

MINIMUM MEAN SQUARE ESTIMATION AND NEURAL NETWORKS

Michael T. Manry¹, Steven J. Apollo², and Qiang Yu³

¹Department of Electrical Engineering, University of Texas at
Arlington, Arlington, Texas 76019

²Lockheed Fort Worth Company, Mail Zone 2615
P.O. Box 748
Fort Worth, Texas 76101

³Worldcom, Inc.
1 William Center
Tulsa, Oklahoma 74101-0949

Abstract

Neural networks for estimation, such as the multilayer perceptron (MLP) and functional link net (FLN), are shown to approximate the minimum mean square estimator rather than the maximum likelihood estimator or others. Cramer-Rao maximum a posteriori lower bounds on estimation error can therefore be used to approximately bound network training error, when a statistical signal model is available for its inputs and the desired outputs are Gaussian. The bounds help the user to determine when to stop training, and to determine how close to optimal the neural net's performance is. When a linear preprocessor is sought to compress raw data, before it is input into a neural network, the bounds can be used to determine the relative optimality of several candidate linear preprocessors or transforms. A method is proposed for re-ordering the rows of the preprocessor's transform matrix. It is shown that a single linear transformation can be used, even when more than one parameter is estimated by the network.

Published in : *Neurocomputing*, vol. 13, September 1996, pp. 59-74.

I. Introduction

In the past several years, neural networks [1-4] have been used for many tasks, including classification and mapping. There have been many useful theoretical results concerning the capabilities of neural nets. They have been shown to approximate Bayes classifiers when trained using the mean square error (MSE) objective function [5,6]. They have been shown to have good approximation capabilities [7-10]. Bounds on mapping performance, in the absence of noise, have been found [11]. In parameter estimation [12], which can be considered to be a mapping problem in which the inputs have noise, neural networks are beginning to find use [13-15]. One major parameter estimation application is the inversion of terrain parameters from microwave measurements, in remote sensing [16-20]. Another major application is power load forecasting [21,22], in which an electric utility uses past and present power loads, past and present temperature, and other inputs to predict power load one hour, one day, or one week in the future.

Several problems remain in neural net parameter estimators. It is not known whether such estimators emulate maximum likelihood estimators or minimum mean-square estimators. Similarly, it is not known whether straight Cramer-Rao bounds or Cramer-Rao maximum a posteriori (MAP) bounds on estimation error variance are appropriate. It is not clear how close neural network parameter estimators come to being optimal. Because of the scaling difficulties of neural nets, it is advantageous to compress their inputs via linear transforms. However, it is not clear which of the many available transforms should be used. It is not clear which subset of the transform coefficients are theoretically the most useful. It has been observed that network estimation performance can deteriorate as new inputs are added and the network is retrained. It has been found that having a separate network for each parameter often works better than having a single network trained to estimate multiple parameters.

In this paper, neural nets are related to optimal minimum mean square parameter estimators. In section II, the limit of neural network training error is found. It is pointed out that the Cramer-Rao MAP bounds are bounds on the limit of the training error. These bounds require a statistical signal model of the noisy inputs, and can be tight when the parameters are Gaussian. In section III, these bounds are used in the theoretical evaluation of linear pre-processors. An efficient method is shown for evaluating the bounds. The bounds are used to evaluate the relative performances of transforms used as linear pre-processors. In section IV, a method is proposed for re-ordering the transform's coefficients. An objective function is proposed for combining the bounds for the multiple parameter case. An order function is proposed, as a method for determining the best subset of an given transform's coefficients. Examples are provided in section V to illustrate the proposed methods. In our analyses of the second example, it becomes clear why adding extra input features can sometimes cause a decrease in the performance of the network.

II. Parameter Estimation Using Neural Networks

Here we show that when a neural network, such as the MLP, is applied to the problem of parameter estimation, it approximates the optimal minimum mean square (MMS) estimator [12] in an analogous fashion. Additionally, lower bounds on the variances of MMS estimation errors are presented.

A. Limit of Neural Network Training Error

Consider an M -dimensional random vector \mathbf{V} representing an observation. Let $\boldsymbol{\theta}$ be an N_p -dimensional vector of parameters we wish to estimate. The MMS estimate $\boldsymbol{\theta}_{MMS}$ which minimizes the mean square error

$$\varepsilon_{MS} = E[(\boldsymbol{\theta} - \theta_{MMS})^T(\boldsymbol{\theta} - \theta_{MMS})] \quad (1)$$

is known [12] to be

$$\theta_{MMS} = E[\boldsymbol{\theta}|\mathbf{V}] \quad (2)$$

In many cases, equation (2) is difficult, or impossible to evaluate analytically. More tractable alternatives such as maximum likelihood (ML) and MAP estimation are frequently used instead for algorithm development and performance bounds. These are discussed in the next sub-section. However, one need not compromise if a neural network-based approach is adopted as the network can approximate the regression characteristic of equation (2). This is demonstrated theoretically below.

Let $F(\mathbf{V}, \mathbf{W})$ represent the vector of output unit activations in a neural net. The output units are assumed to have activation functions which cover the ranges of the parameters to be estimated. W is the vector of network weights. The network's training error is written as

$$\varepsilon_s(W) = \frac{1}{N_v} \sum_{k=1}^{N_v} \|F(V^k, W) - \theta^k\|^2 \quad (3)$$

where (V^k, θ^k) $k = 1, 2, \dots, N_v$ represents a training set drawn from a population that is representative of the true statistics of \mathbf{V} and $\boldsymbol{\theta}$.

