A Review on WHAT NEURAL NETWORK IS and its Applications

Authors Name: Modi Parth Daxesh Department of **Electronics and Communication Engineering, Vishwakarma Government Engineering College,** Ahmedabad, Gujarat, India 380015.

Abstract: Have you ever been wondered how human brain recognize things and makes the decisions depending on the things it sees. Obviously there are some processing elements which makes decisions and they are nothing but Neurons. Human brain consist of millions of neurons and they are interconnected with each other. This papers reviews what neural network is and how it works and its applications in implementing Boolean functions and also image detection and classification.

Keywords: Neural Networks, Training a neural nets, Sigmoid function.

1. INTRODUCTION

Firstly motivation of learning Neural Networks is as follows: Have you ever wondered how our human brain recognize things around and make decisions? For instance we see a cat and immediately makes a decision that it's a cat. This decision making happens due to Neurons present in our brain and their number is millions in human brain! These are the processing elements which process the information which is seen by our eyes and makes the decision after processing. You can compare this with the millions of transistors which are the processing elements in a microprocessor!

This paper reviews what neural network is and most importantly how it works (from scratch) and idea is proposed to implement VLSI circuits using Neural Networks (NN).

2. What is Neural Network?

Obviously we can't make neurons which are inside brain. But the first artificial neuron made is known as PERCEPTRON. Many different neuron models came after that but the most widely used till date is Sigmoid Neuron. Figure2.1 show the Perceptron. It has one or more inputs and single output and inputs are weighted and the weighted sum compared with threshold decide the output as 1 or 0.Thus we can say it is a decision making model just like our brain but depending on inputs and weights unlike biological neurons. The decision making process is mathematically shown as follows:

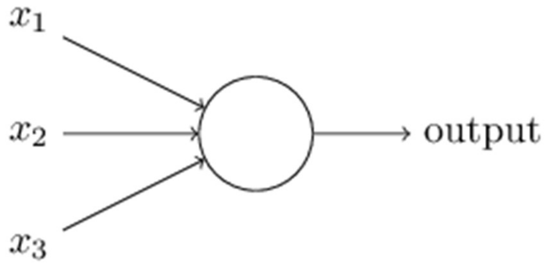


Figure 2.1 The Perceptron

$$\text{Output} = \begin{cases} 0 \text{ if } \sum_j w_j x_j \leq \text{threshold} & \dots(1) \\ 1 \text{ if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

3. How Neural Networks works?

Importance of Weights: Suppose that you love watching movies and you have to take decision of going to it or not by weighing the following inputs (factors).let say2 inputs are x1, x2.

Let x1= Is the price of ticket too expensive?

x2= Is the weather bad?

And 2 weights associated with them are w1, w2. We will make decision depending on the values of weight or in other words we can say that if value if w1 is higher than w2 then we will decide in favor of input x1 and vice versa. For example,

x1=1 is ticket is expensive and x1 = 0 means ticket is not expensive

x2=1 means weather is good and x2 = 0 means weather is bad case a) If you are a diehard fan of a movie you will definitely go for it irrespective of weather conditions and so in this case value if w2 = 3 (say) > w1=1 (say)

case b) If your mood to watch a movie depends on weather conditions then w2 > w1.

Choosing larger values of weights of particular condition indicates that it matters more to you than the other one.

Choosing Threshold: Choosing threshold appropriately will make the decision making model more accurate. Finally choosing threshold as 6 (just any arbitrary value but make sure that it gives correct decision making) indicates with above mentioned weights will make a correct decision making model. You can verify the correctness using output equation.

The value of weights can also be negative.

4. Multilayer Neural Networks

The perceptron just don't have an input layer and output layer. In fact there are number of layers between them which are called hidden layers and they are used to increase the accuracy of the system. More number of hidden layers means higher degree of accuracy. Each layer of the model works independently and compute its result and gives its output to the next layer. More number of layers create deep network and the network results is termed as Deep learning.

The multilayer perceptron is shown in the figure 4.1.

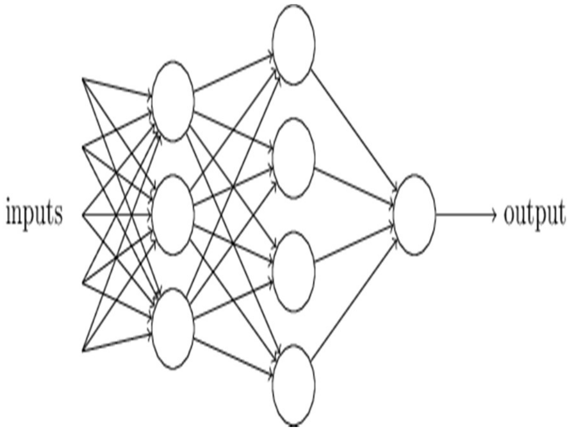


Figure 4.1 Multilayer Perceptron

This perceptron can be used to make more complex decisions just like our mind. The first layer perceptron makes decision simply by weighing inputs. This output is forwarded to second layer and it makes more complex decision and so on. Thus increasing the number of hidden layers increase the efficiency of the perceptron and helps in taking sophisticated decisions. Figure 4.1 also shows architecture of neural networks.

