks, Vol. 5, pp. 903–909, 1992.

[8] M. Bichsel and P. Seiz, "Minimum Class Entropy: A Maximum Entropy Approach to Layered Networks," Neural Networks, 2: 133–141, 1989.

[9] R. Linsker, "How to Generate Ordered Maps by Naximizing the Mutual Information between Input and Output Signals," Neural Computation, Vol. 1, No. 3, pp. 402–411, 1989.

[10] R. Battiti, "Using Mutual Information for Selecting Features in Unsupervised Neural Net Learning," IEEE Trans. on Neural Networks, Vol. 5, No. 4, pp. 537–550, July 1994.

[11] P. G. Hoel, Introduction to Mathematical Statistics, Wiley, New York, 1984.

[12] A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw–Hill, Princeton, 1984.

[13] D. G. Luenberger, Optimization by Vector Space Methods, Wiley, New York, 1969.