Deep Learning and its Applications

CS671

Course Instructor : Dr. Aditya Nigam

Assignment 1 : Question 3

Layer API - A Simple Neural Network

Aman Khandelwal	B16007
Amrendra Singh	B16010
Bharat Lodhi	B16015

https://github.com/ashking13th/deepLearning

1 Problem Statement

You are required to make a simple fully connected network to classify the MNIST datasetand the dataset you made in part 1. The caveat here is that you can not use layer APIs already provided in your library of choice. You will need to code a dense layer API as exhaustive as possible using in-built operations for matrix multiplication, etc.

2 Model

For MNIST dataset - 1 hidden layer, 20 neurons(hidden layer), 784 neurons(input layer), 10 neurons(Output layer), epochs = 100, batch size = 200, learning Rate = 0.01.
For Ques-1 dataset - 1 hidden layer, 10 neurons(hidden layer), 2352 neurons(input layer), 96 neurons(Output layer), epochs = 100, batch size = 200, learning Rate = 0.01. Used the below rule for number of neurons for hidden layer - Upper bound on number of hidden neurons that won't result in over-fitting is: r

$$N_h = \frac{N_S}{\alpha * (N_i + N_O)} \tag{1}$$

$nkhN_i$ = number of input neurons
N_O = number of output neurons
N_S = number of samples in training set
α = an arbitrary scaling factor usually 2-10

