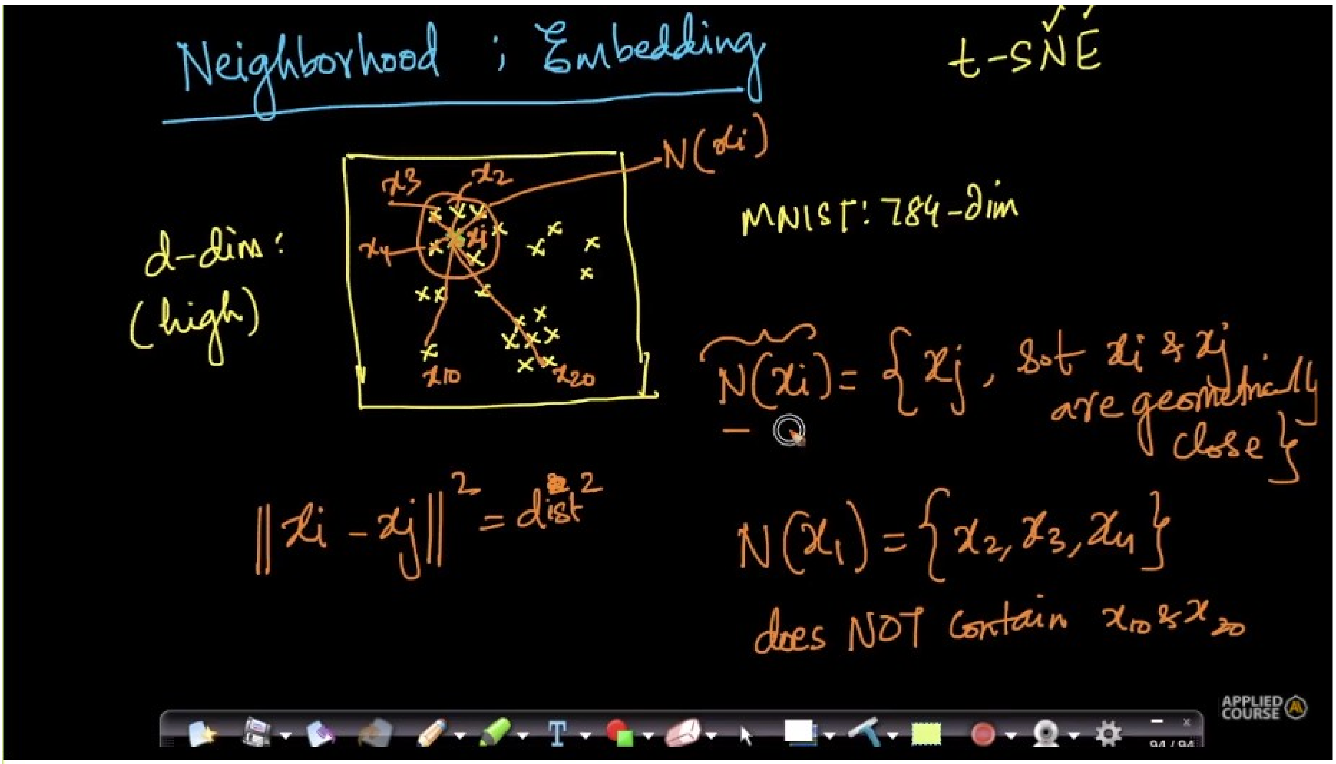
NOTES

TSNE:

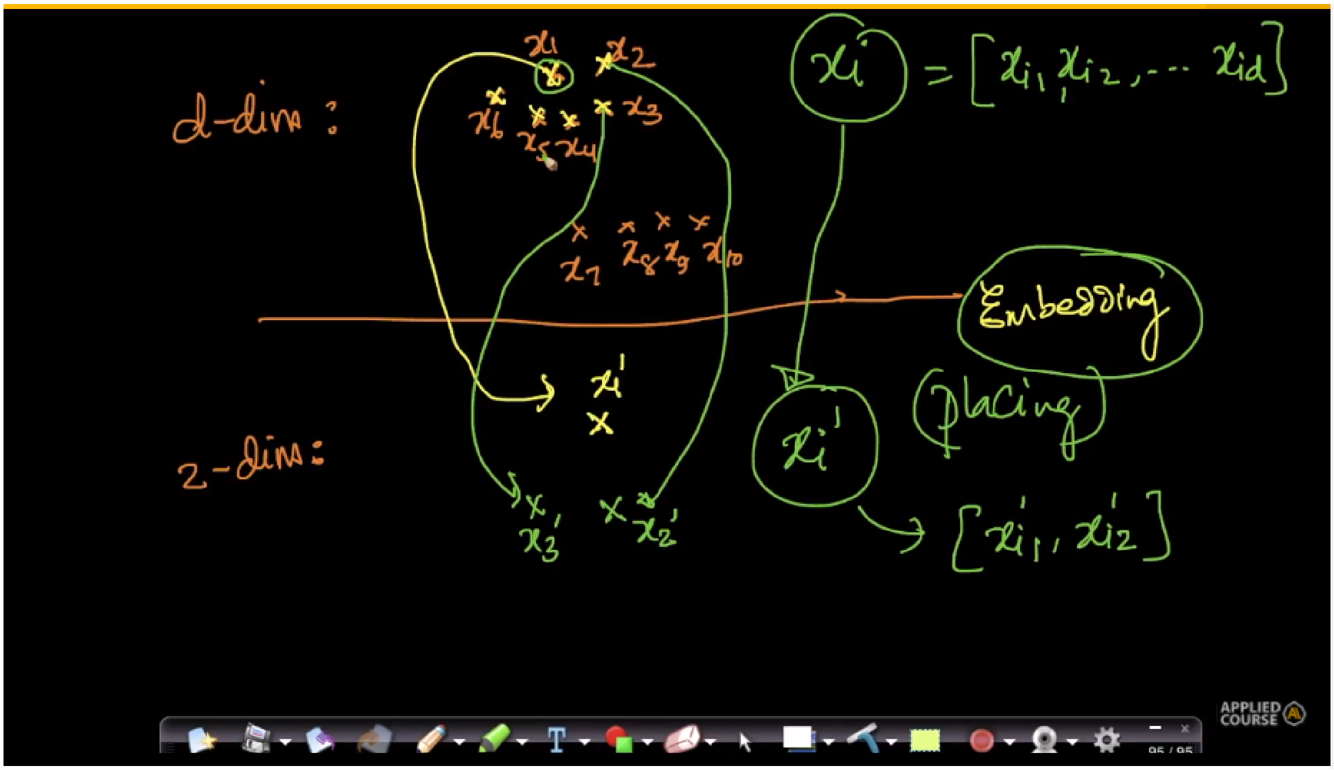
TSNE tries to preserve the global structure of the data.

Neighborhood of a point, Embedding

Neighborhood: here x1, x2 and x3 are the neighborhood od the point xi and x10, x20 are the far away points.



Embedding: creating the point in low - dim from d-dim space is called embedding.

