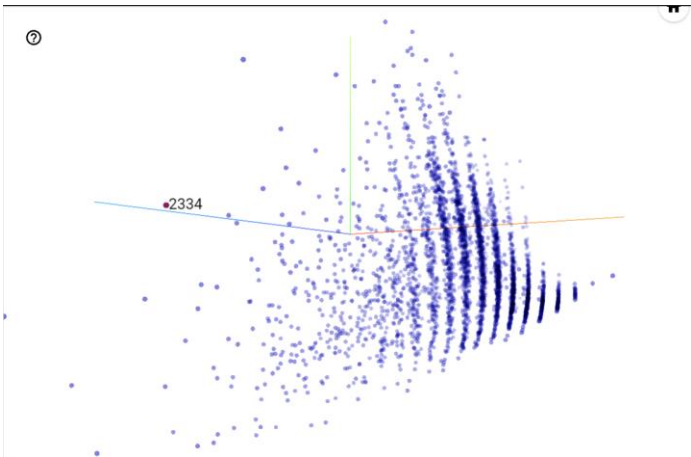
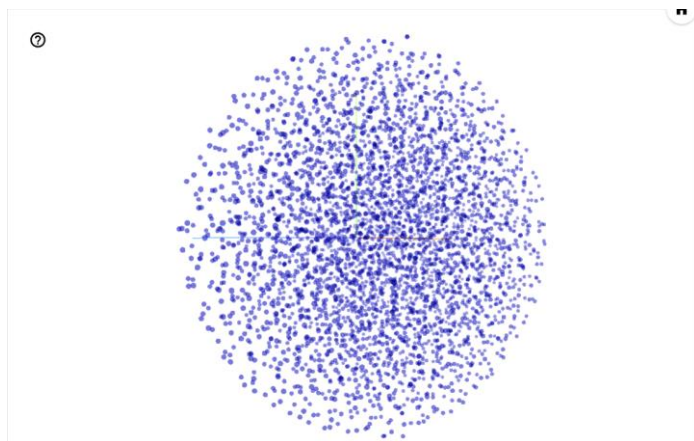


计算机科学与技术学院可视化技术实验报告

| | | |
|---|----------|-----------------|
| 实验题目：数据降维 | | 学号：201900150221 |
| 日期：10.9 | 班级：19 智能 | 姓名：张进华 |
| Email：zjh15117117428@163.com | | |
| 实验目的： 1. 体验 tensorflow 的降维 2. 选择数据进行降维，并比较 tsne, pca, isomap 等方法的区别 | | |
| 实验软件和硬件环境： Visual studio Code python 3.9.7 Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz | | |
| 实验步骤 步骤一 体验 tensorflow 的降维 首先下载数据集 abalone, 此数据集用来通过物理测量预测鲍鱼的年龄，而在我的实验中，我将鲍鱼的性别作为分类的 label, 对其他属性信息进行降维，在导入之前将其处理成为 tsv 格式文件 数据集 PCA 降维可视化效果如下：  <p>The image shows a scatter plot of data points (blue dots) in a 2D space. The points are clustered into several distinct groups, indicating that PCA has successfully separated the data into different classes. A single point is labeled '2334'.</p> 可以看到数据集降维的效果并不是特别好 数据集 t-sne 降维可视化效果如下： | | |



当迭代次数较少的时候, 各数据集其实并不能较好的分开, 当迭代次数逐渐增大的时候其实可以较好地分离不同 label 的数据

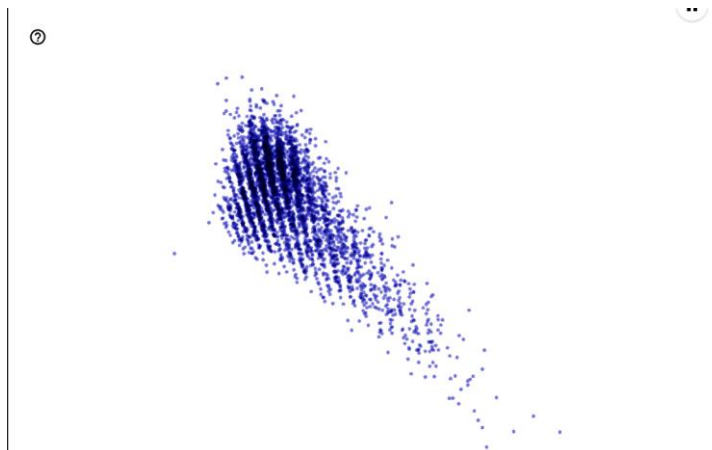
数据集 UMAP 降维可视化效果如下:



