

ELL 409 Assignment 1

By Gauri Gupta

2017MT60207

Neural Networks: Back Propagation

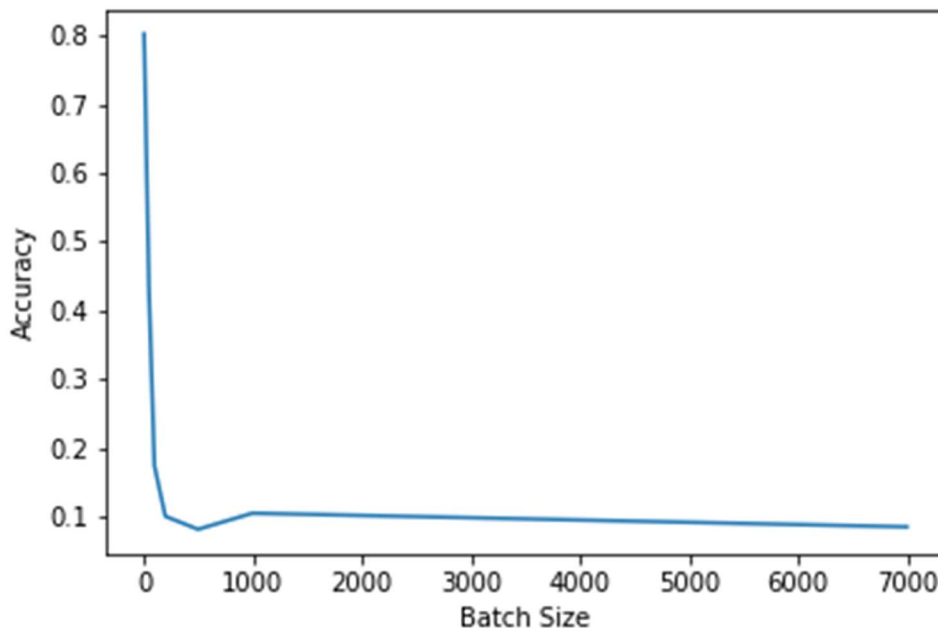
Dataset: We are given an MNIST dataset to train our model. The MNIST database is a large database of handwritten digits that is commonly used for training various image processing systems.

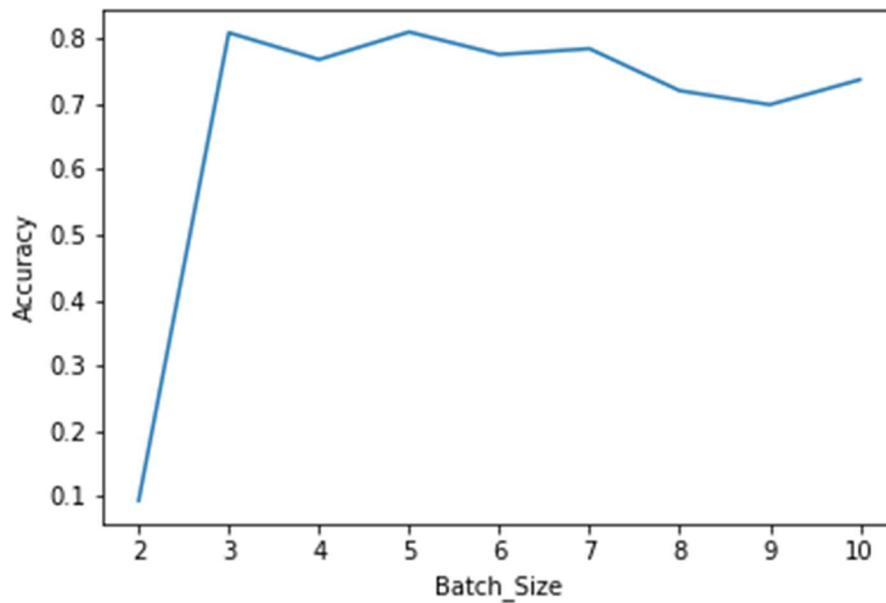
Aim: To implement a multiclass neural network classifier that classifies each image as corresponding digit.