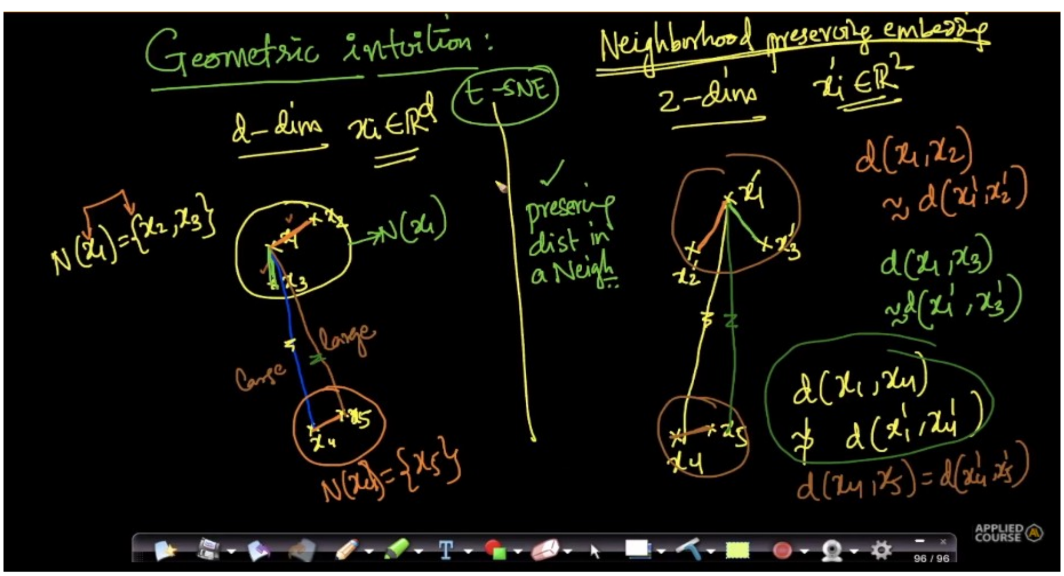


Geometry:

The distances of point are preserved by embedding into low-dim space from d-dim space.

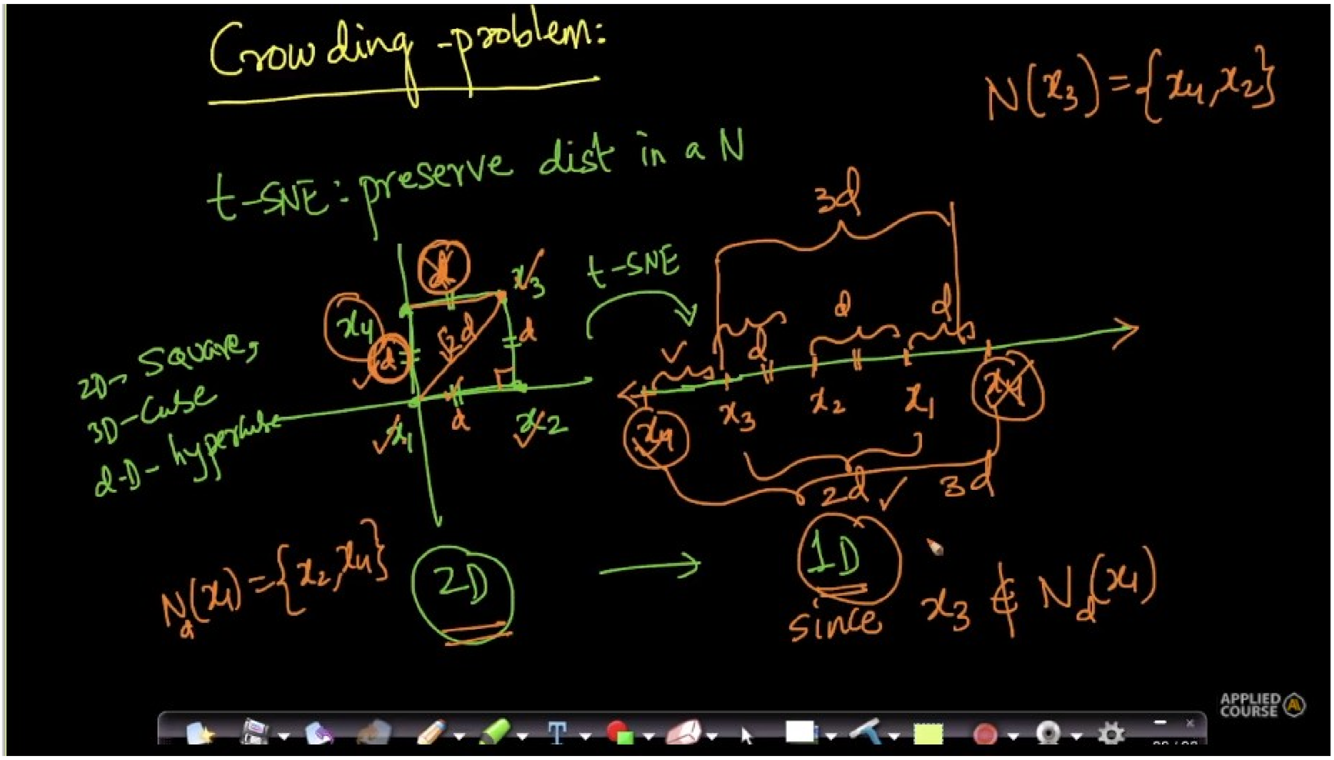
The actual distances of the points in the d-dim are not same as the low-dim space.

