Machine Learning with Big Data
Assignment 2
M22MA003

Q1. Download the MNIST dataset from here. Pick its training and test sets. Each sample in the dataset is represented by a 784-dimensional vector. There are ten classes (the ten digits, `0' to `9'), and each sample is associated with one class.
(1) Using the raw binary features, implement the streaming Naive Bayes algorithm and classify the test data.

METHOD 1 :-

- First of all we have imported necessary libraries and after that downloaded the MNIST dataset using the pytorch datloader and then split the dataset and prepared the dataloader as train and test .
- Next we have defined the neural network model architecture
- **`Define hyperparameters`**
    - `num_epochs = 5`
    - `batch_size = 128`
    - `learning_rate = 0.001`
- Next we have defined the model by defining class **class NaiveBayes(nn.Module):**
- Next we have called the model and defined the loss fn and optimiser as
    - model = NaiveBayes(num_features=28*28, num_classes=10)
    - optimizer = optim.Adam(model.parameters(), lr=learning_rate)

- Next we have trained the model and after that tested the model for the naive bayes algo.
- We have also calculated the train and testing time of the model
- All the results are shown below

METHOD 2:-

- Here we have used basic model in which we have updated the model and then predict defined under the class streaming naive bayes and after that we have called the naive bayes fn which was define in which we have called the class object and then updated the training part and then predict the value.
- Implementation was for 784 dimensions and also training and testing time was shown

Results:-

METHOD 1

Epoch [1/5], Step [100/469], Loss: -1.1734
Epoch [1/5], Step [200/469], Loss: -1.1731
Epoch [1/5], Step [300/469], Loss: -1.1692
Epoch [1/5], Step [400/469], Loss: -1.1837
Epoch [2/5], Step [100/469], Loss: -1.1820
Epoch [2/5], Step [200/469], Loss: -1.1881
Epoch [2/5], Step [300/469], Loss: -1.1876
Epoch [2/5], Step [400/469], Loss: -1.1879
Epoch [3/5], Step [100/469], Loss: -1.1933
Epoch [3/5], Step [200/469], Loss: -1.1821
Epoch [3/5], Step [300/469], Loss: -1.2061
Epoch [3/5], Step [400/469], Loss: -1.1938
Epoch [4/5], Step [100/469], Loss: -1.1888
Epoch [4/5], Step [200/469], Loss: -1.1812
Epoch [4/5], Step [300/469], Loss: -1.1786
Epoch [4/5], Step [400/469], Loss: -1.1864
Epoch [5/5], Step [100/469], Loss: -1.1935
Epoch [5/5], Step [200/469], Loss: -1.1731
Epoch [5/5], Step [300/469], Loss: -1.1715
Epoch [5/5], Step [400/469], Loss: -1.2157
Test Accuracy of the model on the 10000 test images: 51.09 %
Training and testing time are:115.5454and 2.7727

Classification on the test set for n_dimension = 100 are:

[tensor([1, 4, 7, 0, 2, 9, 1, 2, 3, 3, 0, 4, 1, 0, 7, 4, 2, 1, 4, 1, 1, 2, 6, 7,
1, 0, 1, 2, 0, 7, 7, 1, 4, 0, 9, 4, 1, 7, 7, 7, 1, 1, 8, 4, 7, 4, 1, 6,
3, 6, 7, 4, 0, 4, 6, 0, 3, 7, 3, 1, 0, 3, 1, 4, 2, 1, 6, 6, 9, 0, 1, 0,
0, 9, 7, 1, 4, 1, 9, 3, 1, 2, 6, 1, 3, 3, 1, 0, 2, 9, 9, 2, 1, 9, 7, 3,
1, 1, 2, 2, 2, 0, 3, 6, 6, 2, 0, 7, 3, 2, 9, 9, 7, 2, 1, 1, 1, 6, 1, 2,
3, 2, 1, 2, 1, 2, 6, 3]), tensor([3, 3, 2, 0, 4, 2, 3, 7, 0, 9, 2, 2, 7, 1, 6, 7, 2, 7, 3, 6, 0, 4, 2, 6,
1, 4, 1, 3, 2, 0, 4, 1, 2, 0, 3, 6, 2, 4, 8, 1, 7, 8, 1, 9, 4, 4, 6, 1,
7, 9, 7, 9, 7, 9, 7, 0, 9, 2, 9, 4, 0, 1, 9, 1, 0, 1, 0, 1, 7, 7, 2, 9,
8, 0, 9, 7, 7, 4, 3, 4, 6, 2, 0, 1, 2, 1, 3, 0, 9, 7, 1, 0, 3, 8, 0, 1,
7, 4, 3, 1, 7, 1, 7, 7, 4, 3, 1, 1, 6, 6, 3, 7, 4, 2, 3, 1, 7, 7, 6, 1,
3, 9, 3, 7, 3, 7, 1, 1]), tensor([9, 7, 6, 0, 3, 7, 1, 1, 1, 7, 0, 7, 4, 0, 4, 0, 7, 3, 2, 2, 7, 3, 6, 7,
4, 3, 7, 1, 2, 6, 7, 4, 7, 4, 3, 4, 2, 6, 0, 8, 0, 0, 4, 9, 1, 2, 7, 1,
1, 4, 4, 1, 3, 4, 4, 0, 0, 4, 7, 3, 7, 1, 4, 4, 7, 9, 1, 9, 0, 1, 7, 0,
1, 7, 7, 9, 7, 4, 9, 9, 2, 1, 4, 2, 7, 1, 7, 4, 7, 7, 0, 3, 7, 1, 1, 3,
0, 2, 9, 7, 0, 7, 1, 3, 4, 2, 6, 7, 3, 0, 2, 4, 2, 4, 1, 7, 6, 9, 2, 3,
1, 7, 7, 0, 4, 2, 4, 1]), tensor([2, 7, 0, 6, 9, 1, 6, 3, 2, 7, 1, 4, 6, 4, 0, 6, 0, 9, 4, 3, 9, 6, 0, 0,

