山东大学 计算机科学与技术 学院

可视化技术 课程实验报告

学号：201900130176	姓名： 李伟国	班级： 智能
实验题目：		
实验学时：	实验日期：	2021/10/16

实验目的：

对 mnist 数据分析不同的参数对 t-sne 结果的影响（60k 的数据量，只是用其中的 train set 就可以）

硬件环境：

处理器：AMD Ryzen 5 3600 6-Core Processor 3.60 GHz

Ram 16.0 GB

软件环境：

实验步骤与内容：

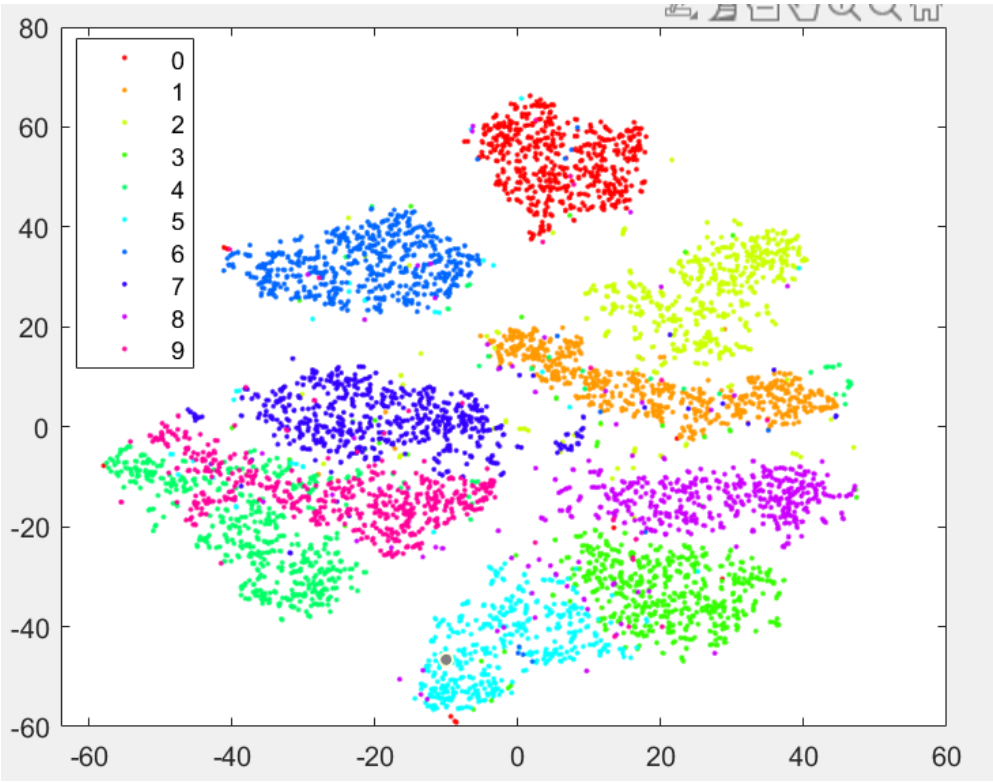
数据的基本信息

测试的数据时 MNIST 是手写识别的图片的集合，每个图片是 28*28 的大小。并且每个像素是单通道的灰度图。

将这 28*28 的每张图片 strench 成一个个的列向量，因为是单通道的图片，所以每张图片的维度都是 28*28 = 784，即维度是 784 维度的，现在采用 t-sne 方法进行降维。

