L.E.Atlas, J. Conner, D.C. Park, M.A. El-Sharkawi, R.J. Marks II, A. Lippman, R. Cole and Y. Muthusamy,
"A performance comparison of trained multi-layer perceptrons and trained classification trees",
Proc. 1989 IEEE International Conference on Systems, Man and Cybernetics,
(Hyatt Regancy, Cambridge, Massachusetts, 14-17 Nov. 1989), pp.915-920.

A PERFORMANCE COMPARISON OF TRAINED MULTI-LAYER PERCEPTRONS AND TRAINED CLASSIFICATION TREES

Les Atlas, Jerome Connor, Dong Park,
Mohamed El-Sharkawi, and Robert Marks II

Dept. of Electrical Engrg., FT-10
University of Washington
Seattle, WA 98195

Alan Lippman

Dept. of Statistics, GN-22
University of Washington
Seattle, WA 98195

Ronald Cole and Yeshwant Muthusamy

Dept. of Computer Science & Engrg.
Oregon Graduate Center
Beaverton, OR 97006

ABSTRACT

Multi-layer perceptrons and trained classification trees are two very different techniques which have recently become popular. Given enough data and time, both methods are capable of performing arbitrary non-linear classification. These two techniques, which developed out of different research communities, have not been previously compared on real-world problems. We first consider the important differences between multi-layer perceptrons and classification trees and conclude that there is not enough theoretical basis for the clear-cut superiority of one technique over the other. For this reason, we performed a number of empirical tests on quite different problems in power system load forecasting and speaker-independent vowel identification. We compared the performance for classification and prediction in terms of accuracy outside the training set. In all cases, even with various sizes of training sets, the multi-layer perceptron performed as well as or better than the trained classification trees. We are confident that the univariate version of the trained classification trees do not perform as well as the multi-layer perceptron. More studies are needed, however, on the comparative performance of the linear combination version of the classification trees.

I. INTRODUCTION

Many real-world problems are difficult to represent by specific analytical relationships or rules. It has been the hope of many researchers that these problems could be amenable to training by example. Namely, the full input/output relationships could be captured via a set of examples (training data) and these examples could then be used to train a system to automatically classify or estimate unknown outputs associated with new inputs (test data). In this paper we study both regression and classification systems. A regression system can generate an output Y for an input X, where both X and Y are continuous and, perhaps, multi-dimensional. A classification system can generate an output class, C , for an input X, where X is continuous and multi-dimensional and C is a member of a finite alphabet.

The use of trained systems has been studied by many researchers in the past (e.g. [1-4]). However, there has been a recent surge of interest in trainable classifiers as manifest by artificial neural networks (ANN's). In particular the multi-layer perceptron (MLP) has been shown to be able to be trained by example to solve the non-linearly separable exclusive-OR problem [5], and this architecture has been linked to previous neural-like processors [6, 7]. Less known to the engineering community is the statistical technique of classification and regression trees (CART) which was developed during the years 1973 [8] through 1984 [9]. As we will describe in the next section, CART can also be trained to solve the exclusive-OR problem and, furthermore, the solution it provides is extremely easy to interpret. There have been no links made between CART and biological neural networks. However, the possible applications and paradigms used for MLP and CART are very similar.

The authors of this paper represent diverse interests in problems which have the commonality of being very important and potentially well-suited for trainable classifiers. The load forecasting problem, which is partially a regression problem, uses past load trends to predict the critical needs of future power generation. The vowel recognition problem is representative of the difficulties in automatic speech recognition due to variability across speakers and phonetic context.

In all cases large amounts of real data were used for training and disjoint large data sets were used for testing. We were very careful to ensure that the experimental conditions were identical for the MLP and CART. We concentrated only on performance as measured in error on the test set and did not do any formal studies of training or testing time. (CART was, in general, quite a bit faster).

In all cases, even with various sizes of training sets, the multi-layer perceptron performed as well as or better than the trained classification trees. We also believe that integration of many of CART's well-designed attributes into MLP architectures could only improve the already promising performance of MLP's.

II. BACKGROUND

A. Multi-Layer Perceptrons

The name "artificial neural networks" has in some communities become almost synonymous with multi-layer perceptrons (MLP's) trained by back-propagation. Our power studies made use of this standard algorithm and our vowel studies made use of a conjugate gradient version [10] of back-propagation. In all cases the training data consisted of ordered pairs {(X,Y)} for regression, or {(X,C)} for classification. The input to the network is X and the output is, after training, hopefully very close to Y or C. The network consists of a number of "neuron-like" units which multiply neural inputs by weights, sum the products and then pass through an instantaneous sigmoid nonlinearity. Some of these units connect to elements of X. The distinctive feature of multi-layer perceptrons is that not all units connect

CH2809-2/89/0000-0915 $1.00 © 1989 IEEE

to elements of X; these "hidden units" connect to outputs of other units, thus giving a multi-layer architecture. The goal of training is to best estimate the multiplicative weights to make the network outputs as close (usually a square-error measure) to Y or *C* as possible.

The usual back-propagation training equations can be seen in Rumelhart et al [5] or in a tutorial by Lippmann [11]. A typical paradigm for training is to initialize all weights randomly and to then update the weight based on a repetitive scanning through the training set. When the error within the training set is considered low enough, the network is considered to be "trained" and ready for testing. In some cases training also involves the choice of network topology, e.g., number of layers and hidden units. In our experiments we were careful to make these design choices only with our training data, thus avoiding positively biased results on the test sets.

When MLP's are used for regression, the output values, Y, can take on real values between 0 and 1. This normalized scale was used as the prediction value in the power forecasing problem. For MLP classifiers the output is formed by taking the (0, 1) range of the output neurons and either thresholding or finding a peak. For example, the vowel study used the maximum of the 12 output neurons to determine the vowel class.

B. Classification and Regression Trees (CART)

CART has already proven to be useful in diverse applications such as radar signal classification, medical diagnosis, and mass spectra classification [9]. Given a set of training examples $\{(X,C)\}$, a binary tree is constructed by sequentially partitioning the p-dimensional input space, which may consist of quantitative and/or qualitative data, into p-dimensional rectangles. The trained classification tree divides the domain of the data into non-overlapping regions, each of which is assigned a class label *C*. For regression, the estimated function is piecewise constant over these regions.

When used for classification, the first step of the CART algorithm is to find a hyper-plane that best separates the training examples. In a classification problem with two classes, $C \in \{A,B\}$, the best possible hyper-plane would be one with all the training examples of class A on one side and all examples of class B on the other. In most cases no ideal "split" of the data is possible. Sensible functions are used to measure the quality of non-ideal splits; by minimizing these functions over all possible splits, the best split is found. An example of such a function for the two class problem is:

$$p_L n_L(A) n_L(B) + p_R n_R(A) n_R(B)$$

where p_L and p_R is the proportion of training points in the "left" and "right" side respectively. $n_L(C)$ and $n_R(C)$ are the number of data from classes C=A or C=B which are placed in the left or right side of the split.

Once the first split of the data space has been made, the next step in CART is to consider the split training examples as two completely unrelated sets -- those examples on the left of the selected hyper-plane, and those on the right. CART then proceeds as in the first step, treating each subset of the training examples independently. A question which had long plagued the use of such sequential schemes was: when should the splitting stop? CART implements a novel, and very clever approach; splits continue until every training example is separated from every other, then a pruning criterion is used to sequentially remove splits.

C. Relative Expectations of CART and MLP

The non-linearly separable exclusive-OR problem is an example of a problem which both CART and the MLP can solve with zero error. The training data is a set of 2-input, 1-output ordered pairs, (X_1,X_2,Y), which represent the binary logic behavior as: (00,0), (01,1), (10,1), and (11,0). The left side of figure 1 shows a trained MLP solution to this problem and the right side shows the very simple trained CART solution. For the MLP the values along the arrows represent trained multiplicative weights and the values in the circles represent trained scalar offset values. For the CART figure, *y* and *n* represent yes or no answers to the trained thesholds and the values in the circles represent the output Y. It is interesting that CART did not train correctly for equal numbers of the four different input cases and that one extra example of one of the input cases was sufficient to break the symmetry and allow CART to train correctly. (Note the similarity to the well-known requirement of random and different initial weights for training the MLP).

