Iterative Improvement of Image Classifiers Using Relaxation

by

L.M. Liu, M.T. Manry, F. Amar, M.S. Dawson and A.K. Fung

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

Abstract

A new objective function for neural net classifier design is presented, which has more free parameters than the classical objective function. An iterative minimization technique for the objective function is derived which requires the solution of multiple sets of numerically ill-conditioned linear equations. A numerically stable solution to the neural network design equations, which utilizes the conjugate gradient algorithm and a relaxation algorithm, is presented. The design method is applied to networks used to classify SAR imagery from remote sensing. The improvement of the iterative technique over classical design approaches is clearly demonstrated.

I. Introduction

Two commonly used neural network classifiers are the functional link neural network (FLNN) [1] and the multilayer perceptron (MLP) neural network [2]. The MLP and FLNN approximate the general Bayes discriminant [3,4]. FLNNs and MLPs are designed by minimizing the standard training error,

$$E = \sum_{k=1}^{N_c} E(k) \tag{1}$$

where N_c is the number of classes and $E(k)$, the mean-squared error for the kth output, is defined as

$$E(k) \equiv \frac{1}{N_v} \sum_{p=1}^{N_v} [\, t_p(k) - O_p(k)]^2 \tag{2}$$

where $t_p(k)$ denotes the kth desired output for the pth input pattern, $O_p(k)$ denotes the kth observed output for the pth input pattern, and N_v denotes the total number of training patterns. In this paper, we assume that $t_p(i_c) = b$ and $t_p(i_d) = -b$ where i_c denotes the correct class number for the current training pattern, i_d denotes any incorrect class number for that pattern, and b is a positive constant.

Generally, the output of a neural network classifier can be written as

$$O_p(k) = \sum_{j=1}^{N_u(k)} w(k, j)\, X_p(j) \tag{3}$$

where $w(k,j)$ denotes the weight connecting the jth unit to the kth output unit, and $N_u(k)$ denotes the number of units feeding signals to the kth output unit. In the FLNN, $X_p(k)$ may represent a multinomial combination of the N inputs x_n whereas in the MLP, $X_p(k)$ may represent a sigmoidal activation for a hidden unit. Both the MLP and the FLNN can be designed by solving linear sets of equations [1,5-11]. A conjugate gradient (CG) solution to these sets of equations has been given that works, even though the equations are ill-conditioned [7-11].

The error function E in (1) is too restrictive in at least two ways. First, if each individual output vector has a different constant bias add to it, E could be increased or decreased, but this will have no effect on the classification error. Second, if an output has the correct sign but a magnitude larger than b, E will increase while the classification error will be unaffected. In order to take advantage of these facts, we have developed the Output Reset (OR) algorithm, which uses a relaxation approach in combination with CG. In section II the OR algorithm is derived and theorems concerning the algorithm are given. The applications of OR-designed MLPs is demonstrated in section III.

II. The output reset algorithm

A. Derivation

In the OR algorithm we (1) give each output vector the specific bias which minimizes E, and (2) set the desired output equal to the actual output when the output has the correct sign but is larger than b. The error function

1058-6393/95 $4.00 © 1995 IEEE

E can be modified as

$$E' \equiv \frac{1}{N_v} \sum_{p=1}^{N_v} \sum_{i=1}^{N_c} [\, t'_p(i) - O_p(i) \,]^2 \qquad (4)$$

where $t_p'(i) = t_p(i) + a_p + d_p(i)$ and where $d_p(i)$ is a function of p and i to be defined later. Our goal is to find a_p, $d_p(i)$ and $O_p(i)$ that minimize E', under the following conditions :

Condition (1); The difference $| t_p'(i_c) - t_p'(i_d) |$ must be larger than or equal to 2b. Without this condition, E' can be minimized by letting the network weights and the difference $| t_p'(i_c) - t_p'(i_d) |$ be equal to zero.

Condition (2); Each change made to a_p, $d_p(i)$ and $O_p(i)$ (through changes in the network weights), must reduce E'.