TSNE is the neighborhood preserving algorithm.

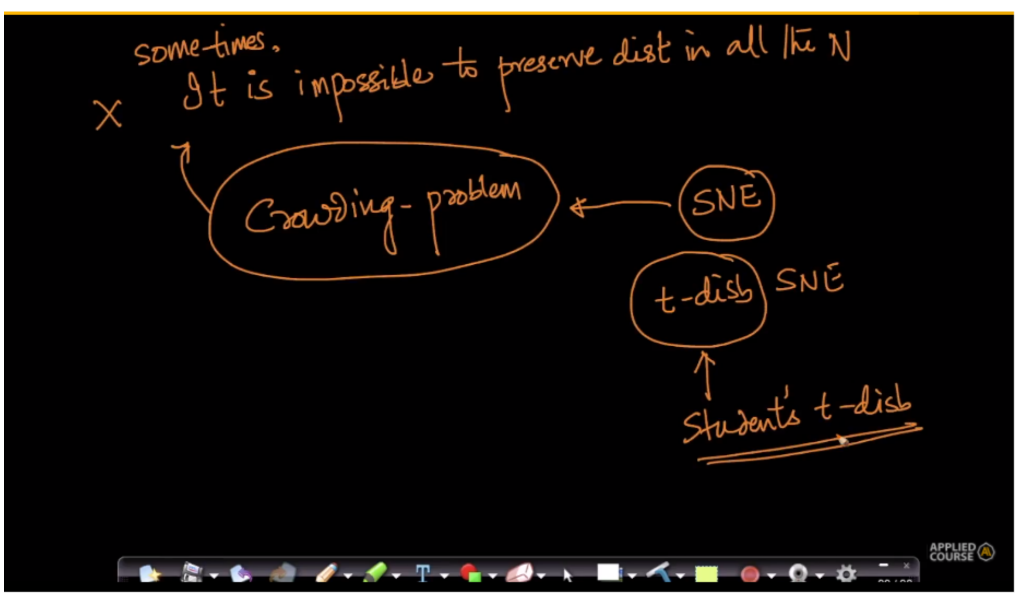


Crowding problem:

Here x3 is not in the neighborhood of x1 at a distance of “d”.



Sometimes, It is impossible to preserve all the distances of the data points. This is called **Crowding problem.** In order to overcome the crowding problem. T-distributed SNE was proposed where in which the algorithm tries its best to minimize the **crowding problem** seen above.



Applying t-SNE:

https://distill.pub/2016/misread-tsne/

Do not stick to one value of **perplexity** and **number of iterations.** Trying various values to achieve the best parameters. **T-SNE** is the iterative algorithm not deterministic.

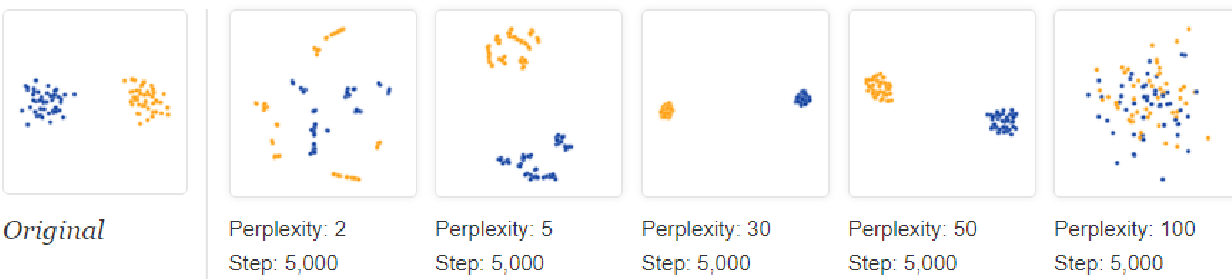
NOTE: After achieving the stable visualization of the data. The T-SNE interpretation does not change with the increase in number of iterations. **More the number of iterations the better is the interpretation.**

