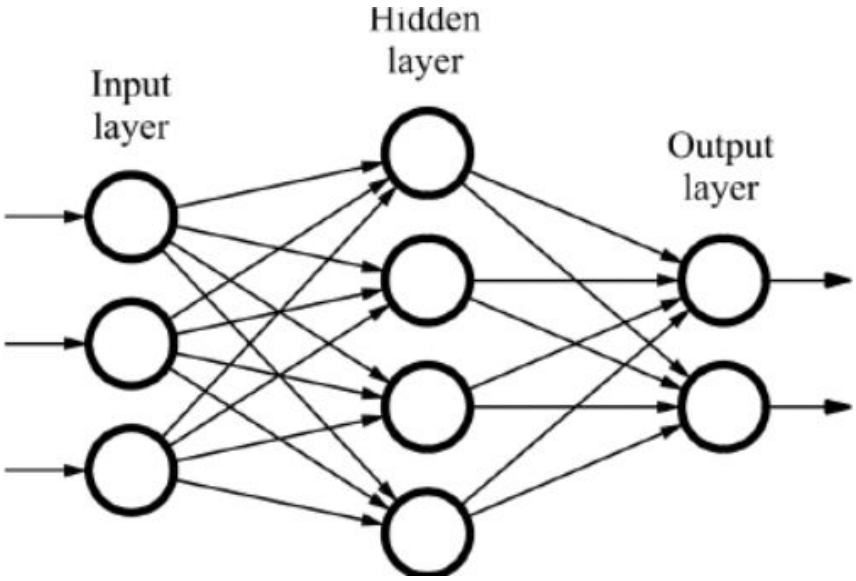
Machine Learning Assignment 2
Neural Network

Team Members→

1. Atharva Sune 2017A7PS0183H
2. Ameya Salankar 2017A7PS0182H
3. Aayush Laddha 2017A8PS0717H

Neural Networks

Neural Networks are a class of algorithms, modeled around the human brain, which are designed to find and interpret hidden patterns in data.



The above picture depicts a 2 layer neural network (because 2 weight matrices).
In the above neural network we have 2 weight matrices W^1 between input and hidden layer and W^2 between hidden and output layer.

Every layer (except the input)performs the following function:

Let the input to the layer be denoted by A, let the weight matrix be denoted by W and bias vector be denoted by B then the output of the layer is

$$Z = g((W.T * A) + B)$$

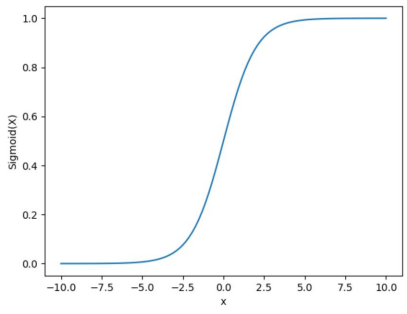
Where W.T = transpose of weight matrix, g is called activation function

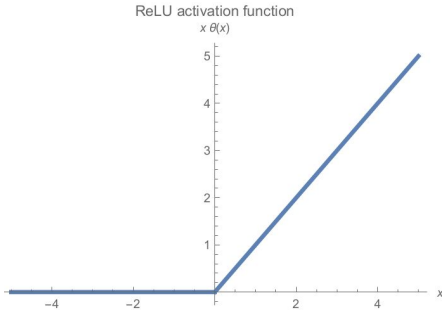
The given problem is a classification problem hence the activation function for the last layer is ***sigmoid.*** For the rest of the hidden layers there are various options available and in our implementation we have implemented ***Relu, LeakyRelu.***

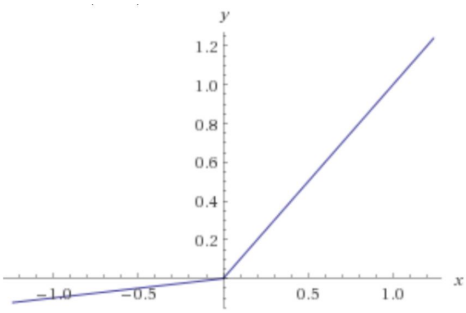
$$\sigma(x) = \frac{1}{1 + e^{(-x)}}$$

$$relu(x) = x \text{ if } x > 0 \text{ else } 0$$

$$leaky\ relu(x) = x \text{ if } x > 0 \text{ else } Cx, \text{ where c is a small constant, } c < 1$$







Model and Dataset

For the training the model we have implemented 2 methods, SGD (Stochastic gradient Descent) and Vanilla Gradient Descent. Also neural networks with various hidden layers have been trained. In the assignment we have experimented with different weight initializations as well (uniform and gaussian). It was observed that the accuracy when initialised with uniform was better than when initialized with gaussian.

