

作业 3

元敬哲 2022010657

2024 年 5 月 25 日

理论部分

1 单选题 (15 分)

1.1 D

1.2 C

1.3 D

1.4 D

1.5 B

2 计算题 (15 分)

2.1 给定两个类别的样本分别为:

$$\omega_1 : \{(3, 1), (2, 2), (4, 3), (3, 2)\}$$

$$\omega_2 : \{(1, 3), (1, 2), (-1, 1), (-1, 2)\}$$

试利用 LDA, 将样本特征维数压缩为一维。

解:

$$\text{对于该两类样本, 有: } S_1 = \begin{bmatrix} 0.5 & 0.25 \\ 0.25 & 0.5 \end{bmatrix}, S_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}。$$

$$\text{并且 } \mu_1 = [3, 2], \mu_2 = [0, 2], \mu = [1.5, 2]$$

其类内、类间散度矩阵为:

$$S_b = \begin{bmatrix} 2.25 & 0 \\ 0 & 0 \end{bmatrix}, S_w = \begin{bmatrix} 0.75 & 0.375 \\ 0.375 & 0.5 \end{bmatrix}$$

所以求矩阵 $S_w^{-1}S_b$ 的特征值的最大值所对应的特征向量。可求得

$$\lambda = 4.8, \text{ 投影方向 } w_{LDA} = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix} \text{ 所以压缩为一维后的样本变成:}$$

$$w_1 : \{3, 2.8, 5, 3.6\}, w_2 : \{2.6, 2, -0.2, 0.4\}$$

2.2 模型训练通常需要大量的数据，假设某采集的数据集包含 80% 的有效数据和 20% 的无效数据。采用一种算法判断数据是否有效，其中无效数据被成功判别为无效数据的概率为 90%，而有效数据被误判为无效数据的概率为 5%。如果某条数据经过该算法被判别为无效数据，则根据贝叶斯定理，这条数据是无效数据的概率是多少？(提示：全概率公式 $P(Y) = \sum_{i=1}^N P(Y|X_i)P(X_i)$)

解：

不妨记该数据集中某条数据被判无效数据为事件 Y，它是无效数据为事件 X，则：

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} = \frac{0.9 \times 0.2}{0.8 \times 0.05 + 0.2 \times 0.9} = 81.8\%$$

2.3 设有两类正态分布的样本集，第一类均值为 $\mu_1 = [2, -1]^T$ ，第二类均值为 $\mu_2 = [1, 1]^T$ 。两类样本集的协方差矩阵和出现的先验概率都相等： $\Sigma_1 = \Sigma_2 = \Sigma = \begin{bmatrix} 4 & 2 \\ 2 & \frac{4}{3} \end{bmatrix}$ ， $p(\omega_1) = p(\omega_2)$ 。试计算分类界面，并对特征向量 $x = [6, 2]^T$ 分类。

解：

因为两类样本的协方差矩阵和先验概率均相等，可以得出线性判别函数为： $g_{LDF}(x) = A_i^T x + b_i$ 。其中 $A_i = \Sigma_0^{-1} \mu_i$ ， $b_i = -\frac{1}{2} \mu_i^T \Sigma_0^{-1} \mu_i$ 。

所以

$$A_1 = [3.5, -6]^T, b_1 = -6.5$$

$$A_2 = [-0.5, 1.5]^T, b_2 = -0.5$$

判别函数 $g(x) = g_1(x) - g_2(x)$ 。所以决策面方程为： $g(x) = 0$ ，即

$$4x - 7.5y - 6 = 0$$

把特征向量 $x = [6, 2]^T$ 带入判断方程，可得 $g(x) = 3 > 0$ ，所以属于第一类。

2.4 给定异或的样本集

$$D = \{((0, 0)^T, -1), ((0, 1)^T, 1), ((1, 0)^T, 1), ((1, 1)^T, -1)\}$$

该样本集是线性不可分的，可采用如下所示的多项式函数

$\phi(\mathbf{x})$ 将样本 $D = \{(\mathbf{x}_n, y_n)\}$ 映射为 $D_\phi = \{(\phi(\mathbf{x}_n), y_n)\}$ ，其中 $\phi(\mathbf{x})$ 满足

$$\phi_1(\mathbf{x}) = 2(x_1 - 0.5)$$

$$\phi_2(\mathbf{x}) = 4(x_1 - 0.5)(x_2 - 0.5)$$

(1) 给出映射后的样本集；

(2) 在映射后的样本集中，设计一个线性 SVM 分类器，给出支持向量及分类界面。

解：

(1) 映射后的样本集为：

$$D_\phi = \{((-1, 1)^T, -1), ((-1, -1)^T, 1), ((1, -1)^T, 1), ((1, 1)^T, -1)\}$$