Observations:

Studied various hyperparameters such as learning rate, batch size, activation function, number of hidden layers, number of neurons in each layer and their effects on the model performance.

1. Batch Size

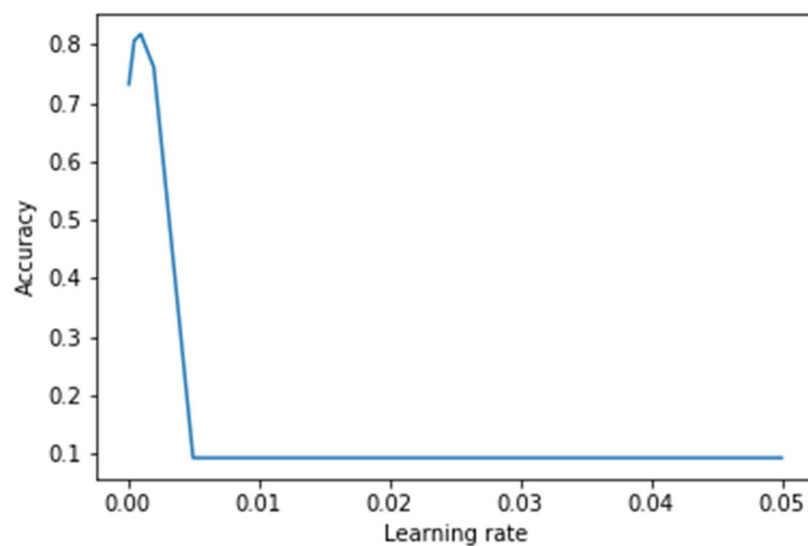




Maintaining the other parameters at constant value (learning rate = 0.001, layer number = 2, no. of neurons in the hidden layer = 275, epoch = 1) I changed the batch size and obtained these results.

As I increased the batch size further beyond 10, the accuracy decreased sharply with a minimum value obtained at 50. Between 2 to 10, the accuracy fluctuates with a maximum value of accuracy obtained at a batch size = 3

2. Learning Rate:

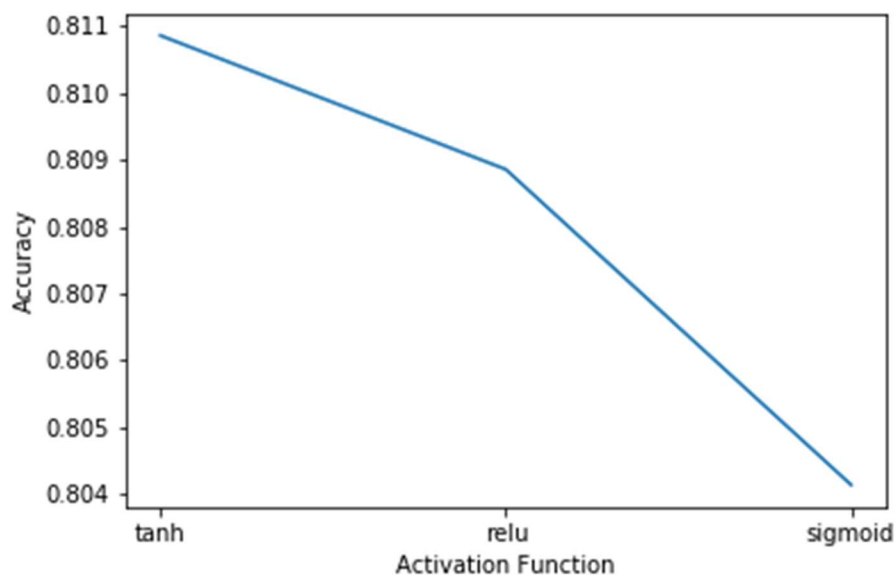


Maintaining the other parameters at constant value (batch size = 5, layer number = 2, no. of neurons in the hidden layer = 275, epoch = 1), changed the learning rate

As, from the graph on increasing the learning rate from 0 to 0.001 the accuracy increases with a maximum accuracy at 0.001 i.e. 0.83. Upon further increasing the learning rate, there is a sharp decrease in the accuracy which becomes constant after $l = 0.005$. This is because the learning rate > 0.005 , is too high to train the model and the minima of the gradient is never achieved. Thus, the accuracy is about 10% in this case.