可以看到 UMAP 降维后的效果很奇怪

数据集 custom 降维可视化效果

custom 降维效果看不出来和原数据有什么区别



步骤二 降维比较 t-sne, pca, isomap 等方法的区别

这里仍使用上面的数据进行降维可视化，第一列为 label，实验数据集可视化效果如下：

| | Sex | Length | Diameter | Height | Whole weight | Shucked weight | Viscera weight | Shell weight | Rings |
|-----------------------|-----|--------|----------|--------|--------------|----------------|----------------|--------------|-------|
| 0 | M | 0.455 | 0.365 | 0.095 | 0.5140 | 0.2245 | 0.1010 | 0.1500 | 15 |
| 1 | M | 0.350 | 0.265 | 0.090 | 0.2255 | 0.0995 | 0.0485 | 0.0700 | 7 |
| 2 | F | 0.530 | 0.420 | 0.135 | 0.6770 | 0.2565 | 0.1415 | 0.2100 | 9 |
| 3 | M | 0.440 | 0.365 | 0.125 | 0.5160 | 0.2155 | 0.1140 | 0.1550 | 10 |
| 4 | I | 0.330 | 0.255 | 0.080 | 0.2050 | 0.0895 | 0.0395 | 0.0550 | 7 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 4172 | F | 0.565 | 0.450 | 0.165 | 0.8870 | 0.3700 | 0.2390 | 0.2490 | 11 |
| 4173 | M | 0.590 | 0.440 | 0.135 | 0.9660 | 0.4390 | 0.2145 | 0.2605 | 10 |
| 4174 | M | 0.600 | 0.475 | 0.205 | 1.1760 | 0.5255 | 0.2875 | 0.3080 | 9 |
| 4175 | F | 0.625 | 0.485 | 0.150 | 1.0945 | 0.5310 | 0.2610 | 0.2960 | 10 |
| 4176 | M | 0.710 | 0.555 | 0.195 | 1.9485 | 0.9455 | 0.3765 | 0.4950 | 12 |
| 4177 rows × 9 columns | | | | | | | | | |

可以看到数据集为八维，无法对原数据进行可视化，然后将第一列性别数据作为 label 进行分类，并进行可视化

1- 查看数据相关性

将输入数据 X 中的各个属性进行相关性分析，将彼此之间的关系映射到 0-1 之间，代表线性相关，绘制相关性散度矩阵如下：

| | Length | Diameter | Height | Whole weight | Shucked weight | Viscera weight | Shell weight | Rings |
|----------------|--------|----------|--------|--------------|----------------|----------------|--------------|-------|
| Length | 1.00 | 0.98 | 0.82 | 0.92 | 0.89 | 0.90 | 0.89 | 0.55 |
| Diameter | 0.98 | 1.00 | 0.83 | 0.92 | 0.89 | 0.89 | 0.90 | 0.57 |
| Height | 0.82 | 0.83 | 1.00 | 0.81 | 0.77 | 0.79 | 0.81 | 0.55 |
| Whole weight | 0.92 | 0.92 | 0.81 | 1.00 | 0.96 | 0.96 | 0.95 | 0.54 |
| Shucked weight | 0.89 | 0.89 | 0.77 | 0.96 | 1.00 | 0.93 | 0.88 | 0.42 |
| Viscera weight | 0.90 | 0.89 | 0.79 | 0.96 | 0.93 | 1.00 | 0.90 | 0.50 |
| Shell weight | 0.89 | 0.90 | 0.81 | 0.95 | 0.88 | 0.90 | 1.00 | 0.62 |
| Rings | 0.55 | 0.57 | 0.55 | 0.54 | 0.42 | 0.50 | 0.62 | 1.00 |

