COMP 6915 - Machine Learning
Assignment 5 Report

by
© Soroush Baghernezhad
& Ramyar Zarza
& Mantra .

Instructor: Karteek Popuri
Department of Computer Science & Scientific Computing
Memorial University of Newfoundland

March 2025

St. John's Newfoundland

Question 1:

In this part, we first identify missing values and duplicated samples. Since neither is present, we proceed to split the data into training and testing sets based on a given condition. To satisfy this condition, we first count the number of samples for each category $(0, 1, \ldots)$, which results in the following distribution:

$$[5894, 6764, 5978, 6096, 5858, 5400, 5929, 6302, 5829, 5950]$$

Next, we select the minimum value among these counts, which is 5400, and take 90% of this value as the minimum number of samples required for each category is 0.9*5400. Finally, the percentage of each category in the test data is as follows:

$$[17.5, 28.1, 18.7, 20.3, 17.0, 10.0, 18.0, 22.9, 16.6, 18.3]$$

which satisfies the condition of having at least 10% in each category.

Question 2:

In this experiment, performing hyperparameter tuning using the entire training dataset is time-consuming when searching for the optimal k. Therefore, for generating the plot below, the code was initially run using the entire training set. However, to speed up subsequent runs, only 500 samples from each class were selected for hyperparameter tuning. Once the optimal k was determined, the entire dataset was used to train the final model. As shown in the figure below, the best value for k is $k = 1$. After training the model with $k = 1$, the error rate was found to be 2.56%.

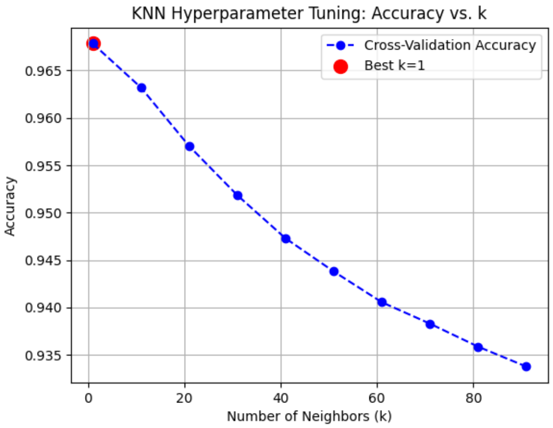

Figure 1: Hyperparameter tuning for k in K-Nearest Neighbors (KNN)

Question 3:

In this experiment, a Support Vector Machine (SVM) with a polynomial kernel was used. For the regularization parameter C, the values tested were:

$$C = [0.1, 1, 10, 50, 100]$$

and for the polynomial degree d, the tested values were:

$$d = [2, 3, 4, 5]$$

A grid search was performed using only a subset of the data (500 samples per class). The results indicated that the optimal parameters were:

$$C = 50, \quad d = 2.$$

Using these optimal parameters, the final model was trained, resulting in an error rate of 1.86%, which was an improvement over the KNN approach.

Question 4:

In this experiment, the `FastAPI` module was used to create a simple Multi-Layer Perceptron (MLP) and train it. Five different architectures for the hidden layers were tested:

$$\begin{gathered} [20, 20, 20], \\ [10, 10, 10, 10], \\ [128, 64], \\ [64, 32, 16], \\ [256, 128]. \end{gathered}$$

Each model was trained for 5 epochs, after which the best-performing architecture was selected and trained for an additional 10 epochs. The optimal architecture was found to be:

$$[256, 128].$$

The final error rate was 2.55%, which is comparable to the KNN approach but worse than the SVM model.

Question 5:

In this experiment, a Convolutional Neural Network (CNN) architecture was used. As CNN models are the best choice for image classification tasks, a small model was designed with two convolutional layers followed by two fully connected layers. The total number of parameters in the model was 421,000. The model was trained for 30 epochs, and the progress was remarkable. The final error rate achieved was 0.11%. The trained model was saved as:
`Q5_model.pth`.