(2) 设计线性 SVM 分类器，可以将问题对偶地转化为优化目标：

$$L = \max_{\alpha} \left(\sum_{i=1}^4 \alpha_i - \frac{1}{2} \sum_{i=1}^4 \sum_{j=1}^4 \alpha_i \alpha_j y_i y_j x_i^T x_j \right)$$

目标函数对 α_i 分别求导，并令导数为 0，且满足约束条件。可得：

$$\begin{cases} 2\alpha_1 + 2\alpha_3 = 1 \\ 2\alpha_2 + 2\alpha_4 = 1 \\ -\alpha_1 + \alpha_3 + \alpha_2 - \alpha_4 = 0 \end{cases}$$

所以得出如下解：

$$\begin{cases} \alpha_1 = \frac{1}{4} \\ \alpha_2 = \frac{1}{4} \\ \alpha_3 = \frac{1}{4} \\ \alpha_4 = \frac{1}{4} \end{cases}$$

因此四个样本都是支持向量。 $w = \sum \alpha y \phi(x) = [0, -1]^T$ 。带入样本 x_1 ，可得偏置量 $b = 0$ 。因此分界面为： $y = 0$

2.5 使用 KMeans 算法对 2 维空间中的 6 个点

(0, 2), (2, 0), (2, 3), (3, 2), (4, 0), (5, 4) 进行聚类, 距离函数选择

欧氏距离 $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 。

(1) 起始聚类中心选择 (0,0) 和 (4,3), 计算聚类中心;

(2) 起始聚类中心选择 (1,4) 和 (3,1), 计算聚类中心。

解:

(1) 若起始聚类中心为 (0,0) 和 (4,3), 则点 (0,2), (2,0) 属于一个聚类, 点 (2,3), (3,2), (4,0), (5,4) 属于另一个聚类。下一次的聚类中心分别为 (1,1) 和 (3.5, 2.25)。以此为中心的聚类与上次相比不变, 所以聚类中心为 (1,1) 和 (3.5, 2.25)。

(2) 若起始聚类中心为 (1,4) 和 (4,3), 则点 (0,2), (2,3), (4,0) 属于一个聚类, 点 (2,0), (3,2), (5,4) 属于另一个聚类。下一次的聚类中心分别为 $(2, \frac{5}{3})$ 和 $(\frac{10}{3}, 2)$ 。以此为中心的聚类是: 点 (0,2), (2,3) 属于一个聚类, 点 (2,0), (3,2), (4,0), (5,4) 属于另一个聚类。所以聚类中心为 (1, 2.5) 和 (3.5, 1.5)。再以此为聚类中心的聚类不发生变化, 所以聚类中心为 (1, 2.5) 和 (3.5, 1.5)。

编程部分

3 编程作业报告

3.1 程序验证

运行 check.py 进行检查 svm_hw.py 下各模块前向传播和反向传播过程的正确性:

```
PS D:\清华文件\大二下\媒体与认知\hw3_release\hw3_release\code> python check.py
Linear successully tested!
Hinge successfully tested!
SVM_HINGE successfully tested!
PS D:\清华文件\大二下\媒体与认知\hw3_release\hw3_release\code> |
```

图 1: 验证 svm_hw.py

可见代码通过了检验。

3.2 数据预处理

在进行 SVM 的训练之前，需要利用补全好的 `data_preprocess.py`，读取特定的 2 类交通标志数据集，利用第 2 次编程作业预训练好的卷积网络从图像中提取 2048 点特征，并采用 PCA 降至 2 维，最后保存为相应的数据集文件 (`train.pt`、`val.pt`、`test.pt`)。本次作业采用交通标志数据集中的 B 和 C 两类。预处理部分结果如下：

