Name: Parth Bhatt
Student ID: 1220528816

Project 1 Naive Bayes Classifier

Introduction

The objective of this project was to develop Naive Bayes Classifier to distinguish images from the MNIST dataset and to measure the accuracy of the developed classification model. The sub-dataset of MNIST is provided with 5000 digit “0” images and 5000 digit “1” images for training the Naive Bayes Classifier. A Testing dataset containing 980 digit “0” images and 1135 digit “1” images is provided to classify those test images and find out the accuracy of the model.

Approach

The following steps were performed to classify the images and find the accuracy of Naive Bayes Classifier:

- From the training data, first we calculate the features(average and standard deviation) of each picture(28X28) by using the for loop and numpy average and std function(**numpy.avg()** and **numpy.std()**).
- The above step gives us the average(feature 1) and standard deviation(feature 2) of each picture, which is a numpy array with 5000 elements respectively, in the training dataset for digit “0” and digit “1”.
- Then, we extract parameters from the average and standard deviation of the training dataset of digit “0” and digit “1” by getting mean and variance features using numpy function mean and var(**numpy.mean()** and **numpy.var()**).
- The 8 parameters we extract are:
    - Mean of average of the dataset of digit “0” and “1”.
    - Variance of average of the dataset of digit “0” and “1”.
    - Mean of standard deviation of the dataset of digit “0” and “1”.
    - Variance of standard deviation of the dataset of digit “0” and “1”.
- Now, we calculate the average(feature 1) and standard deviation(feature 2) of the testing dataset of digit “0” and digit ”1” using numpy average and std functions(**numpy.avg()** and **numpy.std()**).
- Function **NB_probability** is defined to calculate the gaussian probability of any point x_i in the testing dataset using the parameters calculated for digit “0” and “1”.
- Function **decide0** takes the features of test dataset for digit “0” and using the Naive Bayes probability function, it classifies the test data whether it is digit “0” or digit “1”.It also calculates the **accuracy of Naive Bayes model** to predict digit “0”.

- Similarly, function **decide1** takes the features of test dataset for digit “1” and using the Naive Bayes probability function, it classifies the test data whether it is digit “0” or digit “1”.It also calculates the **accuracy of Naive Bayes model** to predict digit “1”.

Challenges

- Initially, as the data was provided already. It took a few minutes to understand how the data was structured and that understanding was necessary to calculate the average and standard deviation of each picture using the pixel values in the numpy array.
- The formula to calculate the mean, standard deviation, and variance is easy to implement in python but finding numpy library functions from the documentation was really helpful in implementing the Naive Bayes Classifier.

Conclusion

- The below parameters are calculated from the training dataset of digit “0” and “1” for two-class naive bayes classifier:
    - (No.1) Mean of feature1 for digit0 - **44.2312242347**
    - (No.2) Variance of feature1 for digit0 - **115.259232406**
    - (No.3) Mean of feature2 for digit0 - **87.4581326749**
    - (No.4) Variance of feature2 for digit0 - **100.901069465**
    - (No.5) Mean of feature1 for digit1 - **19.3993681122**
    - (No.6) Variance of feature1 for digit1 - **31.1453053306**
    - (No.7) Mean of feature2 for digit1 - **61.4156445259**
    - (No.8) Variance of feature2 for digit1 - **81.208543733**
- The accuracy of the Naive Bayes Classification for testing dataset of digit “0” and “1”:
    - The accuracy of testing dataset of digit “0” - **0.917346938776**
    - The accuracy of testing dataset of digit “1” - **0.923348017621**
- The accuracy mentioned above is considered good with respect to classifying images with digit “0” and digit “1” but we can improve this accuracy to nearly 98-99% if we use other models like deep learning-Convolutional Neural Network (CNN) models because the convolution network learns through the layers.

`