Link to Google Colab:
https://colab.research.google.com/drive/13Dr8nemaMEzRSSFLc0Rc81PaLh4TGoAq?usp=sharing

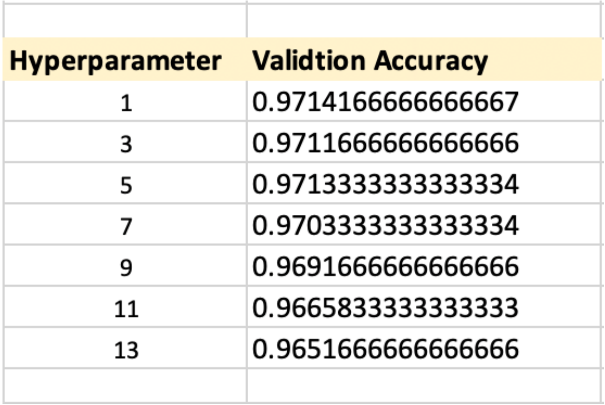
The problem is to implement a k-NN classifier and train the classifier using the training set and tune the hyperparameters to optimize the classifiers performance on the validation set. In other words, we are asked to find the optimal number of nearest neighbors to use.

The size of our training set (X_train) is 60000 and the size of our test set (X_test) is 10000. The training set is used to train the algorithm (i.e. train the model) to recognize the patterns between the data that is provided as input and its corresponding output. The test set on the other hand, is used for evaluating the performance of the trained k-NN model. The test set is being used to understand the degree that the trained k-NN model has accuracy in predicting unseen data points. Finally the validation set is for arranging the hyperparameters. Hyperparameters are set prior to training such as the k-NN. So we will be using the validation set to find the optimal value of n neighbors, by using the values [1, 3, 5, 7, 9, 11, 13] and evaluate the performance of the classifier on the validation set for each value. Before spliting the data we have used the shuffle method on the dataset to ensure that the training and test sets are representative of the entire dataset and to avoid any bias due to the ordering of the data. Then in our task we have reserved 20% of the training data for validation and use the remaining 80% to train our model. You may see that the train_test_split method involves randomly splitting the dataset into two subsets. Here want to ensure that our model can have an accurate performance on the unseen data (high validation accuracy), to ensure this our practice is to split the data into two sets such that the model is trained on the training set and the performance of the model is evaluated on the validation set. Overall by splitting data into two parts, we have not only allowed the model to learn from the training data but also ended up having a way to measure its performance.

k-NN is a supervised machine learning algorithm which operates by finding the k closest data points to a given point and using these points to make predictions. In this homework I have used k-Nearest Neighbor algorithm to find an accuracy. So, with different k values between 1 and 13, I used the data sets we split before. I used training data to train our model with the changing hyperparameter k. Then, I validate the model and store the accuracies in order to later decide on the k value that yields the best accuracy result. Once I found the best_k value which gives best validation accuracy then the data is tested with the best_k value. If the model performs well on the validation set we can be confident that it will also be efficient on the unseen data.

We did not perform any feature extraction or preprocessing in this homework.

You may see the table containing different k values and their corresponding accuracy values below:



Hyperparameter	Validtion Accuracy
1	0.9714166666666667
3	0.9711666666666666
5	0.9713333333333334
7	0.9703333333333334
9	0.9691666666666666
11	0.9665833333333333
13	0.9651666666666666

From the table, we can see that k = 1 value yielded the best accuracy result with 0.9714166666666667 thus I have selected k = 1 as my model.

We have obtained our results on the validation set with the k-NN classification approach. I have tried different k values (n_neighbors) and observed that each of them have different accuracy rates according to the accuracy scores that I have calculated using the accuracy_score function. It can be seen that k = 1 (97%) has the best validation accuracy and k = 13 (96%) has the lowest validation accuracy. At this point I want to explain more about the concept of validation accuracy and also the meaning of the having a high or a low validation accuracy. Validation accuracy is the measure of how well the k-NN algorithm is performing on the previously unseen data. Our aim to choose the k value with high validation accuracy in the sense that a high validation accuracy indicates that the algorithm is able to generalize well to the new data. When an algorithm is able to generalize well on unseen data this means that, our model can also have accurate performance on the new data. We want to avoid a low validation accuracy because it would indicate that the algorithm is not sufficient and effective in the sense that it is not well enough to be able to generalize on the unseen data.

Zeynep Kurtuluş
29045