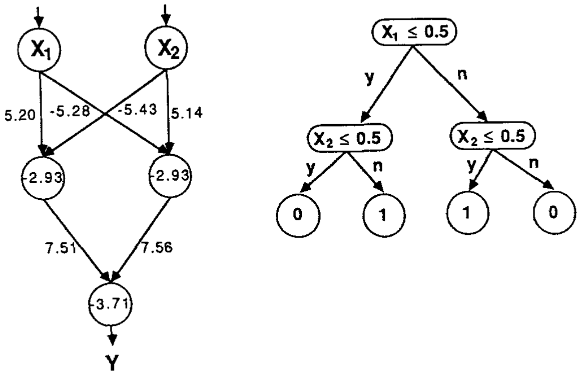

Figure 1. The left side shows an MLP solution and the right side shows a CART solution to the exclusive-OR problem.

CART trains on the exclusive-OR very easily since a rectangle-based partition in the input space is a perfect solution. In general, the MLP will construct classification regions with smooth boundaries, whereas CART will construct regions with 'sharp' corners (each region being, as described previously, an intersection of half planes). We would thus expect MLP to have an advantage when classification boundaries tend to be smooth and CART to have an advantage when they are sharper.

Other important differences between MLP and CART include:

- CART uses a pruning technique to adjust the size of the tree and avoid both underfitting and overfitting. A MLP avoids overfit and underfit by using a hand selected number of hidden units. Both pruning and selection of the number of hidden units can be implemented by using data from a second training set (independent of the first).

- A MLP becomes a classifier through an *ad hoc* application of thresholds or peak-picking to the output values(s). Great care has gone into the CART splitting rules and the usual MLP approach is rather arbitrary.

- A trained MLP represents an approximate solution to an optimization problem. The solution may depend on initial choice of weights and on the optimization technique used. For complex MLP's many of the units are independently and simultaneously adjusting their weights to best minimize output error.

- MLP is a distributed topology where a single point in the input space can have an effect across all units or analogously, one weight, acting alone, will have minimal affect on the outputs. CART is very different in that each split value can be mapped onto one segment in the input space. This behavior of CART make it much more useful for data interpretation. A trained tree may be useful for understanding the structure of the data. The usefulness of MLP's for data interpretation is much less clear.

The above points, when taken in combination, do not make a clear case for either MLP or CART to be superior for the best performance as a trained classifier. We thus believe that the empirical studies of the next sections, with their consistent performance trends, will indicate which of the comparative aspects are the most significant.

III. LOAD FORECASTING

A. The Problem

The ability to predict electric power system loads from an hour to several days in the future can help a utility operator to efficiently schedule and utilize power generation. This ability to forecast loads can also provide information which is able to be used to strategically trade energy with other generating systems. In order for these forecasts to be useful to an operator, they must be accurate and computationally efficient.

Previous approaches to load forecasting rely upon either time series or regression analysis. In general, the regression approach assumes that the load pattern is heavily dependent on the weather and finds an approximate functional relationship between weather variables (such as temperature and wind speed) and the load. A future load can then be predicted by inserting a weather forecast into the previously determined functional relationship. An example of this approach is the work of Gupta and Yamada [12]. The time series approaches assumes that the load fluctuations are composed of many periodicites, such as daily, weekly, and seasonal variations. These periodicities can be considered stochastic and parameters to describe them can be estimated by, say, autoregressive moving average [13] or spectral expansion [14] techniques.

We assume the nature of this problem to be a mixture of a true regression (based on a weather forecast) and a time series prediction. This assumption is the basis of our approach to utilize trained classifiers. More details on this approach and our MLP results can be found in Park et al [15].

B. Methods

Hourly temperature and load data for the Seattle/Tacoma area were provided for us by the Puget Sound Power and Light Company. Since weekday forecasting is a more critical problem for the power industry than weekends, we selected the hourly data for all Tuesday through Fridays in the interval of November 1, 1988 through January 31, 1989. These data consisted of 1368 hourly measurements that consisted of the 57 days of data collected.

