

COMPARATIVE STUDY ON ARTIFICIAL NEURAL NETWORK ALGORITHMS

Neenu Joseph¹, Melody M Fernandes² and Mr. Santhosh Rebello³

Abstract- We want our computers to perform many pattern recognition problems that are complex. We borrow features from the physiology of the brain as the basis for our new processing models, since our conventional computers are not suited to these types of complex pattern recognition. Hence, the technology has come to be known as artificial neural systems (ANS) technology, or simply neural networks. Various algorithms are used for the analysis of artificial neural networks. In this paper we are comparing SLP and MLP algorithms used in artificial neural networks.

Keywords –*Artificial Intelligence, Machine Learning, Algorithms, Neural Computing, Pattern Recognition*

I. INTRODUCTION

Neural Networks is a field of Artificial Intelligence (AI) that finds data structures and algorithms for learning and classification of data which is imitated from that human brain. Computer with conventional programming methods cannot perform tasks that humans perform naturally fast, such as the recognition of a familiar face. A program can learn by examples, by applying Neural Network techniques. It can create an internal structure of rules to classify different inputs, such as recognizing images.

In this paper we have presented different multi artificial neural network algorithms and its comparative study.

II. PROBLEM STATEMENT

A comparative study of single layer perceptron network and multilayer perceptron network algorithm and we will conclude on an algorithm that is better for learning techniques.

III. NEURAL NETWORK ARCHITECTURE ALGORITHMS

A. *The Simple Neuron Model-The Single Layer Perceptron(SLP)–*

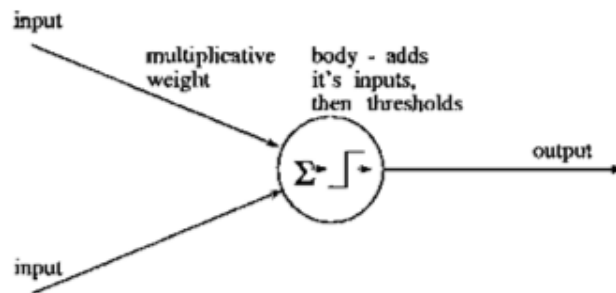
The simple neuron model is made by imitating the human brain neurons. Through dendrites a neuron in the brain receives chemical inputs from other neurons. If the threshold is exceeded, the

¹ *Aloysius Institute of Management and Information Technology St.AloysiusCollege(AUTONOMOUS) Beeri, Kotekar, Mangalore, Karnataka, India.*

² *Aloysius Institute of Management and Information Technology St.AloysiusCollege(AUTONOMOUS) Beeri, Kotekar, Mangalore, Karnataka, India.*

³ *Aloysius Institute of Management and Information Technology St.AloysiusCollege(AUTONOMOUS) Beeri, Kotekar, Mangalore, Karnataka, India.*

neuron fires its own impulse on to the neurons it is connected to by its axon. Below is a very simplified figure of the neurons of the brain since it is connected to about 10000 other neurons. The simple perceptron models this behaviour in the following way. First the perceptron receives several input values ($x_0 - x_n$). The connection for each of the inputs has a weight ($w_0 - w_n$) in the range 0-1. The Threshold Unit sums the inputs, and if the sum of the inputs exceeds the threshold value, a signal is sent to output. Otherwise the signal is not sent. The SLP learns by adjusting the weights to achieve the desired output.



With one perceptron, it is only possible to distinguish between two pattern classes, with the visual representation of a straight separation line in pattern space

Algorithm:

- a. Initialize the weights and bias to zero. α is set to 1 ($\alpha=1$)
- b. Perform steps c to g till the stopping condition is false
- c. Perform steps d to f for each training vector and target output pair(s:t)
- d. The input layer containing the input units is applied with identity activation function $x_i=s_i$
- e. Calculate the net input to output of the network

$$Y_{in} = b + \sum x_i w_i \quad (1)$$

Where $i=1$ to n and n is the number of input neurons at input layer.

