(python + numpy + scipy) s1621503

Inf2B-CW2

Task 1 Report

➔ my_knn_classify:

- to compute the squared euclidean distances between the training and test vectors, a fully vectorised approach was imperative for low computation time, as follows:
    - calculate (48000, 7800) matrix of distances between each training and test vector as specified in FAQ - this was crudely done in the first attempt - however, Sourav Dey[1] provides some insight into how to directly create this matrix in a simpler way using a fully vectorised expression for euclidean distance: $(x-y)^2 = x^2 + y^2 - 2xy$; and a numpy trick for broadcasting the additions: creating a new axis when adding x^2 and y^2 to obtain a 2D array, to then easily subtract $2xy$ from it.
- to find the closest neighbours, utilise `argsort()`[2] to find their indices in the distances matrix, which will in turn correspond to the respective class in the `Ctrn` training vector class matrix
    - find the most common occurrence (mode) of class from training data cross-checked with k nearest neighbours, assign the outcome as the class prediction for the test vector

Time elapsed approx (in seconds) [DICE environment, command line]	distances	sort	total
	13.10	25.51	38.65

Statistics

- from the data collected, the best accuracy is obtained when a value of $k = 3$ is chosen
- there appear to be diminishing returns in accuracy when upwards of 5 nearest neighbours are selected, as more chaos is introduced into the system
- works surprisingly well even when considering just a single nearest neighbour

k	N	Nerrs	acc
1	7800	1083	86.10%
3	7800	1049	86.54%
5	7800	1061	86.40%
10	7800	1136	85.44%
20	7800	1228	84.26%

[1] https://medium.com/dataholiks-distillery/l2-distance-matrix-vectorization-trick-26aa3247ac6c

[2] an attempt was made at foregoing full `argsort()` in order to shave a few seconds off the sorting time, namely by performing a partial sort using `argpartition()` to obtain the indices of the k lowest values in the array without actually sorting the whole array; however, these indices would need to be used to re-fetch values from the distance matrix, to then perform another sort, which unfortunately means that their original indices would have been lost in the process, rendering them unclassifiable; because of this, the idea of partitioning was not developed further