

Optimal Pruning of Feedforward Neural Networks Based upon the Schmidt Procedure.

F. J. Maldonado
Williams-Pyro, Inc.
200 Greenleaf Street
Fort Worth, TX 76107

M. T. Manry
University of Texas at Arlington
Electrical Engineering Dept
Nedderman Hall, Rm 501
416 Yates Street
Arlington, TX 76010

Abstract.

A common way of designing feed forward networks is to obtain a large network and then to prune less useful hidden units. Here, two non-heuristic pruning algorithms are derived from the Schmidt procedure. In both, orthonormal systems of basis functions are found, ordered, pruned, and mapped back to the original network. In the first algorithm, the orthonormal basis functions are found and ordered one at a time. In optimal pruning, the best subset of orthonormal basis functions is found for each size network. Simulation results are shown.

1. Introduction

The most common approaches for obtaining neural network topologies are growing methods and pruning methods. In growing methods, new hidden units are added during the training process [4][12]. A drawback of growing methods is that the network can get trapped in local minima and they are sensitive to initial conditions. In pruning methods, a large network is trained and then less useful nodes or weights are removed [11][1][8][15].

Pruning methods can be classified as sensitivity based methods and penalty-term methods. In sensitivity methods the error sensitivity to the removal of an element is estimated, and according to this is selected the elements to remove. In penalty-term methods a cost function is used that drives unnecessary weights nearly to zero. Since the cost function may include sensitivity terms there could be some overlap between both methods.

In this work pruning methods are presented that use a modified version of the Gram-Schmidt procedure [18] to obtain the Multilayer Perceptron (MLP) configuration with good performance. It is explained for the case of a three layer MLP with full connectivity [5]. The process starts using a similar approach to the method reported by Kaminsky and Strumillo [18] for Radial Basis Functions. Then using a modified Schmidt procedure the system is

transformed into a new system that consists of linearly independent and orthonormal basis functions. The new system is expressed in terms of the estimated autocorrelation and crosscorrelation matrix elements of the original neural network. Then taking advantage of the orthonormal condition an expression is obtained for the output weights. Using these results the error of the MLP can be calculated for different size hidden layers, and also for different combinations of hidden units. Using these results a method for evaluating basis functions is defined and algorithms for getting the best configuration of the MLP are given. Once the hidden units (orthonormal basis functions) are selected their weights are transformed into the final form of the network.

The paper is organized as follows way. In section 2 the MLP and the required notation are defined. In section 3 a modified Schmidt process is presented as well as necessary weight transformations. Sections 4 and 5 explain basic and optimal pruning respectively. Results are given in section 6. Final comments and conclusions are given in section 7.

2. Multilayer perceptron and modified Schmidt procedure

Multilayer perceptron

Figure 1 depicts feedforward MLP, having one hidden layer with N_h nonlinear units and an output layer with M linear units. From Figure 1, the net value ($net_{p,j}$) and the output value ($O_{p,j}$) for the j^{th} -hidden unit for the p^{th} training pattern are defined as $O_{p,j} = f(net_{p,j})$ with

$$net_{p,j} = \sum_{i=1}^{N+1} w(j,i) \cdot x_{p,i} \quad 1 \leq p \leq N_v, \quad 1 \leq j \leq N_h \quad (1)$$

Here the threshold of the j^{th} node is handled by letting $x_{p,N+1}$ be one. Weight $w(j,i)$ connects the i^{th} input to the j^{th} hidden unit.

For the MLP the most common activation functions are the sigmoid function for hidden layers and linear

functions for the output layer. In a two layer MLP the j^{th} output in the hidden layer is given by,

$$O_{pj} = f(\text{net}_{pj}) = \frac{1}{1 + e^{-\text{net}_{pj}}} \quad (2)$$

The k^{th} output for the p^{th} pattern is,

$$y_{pk} = \sum_{i=1}^{N+1} w_o(k, i) \cdot x_{pi} + \sum_{j=1}^{N_h} w_o(k, j + N + 1) \cdot O_{pj} \quad (3)$$

where $1 \leq k \leq M$. There are N_v training patterns denoted by $\{(x_p, t_p)\}_{p=1}^{N_v}$ where each pattern consists in an input vector x_p and a desired output vector t_p . For the p^{th} pattern, the N input values are x_{pi} ($1 \leq i \leq N$) and the M desired output values are t_{pk} ($1 \leq k \leq M$).

