J.N. Hwang, R.J. Marks II and L.E. Atlas, "Neural network research at the University of Washington - recent results and applications", Northcon/90 Conference Record, (Western Periodicals Co., North Hollywood, CA), Seattle WA, October 9-11, 1990, pp. 263-268

OCTOBER 9 - 11, 1990
SEATTLE, WASHINGTON

CONFERENCE RECORD

Seattle and Portland Sections, Institute of Electrical and Electronic Engineers, IEEE

Cascade Chapter, Electronic Representatives Association, ERA

Electronic Manufacturers Association

Neural Network Research at the University of Washington: Recent Results and Applications*

Jenq-Neng Hwang, Les Atlas, Robert J. Marks II

Interactive Systems Design Laboratory
Department of Electrical Engr., FT-10
University of Washington
Seattle, WA 98195

1 Classification and Regression with Multilayer Perceptrons

There have been a number of applications proposed for artificial neural network computational structures. The neural network architecture currently receiving most attention as a viable computational paradigm with competitive performance attributes is the multilayer perceptron. A multilayer perceptron is a feed-forward neural network which has one or more layers of *hidden neurons* between the input and output layers. The weights of a multilayer perceptron are typically trained using *back-propagation* learning, an iterative gradient descent algorithm designed to minimize the mean squared error between the the desired target vector and the actual output vector [1].

Two major applications of the multilayer perceptrons are *Classification* and *Regression*. After the network is trained by available data, the task of classification is to identify which *discrete* class or classes the incoming input pattern belongs to. The task of regression, on the other hand, is to retrieve the most suitable output response (continuous nonlinear mapping of) the incoming input pattern.

2 Comparative Studies in Classification and Regression

Does a multilayer perceptron perform better than other classifiers and regression machines? By comparison with some other high performance classifiers and regression machines, the current answer is yes - but not by much. Possibly there is an underlying limit of performance placed on all classifiers and regression machines that cutting edge algorithms are approaching. If so, then secondary performance attributes such as training speed and implementation ease must be addressed as primary.

Other artificial neural networks have fallen from favor in an application sense because, quite simply, they are not competitive with other more conventional approaches. The same question must be posed in regard to the multilayer perceptrons. Does the layered perceptron perform better than other classifiers or regression machines programmed from examples using supervised learning? Although abstract analysis of this question may be possible in some cases, it must ultimately be answered in regard to actual data. Comparisons of the multilayer perceptrons have been performed with nearest neighbor lookup, classification and regression trees (CART) and projection pursuit learning networks for such problems as

*The majority of the contributions of this paper are the result of multitudinous hours of collaboration at the ISDL. The other major contributors are Profs. C. H. Chan, M. Damborg, M. A. El-Sharkawi, R. Ladner, D. Martin, and J. Vagner. Drs. M. Aggoune, J. J. Choi, D. Hoskins, S. Oh, and D. C. Park have also offered significant contributions.

speech (classification), power security assessment (classification), load forecasting (regression), and some specially designed nonlinear mappings (regression). In each case, have shown the multilayer perceptron to perform better in terms of classification or regression accuracy, albeit at a higher computational and memory price.

In comparison with nearest neighbor lookup, the layered perceptron was shown to interpolate much more smoothly and with greater accuracy for the problem of power security assessment [2, 3].

Here are some accuracy figures contrasting the layered perceptron with CART. Details of the experiments can be found in [4, 5]. Indeed, these papers must be consulted to give significant meaning to the statistics that follow. In power load forecasting [6], current and forecasted temperature and current load demand is used to forecast the future power load demand. For this problem, the worst perceptron performance was an error of 1.78%. CART produced an error of 1.68%. For speaker independent vowel classification, the perceptron again had a higher correct classification rate than CART, 47.4% to 38.2%. In the power security assessment problem, the state of a power system is determined to be safe or in jeopardy. Applied to this problem, the perceptron again had a lower error rate - 0.78% to 1.46% [6, 7]. In the form of CART used in this experiment, the feature space was initially divided into planes that were perpendicular to the axes. In a higher order form of CART, these planes can be oriented at angles. The higher order form of CART has given preliminary results that are closer in performance to the layered perceptron. However, in all of our studies so far, CART has never done better than a layered perceptron.

