

Query Learning Based on Boundary Search and Gradient Computation of Trained Multilayer Perceptrons*

Jenq-Neng Hwang, Jai J. Choi[†], Seho Oh, Robert J. Marks II

Department of Electrical Engr., FT-10
University of Washington
Seattle, WA 98195

Abstract

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 (*query*), to give a correct classification. Through interaction with a partially trained classifier, efficient query learning uses an oracle to produce training data with high information content. This paper presents a novel approach for query based neural network learning. Consider a layered perceptron partially trained for binary classification. The single output neuron is trained to be either a 0 or a 1. A test decision is made by thresholding the output at, say, $\frac{1}{2}$. The set of inputs that produce an output of $\frac{1}{2}$, forms the classification boundary. We adopted an inversion algorithm for the neural network that allows generation of this boundary. 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 pair locations 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. An application example to power security assessment is given.

1 Introduction

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, however, 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 the good classifier training data at low cost.

One method of query learning makes use of a partially trained classifier. The classifier locates points of confusion which the classifier has difficulty in classifying. These data points are then

*This work is supported through grants from the National Science Foundation and the Washington Technology Center. Discussions with L. Atlas, M. A. El-Sharkawi, D. Cohen and R. Ladner are gratefully acknowledged.

[†]J. J. Choi was supported, in part, by a developmental grant from Physio Control Inc.

taken to the oracle for proper classification. The properly classified points are then introduced as additional training data for the classifier. The use of queries through 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 properly done, the use of queries can reduce the cost of data drastically from the case where examples are generated at random [1].

2 Inversion of Multilayer Perceptrons

Multilayer perceptrons are feed-forward neural networks, which have one or more layers of *hidden neurons* between the input and output layers. The system dynamics in the retrieving phase of an L -layer neural net can be described by the following equations:

$$u_i(l+1) = \sum_{j=1}^{N_l} w_{ij}(l+1)a_j(l) + \theta_i(l+1) \quad (1)$$

$$a_i(l+1) = f(u_i(l+1)) \quad 1 \leq i \leq N_{l+1}, \quad 0 \leq l \leq L-1 \quad (2)$$

where $a_j(l)$ denotes the activation value of the j^{th} neuron at the l^{th} layer and f is the nonlinear activation function.

The inversion of a network will generate the input $\{a_j(0)\}$ (or inputs) that can produce the desired output vector. By taking advantage of the *duality* between the weights and the activations in minimizing the mean squared error between the the desired target values $\{t_i\}$ and the actual output values $\{a_i(L)\}$, the iterative gradient descent algorithm can also be applied to obtain the desired input.

$$a_j(0) \Leftarrow a_j(0) - \eta \frac{\partial E}{\partial a_j(0)} \quad (3)$$

where $E = \frac{1}{2} \sum_{i=1}^{N_L} (t_i - a_i(L))^2$.

The idea is similar to the BP algorithm [2], where the error signals are propagated back to tell the weights the manner in which to change in order to decrease the output error. The inversion algorithm back-propagates the error signals down to the input layer to update the activation values of input units so that the output error is decreased [3].

3 Boundary Search and Gradient Computation

Boundary Search for Region of Maximum Ambiguity After a multilayer perceptron is well trained by the BP algorithm, it is very instructive to locate the *classification boundary* (or the region of maximum classification ambiguity). Without loss of generality and also for simplicity of illustration, only a binary classification (two class) example is given below. We want to train a single output neuron, $a_1(L)$, to be either “0” or “1”. A good strategy is to evaluate the boundary corresponding to an output value of 0.5. This can be done by inverting the network with several randomly selected initial input data points. The inversion algorithm will