Example training algorithms are Backpropagation [14], Output Weight Optimization – Hidden Weight Optimization [3], and Genetic Algorithms [10][2][6]. The mapping error for the p^{th} pattern is

$$E_p = \sum_{k=1}^M [t_{pk} - y_{pk}]^2 \quad (4)$$

In order to train a neural network, for one epoch the mapping error for the i^{th} output unit is defined as

$$E(i) = \frac{1}{N_v} \sum_{p=1}^{N_v} [t_{pi} - y_{pi}]^2 \quad (5)$$

The overall performance of a MLP network, measured as Mean Square Error (MSE), can be written as

$$E = \sum_{i=1}^M E(i) = \frac{1}{N_v} \sum_{p=1}^{N_v} E_p \quad (6)$$

3. Schmidt procedure for Neural Nets

The output of the network in (3), can be rewritten as

$$y_i = \sum_{k=1}^{N_u} w_o(i, k) \cdot x_k \quad (7)$$

where $x_k = O_{p(k-N-1)}$ for $N+1 < k \leq N_u$ where N_u is the total number of units equal to $N + N_h + 1$. In equation (7), the signals x_k are the raw basis functions for producing y_i .

The normal Gram-Schmidt procedure [17] is a recursive process that requires obtaining scalar products between raw basis functions and orthonormal basis functions. The disadvantage in this process is that it requires one pass through the training data to obtain each new basis function. In this section a more useful form of the Schmidt process is reviewed, which will let us express the orthonormal system in terms of autocorrelation elements.

Basic Algorithm

The m^{th} orthonormal basis function x_m' , can be expressed as

$$x_m' = \sum_{k=1}^m a_{mk} x_k \quad (8)$$

From equation (8), for $m = 1$, the first basis function is obtained as

$$x_1' = \sum_{k=1}^1 a_{1k} x_k = a_{11} x_1 \quad (9)$$

$$a_{11} = \frac{1}{\|x_1\|} = \frac{1}{r(1,1)^{1/2}} \quad (10)$$

where

$$r(i, j) = \langle x_i, x_j \rangle = \left(\frac{1}{N_v} \right) \sum_{p=1}^{N_v} x_{pi} x_{pj} \quad (11)$$

For values of m between 2 and N_u , c_i is first found for $1 \leq i \leq m-1$ as

$$c_i = \sum_{q=1}^i a_{iq} r(q, m) \quad (12)$$

Then obtain m coefficients b_k as,

$$\begin{cases} b_k = -\sum_{i=k}^{m-1} c_i a_{ik} & 1 \leq k \leq m-1 \\ b_m = 1 \end{cases} \quad (13)$$

Finally for the m^{th} basis function the new a_{mk} coefficients (for $1 \leq k \leq m$) are found as

$$a_{mk} = \frac{b_k}{\left[r(m, m) - \sum_{i=1}^{m-1} c_i^2 \right]^{1/2}} \quad (14)$$

Equating y_i in (7) to

$$y_i = \sum_{q=1}^{N_u} w_o'(i, q) x_q' \quad (15)$$

where the weights in the orthonormal system are

$$w_o'(i, q) = \sum_{k=1}^q a_{qk} \langle x_k, t_i \rangle = \sum_{k=1}^q a_{qk} c(i, k) \quad (16)$$

and using (8) we obtain output weights for the system as

$$w_o(i, k) = \sum_{q=k}^{N_u} w_o'(i, q) a_{qk} \quad (17)$$

Substituting (15) into (5) we obtain $E(i)$ in the orthonormal system as

$$E(i) = \left\langle \left(t_i - \sum_{k=1}^{N_u} w_o'(i, k) x_k' \right), \left(t_i - \sum_{q=1}^{N_u} w_o'(i, q) x_q' \right) \right\rangle \quad (18)$$

If we decide to use the first N_{hd} hidden units in our original network, the training error is

$$E(i) = \langle t_i, t_i \rangle - \sum_{k=1}^{N+1+N_{hd}} (w_o'(i, k))^2 \quad (19)$$

Modifying (17) the output weights would be

$$w_o(i, k) = \sum_{q=k}^{N+1+N_{hd}} w_o'(i, q) a_{qk} \quad (20)$$

4. Ordered pruning

In section 3, no attempt is made to order the hidden units according to their usefulness. In this section we modify the Schmidt procedure so that during pruning useless basis functions x_m' are eliminated.

