

上机实验报告

|  |  |
| --- | --- |
| 题目： | 运用支持向量机识别手写数字 |

|  |  |
| --- | --- |
| 姓 名： | **李梓焓** |
| 学 号： | **1700011735** |
| 组 别： | **无** |
| 实验日期： | **2019.11.07** |

**摘要** 本实验运用支持向量机，采用MMIST手写数字数据集进行训练，并对其进行测试，计算相应的混淆矩阵、准确率、召回率，从而得出结论：

**关键词** 支持向量机，识别，手写数字

1. **引言**

分类问题可分为线性可分问题和线性不可分问题，对于线性可分问题，支持向量机的解决思路，是找到能正确划分训练数据集并且几何间隔最大的分离超平面，换句话说就是不仅将正负实例点分开，而且对最难分的实例点（离超平面最近的点）也有足够大的确信度将它们分开；对于线性不可分问题，支持向量机则利用核函数，将线性不可分问题通过某种方法转换为线性可分问题。

在线性可分的情况下，训练数据集的样本点中与分离超平面距离最近的样本点的示例称为支持向量，支持向量是使约束条件成立的点。在决定分离超平面时，只有支持向量起作用，而其他实例点并不起作用。如果移动支持向量，将改变所求的解；但是如果在间隔边界以外移动其他实例点，甚至去掉这些点，则解不发生改变，由于支持向量在确定分离超平面中起着决定性的作用，所以将这种分类称为支持向量机。支持向量的个数一般很少，所以支持向量机由很少的“重要的”训练样本确定。

本次上机实验采用作业附带的mlc\_mnist.py，从MMIST手写数字数据集提供的文件读入数据，然后用Scikit-learn自带的SVC，训练支持向量机和测试，并分别用confusion\_matrix()、precision\_score()和recall\_score()计算混淆矩阵、准确率、召回率。

**2 实验部分**

**2.1 仪器**

2.1.1硬件

Surface Pro（第5代，处理器参数：Intel® Core™ i5-7300U CPU @ 2.60GHz，2.71 GHz，2个内核，4个逻辑处理器；内存容量：8.00 GB）

2.2.2软件

操作系统：Windows 10家庭版，版本1903

开发环境：Visual Studio 2019 Community（已预装64位Miniconda 3，含64位Python 3.7.4、Conda 4.7.12、NumPy 1.16.5、Scikit-learn 0.21.2、joblib 0.13.2）

2.2.3训练和测试数据

train-images-idx3-ubyte和train-labels-idx1-ubyte：训练用手写数字和标签（均转存为二进制文件），共计6万个样本。

t10k-images-idx3-ubyte和t10k-labels-idx1-ubyte：测试用手写数字和标签（均转存为二进制文件），共计1万个样本。

**2.2 实验过程**

2.2.1训练和测试数据的读入

采用作业附带的mlc\_mnist.py，读取训练样本和测试样本的二进制文件，并转换为numpy的ndarray。

2.2.2训练支持向量机

采用Scikit-learn自带的MLPClassifier，利用读入的训练样本，训练支持向量机。其中缓冲大小为1024（单位MB），gamma采用“scale”，核函数为“rbf”，核函数的最高次数degree为3，收敛限为0.0001，惩罚系数为1.0，其余参数采用默认参数。

2.2.3预测测试集并评价预测结果

利用训练好的神经网络对测试样本进行预测，然后分别用confusion\_matrix()、precision\_score()和recall\_score()计算混淆矩阵、准确率、召回率。

