**模型融合的Mnist 字符识别**

**1. 实验目的：**用模型融合的方法，识别mnist字符，以提高最后的识别精度。

**2. 实验环境：**python3.7.6, ubuntu16以及一些python的包。

**3. 实验过程：**

**3.1. 训练数据、测试数据的选择**

MNIST数据库的来源是两个数据库的混合,一个来自Census Bureau employees(SD-3),一个来自high-school students(SD-1)；有训练样本60000个，测试样本10000个。训练样本和测试样本中，employee和student写的都是各占一半。60000个训练样本一共大概250个人写的。训练样本和测试样本的来源人群没有交集。

sklearn.datasets的插件包中提供了一个接口，通过load\_digits结果函数可以获取该数据集合的一部分数据，总共1797张图，每张图被缩放到8\*8的大小，也就是64个像素，该接口读取后，返回一个数据大小为(1797, 64)的二维矩阵。下图是一部分一些字符的可视化样例：

|  |
| --- |
|  |
| 图1. 字符样例图 |

在本实验中，选择前面的1000个图片作为训练样本，后面的797个作为测试样本。

**3.2. 基础模型参数的选择，模型训练，精度测试**

分别选择随机森林算法、以svm（支持向量机)为基础模型的Adaboost算法和GBDT（梯度提升树）算法分别对以上的数据集进行建模，用准确率（acc）作为评价指标，分别对模型进行性能的评价。

**3.2.1 随机森林算法**

在用随机森林建模之前，先用sklearn提供的GridSearchCV函数，对随机森林模型的参数进行搜索，以找到一个比较优质的参数，在本实验中，进行搜索到参数如下：

param\_grid = {

'n\_estimators': [100, 200, 500],

'max\_features': ['auto', 'sqrt', 'log2'],

'max\_depth' : [4,5,6,7,8],

'criterion' :['gini', 'entropy'],

}

GridSearchCV(estimator=RandomForestClassifier(random\_state=42), param\_grid=param\_grid, cv=5)

最后，经过搜索后，最优质的如下：

{'criterion': 'entropy', 'max\_depth': 8, 'max\_features': 'log2', 'n\_estimators': 500}

搜索完成后，用搜索得到的参数，建立随机森林模型，然后用训练好的模型进行测试，得到测试的acc是0.929737。

**3.2.2 Adaboost算法**

在本实验中，为了追求模型的速度，Adaboost算法的基础模型选择的是线性SVM模型。首先，也用sklearn提供的GridSearchCV函数，对**Adaboost**的参数进行搜索，主要搜索的是模型个数n\_estimators和学习率，搜索参数实验的主要代码如下：

params\_grid = {'n\_estimators': [50, 100,

"learning\_rate": [0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2],

}

svc=SVC(probability=True, kernel='linear')

clf\_adaboost = GridSearchCV(AdaBoostClassifier(base\_estimator=svc), params\_grid)

最后，经过搜索后，最优质的如下：

{'learning\_rate': 0.05, 'n\_estimators': 200}

搜索完成后，用搜索得到的参数，建立以线性svm为基础模型的**Adaboost**的模型，然后用训练好的模型进行测试，得到测试的acc是0.947302。

**3.2.3 GBDT（梯度提升树）算法**

在本实验中，为了追求模型的精度，也**对GBDT（梯度提升树）算法的参**数进行搜索，主要搜索的参数如下：

param\_grid = {

'n\_estimators': [100, 200, 500],

'max\_features': ['auto', 'sqrt', 'log2'],

'max\_depth' : [4,5,6,7,8],

}

clf\_gbdt = GridSearchCV(GradientBoostingClassifier(), param\_grid, cv=10, n\_jobs=-1)

最后，经过搜索后，最优质的如下：

{'max\_depth': 4, 'max\_features': 'log2', 'n\_estimators': 500}

搜索完成后，用搜索得到的参数，建立GBDT（梯度提升树）算法模型，然后用训练好的模型进行测试，得到测试的acc是0.922208。

**3.3 多模型融合策略及其结果对比**

为了尽可能提高模型的精度和模型的鲁棒性，在本实验中，对上述的三个模型进行了融合，为了对比，采用了两种融合的方法：对分数取均值，加权投票的方法。

**3.3.1 均值的方法：**

a. 先用三个模型对测试数据进行测试，得到每一个样本在三个模型下面的类别分数；

b. 分别对以上的三个模型得到的分数，进行样本维度的平均。

c. 对于一个样本，10个类别对应的分数平均值，谁大，该样本就判定为谁。

最后，得到的acc是0.942284

**3.3.2 加权投票的方法：**

用sklean包提供的ensemble.VotingClassifier将三个模型进行加权投票，因为上述实验中，第二个模型的准确率较高，所以被给与的权重也较大。最后，随机森林模型、Adaboost模型，GBDT（梯度提升树）模型选择的权重比例为 [3,5,2]，得到的acc是0.952321。

**3.3.3 结果对比**

下表1是以上所有方法，在797张图片上面的测试结果，可以看到，加权投票的方法可以取得最好的结果。

表1. 结果精度对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 随机森林模型 | Adaboost | GBDT | 融合mean | 融合vote |
| Acc | 0.929737 | 0.947302 | 0.922208 | 0.942284 | **0.952321** |

对加权投票的方法的结果做了一个混淆矩阵的可视化，如下图2：

|  |
| --- |
|  |
| 图2. 混淆矩阵 |

从混淆矩阵中可以看到1被错误识别成了9的数目最多，有9个；另外3被识别成了8的样本，也比较多，一共有6个。

**4. 总结与问题**

4.1 模型融合可以提高模型的精度。在本实验中，加权投票的方法，相对单个最好的模型，可以将acc提高0.5%（0.952321-0.947302）。

4.2融合方法的选择很重要。以上实验中，分数的均值融合，整体acc相对的最好的模型，并没有提高。融合的策略有很多，需要用实验多尝试，或者用搜索的方法，搜索出一个最好的融合策略。

4.3 单个模型参数的搜索还可以更优。因为时间原因，对单个模型参数搜索的时候，还不够充分，后面可以增加搜索的参数，还可以尝试增大单个参数的搜索范围，可能会取得更高的准确率。

4.3 深度学习的方法还没有来的尝试。随着深度学习越来越广泛，已经在很多领域取得了很好的效果，在本实验中，也可以用深度学习的方法进行尝试，有可能会取得更高的acc。