The dataset had to be scaled first before applying the model. For scaling we have subtracted the mean of a feature from all the values and then divided the result by the standard deviation of the feature. This process is called standardization.

$Z = \frac{X - \mu}{\sigma}$, where Z = new value, X = original value, μ = mean and σ = std. deviation

Then an 80-20 split is performed on the dataset, to get the train and test split .

Accuracy Statistics

The accuracy statistics of some of the models that we are as follows →
********* Neural_Network_Relu_1_Hidden_layers_SGD_uniform **********
========== Train Statistics ===============
Accuracy:88.4417808219178 F-Score: 88.47139197267292

==========
========== Test Statistics ==========
Accuracy:86.98630136986301 F-Score: 86.98630136986301
==========

********* Neural_Network_Relu_3_Hidden_layers_SGD_uniform **********
========== Train Statistics ==========
Accuracy:88.6986301369863 F-Score: 88.85135135135135
==========
========== Test Statistics ==========
Accuracy:86.3013698630137 F-Score: 86.39455782312926
==========

********* Neural_Network_Relu_1_Hidden_layers_VGD_uniform **********
========== Train Statistics ==========
Accuracy:89.04109589041096 F-Score: 89.05982905982906
==========
========== Test Statistics ==========
Accuracy:87.67123287671232 F-Score: 87.67123287671232
==========

********* Neural_Network_Leaky_relu_1_Hidden_layers_SGD_uniform **********
========== Train Statistics ==========
Accuracy:88.4417808219178 F-Score: 88.47139197267292
==========
========== Test Statistics ==========
Accuracy:87.32876712328768 F-Score: 87.37201365187713
==========

********* Neural_Network_Leaky_relu_3_Hidden_layers_SGD_uniform **********
========== Train Statistics ==========
Accuracy:89.04109589041096 F-Score: 89.29765886287626
==========
========== Test Statistics ==========
Accuracy:87.32876712328768 F-Score: 87.62541806020067
==========

********* Neural_Network_Leaky_relu_1_Hidden_layers_VGD_uniform **********
========== Train Statistics ==========
Accuracy:89.46917808219177 F-Score: 89.49615713065755
==========
========== Test Statistics ==========
Accuracy:86.98630136986301 F-Score: 86.89655172413794
==========

==========

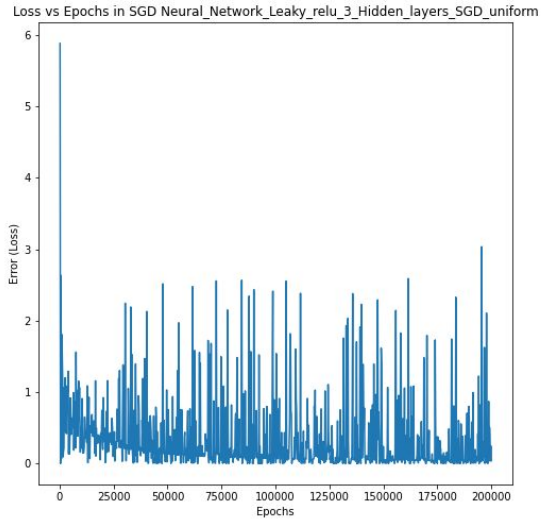
********* Neural_Network_Leaky_relu_3_Hidden_layers_VGD_uniform **********
========== Train Statistics ==========
Accuracy:89.29794520547945 F-Score: 89.28877463581834
==========
========== Test Statistics ==========
Accuracy:87.32876712328768 F-Score: 87.19723183391004
==========

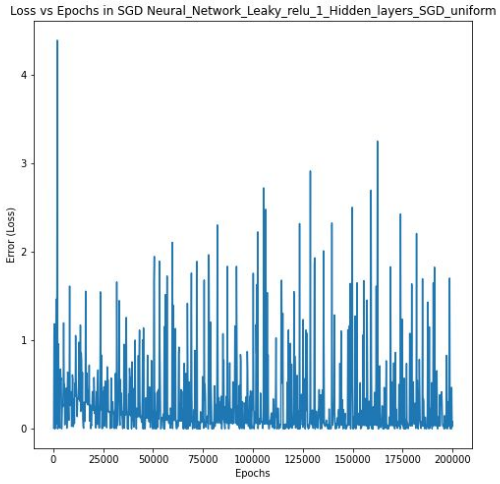
It can be observed from the statistics, that generally (in most cases) there is an increase in the accuracy as the number of layers increases. Also a slight increase is observed in the accuracy when VGD is performed instead od SGD . It can be attributed to the fact that in VGD, the model passes through the complete data unlike SGD.

