Implementation of a variable step size backpropagation algorithm

1Deepak Gupta, 2Ravi Kumar
Electronic & communication
Thapar University Patiala, India
Email: ravi.kumar@thapar.edu

Abstract—This paper reports the effect of the step-size (learning rate parameter) on the performance of the backpropagation algorithm. Backpropagation algorithm (BP) is used to train multilayer neural network. BP algorithm is the generalized form of the least mean square (LMS) algorithm. In this proposed backpropagation algorithm different learning rate parameter are used in different layer. The learning rate parameter of a backpropagation algorithm is an important parameter as it determines the amount of correction applied as the network adapts from one iteration to the next. Choosing the appropriate step size is not always easy, usually requiring experience in neural network design. Hardware implementations of the proposed backpropagation algorithm are also shown in this work. For effective hardware utilizations artificial neural model is based on serial processing structure.

Keywords—Neural network, Backpropagation algorithm, Least mean square algorithm, learning rate parameter.

I. INTRODUCTION

Neural networks works as self learning machines whose synaptic weights adjust themselves to the changing environment. The working of the neural network is inspired by the biological neural network. Neural network is used in many application such as pattern recognition, speech classification, feature extraction, chip failure analysis etc [1][2].

Synaptic weights of the neural network are adjusted using adaptive algorithm [3].The learning capability of the neural network is largely dependent on the training algorithm. The multilayer neural network is trained using backpropagation algorithm. In Function approximation, feature extraction and pattern recognition application multilayer neural network is trained using backpropagation algorithm [4].Multilayer neural network consists three layer. The first layer is called input layer, second hidden layer and third is output layer [5]. The number of hidden layer and number of neuron in the hidden layer are also dependent on the training algorithm. The learning capability of the training algorithm is highly dependent on learning rate parameter (Step-size).

The step size (η) of the training algorithm should be selected such that it is neither too small nor too large, as it affects the performance of the training algorithm by changing the convergence rate, satiability and mean square error (MSE). A small step-size is increased the time for the neural network to learn but accuracy increases. On the other hand learning rate parameter is too high the training algorithm become unstable and never be converge to the goal so Learning rate parameter is behave like a memory for the training algorithm. As a rule of thumb the choice of the learning rate parameter decides the convergence and stability of the training algorithm.

This paper is organized in the following four sections. The first sections describe in brief backpropagation algorithm. Next section is the proposed backpropagation algorithm Experimental results and discussion have been presented in the subsequent section and finally conclusions drawn from this study have been summarized.

II. BACKPROPAGATION ALGORITHM

In back-propagation learning synaptic weight are modified by presynaptic activity and error signal. These signals are independent from the post synaptic activity. The conventional back propagation algorithm is used following equation to update the weight of the hidden and output neurons when a finite length input patterns are given:

\[ \Delta w_{ij}(n) = \Delta w_{ij}(n-1) + \eta \delta_j(n) y_i(n) \]  

(1)

Where, \( \eta \) is the learning rate parameter and \( \Delta w_{ij}(n) \) is the change in weight incident on neuron j from neuron i.

\[ \Delta w_{ij}(n) = \eta \delta_j(n) y_i(n) \]  

(2)

Where, \( \delta_j(n) \) is the gradient of the error for output neuron j at iteration n and is given by

\[ \delta_j(n) = e_j(n) \| y_j(n) \| \]  

(3)

Similarly, gradient of the error for hidden layer neuron j at iteration n is given by
\[ \delta_i(n) = \left[ \sum_k \delta_k(n) w_{ik}(n) \right] w_i(n) \] (4)

The above mentioned backpropagation algorithm has two stage operation. In Forward stage input vector is applied to the input node and response are propagated through the network. In backward stage synaptic weights of the layer are updated using the above mentioned four equation. The backpropagation algorithm uses single learning rate parameter for each layer.

IV. PROPOSED BACKPROPAGATION ALGORITHM

In the proposed backpropagation algorithm different learning rate parameter value is used for each layer. The equation (3) and (4) in section II shows that local gradient term for each layer is different. Local gradient term for the hidden layer is less compare to output layer. The reason behind this is that small portion of the error signal is propagated to the hidden layers. The convergence of the network is highly dependent on the weights of the hidden layer. Larger learning rate will be help to obtained optimum weight of the hidden layer. The error signal is high for output layer so learning rate parameter should be less compare to a hidden layer. The modified weight update term for each layer is given by the following equation:

\[ \Delta w_{ij}(n) = \Delta w_{ij}(n-1) + \eta_2 \delta_i(n) y_j(n) \] (5)

\[ \Delta w_{ij}(n) = \Delta w_{ij}(n-1) + \eta_2 \delta_i(n) y_j(n) \] (6)

Where \( \eta_2 \) is grater then the \( \eta_1 \). The equation (4) is the weight update term for the output layer and equation (5) is the weight update term for the hidden layer. The Modified weight update equation will be help to improve convergence rate and structure complexity.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The present work has been carried out on MATLAB® platform and Xilinx ISE 14.5. The proposed algorithm has been implemented on Spartan-3E Fpga. The simulation is performed on the neural network toolbox. In this work 4-3-3 neural structure are chosen for implement this work. The architecture is optimized for the classification of IRIS data using backpropagation algorithm implemented on a MATLAB program. Fixed point number system is used for encoding the neural network parameters and performing the computations. Serial structure is used for neuron. Serial learning is used in this work.

![Neuron serial processing computing model.](image)

Fig. 1. Neuron serial processing computing model.

<table>
<thead>
<tr>
<th>TABLE I. TIMING REPORT OF 4-3-3 NETWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clock frequency(MHz)</strong></td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>86.663</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II. DEVICE UTILIZATION SUMMERY OF 4-3-3 NETWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logic Utilization</strong></td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>Number of Slice Flip Flops</td>
</tr>
<tr>
<td>Number of 4 input LUTS</td>
</tr>
<tr>
<td>Number of occupied Slices</td>
</tr>
<tr>
<td>Number of MULT18X18s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III. POWER ANALYSIS OF 4-3-3 NEURAL NETWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static Power (mw)</strong></td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>52.43</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

The proposed algorithm is useful in terms of hardware implementation and faster convergence. Different step-size for each layer not only reduced the number of neurons in the layers but also reduces the local minima problem. The local minima prevent the network to reach the goal. The optimum value of the learning rate parameter for each layer can be found using trial method. This approach can be useful in pattern recognition, function approximation and data compression.

REFERENCES