METHOD 2:-

Accuracy of 784 dimension MNIST dataset using NB algo:-84.4900 %
Training and testing time are:0.3125and 4.7613

classification on the test set are
[7, 2, 1, 0, 4, 1, 4, 9, 4, 9, 0, 6, 9, 0, 1, 3, 9, 7, 3, 4, 9, 6, 6, 5, 4, 0, 7, 4, 0, 1, 3, 1, 3, 0, ……]

(2) Project the binary features into a lower dimensional space using a dimensionality reduction technique of your choice (such as PCA, LDA, t-SNE, etc.), by varying the dimensionality in the range {50,100,200}, and classify the test data using the same algorithm.

ANS:

Here we have basically applied the pca over the dataset to reduce the dimension of the MNIST dataset and after that

We have applied the above two **METHODS** in the 3 sets of dimensions-100,200,500 using PCA

As in the first part of the question ,we have defined the two methods for applying naive bayes process.

1. Neural Network (Pytorch): **METHOD 1**
2. Streaming NaiveBayes from scratch (including Loglikelihood) : **METHOD 2**

Next we have classified on the test dataset and also calculated the training and testing time in each method in each dimensions.

Method 1:-

<u>For dimension-100</u>
Epoch [1/5], Step [100/469], Loss: -1548.6983
Epoch [1/5], Step [200/469], Loss: -1621.7729
Epoch [1/5], Step [300/469], Loss: -1675.6318
Epoch [1/5], Step [400/469], Loss: -1591.1667
Epoch [2/5], Step [100/469], Loss: -1464.2512
Epoch [2/5], Step [200/469], Loss: -1374.0240
Epoch [2/5], Step [300/469], Loss: -1523.7629
Epoch [2/5], Step [400/469], Loss: -1459.0886
Epoch [3/5], Step [100/469], Loss: -1527.0059
Epoch [3/5], Step [200/469], Loss: -1481.8685
Epoch [3/5], Step [300/469], Loss: -1525.2034
Epoch [3/5], Step [400/469], Loss: -1424.3986
Epoch [4/5], Step [100/469], Loss: -1489.9663
Epoch [4/5], Step [200/469], Loss: -1508.3957

Epoch [4/5], Step [300/469], Loss: -1600.7225
Epoch [4/5], Step [400/469], Loss: -1560.7544
Epoch [5/5], Step [100/469], Loss: -1490.0295
Epoch [5/5], Step [200/469], Loss: -1489.9226
Epoch [5/5], Step [300/469], Loss: -1605.6326
Epoch [5/5], Step [400/469], Loss: -1657.5626
Test Accuracy of the model on the 10000 test images: 11.98 %
Training and testing time are:18.3767 and 0.0742

For dimension -200
<class '__main__.ReducedMNISTDataset'>
<class '__main__.ReducedMNISTDataset'>
Epoch [1/5], Step [100/469], Loss: -1569.1654
Epoch [1/5], Step [200/469], Loss: -1610.3156
Epoch [1/5], Step [300/469], Loss: -1476.8329
Epoch [1/5], Step [400/469], Loss: -1361.0752
Epoch [2/5], Step [100/469], Loss: -1552.6142
Epoch [2/5], Step [200/469], Loss: -1321.4903
Epoch [2/5], Step [300/469], Loss: -1489.6055
Epoch [2/5], Step [400/469], Loss: -1537.7365
Epoch [3/5], Step [100/469], Loss: -1583.8174
Epoch [3/5], Step [200/469], Loss: -1520.1951
Epoch [3/5], Step [300/469], Loss: -1497.9484
Epoch [3/5], Step [400/469], Loss: -1556.2628
Epoch [4/5], Step [100/469], Loss: -1340.3122
Epoch [4/5], Step [200/469], Loss: -1515.6806
Epoch [4/5], Step [300/469], Loss: -1790.7095
Epoch [4/5], Step [400/469], Loss: -1519.3068
Epoch [5/5], Step [100/469], Loss: -1413.6588
Epoch [5/5], Step [200/469], Loss: -1557.3858
Epoch [5/5], Step [300/469], Loss: -1625.1346
Epoch [5/5], Step [400/469], Loss: -1563.2300
Test Accuracy of the model on the 10000 test images: 17.75 %
Training and testing time are:19.3666 and 0.1245