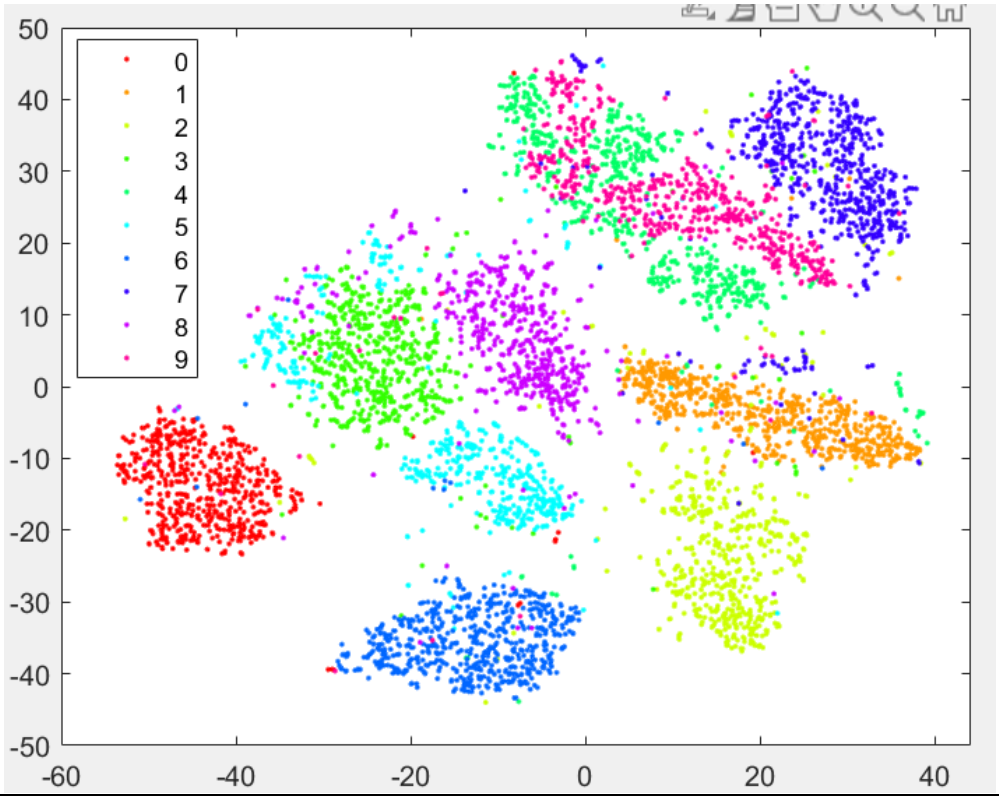
由于数据较多，为了达到测试的目的，这里仅仅使用了 6000 个样本，随机抽选 6000 个样本，做为本次实验的数据集。