Perplexity: Here the perplexity can be taught of **number of neighbors.**

Various interpretations:

Case - 1

- As perplexity increases the shape gets more sensible.

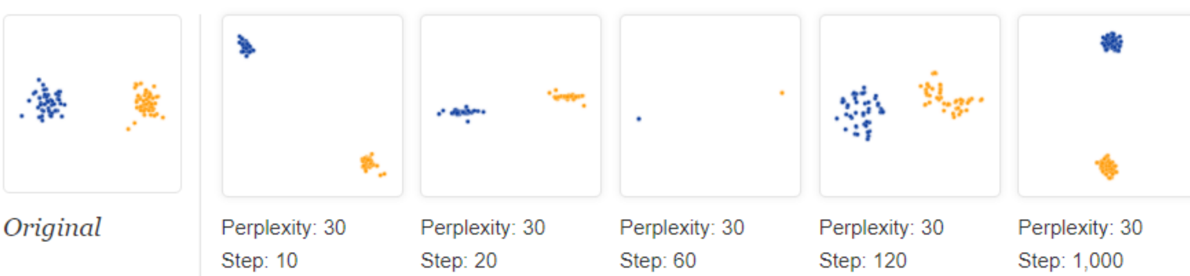


If the **perplexity** is equal to **number of data points** it tries to preserve complete dataset.

- Choose the **perplexity** less than the **number of data points.**

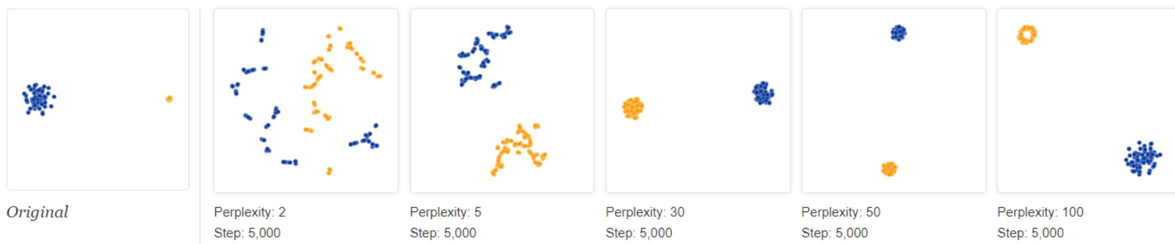
Case – 2

- Always run the **T-SNE** for various values of **iterations.**
- T-SNE is the **stocastic algorithm** it does not give the same plot for the same values of perplexity and number of iterations multiple times.



Case-3

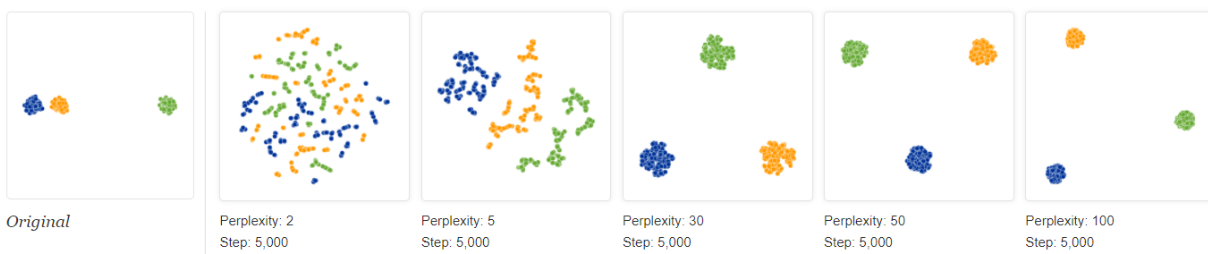
- **T-SNE** expands dense clusters and shrinks the sparse clusters. We cannot come to the conclusion that the data and the **T-sne** plot has the same density as the algorithm is stocastic.



