CHARACTER RECOGNITION USING A SELF-ADAPTIVE TRAINING

Dr. Eng. Shamsuddin Ahmed

College of Business and Economics (AACSB Accredited)
United Arab Emirates University, P O Box 17555, Al Ain, UAE.

and

$Edith Cowan University
School of Engineering and Mathematics
100-Joondalup drive, Joondalup, Perth, WA-6027, Australia.

Email: Dr_Shamsuddin_Ahmed@Yahoo.Com

ABSTRACT

A robust algorithm to train neural network (NN) with individual character presented in pixel format is described. The output of the trained NN is related with the pixel binary format. The trained NN performs the classification based on the original pixel format code presented to the NN. A mapping scheme is proposed to determine the self-adaptive variable learning rates for rapid converge. The proposed character recognition method trains at a faster rate than that of the standard back propagation training method and prevents oscillations during training.

Key Words: Self-adaptive training, ANN, Dynamic variable learning rates, Character recognition.

1. Introduction

A self-adaptive ANN training scheme that identifies variable learning rates is discussed in this study. Consider an ANN error function containing $m$ training weights. The actual error function in higher dimension is decomposed into lower dimensional error function (Ahmed et al., 2000). The transformed error function retains true convex characteristics of the original error function. The training scheme selects the learning rate for a weight parameter say $w_j$, $j=1,2,.....m$, by a small factor, $\Delta w_j$, such that improvement in training is noted. Once this phase is over, all the network weights are then updated and the error function is evaluated.

1.1 Character Recognition Problem

A training experiment is considered to recognize the letters $L$ and $T$. An ANN with nine inputs, two hidden units and a single output unit is considered for this problem. Each input pattern consists of a 3x3 pixel binary image of the letter. The training set is formed by four orientations of each letter as shown in Figure 1 and Figure 2. A 9-2-1 ANN configuration is chosen to compare the results given in Kamarthi and Pittner (1999).

Figure 3 shows the variable learning rates per epoch. As the training continues the variable learning rates as seen in Figure 4 converges to lower magnitude. Consequently the training converges rapidly. The error function value reduces of the order $10^{-3}$ at the early stage of training and further trained at $10^{-6}$ tolerance limit for accuracy. Notice that the function converges as the self-adaptive training parameters gradually reduce to small magnitude. Figure 5 shows training
performance with standard back propagation (BP) training. The training performance is compared in Figure 6 with the standard BP training. The binary input is passed through the 9-2-1 ANN (Figure 7) input neurons and output neuron provides the necessary signal to recognize the L-T input pattern. It is evident from Figure 3 that considerable reduction in error function is achieved within 22 epochs to train the problem. The standard BP takes on average 2178 epochs to train the character recognition problem, while the proposed method takes on average 179 epochs.

2. Identifying self-adaptive learning rates

To describe the self-adaptive iterative training method, an ANN error function $f(w)$ with log activation function is defined in hidden layer. When the algorithm is applied to the error function with an initial arbitrary weight vector $w_k$, at epoch $k$, the algorithm generates a sequence of vectors $w_{k+1}, w_{k+2}, \ldots$ during epoch $k+1, k+2, \ldots$, and so on. The iterative algorithm is **globally convergent** if the sequence of vector converges to a feasible solution set $\Omega$. Consider for example the following training problem, where $w$ is defined over the dimension $E^m$:

$$\text{minimize } f(w) \quad (1)$$

subject to: $w \in E^m$.

Let, $\Omega \subset E^m$ be the solution set, and the application of an algorithmic interpolation map, $B$, generates the sequence $w_{k+1}, w_{k+2}, \ldots$, starting with weight vector $w_k$ such that $(w_k, w_{k+1}, w_{k+2}, \ldots) \in \Omega$, then the algorithm converges globally and the algorithmic map is closed over $\Omega$.

