Machine Learning 2

A1 Report

Junyoung Jung

**(a) Implement an iterative method (using for loop) to classify a single new example. Write**

**down your observations.**

As the iterative method iterates over every value in a tensor, it takes a long time to finish the calculation. A screen shot of a computer

Description automatically generated

It took almost 9 minutes and 20 seconds, when ran in a personal laptop. From the primitive approach, I observed that broadcasting is a feature unneglectable in big set of numbers.

**(b) Use the broadcasting concept you learned in the laboratory session to classify a single**

**new example. Compare against the result from (a).**

Now that broadcasting concept is implemented, the computation is exponentially faster. The time took for the computation was 0.8 seconds

A screenshot of a computer program

Description automatically generated

**(c) Now, implement a k-NN algorithm (starting with k=5) and its training/validation/evaluation**

**code to perform multiclass classification over all digits, using the implementation from (b). Write down your observations.**

As the classification involves the process of distinguishing two numbers where the data is closest to the answer label, now the KNN algorithm improves it such that it sorts the resulting distances of all the data and the answer label’s data. Then it leaves K number of labels and decides the most common data to be the answer label. For the most part, this was correct. However, although as I have increased K from 5 to 100, it remained the same answer, almost at a point where it remarks as erroneous. At K of 1000 it finally made an error, confirming that the algorithm is well constructed.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**(d) Improve the algorithm from (c) [Hint: Try to find the desirable distance function, whichcan be found by googling or going through PyTorch document].**

From “<https://www.geeksforgeeks.org/k-nearest-neighbours/>”, I could find some useful features. For example, where I used slicing and sorting methods for the classified data, the function was already implemented in pytorch as torch.topk, which sorts the tensor in ascending or descending order and leaves k amount of data in the tensor. Also, I switched from direct application of Euclidean distance to pairwise\_distance function in torch.nn.functional module. This enables easier modification of distance hyperparameter, yet, it seemed to result in a slower computation.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

As the screenshot shows, the result took 1.3 seconds. Compared to the unimproved version of KNN, it took more time.

**(e) What are the hyperparameters you can tune?**

The hyperparameter that I could tune was K and the distance function. I used the Euclidean distance for all the calculations.

**(f) Try at least two other options for each hyperparameter. Report the performance for each option.**

*Case 1*: Changing L2 norm distance to L1. The result was slower in general compared to L2. K value was intact.

A screenshot of a computer program

Description automatically generated

*Case 2*: Changing the K value along with norm to L1. The result was even slower. Yet the answers were still correct.

A screenshot of a computer program

Description automatically generated

**(g) You can try more options if you want. What is the final test accuracy?**

For all the options I tried, the ultimate aim was to shorten the computation speed. The matter of models overfit or underfit did not affect the computation speed whatsoever, at a dataset this small is probable, and the result was correct for all cases, where hyperparameters were changed as long as the K value did not increase massively.