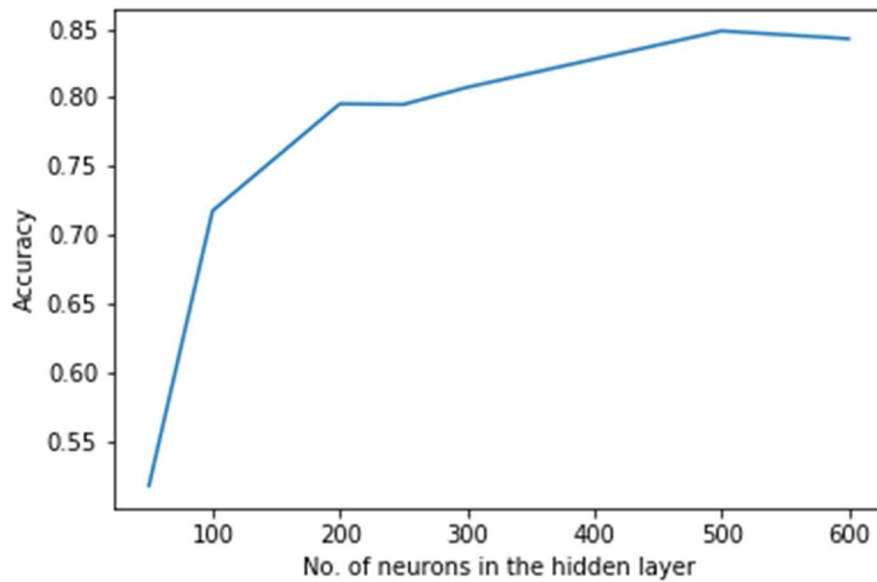
3. Type of Activation Function:

The different types of activation functions that can be used are softmax, relu, tanh, sigmoid etc. However, softmax preferably is used in the classification problem in the final output layer. We vary the activation functions of the hidden layers.



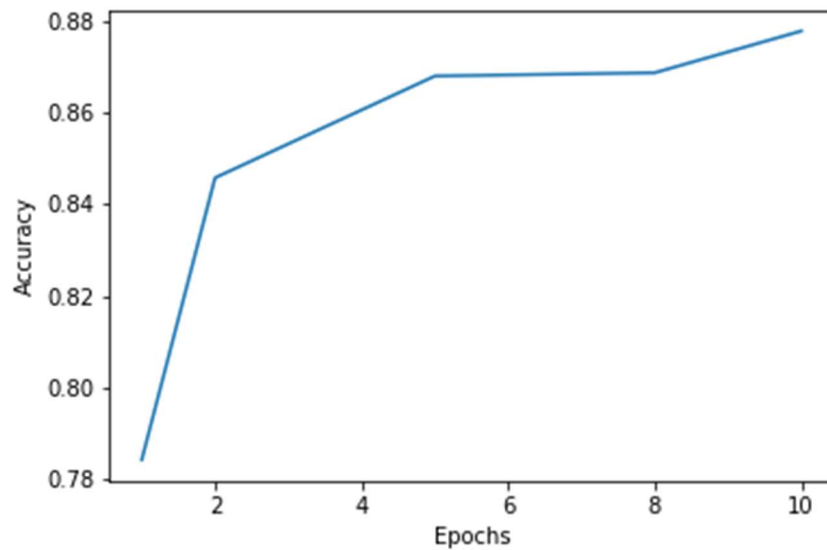
Maintaining the other parameters at constant value (learning rate = 0.001, batch size = 5, layer number = 2, no. of neurons in the hidden layer = 275, epoch = 1), I changed the activation function on the hidden layer and observed that maximum accuracy was obtained using tanh activation function on the hidden layer, followed by relu and sigmoid.

4. Number of neurons in the hidden layer



Maintaining the other parameters at constant value (learning rate = 0.001, batch size = 5, layer number = 2, epoch = 1), I changed the number of neurons in the hidden layer.