Method 1. Changes to a_p

In order to minimize E' with respect to a_p, it is sufficient that the first derivative of E' with respect to a_p be zero, yielding

$$a_p = \frac{1}{N_c} \sum_{i=1}^{N_c} [O_p(i) - t_p(i) - d_p(i)] \qquad (5)$$

After adding a_p to each desired output, the distance between the desired outputs is the same as before. Therefore, the classification performance remains the same, so condition (1) is satisfied. Since a_p is specifically found to minimize E', condition (2) is also satisfied.

Method 2. Changes to $d_p(i)$

Ignoring condition (1), $d_p(i)$ can be found such that the term $[t_p(i)+a_p+d_p(i)-O_p(i)]^2$ is zero, yielding $d_p(i) = O_p(i) - t_p(i) - a_p$, which satisfies condition (2). However, in order to satisfy condition (1), we modify $d_p(i)$ such that $d_p(i_c) \geq 0$, $d_p(i_d) \leq 0$. In summary,

(a) If $O_p(i_c) \geq t_p(i_c) + a_p$
then choose $d_p(i_c) = O_p(i_c) - t_p(i_c) - a_p$

(b) If $O_p(i_d) \leq t_p(i_d) + a_p$
then choose $d_p(i_d) = O_p(i_d) - t_p(i_d) - a_p$

(c) Otherwise, choose $d_p(i_c) = 0$ or $d_p(i_d) = 0$.

Method 3. Changes to $O_p(i)$

After a_p, $d_p(i)$ are found, we minimize E' with respect to the output weights using *Output Weight Optimization* (OWO) [7-11]. Condition (1) is satisfied since $t'_p(i)$ is not changed. Condition (2) is obviously satisfied. It is possible to minimize E' with respect to all of the network's weights [7,8]. However, in this paper, we choose instead to describe and analyze, in more detail, our approach for finding output weights.

B. Theory

Theorem 1. The Output Reset (OR) algorithm leads to convergence of the error function E'.
Proof. The three methods above satisfy condition (2).

In the following derivation, x denotes the input feature vector to be classified. The Bayes discriminant function is defined as $g_x(i) = P(C_i|x)$ [3], which is the probability that the input pattern x belongs to the class C_i. An error function can be defined as

$$e \equiv 4b^2 \sum_{i=1}^{N_c} \int_S [\, F'_x(i) - g_x(i) \,]^2 \, p(x)dx \qquad (6)$$

where S is the set of all input patterns, $F'_x(i)$ is defined as

$$F'_x(i) \equiv \frac{1}{2b} O'_p(i) + \frac{1}{2} \qquad (7)$$

and $p(x)$ is the joint probability density function of the random input vector x. The error function E' is rewritten from eq.(4),

$$E' = \frac{1}{N_v} \sum_{p=1}^{N_v} \sum_{i=1}^{N_c} [\, t_p(i) - O'_p(i) \,]^2 \qquad (8)$$

where $O_p'(i) = O_p(i) - a_p - d_p(i)$.

Theorem 2. As N_v increases, the error function E' approaches e, i.e.

$$\lim_{N_v \to \infty} E' = e + C \qquad (9)$$

Proof. Taking the limit of $E'/4b^2$, as the number of input pattern is large,

$$\lim_{N_v \to \infty} \frac{E'}{4b^2} = \lim_{N_v \to \infty} \frac{1}{N_v} \sum_{p=1}^{N_v} \sum_{i=1}^{N_c} [\, \frac{1}{2b} (t_p(i) - O'_p(i)) \,]^2 \qquad (10)$$