Consider further that the error function is a minimization problem, since it is possible to train such function (Haykin, 1994). Let $\Omega$ be a non-empty compact feasible subset of $E^m$, and if the algorithmic map $B$ generates a sequence: $(w_k) \in \Omega$ such that $f(w)$ decreases at each iteration while satisfying $f(w_k) > f(w_{k+1}) > f(w_{k+2}), \ldots$, then the error function $f(w)$ is a descent function. The expression $w_k$ defines a sequence $(w_1, w_2, w_3, \ldots, w_n)$. In ANN computation $f(w)$ is assumed to posses descent properties, which implies it is convex in nature (Hykin, 1994).
Therefore, it is possible to define a descent direction along which the error function can be trained. Following sections demonstrate how the self-adaptive variable learning rates are generated, such that the training direction is always correct.

**Property 1**

Suppose that \( f : E^m \rightarrow E^1 \) and the gradient of the error function, \( \nabla f(w) \), is defined then there is a directional vector \( d \) such that \( \nabla f(w)^T d < 0 \), and \( f(w + \eta d) < f(w) : \eta \in (0, \delta), \delta > 0 \), then the vector \( d \) is a descent direction of \( f(w) \), where \( \delta \) is assumed arbitrary positive scalar.

A vector defined as directional derivative conceptualizes the direction along which the error function converges to minimum. Also, assume that the error function is smooth and continuous. Therefore the negative gradient of the error function points towards the minimum ANN error space. Next the directional derivative is defined using the following property.

Let, \( f : E^m \rightarrow E^1 \), \( w \in E^m \) and \( d \) is a non-zero vector satisfying \( (w + \eta d) \in E^m \), \( \eta > 0 \) and \( \eta \rightarrow 0^+ \). The directional derivative at location \( w \) along the descent direction \( d \) is given by:

\[
\nabla f(w; d) = \lim_{\eta \rightarrow 0^+} f(w + \eta d) - f(w) \over \eta .
\]

**Property 2**

Let \( f : E^m \rightarrow E^1 \) is a descent function. Consider any training weight \( w \in E^m \) and \( d \in E^m : d \neq 0 \). Then the directional derivative \( \nabla f(w; d) \) of the error function \( f(w) \) in direction \( d \) always exists.

The central theme behind the self-adaptive BP training is the computation of the directional search vector \( d \) and the learning rate parameter \( \eta \) in \( m \) weight space. Since the exact location of the minimum valley is not known immediately, there is an uncertainty in identifying the boundary or region in weight space over which the training may explore. The uncertainty can be reduced if it is possible to eliminate the sections of the error function (Kiefer, 1953), which do not contain the minimum. An interpolation-training map can do this operating over the error function in constrained interval. Hence a closed mapping is established. Therefore, what is needed is a training scheme that explores the constrained region, \( L \), of the error surface. Consider the interpolation map, \( B \), samples the error surface with discrete length in a given direction. Few definitions are needed to describe the interpolation-training map. For convenience define the following expression to update the ANN connection weights:

\[
u_k = w_{k+1} = w_k + \eta_k d_k \quad \text{or} \quad u = w + \eta d .
\]

Now consider an ANN training problem expressed as:

\[
\eta_k = \arg \{ \min f(w + \eta d) \}
\]

subject to : \( \eta_k \in L \)

where, \( L \) defines a closed interval \( L = \{ \eta : \eta \in E^1 \} \) over the error function.
Property 3
The interpolation map that computes learning rates is defined in restricted error space written as:

\[ A: E^m \times E^m \rightarrow E^m \] such that:

\[ A(w, d) = \{ u : u = (w + \eta \ d) | \eta \} \in L \]  \hspace{1cm} (5)

\[ f(u) = \min f(w + \eta \ d) \] \hspace{1cm} (6)

Subject to: \( \eta^* \equiv \eta \in L \).

Suppose that the algorithm B operating on \( f(w) \) produces descent directions. The correct learning rate is obtained during the training phase, when the map is closed as set value mapping (Luenberger, 1984).