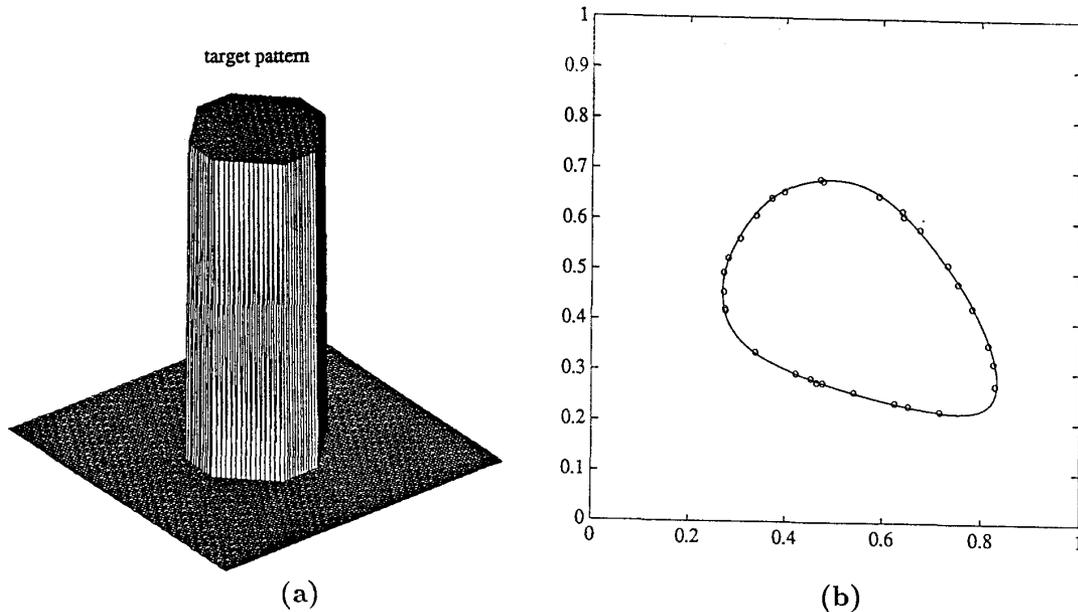


Figure 1: (a) A perspective plot of the hexagon classification region. (b) 30 Classification boundary points.

take some trajectory and gradually move each initial input data toward one specific boundary point [4].

A neural network with a single hidden layer of 10 neurons was trained with 50 randomly selected 2-dimensional data for hexagonal region classification. A perspective plot of the classification region is shown in Figure 1(a) where the classification target is “1” inside the hexagon and “0” without. There are 50 randomly selected training data used for training. After 3000 iterations of the training, 30 classification boundary points corresponding to an output of 0.5 were created using the inversion algorithm and is shown in Figure 1(b).

For classification of multiple classes, where each class (say, the k -th) is represented by the k -th output neuron, the *boundary* (or the region of maximum ambiguity) for the k -th class can be also located by searching the points (in the input space) which give rise to 0.5 activation value for the k -th output neuron regardless the values of other output neurons. To be more specific, the E measure in Eq. 3 is computed based only on the square difference between the target value and actual value of k -th output neuron;

$$E = \frac{1}{2} (t_k - a_k(L))^2.$$

Gradient Computation for Sensitivity of Functional Mapping After the neural network is well trained, the functional mapping relationship between the input and output is established through the weights. For each point in the input space, we can compute the gradient, $\rho_{kj}(L)$, of each output neuron (*e.g.* the k -th), with respect to each input neuron (*e.g.* the j -th). This gradient is a measure of the sensitivity of the functional mapping relationship. This gradient,

$$\varrho_{kj}(L) = \frac{\partial a_k(L)}{\partial a_j(0)}, \quad (4)$$

can be recursively computed from

$$\varrho_{ij}(l) = f'(u(l)) \sum_{k=1}^{N_{l-1}} w_{ik}(l) \varrho_{kj}(l-1), \quad 2 \leq l \leq L \quad (5)$$

where the initial values are given as $\varrho_{kj}(1) = f'(u_k(1))w_{kj}(1)$.

Figure 2(a) illustrates the magnitude of the gradient of each inverted boundary point shown in Figure 1(b).

