Deep Neural Networks

Group-3
Group Members:Shreyas Patel(1401025),Akhil Vavadia(1401095),Amee Bhuva(1401009),Malav Vora(1401062),Manini Patel(164404)
School of Engineering And Applied Science,Ahmedabad University

***Abstract*—The artificial neural network(ANN) is a biologically-inspired methodology to conduct machine learning, intended to mimic human brain.An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process.**

Keywords:Neural Network,Deep Neural Network,Back propagation,Activation Function.

I. Introduction

The First developed ANN model was Perceptrons which take binary inputs and produces a single binary output based on weighted sum and threshold value.Then more general version of ANN came into picture called single layer network,in which there is a neuron for each input and which is called input layer and then wighted sum of input layer neurons is calculated at another layer called hidden layer.Then the last layer is called output layer which does thresholding using activation functions like Rectified linear unit(ReLu) or sigmoid.The networks in which there are more than one hidden layer are called deep neural networks.

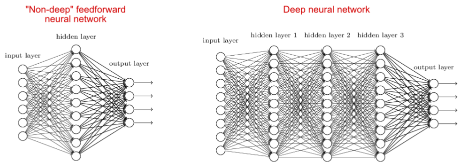


Fig. 1: Single layer vs deep neural network

II. Basic Process

First we create a neural network for that we need to decide required number of hidden layers and nodes per layer.Generally these is selected by trial and error.we should optimally choose number of nodes so that network does not give high error(when too few nodes) and over-fitting(too many nodes).For wights,they are selected at random at start Then, in a typical ”feed forward” neural network, we pass data straight through the network and compare the obtain output to desired output from sample data. From here, we adjust the weights to get output to match your desired output.Our data goes from input, to the layers, in order, then to output. When we go backwards and begin adjusting weights to minimize loss/cost, this is called back propagation.After some iterations(Epoch) when we found there is very less changes in cost function which means our prediction closest to the desired value.

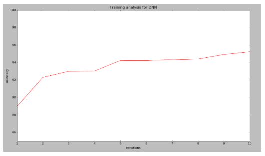


Fig. 2: Accuracy vs Epoch

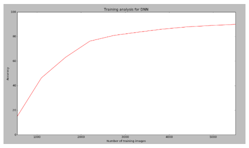
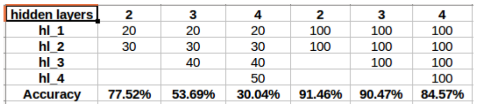


Fig. 3: Accuracy vs No. of training images

III. Implementation

A. Implementation on MNIST dataset



hidden layers	2	3	4	2	3	4
hl_1	20	20	20	100	100	100
hl_2	30	30	30	100	100	100
hl_3		40	40		100	100
hl_4			50			100
Accuracy	77.52%	53.69%	30.04%	91.46%	90.47%	84.57%

Fig. 4: Effect of Change in No. of hidden layers and nodes

Here our task is to recognize number from handwritten images of number. From Fig-2 we can say that as epoch(cycle of forward + backward pass) increases we get better accuracy for initial epochs,for epoch greater than 10 accuracy remains almost constant.Also from Fig-3 we can say that as we increase training data,accuracy increases but it remains constant after sufficient training images.From Fig-4 we can say that as number of hidden layer increases accuracy decreases and also with increase in no of nodes per layer accuracy increases but as we know increase in nodes per layer may result in overfitting so we should choose these parameters optimally.

B. Implementaion For Multiplication Tables

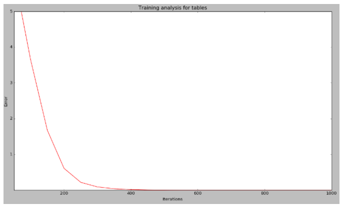


Fig. 5: Error vs No. of iteration

We tried to train our neural network using tables of 1 to 5 where there were two inputs and one output. Then on giving two random numbers as input, the neural network was able to give approximately the product of to numbers. The error decreased as number of iterations were increased for training data set(Fig-5).