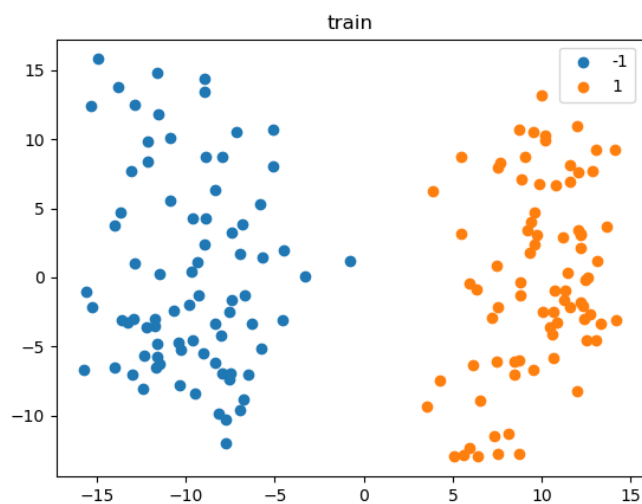


图 2: pre_train

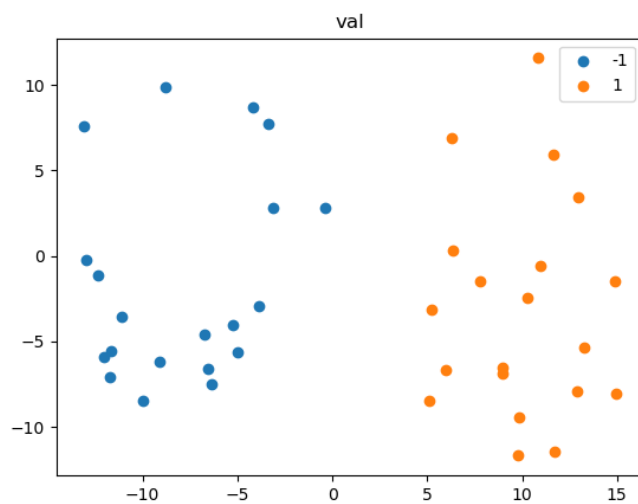


图 3: pre_val

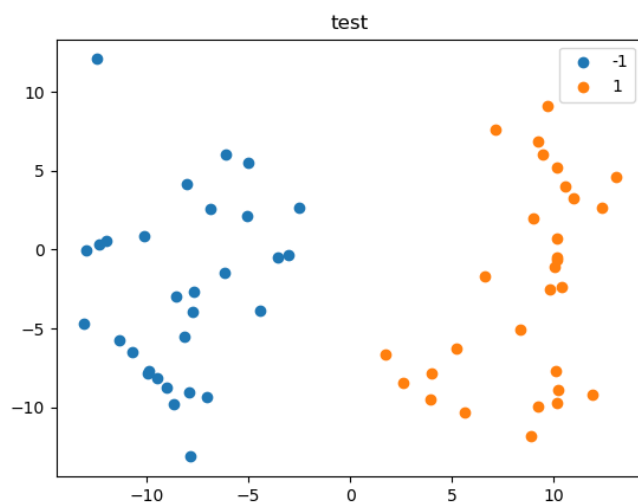


图 4: pre_test

可见两类样本点被彼此尽可能地分散在二维平面上。

3.3 训练、验证及测试

在完成 `train_svm.py` 和 `test_svm.py` 后，可以使用默认参数配置对模型进行训练和测试。在默认参数配置中，正则化系数 C 为 0.001。

(1) 训练模型: 利用 cpu 训练, 结果如下:

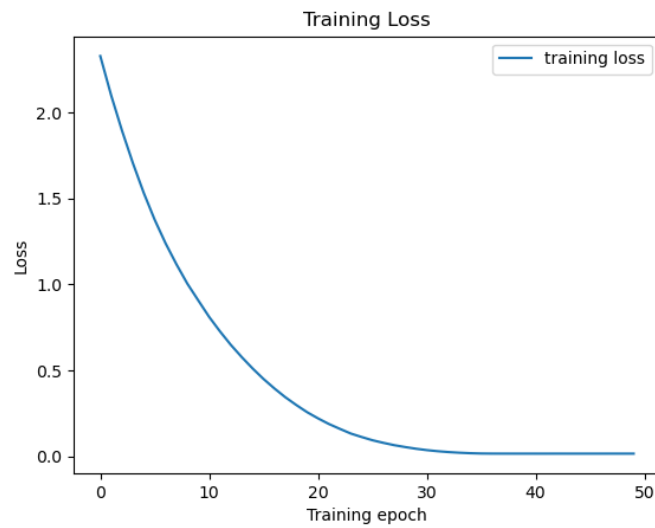


图 5: loss 曲线



图 6: 准确率曲线

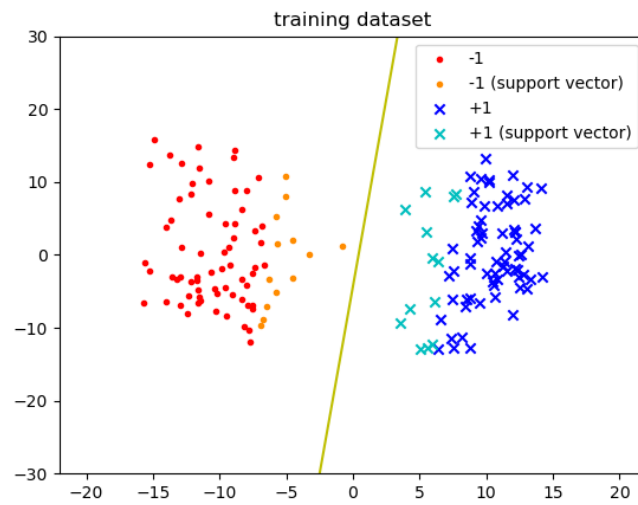


图 7: 训练集上可视化图片

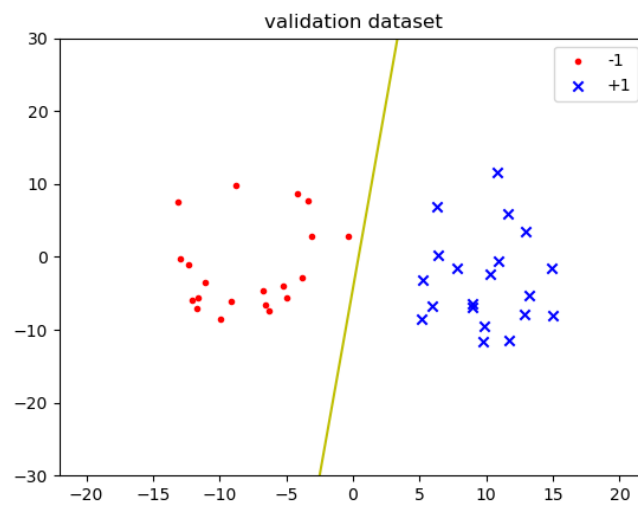


图 8: 验证集上可视化图片

(2) 测试模型: 利用 cpu 测试, 则执行如下命令:

```
PS D:\清华文件\大二下\媒体与认知\hw3_release\hw3_release\code> python test_svm.py --device cpu
Test accuracy = 100.0%
```

图 9: 分类准确率

3.4 调整正则化系数 C ，分析不同的 C 取值对分类效果的影响

分别设置不同的参数 $C=1e-6, 1e-3, 1$ ，对比不同 C 取值下的训练集及验证集上的准确率曲线、训练集及验证集的可视化图片（共 4 张图片），并记录在测试集上的分类准确率，如下：

