Fuzzy Control of Backpropagation

Payman Arabshahi * [1], Jai J. Choi †,
Robert J. Marks II *, Thomas P. Caudell ‡

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

† Boeing Computer Services
P.O. Box 24346, MS 6C-04
Seattle, WA 98124-0346

‡ Boeing Computer Services
P.O. Box 24346, MS 7L-22
Seattle, WA 98124-0346

Abstract

We propose a fuzzy logic controlled implementation of the backpropagation training algorithm for layered perceptrons. The heuristics for adjusting the values of the learning rate η are incorporated into a simple fuzzy control system. This provides automatic tuning of the learning rate parameter depending on the shape of the error surface. Application of this straightforward procedure is shown to be able to dramatically improve training time in some problems.

1 Introduction

The backpropagation learning algorithm [1], has been successfully applied to the training of multilayer feedforward neural networks in a number of practical problems. This algorithm is a gradient descent search in the space of weights of the neural network, and aims to minimize an energy function which is normally defined as the sum of squared errors, where each "error" is the difference between desired (target) values at the output of the network, and actual values obtained during each iteration of the algorithm. Weight changes are performed according to:

$$\Delta \mathbf{w}_n = -\eta \nabla E(\mathbf{w}_n) + \alpha \Delta \mathbf{w}_{n-1} \tag{1}$$

where $\mathbf{w}_n$ is the vector of weight values after the nth iteration; $\Delta \mathbf{w}_n$ is the change in these weights; $E(\mathbf{w}_n)$ is the error function at the nth iteration, η is the learning rate, and α is the momentum parameter.

[1]Electronic mail address: payman@milton.u.washington.edu

0-7803-0236-2 /92 $3.00

Despite the effectiveness of backpropagation, its speed of convergence can be painfully slow, rendering it ineffective in many practical situations where on-line learning is required. The reasons for this have been discussed in detail by Jacobs [2]. Jacobs also presented an overview of heuristics that can be used to accelerate the convergence of the algorithm, suggesting that each weight be given its own learning rate, and that this learning rate be allowed to change over time during the learning process. He also suggested how the learning rate should be adjusted, thus incorporating the heuristics into the *delta-bar-delta rule*. The heuristics suggested that if the error gradient possessed the same sign over several consecutive time steps, the value of η should be increased; and if the sign of ∇E alternated over consecutive time steps, the value of η should be decreased.

However, there are still no general guidelines for choosing the values of η and α, and it is up to the user to come up with values that provide for fast convergence. Fuzzy control of the learning rate η is suggested herein as a solution to this problem. The resulting procedure, although straightforward, can be remarkably effective.

2 Fuzzy Controlled Backpropagation

The central idea behind fuzzy control of the backpropagation algorithm is the implementation of heuristics used for faster convergence in terms of fuzzy IF ... THEN rules. This is done by considering the error E, and the change in error CE to be fuzzy variables taking on the values E_n and CE_n at each iteration n. These values can in turn be categorized as *low*, *medium*, and *high*. We also define a fuzzy variable $\Delta\eta$ taking on a value of $\Delta\eta_n$ at each iteration n representing the amount by which η is updated at the nth iteration. This can take on the values *negative small*, *zero*, and *positive small*. All of these values are expressed in terms of membership functions as shown in Fig.1. Based on the actual (crisp) values of E and CE, we can thus arrive at a (crisp) value for $\Delta\eta$, if we can express the relationship among these variables through fuzzy conditional statements. During each iteration of the backpropagation algorithm therefore, the value of the learning rate η is incremented by $\Delta\eta_n$ based on current values of the error and change in error (E_n and CE_n).

The rules chosen for this application are shown in the form of a table in Table 1. From this table for instance, one can read the following rule:

(1) IF E is low, THEN IF CE is low, THEN $\Delta\eta$ is positive small.

CE / E	Low	Medium	High
Low	PS	PS	ZE
Medium	PS	PS	ZE
High	ZE	ZE	NS

Table 1: Decision Table for the Fuzzy Controller. Table contents represent the value of the fuzzy variable $\Delta\eta$ for a given choice of values for E & CE