Substituting eq.(7) and $t_p(i)$ into eq.(10),

$$\lim_{N_v \to \infty} \frac{E'}{4b^2}$$

$$= \lim_{N_v \to \infty} \frac{1}{N_v} \left(\sum_{x \in S_1} [\ (1 - F_x'(1))^2 + \sum_{j \neq 1} (0 - F_x'(j))^2\ ] \right.$$

$$\left. + \cdots + \sum_{x \in S_{N_c}} [\ (1 - F_x'(N_c))^2 + \sum_{j \neq N_c} (0 - F_x'(j))^2\ ] \right)$$

where S_i denotes the set of input vectors x belonging to the ith class. By following the derivation of Ruck et al.[3] for the multiclass case, we obtain eq.(9), with the constant C defined as

$$C = 4b^2 \sum_{i=1}^{N_c} \int_S g_x(i)\ (1 - g_x(i))\ p(x)dx$$

From Theorem 1 and Theorem 2, we conclude that, if the MLP is large enough, is trained properly and the training set is large enough, the neural net is a good approximation to Bayes classifier.

C. Final output reset algorithm

Using the three methods derived in subsection A, the final OR algorithm is summarized as follows.

(i) Find and store the desired outputs $t_p(i)$.
(ii) For each input vector x,
(a) Calculate the network outputs $O_p(i)$ using the current weight matrix.
(b) Find $t'_p(i)$, iteratively, using methods (1) and (2) from subsection A.
(c) Accumulate the autocorrelations and crosscorrelations needed in the OWO algorithm.
(iii) Find the output weights $w(k,j)$ using OWO. Go back to step (ii) for another iteration, if desired.

In order to compare the performances of the first two methods in subsection A, we propose three versions of the OR algorithm. Algorithm 1 uses methods (1) and (3) from subsection A. Algorithm 2 uses methods (2) and (3) from subsection A. Algorithm 3 iteratively uses all three methods, as given above.

III. Remote sensing application

To test the performance of our design algorithm, we applied an MLP classifier for synthetic aperture radar (SAR) data from three sites, obtained by the NASA/JPL AIRSAR system during the MAESTRO-1 campaign in the summer of 1989 [8-11]. The Flevoland test site is located in Southern Flevoland in the Netherlands. The area is characterized by homogeneous soils and a variety of crop types. The 12 input features used in the classifiers include vertical-vertical (VV), vertical-horizontal (VH), 2and horizontal-horizontal (HH) polarizations and also the total power (TP) from the P, L, and C-bands. The 13 classes include bare soil, lucerne, water, forest, summer barley, flood, potatoes, stem beam, grass, red beat, winter wheat, peas, and sugar beat. Our MLP used two hidden layers with 20 units each. Fig. 1 shows the training and testing results for the MLP network. A Bayes-Gaussian classifier was also tried but did not work. The MLP trained down to 1.56% classification error. The testing error had a minimum of about 1.81 %. In Fig. 2, we show the training results for all three algorithms. Clearly, methods 1 and 2 are both beneficial.

The Belize test area is a tropical rain forest located in Central America. The terrain in this area includes rain forest, bare soil, regrowth, clear-cut, farmland, village, and flooded areas. The four input features included C-band signals with VV, VH, and HH polarizations and also the TP. The 7 classes include Bajo, clearcut, regrowth, forest, farmland, laguna seca, and flood. The MLP had two hidden layers with 10 units each. The training results of the MLP are shown in Fig.3 for all three algorithms. The Bayes-Gaussian classifier was tried, but had training and testing errors of 76 % and 48 % respectively, indicating a numerical problem in the classifier design. The MLP results bottomed out around 10 % for training and testing results. After twenty iterations, the classification performance does not improve.

The Freiburg test site is located near a small village called Vinningen in the area surrounding the collegian city of Freiburg in Germany. This site has three main areas: village, agricultural field, and the forest. The 12 input features include VV, VH, HH, TP features from the P, L, and C-bands. The 4 classes include lake, forest, clearcut, and city. The MLP had two hidden layers with 20 units each. The training results of the MLP are shown in Fig. 4 for all three algorithms.

Although the classification error % is mostly decreasing in figures 1, 2, and 3, it is not guaranteed to do so. This is especially clear in Fig. 4. The training MSE for the Flevoland, Belize, and Freiburg data sets are shown in figures 5, 6, and 7 respectively. The curves are monotonically decreasing as expected.

