ASSIGNMENT 1: ELL409

CLASSIFICATION OF HANDWRITTEN DIGITS USING BACKPROPAGATION ALGORITHM

SUBMITTED BY: ISHA CHAUDHARY
2018EE30614

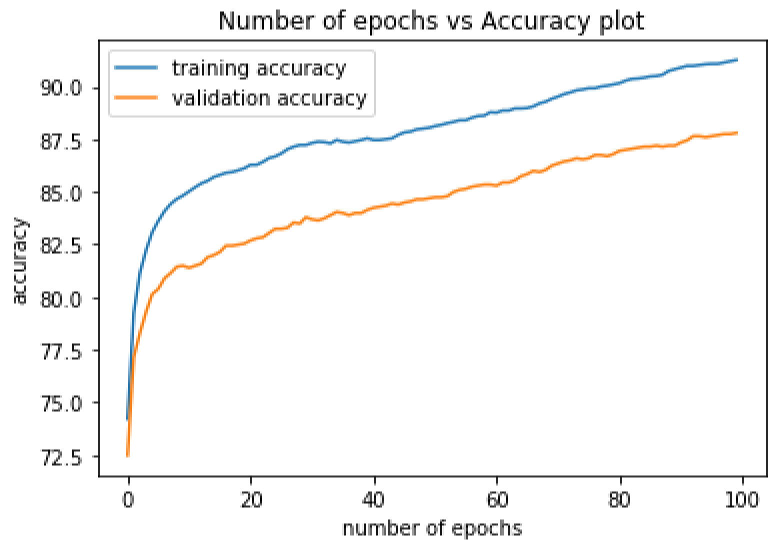
Initial Condition of the network and hyper parameters:

- *Data set*: Subset of MNIST data set of handwritten digits
- *Size of training data*: 7000 samples
- *Size of test set*: 3000 samples
- *Breakup of training and validation set*: 5000:2000
- *Initialization of weights*: Random numbers between -0.1 and 0.1
- *Initial choice of number of hidden layers*: 1
- *Initial choice of number of neurons in the hidden layers*: 128
- *Number of units in input layer*: 784 (28x28) pixel brightness values in the range of 0 to 255 for each input pattern (These inputs were normalized to have values between 0 and 1)
- *Number of units in output layer:* 10
- *Activation Function for the neurons in the hidden layer:* Sigmoid
- *Activation Function for the neurons in the output layer:* to find their activation, sigmoid was used, used argmax for finding the class label for the output neurons

★ The training data is shuffled after every epoch.

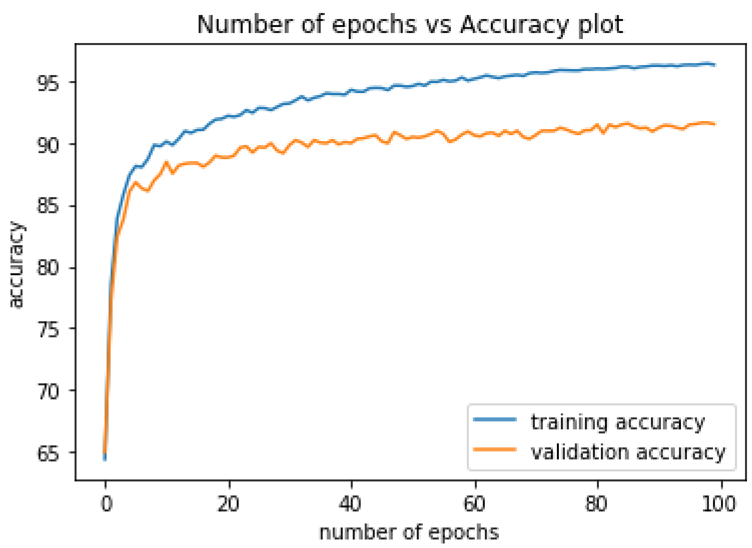
Accuracy obtained with above-mentioned hyper parameters (learning rate = 1, number of epochs = 100, batch size = 40):
Training accuracy = 88.04% ; Validation Accuracy = 85.84%

Changing the activation function for hidden layer to tanh: The computations with tanh were observed to be slower than those with sigmoid.
(learning rate = 2, number of epochs = 100, batch size = 100)
Training accuracy = 99.91% ; Validation accuracy = 87.79%



- *Variation of the training and validation accuracy with the number of epochs*

1.