The projection pursuit learning networks can be regarded as a generalization of single hidden layer neural networks. Unlike the neural network learning, which employs a gradient descent search using fixed set of nonlinear cell functions (e.g., sigmoid), the projection pursuit networks combines least squares fitting and Gauss Newton optimization to nonparametrically estimate the weights as well as the nonlinear cell functions using a one dimensional data-smoother. Several specially designed nonlinear mappings (within a noisy training environment) are used in the comparisons, e.g., parabolic, radial, harmonic, statistically additive, and more complicated statistically multiplicative mappings. The multilayer perceptrons performed consistently better than the projection pursuit networks. For example, when applied to the harmonic function, the perceptrons had a smaller mean squared error compared to projection pursuit networks, 0.071 to 0.254 [8].

3 Query Learning for Classification

One problem associated with trained classifiers is the diminishing return of information content in randomly generated training data obtained with respect to the data set cardinality. In other words, the more that is learned, the harder it becomes to learn something new. To illustrate, consider the classification problem of learning the location of a point a on the interval $0 < a < 1$. We choose a point at random on the unit interval. If it to the right of a, we assign it a value of one. If it is to the left of a, the result is 0. It is clear that, after a number of data points have been generated at random on the unit interval, that a lies somewhere between the rightmost 0 and the leftmost 1. Call this subinterval C. If we generate a new data point that does not lie in the subinterval C, we have learned nothing new. If the new point lies in the subinterval C, then we revise the subinterval and make its duration shorter. Doing so, however, decreases the chance that the next data point contains new information. That is, the probability decreases that the new data point lies in the shorter interval. Thus, in this example, the more we learn about the location of the point a, the harder it is to learn. One approach to counteract this phenomenon is with the use of oracles in query based learning [9].

Consider a binary classification problem, which is totally determined by the classification boundary. Indeed, here is an obvious case where the importance of data to the classification can be noted. Roughly, the closer a feature vector is to the concept classification boundary, the more information it contains. One way to exploit this observation is through interval halving. Between each feature vector classified 0 and each classified 1, there exists a classification boundary. In many cases, taking the geometric midpoint

of these two feature vectors to the oracle will result in a classification point closer to the boundary. This is assured, for example, if the underlying concept is convex. To illustrate interval halving, let's return to the problem of finding the point a on the interval (0,1). After N randomly generated points on this interval, we would expect (in the sense of statistics), that the distance between the right most zero and the left most one is about $1/N$. Using interval halving, on the other hand, this is reduced to about 2^{-N}. The acceleration in learning is indeed remarkable.

In many machine learning applications, the source of the training data can be modeled as an *oracle*. An oracle has the ability, when presented with an example, to give a correct classification. A cost, which can be very expensive (e.g., a supercomputer emulator), is typically associated with this query. The study of queries in classifier training paradigms is therefore a study of the manner by which oracles can provide good classifier training data at low cost.

Our oracle based query learning is developed to help a partially trained classifier to respond to the question: "What don't you yet understand?" The response of the oracle is used as additional training data to clear the classifier's confusion. The properly classified points from the oracle are then introduced as additional training data for the classifier. The use of queries through such a systematic data generation mechanism can be viewed as interactive learning. On the other hand, the use of only available (or randomly generated) data, is passive learning. If done properly, the use of queries can reduce the cumulative cost of data drastically as compared to the case where examples are generated at random [10, 11].

We have proposed a network inversion algorithm for query learning. This inversion algorithm can be regarded as a dual algorithm of back-propagation learning. It allows generation of the network input (or inputs) that can produce any specified output vector. The inversion of a network midway between two classifications results in a classification boundary. This boundary is the locus of input vectors that, with respect to the neural network's representation of the training data, is highly confusing. In addition, for each boundary point, we can generate the classification gradient. The gradient provides a useful measure of the sharpness of the multi-dimensional decision surfaces. Using the boundary point and gradient information, conjugate input pairs are generated and presented to an oracle for proper classification. This new data is used to further refine the classification boundary thereby increasing the classification accuracy. The result can be a significant reduction in the training set cardinality in comparison with, for example, randomly generated data points. This query learning technique has been successfully applied to the power system security assessment problems and will be applied to some inverse problems in remote sensing [12, 13].