IV. Conclusion

In this paper, we propose some basic improvements to the objective function used in training FLNN and MLP classifiers. The method is analyzed in

detail. The linear classifier design equations have been solved using a numerically stable CG approach. The OR algorithm, which is a type of relaxation algorithm, has been detailed. It allows us to iteratively improve our initial solution to the design equations. The OR approach has been used to design MLP classifiers for SAR imagery. Compared to the Bayes-Gaussian classifier and conventionally-designed MLPs, the OR-designed networks have proved superior.

Acknowledgement

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 a grant from the state of Texas.

References

[1] Yoh-Han Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, 1989.

[2] D.E. Rumelhart, J.L. McClelland, and the PDP research group, *Parallel Distributed Processing : Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, MIT Press, 1988.

[3] D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley and B.W. Suter, " The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminabt Function', *IEEE trans. on Neural Network*, Vol.1, No.4, pp.296-298, Dec.1990.

[4] E. A. Wan, "Neural Network Classification: Bayesian Interpretation", *IEEE trans. on Neural Network*, Vol.1, No.4, pp.303-305, Dec.1990.

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

[6] M.A. Sartori, 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, pp.467-471, July 1991.

[7] 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*, Vol 1, pp.502-506, Oct. 1992.

[8] 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.

[9] M.S. Dawson, J. Olvera, A.K. Fung, M.T. Manry, "Inversion of Surface Parameters Using Fast Learning Neural Networks," *Proceeding of IGARSS '92*, Houston, Texas, May 1992, Vol.II, pp.910-912.

[10] M.S. Dawson, A.K. Fung, M.T. Manry, "Sea Ice Classification Using Fast Learning Neural Networks," *Proceeding of IGARSS '92*, Houston, Texas, May 1992, Vol.II, pp.1070-1071.

[11] R.R. Bailey, E.J. Pettit, R.T. Borochoff, M.T. Manry, and X. Jiang, "Automatic Recognition of USGS Land USE/Cover Categories Using Statistical and Neural Network Classifiers," *Proceeding of SPIE OE/Aerospace and remote sensing*, April 12-16,1993, Orlando Florida.