We motivate minimum mean square estimation via neural networks by the following lemma and proof.

Lemma: If the neural network $F(\mathbf{V}, \mathbf{W})$ minimizes $\varepsilon_s(\mathbf{W})$, in the limit as N_v approaches ∞ , then it

is the minimum mean square estimator of θ .

Proof: As N_v becomes arbitrarily large, by the Strong Law of Large Numbers [23], equation (3) will tend to

$$\varepsilon(W) = \lim_{N_v \rightarrow \infty} \varepsilon_s(W) = \iint \|F(V,W) - \theta\|^2 p(V,\theta) dV d\theta \quad (4)$$

where $p(V, \theta)$ is the joint probability density function (pdf) of \mathbf{V} and θ . Since $p(V, \theta) = p(\theta|V)p(V)$ we can rewrite (4) as

$$\varepsilon(W) = \int p(V) \left[\int \|F(V,W) - \theta\|^2 p(\theta|V) d\theta \right] dV. \quad (5)$$

Minimizing equation (5) above is equivalent to minimizing the quantity in brackets

$$\varepsilon'(W) = \int \|F(V,W) - \theta\|^2 p(\theta|V) d\theta \quad (6)$$

which in turn is minimized when $F(\mathbf{V}, \mathbf{W}) = E[\theta|\mathbf{V}] = \theta_{MMS}$. The lemma is proven.

A sufficiently complex MLP can approximate the regression characteristic $E[\theta|\mathbf{V}]$ [7-10] provided that: (1) the topology is of sufficient complexity, (2) the output activation functions have sufficient range to span the parameters to be estimated, (3) unlimited training data are available that are representative of the underlying statistics, and (4) training successfully minimizes $\varepsilon_s(\mathbf{W})$.

B. Lower Bounds on Neural Network Estimation Variance

In MAP estimation [12], rather than minimizing $E[\theta|V]$ directly, one tries to minimize the conditional density

$$p_{\theta|\mathbf{V}}(\theta|V) = \frac{p_{\mathbf{V}|\theta}(V|\theta)p_{\theta}(\theta)}{p_{\mathbf{V}}(V)} \quad (7)$$

evaluated at \mathbf{V} equal to the observation V . The MAP and MMS estimates are equivalent when $p_{\theta|\mathbf{V}}$ has its maximum at θ_{MMS} [12]. The denominator of equation (7) can be ignored as it is a constant that depends only on the observation. For computational convenience we take the log of both sides to yield the log-likelihood function (LLF)

$$\Lambda^{MAP} = \Lambda^{MLE} + \Lambda^{AP}, \quad (8)$$

where $\Lambda^{MAP} = \ln(p_{\mathbf{V},\theta})$, $\Lambda^{MLE} = \ln(p_{\mathbf{V}|\theta})$, and $\Lambda^{AP} = \ln(p_{\theta})$. The superscripts *MLE* and *AP* respectively stand for maximum likelihood estimation and a-priori.

The elements of the MAP Fisher information matrix (FIM) \mathbf{J}^{MAP} are obtained by [12]

$$J_{ij}^{MAP} = E_{\theta}[J_{ij}^{MLE}] + E_{\theta}\left[\frac{\partial\Lambda^{AP}}{\partial\theta_i} \frac{\partial\Lambda^{AP}}{\partial\theta_j}\right], \quad (9)$$

$$J_{ij}^{MLE} \equiv E_N\left[\frac{\partial\Lambda^{MLE}}{\partial\theta_i} \frac{\partial\Lambda^{MLE}}{\partial\theta_j}\right]$$

where $E_{\theta}[\cdot]$ denotes expected value over the parameter vector θ and $E_N[\cdot]$ denotes expected value over the noise. Assume that the elements v_k of \mathbf{V} are modelled as

$$v_k = S_k + E_k \quad (10)$$

where the elements S_k and E_k are respectively elements of the signal and Gaussian noise vectors \mathbf{S} and \mathbf{E} . \mathbf{S} is a deterministic function of the parameter vector θ . The M by M covariance matrix of \mathbf{V} or \mathbf{E} is denoted by \mathbf{C}_V . The elements of \mathbf{J}^{MAP} in (9) can now be evaluated as

$$E_{\theta}[J_{ij}^{MLE}] = E_{\theta}\left[\left(\frac{\partial S}{\partial \theta_i}\right)^T C_v^{-1} \left(\frac{\partial S}{\partial \theta_j}\right)\right],$$

$$E_{\theta}\left[\frac{\partial \Lambda^{AP}}{\partial \theta_i} \frac{\partial \Lambda^{AP}}{\partial \theta_j}\right] = d_{\theta}(i,j)$$
(11)

where $d_{\theta}(i,j)$ denotes an element of \mathbf{C}_{θ}^{-1} , where \mathbf{C}_{θ} is the N_p by N_p covariance matrix of the N_p -dimensional parameter vector $\boldsymbol{\theta}$. Let $(\mathbf{J}^{MAP})^{ij}$ denote an element of $(\mathbf{J}^{MAP})^{-1}$. Then [12],

$$\text{var}(\theta_i' - \theta_i) \geq (\mathbf{J}^{MAP})^{ii}$$
(12)

where θ_i' can be any estimate of θ_i .