下面是 matlab 自带的 T-sne 函数对该数据集降维后的效果



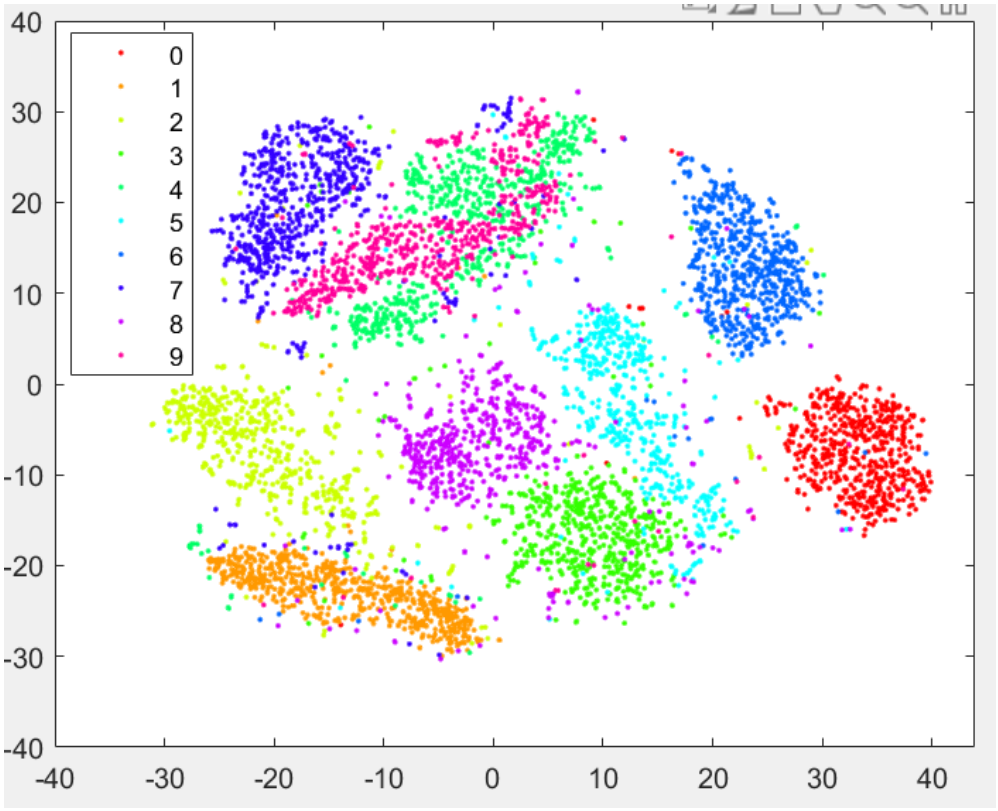
默认参数的情况（peplexity = 30）

通过对上面的降维后的结果可视化后，可以非常容易的发现，几乎每个数字的类别都被分隔的非常好。仿佛达到了一种 聚类的效果，也就是说，在高维空降的聚类在低维度的空间中任然被保留了。但是 peplexity 只有 30 ，并不能对拥有 6000 个数据点的集合有较好的 global structure。所以下来来调整 peplexity 的值。
将 perplexity 调整为 50

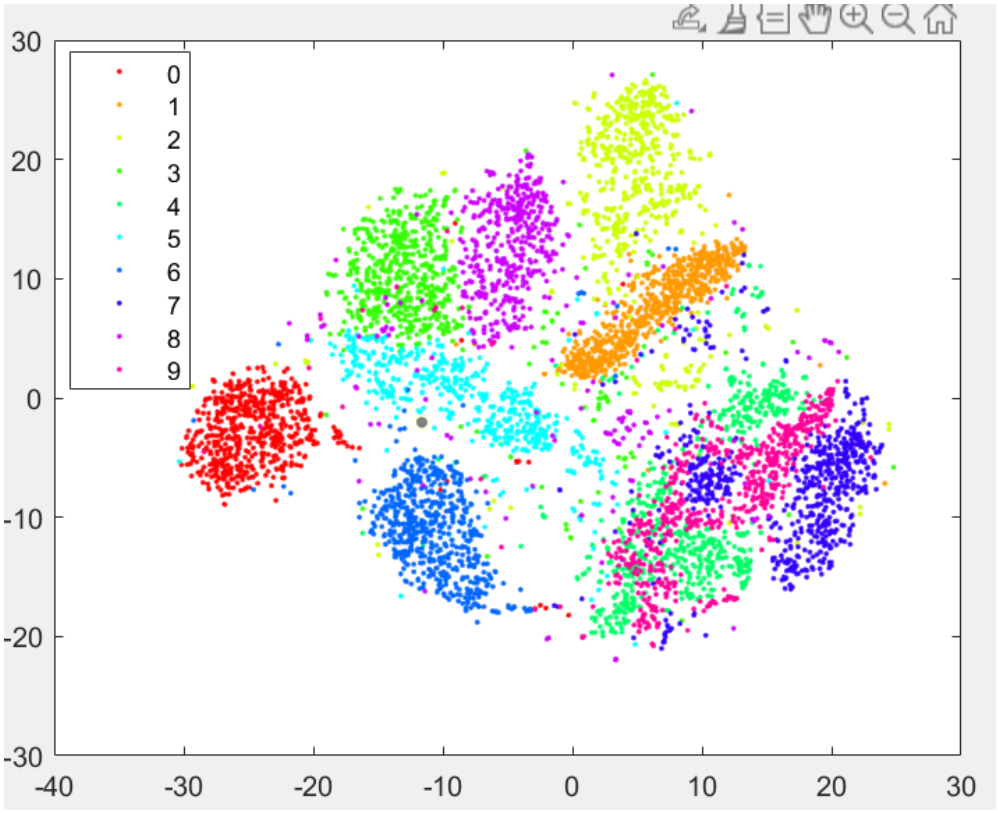


Perplexity = 50

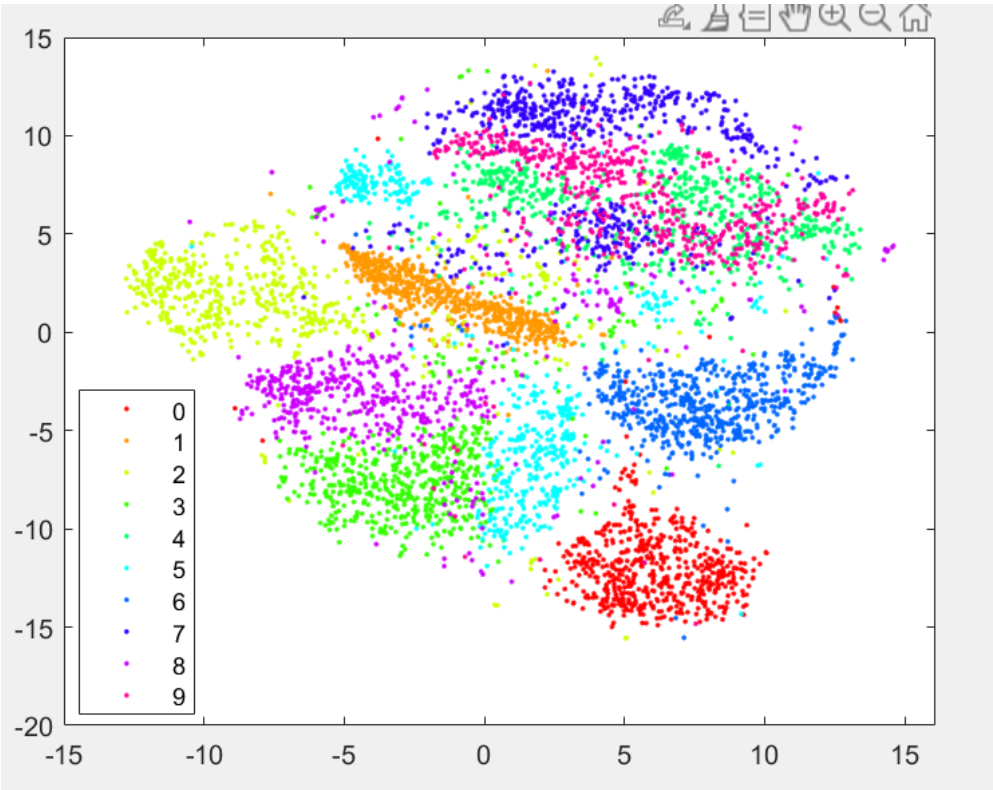
```
>> Y = tsne(origin_data,'Perplexity',100);
```



