Deep Learning

(Assignment 2)

Report

Name: **Qazi Danish Ayub**

Roll no. **MSDS20075**

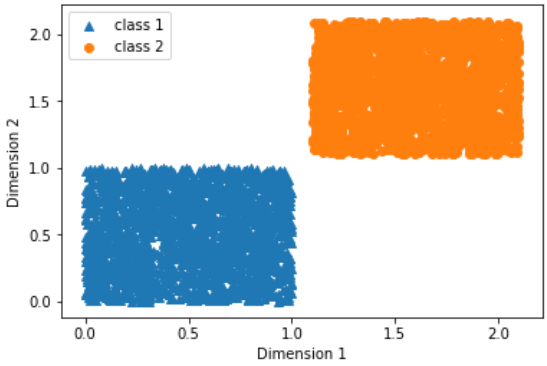
Department of Computer Science,

Information Technology University

Task # 1

Implementation of Neural Network with Sigmoid and cross entropy loss function.

Initially random dataset was generated of about 4000 data points, having two classes 2000 data points each. Figure below shows two classes.

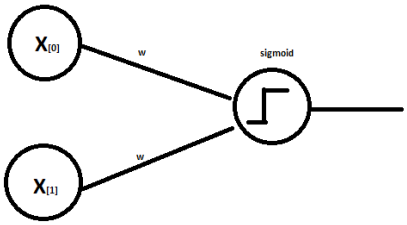


Data was shuffled and split into 3 parts: training, testing and validation.

- Training60%
- Testing 20 %
- Validation 20%

Validation data was used for cross validating and checking errors while training process.

Simple neural network architecture was built with 2 input nodes and 1 output nodes.



Weight vector of **2X1** created and initialized with random values. Bias is also added and initialized with random value.

This is a simple architecture with feed forward neural network. and Sigmoi d function is used.

Sigmoid:

$$S(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid Derivative:

$$S(x) = S(x)(1 - S(x))$$

For calculating the loss cross entropy function is used

Cross Entropy:

$$-1/N \sum_{n=1}^{N} [y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)$$

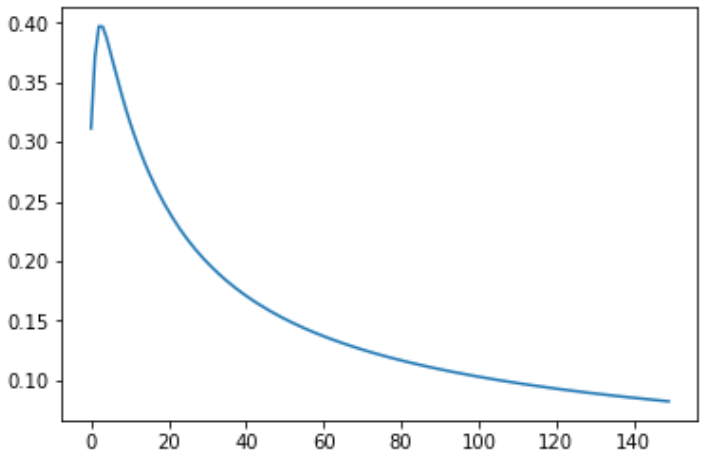
where y_n is actual while $\hat{y}_n$ is predicted value.

In backpropagation loss was calculated and weight vector was updated based on learning rate and loss value.

Parameters for training:

- Number of Epochs: 150
- Learning rate: 0.001

Following graph shows Loss curve generated by loss function while training.



Saving Model:

Model input size, output size, learned updated weights and biases was saved using pickle.

Accuracies:

Although the data is clearly linearly separable 100 % accuracy was achieved.

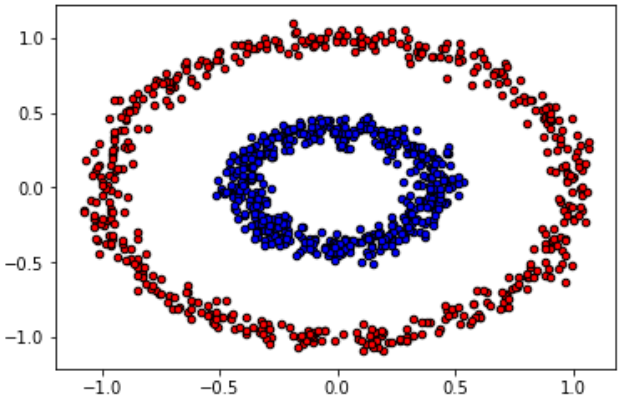
1. Training Data 100%
2. Testing Data 100%
3. Validation 100%

Task # 2

Implementation of Neural Network with a single hidden layer, multiple activation function and cross entropy loss function.