5. How to Train a Neural Network?

You may get surprised but you can adjust your weights and threshold (bias) at your own in order to train the model as per your need to make it more accurate. Revisiting output equation we can simplify it by writing $\sum_j$ as just "w.x" (where w and x are weight vectors and input vectors) and move threshold on other side and replacing it by 'b', where b = -threshold. The modified output equation is as follows

$$\text{Output} = 0 \text{ if } w.x + b \leq 0 \qquad \dots (2)$$

$$= 1 \text{ if } w.x + b < 0$$

Here b is termed as bias of the perceptron.

Thus adjusting weights and bias we can train our model for numerous application like recognizing handwritten digits , object detections and many more and it is obvious to say that the number of applications are not just limited to this. Neural nets have applications in every field!

6. Implementing Boolean functions using Neural Networks

Consider the perceptron shown in figure 6.1. The two inputs are termed as x1 and x2 and the corresponding weights, w1, w2 = -2, are written on top of the line. And the number written inside the last layer is nothing but bias, b=3.

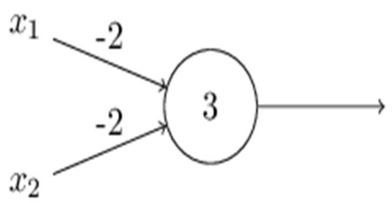


Figure 6.1 NAND gate using NN

When input is x1 = 0 and x2 = 0, output is (-2)(0) + (-2)(0) +3 = 3 which is greater than 0, so output will be 1. When input is x1 = 1 and x2 =1, output is (-2)(1) + (-2)(1) +3 = -1 which is negative so output is 0.

Similarly checking for x1x2=01 and x1x2= 10 we get output as 1. This clearly states that the figure 6.1 implements the NAND gate. This is how we can implement Boolean functions using NN.

Since NAND gate is a universal gate we can implement any Boolean function with it and thus we can realize ant Boolean functions easily.

Actually we devise algorithms which are used to train the neural networks. Also this process of learning is automatic as per the learning algorithms and there is no human intervention.

7. Thinking ahead of perceptron (Sigmoid Neuron)

How learning works? As we know that changing weights and bias train the network so we can say that a small change in weight or bias can be used to train a network and we can get a desired output. But the problem is not that. Sometimes what happens is, a small change in weights or bias can change the output of the perceptron completely from the desired one and so we need a new type of artificial neuron namely Sigmoid neuron[5].

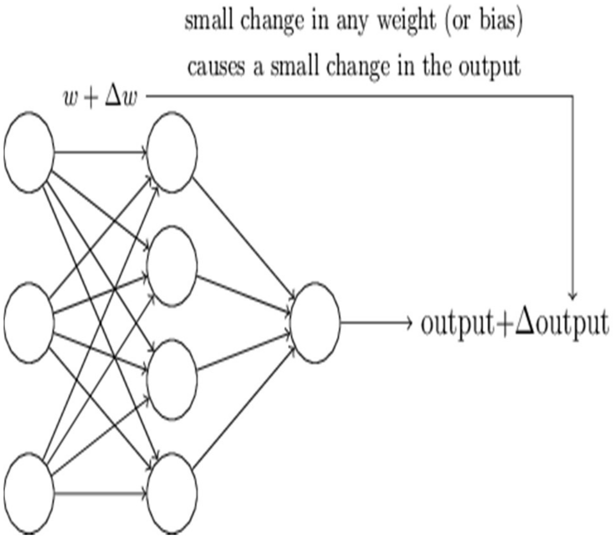

small change in any weight (or bias)
causes a small change in the output
$w + \Delta w$
output+Δoutput

Figure 7.1 How Neural Networks are trained?

Sigmoid neurons are similar to that of Perceptrons but are such that small changes in weights and bias causes only a small change in the final output and thus it overcomes the drawbacks of Perceptrons. Architecture of sigmoid is same as shown in figure 2.1, but here inputs can take any values between 0 and 1 unlike Perceptrons where value was only 0 or 1. Here also we have bias and weights.

