CONF-9107

LA-UR -91-325

LA-UR--91-325

DE91 007552

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE: INFORMATION THEORETIC DERIVATION OF NETWORK ARCHITECTURE AND LEARNING ALGORITHMS

AUTHOR(S): R. D. Jones, C. W. Barnes, Y. C. Lee, W. C. Mead

SUBMITTED TO: IJCNN-91-Seattle Conference
Seattle, WA

DISCLAIMER

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

INFORMATION THEORETIC DERIVATION OF NETWORK ARCHITECTURE AND LEARNING ALGORITHMS

R. D. JONES[a], C. W. BARNES, Y. C. LEE, and W. C. MEAD

Los Alamos National Laboratory
Los Alamos, New Mexico 87545 USA
[a]*rdj@lanl.gov*

ABSTRACT

Using variational techniques, we derive a feedforward network architecture that minimizes a least squares cost function with the soft constraint that the mutual information between input and output be maximized. This permits optimum generalization for a given accuracy. A set of leearning algorithms are also obtained. The network and learning algorithms are tested on a set of test problems which emphasize time series prediction.

1. Introduction

The goal of feedforward networks is to maximize network approximation accuracy while also maximizing network generalizability. The accuracy of the output is typically obtained by implementation of learning algorithms which minimize a least squares cost function. Generalizability can be obtained through maximization of a cost function proportional to the mutual information between input and output.[1] These techniques generalize work of Kohonen.[2] In this paper we demonstrate how the optimization of a composite cost function that contains both a least squares term and a mutual information term can lead to an architecture and to learning algorithms which yield optimum generalization for a given accuracy.

2. The Architecture

We can now examine the probability distribution which minimizes the cost function

$$F \equiv E - \tau S, \tag{1}$$

where

$$E = \frac{N}{2} \int_I dx \int_O dy\ \epsilon(x,y) P(y \mid x) P(x)$$

and

$$S = N \int_I dx \int_O dy\ P(y \mid x) P(x)\ log\left[\frac{P(y \mid x)}{P(y)}\right].$$

Here, $\epsilon(x,y)$ is

$$\epsilon(x,y) \equiv \frac{1}{2}[y - \phi(x)]^T [y - \phi(x)]$$

where

$$\phi(x) = \int_O dy\ y P(y \mid x) \tag{2}$$

and

$$P(y) = \int dx\ P(y \mid x) P(x). \tag{3}$$

Here, E is the least squares portion of the cost function and S is the mutual information. We find the functional form of $P(y \mid x)$ by equating the functional deriative to zero. We obtain a relation between

the conditional probability, $P(y \mid x)$ known as the ***posterior***, and the pre-measurement probability, $P(y)$ known as the ***prior***. In general, these two probabilities are related through Bayes theorem,

$$P(y \mid x) = \frac{P(x \mid y)}{P(x)} P(y)$$

where

$$P(x) = \int_O dy \; P(x \mid y) P(y).$$

The conditional probability, $P(x \mid y)$, is known as the ***likelihood***.

We can regard Eqs. 2 and 3 as well as the normalization condition,

$$1 = \int_O dy \; P(y \mid x),$$

as soft constraints. The cost function can be modified to read

$$F' = N \int_I dx \int_O dy \; P(y \mid x) P(x) \left[\epsilon(x,y) - \tau \, log \left[\frac{P(y \mid x)}{P(y)} \right] + \alpha(x) + \beta(y) + \gamma^T(x) y \right]$$

where $\alpha(x) \in \Re$, $\beta(y) \in Re$, and $\gamma(x) \in \Re^m$ are functions that are determined by the constraints. Setting the functional derivative,

$$\frac{\delta F'}{\delta P(y \mid x)},$$

equal to zero yields

$$\epsilon(x,y) - \tau \, log \left[\frac{P(y \mid x)}{P(y)} \right] + \alpha(x) + \beta(y) + y\gamma(x) - \tau = 0.$$

Solving for $P(y \mid x)$ obtains

$$\phi(x) = \frac{1}{P(x)} \int_O dy \; y \frac{E(x,y)}{\int dx E(x,y)} P(y) \qquad 3$$

where

$$E(x,y) \equiv exp \left[- \frac{y^T y - (\gamma(x) - \phi(x))^T y + \phi^T(x)\phi(x)}{2\tau} \right].$$

This is an integral equation which contains the arbitrary function, $\gamma(x)$. In general, $\gamma(x)$ cannot be chosen such that Eq. 3 is exactly satisfied. However, it can be approximately satisfied if

$$\gamma(x) \equiv 2\phi(x).$$

This forces the integrand in Eq. 3 to be sharply peaked around $y = \phi(x)$. $E(x,y)$ becomes

$$E(x,y) = exp \left[- \frac{\epsilon(x,y)}{\tau} \right]. \qquad 4$$

We see from Eq. 2 that the posterior is

$$P(y \mid x) = \frac{1}{P(x)} \frac{E(x,y)}{\int dx E(x,y)} P(y).$$

This yields for the likelihood

$$P(x \mid y) = \frac{E(x,y)}{\int dx E(x,y)}. \qquad 5$$

The output of the network can be written

$$\phi(x) = \frac{1}{P(x)} \int_O dy\, yP(x \mid y)P(y). \qquad 6$$

Equations 3 to 6 define a near optimum architecture given the soft constraints imposed on the least squares minimization. From Eqs. 5 and 4 it can be seen that the role of the Lagrange multiplier, τ, is to spread the input probability, $P(x)$. This will degrade the fit of $\phi(x)$ to $f(x)$ but it will improve the generalizability of the network. For very large values of τ the network only acquires the average value of $f(x)$. This is the ultimate generalization.

Suppose we have just presented the network with N training pairs and we had exact knowledge of the desired output. Then $P(y)$ is given by

$$P(y) = \frac{1}{N} \sum_{p=1}^{N} \delta(y - f(x_p)).$$

Thus $P(x)$ can be written

$$P(x) = \int dy P(x \mid y)P(y) = \frac{1}{N} \sum_{p=1}^{N} P(x \mid f(x_p)).$$

The posterior becomes

$$P(y \mid x) = \sum_{p=1}^{N} \frac{P(x \mid f(x_p))}{\sum_{q=1}^{N} P(x \mid f(x_q))} \delta(y - f(x_p))$$

and the network output becomes

$$\phi(x) = \frac{\sum_{p=1}^{N} f(x_p)P(x \mid f(x_p))}{\sum_{q=1}^{N} P(x \mid f(x_q))}.$$

This architecture is a linear superposition of nonlinear elements. We can make the identifications

$$P(y \mid x) = \sum_{j=1}^{M} u(x; b_j)v(y - a_j)$$

$$u(x; b_p) \equiv P(p \mid x) = \frac{P(x \mid f(x_p))}{\sum_{q=1}^{N} P(x \mid f(x_q))},$$

and

$$v(y - f(x_p)) \equiv P(y \mid p) = \delta(y - f(x_p)).$$

Moreover, we can write the specific form of the nonlinear element as

$$u(x, b_p) = \left[\sum_{q=1}^{N} \frac{E(x, f(x_q))}{\int dx E(x, f(x_q))} \right]^{-1} \frac{E(x, f(x_p))}{\int dx E(x, f(x_p))}, \qquad 7$$

where

$$E(x, f(x_p)) = exp\left[-\frac{\epsilon(x, f(x_p))}{\tau} \right]. \qquad 8$$

Equations 7 and 8 give the optimum architecture given that the desired output is perfectly known. The architecture is self-consistant. The output must be known in order to specify ϵ and ϵ must be known

in order to specify the output. We can remove the self-consistancy in the limit of small τ. In that case the function $E(x, f(x_p))$ is sharply peaked about $\phi(x) = f(x_p)$. If we Taylor expand $\phi(x)$ about $f(x_p)$ the energy, ϵ, becomes

$$\epsilon(x, f(x_p)) \approx \frac{1}{2}\left[(x - x_p)^T \left(\frac{\partial\phi(x)}{\partial x}\right)_{x=x_p}\right]\left[(x - x_p)^T \left(\frac{\partial\phi(x)}{\partial x}\right)_{x=x_p}\right]^T.$$

Here,

$$\left(\frac{\partial\phi(x)}{\partial x}\right)_{x=x_p}$$

is an $n \times m$ matrix. Equation 7 becomes

$$u(x, b_p) = \frac{|\gamma_p^{-1}|^{-1/2} exp\left[-(x - x_p)^T \gamma_p (x - x_p)\right]}{\sum_{q=1}^{N} |\gamma_q^{-1}|^{-1/2} exp\left[-(x - x_q)^T \gamma_q (x - x_q)\right]} \qquad 9$$

where γ_p is an $n \times n$ matrix given by

$$\gamma_p \equiv \frac{1}{2\tau}\left(\frac{\partial\phi(x)}{\partial x}\right)_{x=x_p}\left(\frac{\partial\phi(x)}{\partial x}\right)^T_{x=x_p}.$$