It is, therefore, now possible by any standard technique; though difficult; to identify learning rate, \( \eta_k \), for each epoch \( k \) while descending along the minimum trajectory and exploring all the \( m \) dimensions. The error function as a result monotonically converges to a minimum value. The algorithm gradually converges to the acceptable minimum limit in few epochs.

2.1 Training

One of the most difficult computational steps in identifying the trained weights in ANN is that the ANN error function contains several local minimums within the reasonable range of \( w_j \). The reduced ANN error function can be considered a continuous function of \( m \) training weights \( w_j, j=1,2,3,\ldots, m, \) describing a hyper surface in \( m \) dimensional space (Moller, 1997). Suitable reduction of the error surface constitutes a parabolic hyper surface, and the learning rates are determined from the reduced function with initial random weight estimated at the beginning of training. As a result, few epochs are needed to converge to the minimum trajectory of the error function.

3. Character recognition problem

Ten simulation experiments are carried out to test the proposed training method against the standard BP training method (Rumelhart et al., 1989) with small random starting weights. Simulation results are shown in Table 1 and Table 2. The small random starting weights are used to initiate the training. Table 1 shows the result against the standard BP training method. The average number of epoch with the self-adaptive back propagation method is 178.8. The maximum and minimum values in numbers of epoch are 323 and 53 respectively. The average terminal function value is \( 1.28 \times 10^{-9} \), which indicates that the ANN is trained to produce accurate results.

The average number of epoch is 2177.7 with the standard BP training. The maximum and minimum numbers of epoch are 3165 and 802 respectively. The relative efficiency of the self-
adaptive back propagation-training algorithm over the standard BP-training algorithm in number of epoch is (2177.7/178.8) 12.18.

Table 2 shows the comparison with different training methods including the result given in Kamarthi et al. (1999). The proposed method performs better than the weight extrapolation method suggested by Kamarthi et al. (1999). They train L-T letter recognition problem with 1811 epoch and the terminal function value is 0.00001 at the end of training. The self-adaptive training method finds terminal function value, which is comparatively less than the weight extrapolation and standard BP training method.

![Table 2](image)

**Table 2 Training performance with 9-2-1 ANN: L-T letter recognition problem**

<table>
<thead>
<tr>
<th>Expt #</th>
<th>Self-adaptive back propagation</th>
<th>Standard back propagation</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-T Letter Recognition</td>
<td>Epoch</td>
<td>Function value</td>
</tr>
<tr>
<td>Mean</td>
<td>178.8</td>
<td>1.28E-09</td>
</tr>
<tr>
<td>Median</td>
<td>131.5</td>
<td>5.2E-10</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>97.54406</td>
<td>2.27E-09</td>
</tr>
<tr>
<td>Range</td>
<td>270</td>
<td>7.54E-09</td>
</tr>
<tr>
<td>Minimum</td>
<td>53</td>
<td>9.06E-11</td>
</tr>
<tr>
<td>Maximum</td>
<td>323</td>
<td>7.63E-09</td>
</tr>
</tbody>
</table>

Table 1 Comparison with standard back propagation and self-adaptive training method (9-2-1 ANN: L-T letter recognition problem)

**CONCLUSION**

A strictly self-adaptive training scheme to train forward ANN is shown. The important feature of this ANN training is that the learning rates are dynamically computed each epoch by an interpolation map. The ANN error function is transformed into a lower dimensional error space and the reduced error function is employed to identify the variable learning rates. As the training progresses the geometry of the ANN error function constantly changes and therefore the interpolation map always identifies variable learning rates that gradually reduces to a lower magnitude. As a result the error function also reduces to a smaller terminal function value. It improves over the BP training method and the weight interpolation method demonstrated by Kamarti et al., (1999) in character recognition problem. The proposed self-adaptive training method needs 179 epochs, while the standard BP method needs 2178 epoch and the weight extrapolation method needs 1811 epoch to train the L-T character recognition problem with 9-2-1 ANN configurations.

**REFERENCES**

Computations, 2000, Berlin, Germany