4 Constrained Inversion for Regression

Thanks to the capability of neural networks' approximating (identifying) most classes of continuous nonlinear function, multilayer perceptrons are widely used in nonlinear regression applications. The forward problem in a nonlinear functional mapping is to obtain the best approximation of the output vector given the input vector. The inverse problem, on the other hand, is to obtain the best approximation of the input vector given a specified output vector, i.e., to find the inverse function of the nonlinear mapping, which might not exist except when the constraints are imposed on. This leads to the constrained inverse problems for a trained nonlinear mapping. These problems can be found in a wide variety of applications in dynamic control of nonlinear systems and nonlinear constrained optimization. Most neural networks previously proposed for training the inverse mapping either adopted an one-way constraint perturbation or a two-stage learning. Both of these approaches are very laborious and unreliable.

Instead of using two neural networks for emulating the forward and inverse mappings separately, we applied the network inversion algorithm, which works directly on the network used to train the forward mapping, yielding the inverse mapping [14]. Our approach uses one network to emulate both of forward and inverse nonlinear mapping without explicitly characterizing and implementing the inverse mapping.

Furthermore, our single network inversion approach allows us to iteratively locate the optimal inverted solution which also satisfies some constraints imposed on the inputs. It also allows best exploitation of the sensitivity measure of the inputs to outputs in a nonlinear mapping. This constrained inversion techniques has been successfully applied to frequency selective surface design [15] and adaptive nonlinear control [16].

5 Recursive Learning of Neural Networks

While multilayer perceptrons provide a very powerful nonlinear modeling capability, back-propagation learning can be very slow and inefficient. In linear adaptive filtering, the analog of the back-propagation algorithm is the least-mean-squares (LMS) algorithm. Steepest descent-based algorithms such as back-propagation or LMS are *first order* in that they utilize first derivative or gradient information about the training error to be minimized. To speed up the training process, *second order* algorithms may be employed that take advantage of second derivative or Hessian matrix information.

Second order information can be incorporated into multilayer perceptron training in different ways. In many applications, especially in the area of pattern recognition, the training set is finite. In these cases, block learning can be applied using standard nonlinear optimization techniques. These techniques iteratively minimize the error for the *complete* training set and are based on the ability to evaluate the total error and its derivatives with respect to the network parameters at arbitrary values of the parameters. In essence, the training set is rerun for each function and derivative evaluation.

Recursive training seeks to adjust the network parameters as training patterns are presented, rather than after a complete pass through the training set. This approach is necessary when the training set is infinite or at least orders of magnitude larger than the number of network parameters. In principle, each training pattern is seen only once, as would be the case in a time series filtering operation. Although in practice, recursive training is commonly applied to finite training sets by repeated application of the same set of patterns. Underlying this approach is the assumption that the knowledge to be extracted from the training set is of a statistical nature and that no single training pattern contains unique information.

In linear adaptive filter, recursive least squares (RLS) algorithms implement second order recursive training. The basic approach that underlies RLS algorithms for linear adaptive filters may also be applied to the training of feed-forward ANNs in general and multilayer perceptrons in particular. RLS-like algorithms for multilayer perceptrons have been derived in a variety of ways and use different amounts of second order information. These derivations include the application of the extended Kalman filtering equations to MLP training, neuron-local linearizations of the sigmoid functions, and quadratic MLP error approximations.

An approximate least squares formulation of the training problem is derived from a linearization of the error function, which yields a quadratic squared-error function. We can show what approximations are necessary both for this least squares approximation and for its recursive solution. We also illustrate how the formulated least squares problem can be solved by either conventional RLS recursions or by the more numerically stable QR decomposition-based methods popular in linear least squares filtering [17].

6 Acknowledgements

Portions of this work were supported by the National Science Foundation, Puget Power and Light Company and the Washington Technology Center.

References

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Parallel Distributed Processing (PDP): Exploration in the Microstructure of Cognition (Vol. 1)*, Chapter 8, MIT Press, Cambridge, Massachusetts, 1986.

