J.J. Choi, S.Oh & R.J. Marks II, "Training layered perceptrons using low accuracy computation",
Proceedings of the International Joint Conference on Neural Networks, Singapore, 18-20 Nov 91, IEEE Press, pp.554-559.

Training Layered Perceptrons Using Low Accuracy Computation

Jai J. Choi
Boeing Computer Services, P.O.Box 24346, 6R-08, Seattle, WA 98124

Seho Oh, Robert J. Marks II
Dept.EE, FT-10, Univ. of Washington, Seattle, WA 98195

Abstract

A fundamental limitation of the use of error back propagation (BP) in the training of a layered feed forward neural network is the high degree of required computational accuracy. For each iteration, the weights typically change at low significant digits. Thus, layered perceptrons cannot be trained with error back propagation using low accuracy analog circuitry. Since analog implementations of layered perceptrons perform quite fast in comparison with their digital counterparts, this is indeed unfortunate. The training of the layered perceptron, however, is simply a minimization search in weight space to which any one of a number of search algorithms can be applied. Certain other search approaches do not require the computational precision needed by error back propagation. In this paper, we demonstrate the *random search* approach to training layered perceptrons can be performed using low accuracy computational precision and, therefore, can be implemented using analog computational accuracy. In spite of numerical stability, random search techniques suffer from ever increasing search time as dimensionality grows. In response, we introduce a modified random search technique (Improved Bidirectional Random Optimization - IBRO) to improve the search accuracy per iteration. Consequently our scheme will reduce overall search iterations dramatically. We compare the performance of IBRO with that of the bidirectional random optimization (BRO) method through simulations.

1 Introduction

Finding a global minimum of cost or error function is a problem common to many applications. In contrast to steepest descent approaches, the random optimization (RO) technique [1, 3, 5, 6, 9] is an optimization technique for which no gradient information is required. The approach is known for its simplicity, modest computational complexity and low computational accuracy demands, low memory requirement, and its capability of handling multi-modal objective functions.

The training of a layered perceptron can be viewed as a minimization search in weight space to which any search algorithms can be applied. The back propagation (BP) algorithm seeks a minimum based on the local gradient of the objective function. A fundamental limitation of the use of BP is the high degree of required computational accuracy. For each iteration, the weights typically change at low significant digits. Thus, layered perceptrons cannot be trained with BP using low accuracy analog circuitry.

In this paper, we perform a preliminary investigation into the performance of RO as a layered perceptron training algorithm. We experimentally demonstrate that random search methods can be used to train a layered perceptron using low accuracy computations.

CH 3065-0/91/0000-0554 $1.00 © IEEE

2 Random optimization method

The RO technique was first introduced in 1953 by Anderson [1] and developed by Rastrigin [7], Karnopp [5], Matyas [6], and Solis [9]. Recently Baba [3] suggested its use in training feedforward multi-layer perceptron.

Although there have been numerous variations on this approach, the basic idea behind RO can be explained as follows. Starting from an initial point $\vec{x}(0)$ in the weight space, we compute $\hat{x}(0) = \vec{x}(0) + \vec{\xi}(0)$ where $\vec{\xi}(0)$ is a randomly chosen vector. A typical choice of $\vec{\xi}(0)$ is a Gaussian random vector. The objective function $f(\hat{x}(0))$ is then evaluated. If an improvement in the cost function[1] is measured, then $\vec{x}(1) = \hat{x}(0)$. Otherwise, $\vec{x}(1) = \vec{x}(0)$.

One of the most attractive features of the RO method is that the resulting search converges in probability to the *global* minimum. Specifically, for many probability distributions, including Gaussian and Laplace, *the sequence of search vector* $\{\vec{x}(k)\}_{k=1}^{\infty}$ *converges to the global minimum point* $\vec{x}_{opt}$ *in the sense that,*

$$\lim_{k\to\infty} \Pr[\,\rho(\vec{x}(k), \vec{x}_{opt}) > \delta\,] = 0,$$

where $\rho(\,\cdot\,,\,\cdot\,)$ represents the distance measure between the two vectors and δ is a small positive number ($\delta > 0$). The proofs are in references [2], [6], and [9].

Another important advantage of the random search method is its simplicity. It does not require evaluation of gradient information that, for example, is required in the BP algorithm. The RO algorithm requires no imposition of learning parameters except for the variance of the random number generator. By applying this method to train a multilayer feed forward network, Baba [3] empirically has shown the effect of the variance of Gaussian random vector. Obviously, a small variance will confine the scope of the search space and result in a slower albeit more accurate search.

