[CV21] Assignment 1 - Di Zhuang

2.1 Implement the distance function

In distance function, the distance between x and points in X are computed. In particular, I used the None type to expand the dimension of x, and applied torch.cdist() provided by PyTorch to compute the distances. The function returns a tensor “dist”.

2.2 Implement the gaussian function

In gaussian function, the weights of the points in X are computed according to the distances computed in the distance function. As for the kernel of the “kernel density estimation” method, I chose the gaussian function. The function returns a tensor “weight”.

2.3 Implement the update_point function

In the update_point function, the updated coordinates of the point x are computed according to the weights computed in the gaussian function. In particular, the updated coordinates of point x is equal to the weighted average of the points in X. The function returns a tensor “x”.

2.4 Accelerating the naive implementation

To accelerate the computation, we may batchify our input with batch size equal to the number of points. In this way, no for loop is used in the computation. All points are processed in one go. The running time of my algorithm is around 12.587 seconds for the for-loop-based version and around 2.404 seconds for the batch-processing-based version.