III. Theoretical Evaluation of Linear Pre-processors

Neural networks require training for use as parameter estimators. Since the time required for typical training methods, such as back-propagation (BP), grows at super-linear rates with the number of inputs, compression of the data through transforms yields a great cost advantage. Here we explore the use of linear transformations to reduce the amount of data presented to the neural network for processing.

Consider an observation vector $\mathbf{Z} = (z_1, z_1, \dots, z_N)^T$, with mean vector, $E_N[\mathbf{Z}] = \mathbf{s} = (s_1, s_2, \dots, s_N)^T$, and covariance matrix C_Z . A linear transformation of \mathbf{Z} into a feature vector \mathbf{V} can be expressed as the matrix multiplication:

$$\mathbf{V} = \Phi \cdot \mathbf{Z}$$
(13)

where Φ is an M by N transformation matrix and \mathbf{V} is an M by 1 vector. In this case, a smaller ($M < N$), transformed feature vector can reduce the number of inputs required in a neural network-based

estimator, which reduces the size of the network topology and hence the computation speed.

In Fig. 1, a parameter estimation system is shown which consists of a linear preprocessor and an MLP neural network. Given a set of linear transform matrices Φ ,

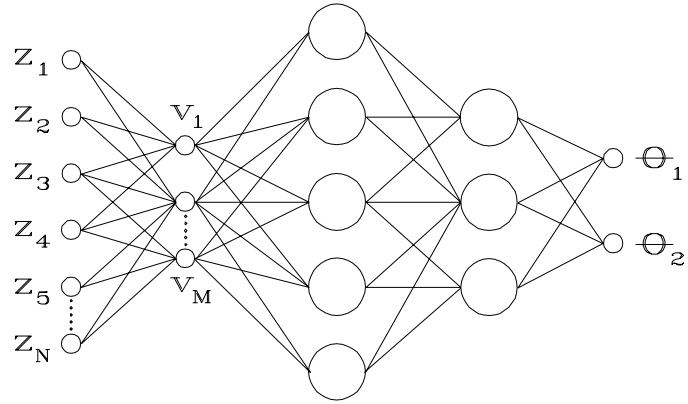


Figure 1. MLP Network With Linear Pre-Processor

our goal is to determine which linear transform is the most optimal. For a given parameter θ_n from the vector θ , and for a given value for M , the optimal transform is that which yields the lowest Cramer-Rao MAP bound on $\text{var}(\theta_n' - \theta_n)$.

A. Efficient Bound Calculation

In this subsection we develop efficient methods for calculating the MAP bounds. The covariance matrix of the feature domain noise, C_v , is calculated as

$$C_v = \Phi C_z \Phi^T \quad (14)$$

For simplicity, let's assume that (1) \mathbf{V} has Gaussian noise, (2) the covariance matrix C_v is diagonal, and that (3) θ is Gaussian. The first assumption is usually good, even if the noise vector \mathbf{E} is non-Gaussian, because of the central limit theorem. Note that the noiseless signal component \mathbf{S} of \mathbf{V} is not Gaussian, since elements of \mathbf{s} are not statistically independent. The second assumption is good whenever the elements of \mathbf{E} are from a stationary noise sequence and when Φ corresponds to an orthogonal transform. The third assumption is necessary if the MAP bounds are to be tight. The

MAP FIM element in (9) and (11) can be rewritten as

$$J_{ij}^{MAP}(M) = E_{\theta} \left[\sum_{m=1}^M \frac{\partial S(m)}{\partial \theta_i} \frac{\partial S(m)}{\partial \theta_j} d_{\mathbf{v}(m,m)} \right] + d_{\theta}(i,j) \quad (15)$$

where the argument M in $J_{ij}^{MAP}(M)$ denotes the number of features v_k used and where $d_{\mathbf{v}(m,n)}$ denotes an element of $\mathbf{C}_{\mathbf{v}}^{-1}$. The recursive calculation of $\mathbf{J}^{MAP}(k+1)$ from $\mathbf{J}^{MAP}(k)$ can be developed from (15) as

$$\begin{aligned} J_{ij}^{MAP}(k) &= d_{\theta}(i,j) + \sum_{m=1}^k u_{ij}(m), \\ J_{ij}^{MAP}(k+1) &= J_{ij}^{MAP}(k) + u_{ij}(k+1), \\ u_{ij}(m) &\equiv E_{\theta} \left[\frac{\partial S(m)}{\partial \theta_i} \frac{\partial S(m)}{\partial \theta_j} \right] \cdot d_{\mathbf{v}(m,m)}, \\ \frac{\partial S(m)}{\partial \theta_i} &= \sum_{k=1}^N \Phi_{mk} \frac{\partial s_k}{\partial \theta_i} \end{aligned} \quad (16)$$

where the $u_{ij}(k)$ can be computed and stored prior to the calculation of the FIM, Φ_{mk} is an element of the transform matrix Φ , and where s_k is $E_N[z_k]$.

B. Transform Evaluation Method

Many transforms are candidates for compressing the vector \mathbf{Z} into the feature vector \mathbf{V} . These include fast transforms [24] such as the fast Fourier transform (FFT) implementation of the discrete Fourier transform (DFT), the fast Walsh transform (FWT), the discrete cosine transform (DCT), and the wavelet transform [25]. Another candidate transform is the Karhunen-Loeve transform (KLT) [12,26], which does not have a fast implementation. The KLT is theoretically optimal [26] for compressing data for accurate reconstruction. This is no guarantee that it is optimal

if the goal is parameter estimation from transformed data. In practice, as we shall see, the KLT can give very good compression results for some data sets. A methodology for comparing feature sets, in order to determine their optimality relative to each other, is given below.

- (1) If p_{θ} is known (e.g. in MAP estimation), select an appropriate method to numerically evaluate the Cramer-Rao MAP bounds in (9) and (12). Use (16) for the case where \mathbf{C}_v is diagonal and θ is Gaussian.
- (2) For the raw observation case ($\Phi = \mathbf{I}$ where \mathbf{I} denotes the identity matrix), find the bounds. The raw observation case gives an indication of what the theoretical ideal is for the transformed observation case.
- (3) For each transform to be evaluated, increase the number of features until the Cramer-Rao bound for the transform case approaches acceptably close to that for raw time domain data. For a given number of features M , the best transform is that with the lowest bound.

IV. Optimal Ordering of Transform Feature Sets

There are some significant problems with step (3) in the previous subsection. There is no guarantee that a transform's natural order yields the best results. Multiple parameters complicate this problem further. If a signal model is bandpass for example, the low frequency DFT coefficients may do poorly, while coefficients close to the signal's center frequency do well.

A. Objective Function

For the one parameter case, it is fairly easy to order the transform coefficients. When adding a single new coefficient to the feature vector, we merely add that coefficient which most decreases the bound on the parameter's error variance. However, it is not clear how to order a transform's

coefficients according to their performance for the case of multiple parameters. For the two parameter case for example, the first parameter may be optimally estimated using features (v_1, v_2, v_3) while the second parameter may be optimally estimated using features (v_2, v_5, v_{11}) . Even worse, parameter θ_1 may be best estimated using DCT coefficients while parameter θ_2 is best estimated using DFT coefficients. When we have multiple parameters, it is necessary to define an objective function to help reconcile conflicting results from the individual parameters. The objective function that we have chosen is the weighted sum of Cramer-Rao MAP bounds,

$$O(k) \equiv \sum_{i=1}^{N_p} w_i \cdot (J^{MAP}(k))^{ii} \quad (17)$$

where $(J^{MAP}(k))^{ii}$ is the MAP bound on the i th parameter's error variance for the case of k features, w_i is a positive weight, and N_p is the number of parameters to be estimated.

B. Order Function Determination for the Multiple Parameter Case

Given that a transform matrix Φ has been chosen, using the procedure of section III, our goal is to develop a procedure for finding the best M -element subset of the N original features, v_k . However, the number of M -element subsets of N features is $N!/((N-M)!M!)$, which can be very large. Our solution to this difficulty is to order the features so that our best M -element subset consists of the first M elements of an ordered feature set. In other words, the feature subset of size M is formed by finding the best new feature to add to the subset of size $M-1$.

Let $o(k)$ denote an integer-valued order function such that (1) $o(k)$ takes on unique values between 1 and N as k varies from 1 to N and that (2) the features $v_{o(k)}$ for $1 \leq k \leq M$ are optimal for every positive value of M between 1 and N . Now, (16) can be rewritten as

$$\begin{aligned}
J_{ij}^{MAP}(k) &= d_{\theta}(i,j) + \sum_{m=1}^k u_{ij}(o(m)), \\
J_{ij}^{MAP}(k+1) &= J_{ij}^{MAP}(k) + u_{ij}(o(k+1))
\end{aligned} \tag{18}$$

For the multiple parameter case, we pick $o(1)$ as

$$o(1) = \operatorname{argmin}\{ O(1) \}$$

where the argmin is over $o(1)$ and

$$J_{ij}^{MAP}(1) = d_{\theta}(i,j) + u_{ij}(o(1))$$

To find the best value of $o(k+1)$ given $o(1)$ through $o(k)$, we choose $o(k+1)$ as

$$o(k+1) = \operatorname{argmin}\{ O(k+1) \}$$

where the argmin is over $o(k+1)$ and

$$J_{ij}^{MAP}(k+1) = J_{ij}^{MAP}(k) + u_{ij}(o(k+1))$$

V. Experimental Results

A. Finding the Best of Four Transforms

As a first example, we use the method of section III to find the best of four transforms. We generated a signal z_n with the signal model

$$z_n = A \cdot e^{-\frac{n}{\tau}} + n(n)$$

where $n(n)$ was zero-mean Gaussian white noise with a standard deviation of .1 and n varies from

0 to 127. The random parameters A and τ had uniform probability density functions which

varied from 1 to 2 for A and varied from 10 to 20 samples for τ . The covariance matrix for these parameters was used in (16), as if the parameters were Gaussian. The number of training vectors was $N_v = 1000$. As a first task, we can determine which transform out of a group of four is the most optimal.

The four transforms to be

evaluated, using the approach of section III, are the discrete Fourier transform (DFT), the Karhunen-Loeve transform (KLT), the fast walsh transform (FWT), and the discrete cosine transform (DCT).

In Fig. 2, the bounds for A for all four transforms are plotted versus the number of features used. The bound for the 128 time domain samples is plotted for reference. In Fig. 3, the bounds for all four transforms and the raw time domain data are plotted versus the number of features for the time constant τ .