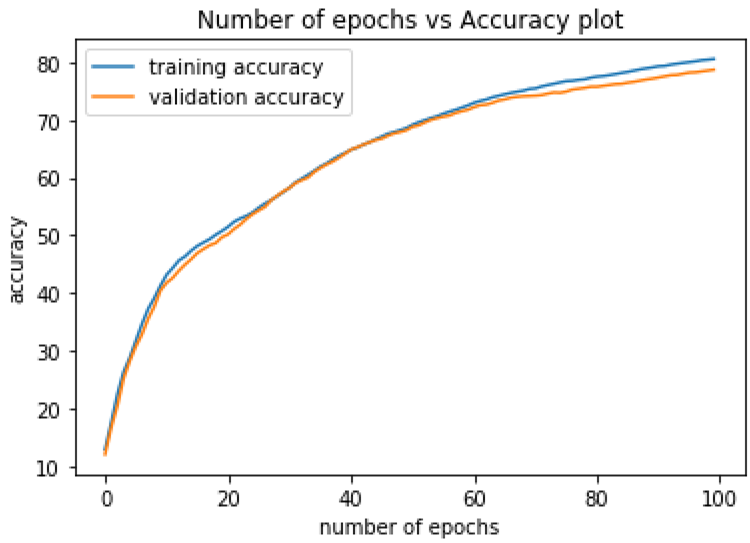


Batch size = 4; Learning rate = 2

<u>Inference from the plot:</u> This is a trend in the accuracy as the number of epochs varies till 100. Clearly, the training error and validation error increases as the network is learning the pattern. The anticipated occurence of the network overfitting over the training data when low validation accuracy and high training accuracy occur does not occur here as the number of epochs over the data in this case is very less. (The number of epochs are limited by the time available and the capacity of the machine on which this network is being run.) If the number of epochs are increased

considerably, a dip in the validation accuracy is predicted owing to the occurence of overfitting on the training data.
Due to the big learning rate, the graph is seen to oscillate a bit, and does not achieve a steady value of accuracy over the training data.

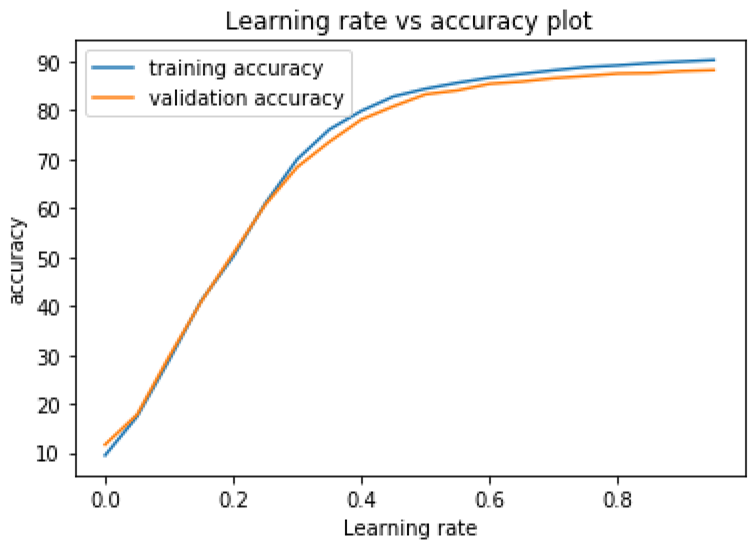
2.



Learning rate: 1; Batch size: 100

This plot also depicts similar characteristics as earlier one.

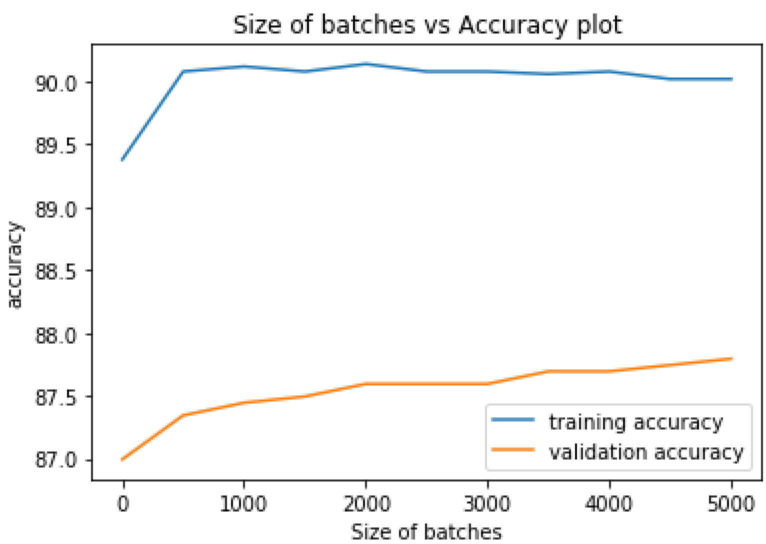
- *Varying the learning rate and checking the trend in the accuracy of the network*



Batch size = 100; Epoch = 50

<u>Inference from the plot:</u> As the learning rate is varied over 0 to 1, the accuracy is observed to increase, as depicted in the above graph. This happens if the learning rate is changed in the steps of 0.05 with every trial.

- *<u>Variation of the training and validation accuracy with batch size</u>*



Number of epochs = 5; learning rate = 1

Inference from the plot: The above trend of variation of the training and validation set accuracies shows that the training accuracy saturates around 90% as the batch size increases, whereas the validation accuracy rises with batch size. This indicates the effectiveness of using the entire training set for training network over stochastic gradient descent, which is just a good approximation. The increase in accuracy with increase in batch size also demonstrates the effectiveness of Mini Batch Gradient Descent over Stochastic Gradient Descent.

- *Changing the number of hidden layer neurons to 64:* (learning rate = 1, no. of epochs = 100, batch size = 40)
  Training accuracy = 88.8%
  Validation accuracy = 85.84%

- *Introducing another hidden layer with 128 units* (learning rate = 2, no. of epochs = 100, batch size = 100)*:*
  Training Accuracy = 86.88%
  Validation Accuracy = 84.5%