The sigmoid function is given by:

$$\sigma(z) \equiv 1/1+e^{-z}$$

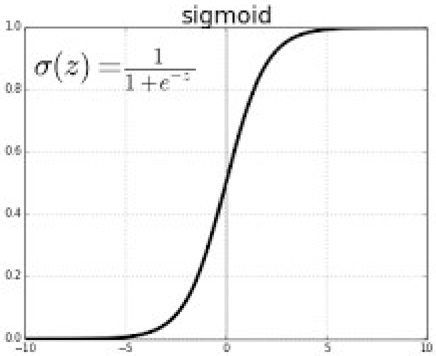


Figure 7.2 Sigmoid function

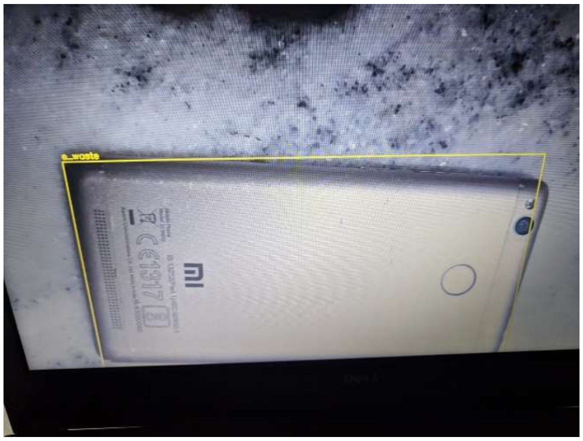
The output of the sigmoid neuron is as follows:

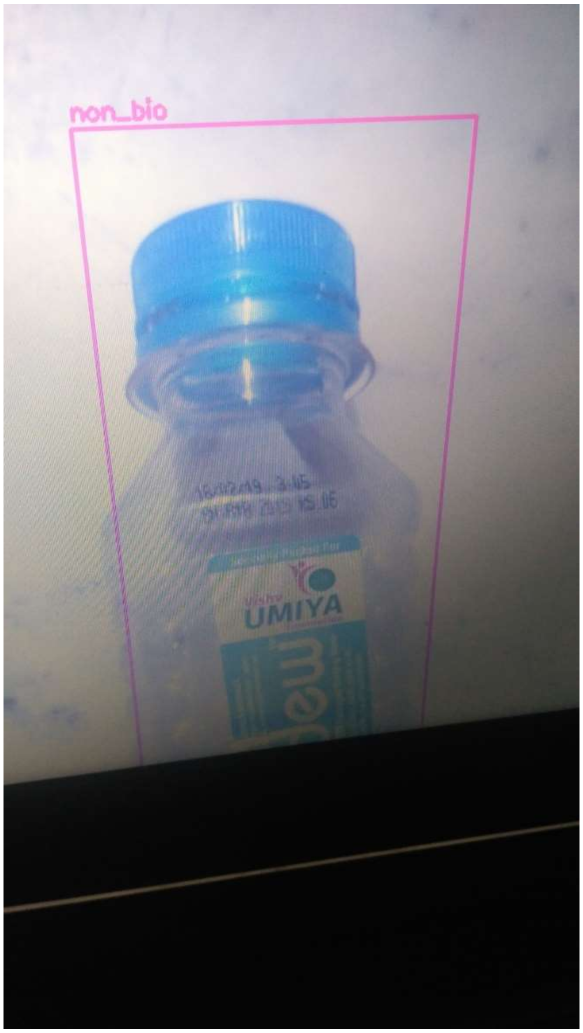
$\text{Output}=1/1+\text{e}^{(-\sum jwjxj-b)}$

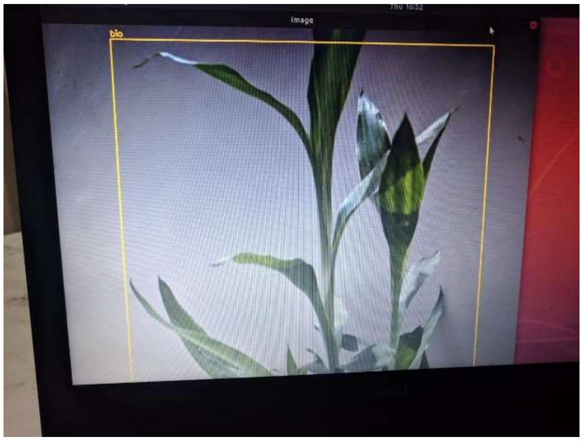
These functions are called activation functions since they activate the neuron of next layer depending on the inputs from the previous layer. Different activation functions are further found in [1].

8. My Project

Below images shows the detection and classification of objects using neural networks. The project is intended to classify dry and wet waste i.e. waste segregation using just a camera (Webcam) to see images and NN model to make decision.The code environment used is Python [6].







9. Conclusion

The term biological neuron which is present inside our human brain and the corresponding artificial neurons which are used for processing and decision making were discussed. The neural network architecture along with the significance of weights and bias were discussed and how weights and bias are useful in training the neural network model. The project “waste segregation” is demonstrated in section 8. Again as stated earlier the domain of applications is not limited it is used in every branch of Engineering and solving real life problems.

10. References

[1] Neural networks by Simon Haykin

[2] Deep learning course by Prof Andrew ng

[3] Neural networks by D.Kreisel

[4] Deep learning by Prof. Mitesh Khapra (IIT Madras)

[5] http://neuralnetworksanddeeplearning.com/

[6]https://www.youtube.com/watch?v=hysqoAeIV5A