2.2.4保存模型

采用joblib的dump()，保存训练模型。

2.2.5支持向量机核函数类型与训练效果的关系

使用固定的随机种子，重复2.2.1-2.2.4的流程，每次仅改变支持向量机采用的核函数类型，然后计算支持向量机预测结果的混淆矩阵、准确率、召回率。

2.2.6支持向量机核函数最高次数与训练效果的关系

使用固定的随机种子，重复2.2.1-2.2.4的流程，每次仅改变支持向量机核函数的最高次数，然后计算支持向量机预测结果的混淆矩阵、准确率、召回率。

2.2.7支持向量机惩罚系数与训练效果的关系

使用固定的随机种子，重复2.2.1-2.2.4的流程，每次仅改变支持向量机的惩罚系数，然后计算支持向量机预测结果的混淆矩阵、准确率、召回率。

**3 数据与结果分析**

**3.1固定随机状态后支持向量机预测效果**

random\_state设定为1551，按2.2.1-2.2.4进行实验，得混淆矩阵，如表1所示：

**表1** 固定随机状态后支持向量机预测结果的混淆矩阵

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实际\预测 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 967 | 0 | 2 | 1 | 0 | 3 | 3 | 2 | 2 | 0 |
| 1 | 0 | 1125 | 5 | 0 | 0 | 1 | 2 | 0 | 2 | 0 |
| 2 | 5 | 1 | 996 | 2 | 2 | 0 | 1 | 15 | 9 | 1 |
| 3 | 0 | 0 | 3 | 980 | 1 | 7 | 0 | 12 | 7 | 0 |
| 4 | 0 | 0 | 13 | 0 | 945 | 2 | 3 | 7 | 2 | 10 |
| 5 | 2 | 0 | 2 | 11 | 1 | 857 | 6 | 5 | 6 | 2 |
| 6 | 6 | 2 | 0 | 0 | 4 | 8 | 927 | 6 | 5 | 0 |
| 7 | 1 | 6 | 13 | 3 | 3 | 0 | 0 | 989 | 0 | 13 |
| 8 | 3 | 0 | 6 | 5 | 6 | 10 | 3 | 12 | 926 | 3 |
| 9 | 4 | 5 | 6 | 11 | 13 | 2 | 0 | 21 | 3 | 944 |

**表2** 固定随机状态后支持向量机预测结果的准确率和召回率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模型得分 | 微平均准确率 | 宏平均准确率 | 加权平均准确率 | 微平均召回率 | 宏平均召回率 | 加权平均召回率 |
| 0.9656 | 0.9656 | 0.9657 | 0.9658 | 0.9656 | 0.9652 | 0.9656 |

**3.2支持向量机核函数类型与训练效果的关系**

random\_state设定为1551，重复2.2.1-2.2.4的流程，每次仅改变支持向量机采用的核函数类型，其他参数不变，结果如表3所示。

**表3** 支持向量机采用的核函数类型与训练效果关系表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 采用的核函数类型 | rbf(高斯函数) | poly(多项式函数) | linear(线性函数) | sigmoid(逻辑函数) |
| 模型得分 | 0.9656 | 0.9611 | 0.9281 | 0.8885 |
| 微平均准确率 | 0.9656 | 0.9611 | 0.9281 | 0.8885 |
| 宏平均准确率 | 0.9657 | 0.9615 | 0.9279 | 0.8876 |
| 加权平均准确率 | 0.9658 | 0.9619 | 0.9283 | 0.8885 |
| 微平均召回率 | 0.9656 | 0.9611 | 0.9281 | 0.8885 |
| 宏平均召回率 | 0.9652 | 0.9610 | 0.9270 | 0.8868 |
| 加权平均召回率 | 0.9656 | 0.9611 | 0.9281 | 0.8885 |

**3.3支持向量机核函数最高次数与训练效果的关系**

random\_state设定为1551，核函数设定为“poly”（这是因为核函数最高次数仅对多项式核函数有效），重复2.2.1-2.2.4的流程，每次仅改变支持向量机核函数的最高次数，其他参数不变，结果如表4所示。

**表4** 支持向量机核函数的最高次数与训练效果关系表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 核函数的最高次数 | 2 | 3 | 4 | 5 |
| 模型得分 | 0.9712 | 0.9611 | 0.8939 | 0.7595 |
| 微平均准确率 | 0.9712 | 0.9611 | 0.8939 | 0.7595 |
| 宏平均准确率 | 0.9711 | 0.9615 | 0.9219 | 0.8920 |
| 加权平均准确率 | 0.9712 | 0.9619 | 0.9232 | 0.8937 |
| 微平均召回率 | 0.9712 | 0.9611 | 0.8939 | 0.7595 |
| 宏平均召回率 | 0.9710 | 0.9610 | 0.8925 | 0.7553 |
| 加权平均召回率 | 0.9712 | 0.9611 | 0.8939 | 0.7595 |