- T-sne does nothing with dense clusters.

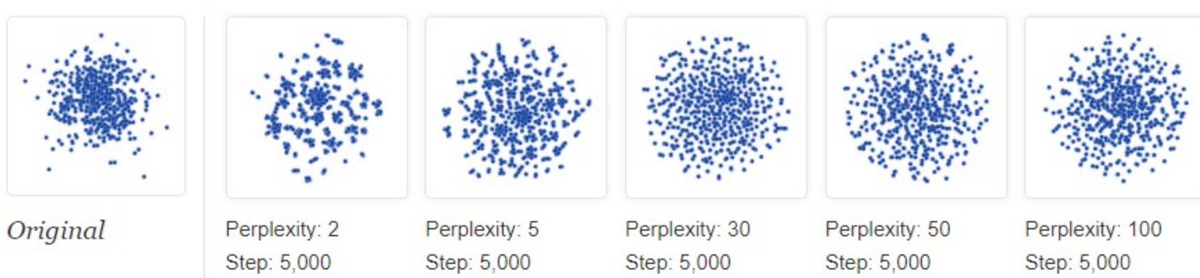
Case – 4

- The predefined distances between the clusters is not preserved by T-sne.



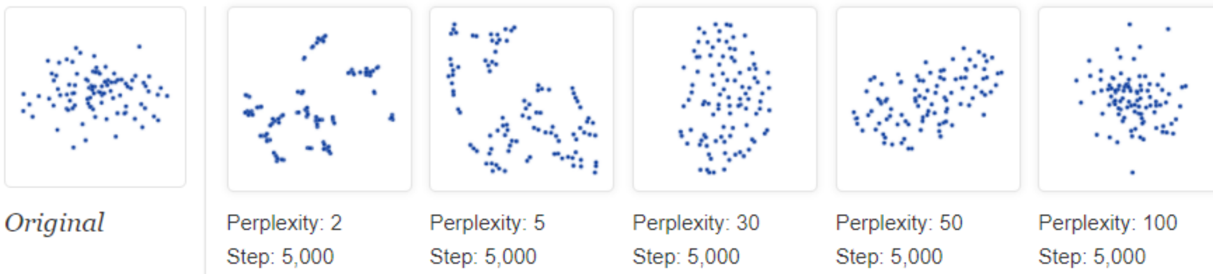
Case – 5

- For random data apply T-sne for various values of perplexity and get sense of the data.
- **Don't come to conclusion for one value of perplexity in case of random data though the clusters are formed for low perplexity which is EXTREMELY DANGEROUS**



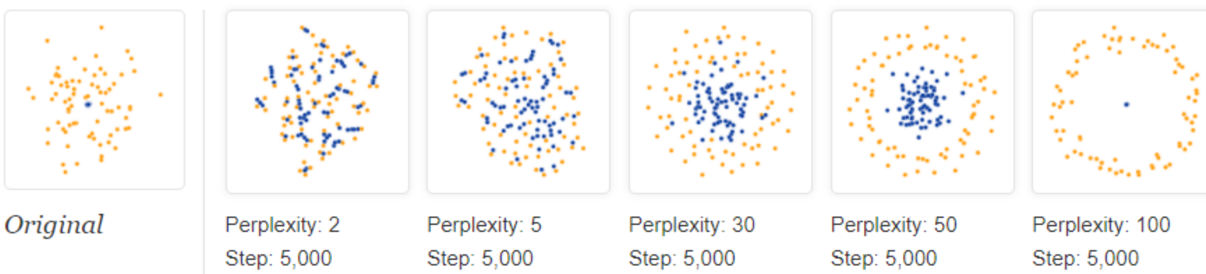
Case – 6

- The same situation for the data as **case-5** in case the data have some **shape** which is **EXTREMELY DANGEROUS** situation as in case – 5.