In both plots, we see that only two

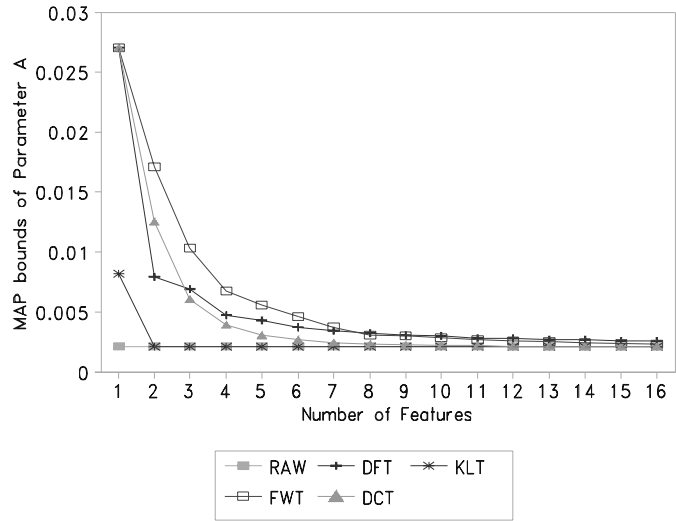


Figure 2. MAP Bounds on Parameter A for 4 Transforms

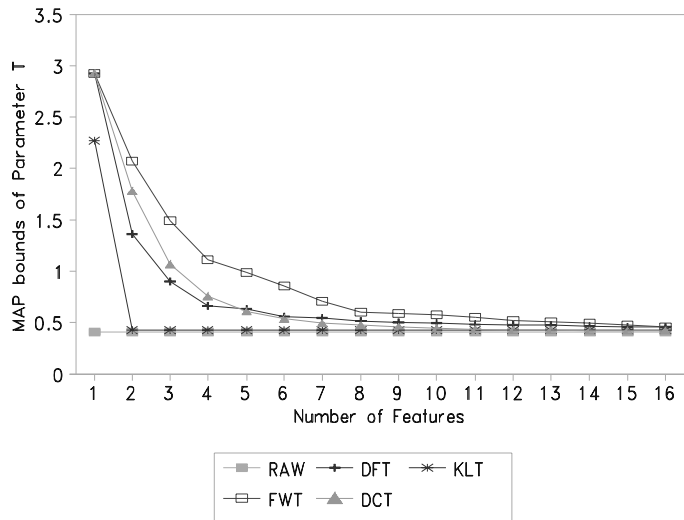


Figure 3. MAP Bounds on Parameter τ for Four Transforms

KLT features are needed to closely approach the performance of the raw data case. The DCT and DFT perform as well as the KLT, only after 8 or 9 features are used. The FWT seems less effective than the other transforms, especially when estimating the time constant τ .

As a second task, we evaluated the performance of the KLT features in an MLP network.

MLPs trained to estimate both parameters had topologies of the form M-10-10-2. In other words, there were M input transform coefficients where M varies between 2 and 16. The networks had 10 units in each of two hidden layers.

The networks were trained via five iterations of output weight optimization [16-19,27-29], which solves linear equations for output weights. The particular variation of OWO used is called OWO-BP [19], which uses backpropagation (BP) to improve hidden unit weights. The training and testing sets each had