Several techniques of input and output pairing were tried and after some investigation [15] we found that a good choice of data organization for our trainable classifier was

$$\{(X\,;\,Y)\} = \{(k, L_{k-2}, L_{k-1}, T_{k-2}, T_{k-1}, T_k; L_k)\}$$

where k was the hour(1-24) of the day and L_i and T_j signify the load and temperature at the i-th and j-th hour respectively. The input thus consists of the hour, 2 past load and temperature readings plus the current temperature. The actual current temperature was used during training and the predicted temperature was used during test, thus representative of the actual technique of relying upon weather reports. The output part is the predicted load, L_k.

These data were presented to both the MLP and the CART classifier as a 6-dimensional input with a single, real-valued output. the MLP required that all values be normalized to the range (0, 1). These same normalized values were used with the CART technique. Our training and testing process consisted of training the classifiers on 53 days of the data and testing on the 4 days left over at the end of January 1989. Our training set consisted of 1272 hourly measurements and our test set contained 96 hourly readings.

The MLP we used in these experiments had 6 inputs (plus the trained constant bias term) 10 units in one hidden layer and one output. This topology was chosen by making use of data outside the trainng and test sets. A standard error backpropagation rule [5] was used for training the interconnect weights. The CART system made use of programs from California Statistical Software [16] and was set up to design regression trees.

C. Results

Errors for this problem are different than they were from the previous classification problem. In particular the output of the two techniques were real numbers, and an error measure was needed to describe how far these predicted values, in testing, deviated from the true value. We chose to use a l_1 norm, namely

$$\text{percent error} = \frac{1}{120} \sum_{k=1}^{120} \frac{|L_k - L_k'|}{L'_k} \quad \times 100\%$$

where L_k was the predicted load and L_k' was the true load at hour k.

The comparative results are listed in Table 1. While both techniques gave quite low error rates, it is very notable that the MLP results were almost always better than the CART results and the worst MLP result (1.78%) was close to the best CART result (1.68%). The ratio of the averages of the independent CART and MLP experiments was about 2.1. This difference is, for real applications, sizable and significant. It is also worth noting that the trained MLP offers performance which is at least as good as the current techniques used by the Puget Sound Power and Light Company.

	PERCENT ERROR MLP	 CART
Test day 1	1.78%	3.33%
Test day 2	1.08	2.83
Test day 3	1.39	3.63
Test day 4	1.30	1.68
Average	1.39%	2.86%

Table 1. Comparison of MLP and CART on load forecasting.

Figure 2 depicts a comparative example of 3 days of data. With the exception of most of the positive and negative peaks, CART and MLP performed similarly well. We tended to see more error outliers for CART than for MLP, but our observations have not yet conclusively shown that peak deviations are the main reason for CART's relatively poorer performance.

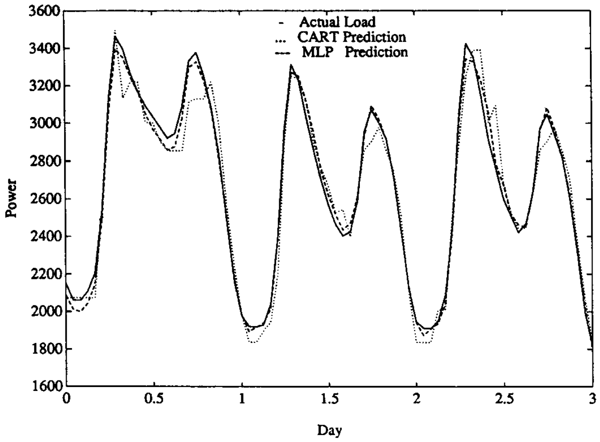

Figure 2. Hourly load predictions for CART and MLP.