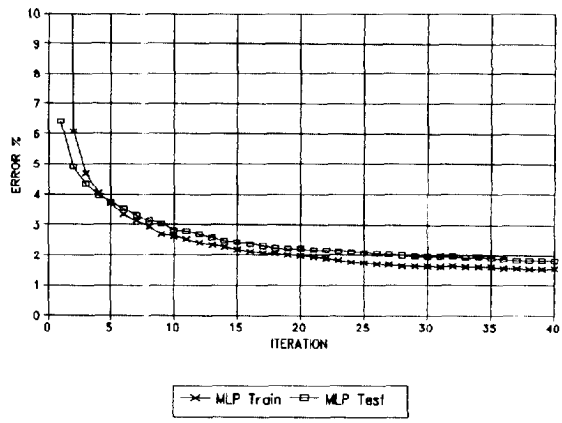

Fig. 1. Flevoland Training and Testing Results

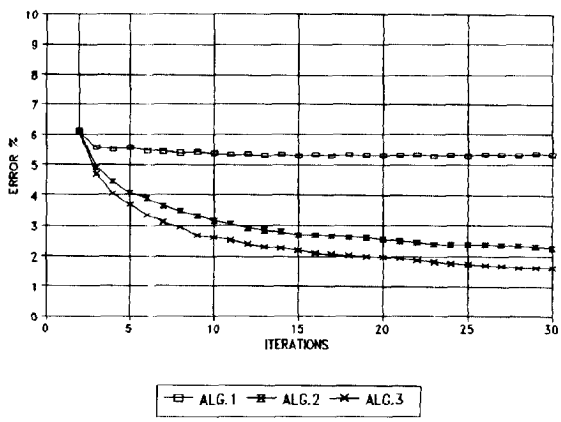

Fig. 2. Training Results for Three Algorithms

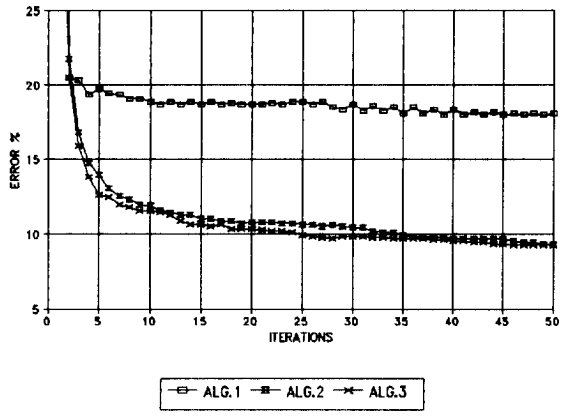

Fig. 3. Belize Training Results

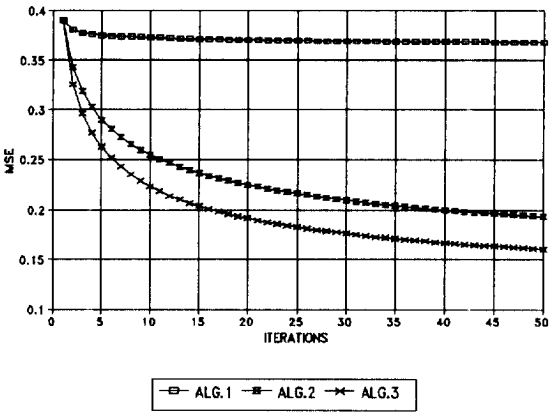

Fig. 6. MSE for Belize Training

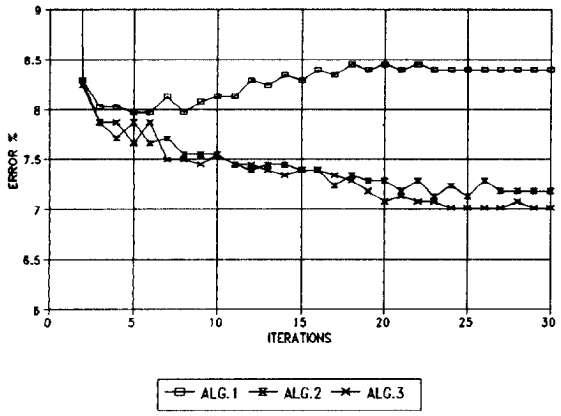

Fig. 4. Freiburg Training Results

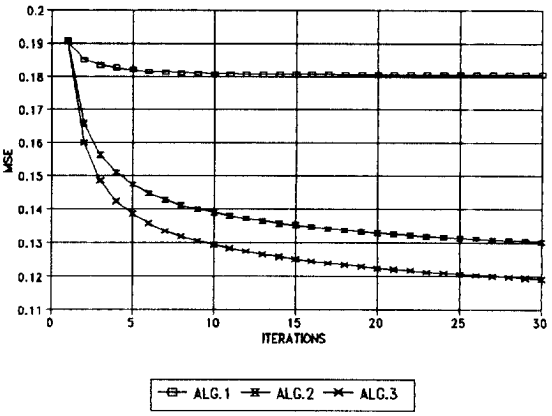

Fig. 7. MSE for Freiburg Training

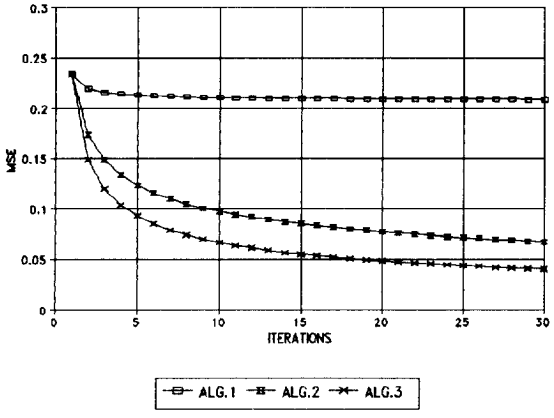

Fig. 5. MSE for Flevoland Training