Design and Performance Analysis of and Gate using Synaptic Inputs for Neural Network Application

S. Soundarya
UG Student
Department of Electronic and Communication Engineering
Saveetha School of Engineering, Chennai

Vamshi. G
UG Student
Department of Electronic and Communication Engineering
Saveetha School of Engineering, Chennai

Soundarajan. M
UG Student
Department of Electronic and Communication Engineering
Saveetha School of Engineering, Chennai

Ram Kumar. A
Assistant Professor
Department of Electronic and Communication Engineering
Saveetha School of Engineering, Chennai

Abstract

Logic gates are one of the main constituents to design and integrate a chip. With advent of vision and method in neural network, the complexity can be clear alongside this knowledge to a remarkable extent. The intention of this paper is to focus on how to design a perceptron neuron to purpose as an AND gate and to examine its performance. The neural network toolbox in MATLAB is delineated for training, validating and assessing the neural network. The performance of the AND gate using neural network is proved by comparing its results with the results of the generated code for AND gate in MATLAB.

Keywords: neural network, AND gate, MATLAB, learning algorithm

I. INTRODUCTION

The main computing elements of a human brain are neurons. A human brain approximately has 10^{12} cells with 10^{15} interconnections for the processing of information. In general human nervous system, data is transmitted between different neurons is done by using coded pulse streams. The human brain is able to perform 3.6x10^{15} operations based on synaptic input per second, and the efficiency of computation is almost 3x10^{14} synaptic operations per joule. The normal human brain is very much able to perform highly complex computation in real time. If we compare the computational efficiency of human brain with a general microprocessor human brain is at least seven times of magnitude more efficient than the general microprocessor.

Generally logic gates procedure the signals that embody TRUE or FALSE and produces output either TRUE or FALSE according to their design. The basic logic gates are AND, NAND, OR and NOR gates. Fig 1.1 shows the symbol for basic AND gate.

![Fig. 1.1: Symbolic representation of AND gate](image)

An arrangement of one input layer of neuron feeding forward to one output layer of neuron is called as a perceptron. The perceptron is an algorithm for supervised association of an input into one of the countless probable non-binary outputs. It is a kind of linear classifier, i.e. an association algorithm that makes it forecasts established in a linear predictor purpose joining a set of weights alongside the feature vector. In this paper, the mathematical background of neural network will be studied firstly. Then, Neural Web Toolbox in MATLAB will be utilized to develop the neural web for assisting the analyzing of an AND gate.

II. BASIC CONCEPTS

A. Neuron:
The human brain is a collection of concerning 10 billion interconnected neurons. Every single neuron is a cell that uses biochemical replies to accord, procedure and send information. It is an electrically excitable cell that procedures and transmits data across mechanical and chemical signals. These signals amid neurons transpire via synapses, enumerated connections
alongside supplementary cells. Neurons can link to every single supplementary to form neural networks. A neuron’s dendritic tree is related to a thousand bordering neurons. As one of those neuron fires, an affirmative or negative price is consented by one of the dendrites.

B. **Neural Network:**
Neural network is a computer arrangement of biological neurons, composed of nonlinear computational agents working in parallel. A neuron’s dendritic tree is related to a thousand bordering neurons. As one of those neurons fire, an affirmative or negative price is consented by one of the dendrites. The strengths of all the consented prices are added jointly across the procedures of spatial and temporal summation. Spatial summation occurs after countless frail signals are modified into a solitary colossal one, as temporal summation converts a quick sequence of frail pulses from one basis into one colossal signal. The aggregate input is next bypassed to the soma. The soma and the encircled nucleus don't frolic a momentous act in the processing of incoming and outgoing data.

Each connection is allocated a comparative weight, additionally shouted connection strength, or synaptic strength. The output at every single node depends on the threshold, additionally shouted bias or offset, enumerated and a transfer (activation) function.

C. **Feed Forward Neural Network:**
A feed forward neural network is an artificially made neural network whereas connections amid the constituents do not form a managed cycle. This is disparate from recurrent neural networks. The feed forward neural network was the early and simplest kind of artificially made neural network devised. In this network, the data moves in merely one association, onward, from the input nodes, across the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

D. **Single Layer Perceptron:**
The simplest kind of neural network is a single-layer perceptron network, which consists of a solitary layer of output nodes; the inputs are fed undeviatingly to the outputs via a sequence of weights. In this method it can be believed the simplest kind of feed-forward network. The sum of the produce of the weights and the inputs is computed at every single node and if the worth is above a little threshold (typically 1) then the neuron fires and seizes the activated, or else it seizes the deactivated worth (typically -0).

### III. DESIGN METHODOLOGY

![Flow chart of system design](image)

From Fig 3.1 shows the flow chart of the system design. Firstly the inputs and the required outputs (targets) are given to the network and a new perceptron is created with the required number of hidden layers according to our application because the increase in the number of hidden layers will increase the accuracy of the network. Then the network will be trained with the given set of inputs and target values and then it is validated and checked. Here for the implementation of AND gate we used 70% of the given data for training purpose, 15% for validating purpose and another 15% for checking purpose. This can be done by changing the percentage of the data usage for different purposes, finally performance and regression curves are plotted and analyzed.