IV. SPEAKER-INDEPENDENT VOWEL CLASSIFICATION

A. The Problem

For our last comparative study we chose a very difficult task: speaker-independent classification of vowels excised from continuous speech. Speaker-independent vowel recognition is difficult because of the many sources of variability that influence the physical realization of a vowel. The spectrum of a vowel is determined in part by the length of the speaker's vocal tract; the same vowel produced by two speakers may differ by over 1000 Hz in the location of the resonant frequencies (formants) important for classification [17]. Variation is also provided by the context in which the vowel occurs. For example, the second lowest vowel formant of /ih/ in "kick" and "Lil" may differ by as much as 1000 Hz in the two contexts. There are several other sources of variation, such as speech rate, syllable stress and word stress.

To make the task even more difficult in the present experiments the classifiers were presented only with information from a single spectral slice. The spectral slice was taken from the center of the vowel, where the effects of coarticulation with surrounding phonemes are least apparent.

There are several interesting reasons to investigate vowel recognition using a single spectral slice. First, the most successful speech recognition systems today [18] model phonemes as *sequences* of spectra. By examining classification performance using a single spectra slice, we can gain some insight about the relative performance of these systems at the phonetic level. An important feature of any representation is how well it preserves phonetic information. Classification using a single spectral slice provides a good measure of this capability. Second, the present experiments address the question of how best to present information about the spectrum to a classifier. Should the classifier be presented with the complete set of spectral coefficients, or will some processing scheme produce better classification results?

B. Methodology

The training and test sets for the experiments consisted of featural descriptions, X, paired with an associated class, *C*, for each vowel sample. The 12 monophthongal vowels of English, shown listed in Table 2, were used for the classes.

Phone	Example	Phone	Example
/iy/	b*ea*t	/ah/	b*u*tt
/ih/	b*i*t	/uw/	b*oo*t
/eh/	b*e*t	/uh/	b*oo*k
/ae/	b*a*t	/ao/	b*ough*t
/ix/	ros*e*s	/aa/	c*o*t
/ax/	th*e*	/er/	b*ir*d

Table 2. The 12 classes for the vowel problem.

The vowels were excised from the wide variety of phonetic contexts in utterances of the TIMIT database, a standard acoustic phonetic corpus of continuous speech, displaying a wide range of American dialectical variation [19, 20]. The diphthongs /oy/, /ay/, /ey/, /aw/, and /ow/ were excluded because they are characterized by spectral change, and are therefore inappropriate for experiments using information from a single spectral slice. The training set consisted of 4104 vowels from 320 speakers. The test set consisted of 1644 vowels from a different set of 100 speakers.

A 256-point real DFT was computed on each utterance, with a 10 ms Hamming window and 3 ms increment, yielding 128 spectral energy coefficients every 3 ms frame of the utterance. Since the important information about vowel identity is found below 4 kHz., only the first 64 spectral coefficients (0-4 kHz) were used. Using the hand-segmented phonetic transcriptions provided by the TIMIT database, the center frame of each vowel token was located, and the first 64 spectral coefficients corresponding to this frame were extracted. The coefficient values (in dB) were then normalized to lie between 0 and 1. Normalization was done by computing the "relative value" of each coefficient with respect to the maxima and minima in the 64 coefficients,

$$\text{normalized value} = \frac{(X_i - \text{min})}{(\text{max} - \text{min})}$$

where X_i is the value of any spectral coefficient, *max* is the value of the largest of the 64 spectral coefficients, and *min* is the value of the smallest of the 64 spectral coefficients.

The MLP systems consisted of 64 inputs (the number of DFT coefficients) and 12 outputs. There was one hidden layer which consisted of 40 units.

The networks were trained using backpropagation with conjugate gradient optimization [10]. The procedure for training and testing a network proceeded as follows: The network was trained on 100 iterations through the 4104 training vectors. The trained network was then evaluated on the training set and a different set of 1644 test vectors. This process was continued and the performance of the network on the training and test vectors was recorded after every 100 iterations through the training set. The training was stopped when the network had converged; convergence was observed as a consistent decrease or leveling off of the classification percentage on the test data over successive sets of 100 iterations.

The CART system was trained using two separate computer routines. One was the CART program from California Statistical Software; the other was a routine we designed ourselves. We produced our own routine to ensure a careful and independent test of the CART concepts described in Breiman et al [9].

