Homework 2

张思源　　*21110850018*

October 21, 2021

1 Ex1

用 SVM 解决 XOR 问题 (如下表), 核函数为 $K(x, x_i) = (1 + x^T x_i)^2$.

序号	输入向量	输出
1	**[-1,-1]**	**-1**
2	**[-1,+1]**	**+1**
3	**[+1,-1]**	**+1**
4	**[+1,+1]**	**-1**

利用 Python 的 sklearn 库的 SVM 方法, 自定义核函数为 $K(x, x_i) = (1 + x^T x_i)^2$(具体见代码附件), 可以得到分类结果如下图: 可以看出此多项式核很好的将 XOR 问题的不同点做了分类, 并清晰的画出分类超曲面 (即曲线)。

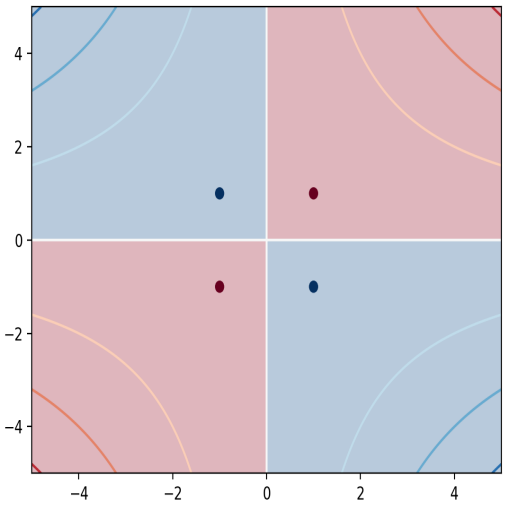

Figure 1: Result of SVM on XOR problem

2 Ex2

利用分类模型实现中 MNIST 手写数字 (0-9) 分类:

1. 数据集的可视化和统计分析;

2. 使用 SVM

在导入 mnist 数据集后, 首先对数据集进行可视化, 只需要对 DataFrame 格式的图片矩阵化, 然后进行 reshape 即可，具体结果如下图所示:

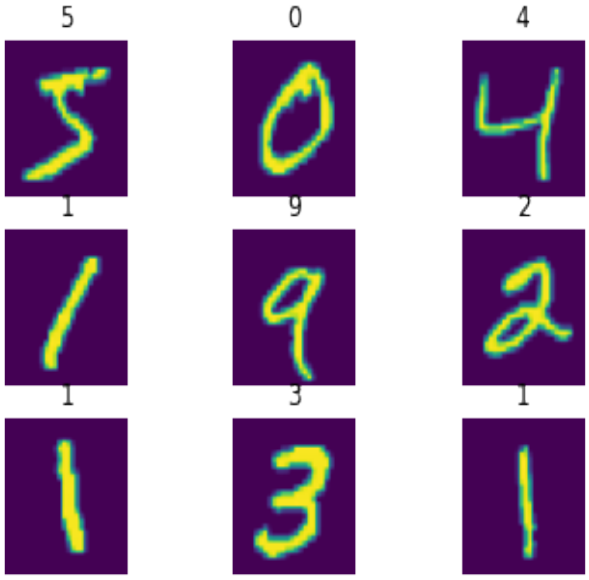

Figure 2: Some examples of mnist

之后, 对各个标签 (即数字 0 到 9) 对应的向量化图像的数目进行统计, 以判断是否存在数据差异过大的情况, 因为在数据差异过大的情况下直接对数据进行分类操作易出现问题. 经过统计, **各个标签对应的向量化图像的数目差距不大，且由于数据维数较高 (784 维)，数据的凸包几乎不可能不相交，故数据是线性不可分的，这是 mnist 数据集最重要的特点**. 具体如下图所示:

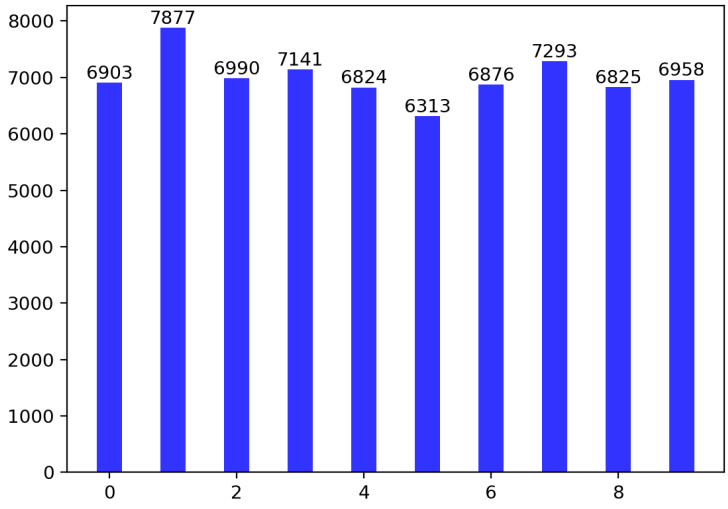

Figure 3: Number of data corresponding to different labels

为了更好说明这一步工作的重要性, 这里引入一个特殊的例子:

e.g. 1 *考虑同一个数据集即 mnisit 数据集, 这时任务考虑为二分类任务, 即区分对于一个手写数字, 是 0 还是非 0. 不妨假设 0 到 9 各个数字对应的图像数目是相等的, 此时我们训练分类器的方法为无论标签为哪个数字, 对应的输出结果均为非 0, 这样我们得到的结果准确率可达 90%, 但是召回率会非常低, 这说明当不同类规模差距较大时, 我们需要对其进行预处理.*

在此基础上, 利用留出法, 划出 70000 个数据中的后 10000 个数据为测试集, 其余数据为训练集, 选取 kernel 为 rbf, 即高斯核函数, 去训练模型, 并记录模型的训练与测试时间, 在 2.6 GHz 六核 Intel Core i7 的 cpu,python3.8 下训练时间和测试时间如下图所示:

```
[Running] python -u "/Users/zhangsiyuan/Desktop/ISTBI/深度学习与神经网络/homework3/zhangsiyuan_SVM_for_mnist.py"
训练时间为 216 s
测试时间为 81 s

[Done] exited with code=0 in 333.946 seconds
```

Figure 4: Training and testing time of SVM

最终模型在测试集上的准确率为 0.9792, 可以看出模型较好的对 10 种手写数字图像进行了分类. 同时模型的混淆矩阵如下图, 可以看出模型同时有着较高的召回率和精确率.

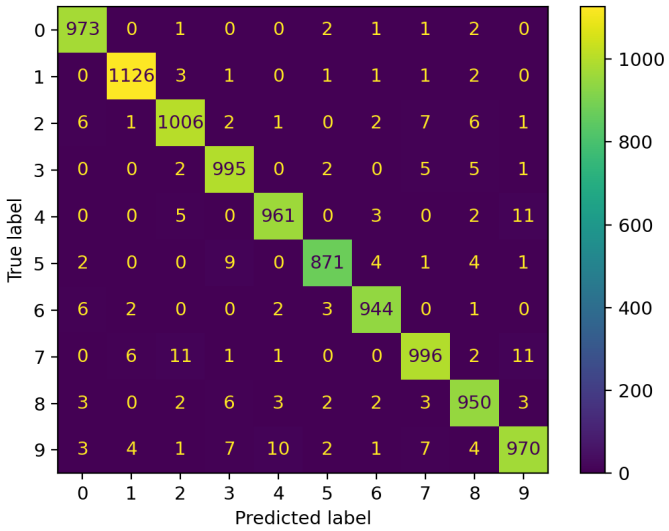

Figure 5: Confusion matrix

同时可以看出, 这个模型的速度直观上感受比较慢, 为了印证这一结论, 这里引入 KNN(K-Nearest-Neighbors) 分类器, 选择邻节点个数为 4, 以距离为权重, 在此基础上训练模型, 得到的测试集上的准确率为 0.9714, 此时准确率差距不大, 此时在相同计算力和计算环境下训练时间和测试时间如下图所示:

Figure 6: Training and testing time of KNN

可以看出此时, 训练和测试的时间大大缩短, **这里尝试给出个人的理解: 因为 SVM 是一个二分类的分类器, 即 SVM 是将数据分为正类和负类, 因此在执行多分类任务如本题时,SVM 会随机挑选两类作为分类目标, 最终会遍历所有的取值可能, 即 $C_{10}^2 = 45$ 种可能, 然后对于每个分类任务都会形成一次投票, 最终得票多的类为最终的分类. 对于一般的情况, 若假设有 n 类, 则共会执行 $C_n^2 = n(n+1)2$ 次二分类任务, 这大大削弱了程序的计算效率.**

References

[1] 李航. 统计学习方法 [M]. 清华大学出版社, 2012.

[2] Goodfellow, Ian, et al. Deep Learning[M]. MIT Press, 2016.

[3] Peter Harrington, 李锐…[等. 机器学习实战 [M]. 人民邮电出版社, 2013.