[2] M.E. Aggoune, L.E. Atlas, D.A. Cohn, M.J. Damborg, M.A. El-Sharkawi and R.J. Marks II. Artificial neural networks for static system security assessment. *Proc. IEEE International Symposium on Circuits and Systems*, pp. 490-494, May 1989, Portland (invited paper).

[3] M.A. El-Sharkawi, R.J. Marks II, M.E. Aggoune, D.C. Park, M.J. Damborg and L.E. Atlas. Dynamic security assessment of power systems using back error propagation artificial neural networks. *Proceedings of the 2nd Annual Symposium on Expert Systems Applications to Power Systems*, Seattle, pp. 366-370, 17-20 July 1989,

[4] 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. IEEE International Conference on Systems, Man and Cybernetics*, pp. 915-920, Cambridge, Massachusetts, Nov. 1989.

[5] L.E. Atlas, R. Cole, Y. Muthusamy, A. Lippman, G. Connor, D.C. Park, M. El-Sharkawi, and R.J. Marks II. A performance comparison of trained multi-layer perceptrons and classification trees, *Proceedings of the IEEE*, August 1990 (in press).

[6] M.J. Damborg, M.A. El-Sharkawi and R.J. Marks II. Potential applications of artificial neural networks to power system operation. *Proc. IEEE International Symposium on Circuits and Systems*, pp. 2933-2936, May, 1990, New Orleans, Louisiana (invited paper).

[7] M. Aggoune, M.A. El-Sharkawi, D.C. Park, M.J. Damborg and R.J. Marks II, Preliminary results on using artificial neural networks for security assessment. *Proceedings of the Power Industry Computer Applications Conference*, pp. 252-258, June 1989, Seattle, WA.

[8] M. Maechler, D. Martin, J. Schimert, M. Csoppenszky, and J. N. Hwang. Projection pursuit learning networks for regression. Submitted to *Conf. on Neural Information Processing Systems*, November 1990, Denver, Colorado.

[9] L.E. Atlas, D. Cohn, R. Ladner, M. El-Sharkawi, R.J. Marks II, M. Aggoune, and D.C. Park. Training connectionist networks with queries and selective sampling. *Proc. Conf. on Neural Information Processing Systems*, pp. 566-573, November 1989, Denver, Colorado.

[10] J.N. Hwang, J.J. Choi, S. Oh, and R.J. Marks II. Classification boundaries and gradients of trained multilayer perceptrons. *Proc. IEEE International Symposium on Circuits and Systems*, pp. 3256-3259, May, 1990, New Orleans, Louisiana.

[11] J.N. Hwang, J.J. Choi, S. Oh, and R.J. Marks II. Query learning based on boundary search and gradient computation of trained multilayer perceptrons. *Proc. International Joint Conference on Neural Networks*, San Diego, June, 1990 (to appear).

[12] C.M. Lam, D.C. Park, L. Tsang, R.J. Marks II. A. Ishimaru, and S. Kitamura. Determination of particle distribution using a neural network trained with back-scatter measurement. *Proc. IEEE Ap-S International Symposium & URSI Radio Science Meeting*, May 1990, Dallas, Texas.

[13] A. Ishimaru, R.J. Marks II, L. Tsang, C.M. Lam, D.C. Park and S. Kitamaru. Optical sensing of particle size distribution by neural network technique. *Proc. International Geoscience and Remote Sensing Symposium*, May 1990 , Washington D.C.

[14] J.N. Hwang, C. H. Chan. Iterative constrained inversion of neural networks and its applications. *Proc. Conference on Information Systems and Sciences*, Princeton, March, 1990.

[15] J.N. Hwang, C. H. Chan, and R.J. Marks II. Frequency selective surface design based on iterative inversion of neural networks. *Proc. International Joint Conference on Neural Networks*, San Diego, June, 1990 (to appear).

[16] J. N. Hwang, D. A. Hoskins, and J. Vagners. Iterative inversion of neural networks and its application to adaptive control. Submitted to *Conf. on Neural Information Processing Systems*, November 1990, Denver, Colorado.

[17] P. S. Lewis and J. N. Hwang. Recursive least squares learning algorithms for neural networks. *Proc. SPIE's International Symposium on Optical and Optoelectronic Applied Science and Engr.* July 1990, San Diego, Calif.