# 扑克牌分类

## 1问题阐述

扑克牌是52张牌，红心(Hearts), 黑桃(Spades), 方片(Diamonds), 梅花(Clubs),每一种各13张，分别为1，2，3，4，5，6，7，8，9，10，J，Q，K组成。数据集中，每个样本包含了5张扑克牌的数据，这五张扑克牌是由一副标准扑克拍中随机抽取，每张扑克牌由两个特征进行表示：花色和数字，因此一个样本中包含10个扑克牌特征。花色（红心，黑桃，方片，梅花）分别由数字1，2，3，4表示，扑克牌牌面数字（1，2，.....，Q，K）由数字1，2，...，13表示。

表1 数据集描述

|  |  |
| --- | --- |
| 数据集类型 | 多分类 |
| 训练集样本数目 | 1000000 |
| 测试集样本数目 | 25010 |
| 数据集单样本特征数目 | 10 predictive attributes  1 goal attribute |
| 有无缺失值 | 无 |

数据集特征说明：

第1，3，5，7，9列分别表示第1，2，3，4，5张牌的花色，第2，4，6，8，10列分别表示第1，2，3，4，5张牌的牌面数字。

第11列表示纸牌所归于的类别，共10种，分别用0--9表示，每种数字所代表的含义为：

0: Nothing in hand; not a recognized poker hand

1: One pair; one pair of equal ranks within five cards

2: Two pairs; two pairs of equal ranks within five cards

3: Three of a kind; three equal ranks within five cards

4: Straight; five cards, sequentially ranked with no gaps

5: Flush; five cards with the same suit

6: Full house; pair + different rank three of a kind

7: Four of a kind; four equal ranks within five cards

8: Straight flush; straight + flush

9: Royal flush; {Ace, King, Queen, Jack, Ten} + flush

任务为：预测5张纸牌所归于的类别。

训练集和测试集已随机分割好，统一使用训练集训练模型，使用测试集预测数据。

衡量指标：评价指标可选，即指标一和二任选。

评价指标一：类别预测正确的的数目。

评价指标二：计算每一类的预测准确率precision，预测准确率precision :



基于是多分类，可尝试使用top1,top2,top3预测准确率， 最后得出一个总的mAP值。

# 2技术原理

DNN,即Deep Neural Network,是指深度神经网络算法，它是深度学习的基础。

相比于传统的感知机，深度网络加入了隐藏层且隐藏层可以有多层，输出层的神经元也可以有多个输出。感知机的激活函数虽然简单但处理能力有限，深度网络对激活函数做了扩展，比如Sigmoid、tanx、softmax、和Relu等。

DNN内部的神经网络可以分为3类，即输入层、隐藏层和输出层。一般来说第一层是是输入层，最后一层是输出层，而中间的层数都是隐藏层。DNN的前向传播算法就是利用若干个权重系数矩阵W,偏倚向量b来和输入值向量x进行一系列线性运算和激活运算，从输入层开始，一层层的向后计算，一直到运算到输出层，得到输出结果为值。DNN的反向传播算法是知道输入层输出层的神经元数以及中间隐藏层的若干神经元，找到合适的所有隐藏层和输出层对应的线性系数矩阵W，偏倚向量b，让所有的训练样本输入计算出的输出尽可能的等于或很接近样本输出。

本文用的深度学习框架是Tensorflow，使用了其中的API分类接口即tf.estimator.DNNClassifier。

feature\_columns用来指定特征列，hidden\_units则

用于定义隐藏层的层数以及每层的神经元数，n\_classes用于指定需要分多少类。

DNNClassifier.train根据所给数据input\_fn对模型进行训练，可以指定神经网络训练的步数。

DNNClassifier.evaluate根据数据input\_fn，对模型进行验证。对于每一步，执行input\_fn，然后返回数据的一个batch，直到验证完成。

DNNClassifier.predict则是根据给出的特征进行预测，得到预测结果。

# 3实验方法

1. 加载数据集数据并进行数据分割。

poker-hand.names是对Poker Hand Dataset数据集的解释，共有10个标签：Nothing in hand，One pair，Two pairs，Three of a kind，Straight，Flush，Full house，Four of a kind，Straight flush，Royal flush。训练集样本数目为1000000，测试集样本数目为25010。

采用pandas模块可将poker-hand-training.data和poker-hand-test.data两个文件读取为CSV文件格式，并赋予各列表头名称。然后对数据集的特征和标签进行分割，在数据集中，前10列为特征，最后1列为标签，根据CSV对各列进行处理，即可实现数据集数据的分割，得到train\_x, train\_y, test\_x, test\_y。