1. $C=1e-6$

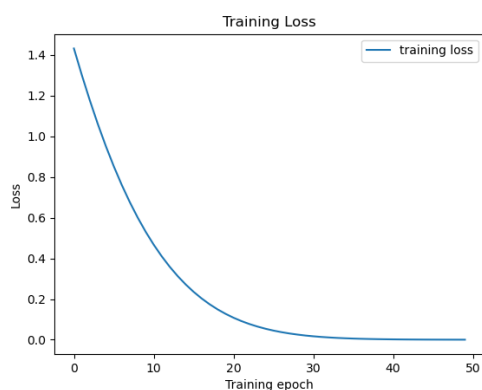


图 10: $C=1e-6$ 时 loss 曲线

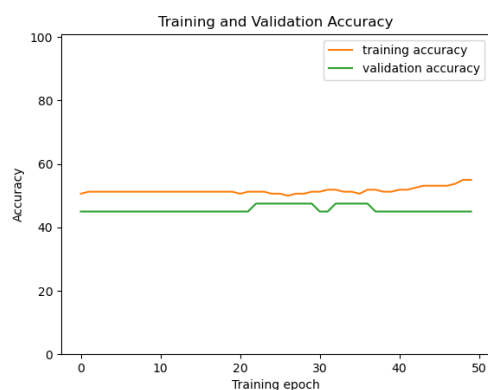


图 11: $C=1e-6$ 时准确率曲线

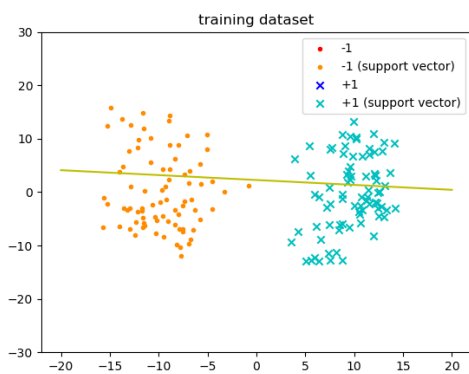


图 12: $C=1e-6$ 时训练集图片

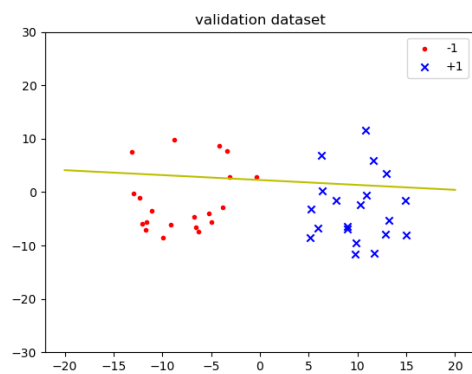
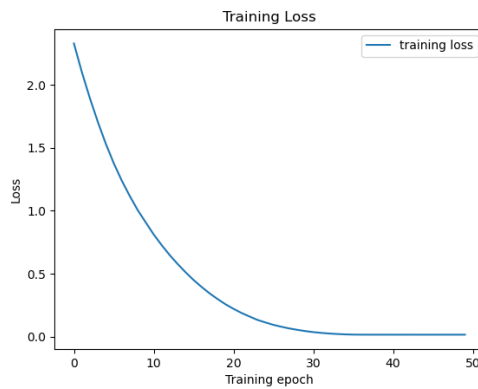
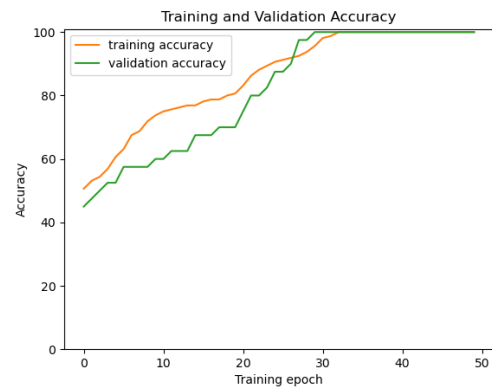
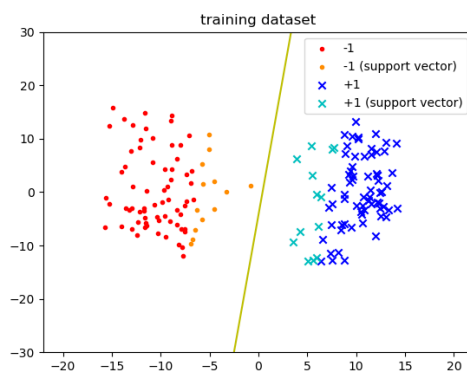
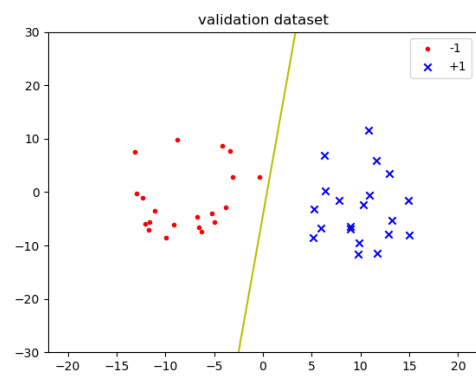


图 13: $C=1e-6$ 时验证集图片

```
PS D:\清华文件\大二下\媒体与认知\hw3_release\hw3_release\code> python test_svm.py --device cpu
Test accuracy = 58.3%
```

图 14: $C=1e-6$ 时分类准确率

2. $C=1e-3$

图 15: $C=1e-3$ 时 loss 曲线图 16: $C=1e-3$ 时准确率曲线图 17: $C=1e-3$ 时训练集图片图 18: $C=1e-3$ 时验证集图片

```
PS D:\清华文件\大二下\媒体与认知\hw3_release\hw3_release\code> python test_svm.py --device cpu
Test accuracy = 98.3%
```