5. Epochs



Maintaining the other parameters at constant value (learning rate = 0.001, batch size = 5, layer number = 2, no. of neurons in the hidden layer = 275), the accuracy increased as I increased the number of epochs.

6. Number of hidden layers:

Maintaining the other parameters at constant value (learning rate = 0.001, batch size = 5, epoch=1), I varied the number of hidden layers and obtained the following results.

The accuracy obtained using 1 hidden layer with 275 neurons is 85.4% (0.8548571428571429)

The accuracy obtained using 2 hidden layers with neurons 275 and 100 respectively is around 10.9% (0.10928571428571429)

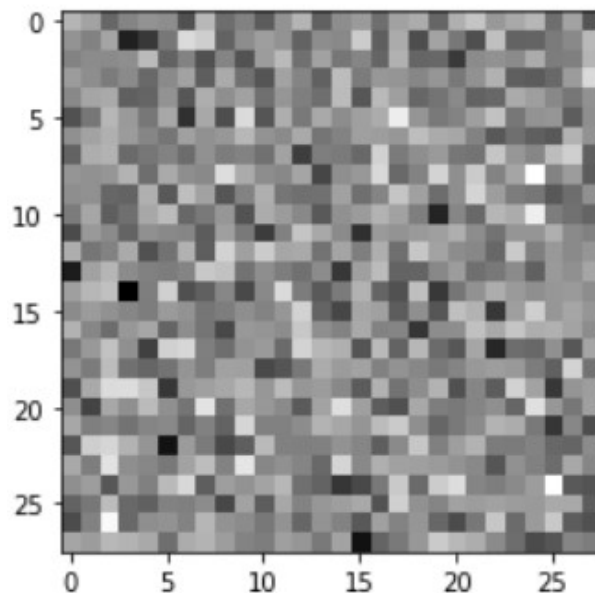
The accuracy obtained using 3 hidden layers with neurons 275, 50 and 50 respectively is around 10.9% (0.10928571428571429)

Thus, in my training model, I achieved a high accuracy using 1 hidden layers.

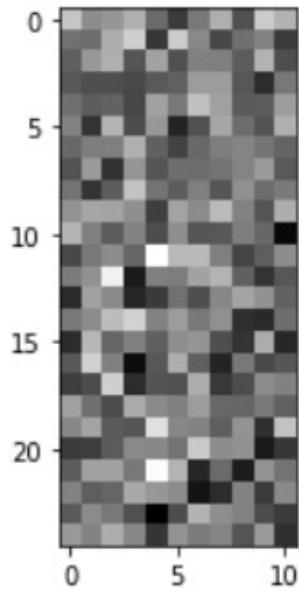
Hidden layer visualization

Plotting the weight vectors obtained on using 1 hidden layer, we get an image of how the first neuron sees the input after it has been activated.

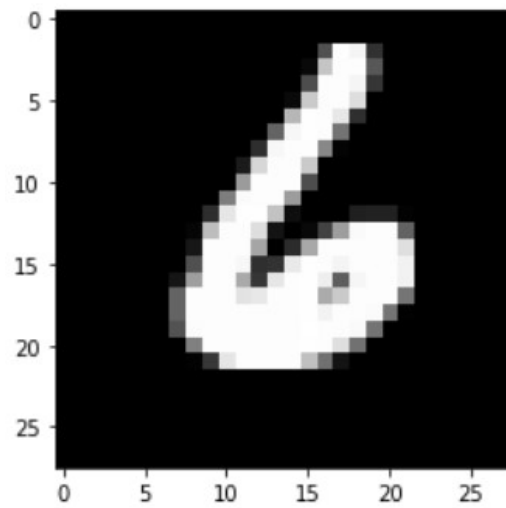
The visualization of the weights of the first layer with 784 neurons