C. Results

When using the scaled spectral coefficients to train both techniques, the MLP correctly classified 47.4% of the test set while CART employing uni-variate splits performed at only 38.2%.

One reason for the poor performance of CART with uni-variate splits, may be that each coefficient (corresponding to energy in a narrow frequency band) contains little information when considered independently of the other coefficients. For example, reduced energy in the 1 kHz band may be difficult to detect if the energy in the 1.06 kHz band was increased by an appropriate amount. The CART classifier described above operates by making a series of inquiries about one frequency band at a time, an intuitively inappropriate approach.

We achieved our best CART results, 46.4% on the test set, by making use of arbitrary hyper-planes (linear combinations) instead of univariate splits. This search-based approach gave results which were within 1% of the MLP results.

These seemingly low scores came from a task with chance performance of about 8.3%. Nevertheless, the high error rates we found are probably indicative of the extreme difficulty in identifying speaker-independent vowels from a single spectral slice. In fact, listeners presented with vowel sounds excised from the TIMIT data base agree on vowel labels only about 65% of the time [21]. There is a large difference between this problem, with its apparent high optimal error rate, and the power security problem, which had a 0% optimal error rate.

V. CONCLUSIONS

Table 3 lists a summary of comparative results for both problems. This table includes, where suitable, the best performing case for both CART and the MLP. The load forecasting and the vowel classification problem differed in more than just application area; they each have a very different underlying structure. The problem of vowel recognition gave MLP and CART results which are not immediately applicable. However, the impressive performance of the MLP technique for load forecasting at least equaled the techniques currently used by the Puget Power and Light Company.

Note that the best CART performance on the vowels was very close to the MLP result. This CART result made use of linear combination splits. Also, other MLP vowel experiments which made use of more features and gave performance as high as 55.2% correct [22]. We are currently in the process of applying linear combination splits to the load forecasting problem and to the larger vowel feature set.

	Lowest Error Rate	
	MLP	CART
Load Forecasting	1.39%	2.86%
Vowel Recognition	52.6%	53.6%

Table 3. Summary of best results on load forecasting and vowel recognition.

There are several possible reasons for the superior performance of the MLP technique, all of which we are currently investigating. One advantage may stem from the ability of MLP to easily find correlations between large numbers of variables. CART is best suited for finding significant properties of single features (i.e. those whose range is associated with a given class). Although it is possible for CART to form arbitrary nonlinear decision boundaries, the efficiency of the recursive splitting process may be inferior to MLP's nonlinear fit. Another relative disadvantage of CART may be due to the successive nature of node growth. For example, if the first split that is made for a problem turns out, given the successive splits, to be sub-optimal, it becomes very inefficient to change the first split to be more suitable.

We feel that the careful statistics used in CART could also be advantageously applied to MLP. The superior performance of MLP is not yet indicative of best performance and it may turn out that careful application of statistics may allow further advancements in the MLP technique. It also may be possible that there would be input representations that would cause better performance for CART than for MLP.

There also have been developments in trained statistical classifiers since the development of CART. More recent techniques, such as projection pursuit [23], may prove as good as or superior to MLP. This continued interplay between MLP techniques and advanced statistics is a key part of our ongoing research.

ACKNOWLEDGEMENTS

The authors wish to thank Professor R.D. Martin of the University of Washington Department of Statistics and Professors Aggoune, Damborg, and Hwang of the University of Washington Department of Electrical Engineering for their helpful discussions. David Cohn and Carlos Rivera assisted with many of the experiments.

We also would like to thank Milan Casey Brace of Puget Power and Light for providing the load forecasting data.

This work was supported by a National Science Foundation Presidential Young Investigator Award for L. Atlas and also by separate grants from the National Science Foundation and Washington Technology Center.

References

1. N. J. Nilsson, **Learning Machines**, McGraw-Hill, New York, 1965.

2. A. G. Arkadev and E. M. Braverman, **Computers and Pattern Recognition**, Thompson Book Co., Inc., Washington, DC, 1966.