3 Improved random optimization method

Various modifications of the basic RO algorithm are possible. The bidirectional random optimization (BRO) method searches both reverse ($\vec{x}(k) - \vec{\xi}(k)$) and forward ($\vec{x}(k) + \vec{\xi}(k)$) direction to compare the objective function. This method has been empirically proven to be more effective than the conventional RO method [3, 9]. The BRO also keeps track of the center of random search vectors $\vec{b}(k)$ as shown in the Table 1 (the introduction of $\vec{b}(k)$ is due to Matyas [6]).

Step 1: Initialize $\vec{x}(0)$, $\vec{b}(0) = \vec{0}$, and set $k = 0$.

Step 2: Obtain $\vec{\xi}(k)$. The mean of the random vector $\vec{\xi}(k)$ is $\vec{b}(k)$.

Step 3: Let

$a_0 = f(\vec{x}(k))$

$a_{-1} = f(\vec{x}(k) - \vec{\xi}(k))$

$a_1 = f(\vec{x}(k) + \vec{\xi}(k))$.

Step 4:

if a_{-1} is minimum, $\vec{x}(k+1) = \vec{x}(k) - \vec{\xi}(k)$ and $\vec{b}(k+1) = \vec{b}(k) - 0.4\vec{\xi}(k)$.

---

[1] In the case of the layered perceptron, the cost function is the error function.

if a_1 is minimum, $\vec{x}(k+1) = \vec{x}(k) + \vec{\xi}(k)$ and $\vec{b}(k+1) = 0.2\vec{b}(k) + 0.4\vec{\xi}(k)$.
if a_0 is minimum, $\vec{x}(k+1) = \vec{x}(k)$ and $\vec{b}(k+1) = 0.5\vec{b}(k)$.

Step 5: Set $k = k+1$ and return to **Step 2**.

Table 1 Bidirectional random optimization (BRO) algorithm (Solis & Wets, 1981).

We modified the BRO to increase the search probability in every iteration. Let's assume the situation where both forward and backward search fails to find lower objective function values as illustrated in Fig. 1. Define, $F_\alpha = \{\vec{y} \mid f(\vec{y}) < f(\vec{x})\}$, and

$$\alpha = \frac{f(\vec{x} - \vec{\xi}) - f(\vec{x} + \vec{\xi})}{f(\vec{x} - \vec{\xi}) + f(\vec{x} + \vec{\xi}) - 2f(\vec{x})}.$$

For $|\alpha| < 1$, if $\vec{x} + \alpha\vec{\xi} \in F_\alpha$, then we can increase the search probability and thus improve BRO with relatively low additional computing costs. Notice that the only additional computation is **Step 5** shown in Table 2. We compare our improved BRO (IBRO) to BRO to demonstrate the speed upgrade by simulating 6-bit parity problem. The corresponding learning curves, shown in Figure 2, are obtained through 10 ensembles of experiments.[2] The learning gains were 0.3, 0.5, 0.75 and the momentum gain was chosen to be between 0.3 and 0.8 for the BP algorithm. The feedforward network has one hidden layer with sigmoidal non-linear units. Different numbers of hidden units were employed (from 5 to 15 units). Sometimes the simulation never escapes local minima, and we restart with a different configuration. *The IBRO, on the other hand, never fails to reach the zero error value (global minimum).* We use 0.1, 0.01, 0.001 variances for the Gaussian random variables.

Step 1: Initialize $\vec{x}(0)$, $\vec{b}(0) = \vec{0}$, and set $k = 0$.

Step 2: Obtain $\vec{\xi}(k)$. The mean of the random vector $\vec{\xi}(k)$ is $\vec{b}(k)$.

Step 3: Let

$$a_0 = f(\vec{x}(k))$$
$$a_{-1} = f\left(\vec{x}(k) - \vec{\xi}(k)\right)$$
$$a_1 = f\left(\vec{x}(k) + \vec{\xi}(k)\right).$$

Step 4:

if a_{-1} is minimum, $\vec{x}(k+1) = \vec{x}(k) - \vec{\xi}(k)$ and $\vec{b}(k+1) = \vec{b}(k) - 0.4\vec{\xi}(k)$. Go to **Step 6**.
if a_1 is minimum, $\vec{x}(k+1) = \vec{x}(k) + \vec{\xi}(k)$ and $\vec{b}(k+1) = 0.2\vec{b}(k) + 0.4\vec{\xi}(k)$. Go to **Step 6**.
if a_0 is minimum, i.e., ($a_{-1} > a_0$ *and* $a_1 > a_0$), Go to **Step 5**.