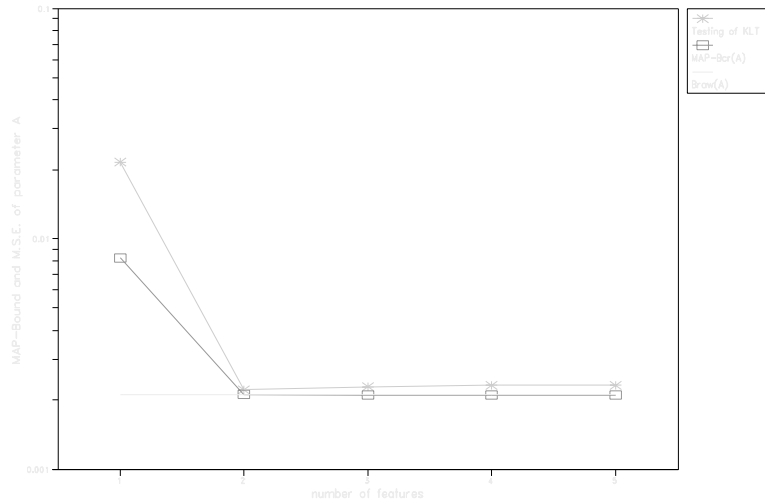


Figure 4. Bounds and Testing MSE for KLT and Parameter A

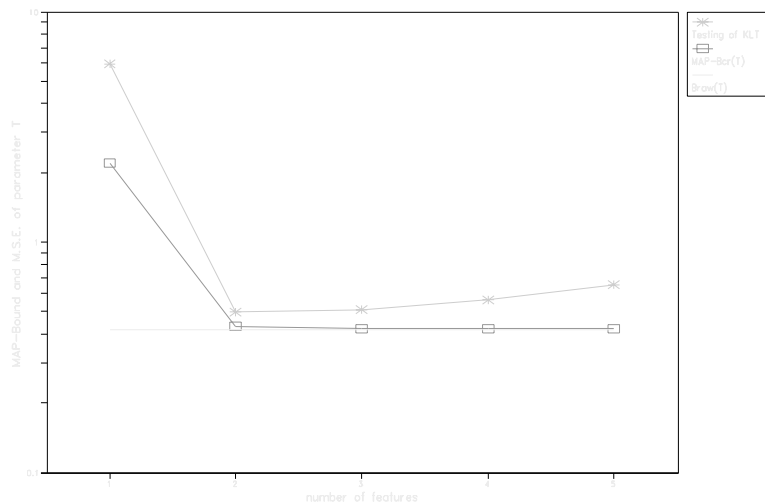


Figure 5. Bounds and Testing MSE for KLT and Parameter τ

5,000 patterns. In the function $O(k)$, $w_1 = .1$ (for the amplitude A) and $w_2 = .9$. The Cramer-Rao MAP bounds on $\text{var}(A-A')$ for raw data and KLT feature data are given in Fig. 4. Bounds for estimates of τ are shown in Fig. 5.

The testing results for the corresponding MLPs are also shown in the figures. The MLPs performed well and closely approached the bounds. Note however, that the performance in estimating τ decreases as the number of features increases past 2. This occurs because the extra features act as noise, making the training more difficult.

B. Optimizing Transform Feature Order

As a second example, we use the method of section IV find the best feature order when the DFT is used to compress a time domain signal. We generated a signal z_n with the signal model

$$z_n = A \cdot e^{-\frac{n}{\tau}} \cdot \sin(\omega \cdot n) + n(n)$$

where $n(n)$ was zero-mean Gaussian white noise with a standard deviation of .1. The random parameters A and τ had independent Gaussian probability density functions with respective means of 1.5 and 15, and with respective standard deviations of .1 and 1.5. The frequency ω had a value of .5 radians. MLPs trained to estimate both parameters had topologies of the form M-10-10-2 as before, and OWO-BP training was used. As before, the training and testing sets each had 5,000 patterns.

We evaluated the performance of real and imaginary parts of DFT features. In the function $O(k)$, $w_1 = .1$ (for the amplitude A) and $w_2 = .9$. The Cramer-Rao MAP bounds on the variance of the A estimate, for raw data, DFT feature data, and optimally ordered DFT features are given in Fig. 6.

Bounds for estimates of τ are shown in Fig. 7. The testing results for the corresponding MLPs are also shown in the figures. In figures 6 and 7, note that ordered features have lower bounds, denoted as *optimal* *MAP Bcr*, than do the non-ordered DFT features

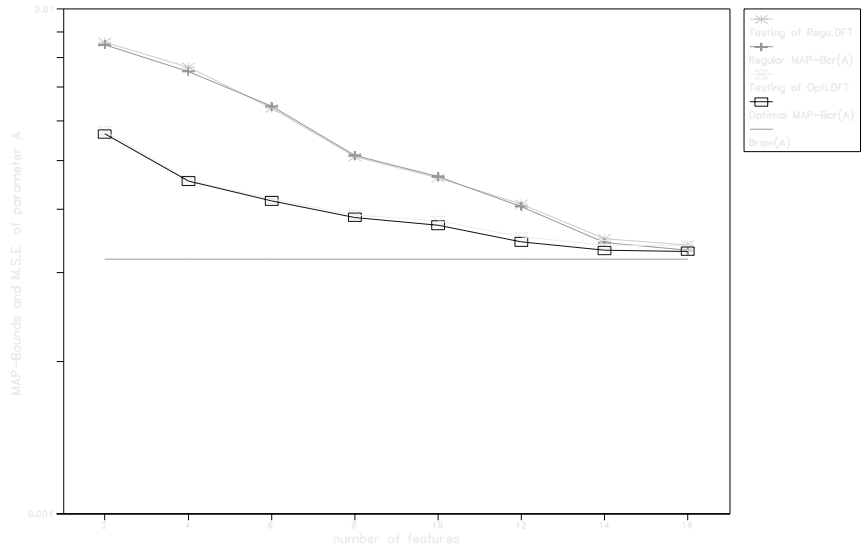


Figure 6. Bounds and Testing MSE for Parameter A

whose bounds are denoted by *MAP-Bcr*. In fact, the optimal MAP bounds are very close to the bounds for raw time domain data.

However, note that the estimates of τ require two features while the estimation of A requires many more. The additional features, although useful for estimating A , are merely noise as far as τ is concerned. This explains why neural net performance

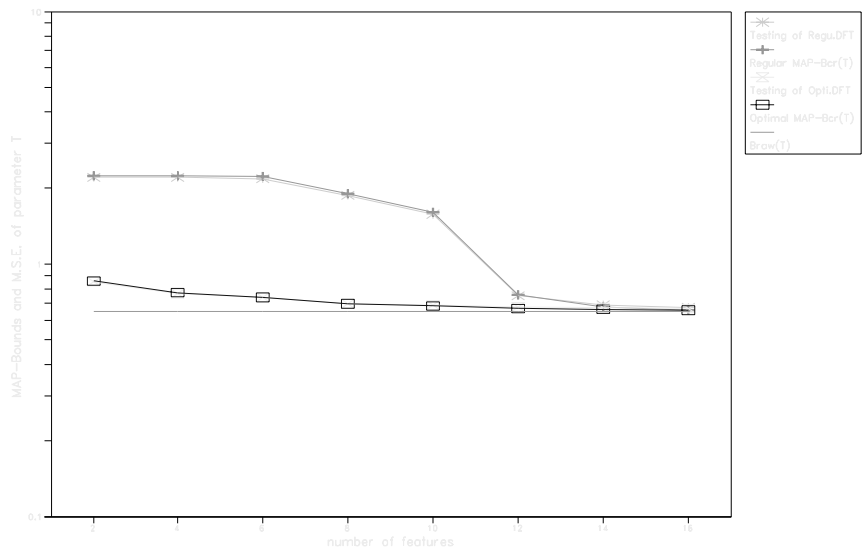


Figure 7. Bounds and Testing MSE for Parameter τ

can deteriorate when the number of inputs is increased.

