《人工智能基础》课程实验一报告

一、数据集描述

本次使用的数据集是数字0~9的手写体图片数据集，图片已归一化为以手写数字为中心的8*8规格的图片。数据包括来自于43个人的1797张图片，原始图像为32×32像素的尺寸，使用NIST的预处理工具将其转化为8×8像素的图像，每个像素有16种灰度值。

数据集由训练集与测试集两个部分组成，比例大概为80%：20%，各部分规模如下：
训练集：1437个手写体图片及对应标签
测试集：360个手写体图片及对应标签

二、神经网络+监督学习

1.模型建立

搭建3层的神经网络，输入图像为8×8像素，因此输入神经元有64个，每个神经元对应一个像素点。
最终将数据分为10类，于是输出神经元有10个，最后对这个10个神经元的输出通过softmax来分类。

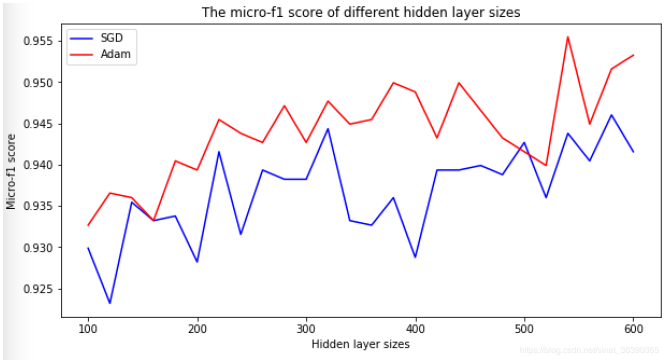
2.参数调节

（1）隐藏神经元个数:从100至600

（2）激活神经元种类:identity，tanh，relu

（3）优化器：Adam、SGD

固定神经元为relu，变化隐藏神经元个数，模型f-1值变化如下。发现Adam优化器效果较好，模型效果随隐藏神经元个数变多，有些许提升。



固定优化器为Adam，变化隐藏神经元个数，模型f-1值变化如下。发现relu和tanh激活神经元效果较好，模型效果随隐藏神经元个数变多，有一些提升。

3.结果

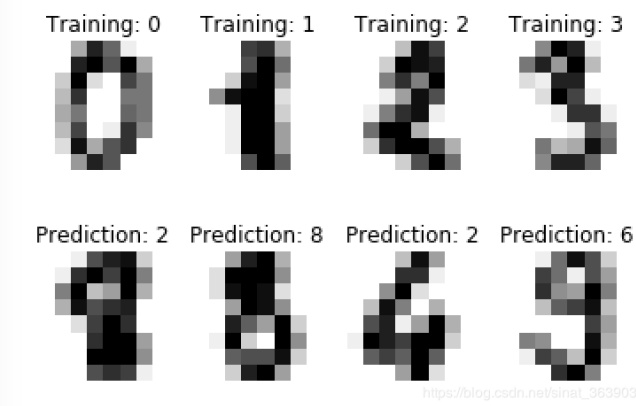
设置模型隐藏层为350，优化器为Adam，激活神经元为relu，学习率0.001。
运行后得到模型的各项评估指标如下：

	precision	recall	f1-score	support
0	1.00	1.00	1.00	27
1	0.94	0.97	0.96	35
2	1.00	0.97	0.99	36
3	0.97	1.00	0.98	29
4	1.00	0.97	0.98	30
5	0.97	0.97	0.97	40
6	1.00	0.98	0.99	44
7	0.93	1.00	0.96	39
8	0.97	0.95	0.96	39
9	0.97	0.95	0.96	41
avg / total	0.98	0.97	0.98	360

混淆矩阵为：

```
Confusion matrix:
[[27  0  0  0  0  0  0  0  0  0]
 [ 0 34  0  0  0  0  0  0  1  0]
 [ 0  0 35  0  0  0  0  1  0  0]
 [ 0  0  0 29  0  0  0  0  0  0]
 [ 0  0  0  0 29  0  0  1  0  0]
 [ 0  0  0  0  0 39  0  0  0  1]
 [ 0  1  0  0  0  0 43  0  0  0]
 [ 0  0  0  0  0  0  0 39  0  0]
 [ 0  1  0  1  0  0  0  0 37  0]
 [ 0  0  0  0  0  1  0  1  0 39]]
```