If the data is preprocessed such that each input affects the output nearly independently and redundant and irrelevant inputs are not present then Eq. 9 can be simplified to

$$u(x, b_p) = \frac{\beta_p^{n/2} exp\left[-\beta_p (x - x_p)^T (x - x_p)\right]}{\sum_{q=1}^{N} \beta_q^{n/2} exp\left[-\beta_q (x - x_q)^T (x - x_q)\right]} \qquad 10$$

where γ can be written as a scalar β times a unit matrix. Writing Eq. 9 as Eq. 10 is a serious and drastic approximation. The process of simplifying the matrix γ_p is known as ***feature extraction.***

The network output becomes

$$\phi(x) = \sum_{p=1}^{N} f(x_p) u(x, b_p) \qquad 11$$

where u is given by Eq. 10. We will permit the quantities $f(x_p)$, x_p, and β_p to be adjustable parameters that minimize a given cost function. We can thus associate β_p and x_p with the parameters b_p in the function $u(x, b_p)$.

We can relate Eq. 11 to a network architecture. Note the identity,

$$g(x) = \frac{\sum_{j=1}^{N} g(x)\rho_j(x)}{\sum_{j=1}^{N} \rho_j(x)}. \qquad 12$$

Here, $\rho_j(x)$ is a localized function of x about some a_j. Hence, $g(x)$ on the right of Eq. 12 can be approximated by its Taylor expansion about a_j. We have then,

$$\phi(x) = \sum_{j=1}^{N} [f_j + (x - a_j) \cdot d_j] \frac{\rho_j(x)}{\sum_j \rho_j(x)}$$

for an approximation to $g(x)$. If we compare with Eq. 10 and 11 we can write a specific form for the local function, ρ,

$$\rho_j(x) = \beta_p^{n/2} exp\left[-\beta_p (x - x_p)^T (x - x_p)\right]. \qquad ???$$

This net is similar to the radial basis function net of Moody and Darken[3] but differs in two ways, the use of normalization and also a linear term, $(x - a_j) \cdot d_j$. The use of a normalization term was suggested but not pursued by Moody and Darken. The addition of these two terms is responsible for the reduction in the amount of training data needed to obtain reasonable approximations. As in the case with radial basis functions, the training of f_j and d_j is linear and hence very fast. The widths of the basis functions can also be trained. This training is nonlinear.

3. Learning Algorithms

The training for the linear quantities, f_j and d_j is given by

$$f_j^{p+1} = f_j^p + \alpha[g(x_p) - \phi(x_p)]\frac{\rho_j(x_p)\sum_k \rho_k(x_p)}{\sum_k [\rho_k^2(x_p) + \beta_k(x - a_k)^2\rho_k^2(x_p)]}$$

and

$$d_j^{p+1} = d_j^p + \alpha[g(x_p) - \phi(x_p)]\frac{\beta_j(x_p - a_j)\rho_j(x_p)\sum_k \rho_k(x_p)}{\sum_k [\rho_k^2(x_p) + \beta_k(x - a_k)^2\rho_k^2(x_p)]}.$$

Here, x_p is a training input vector and $g(x_p)$ is the training output for the p^{th} on-line presentation of data to the network. The superscripts p and $p+1$ indicate values of the weights before and after the p^{th} presentation of training data. The learning rate, α, is usually taken to be 1/3. This training algorithm or modifications of it have been used in time series prediction and in control.[4-6]

We can also train the basis function widths. Equation 7 can be written

$$F' = N \int_I dx \int_O dy\ P(y \mid x)P(x)\left[-\epsilon(x,y) - \tau\, log\left[\frac{P(y \mid x)P(x)\int dx E(x,y)}{P(y)}\right] + \tau\right]$$

We have

$$\int dx E(x, a_j) \approx \left(\frac{2\pi}{\beta_j}\right)^{n/2}.$$

If we apply gradient descent learning to this cost function we find

$$\beta_j^{p+1} = \beta_j^p - \eta\frac{n\beta_j^p}{2}\ \rho(x_p; b_j^p, \beta_j^p)\ v[\phi(x_p) - a_j]$$

$$+\eta(\beta_j^p)^2\left[\frac{n}{2\beta_j^p} - (x_p - b_j^p)^T(x_p - b_j^p)\right]\ \rho(x_p; b_j^p, \beta_j^p)\ v[\phi(x_p) - a_j]\ K[x_p, \phi(x_p); a, b, \beta] \qquad 13$$

where K is now given by

$$K(x_p, y; a, b, \beta) \equiv log\left[\frac{\pi(x_p, y; a, b, \beta)\ \pi(x; b, \beta)}{\pi(y; a)\ \beta^{n/2}}\right] + \frac{1}{2\tau}[a_j - \phi(x)]^T[a_j - \phi(x)].$$

In each of these algorithms the conscience is affected indirectly by the errors that are being made in the approximation of f.