Note that the neural net testing results are very close to the corresponding MAP bounds. When this occurs, the neural net performance is clearly adequate, and no further training is necessary.

VI. Conclusions

In this paper, we have shown that a neural network can approximate the minimum mean square estimator arbitrarily well, provided that it is of adequate size and is well-trained. We have described the utility of linear transformations to compress data efficiently for purposes of estimation via neural net techniques. A method for comparing transforms based upon transform domain error bounds was presented and a method for improving the transform through re-ordering was described. The bounds give clues as to how many transform coefficients are necessary and when training can be stopped. They also help us to understand why it is often productive to have separate MLPs for each parameter to be estimated.

Several problems remain to be solved. Although equation (16) works for the non-Gaussian parameter case, as demonstrated in our first example, the validity of this needs to be proven. Also, bounds need to be extended to the case where no signal model is given. These problems will be addressed in future papers.

Acknowledgements

This work was funded by NASA under Grant NAGW-3091, by the NSF under grant IRI-9216545, by EPRI under grant RP 8030-09, and by grants from Mobil Research and Development Corporation, E-Systems Garland Division, and the Advanced Technology Program of the state of Texas. Also, we thank the reviewers, whose many insightful comments improved the clarity of this paper.

VII. References

- [1] P. Werbos, "Beyond regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Ph.D. dissertation, Committee on Appl. Math., Harvard Univ., Cambridge, MA, Nov. 1974.
- [2] Y-H Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing Company, Inc., New York, 1989.
- [3] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation," in D.E. Rumelhart and J.L. McClelland (Eds.), *Parallel Distributed Processing*, Vol. I, Cambridge, Massachusetts: The MIT Press, 1986.
- [4] T. Kohonen, G. Barna, and R. Chrisley, "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies," *IEEE International Conference on Neural Networks*, San Diego, Ca., Vol. 1, pp. 61-68, 1989.
- [5] D. W. Ruck, S. K. Rogers, et. al., "The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function," *IEEE Trans. on Neural Networks*, vol. 1, pp. 296-298. Dec.1990.
- [6] E. A. Wan, "Neural Network Classification: a Bayesian Interpretation," *IEEE Trans. on Neural Networks*, vol. 1, pp. 303-305.
- [7] G. Cybenko, "Approximations by Superpositions of a Sigmoidal Function," *Math. Contrl., Signals, Syst.*, Vol. 2, pp. 303-314, 1989.
- [8] Eric Hartman, James D. Keeler, and Jacek M. Kowalski, "Layered Neural Networks with Gaussian Hidden Units as Universal Approximations," *Neural Computation* Vol. 2, No. 2, 1990.
- [9] Mu-Song Chen and M.T. Manry, "Back-propagation Representation Theorem Using Power Series," in *Proceedings of IJCNN*, Vol. I, pp. 643-648, 1990.
- [10] J. Wray and G.G.R. Green, "Neural Networks, Approximation Theory, and Finite Precision Computation," *Neural Networks*, Vol. 8, No. 1, 1995, pp. 31-37.
- [11] A.R. Barron, "Approximation and Estimation Bounds for Artificial Neural Networks," *Machine Learning*, Vol. 14, pp. 115-133, 1994.
- [12] H. L. Van Trees, *Detection, Estimation, and Modulation Theory - Part I*, New York, NY: John Wiley and Sons, 1968.
- [13] G. Govind and P.A. Ramamoorthy, "Multi-Layered Neural Networks and Volterra series: the Missing Link," *Proc. of the International Conference on Systems Engineering*, Pittsburgh, PA. August 1990.