3. W. S. Meisel, **Computer-Oriented Approaches to Pattern Recognition**, Academic Press, New York, 1972.

4. R. O. Duda and P. E. Hart, **Pattern Classification and Scene Analysis**, John Wiley and Sons, New York, 1973.

5. D. E. Rumelhart, G. Ę. Hinton, and R. J. Williams, Learning Internal Representations by Error Propagation," Ch. 2 in **Parallel Distributed Processing**, D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, MIT Press, Cambridge, MA, 1986.

6. W. S. McCuloch and W. H. Pitts, "A Logical Calculus of the Ideas Imminent in Nervous Activity," **Bulletin of Mathematical Biophysics, 5**, pp. 115-133, 1943.

7. F. Rosenblatt, **Principles of Neurodynamics**, Spartan Books, New York, 1962.

8. W. S. Meisel and D. A. Michalpoulos, "A Partitioning Algorithm with Application in Pattern Classification and the Optimization of Decision Trees," **IEEE Trans. Computers**, C-22, pp. 93-103, 1973.

9. L. Breiman, J. H. Friedman, R. A Olshen, and C. J. Stone, **Classification and Regression Trees**, Wadsworth International, Belmont, CA, 1984.

10. E. Barnard and D. Casasent, "Image Processing for Image Understanding with Neural Nets," in *International Joint Conference on Neural Nets*, (1989). (Submitted for publication).

11. R. P. Lippmann, "An Introduction to Computing with Neural Nets," **IEEE Acoustics, Speech and Signal Processing Magazine**, pp. 4-22, 1987.

12. P. Gupta and K. Yamada, "Adaptive Short-Term Forecasting of Hourly Loads Using Weather Information," **IEEE Tr. on Power App. and Sys.**, vol. PAS-91, pp. 2085-2094, 1972.

13. S. Vemuri, W. Huang, and D. Nelson, "On-line Algorithms For Forecasting Hourly Loads of an Electric Utility," **IEEE Tr. on Power App. and Sys.**, vol., PAS-100, pp. 3775-3784, Aug. 1981.

14. W. Christiaanse, "Short-Term Load Forecasting Using General Exponential Smoothing," **IEEE Tr. on Power App. and Sys.**, vol. PAS-90, pp. 900-910, Apr., 1971.

15. D. Park et al, "Electric Load Forecasting Using an Artificial Neural Networks," Submitted to **IEEE Power Engineering Systems**, Winter Meeting, 1990.

16. California Statistical Software, Inc., 961 Yorkshire Court, La Fayette, CA 94549.

17. G. E. Peterson and H. L. Barney, "Control Methods Used in a Study of the Vowels," *Journal of the Acoustical Society of America* 24, pp. 175-184 (1952).

18. K. Lee and H. Hon, "Speaker-independent Phoneme Recognition Using Hidden Markov Models," CMU-Computer Science-88-121, Dept. of Computer Science, Carnegie-Mellon University (1988).

19. W. Fisher, G. Doddington, and K. Goudie-Marshall, "The DARPA Speech Recognition Research Database: Specification and Status," pp. 93-100 in *Proceedings of the DARPA Speech Recognition Workshop*, (February, 1986).

20. L. Lamel, R. Kassel, and S. Seneff, "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus," pp. 100-110 in *Proceedings of the DARPA Speech Recognition Workshop*, (February, 1986).

21. M. Phillips, "Speaker-independent classification of vowels and diphthongs in continuous speech," in *Proc. of the 11th International Congress of Phonetic Sciences*, Estonia, USSR (1987).

22. R. Cole, Y.K. Muthusamy and L.Atlas, Speaker-Independent Vowel Recognition : Comparison of Backpropagation and Trained Classification Trees," Proceedings IEEE International Conference on System Sciences - Neural Networks and CAS Related Emerging Technologies, Kailua-Kona, Hawaii, 3-6 January, 1990 (to appear).

23. J. H. Friedman and W. Stuetzle, "Projection Pursuit Regression," *J. Amer. Stat. Assoc.* 79, pp. 599-608, 1984.