# SVM分类实验报告

牛远卓

2022/05/15

## 数据集

数据集简介：Omniglot 数据集包含来自50 个不同字母的1623个不同手写字符，如下图所示。

数据规模：共200个类别，每个类别有20个样本，15个作为训练样本，另外5个作为测试样本。每个样本为28\*28。



由于数据集为mat格式，我yongsvm分类时不需要做任何转换就能直接分类

## 实验过程

### 数据导入

实验先将（200，15，28，28）的训练数据和（200，5，28，28）的测试数据分别通过flatten导入到训练和测试的一维数组中。再将他们转换成numpy.array的形式，这样就可以直接放入svm中了。

### svm分类

我用的svm分类是用了Pytorch自带的包：sklearn.svm。然后调了些不同的参数，比如kernal,C,gamma等，来找出测试正确率最高的svm模型。

在选择kernal这一任务上，我参考了吴恩达老师的看法。

1.当 n 很大，m相对于n较小时。（如 n =10000，m = 10-1000）  
选用模型：逻辑回归、线性SVM（其实就是普通不带核函数的SVM，视频中也叫线性核函数。）  
个人理解：当特征维度过高，每个样本提供的信息已足够训练（如在已知房子的大小、装修情况、交通等等特征后，很容易退出房子单价。），这样的模型很容易过拟合。如果还采用带核函数的SVM，会使得模型更加容易过拟合，所以采用线性模型就够了。  
解决方法：降维、增大数据量、正则等等  
PS.听过一句话，高维线性模型等于低维非线性模型，结合核函数很好理解。

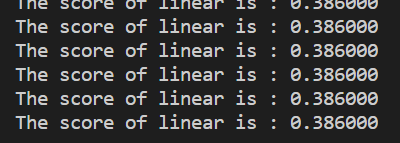
2.n小，m一般大时。（n = 1-1000 , m = 10-10000/20000）  
选用模型：非线性SVM（如高斯核SVM）  
个人理解：此时样本数相比于特征数可能差别不是很大，样本特征提供的信息可能不足，为了防止模型欠拟合，所以吴恩达建议使用非线性SVM。

3.n小，m很大时。（n = 1-1000 , m > 50000）  
选用模型：非线性SVM 或 逻辑回归、线性SVM。  
个人理解：先说吴恩达提到这种大批量训练集情况下使用非线性SVM也不是不可以，只不过核函数的存在会拖累运行速度。使用他建议 增加/创建 新的特征，然后再使用逻辑回归、线性SVM。这个我的理解是，由于特征少，训练得来的模型可能欠拟合，此时需要增加新的特征，获取更多特征信息。（欲向第一种情况靠拢）

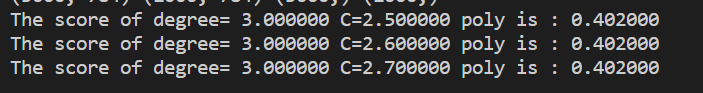
## 实验结果

我主要测了kernal为linear，poly与rbf的情况，发现:

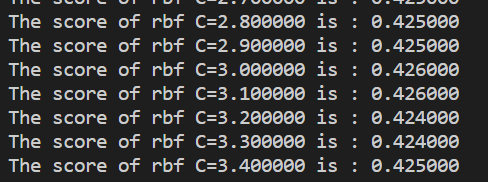
kernal=linear时，模型对其他参数都不敏感，毕竟是线性模型，能有啥超参数呢。正确率0.386。



kernal=poly时，模型degree敏感，毕竟是多项式模型。degree=3时，正确率最高，为0.402。



kernal=rbf时，模型对C敏感，C=3.0或3.1时，正确率最高，为0.426。



由于训练结果不太理想，我查了查能改善的方法。

我希望用数据增强，增强一下数据集的多样性。但是Pytorch自带的方法只支持图像类型的数据，于是我又将Mat类型的数据转换成了图像的，分类存好。通过imagefolder将数据导入进来。

除了这一方面，我有想了向更主要的问题：svm主要是将一些不同的点再维度空间内通过超平面分类，所以希望数据是再这些超平面上按照类型聚拢在一起。所以我希望用一个映射将这些mat文件的数据按类型自动聚合在一起。我用的模型是alexnet，loss function就是一个类型中不同点的欧氏距离，optimizer用的是lr=0.01,momentum=0.9的SGD。

通过上述的方法预处理完数据集，再进入测试后效果好了很多。

之后用了上述的线性kernal重新跑了一遍，正确率0.978



## 实验收获

这次学习了svm的主要架构以及各项参数的物理意义。

C : float, default=1.0

正则化参数. 正则化的强度与C的大小成反比. 但必须大于0. 用的是l2正 则化。

kernel :

核函数{'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'}, default='rbf'

degree : int, default=3

多项式的度数

gamma : {'scale', 'auto'} or float, default='scale'

'rbf', 'poly' and 'sigmoid'中的核系数

如果gamma是scale,那么gamma=1/(n\_features \* X.var())

如果gamma是auto,那么gamma=1/n\_features

coef0 : float, default=0.0

在核函数中的独立变量

只在poly和sigmoid有用。