展示图像分类精度图

这里用t-sne来降维

Inception-v3在倒数第二层是2048维的，现在我有2048维的特征向量，将它降维到平面空间上去。

http://www.datakit.cn/blog/2017/02/05/t\_sne\_full.html

t-distributed stochastic neighbor embedding

在高斯空间中获取一组点，并在低维空间通常是2D平面上找到这组点的表示，非线性降维算法。

SNE：是通过仿射变换将数据映射到概率分布上，

1. SNE构建一个高维对象之间的概率分布，使得相似的对象有更高的概率选择，而不相似的对象有较低的概率被选中。
2. SNE在低维空间里在构建这些点的概率分布，使得这两个概率分部之间尽可能的相似。

这个算法只能处理数据，不能训练数据之后再处理。

SNE原理推导证明：

。。。。。。

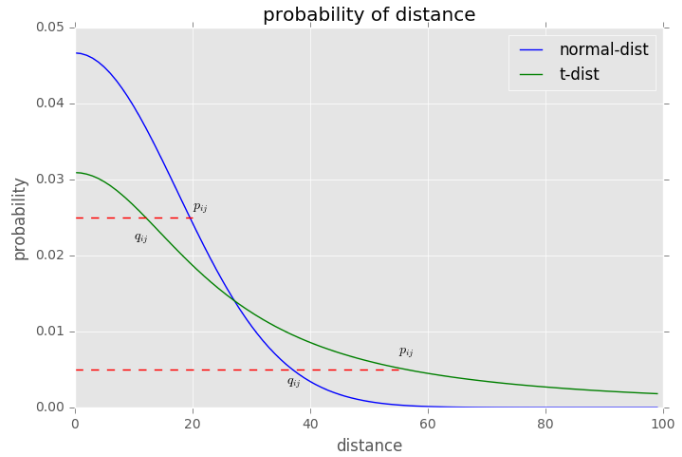
SNE提出了很好的可视化方法，但是它很难优化，而且存在拥挤问题。

t-SNE：使用对称版的SNE，简化梯度公式

低维空间下，使用t分布替代高斯分布表达两点之间的相似度

拥挤问题：各个簇聚集在一起，无法区分。

对称SNE实际上在高维度下另外一种减轻”拥挤问题”的方法：在高维空间下我们使用高斯分布将距离转换为概率分布，在低维空间下，我们使用更加偏重长尾分布的方式来将距离转换为概率分布，使得高维度下中低等的距离在映射后能够有一个较大的距离。



T分布是低相似度，可以看到于较大相似度的点，t分布在低维空间中的距离需要稍小一点；而对于低相似度的点，t分布在低维空间中的距离需要更远。这恰好满足了我们的需求，即同一簇内的点(距离较近)聚合的更紧密，不同簇之间的点(距离较远)更加疏远。

算法过程：

1.Data：

2.计算cost function的函数：困惑度

3.优化参数：设置迭代次数T，学习速率n，动量

4.目标结果是低维数据表示

5.开始优化

6.结束

不足：

1. 主要用于可视化，很难用于其他目的。比如测试集合降维，因为他没有显式的预估部分，不能在测试集合直接降维；比如降维到10维，因为t分布偏重长尾，1个自由度的t分布很难保存好局部特征，可能需要设置成更高的自由度。

2. t-SNE倾向于保存局部特征，对于本征维数(intrinsic dimensionality)本身就很高的数据集，是不可能完整的映射到2-3维的空间

3. t-SNE没有唯一最优解，且没有预估部分。如果想要做预估，可以考虑降维之后，再构建一个回归方程之类的模型去做。但是要注意，t-sne中距离本身是没有意义，都是概率分布问题。

4. 训练太慢。有很多基于树的算法在t-sne上做一些改进

Sk-learn:

t-distributed Stochastic Neighbor Embedding

t-SNE是高维数据可视化的工具。它将数据点之间的相似性转换为联合概率，并试图使低维嵌入和高维数据的联合概率之间的Kullback-Leibler散度最小化，t-SNE具有非凸的成本函数，即具有不同的初始化，我们可以得到不同的结果。

（非凸函数，即布局最优解非全局最优解，最明显的就是多峰函数）

如果数据特征很高，那么就强烈要求去使用另一个维度降低的方法，比如说密集数据的PCA和稀疏数据的随机梯度下降，将数据的维度降低到一个合理的范围。这样可以抑制一些噪声，加快样本间成对距离的计算。

class sklearn.manifold.**TSNE**(n\_components=2, perplexity=30.0, early\_exaggeration=12.0, learning\_rate=200.0, n\_iter=1000, n\_iter\_without\_progress=300, min\_grad\_norm=1e-07, metric=’euclidean’, init=’random’, verbose=0, random\_state=None, method=’barnes\_hut’, angle=0.5)

https://yq.aliyun.com/articles/70733