Step 5: Compute $\alpha = (a_{-1} - a_1)/(a_{-1} + a_1 - 2\,a_1)$ and let $a_\alpha = f\left(\vec{x}(k) + \alpha\,\vec{\xi}(k)\right)$.
if $a_\alpha < a_0$, then $\vec{x}(k+1) = \vec{x}(k) + \alpha\vec{\xi}(k)$ and

[2]The comparison of performance in terms of iterations is, in some sense, unfair. The computational complexity of the BRO and IBRO are significantly less on a per iteration basis than that of a BP iteration step.

$\vec{b}(k+1) = \alpha\vec{b}(k) - 0.4\vec{\xi}(k)$ **when** $-1 < \alpha < 0$.
$\vec{b}(k+1) = 0.2\alpha\ \vec{b}(k) + 0.4\vec{\xi}(k)$ **when** $0 < \alpha < 1$.
$\vec{b}(k+1) = 0.5\ \vec{b}(k)$ **when** $\alpha = 0$.
Else $\vec{x}(k+1) = \vec{x}(k)$ **and** $\vec{b}(k+1) = 0.5\ \vec{b}(k)$.
Step 6: Set $k = k+1$ and return to **Step 2.**

Table 2 Improved bidirectional random optimization algorithm.

4 Effect on finite word length representation of weight values

In this section, we experimentally show the sensitivity of RO to finite word length. The ability to use RO at low precision levels removes a significant obstacle for classifier and regression machine training using analog technology. For example, when the LMS algorithm is implemented in VLSI using fixed point arithmetic, experiments show that approximately 16 - 20 bits are required to represent weight coefficients in order to achieve necessary performance [10].

The random search methods are, by their very nature, numerically stable. Clearly an algorithm that does not require the function to be accurately computed is unlikely to depend on exact arithmetic utilizing these function values. We only need to evaluate the function if the objective value of the new search point is lower or higher than the previous one.

The simulation of low precision is performed by truncating the weight values at the n^{th} digit after the decimal point. To illustrate, consider the 5-bit parity problem. A typical learning curve is shown in Figure 3 for different degrees of truncation. The BRO method exhibits convergence on finite word representation up to first digit truncation. The BP method, however, was not able to perform consistent learning after 5^{th} digit truncation.

References

[1] R. Anderson, "Recent advances in finding best operating conditions", *J. Amer. Statist. Assoc.*, Vol. 48, pp. 789–798 (1975).

[2] N. Baba and T. Shoman, "A modified convergence theorem for a random optimization method," *Information Sciences*, Vol. 13, pp.159–166 (1977).

[3] N. Baba, "A new approach for finding the global minimum of error function of neural networks," *Neural Networks*, Vol. 2, pp.367–373 (1989).

[4] R.A. Jarvis, "Adaptive global search by the process of competitive evolution," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC–5, No. 8, pp.297–311 (1975).

[5] D.C. Karnopp, "Random search techniques for optimization problems," *Automatica*, Vol. 1, pp.111–121 (1963).

[6] J. Matyas, "Random optimization," *Automation and Remote Control*, Vol.26, pp.246–253 (1965).

[7] L.A. Rastrigin, "The convergence of the random search method in the extremal control of a many-parameter system," *Automation and Remote Control*, Vol. 24, pp.1337–1342 (1963).

[8] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, *Parallel Distributed Processing (Vol.1)*, MIT Press, Cambridge, Massachussetts, 1986.

[9] F.J. Solis, and J.B. Wets, "Minimization by random search techniques," *Mathematics of Operation Research*, Vol. 6, pp.19–30 (1981).

[10] M.S. Song, P. Yang, and K. Shenoi, "Nonlinear compensation for finite word length effects of an LMS echo canceler algorithm suitable for VLSI implementation," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pp.1487–1490 (1988).

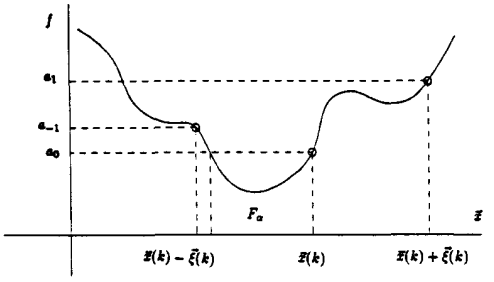

Figure 1: Improved bidirectional random search.