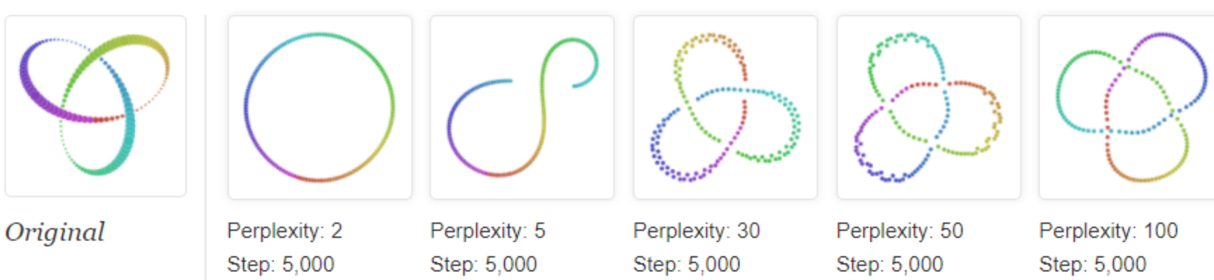
Case – 7

- Always run with various values of perplexity and number of iterations.
- Tsne does not replicate the results.



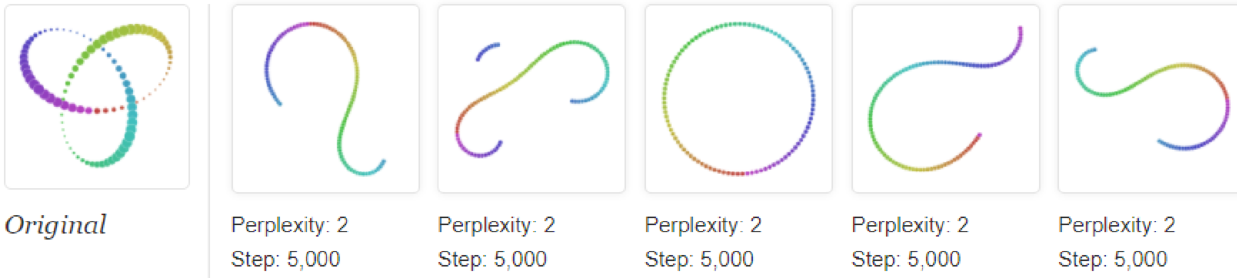
Case – 8:

- Same topology can be achieved for the various values of perplexity. **Do not stick to one perplexity.**

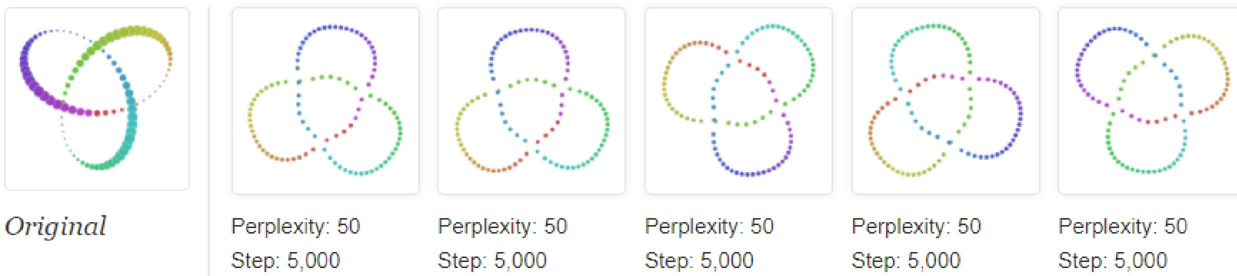


- As the perplexity increases the shape is achieved.

Case – 9



Five runs at perplexity 50, however, give results that (up to symmetry) are visually identical. Evidently some problems are easier than others to optimize.



- From the plots we can infer that do not run tsne for one value of perplexity. The first plot gives different types of plots for same perplexity and number of iterations on running the algorithm various times.

TSNE clustered the points together which looks visually similar.