CS671 - Deep Learning and its Applications

Programming Assignment 1

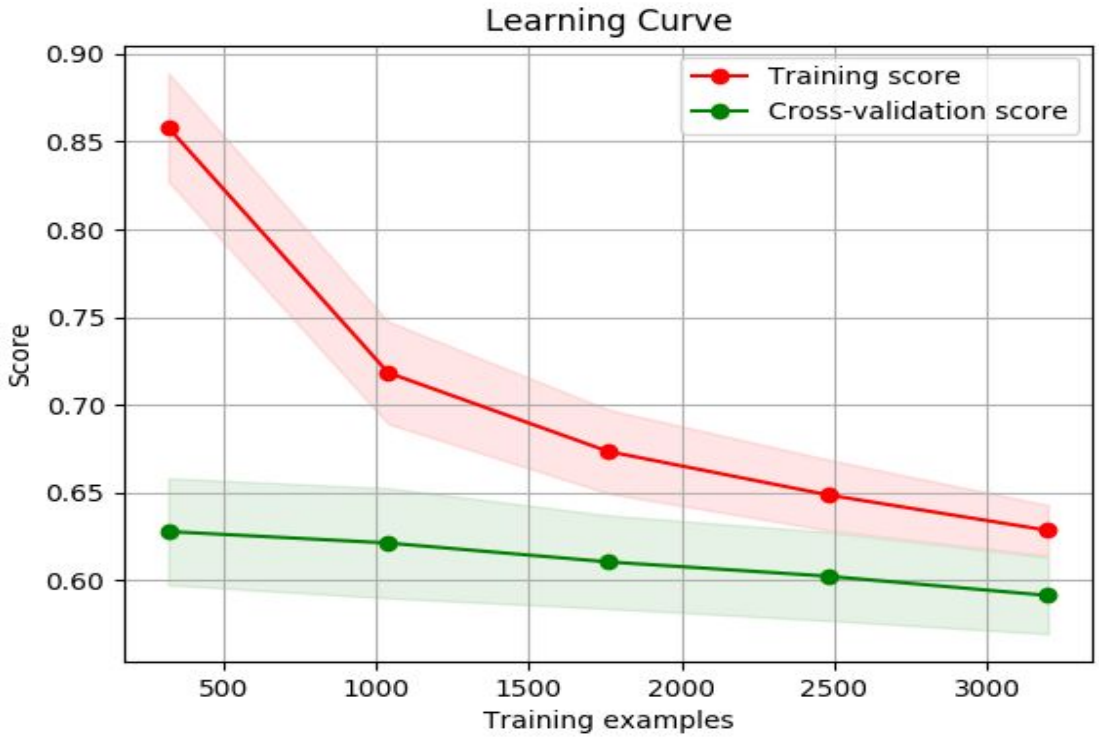
Group - 20

Problem Statement

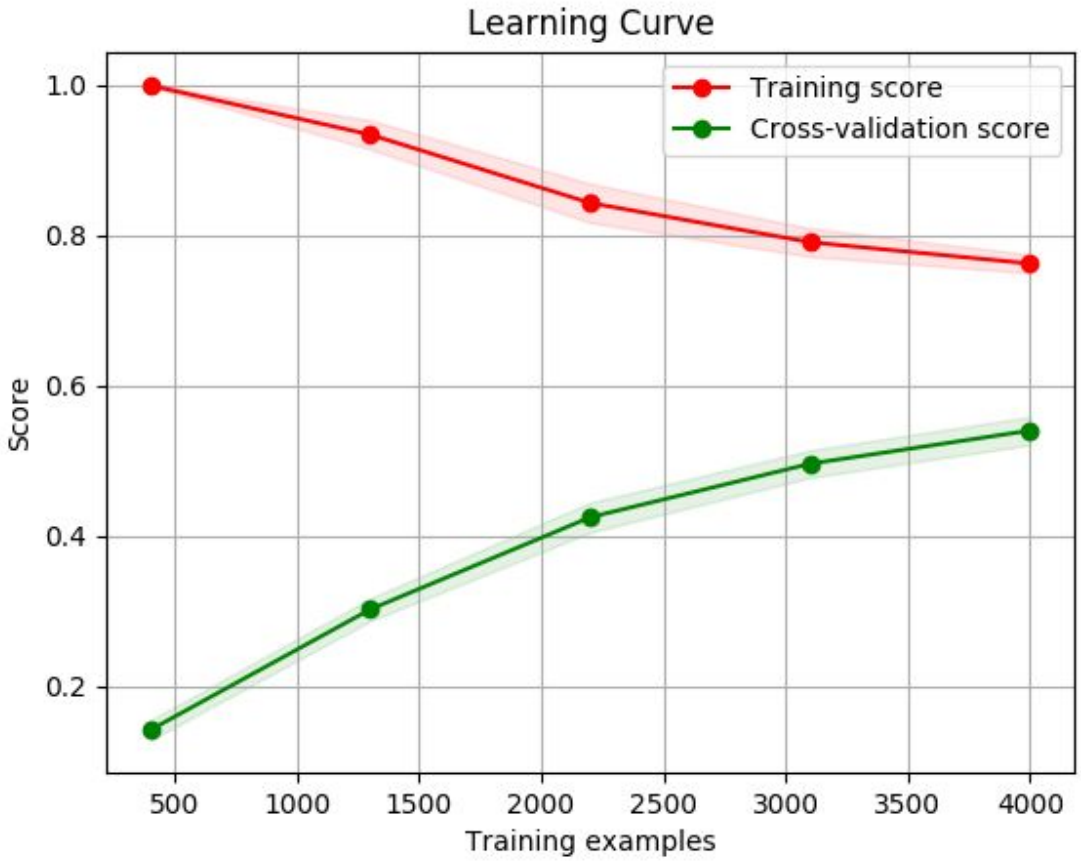
You are required to make a simple fully connected network to classify the MNIST dataset and the dataset you made in part 1. The caveat here is that you can not use layer APIs already provided in your library of choice. You will need to code a dense layer API as exhaustive as possible using in-built operations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