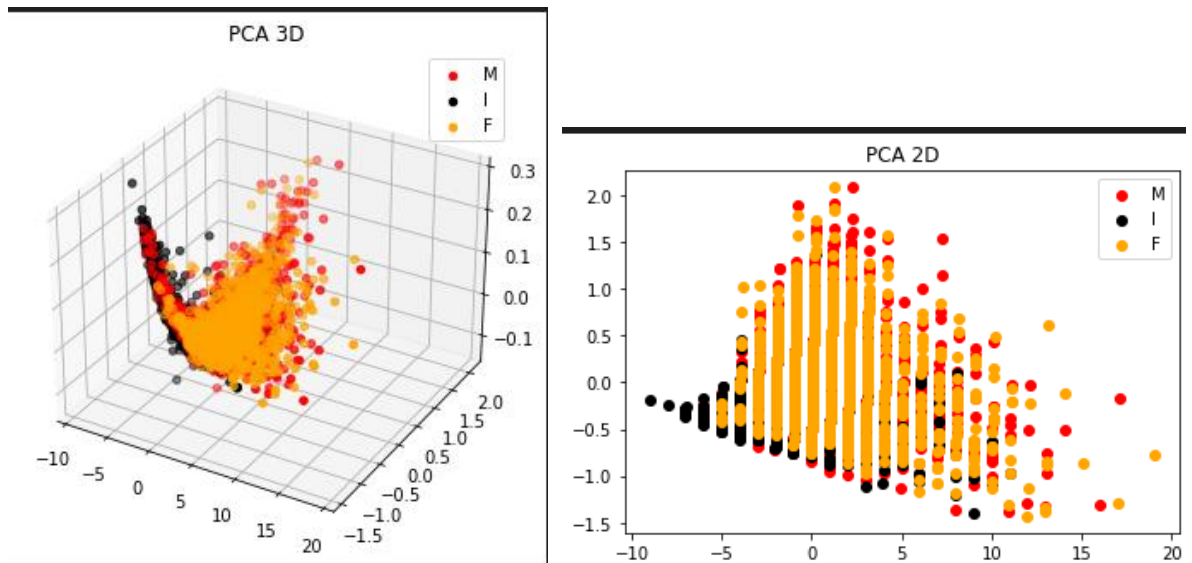
2-PCA 降维可视化

PCA 是一种将 n 维数据正交分解到 k 维的降维方法，选择了方差最大的 k 个方向作为数据维度保留方向，是一种由原空间线性组合降维的方法。

算法步骤：

- <1>. 设有 m 条 n 维数据，将原始数据按列组成 n 行 m 列矩阵 X；
- <2>. 将 X 的每一行进行零均值化，即减去这一行的均值，求出协方差矩阵；
- <3>. 求出协方差矩阵的特征值及对应的特征向量；
- <4>. 将特征向量按对应特征值大小从上到下按行排列成矩阵，取前 k 行组成矩阵 P，即为降维到 k 维后的数据。

绘制 PCA 降维分类图效果如下：



可以发现 PCA 降维的效果不是很好，降维后无法清楚的将三类数据区分

3-isomap 降维可视化

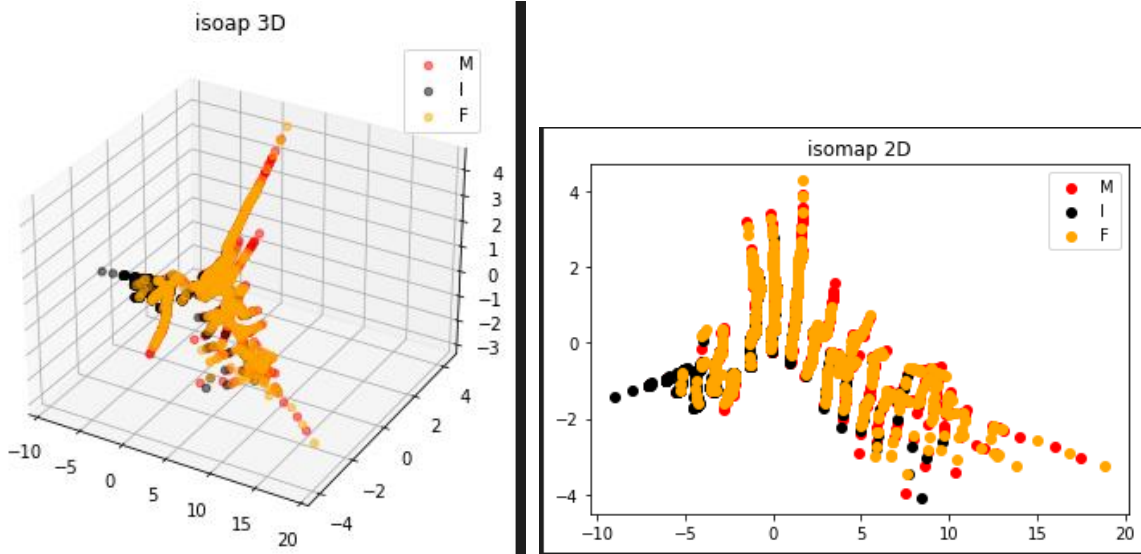
MDS（多维缩放）降维是一组对象之间的距离的可视化表示，也可以当做一种无监督降维算法使用。而 Isomap（等度量映射）是在 MDS 算法的基础上衍生出的一种非迭代的全局优化算法，它是一种等距映射算法，也就是说降维后的点，两两之间距离不变，这个距离是测地距离。

Isomap 算法没有多少公式推导的内容，它的创新点是引入测地线距离和提出对应的距离计算方法。此算法出发点，是认识到流形在高维空间中，两个样本之间的距离不该直接使用欧式距离计算直线距离，更应该是采用“测地线”距离，就像我们日常生活中送快递的例子，两个城市之间如果没有直达的路线，快递就会经过许多中转站才能送到，Isomap 通过将数据点连接起来构成一个邻接 Graph 来离散地近似原来的流形，而测地距离也相应地通过 Graph 上的最短路径来近似了