Let $j(m)$ be an integer valued function that specifies the order in which raw basis functions x_k are processed into orthonormal basis functions x_k' . Then x_m' is to be calculated from $x_{j(m)}$, $x_{j(m-1)}$ and so on. This function also

defines the structure of the new hidden layer where $1 \leq m \leq N_u$ and $1 \leq j(m) \leq N_u$. If $j(m) = k$ then the m^{th} unit of the new structure comes from the k^{th} unit of the original structure.

Given the function $j(m)$, and generalizing section 3, the m^{th} orthonormal basis function is described as

$$\dot{x}_m = \sum_{k=1}^m a_{mk} x_{j(k)} \quad (21)$$

Initially, \dot{x}_1 is found as $a_{11} x_{j(1)}$ where

$$a_{11} = 1/\|x_{j(1)}\| = 1/r(j(1), j(1))^{1/2} \quad (22)$$

For $2 \leq m \leq N_u$, we first perform

$$c_i = \sum_{q=1}^i a_{iq} r(j(q), j(m)) \quad , \quad (23)$$

for $1 \leq i \leq m-1$. Second, we set $b_m = 1$ and get

$$b_k = -\sum_{i=k}^{m-1} c_i a_{ik} \quad , \quad (24)$$

for $1 \leq k \leq m-1$. Lastly, we get coefficients a_{mk} as

$$a_{mk} = \frac{b_k}{\left[r(j(m), j(m)) - \sum_{i=1}^{m-1} c_i^2 \right]^{1/2}} \quad , \quad (25)$$

for $1 \leq k \leq m$. $w_o'(i, k)$ is found as

$$w_o'(i, m) = \sum_{k=1}^m a_{mk} c(i, j(k)) \quad 1 \leq i \leq M \quad (26)$$

Basic pruning.

The goal of pruning is to find the function $j(m)$ which defines the structure of the hidden layer. Here it is assumed that the original basis functions are linearly independent i.e. the denominator of equation (25) is not zero.

Since we want the effects of inputs and the constant “1” to be removed from orthonormal basis functions, the first $N+1$ basis functions are picked as,

$$j(m) = m \quad \text{for} \quad 1 \leq m \leq N+1 \quad , \quad (27)$$

The selection process will be applied to the hidden units of the network. We now define notation that helps us specify the set of candidate basis function to choose in a given iteration. First, define $S(m)$ as the set of indices of chosen basis functions where m is the number of units of the current network (i.e. the one that the algorithm is processing). Then $S(m)$ is given by

$$S(m) = \begin{cases} \{\emptyset\} & \text{for } m = 0 \\ \{j(1), j(2), \dots, j(m)\} & \text{for } 0 < m \leq N_u \end{cases} \quad (28)$$

Starting with an initial linear network having 0 hidden units, where m is equal to $N+1$, the set of candidate basis functions is clearly $S^c\{m\} = \{1, 2, 3, \dots, N_u\} - S(m)$, which is $\{N+2, N+3, \dots, N_u\}$. For $N+2 \leq m \leq N_u$, we obtain $S^c(m-1)$. For each trial value of $j(m) \in S^c\{m-1\}$ we perform operations (23), (24), (25), and (26). Then $P(m)$ is

$$P(m) = \sum_{i=1}^M [w_o'(i, m)]^2 \quad (29)$$

The trial value of $j(m)$ that maximizes $P(m)$ is found. Assuming that $P(m)$ is maximum when testing the i^{th} element, then $j(m) = i$. $S(m)$ is updated as

$$S(m) = S(m-1) \cup \{j(m)\} \quad (30)$$

Then for the general case the candidate basis functions are, $S^c(m-1) = \{1, 2, 3, \dots, N_u\} - \{j(1), j(2), \dots, j(m-1)\}$ with $N_u - m + 1$ candidate basis function. By using equation (29) after testing all the candidate basis function, $j(m)$ takes its value and $S(m)$ is updated according to equation (30). Defining N_{hd} as the desired number of units in the hidden layer, the process is repeated until $m = N+1+N_{hd}$. Then the orthonormal weights are mapped to normal weights according to equation (20).