图 19: $C=1e-3$ 时分类准确率

3. $C=1$

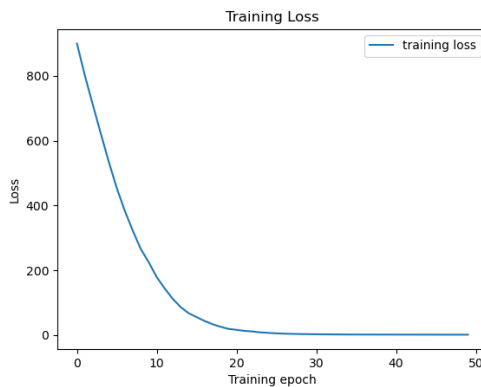


图 20: C=1 时 loss 曲线

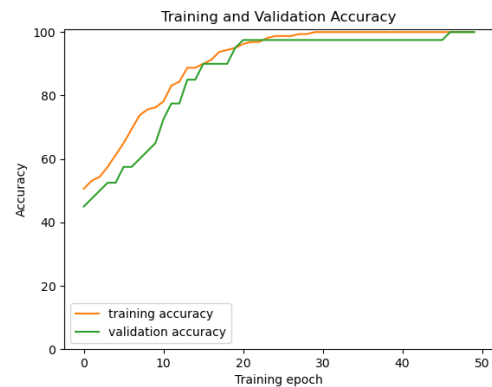


图 21: C=1 时准确率曲线

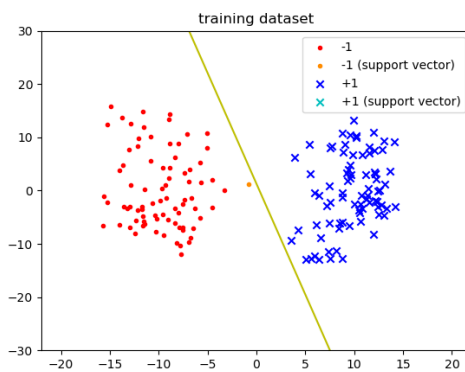


图 22: C=1 时训练集图片

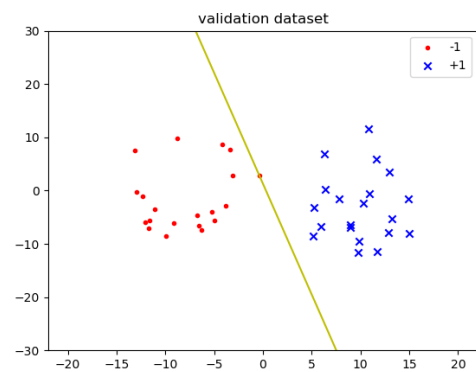


图 23: C=1 时验证集图片

```
PS D:\清华文件\大二下\媒体与认知\hw3_release\hw3_release\code> python test_svm.py --device cpu
Test accuracy = 98.3%
```

图 24: C=1 时分类准确率

由上面不同 C 取值下的可视化图片, 可见 C 取值越大, 拟合程度越高, 越容易过拟合。比如 $C=1$ 时, 此时模型误差要求很小, 因此可见训练集中几乎没有支持向量。划分的直线与两类的样本聚集区域边界上的点距离很近。此时产生过拟合, 导致了测试集中出现了一些错误。

而 C 取值越小时, 则会出现欠拟合。比如 $C=1$ 时, 此时模型误差要求很宽, 因此可见训练集中几乎全是支持向量。划分的直线横穿两类样本区域。此时产生欠拟合, 导致了测试集中只有一半的准确率。

因此 C 的值要合理选取, 才能使得数据被很好地线性分类, 比如 C

取值为 $1e-3$ 。

3.5 总结与反思

本次实验遇到了不少程序中的细节问题。比如 Traffic_Dataset 的文件路径以及在第 2 次作业中的 networks.py 在 self.conv_net 中调用 ConvBlock() 时遗漏了 use_batch_norm 参数。还有计算 out 和 label 时遇到 “Jacobian mismatch for output 0 with respect to input 0” 类似的问题。但都在助教的文档帮助下一一解决了。

本次实验使用了大语言模型来帮助我理解了一些 torch 中函数的输入输出，以及学习一些 python 其他函数的用法，比如求矩阵范数。