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

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 )