5. Optimal Pruning

As in the basic pruning case the optimal pruning algorithm uses the first $N+1$ units (inputs and threshold) without reordering, so that $j(m) = m$ for $1 \leq m \leq N+1$. The first $N+1$ orthonormal basis functions are generated by obtaining the corresponding a_{nk} coefficients (with $1 \leq n \leq N+1$ and $1 \leq k \leq n$) according to equations (25), (23) and (24). In addition, set $S(m)$ ($m = N+1$) is given by equation (28).

Next the algorithm looks for the optimal hidden units of the network, where the unknowns are the values of $j(N+2)$ to $j(N+1+N_{hd})$. The initial set of candidate basis functions is $S^c(N+1)$ with N_h elements.

Then N_{hd} defines the basis function set size for the algorithm. In testing the candidate basis function set, all combinations of N_{hd} hidden units are taken from $S^c(N+1)$. In optimal pruning we test all possible subsets of N_{hd} hidden units.

Defining n_c as the number of orderings of N_{hd} hidden units, the number of networks to test is given by

$$n_c = \binom{N_h}{N_{hd}} \quad (31)$$

For each candidate network the energy is estimated as

$$P_{total} = \sum_{q=1}^{N_{hd}} P(m_q) = \sum_{q=1}^{N_{hd}} \left(\sum_{i=1}^M [w_o'(i, m_q)]^2 \right) \quad (32)$$

where each value of $j(m_q)$ will correspond to a hidden unit that forms the network being tested. Here m_q defines the q^{th} element of the combination being tested.

Then the final solution will be the set of hidden units that maximize equation (32). In this way $j(m)$ take its final value and $S(N+1+N_{hd})$ is updated as

$$S(N_{hd}) = S(N+1) \cup \{j(m_1), j(m_2), \dots, j(m_{N_{hd}})\} = \{j(1), j(2), \dots, j(N+1), \dots, j(N+1+N_{hd})\} \quad (33)$$

Once that $j(m)$ and a_{nk} ($1 \leq n \leq N+1+N_{hd}$, $1 \leq k \leq n$) are found the final output weights $w_o(i,k)$ are obtained by using equation (20).

6. Results

The pruning algorithms were tested using training data corresponding to four cases. In each case we trained a network (100 epochs) used it as input to the algorithms. The first example corresponds to the task of inverting the surface scattering parameters from an inhomogeneous layer above a homogeneous half space [7], where both interfaces are randomly rough. The network has 8 inputs and 7 outputs, and the file has 1768 training patterns. The inputs consist of eight theoretical values of back scattering coefficient parameters at V and H polarization and four incident angles. The outputs were the corresponding values of permittivity, upper surface height, lower surface height, normalized upper surface correlation length, normalized lower surface correlation length, optical depth and single scattering albedo which had a jointly uniform pdf. The second case corresponds to a neural network that performs demodulation of an FM (frequency modulation) signal containing a sinusoidal message [13]. The third plot was generated using data obtained from TU Electric Company in Texas where the first ten input features are the last ten minute power load in megawatts for the entire TU Electric utility [9]. The desired output is power load fifteen minutes in the future from the current time. All powers were originally sampled every fraction of a second, and averaged over 1 minute to reduce noise. Finally the last case corresponds to the inversion of radar scattering [16]. The training set contains VV and HH polarization at L 30, 40 deg, C 10, 30, 40, 50, 60 deg, and X 30, 40, 50 deg along with the corresponding unknowns rms surface height, surface correlation length, and volumetric soil moisture content in g/cubic cm.

From the figures, we see that optimal pruning is better than basic pruning.

7. Conclusions

In this paper, two pruning algorithms are given for the MLP. Optimal pruning shows better performance than simple pruning but is much more computationally expensive. The modified orthogonalization procedure is more efficient than normal Gram-Schmidt method and provides a more suitable framework for pruning.

8. Acknowledgement.

This work was funded by the Advanced Technology Program of the state of Texas under grant 003656-0129-2001.

References.

- [1] H. Amin, K. M. Curtis, B. R. Hayes Gill, "Dynamically Pruning Output Weights in an Expanding Multilayer Perceptron Neural Network", *Digital Signal Processing Proceedings, 1997 13th International Conference on DSP*, 1997, vol. 2, pp. 991-994.
- [2] P. Arena, R. Caponetto, L. Fortuna, M. G. Xibilia, "Genetic Algorithms to select optimal neural network topology", in *Proc. of the 35th Midwest Symposium on Circuits and Systems*, 1992, vol. 2, pp. 1381-1383.
- [3] H. H. Chen, M. T. Manry and H. Chandrasekaran, "A neural network training algorithm utilizing multiple sets of linear equations," *Neurocomputing*, vol. 25, no. 1-3, pp. 55-72, Apr. 1999.
- [4] F. L. Chung and T. Lee, "Network-growth approach to design of feedforward neural networks", *IEEE Proceedings Control Theory*, 142-5, 486-492 (1995).
- [5] G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Math. Cont. Signal Syst.*, vol. 2, pp. 303-314, 1989.
- [6] J. Davila, "Genetic optimization of NN topologies for the task of natural language processing", *International Joint Conference on Neural Networks IJCNN'99*, 1999, vol. 2, pp. 821-826.
- [7] M.S. Dawson, J. Olvera, A.K. Fung and 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.
- [8] Y. Hirose, K. Yamashita, and S. Hijiya, "Back-propagation Algorithm that varies the number of Hidden Units", *Neural Networks*, vol. 4, pp. 61-66, 1991.
- [9] K. Liu, S. Subbarayan, R.R. Shoultz, M.T. Manry, C.Kwan, F.L. Lewis, J.Naccarino, "Comparison of Very Short-Term Load Forecasting Techniques," *IEEE Transactions on Power Systems*, vol.11, no.2, May 1996, pp. 877-882.
- [10] V. Maniezzo, "Genetic evolution of the topology and weight distribution of Neural Networks", *IEEE Transaction on Neural Networks*, 1994, vol. 5, No. 1, pp. 39-53.
- [11] Ponnappalli, "A formal selection and pruning algorithm for feedforward artificial network optimization", *IEEE Transaction on Neural Networks*, 1999, vol. 10, No. 4, pp. 964-968.
- [12] R. Reed, "Pruning Algorithms - A Survey:", *IEEE Transaction on Neural Networks*, 1993, vol. 4, No. 5, pp. 740-747.
- [13] K. Rohani, M.T. Manry, "The Design of Multi-Layer Perceptrons using Building Blocks," *Proc of IJCNN 91, Seattle WA.*, pp. II-497 to II-502.
- [14] D. E. Rumelhart and J. L. McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA: MIT Press, 1986.
- [15] I. I. Sakhnini, M. T. Manry, and H. Chandrasekaran, "Iterative improvement of trigonometric networks", *International Joint Conference on Neural Networks (IJCNN'99)*, July 1999.
- [16] Y., K. Sarabandi, F.T. Ulaby, "An Empirical Model and an Inversion Technique for Radar Scattering from Bare Soil Surfaces," in *IEEE Trans. on Geoscience and Remote Sensing*, pp. 370-381, 1992.

- [17] G. Strang, *Linear Algebra and its application*. New York: Harcourt Brace, 1988.
- [18] Wladyslaw Kaminski, Pawel Strumillo “Kernel Orthonormalization in Radial Basis Function Neural Networks”, *IEEE Transactions on Neural Networks*, vol. 8, No. 5, pp. 1177 – 1183, 1997.

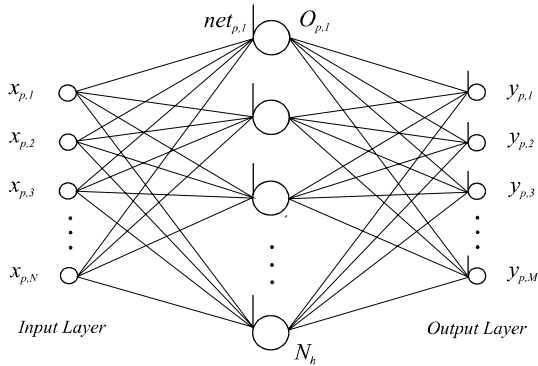


Figure 1. Artificial neural network, two layers MLP.