**3.4支持向量机惩罚系数与训练效果的关系**

random\_state设定为1551，核函数重新设定为“rbf”，重复2.2.1-2.2.4的流程，每次仅改变支持向量机的惩罚系数，其他参数不变，结果如表5所示。

**表5** 支持向量机的惩罚系数与训练效果关系表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 支持向量机的惩罚系数 | 0.1 | 1 | 10 | 100 |
| 模型得分 | 0.9329 | 0.9656 | 0.9719 | 0.9716 |
| 微平均准确率 | 0.9329 | 0.9656 | 0.9719 | 0.9716 |
| 宏平均准确率 | 0.9340 | 0.9657 | 0.9721 | 0.9717 |
| 加权平均准确率 | 0.9341 | 0.9658 | 0.9721 | 0.9717 |
| 微平均召回率 | 0.9329 | 0.9656 | 0.9719 | 0.9716 |
| 宏平均召回率 | 0.9320 | 0.9652 | 0.9716 | 0.9712 |
| 加权平均召回率 | 0.9329 | 0.9656 | 0.9719 | 0.9716 |

**3.5支持向量机与感知神经网络训练效果对比**

由于之前的感知神经网络实验中，random\_state也设定为1551，因此可用来比较两种方法的训练效果，结果如表6所示。

**表6** 支持向量机与感知神经网络训练效果

|  |  |  |  |
| --- | --- | --- | --- |
|  | 最佳单层感知神经网络 | 最佳多层感知神经网络 | 最佳支持向量机 |
| 模型得分 | 0.9691 | 0.9731 | 0.9719 |
| 微平均准确率 | 0.9691 | 0.9731 | 0.9719 |
| 宏平均准确率 | 0.9689 | 0.9728 | 0.9721 |
| 加权平均准确率 | 0.9692 | 0.9731 | 0.9721 |
| 微平均召回率 | 0.9691 | 0.9731 | 0.9719 |
| 宏平均召回率 | 0.9688 | 0.9729 | 0.9716 |
| 加权平均召回率 | 0.9691 | 0.9731 | 0.9719 |

注：最佳单层感知神经网络参数：求解器采用adam，激活函数采用logistic，正则化系数为，使用1层隐藏层，包含200个节点，最大迭代数为500，学习率恒定为；最佳多层感知神经网络参数：求解器采用adam，激活函数采用logistic，正则化系数为，使用2层隐藏层，每个隐藏层各包含200个节点，最大迭代数为500，学习率恒定为；最佳支持向量机参数：缓冲大小为1024（单位MB），gamma采用“scale”，核函数为“rbf”，收敛限为0.0001，惩罚系数为1.0。

**3.6实验结论**

（1）在手写数字识别中，用高斯函数作为核函数，收敛限设为0.0001，训练出来的支持向量机准确率和召回率达90%以上，基本满足手写数字识别的需求。

（2）在其他条件不变的情况下，以高斯函数为核函数，训练出来的支持向量机表现最好，准确率和召回率达96%以上。指数为3的多项式函数的表现稍逊于高斯函数，但准确率和召回率也达到96%以上。线性函数和逻辑函数的表现不如高斯函数和指数为3的多项式函数，其中逻辑函数的表现最差。

（3）在其他条件不变的情况下，对多项式函数而言，随着多项式最高指数由2增加到5，支持向量机的表现逐渐劣化，且耗时逐渐增加，这可能与指数增加所带来的过拟合有关。因此，使用较高指数的多项式函数时，最好要配合较大的惩罚系数，以防止模型出现过拟合现象。

（4）在其他条件不变的情况下，对高斯函数而言，随着惩罚系数从0.1增加到100，支持向量机的表现先出现明显改善，随后出现轻微劣化，这说明，适当增大惩罚系数能极大改善预测结果。

（5）在随机状态相同的情况下，最佳的支持向量机模型与最佳的感知神经网络模型，其训练效果从最差到最好的顺序为：单层感知神经网络 < 支持向量机 < 多层感知神经网络，且准确率和召回率均为97%左右。