### IV. LEARNING PERCEPTRON ALGORITHM

A network can learn in two methods one is supervised learning and the other is adaptive learning. Perceptron is the network that is capable of learning and is only applicable for linearly separable curves such as AND gate. Learning perceptron algorithm
is one of the supervised learning algorithms. It proceeds with a network and changes the weights to each input according to the following three cases

1) Case 1:
When the obtained output is same as the targeted output, then the error will be zero and hence the weights are not adjusted.

2) Case 2:
When the obtained output is 0 but the required output is 1 then the given input will be added to the weights in order to get the accurate value.

3) Case 3:
When the obtained output is 1 but the required output is 0 then the given input will be subtracted from the weights in order to get the accurate value.

A. A General Formula for All the Three Cases

\[ \Delta W = (t-a) P \]  \hspace{1cm} \text{(1)}

And the new weight is

\[ W_{\text{new}} = W_{\text{old}} + \Delta W \]  \hspace{1cm} \text{(2)}

Where

- \( \Delta W \) = change in weight
- \( t \) = target value
- \( a \) = obtained output
- \( p \) = input value
- \( W_{\text{new}} \) = new weight
- \( W_{\text{old}} \) = old weight

V. IMPLEMENTATION TECHNIQUES

In this paper, we are going to discuss about two methods of creating a perceptron which will function as AND gate by using LEARNINGP algorithm. One is by using MATLAB coding and the other is by using neural network tool in MATLAB®. In both the techniques the same design procedure is used and the outputs obtained in both the techniques are depicted below.

A. Training With Known Set of Values:
- First the inputs and the corresponding targets are given to the network.
- A new perceptron is created for the given inputs and targets.
- The network is trained with the 70% of the give set of values, and the rest for validating and testing of the network.
- Corresponding fit, regression and performance graphs are obtained and plotted.

B. Coding:
- The inputs and the corresponding targets are provided using variable.
- Using network functions available in MatLAB, we assign these variables to network.
- The number of iterations for training can be based on the desired value.
- The corresponding output graphs can be plotted using MATLAB.

VI. RESULTS

Fig. 6.1: Coding technique output for and gate
In Fig 6.1 coding implementation method the network is trained with bias learning rules which is similar to learning perceptron algorithm, the given data set is completely used for 11 times to update its weights and to produce results accurately. The performance of the network is analyzed using mean absolute error (MAE) which is one of the ways to compare the predicted output and the obtained output of a neural network.

The step size taken for plotting the curve is 0.1 in both the axis and the peak in the plot depicts the output of AND gate, as the plotting curve used is mesh type which gives the three dimensional view of the output. This can be done by using ones (m, n) in MatLAB which plots m*n matrix of m rows and n columns of ones.

From the Fig 6.2 we can observe that the mean absolute error increases with the increase in the number of epochs till certain limit and after that it again decreases. Here in this case the mean absolute error increased till 3 epochs and then decreased. The best training state performance is obtained at ‘0’ epoch.

From Fig 6.3 the neural network tool box which is used to train the values using MATLAB is shown.
Fig 6.4 shows the performance curve of the AND gate obtained by training the network with known set of values. There are four different colored lines in the epoch vs mean squared error (MSE) plot which gives the average of the squares of the errors. Error is nothing but the difference between the predicted value and the obtained value of the output, in which the dotted line denotes the best performance, blue curve denotes training state performance, green curve denotes validation checks and finally the red one denotes the testing of a particular network. The best validation performance is obtained at epoch 2 in this case.

![Performance plot using NN tool box](image)

Fig 6.5 shows the fit curve of an AND gate when a single input is given in the training method. The inputs and the corresponding target values are given randomly to the network between 0 and 1. The new network is created for the given set of values in which 70% of data set is used for training the network, 15% for validation checks and the rest of the 15% for testing the network. The fit curve is nothing but which fits the given data into the curve. In the network it is trained in such a way that in the given data set from 0 to 1 it should show an output of 1 if its input is greater than or equals to 0.7 or else it should show an output of 0. The different values taken for different purposes are shown in different colors for the sake of convenience.

![Fit plot using NN tool box](image)

The corresponding performance plot is shown in the Fig 6.6 in which the dotted line shows the best performance of the network, blue curve denotes training state performance, green curve denotes validation checks and finally the red one denotes the testing of a particular network. The best validation performance is obtained at epoch 33 in this case which shows that the given data set is completely used for 33 times to update its weights to produce the accurate result.

![Performance plot using NN tool box](image)

**VII. CONCLUSION**

These neural networks find wide range of applications in this fast generation because it is something which is arranged and designed to analyze and take decisions like our human brain which learns on its own from its experience. The same technique is used in this algorithm and is applied for the network to purpose as an AND gate. Further this can be used and implemented for various applications like image compression, character recognition, classifying a given task and a lot more.

**VIII. ADVANTAGES**

- The network can learn from its own experience and can take decision accordingly.
No human involvement is required once if the circuit is trained properly.
With the increase in the number of iterations the accuracy of the network can be increased.
Cost performance and flexibility.

IX. APPLICATIONS

The neural networks find a wide range of applications in different areas which include vehicle controlling, decision making, process controlling, game controlling (which includes chess, backgammon, racing etc.), pattern recognition, face recognition, object recognition, medical diagnosis, and business applications like share market prediction data mining.

REFERENCE