Conjugate Pair Around Boundary for Refining For each inverted boundary point, a conjugate training data pair based on the magnitude of gradient were also created (see Figure 2(b)). More specifically, two points lying on opposite sides of the line passing through the boundary point and perpendicular to the boundary surface are located, with their distances to the corresponding boundary point equal to $1/|\varrho_{kj}(L)|$. The motive behind the selection of this conjugate pair comes from the desire to sharpen the boundary between two distinct classes by narrowing the regions of ambiguity. The oracle will provide true classification (1 or 0) for those newly generated query points. If all three points (the boundary point as well as the conjugate pair) fall into the same class, we neglect the conjugate pair and keep only the boundary point as part of the new training data. Otherwise, we choose all three points to be part of training data. This results in a set of newly generated training data, which will be used to retrain the network along with the originally randomly selected data. The result of query learning was much more favorable as compared with that of conventional learning based on purely randomly selected training data with the same data set cardinality or even a larger size of training data [4].

4 Applications to Power System Security Problem

One of the main aspects of *power system security* is the so-called static or steady state security. This is defined as the ability of the system, after a disturbance such as a line break or other rapid load change, to reach a state within the specified safety and does not violate any operating constraints [5].

Defining multi-dimensional static-security regions for even a small power network is a computationally demanding task. It involves the solution of a nonlinear programming problem with a large number of variables and an equally large number of limit constraints which define the feasible region of operation. In addition, the amount of memory required to store the security status under each probable network configuration will be equally prohibitive.

A 3-layer perceptron consists of 4 inputs, 2 hidden layers (with 20 and 10 units separately), and one output is used to train a 4-dimensional power system security assessment. Figure 3 shows the performance comparison between query based and random sample based training approach for this problem. It can be easily seen that as the training size increases, the relative advantage of the query based system increases significantly. To illustrate how well the query based learning identifying the real boundary, a 2-dimensional (randomly selected) slice of this

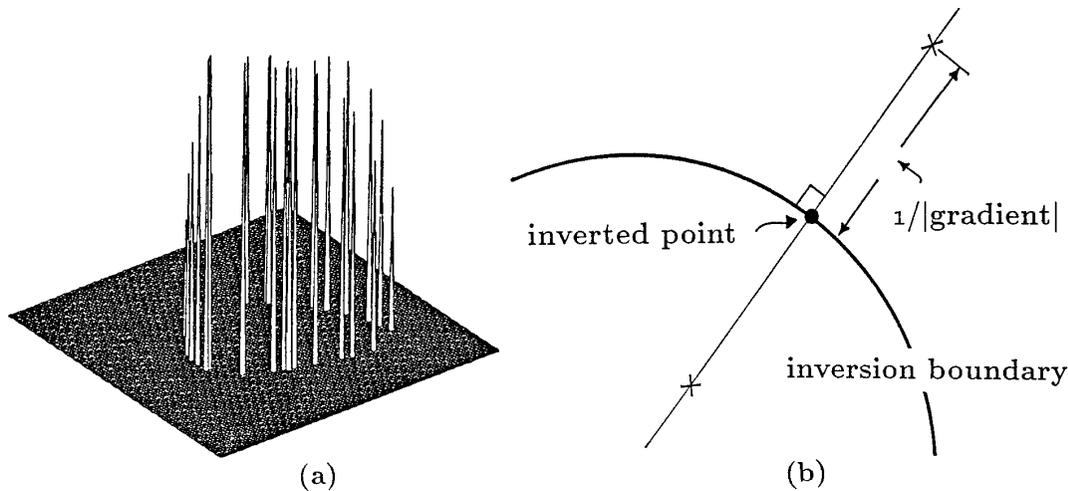


Figure 2: (a) The magnitude of the gradient of each inverted boundary point. (b) A conjugate training data pair based on gradient were created.

4-dimensional power security problem is used for graphical presentation. Figure 4(a) shows the boundary trace result for 5000 randomly selected training data are trained. On the other hand, Figure 4(b) shows the boundary trace result for 5000 query selected training data (including 500 initially randomly selected data and two times of queries from the oracle) are trained.