- f. Apply the activation function

$$Y=f(y_{in}) = 1 \text{ if } y_{in} > \theta \quad (2)$$

$$Y=f(y_{in}) = 0 \text{ if } y_{in} = \theta \quad (3)$$

$$Y=f(y_{in}) = -1 \text{ if } y_{in} < \theta \quad (4)$$

- g. Weights and bias are adjusted by comparing the value of actual output against the desired output.

If $t=y$, then

$$w_i(\text{new})=w_i(\text{old}) \quad (5)$$

$$b_i(\text{new})=b_i(\text{old}) \quad (6)$$

else if $t \neq y$, then

$$w_i(\text{new})=w_i(\text{old})+ \alpha t x_i \quad (7)$$

$$b_i(\text{new})=b_i(\text{old})+ \alpha t \quad (8)$$

- h. Train the network until there is no weight change. This is the stopping condition for the network. If it is not met then start again from step number a.

To improve efficiency, the learning algorithm for the perceptron can be improved in several ways, but the algorithm lacks usefulness as long as it is only possible to classify linear separable patterns.

B. The Multilayer Perceptron (MLP) Or Multilayer Feedforward Network–

The MLP model gives a perceptron structure for representing more than two classes, and also defines a learning rule for this kind of network. The MLP is divided into three layers: input layer, hidden layer and output layer, where each layer in this order gives the input to the next. The extra layers, gives the structure, needed to recognise non-linearly separable classes.

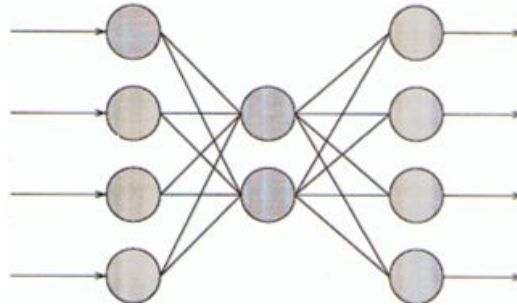


Figure 1. The Multi Layer Perceptron

Algorithm

- Initialize the weights, biases and learning rate suitably.
- Check for stopping condition; if it is false, perform steps c-g.
- Perform steps d-f for each bipolar or binary training vector pair $s : t$.
- Set activation (identity) of each input unit $i = 1$ to n :

$$x_i = s_i \quad (9)$$

- Calculate output response of each output unit $j = 1$ to m :

First, the net input is calculated:

$$Y_{inj} = b_j + \sum_{i=1}^n x_i w_{ij} \quad (10)$$

Then activations are applied over the net input to calculate the output response:

$$y_i = f(y_{inj}) = 1 \text{ if } y_{inj} > \theta \quad (11)$$

$$y_i = f(y_{inj}) = 0 \text{ if } -\theta \leq y_{inj} \leq \theta \quad (12)$$

$$y_i = f(y_{inj}) = -1 \text{ if } y_{inj} < -\theta \quad (13)$$

- Make adjustment in weights and bias for $j = 1$ to m and $i = 1$ to n .

If $t_j \neq y_i$, then

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha t_j x_i; \quad (14)$$

$$b_j(\text{new}) = b_j(\text{old}) + \alpha t_j; \quad (15)$$

else, we have

$$w_{ij}(\text{new}) = w_{ij}(\text{old}); \quad (16)$$

$$b_j(\text{new}) = b_j(\text{old}); \quad (17)$$

- Test for the stopping condition, i.e., if there is no change in weights then stop the training process, else start again from Step c.

Until the error function reaches a certain minimum, training continues on the training set. The network might not be able to correctly classify a pattern, if the minimum is set too high, and the network will have difficulties in classifying noisy patterns if the minimum is set too low.

III. EXPERIMENT AND RESULT

Single layer perceptron algorithm and multilayer perceptron algorithm is compared using the Netbeans IDE.