Perplexity = 100



Perplexity = 200

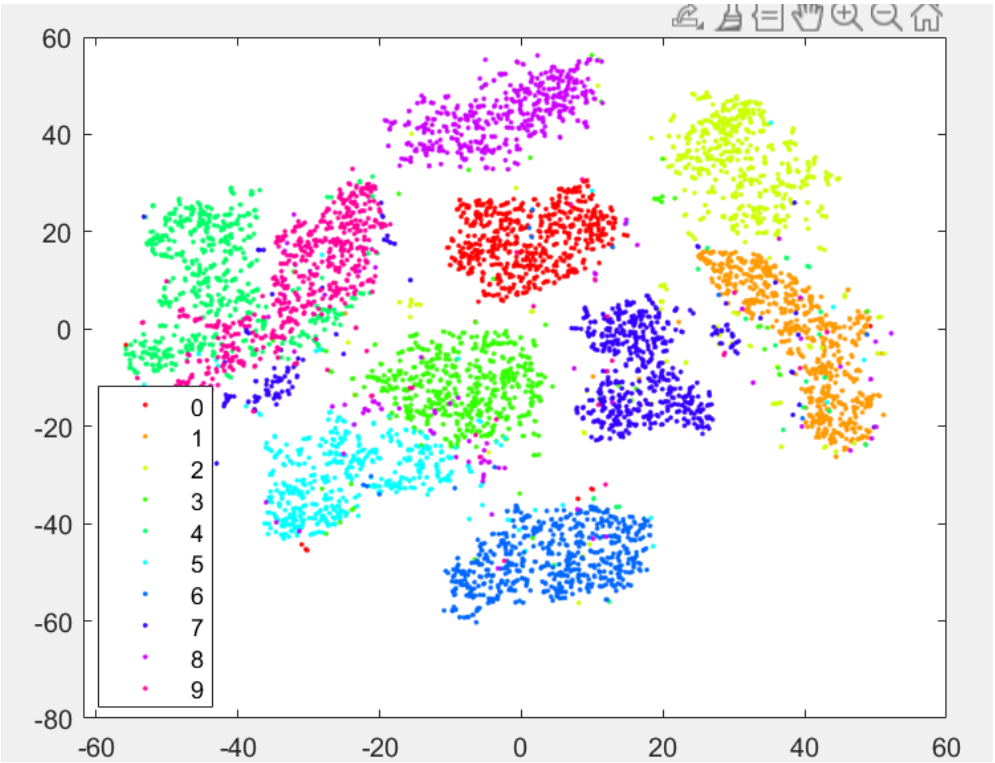


Perplexity = 500

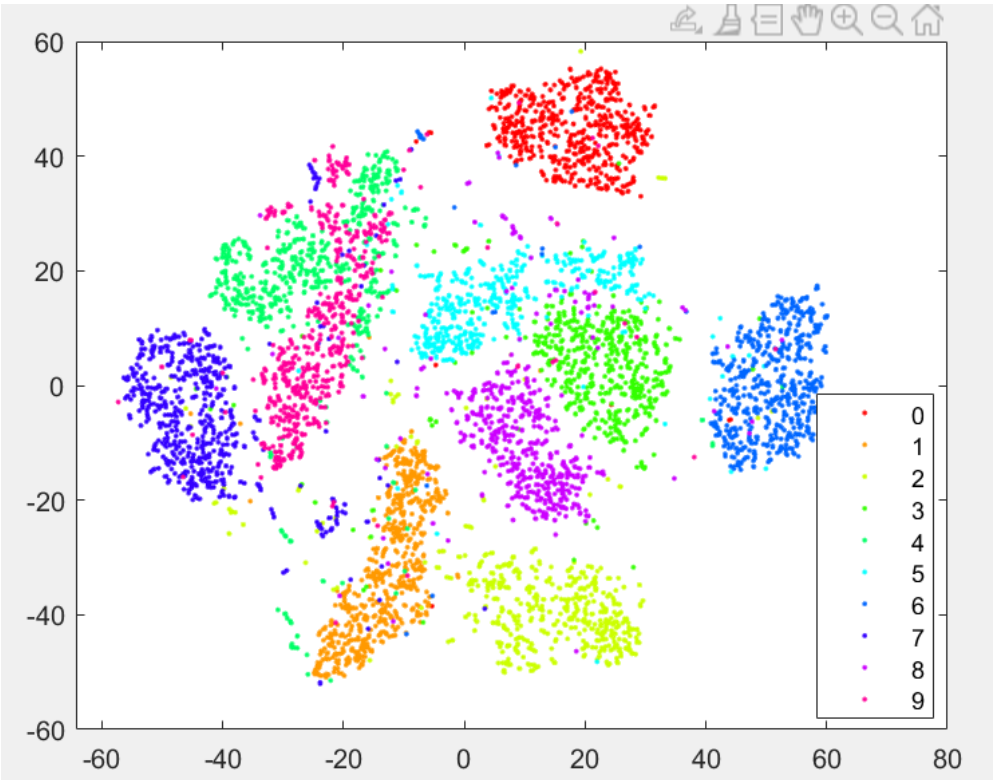
通过调整不同的 perplexity ，我们可以发现，在 perplexity 很大的时候，在 embedding 中其 cluster 的 效果远远不如在默认情况下即 perplexity = 30 的情况下的效果，这是因为，perplexity 越大，所要在低维空间中保持的点与点之间的距离的数目就越多，也就是说要保持原来高维的数据越多，导致降维后的效果会越复杂，就会导致类与类之间的空间会很小，会更加的紧凑。但是，如果 perplexity 设置的很小，类与类之间就不能很好地分离开来

Exaggeration parameter：这个参数增加在连个点之间的吸引力并且允许点移动的更加的自由。默认的情况下，Exaggeration = 4（在 matlab 中是这样），该参数用来指定在数据集上的 natrual cluster。 Exaggeration 设置的越大会让 tsne 学习到跟多的联合概率分布，那么 tsne 也就更能将不同类之间跟清晰的分开。Tsne 在前 99 次迭代中使用这个参数.