3 Learning Curves

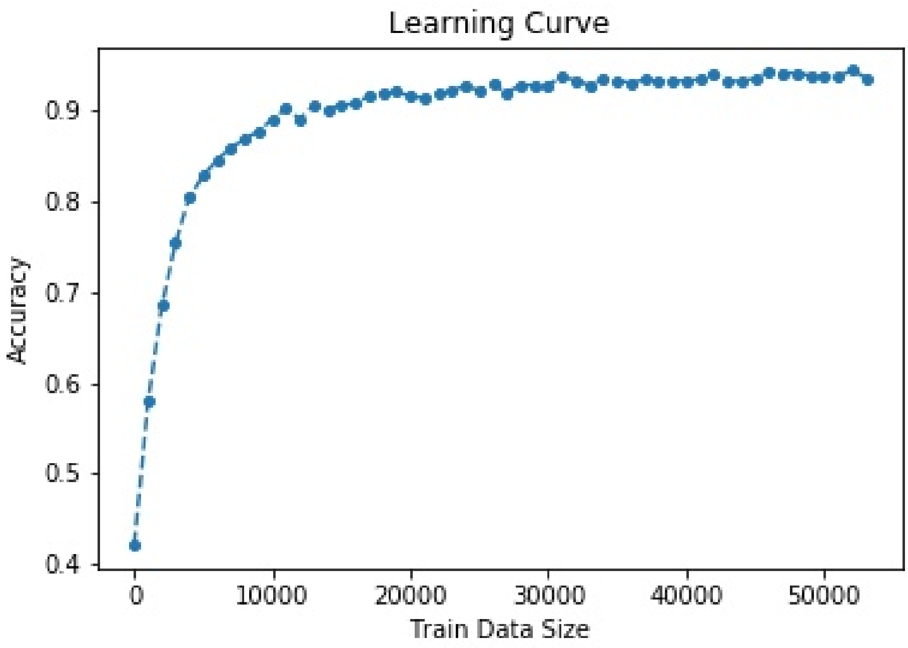

Figure 1. Learning Curve for MNIST Dataset

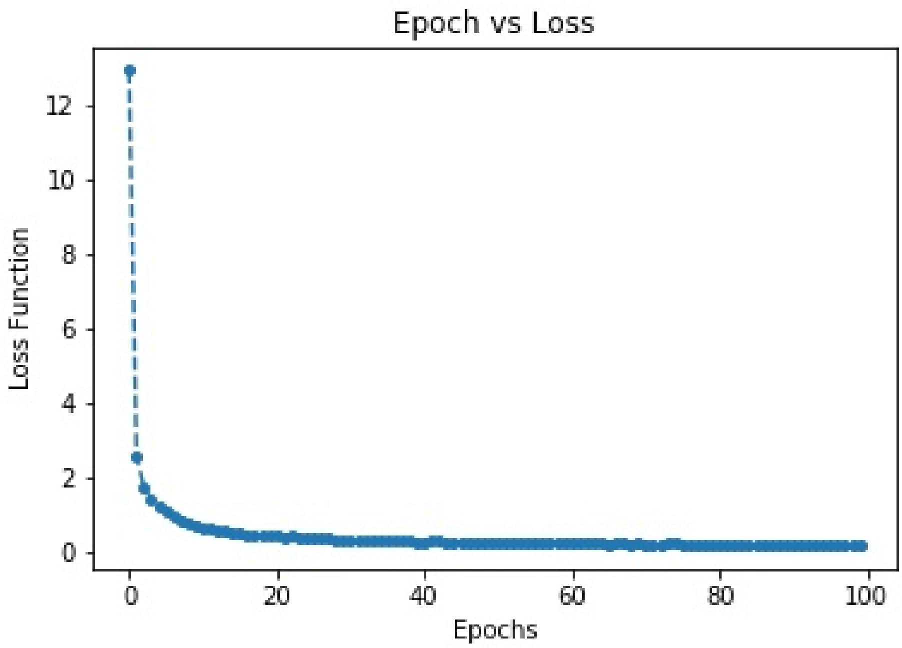

Figure 2. Epoch vs Loss Curve for MNIST Dataset

4 F-scores

5 Confusion Matrices

5.1 MNIST Dataset

A

Digit	0	1	2	3	4	5	6	7	8	9
0	470	0	3	0	3	6	9	0	7	4
1	0	582	2	1	0	0	2	6	1	1
2	2	1	478	14	4	1	2	8	6	1
3	0	1	15	465	0	12	0	4	4	2
4	0	0	5	0	450	0	3	3	5	16
5	2	0	0	17	1	396	6	0	5	2
6	1	2	2	0	9	9	453	0	11	0
7	2	1	2	7	0	2	0	466	1	1
8	1	3	15	6	1	9	4	1	461	9
9	0	0	0	2	7	4	0	3	4	473

Table 1. Confusion Matrix for MNIST Dataset

6 Variations

We experimented with various values of hyper-parameters:

1. **Layer:** We only tried one hidden layer.
2. **Epochs:** We varied the epochs from 5 to 200
3. **Neurons in Hidden Layer:** We varied the number of neurons in the hidden layer between 5 to 100
4. **Batch Size:** We varied the batch size from 1 to 1000

7 Inferences

- On varying the number of training samples from 5000 to 50000 the accuracy increased for MNIST dataset same happened for Ques-1 dataset. This implies more the training data more we are able to predict the new samples correctly.
- On varying the number of neurons in the hidden layer for both the datasets the accuracy first increased and then decreased. This implies that on using more neurons the data was getting overfitted.

- On varying the batch size, first of all for too small batch size the accuracy was poor and also the same happened for too large batch size. In case of large batch size the number of changes in weights were less so learning was not proper for given number of epochs and in case of small batch size the changes in weights were so frequent that important features were lost.

- On varying the number of epochs the accuracy followed the same trend as on varying the number of neurons. This happened because large number of epochs meant very good fit for training set which in turn inplies overfitting.

Bibliography

[1] Stack Overflow
`https://stackoverflow.com`

[2] Towards Data Science
`https://towardsdatascience.com/image-classification-in-10-minutes\`
`-with-mnist-dataset-54c35b77a38d`

[3] Tensorflow Tutorials
`https://www.tensorflow.org/tutorials`

[4] Data Science Blog
`https://medium.com/datadriveninvestor/my-take-at-the-mnist-dataset\`
`-97304dff2057`

[5] Analytics Vidha
`https://www.analyticsvidhya.com/blog/2016/10/`
`an-introduction-to-implementing-neural-networks-using-tensorflow/`