**一、实验说明**

1. 操作流程，准确率均可在代码运行的终端看到。我们将未使用PAC的结果保存在only\_KNN\_result.txt，将使用PAC降维数据的结果保存在KNN+PCA\_result.txt。

2. 代码大约运行 3min。代码使用KNN分类器，原始准确率90.8%，使用PCA后的准确率为92.8%。

3. KNN中K值选3，欧式距离度量，使用多数表决策略。

4. CPA算法中的K值选择：

我们通过实验准确率进行K值的选择。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| K值 | 50 | 100 | 150 | 200 | 250 | 300 |
| Accuracy | 92.8% | 91.8% | 91.0% | 89.6% | 90.0% | 89.6% |

分析：我们对图像进行了2值化处理，这造成了 图像变稀疏，使用PAC降维后，可以提取出图像中主要的影响特征，而忽略对结果有噪音的稀疏特征，所以将特征降维到较低维度，可以得到更好的结果。

5. 请将所需的数据文件：train-images-idx3-ubyte.gz、train-labels-idx1-ubyte.gz、t10k-images-idx3-ubyte.gz、t10k-labels-idx1-ubyte.gz 像代码中那样放在./data目录下，以保证代码正常运行。

6. 有问题请联系我，谢谢！祝每天开心快乐！

**二、实验代码**

KNN.py中是KNN算法相关函数：

L9 – L44 数据读入，数据预处理代码。

L47 – L100 KNN算法相关代码。

PAC.py中是PAC降维的代码。

L5 – L24 PAC降维代码，细节在代码中有明确注释。

L27 – L69 主运行代码。

**三、实验结果**

代码运行结束后即可显示结果，截图如下：