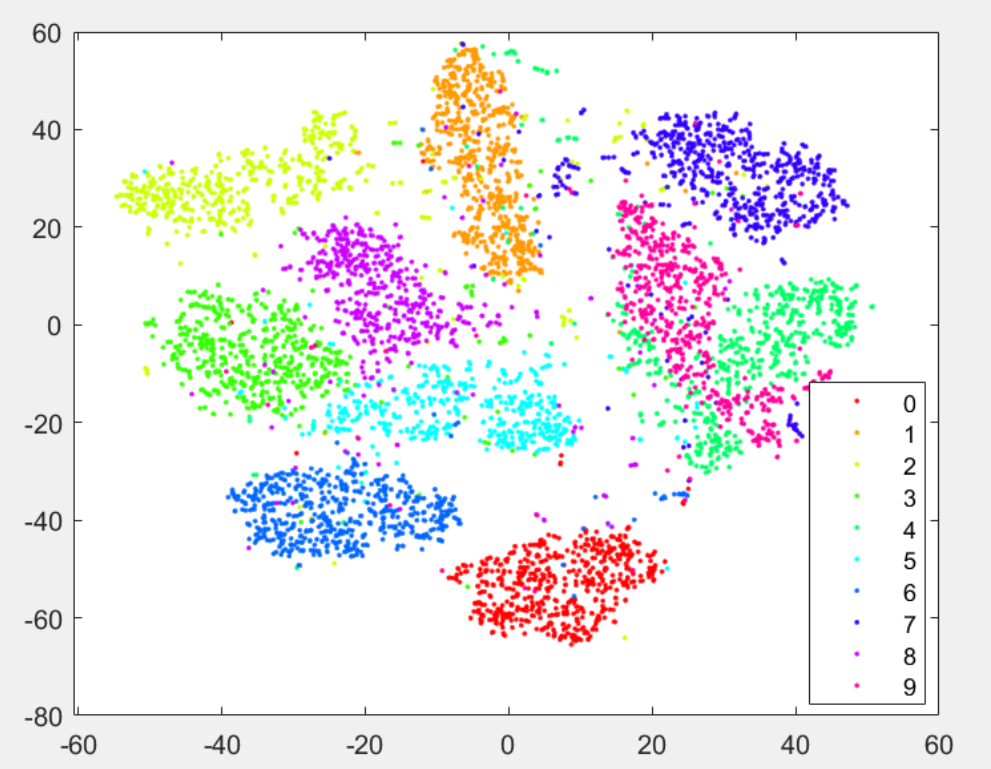
由于 Mnist 训练集上有 10 种类型不同的 cluster，所以不妨将 exaggeration 设置为 10，perplexity 任然设置为 30.



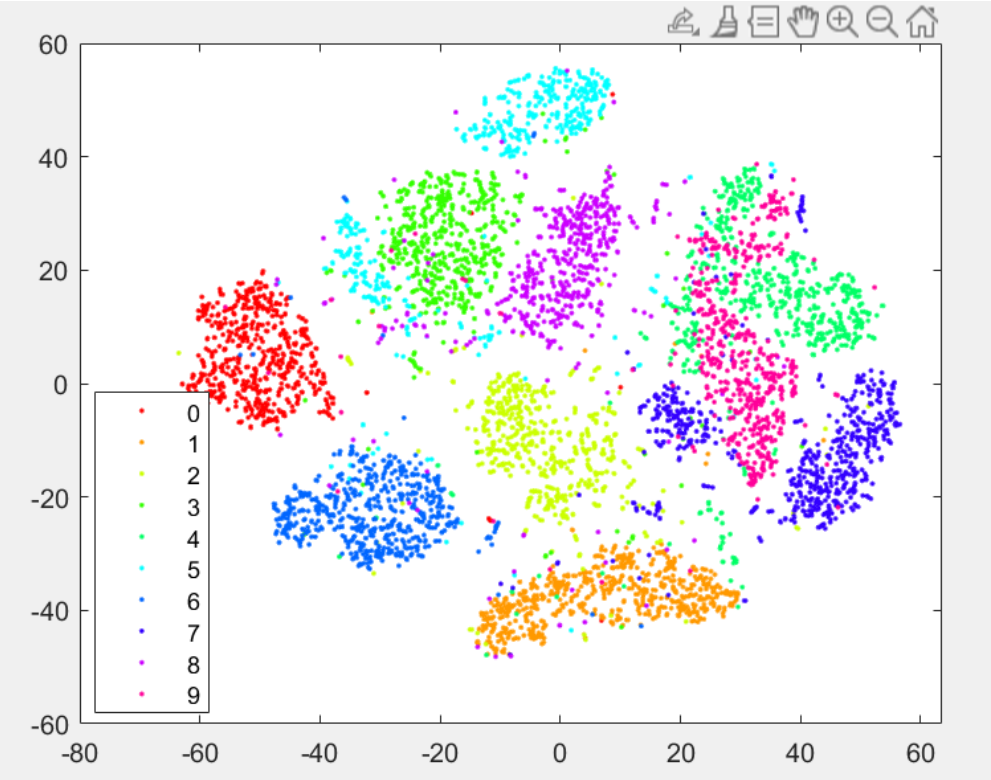
Exaggration = 2



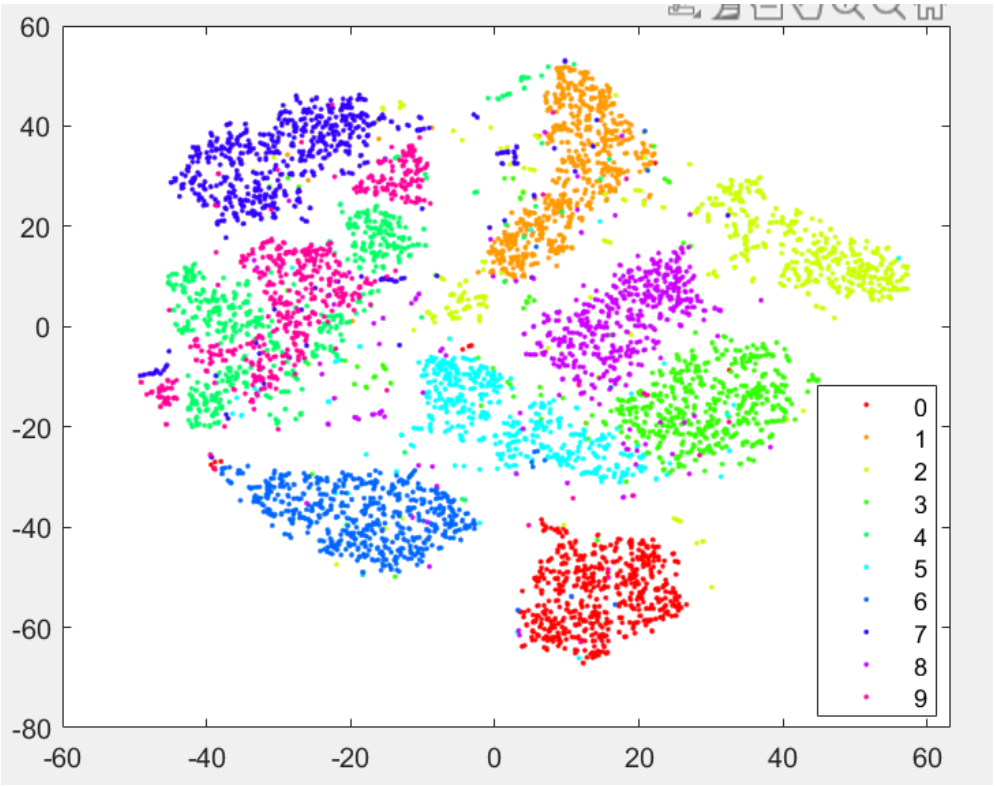
Exaggeration = 10



Exaggration = 12



Exaggeration = 15



Exaggeration = 20

通过对上面不同的 exaggeration 的设置，可以发现，其在 embedding space 中的位置在改变，该参数设置的越大，那么在 embedding space 中各个类之间的空隙也就越大，在将 exaggeration 设置为 2 的时候，可以发现,数字 7 这个 clusters 被分成了 2 块.而随着 exaggeration 的增大，这种情况有被缓解。

深层原理

1）

t-sne 使用高斯分布来衡量两个点的相似性，以要测量的点在为中心（该点称为 point of interest），计算每一个点与该点的距离，并且在高斯分布的概率密度函数上计算出每个点相对于该点的值，直觉告诉我们，越近的点，它们彼此之间就有又更大的相似性。

similarity：$p_{j|i}=\frac{\exp(-||x_i-x_j||_2/2\sigma_{2i})}{\sum_{k\neq i}\exp(-||x_i-x_k||_2/2\sigma_{2i})}$，（以 i 点为 interest，衡量 j 点的相似性），这个式子表示的是经过归一化后的结果，分子相当于是标准的高斯分布的函数值，输入是两个点之间的欧式距离。由于对 J 点作为 interest ，i 点对其的 similarity 并一样（因为，对 i 点和 j 点来说，它们个周围点密度并不是同等程度的，这体现在标准差上），所以用同样的方式计算 P i|j，通过将二者平均一下，作为 j 点和 i 点彼此的 similarity，即：$p_{ij}=\frac{p_{j|i}+p_{i|j}}{2}$。

在 embedding space 中，使用 t-distrubted，该分布函数具有中心点对应的值比

Gaussian 要低，但是尾巴比 Gaussian 要高一点。可以在 embedding space 中计算出两个点的相似度 Pij

不妨分别在 original space 和 embedding space 中根据计算出啦的相似性构造 matrix Q 和 P，t-sne 的目的就是让 Q matrix 和 P matrix 很相似。

通过使用 KL 做为 cost function ，然后用 gradinet descent 的方法来优化损失函数。

还有一点需要做的就是以每个 data point 为中心的 Gaussian function 的 σ 应该怎么设置。这里 t-sne 引入了 perplexity 这个 hyperparameter，perplexity 可以被认为是对最近的 k 个邻居的连续模拟。σ 被设置成可以包含这 k 个邻居的值。这 k 个邻居的值可以认为的设定，是试出来的，有一个最适的取值。即 perplexity 就是在 local 和 global 的一种 tradeoff（权衡），会在 embedding space 空间中保留这 k 个 neighboor 的距离。如果要考虑的邻居越多，那么 Gaussian function 的分布越广才行，即标准差就越大，and vice the visa。 Perplexity 可以认为是一个点附近的有效的近邻点的个数，tsne 对 perplexity 的调整比较有鲁棒性，通常选择 5-50 之间。给定之后用二分搜索寻找合适的 σ 。

在原始的空间中使用 Gaussian 分布，在 embedding 中使用 t – distributed 。

2） tsne 的不足之处

Crowing 问题（tsne 在 embedding space 中使用 t 分布来避免 crowding 问题）

拥挤问题就是说各个簇聚集在一起，无法区分。比如有一种情况，高维度数据在降维到 10 维下，可以有很好的表达，但是降维到两维后无法得到可信映射，比如降维如 10 维中有 11 个点之间两两等距离的，在二维下就无法得到可信的映射结果(最多 3 个点)。

假设数据点在高维度空间中时均匀分布的，intuitively， 那么以第 i 个点为中心的附近的点，在离其越远的地方，点的数量会越多，与到 i 点的距离分布及其不均衡，如果直接将这种距离关系保持到低维，就会出现拥挤的问题

加速:四叉树的加速 Barnes-Hut

3）注意事项

Tsne 具有随机性：每次的实验结果都可能不一样高

Intrisic：如果数据经过 t-sne 后再 2D 平面上的效果不好，可能并不是算法不好，而是数据本身的内在结构不足以在 2D 平面上表示

结论分析与体会：

附录：程序源代码