Evaluation of the rules is best illustrated by an example. Consider the rule (1) shown above. Having obtained (crisp) values for E and CE at the nth iteration, we evaluate their degree of membership in the membership functions of fuzzy sets defining their "values" (*low* and *low* in this case). The minimum value of these two evaluations is chosen and multiplied by the membership function of the consequent fuzzy set (*positive small* in our case), resulting in a *modified* membership function which we choose to represent by $\mu_k(x)$ for rule k. This is repeated for every rule in the rule base, and the modified membership functions $\mu_k(x)$ are summed together to form a composite function $\mu(x)$. The fuzzy centroid of $\mu(x)$ is then chosen as the deterministic control action (value of $\Delta\eta$) to be taken:

$$\Delta\eta_n = \frac{\int x\,\mu(x)\;dx}{\int\;\mu(x)\;dx}. \qquad (2)$$

3 Results

We present here results of comparisons between the convergence speed of standard backpropagation, Jacob's delta-bar-delta rule [2], and fuzzy controlled backpropagation as applied to the 3 bit parity problem. Batch training of the neural network is used. It is seen that the fuzzy controlled backpropagation algorithm is much faster than standard backpropagation or the Delta-bar-delta rule for this example.

4 Discussion

Although we have not explicitly implemented the heuristics presented in [2] by a fuzzy control system, nontheless the control rules can be interpreted as being derived from the general guidelines mentioned in [2]. While our approach in this implementation differs from that of Jacobs' delta-bar-delta rule, it is only natural to assume that whenever Jacobs' heuristics fail in a specific problem, the same will happen with the fuzzy backpropagation technique.

Also, just as backpropagation can not guarantee convergence to a globally minimum solution, neither can fuzzy controlled backpropagation. This is a problem inherent to a localized optimization technique such as steepest descent, of which backpropagation is a special case.

Finally we mention that, although a smaller number of iterations towards convergence is certainly desirable, this aspect of our solution should be cond-sidered jointly with the total number of operations required for each iteration. In this respect however, the total number of operations of the two techniques (plain and fuzzy backpropagation) are not significantly different from each other due to the inherent simplicity of the computations carried out in the fuzzy controller.

Acknowledgments

This work was supported in part by Boeing Computer Services and the Washington Technology Center.

References

[1] D.E. Rumelhart and J.L.McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations* (MIT Press, 1986).

[2] R.A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural Networks,* **1** (1988) 295-307.

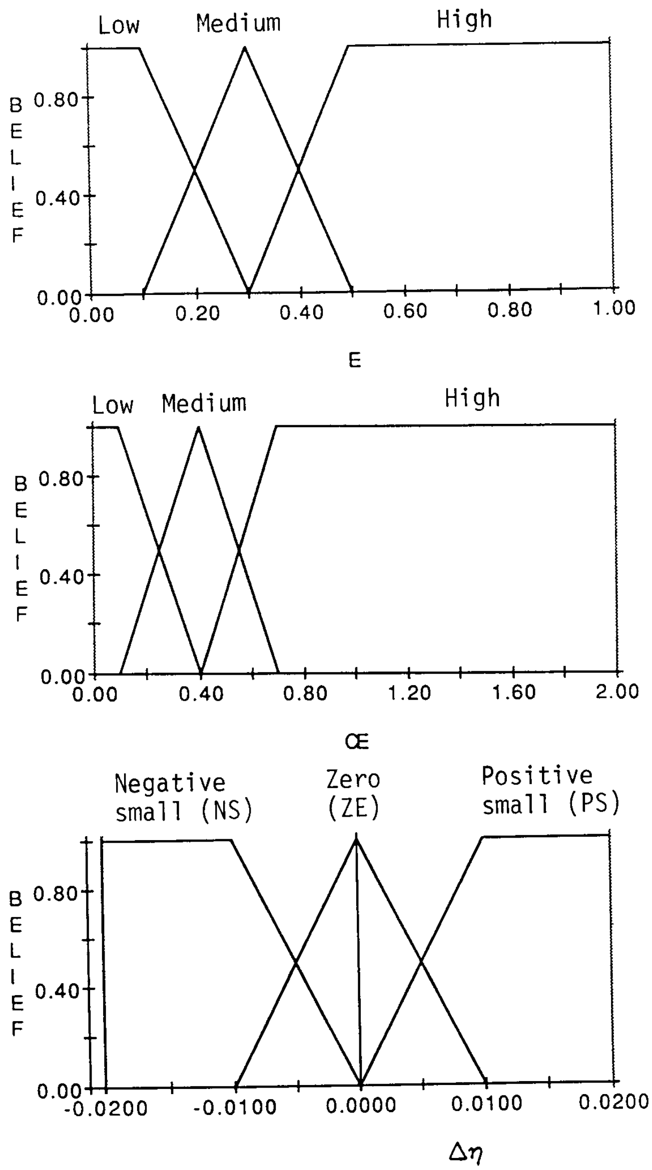

Figure 1: (a) Membership functions for the error (E) and change in error (CE) (b) Membership functions for $\Delta\eta$, the incremental update to the learning rate η