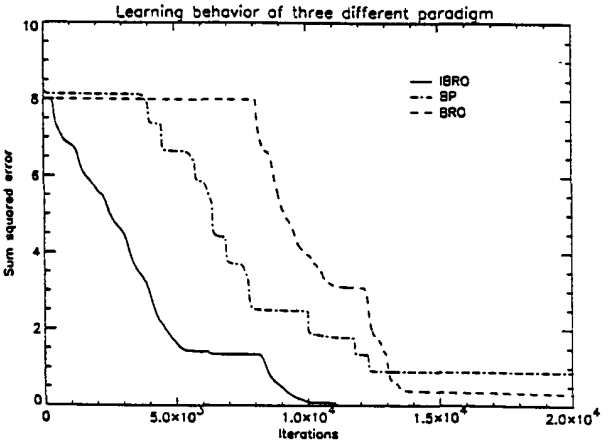

Figure 2

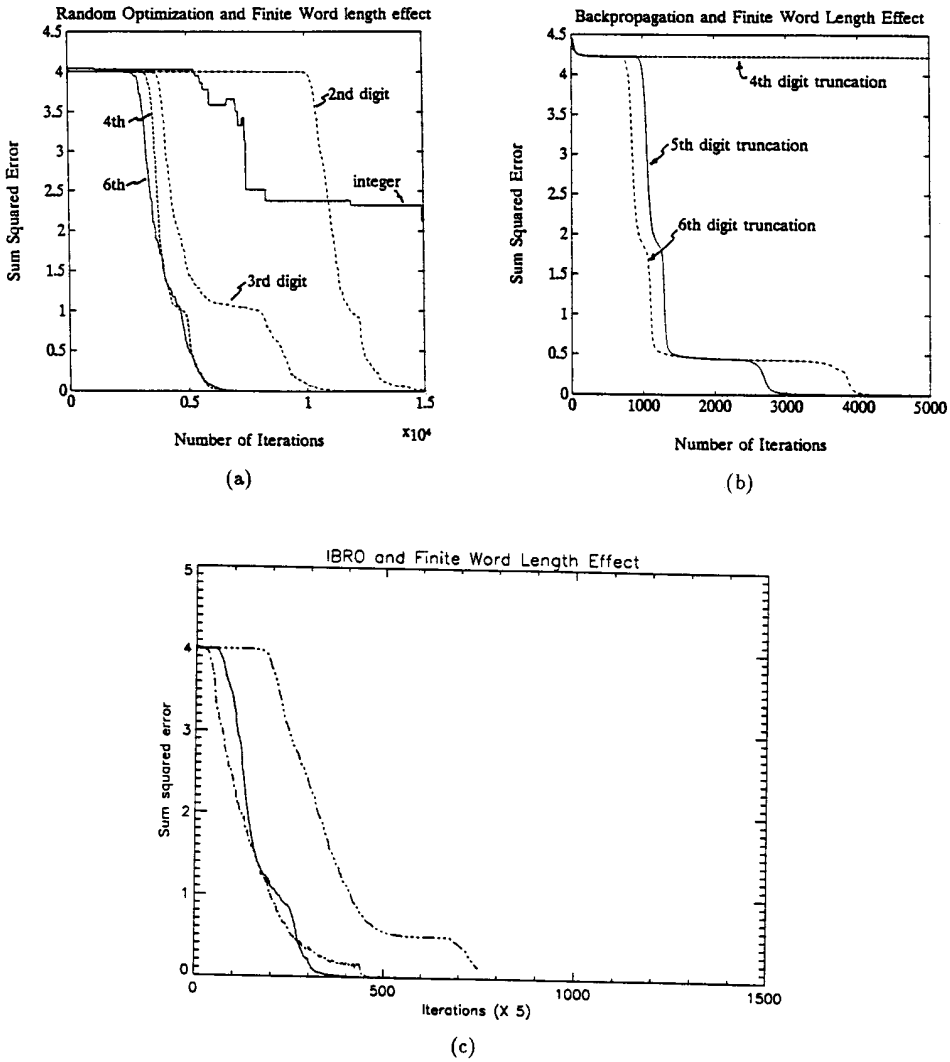

Figure 3: Effects of finite word representation of weight values for the BRO method(a), BP(b) and IBRO(c). 5 bit parity problem is chosen.