References

- [1] D. Cohn, L. E. Atlas, R. Ladner, R. J. Marks II, M. El-Sharkawi, M. Aggoune, and D. C. Parks. Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems*, Denver, November 1989.
- [2] 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, pages 318–362. MIT Press, Cambridge, Massachusetts, 1986.
- [3] A. Linden and J. Kindermann. Inversion of multilayer nets. In *Proc. Int'l Joint Conf. on Neural Networks*, II 425-430, Washington D.C., June 1989.
- [4] J. N. Hwang, J. J. Choi, S. Oh, R. J. Marks, Classification boundaries and gradients of trained multilayer perceptrons. In *Proc. Int'l Symposium on Circuits and Systems*, New Orleans, May 1990.
- [5] L. H. Fink. Power System Security Assessment. In *Proc. IEEE Conference on Decision and Control*, pp. 478-480, December 1984.

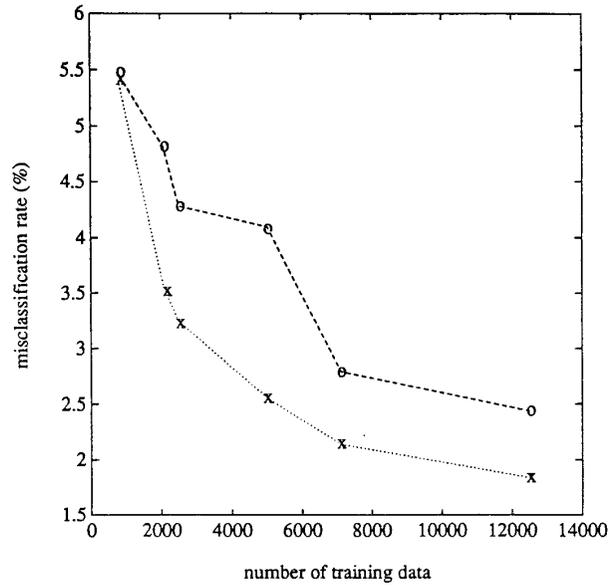


Figure 3: The performance comparison between query based and random sample based training approach for power security problem.

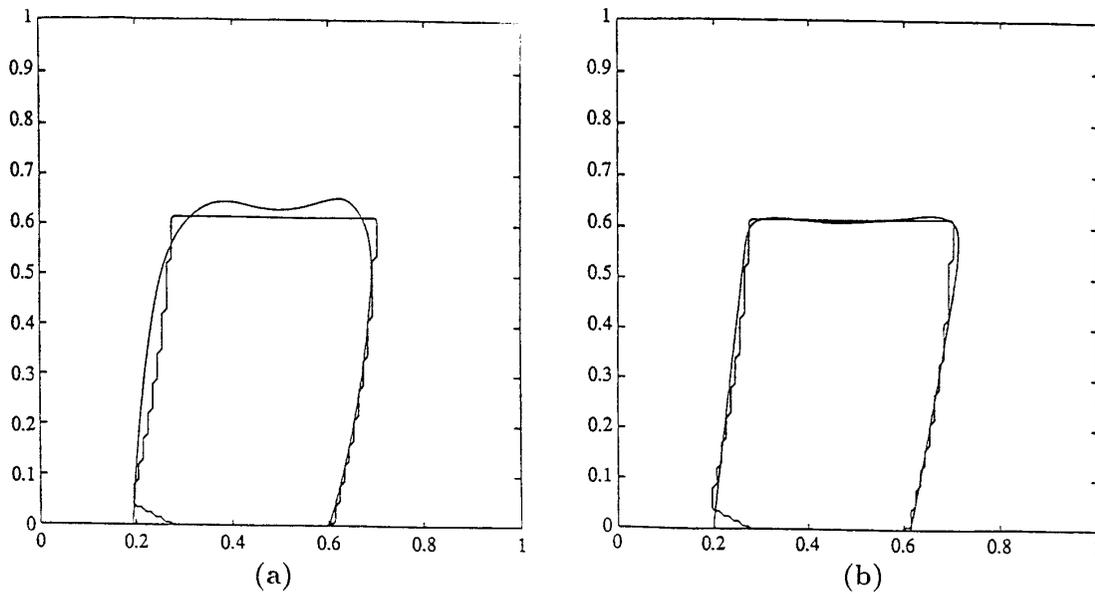


Figure 4: The boundary trace result for a 2-dim slice out of a 4-dim power security problem. (a) After trained by 5000 randomly selected training data. (b) After trained by 5000 query selected training data.