REPORT OF ASSIGNMENT-2

Name: Sai Santhosh Kota
130070042

a)**Feature Engineering:**

The parameters I have changed to acheive a better accuracy are:

1) Batch Size : On increasing the batch size though the computation time decreased , accuracy went down. For example from increasing batch size from 50 to 100 accuracy went down from 0.7 to 0.59. For final simulation I have used a batch size of 50

2) Layer size : Number of neurons in the hidden layer not made much difference in the accuracy. I have studied that the optimal number of neurons should be between input layer size and output layer size. So I have used a hidden layer of size 50, since our input layer size is 108 and output layer size is 1

3) Biases : I have initially implemented with no bias then the accuracy was not so good then I changed the biases to 0.5 which tend to be good

4)Learning rate : I have changed the learning rate from 1 to 0.2 which increased the accuracy

5) Activation Functions : I used logistic function for this case

I have basically used this as my reference: **https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/**

b) **Basic Blocks:**

Preprocessing data: I have used pandas and numpy to read the data and preprocess it. I have used pandas for one hot encoding the data.

Forward Pass: Intially I have taken the weights==>w1(108*50) in first layer and w2(50*1) in second layer using standard_normal function in numpy. Then I have taken each row of the input data i.e., each sample as the input and used the weights to calculate the output.

Calculating the loss: Now I found the error between obtained output and original output.(Normal Euclidean error function). I have calculated this error for batch_size of inputs and accumulated the error.

Backpropogation: Now use backpropagation to update the weights and repeat again till the training data is over.

c) **Comparison between classifiers:**

I have implemented three other classifiers:

a)**k-nearest neighbourhood claasifier:** I have taken k=3 and implemented it using scikit module. I have acheived a accuracy of 0.72 using this.

b) **3 layer neural network :** I have implemented this using keras modules. I have used relu and sigmoid for activation functions.I have used **binary cross entropy** for loss function and used **adam** optimiser. I have acheived a accuracy of 0.734

c) **Support vector classification :** I have implemented this using scikit module.This gave an accuracy of 0.713