Dataset:

Random dataset of 1000 data points was generated, having two classes **Red** and **Blue**, Red label indicated by '**1**' and Blue label indicated with '**0**', as shown in figure below.



Dataset was shuffled and then split into 3 parts.

- Training | 40% | (600 X 2)
- Testing | 20 %| (200 X 2)
- Validation | 20 %| (200 X 2)

For training on data, a simple neural network was built with 1 hidden layer with 3 neurons.

Weight vector for layer1 was initialized of (input_size+1 X 3) dimension, and for layer 2.

Weight W2 is initialized as (hidden layer+1 X 2), **(+1)** added is for bias.

3 activation function was used.

- Sigmoid
- Tanh
- Relu

Sigmoid:

$$S(x) = \frac{1}{1 + e^{-x}}$$

Tanh:

$$T(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

ReLu:

$$f(x) = \max(0,\ x)$$

For calculating the loss cross entropy function is used

Cross Entropy:

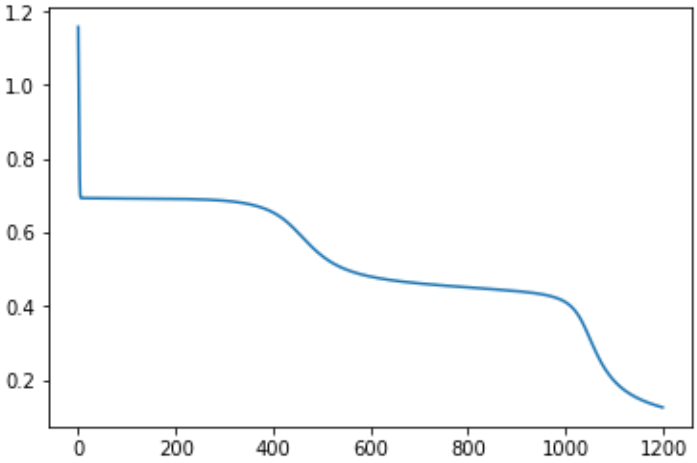
$$-1/N \sum_{n=1}^{N} [y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)$$

where y_n is actual while $\hat{y}_n$ is predicted value.

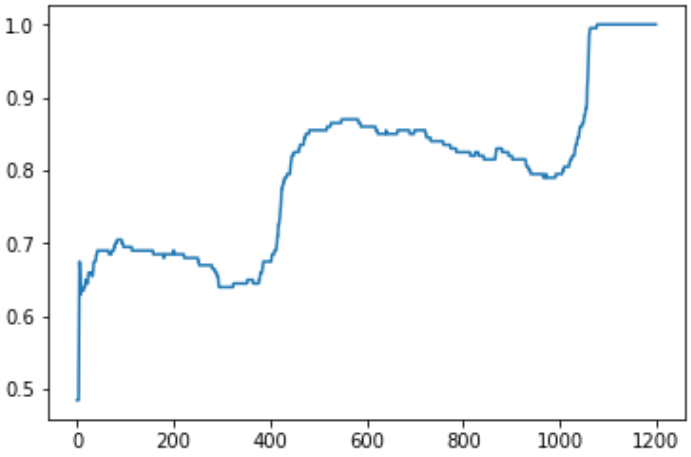
Parameters for training:

- Epochs = 1200
- Learning rate = 0.009
- Activation = Sigmoid

Loss Curve:



Accuracy Graph:



As the data is also clearly separable, we will be able to achieve 100% accuracy.

1. Training Data 100%
2. Testing Data 100%
3. Validation 100%

Task # 3

Implement a multi-layer Neural Network for multi-class classification.

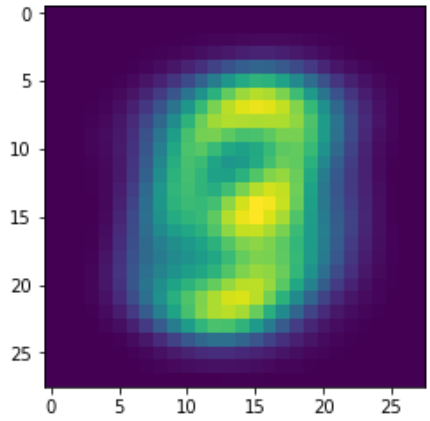
In this task we are required to build a neural network for multitask classification. We are given a dataset of handwritten digits from 0 -9 most known as MNIST.

Dataset contains 60000 samples for training and 10000 samples for testing. Each image is size of 28 X 28.

We first load dataset, and then flatten and reshape datasets from [60000, 28,28] to [60000, 784] to feed into neural network.

Mean:

Mean of all images is calculated.



Subtracted image from mean.