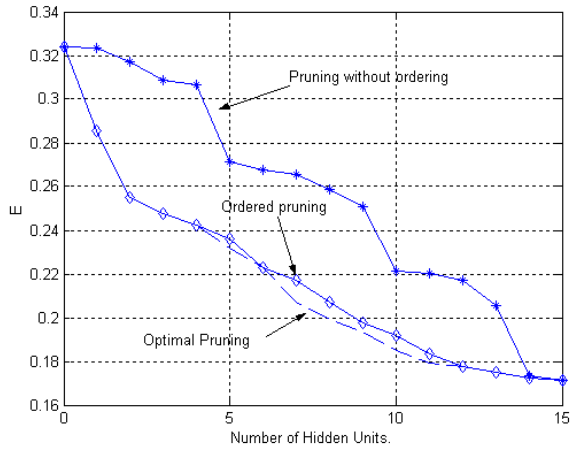


Figure 2. Inversion of Surface scattering parameters.

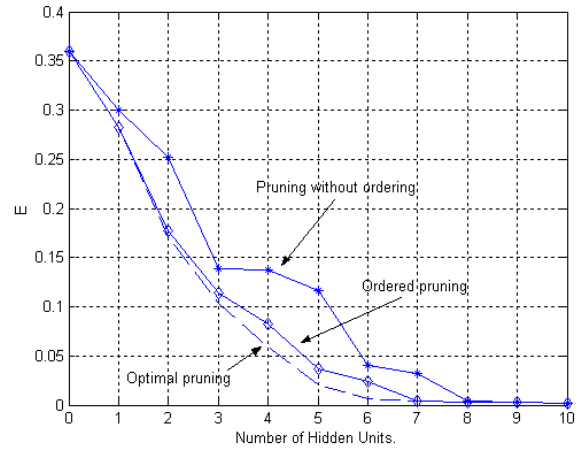


Figure 3. Demodulation of an FM signal

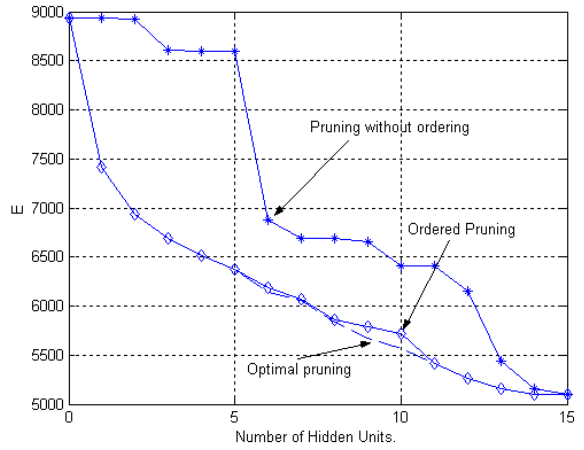


Figure 4. Electrical Power load forecasting.

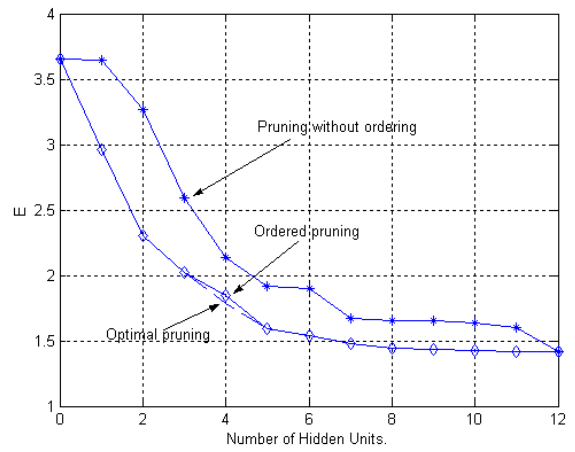


Figure 5. Radar scattering