For dimension - 500
Epoch [1/5], Step [100/469], Loss: -1495.5844
Epoch [1/5], Step [200/469], Loss: -1443.5199
Epoch [1/5], Step [300/469], Loss: -1340.1326
Epoch [1/5], Step [400/469], Loss: -1653.7020
Epoch [2/5], Step [100/469], Loss: -1633.0654
Epoch [2/5], Step [200/469], Loss: -1375.6402
Epoch [2/5], Step [300/469], Loss: -1599.7052
Epoch [2/5], Step [400/469], Loss: -1380.1297

Epoch [3/5], Step [100/469], Loss: -1478.8068
Epoch [3/5], Step [200/469], Loss: -1464.8296
Epoch [3/5], Step [300/469], Loss: -1413.7605
Epoch [3/5], Step [400/469], Loss: -1499.8425
Epoch [4/5], Step [100/469], Loss: -1403.4839
Epoch [4/5], Step [200/469], Loss: -1536.8574
Epoch [4/5], Step [300/469], Loss: -1474.8797
Epoch [4/5], Step [400/469], Loss: -1505.7151
Epoch [5/5], Step [100/469], Loss: -1369.0181
Epoch [5/5], Step [200/469], Loss: -1589.1549
Epoch [5/5], Step [300/469], Loss: -1689.0353
Epoch [5/5], Step [400/469], Loss: -1323.7513
Test Accuracy of the model on the 10000 test images: 15.38 %
Training and testing time are:20.0661 and 0.0810

Method 2:-
Training and testing time are:0.1919 and 2.7441
Accuracy of 100 dimension MNIST dataset using NB algo:-9.8000 %

Training and testing time are:0.1968 and 3.4344
Accuracy of 200 dimension MNIST dataset using NB algo:-9.8000 %

Training and testing time are:0.2237 and 4.8819
Accuracy of 500 dimension MNIST dataset using NB algo:-9.8000 %

Note:- Classification is done in the code since it is big tensor type dataset so it is not mentioned in the report .Also it is done for both the methods

(3) Contrast the training and testing times and accuracy of categorization in (1) and (2). In both quarters, we have determined how long the training and testing phases will take and how accurate they will be.
Ans:-

Category 1:
Method 1:
Test Accuracy of the model on the 10000 test images: 51.09 %
Training and testing time are:115.5454and 2.7727
Category 2:
Method 1:(for 100 dimension)
Test Accuracy of the model on the 10000 test images: 11.98 %
Training and testing time are:18.3767 and 0.0742

Here we have seen that accuracy has been decreased from 51.09 to 11.98 % and also training and testing time has been decreased as the dimension is decreased from 784 to 100.

Category 1:
Method 2:
Accuracy of 784 dimension MNIST dataset using NB algo:-84.4900 %
Training and testing time are:0.3125and 4.7613

Category 2:
Method 2:(for 100 dimension)
Training and testing time are:0.1919 and 2.7441
Accuracy of 100 dimension MNIST dataset using NB algo:-9.8000 %

Here we have seen that accuracy has been decreased from 84.409 to 9.8 % and also training and testing time has been decreased from 0.3 to 0.19 and 4.76 to 2.74 as the dimension is decreased from 784 to 100.

(4) Evaluate the results of (1) and (2) in light of the classification precision you found in assignment 2.
Ans:

K=1, Accuracy=93.579%, Training Time=0.0022, Prediction Time=37.81
:
:
K=4, Accuracy=93.960 %, Training Time=0.0023, Prediction Time=38.31
K=5, Accuracy=94.030 %, Training Time=0.0023, Prediction Time=37.90
Taking the accuracy of all q1,q2 and Ass2 and comparing them we can say ➖

	Q1(method 2)	Q2(method 2)	Assg2(k=1)
Accuracy(%)	84.49	9.80	93.57
Training Time (ms)	0.3125	0.1919	0.002
Testing Time(ms)	4.7613	2.7441	37.81

Colab link
Q1:-https://colab.research.google.com/drive/1GxpTXJJOl_0idVDbIT0f9vMzg88aGUPW?usp=sharing
Colab link Q2:-
https://colab.research.google.com/drive/1r-4dPfHF8fsXodNLh-_0h4swRJd-vwFO?usp=sharing

References:-

1.https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/
2.https://www.analyticsvidhya.com/blog/2022/03/building-naive-bayes-classifier-from-scratch-to-perform-sentiment-analysis/
3.Class Notes