1. 训练DNN网络并进行评估。

采用TensorFlow的tf.estimator.DNNClassifier建立深度网络模型，选用5个隐藏层，每层的神经元数都设置为100，指定n\_classes为10，因为数据集的标签共有10类。

对建立好的深度学习网络模型进行训练，训练数据为训练集数据，对训练数据进行batch处理 ，每个batch定义为100。一共训练50000次。

对模型进行评估，评估数据为测试集数据，同样也是batch处理，每个batch定义为100。

1. 处理Flush, Straight flush, Royal flush情况。

在实验过程中发现，训练好的模型会对Flush, Straight flush, Royal flush这三种情况误判，导致分类错误。因此要对Flush, Straight flush, Royal flush情况进行另外处理。Flush是指同花色，Straight flush是指同花色且数字连续，Royal flush是指同花色且数字为1,10,11,12,13。处理及判断方法如图1所示。

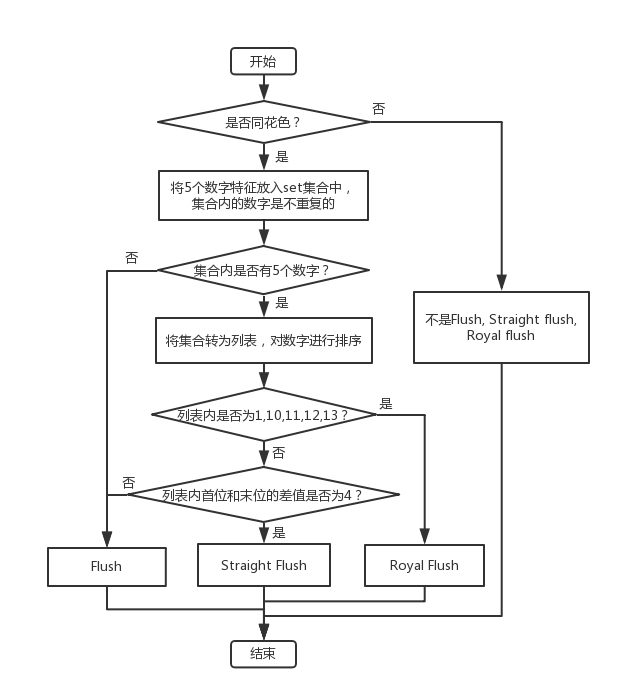


图1 判断Flush, Straight flush, Royal flush

1. 使用测试集验证分类准确率。

使用训练好的分类器对测试集数据进行预测，同样进行batch处理，可以得到测试结果。结合对Flush, Straight flush, Royal flush等情况的处理，得到测试集中类别预测正确的数目、每一类预测正确的数目以及预测准确率。