随机抽取部分分类样例如下：



与其他监督学习方法进行比较，发现神经网络效果基本是最好的，除了SVC
猜测可能是因为数据比较符合SVC的分布

init	timecc	precision		recall	f1-score
SVC	0.12s	0.992	0.992	0.992	0.992
GBC	2.95s	0.964	0.964	0.964	0.964
DecisionTree	0.01s	0.842	0.842	0.842	0.842
KNN	0.04s	0.975	0.975	0.975	0.975
RandomForest	0.02s	0.919	0.919	0.919	0.919
AdaBoost	0.13s	0.242	0.242	0.242	0.242
Bagging	0.09s	0.922	0.922	0.922	0.922
ExtraTree	0.00s	0.756	0.756	0.756	0.756
MLP	0.68s	0.981	0.981	0.981	0.981

三、非监督学习

1.模型建立

选取了几种比较常见的非监督学习算法，分别是K-Means、Birch、Agglomerative Clustering。
其中K-Means又有k-means++初始化、random初始化、PCA-based初始化、MiniBatch几种变体

K-Means

K-means通常被称为劳埃德算法。算法有三个步骤。第一步选择初始质心，最基本的方法是选择来自数据集的样本 。初始化之后，K-means在其他两个步骤之间循环。第一步将每个样品分配到最近的质心。第二步通过获取分配给每个先前质心的所有样本的平均值来创建新质心。计算旧的和新的质心之间的差异，并且算法重复这最后两个步骤，直到该值小于阈值。换句话说，它重复直到质心不显著移动。
簇内平方和标准可以被认为是内部相干簇如何的度量。它有各种缺点：
1）算法假设聚类是凸的和各向同性的，因此对细长簇或具有不规则形状的簇效果不好。
2）但是在非常高维空间中，欧几里德距离往往会膨胀。 在k均值聚类之前运行诸如PCA的维数降低算法可以缓解该问题并加速计算。

Birch

Birch为数据建立了一个特征树（CFT）。数据被有损压缩为一组特征特征节点（CF节点）。CF节点有许多特征子簇（CF Subclusters），位于非终端CF节点中的这些CF子簇可以将CF节点作为子节点。

算法描述：
将新样本插入CF树的根，该CF树是CF节点。然后将其与根的子簇合并，在合并后具有最小半径，受阈值和分支因子条件约束。如果子簇具有任何子节点，则重复执行此操作直到它到达叶子。在叶子中找到最近的子簇后，将递归更新此子簇的属性和父子簇。
如果通过合并新样本和最近的子簇获得的子簇的半径大于阈值的平方，并且如果子簇的数量大于分支因子，则临时为该新样本分配空间。取两个最远的子簇，并根据这些子簇之间的距离将子簇分成两组。
如果此拆分节点具有父子簇并且存在新子簇的空间，则父节点将拆分为两个。如果没有空间，则该节点再次分成两个，并且该过程以递归方式继续，直到它到达根节点。

Agglomerative Clustering

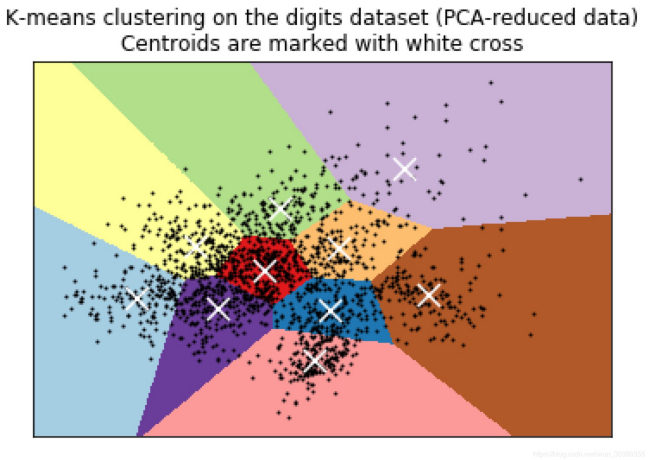
Agglomerative Clustering使用自下而上的方法执行层次聚类：许多簇被连续地合并在一起。以下是用于合并的度量标准：
Ward：最小化所有聚类中的平方差的总和。它是一种方差最小化方法，在这个意义上类似于k均值目标函数，但采用凝聚分层方法。
最大或完整的连接：最小化了簇对之间的最大距离。
平均连锁：最小化了所有簇对之间的距离的平均值。
单个连锁：最小化最近的簇对之间的距离。
AgglomerativeClustering 当它与连接矩阵一起使用时，也可以扩展到大量样本，但是当样本之间没有添加连接约束时计算成本很高：它在每个步骤考虑所有可能的合并。