Learning Curves

Learning curve for MNIST dataset



Learning curve for Line Dataset



F-Scores

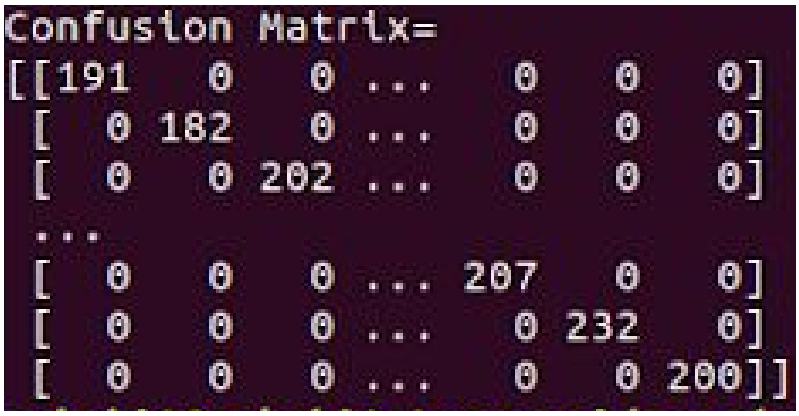
For Line dataset,

F1 score = 99.99 %

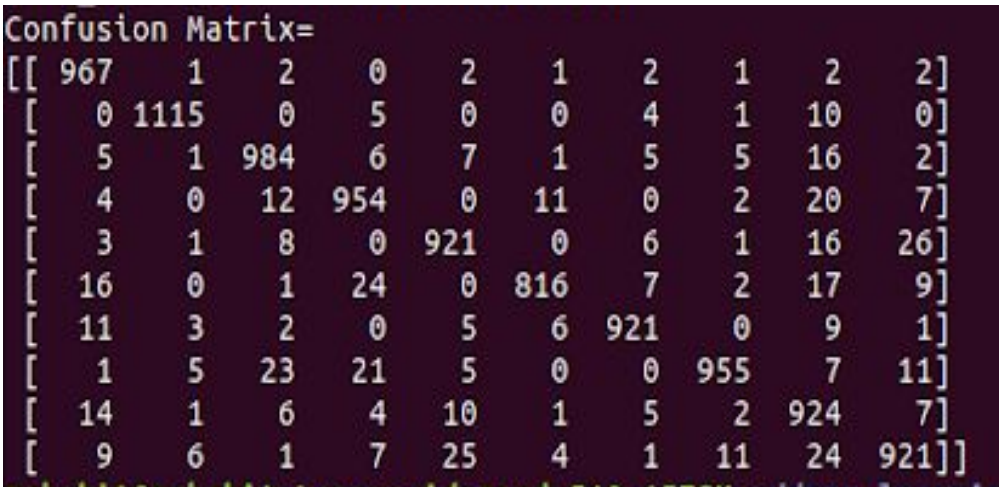
For MNIST dataset,

F1 score = 94.78 %

Confusion Matrices



For full confusion matrix of lines data set created in Line Dataset Generation click here



For full confusion matrix of MNIST dataset click here

Variations Tried

- Increased the number of units in the hidden layer from 64 to 128 for MNIST dataset to prevent underfitting and tried the same number of units in hidden layer for line dataset.
- Decreased the learning rate from 0.1 to 0.01 so that it converges smoothly.
- Tried to train with 5 epochs instead of 1 to improve the accuracy.

Inferences

- We became familiar with the tensorflow usage and computational graph.
- Understood the mathematics behind the **Dense layer API** of tensorflow by implementing it ourselves.
- Increasing the number of hidden layers improves the accuracy.
- Increasing the number of units within a hidden layer helps to prevent underfitting.
- Increasing the number of epochs improves the accuracy but may lead to overfitting.
- Training in a batches is recommended so that the weights are not updated very frequently.
- Accuracy
    - For Line dataset: 99.99 %
    - For MNIST dataset: 94.78 %