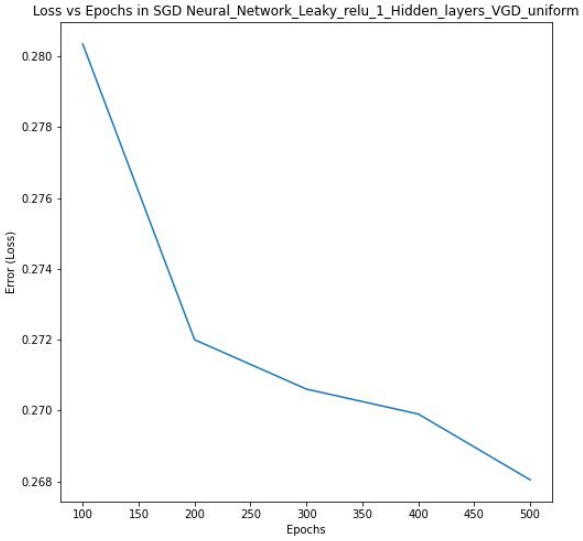
Note: → Accuracy Statistics for all models can be found in results.txt file

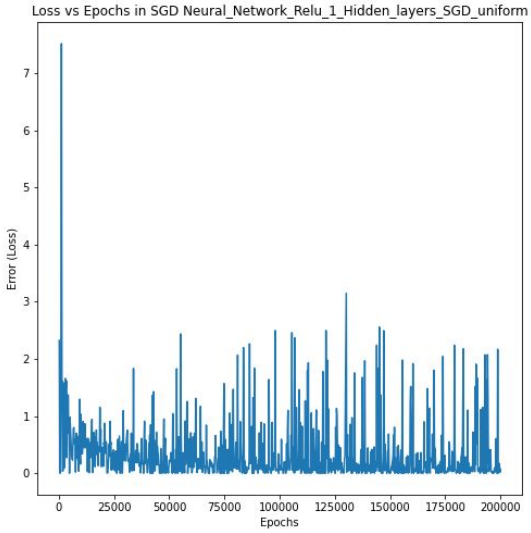
PLOTS

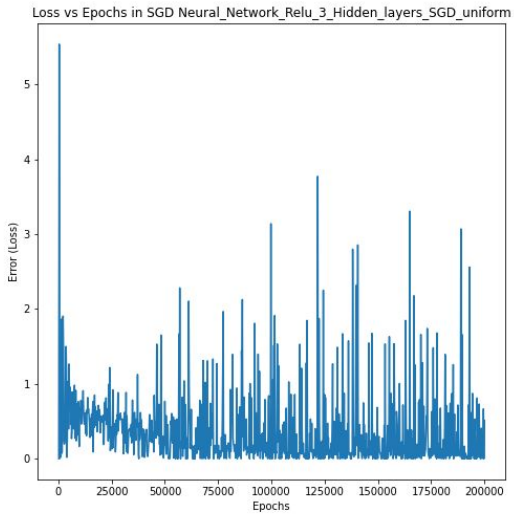
Plots for some of the models that we tested, all plots are available in the plots folder.

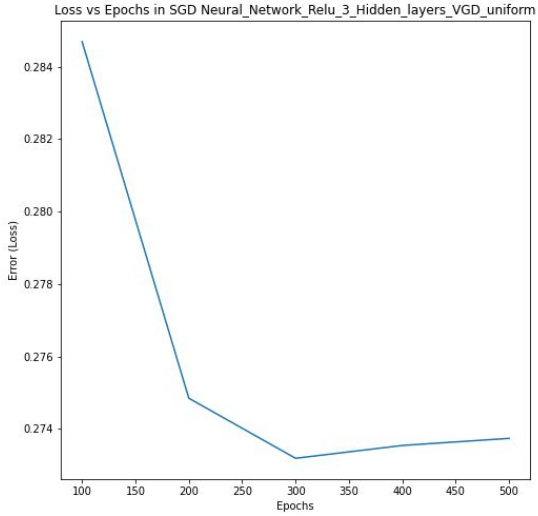












As expected VGD leads to a very smooth curve whereas SGD leads to a very spiked curve, but a definite pattern of decreasing loss is observed in SGD, and it can be seen that it arrives much faster, than in VGD.

Also the number of iterations in VGD are less because in each iteration the model passes through the complete data. So effective number of iterations in VGD = num_epochs x num_examples.