The behavior of this algorithm on the logistic map is displayed in Fig. 1.

Summary

We have derived a network architecture from first principles. The network is designed to optimize both accuracy and generalization. The architecture resembles local radial basis function networks with two important modifications, a normalization which greatly reduces the data requirements and an extra set of gradient style weights which improves interpolation. Performing gradient descent on the composite cost function obtains a learning algorithm for the basis function widths which adjusts the widths for good generalization.

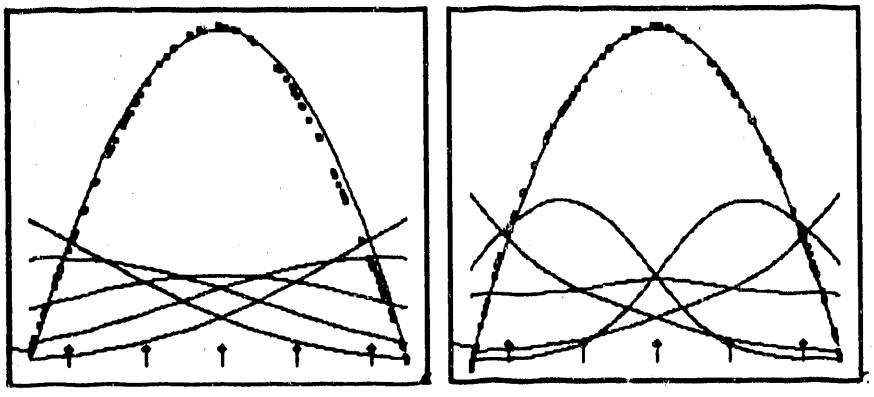

Fig. 1 Behavior of Eq. 1.5-15 on the approximation of the logistic map. The left plot is with no training of the widths, β_j, and the right plot is with training using Eq. 1.5-3. The upper plot is of $x_{p+1} = 4x_p(1 - x_p)$ where x_p is plotted on the abscissa and x_{p+1} is plotted on the ordinate. The solid line is the desired curve. The dots are the predictions of the network. The training on the gradients, d, was turned off. 20000 training patterns were used. Five evenly spaced basis functions were used. The locations are indicated by hash marks with diamonds. Also displayed in each plot are the five curves for the five basis functions. Not the shape adjustment in the right plot. The learning rate, η, was set to 0.3 for the training of the coefficients f_j and to 0.03 for the β training. The root mean square error normalized to the standard deviation of the data was 0.075 for the case with no β training and 0.035 for the case in which the betas were trained.

References

1. R. Linsker, "How to generate ordered maps by maximizing the mutual information between input and output signals," *Neural Computation,* **1**, (1989) 402.

2. T. Kohonen, *Self-Organization and Associative Memory*, (Springer-Verlag, Berlin, 1989).

3. J. Moody and C. J. Darken, "Fast learning in networks of locally tuned processing units," *Neural Computation,* **1**, (1989) 281-294.

4. R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, K. Lee, P. S. Lewis and S. Qian, "Function approximation and time series prediction with neural networks," ***Talk presented at the Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico on September 27, 1989.*** Los Alamos Report LA-UR-90-21. and in *Proceedings of the International Joint Conference on Neural Networks, San Diego, California, June 17-21, 1990.* (1990) I-649-666.

5. W. C. Mead, R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, L. A. Lee, and M. K. O'Rourke, "Using CNLS-Net to Predict the Mackey-Glass Chaotic Time Series," *Proceedings of the International Joint Conference on Neural Networks, Seattle, Washington, 1991.*

6. C. W. Barnes, S. K. Brown, G. W. Flake, R. D. Jones, M. K. O'Rourke, and Y. C. Lee "Applications of Neural Networks to Process Control and Modeling," *Proceedings of the International Joint Conference on Neural Networks, Seattle, Washington, 1991.*

END

DATE FILMED

06 / 07 / 91