Group42 – Assigmnent3 – ML

G. Surya Srikanth – S20170010049

B. Kalyan – S20170010019

Outline Understanding:

* To train a 3 layer multi-layer perceptron with changing hidden layers.

Sigmoid function = 1/1+(e^-x)

* Sigmoid function should be used as activation function in the neurons.
* Given the learning rate is 0.001.

Method:

* Take the data from the file and add randomize it.
* Normalize it by adding a bias term of 1.
* Randomly generate weights initially.
* Values of hidden layer are signum of sum of the product of each node to the weight of the link from its node.

Hidden neuron Val = sig(sum(W.X))

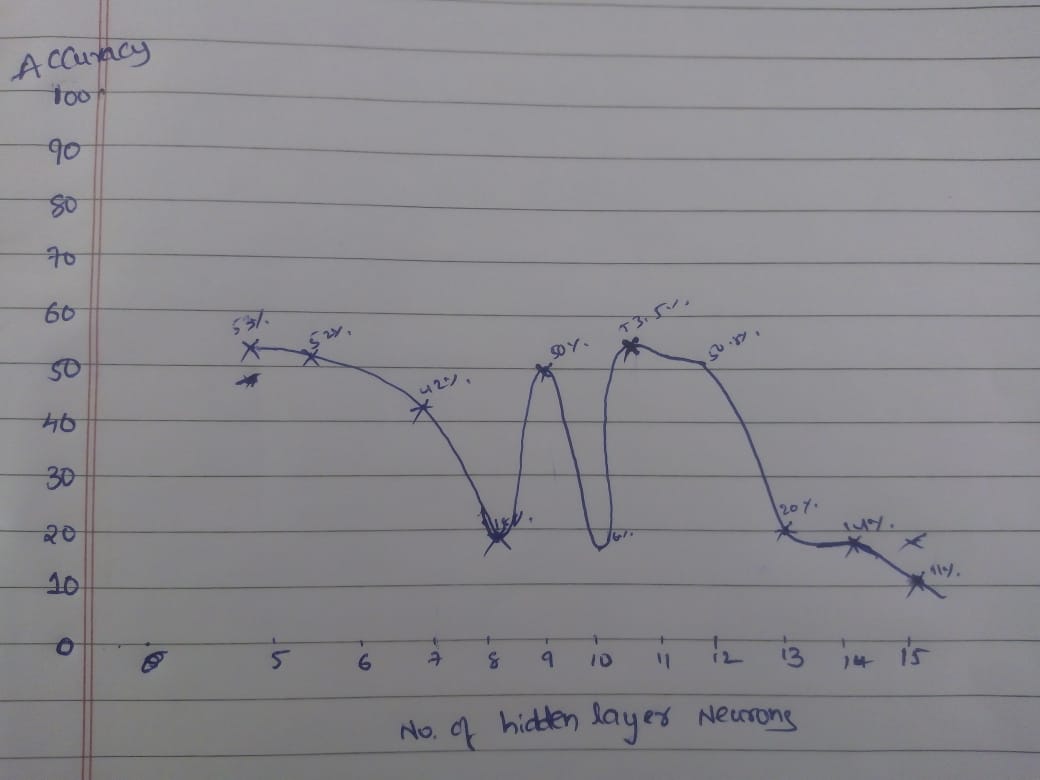
W = weights, X = input layer

* Similarly, forward for the output layer.
* To update the values, use back propagation.
* Find the loss function (here both SSD and cross-entropy)
* Use gradient descent to minimize the loss.
* One can update w2 using this method.
* To update w1 use chain rule to find gradient using loss function for w1.
* When w1 changes the values in hidden layer changes and slowly approach our desired values.
* We can also observe that the loss decreases and classification error also decreases

Observations and results:

## **Graph of Accuracy vs no of neurons in hidden layer using Sum of squared deviation loss**

## **Randomized every time**



## **Graph of Accuracy vs no of neurons in hidden layer using cross-entropy loss**

## **Randomized every time**