整体的步骤流程图如图2所示。

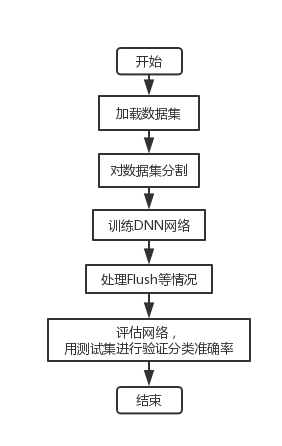


图2 Poker-Hand分类的整体步骤流程图

## 4 实验结果

通过用测试集数据对训练好的DNN网络进行测试，得到了总的分类正确数目以及各类的分类正确数目，并计算得出模型评估结果及准确率。

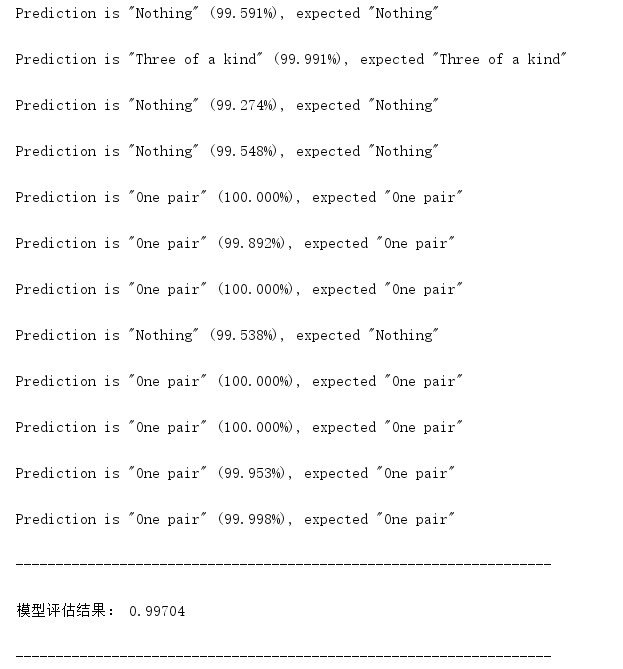


图3 对样本数据进行预测分类

从图3可以看出，训练好的DNN模型基本都能以很高的准确率对样本数据进行分类，对DNN模型进行评估，即运用DNNClassifier.evaluate对模型进行验证，得到的模型评估结果为99.704%。

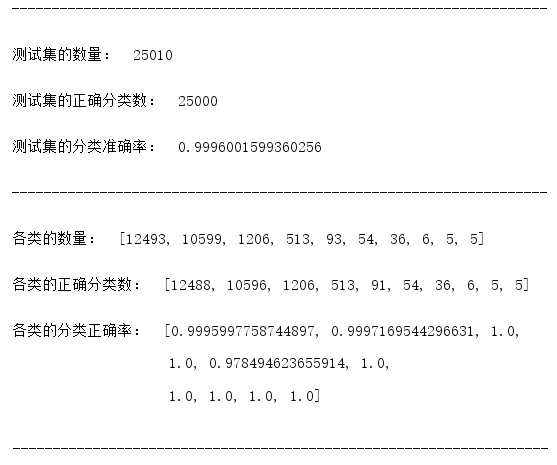


图4 测试集的分类结果

测试集的分类结果如图4所示。其中，测试集的样本数量为25010，正确分类的数目为25000（经过多次测试，正确分类数目均接近于25000），测试集的分类准确率达到了99.96%。

如果仅是使用训练好的DNN模型，并不能对Flush, Straight flush, Royal flush情况进行分类，这三类的分类结果都是0%，即完全分类错误。改进方案后，能够以100% 的准确率对这三种情况进行分类。

在测试集中，10类的样本数量为[12493, 10599, 1206, 513, 93, 54, 36, 6, 5, 5]，而本文方法的各类正确分类数为[12488, 10596, 1206, 513, 91, 54, 36, 6, 5, 5]。‘Nothing’错误分类5个，‘One Pair’错误分类3个，‘Three of a kind’错误分类2个，其余全部正确。各类的分类准确率均很高甚至为100%。

表1 每10000步的loss值

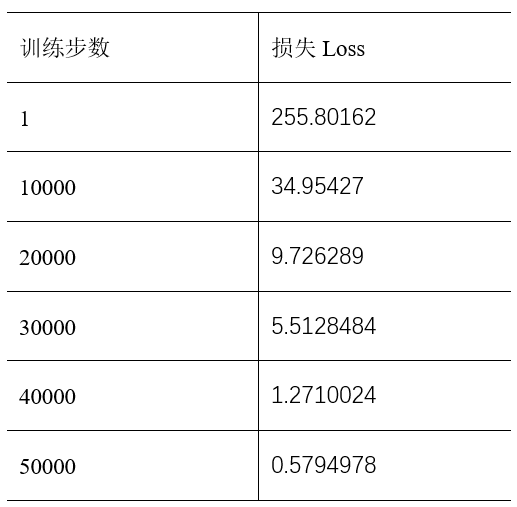


图5 训练步数与损失Loss的关系

当训练次数增加时，损失将会逐渐降低，如图5所示。当训练次数为50000步时，模型的loss损失维持在1.0以内。表1则给出了每10000步的loss损失值。

项目分为2个文件，运行poker\_with\_UI,将显示pyqt制作的UI界面，提供可视化选择操作。运行poker\_without\_UI,直接运行程序，选用Poker\_Test列表中的值进行测试。

为增加可视乎交互效果，基于扑克牌分类制作了一个UI界面，如图6所示，用于测试本文方法对自定义样本的分类准确度。对样本的测试结果如图7所示。



图6 扑克牌分类的UI界面

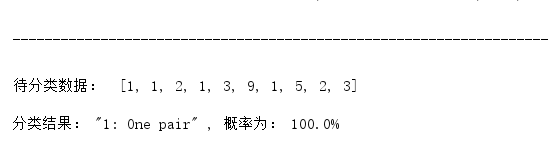


图7 自定义样本的分类结果