- [14] T. M. Jelonek and James P. Reilly, "Maximum Likelihood Estimation for Direction of Arrival Using a Nonlinear Optimizing Neural Network," *Intrnl. Joint Conf. on Neural Networks*, vol. I, pp 253-258, 1990.
- [15] S.J. Apollo, M.T. Manry, L.S. Allen, and W.D. Lyle, "Optimality of Transforms for Parameter Estimation," *Conference Record of the Twenty-Sixth Annual Asilomar Conference on Signals, Systems, and Computers*, Oct. 1992, vol. 1, pp. 294-298.
- [16] M.S. Dawson, J. Olvera, A.K. Fung, M.T. Manry, "Inversion of Surface Parameters Using Fast Learning Neural Networks," *Proc. of IGARSS'92*, Houston, Texas, May 1992, vol. II, pp 910-912.
- [17] M.S. Dawson, A.K. Fung, and M.T. Manry, "Surface Parameter Retrieval Using Fast Learning Neural Networks," *Remote Sensing Reviews*, Vol. 7, pp. 1-18, 1993.
- [18] M.T. Manry, X. Guan, S.J. Apollo, L.S. Allen, W.D. Lyle, and W. Gong, "Output Weight Optimization for the Multi-layer Perceptron," *Conference Record of the Twenty-Sixth Annual Asilomar Conference on Signals, Systems, and Computers*, Oct. 1992, vol 1, pp. 502-506.
- [19] M.T. Manry, S.J. Apollo, L.S. Allen, W.D. Lyle, W. Gong, M.S. Dawson, and A.K. Fung, "Fast Training of Neural Networks for Remote Sensing," *Remote Sensing Reviews*, vol. 9, pp. 77-96, 1994.
- [20] X. Jiang, Mu-Song Chen, M.T. Manry, M.S. Dawson, A.K. Fung, "Analysis and Optimization of Neural Networks for Remote Sensing," *Remote Sensing Reviews*, vol. 9, pp. 97-114, 1994.
- [21] A. Khotanzad, R-C Hwang, and D. Maratuklam, "Hourly Load Forecasting by Neural Networks," presented at the *IEEE PES Winter Meeting*, Columbus, Ohio, February 1993.
- [22] K. Liu, S. Subbarayan, R.R.Shoults, M.T.Manry C.Kwan, F.L.Lewis, and J.Naccarino, "Comparison of Very Short-Term Load Forecasting Techniques," *IEEE Transactions on Power Systems*, to appear.
- [23] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill Book Company, New York, 1965.
- [24] D.F. Elliot and K.R. Rao, *Fast Transforms: Algorithms, Analyses, Applications*, Academic Press, 1982.
- [25] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes in C - The Art of Scientific Computing*, second edition, Cambridge, MA: Cambridge University Press, 1994.

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

[27] S.A. Barton, "A matrix method for optimizing a neural network," *Neural Computation*, vol. 3, no. 3, Fall 1991, pp. 450-459.

[28] M.A. Sartori and P.J. Antsaklis, "A simple method to derive bounds on the size and to train multilayer neural networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 4, July 1991, pp. 467-471.

[29] F. Biegler-Konig and F. Barmann, "A Learning Algorithm for Multilayered Neural Networks Based on Linear Least Squares Problems," *Neural Networks*, Vol. 6, No. 1, 1993, pp. 127-131.

Biography of M.T. Manry

Michael T. Manry was born in Houston, Texas in 1949. He received the B.S., M.S., and Ph.D. in Electrical Engineering in 1971, 1973, and 1976 respectively, from The University of Texas at Austin. After working there for two years as an Assistant Professor, he joined Schlumberger Well Services in Houston where he developed signal processing algorithms for magnetic resonance well logging and sonic well logging. He joined the Department of Electrical Engineering at the University of Texas at Arlington in 1982, and has held the rank of Professor since 1993. He is currently the director of the Image Processing and Neural Networks Laboratory in the Department of Electrical Engineering. In Summer 1989, Dr. Manry developed neural networks for the Image Processing Laboratory of Texas Instruments in Dallas. His recent work, sponsored by the Advanced Technology Program of the state of Texas, E-Systems, Mobil Research, the NSF, and NASA, has involved the development of techniques for the analysis and fast design of neural networks for image processing, parameter estimation, and pattern classification. Dr. Manry has served as a consultant for the Office of Missile Electronic Warfare at White Sands Missile Range, MICOM at Redstone Arsenal, Texas Instruments, Geophysics International, Halliburton Logging Services, Mobil Research, and Verity Instruments. He is a Senior Member of the IEEE.

Biography of Steven J. Apollo

Steven J. Apollo was born in Chicago, Illinois in 1959. He received a B.S. degree from the University of Tulsa in 1981, an M.S. degree from Stanford in 1982 and a Ph.D. from the University of Texas at Arlington in 1991, all in electrical engineering. After being a Bell Labs One-Year-On-Campus fellow at Stanford, he worked at AT&T Information Systems as a Member of Technical Staff from 1982-1984 developing digital modems and multiplexors. From 1984-1990 he was with General Dynamics Fort Worth Division developing algorithms for radar, Electronic Warfare and various tactical fighter avionics systems. From 1989-1990, he served as technical lead for the sensor technologies research while completing Ph.D. course-work. From 1990-1991, he was a graduate research/teaching assistant at the University of Texas at Arlington and a consultant for Mobil Research in Dallas, Texas. Since 1992, Dr. Apollo has been with Lockheed Fort Worth Company. He has performed advanced image processing/fusion algorithm development for multi-spectral imaging sensors including synthetic aperture radar and EO/IR imagers. He has also investigated applications of very high speed DSP systems to these image processing/fusion algorithms as well as very wide bandwidth/high sensitivity RF receivers.

Biography of Qiang Yu

Qiang Yu was born in Nanjing, China in 1962. In 1984 and 1989 respectively, he received the B.S. and M.S. in Electrical Engineering from Nanjing Institute of Posts and Telecommunications in Nanjing, China. From 1984 through 1986 and from 1989 through 1991, he was a lecturer in the Department of Telecommunications at that university, teaching courses in communication theory and digital signal processing. From 1992 through 1994, he was a research assistant in the Image Processing and Neural Networks Laboratory in the Department of Electrical Engineering at the University of Texas at Arlington. He earned his M.S.E.E. in December 1994. Since 1995 he has been with Worldcom Inc. in Tulsa Oklahoma, where he performs software development for telecommunications applications. His current research interests include telecommunications, neural networks, DSP, and image processing.