算法步骤：

- <1>. 对每个样本点 x ，计算它的 k 近邻；同时将 x 与它的 k 近邻的距离设置为欧氏距离，与其他点的距离设置为无穷大；
- <2>. 调用最短路径算法计算任意两个样本点之间的距离，获得距离矩阵 D ；
- <3>. 调用多维缩放 MDS 算法，获得样本集在低维空间中的矩阵 Z ；

绘制 isomap 降维分类图效果如下：



可以看到 isomap 实现效果要比 PCA 好一点，大体上可以看不同 label 数据的分布

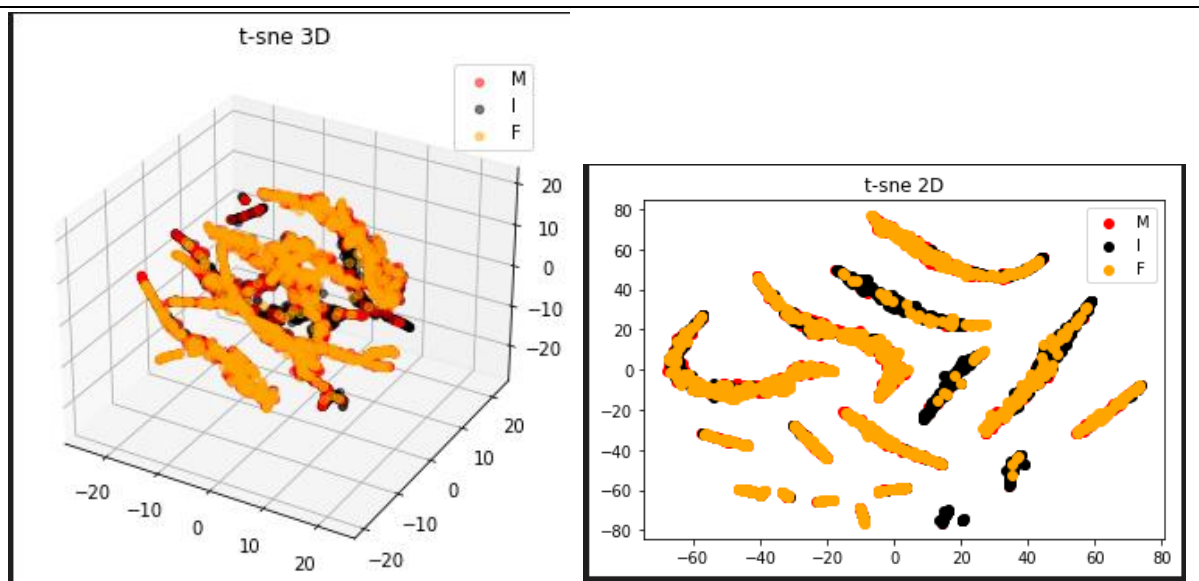
4-t-sne 降维可视化

t-SNE 全称为 t-distributed Stochastic Neighbor Embedding 翻译为 t-随机邻近嵌入，它是一种嵌入模型，能够将高维空间中的数据映射到低维空间中，主要用于高维数据的降维和可视化。

t-SNE 可以算是目前效果最好的数据降维和可视化方法之一，当我们想对高维数据集进行分类，但又不清楚这个数据集有没有很好的可分性（同类之间间隔小、异类之间间隔大）时，可以通过 t-SNE 将数据投影到 2 维或 3 维空间中观察一下：如果在低维空间中具有可分性，则数据是可分的；如果在低维空间中不可分，则可能是因为数据集本身不可分，或者数据集中的数据不适合投影到低维空间。

t-SNE 将数据点之间的相似度转化为条件概率，原始空间中数据点的相似度由高斯联合分布表示，嵌入空间中数据点的相似度由学生 t 分布表示。通过原始空间和嵌入空间的联合概率分布的 KL 散度（用于评估两个分布的相似度的指标，经常用于评估机器学习模型的好坏）来评估嵌入效果的好坏，即将有关 KL 散度的函数作为损失函数（loss function），通过梯度下降算法最小化损失函数，最终获得收敛结果，t-SNE 的缺点很明显：占用内存较多、运行时间长。

绘制 isomap 降维维分类图效果如下：



由图可知效果比上面两种有很大的改进

结论分析与体会：

PCA 相比于其他两种算法计算速度很快，其他两种算法都需要计算距离，所以导致耗费的时间长，样本数量一旦过多或者维度过大往往难以计算。但是 pca 的算法由于是线性的，所以能力有限，有时候很难处理有些问题。而 isomap 则需要设置 knn 算法的超参 k ，超参的设置也很影响降维的效果

Pca 的降维更加兼顾于全局的效果，而 isomap 则注重于特定流形的学习，而 tsne 相比之下则更加关注数据局部特征