<https://github.com/SanaNGU/advanced-deeplearning-course/tree/main/Practice3>

1-

T-SNE is a non-linear dimensionality reduction technique and it is good in preserving the local structure of the data but it fails to preserve distances between different neighborhoods (the distance between the clusters) .

With T-SNE representation, the cluster size does not mean much (a sparse and dense cluster in the original dimension might end up with similar-sized cluster representations in the lower dimension ) which means T-SNE doesn't care about the variance of the data, while the PCA, which is a linear dimensionality reduction technique, finds a linear projection of high data in such a way that the variance of the projected data is maximized.

This means, PCA is trying to preserve the large Euclidean distances found in high dimensional space onto the lower dimensional space , keeping data far apart. The major drawback of PCA is that PCA fails to preserve smaller distance and local structure of the data

PCA tries to preserve the global structure of data points and its main objective does not lie with preserving the relative distance between points(PCA doesn't preserve local structure) but with the overall variance along axes.

**One disadvantage of PCA** is it doesn't consider the distance between the point. This is the main advantage of t-SNE over PCA. t-SNE measures similarities between points in a high dimensional space and looks for local similarities, meaning similarities to nearby points which are much more useful.

To understand more about the differences between TSNE and PCA, I visualized the MNIST dataset in 2-D using both techniques:

<https://github.com/SanaNGU/advanced-deeplearning-course/blob/main/Practice3/TSNE-PCA%20without%20training.ipynb>

Chart, scatter chart

Description automatically generatedA picture containing chart

Description automatically generated

PCA T-SNE

From the figures above, we can realize that T-SNE results a better-clustered embedding than PCA, its worth mentioning that TSNE is not just preserving the distance (ex: put all the 1's together) but it also preserves the shape of the 1's(1's with a similar shape close to each other )

2) Consider three points a, b and c in the high-dimensional space. Let us assume that a and b are very close to each other, and that c is very far away from both of them. How would the loss behave if the following is true for the low-dimensional space? Feel free to use dummy numbers to support your answer.

**• a, b and c are all close to each other.**

**• a and b are close to each other, and c is far away from them.**

**• a, b and c are all far away from each other.**

**• a is far away from both b and c, that are close to each other.**

Since T-SNE care about local structure ( so what it actually cares about is the distance between a, b in low dimensional space compared to their distance in the high dimensional space, so the loss will be small in case 1 and 2 since a and b is close to each other in both high dimensional space and low dimensional space

For the third case, TSNE does not actually care about c but the loss will be high because a and b are far from each other in the low dimensional space

The last case also should be high because this doesn’t preserve the local structure.

To do the calculations, we must first understand how TSNE projects the points in the low dimensional space and how it preserves the local structure.

TSNE represents the high dimensional data by creating probability distribution (P) where the probability of similar points to be picked is high and the probability of dissimilar points to be picked is low. Then it creates a low dimensional space probability distribution Q which preserve the P probability as close as possible.

The cost function in the case of TSNE is the Kullback-Leibler divergence (which is in this case equal

to the cross-entropy up to an additive constant). The cost function is given by the formula: