**Homework# 8**

**0756079 陳冠聞**

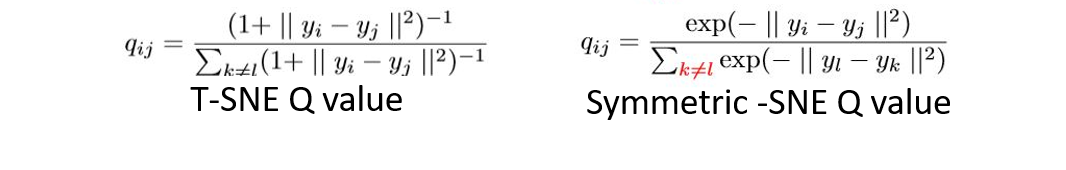
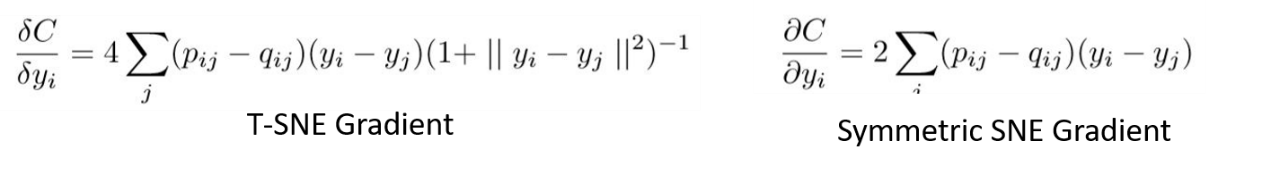
1. **Symmetric-SNE**

t-SNE 與 symmetric SNE的差異為 low-dim similarity Q的計算，

symmetric SNE 採用與 high-dimension 一樣的高斯分布，

t-SNE 採用與 t-student 分布，因此將 t-SNE 修改為 symmetric SNE 主要需要修改 (1) 計算Q (2) 計算Gradient

公式如下圖

1. 計算 Q value
2. 計算Gradient

修改的 code 如下

1. 計算 Q value

C:\Users\nilab\Desktop\擷取.PNG

改成

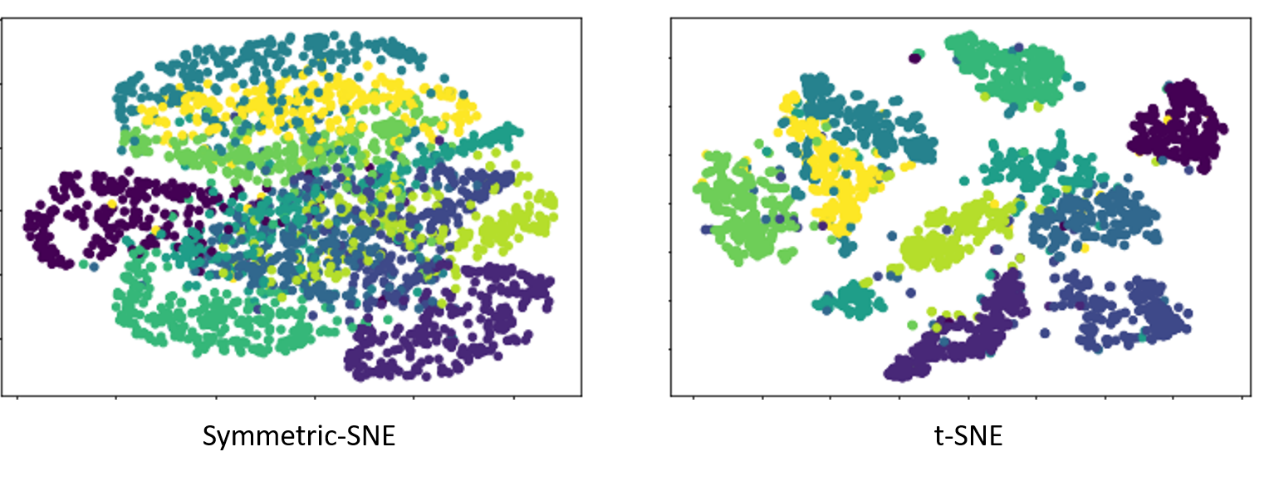
C:\Users\nilab\Desktop\擷取.PNG

1. 計算 Gradient

改成

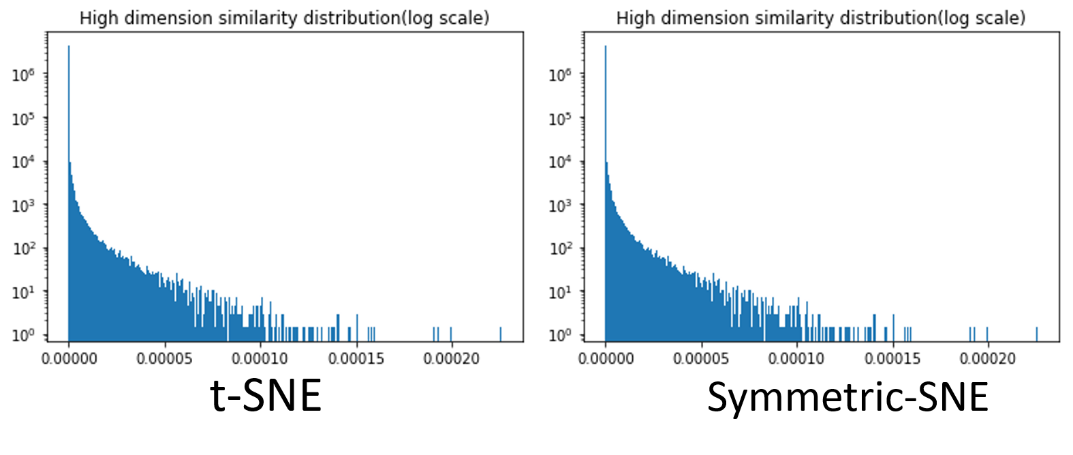
1. **Visualize the embedding**

使用 default parameters (initial\_dims=50, perplexity=30.0)

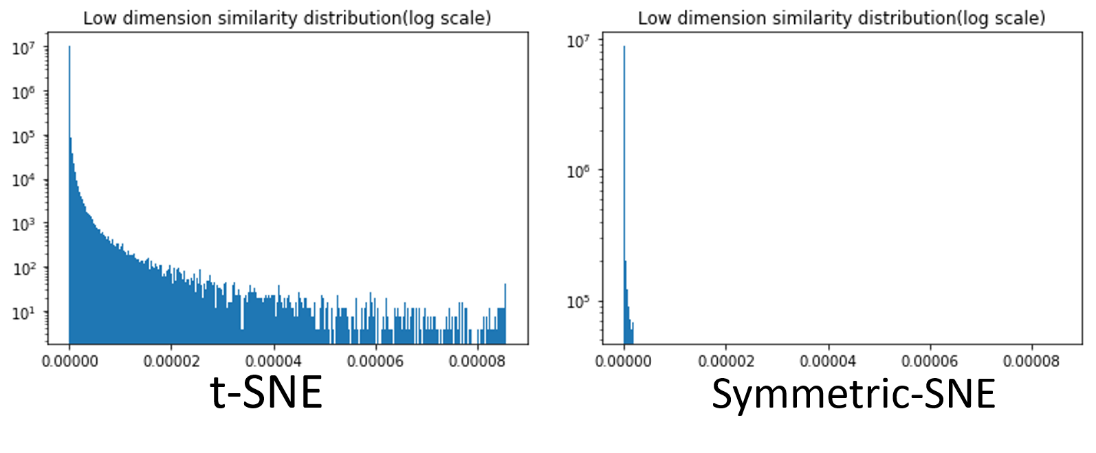
可看出比起 Symmetric-SNE, t-SNE的crowded problem改善了很多，因為t-SNE使用的 t分布相較於常態分佈，對於高相似度的點在低維空間的距離較小，而對於相似度低的點在低維空間的距離則較大，因此可以減少 crowded problem的問題。

1. **Visualize the distribution**

High-dimension

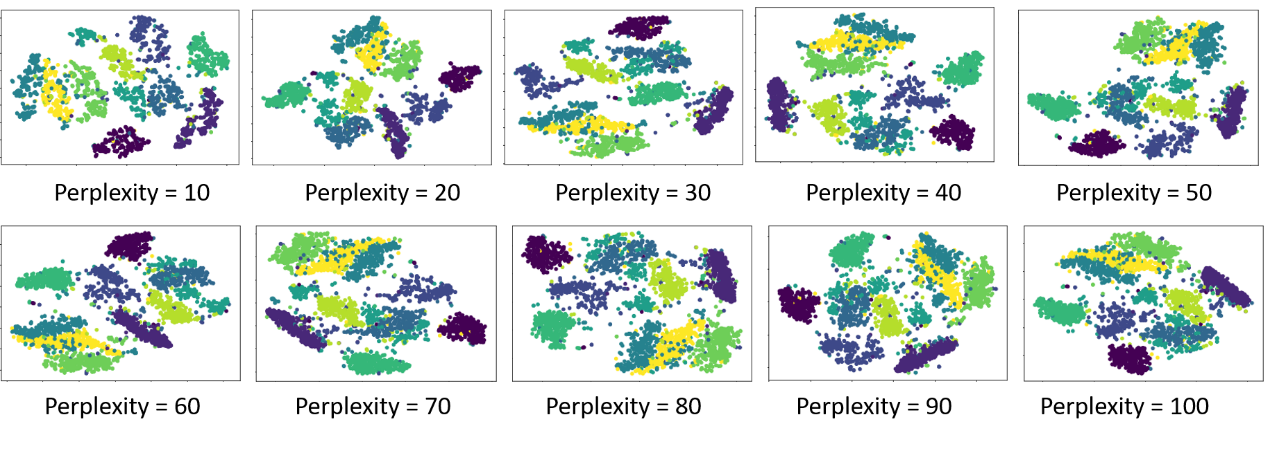


low-dimension



從上圖可以看出 t-SNE 相比 Symmetric-SNE，在 low-dimension 的 similarity distribution 的 range 較大，這是因為 t分布比起常態分布較為寬矮，而這也是為何 t-SNE 較能減輕 crowded problem的原因。

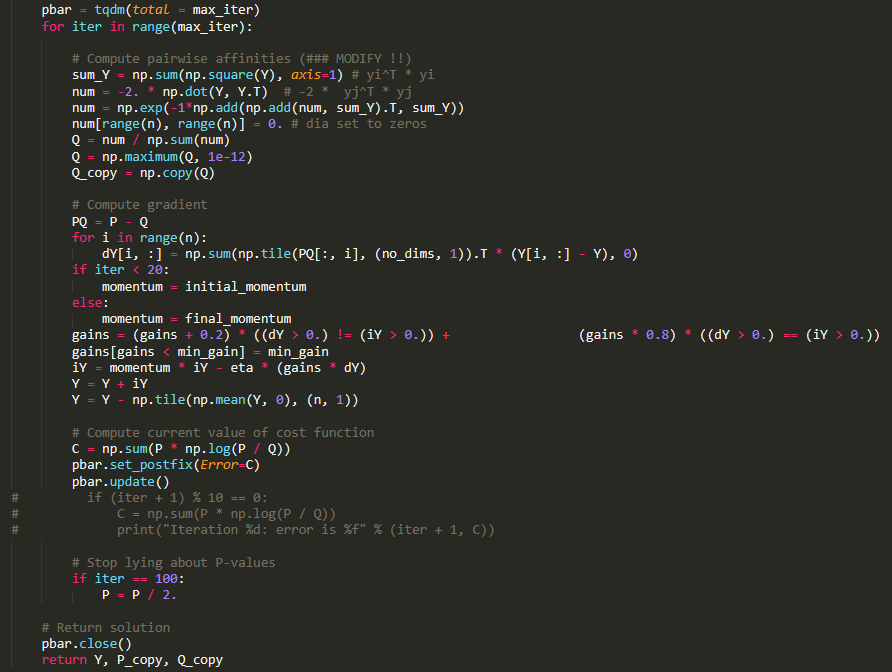
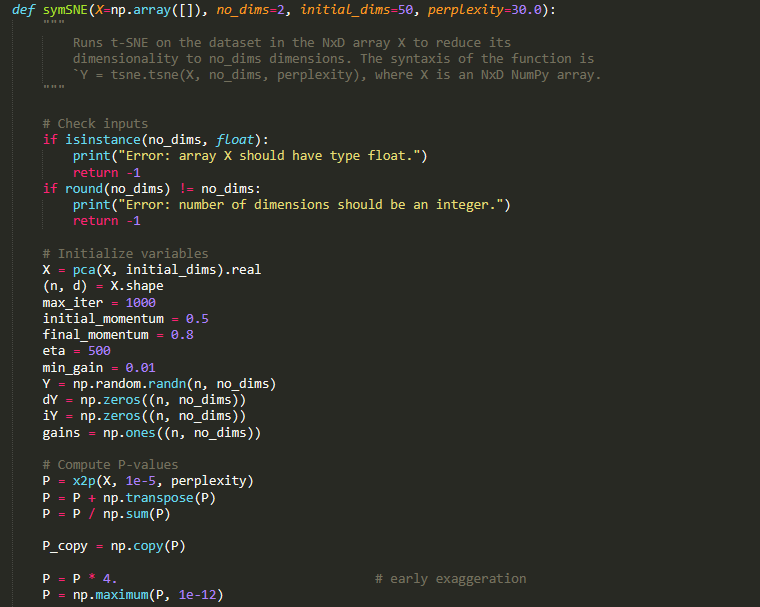
1. **Diﬀerent settings of perplexity**