Single layer perceptron network

Enter value for x1,x2,x3,x4 and t

```
1 1 1 1 1
1 -1 -1 1 -1
-1 1 -1 -1 1
1 1 1 -1 -1
```

The final output

x1	x2	x3	x4	t	y	yin	dw1	dw2	dw3	dw4	db	w1	w2	w3	w4	b
1	1	1	1	1	0	-1	1	1	1	1	1	1	1	1	1	1
1	-1	-1	1	-1	1	1	-1	1	1	-1	-1	0	2	2	0	0
-1	1	-1	-1	1	0	-1	-1	1	-1	-1	1	-1	1	-1	-1	1
1	1	1	-1	-1	1	1	-1	-1	-1	1	-1	-2	0	-2	0	0

Multilayer perceptron network

Enter alpha

1

Enter theta

0

case1Enter the X1,X2,X3 & X4

1 1 1 -1

case2Enter the X1,X2,X3 & X4

1 -1 -1 1

case3Enter the X1,X2,X3 & X4

-1 1 -1 1

case4Enter the X1,X2,X3 & X4

-1 1 1 1

Enter the target for o/p value

-1 -1 -1 1

1 -1 -1 -1

1 -1 1 -1

-1 -1 -1 1

x1 x2 x3 x4 t yin y dw1 dw2 dw3 dw4 db w1 w2 w3 w4 w4 b

 1 1 1 -1 -1 0 0 -1 -1 -1 1 -1 -1 -1 -1 1 -1

 1 -1 -1 1 -1 1 1 -1 1 1 -1 -1 -2 0 0 0 -2

 -1 1 -1 1 -1 0 0 1 -1 1 -1 -1 -1 -1 1 -1 -3

 -1 1 1 1 1 -3 -1 -1 1 1 1 1 -2 0 2 0 -2

x1 x2 x3 x4 t yin y dw1 dw2 dw3 dw4 db w1 w2 w3 w4 w4 b

 1 1 1 -1 1 0 0 1 1 1 -1 1 1 1 1 -1 1

1 -1 -1 1 -1 -1 -1

x1 x2 x3 x4 t yin y dw1 dw2 dw3 dw4 db w1 w2 w3 w4 w4 b

 1 1 1 -1 1 0 0 1 1 1 -1 1 1 1 1 -1 1

1 -1 -1 1 -1 -1 -1

x1 x2 x3 x4 t yin y dw1 dw2 dw3 dw4 db w1 w2 w3 w4 w4 b

 1 1 1 -1 -1 0 0 -1 -1 -1 1 -1 -1 -1 -1 1 -1

1 -1 -1 1 -1 1 1 -1 1 1 -1 -1 -2 0 0 0 -2

-1 1 -1 1 -1 0 0 1 -1 1 -1 -1 -1 -1 1 -1 -3

-1 1 1 1 1 -3 -1 -1 1 1 1 1 -2 0 2 0 -2

IV. CONCLUSION

The MLP can be compared to the single layer perceptron by reviewing the patternclassification problem. The SLP can perform only simple binary operations. We can construct the XOR when advancing to using several unit layers. Even though we here the MLP is a much more convenient classification network, the MLP network is not guaranteed to find convergence. The MLP risks ending up in a situation where it is impossible for it to learn to produce the right output. This state of a MLP is called a local minimum.

REFERENCES

- [1] Bumptrees for Efficient Function, Constraint, and Classification by Stephen M. Omohundro, International Computer Science Institute
- [2] <http://nips.djvuzone.org/djvu/nips03/0693.djvu>
- [3] GA-RBF A Self-Optimising RBF Network by Ben Burdsall and Christophe Giraud-Carrier
- [4] <http://citeseer.nj.nec.com/71534.html>
- [5] Improving Classification Performance in the Bumptree Network by optimising topology with a Genetic Algorithm by Bryn V Williams, Richard T. J. Bostock, David Bounds, Alan Harget
- [6] <http://citeseer.nj.nec.com/williams94improving.html>

-
- [7] Evolving Fuzzy prototypes for efficient Data Clustering by Ben Burdsall and Christophe Giraud-Carrier
 - [8] <http://citeseer.nj.nec.com/burdsall97evolving.html>
 - [9] Neural Networks by Christos Stergiou and Dimitrios Siganos
 - [10] http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html
 - [11] Kohonen self-organising networks with 'conscience' by Dr. M. Turhan Taner, Rock Solid Images
 - [12] <http://www.rocksolidimages.com/pdf/kohonen.pdf>