The visualization of the weights of the second layer with 275 neurons

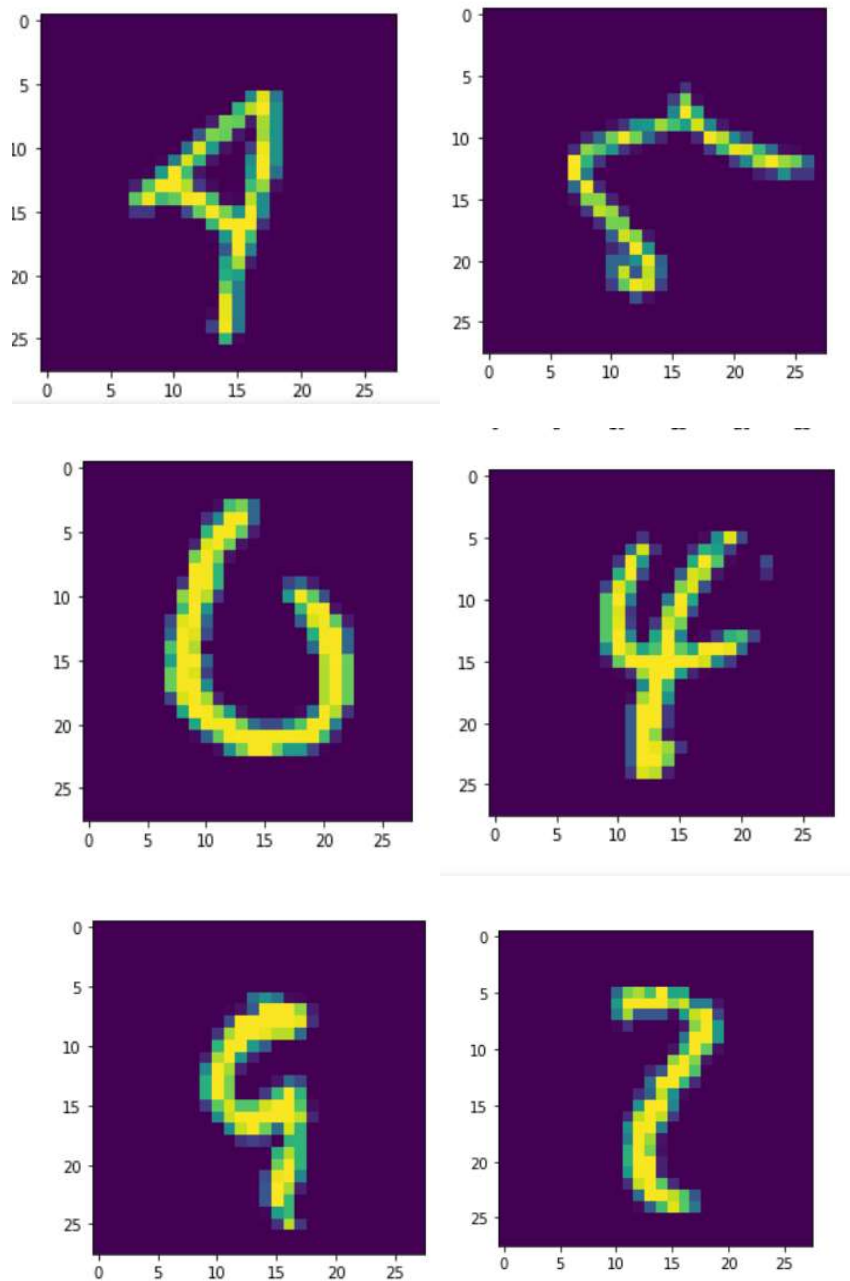


Whereas the input looks like this:



Misclassifications

The following images were misclassified when I trained my data set.



The above are some examples of misclassified images

What we can observe in these images are that the misclassification occurred because :

1. The image was rotated and flipped in such a way that it closely resembled some other digit (as in image 1 which looks like a 7 but is a 4)
2. The image was vaguely drawn and could look like no digit at all(as in image 4)
3. The image was in parts, i.e., the writing was not continuous which led to confusions (as in image 3)
4. The image was so thick that the boundary seemed like another part of the image

There could have been other reasons but mostly these were the ones which were observed especially rotation or flipping affected the decision/classification the most.