Perplexity 決定了 effective nearest neighbors，在本次作業中實驗了 t-SNE Perplexity 從 10 到 100（間隔為10），結果如下

可看到當 Perplexity 逐漸變大時，會逐漸有 crowded problem，因為當依賴的鄰居變多時，很容易造成彼此分不開的情況。

1. **Code Detail**

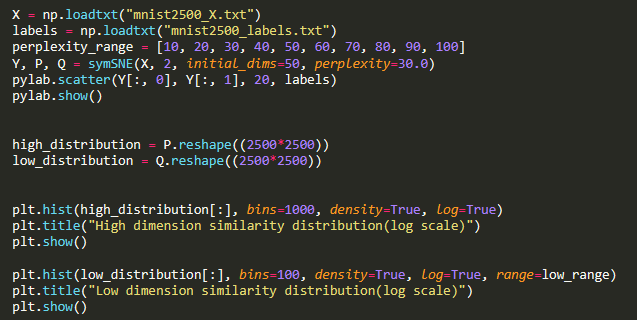
只放上與原 reference 不同的程式碼，其他詳見 SNE.py

****SymSNE 的 function

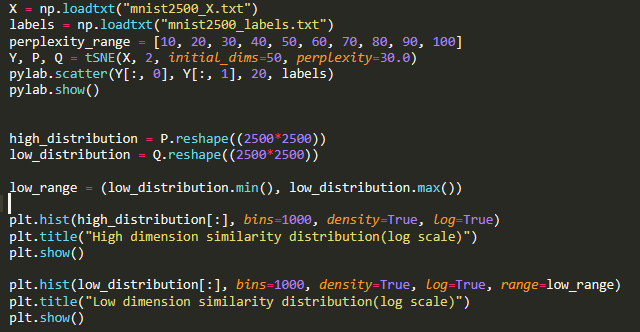
回傳 similarity matrix 方便畫distribution

與 t-SNE 相異處

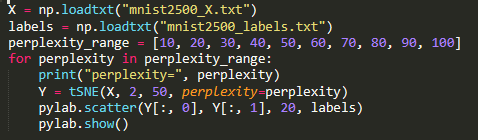
畫t-SNE high-dim及low-dim 的 similarity distribution

****

畫sym-SNE high-dim及low-dim 的 similarity distribution

****

畫不同perplexity 的t-SNE

****