2.结果

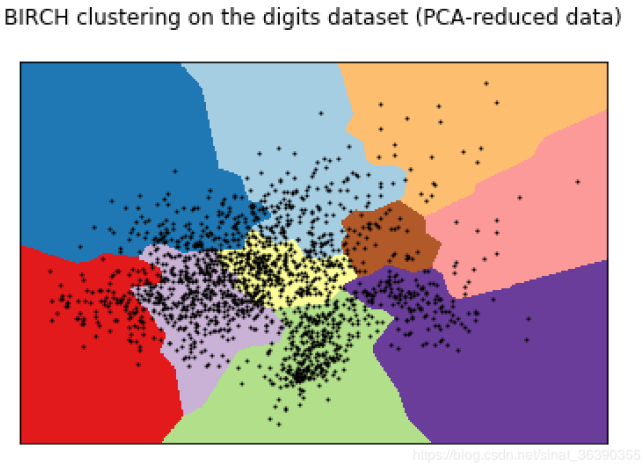
聚类得到的结果如下：

```
n_digits: 10,    n_samples 1797,        n_features 64
_____
init            time    homo    compl   v-meas  ARI     AMI     silhouette
k-means++       0.15s   0.602   0.650   0.625   0.465   0.598   0.146
random          0.14s   0.669   0.710   0.689   0.553   0.666   0.147
PCA-based       0.02s   0.671   0.698   0.684   0.561   0.668   0.118
Birch           0.26s   0.758   0.836   0.796   0.664   0.756   0.121
AggClustering   0.16s   0.758   0.836   0.796   0.664   0.756   0.114
MiniBatchKMeans 0.02s   0.626   0.664   0.644   0.513   0.622   0.115
_____
```

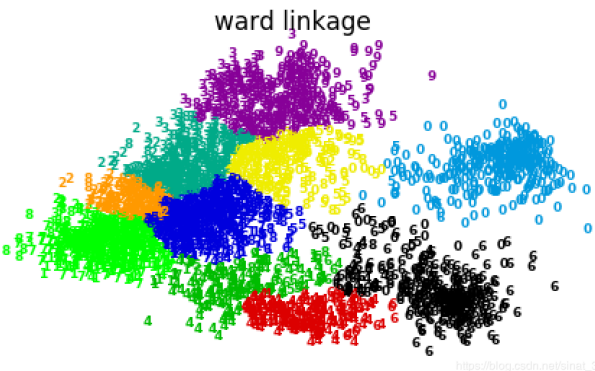
可视化K-Means聚类图：

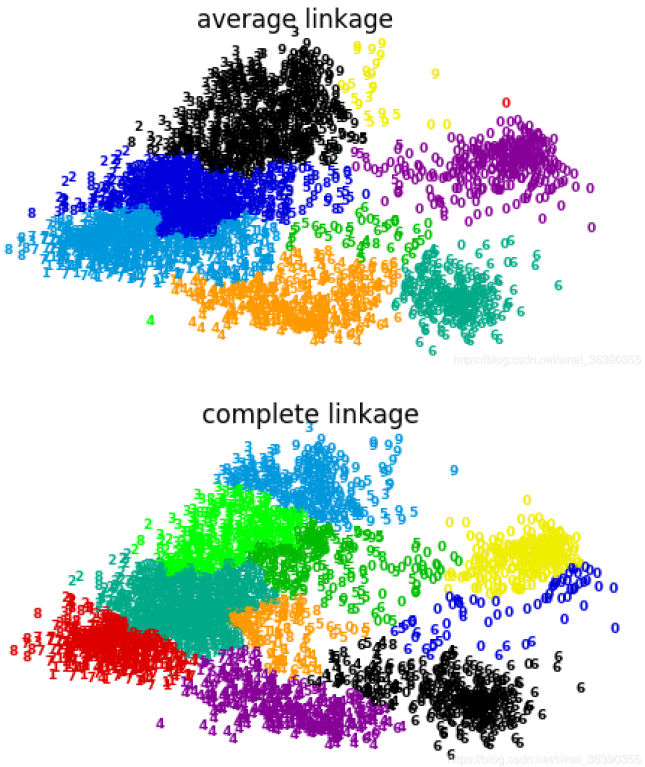


可视化Birch聚类图：



可视化Agglomarative聚类图：





四、结果分析

通过以上结果可以看出，监督学习的效果明显比非监督学习的效果好一些。监督学习的acc基本在80%～90%，MLP更是达到了98%。而非监督学习则只有60%～70%。
其实通过非监督学习的效果图就可以感受到，聚类只能把空间上相关的点聚在一起，不同的聚类方法只是定义空间相似性和形状的差异。但是有很多时候，数据的分布不很规整，这样必然就会丢失很多信息，造成模型性能的差异。
但是对于大规模数据的处理来说，标注数据很难获得，这时非监督学习就可以作为一个折中方案来应用。目前也出现了很多半监督学习方法的出现，旨在降低构建模型所使用的标注数据成本。