

本科生实验报告

学生姓名：谢汶余

学号：23336263

实验名称：机器学习作业 1

问题:

根据提供的数据，训练一个采用在不同的核函数的支持向量机 SVM 的 2 分类器，并验证其在测试数据集上的性能。

要求:

- 1) 考虑两种不同的核函数：i) 线性核函数; ii) 高斯核函数
- 2) 可以直接调用现成 SVM 软件包来实现
- 3) 手动实现采用 hinge loss 和 cross-entropy loss 的线性分类模型，并比较它们的优劣

一、SVM 模型的一般理论

支持向量机 (SVM) 是一种强大的监督学习算法，主要用于分类和回归问题。其核心思想是找到一个最优超平面，将不同类别的数据分开，并最大化类别之间的边界。对于线性可分的数据，SVM 通过寻找一个超平面来实现分类（线性核）；对于线性不可分的数据，SVM 通过核函数将数据映射到高维空间，使其在高维空间中线性可分（高斯核）。其学习策略是间隔最大化，最终可转化为一个凸二次规划问题的求解。支持向量机的学习算法是求解凸二次规划的最优化算法，具有完善的数学理论基础。这些理论构成了 SVM 模型的核心，广泛应用于各种机器学习任务中。

二、采用不同核函数的模型性能比较及分析

本实验采用了线性核函数 (LINEAR) 以及高斯核函数 (RBF)：

LINEAR: $K(x_i, x_j) = x_i^T x_j$ ，只能用来处理线性可分数据，计算速度快，适合高维数据（如本实验的 $28 \times 28 = 784$ 维图像特征）。其决策边界是直线（二维）或超平面（高维）。

RBF: $K(x_i, x_j) = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}$ ，通过调节参数 σ 控制径向作用范围，能处理非线性数据，决策边界更灵活

性能差异原因：手写数字 8 和 9 的特征存在非线性差异（如局部轮廓的弯曲程度），高斯核能捕捉这些非线性关系，因此在实验中准确率会高于线性核。但线性核计算效率更高，且在高维数据上不易过拟合。

三、采用 hinge loss 的线性分类模型与 SVM 模型之间的关系

采用铰链损失函数的线性分类模型和 SVM 模型关系密切，两者损失函数：

HINGE LOSS: $L(y, f(x)) = \max(0, 1 - y \cdot f(x))$ ，其中 y 取值为 -1 或 1， $f(x) = w^T x + b$ ，当样本被分类正确且足够自信的时候（间隔大）的时候不惩罚，只惩罚分类错误或分类正确但不够自信的样本（间隔小），且距离越大，损失越大

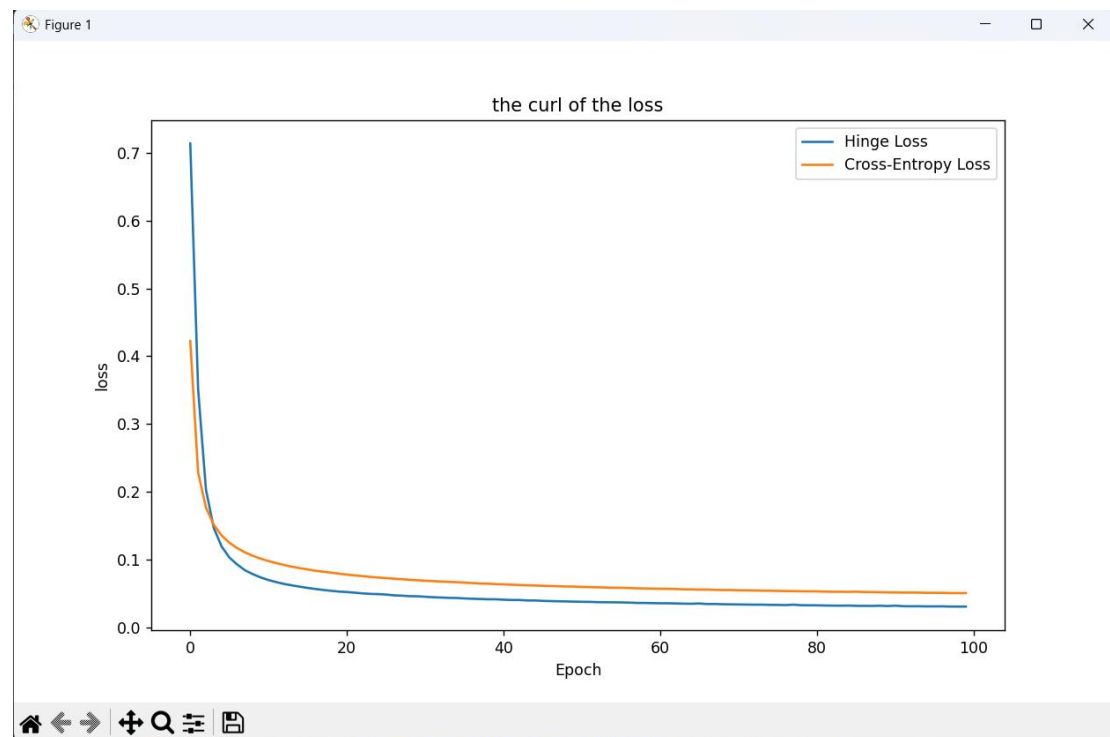
而线性 SVM 是采用 hinge loss 且附加 L2 正则化的线性分类模型。

四、采用 hinge loss 线性分类模型与 cross-entropy loss 线性分类模型比较

损失函数不同；

标签不同，hinge loss 的是 -1 和 1，交叉熵损失函数是 0 和 1；

优化目标不同，hinge loss 关注最大化边界，交叉熵关注输出贴近真实标签概率；



损失函数曲线如上：

hinge loss 训练前期损失从 0.7 陡峭下降，快速收敛到 0.1 以下，后期趋于平稳。这是因为 Hinge Loss 对“易分样本”（分类正确且间隔足够大的样本）损失为 0，仅关注“难分样本”的优化，因此在模型初步学到有效分类边界后，损失会快速下降并稳定；而交叉熵函数损失从 0.4 平滑下降，整体趋势更均匀。这是因为 Cross-Entropy Loss 对所有样本的预测偏差都敏感（无论样本是否易分），需要逐步优化每个样本的概率预测，因此收敛过程更平缓。

五、训练过程（包括初始化方法、超参数参数选择、用到的训练技巧等）

1. 数据预处理

数据加载：从 `processed_8_9` 目录加载手写数字 8 和 9 的训练集（11800 样本）和测试集（1983 样本），图像尺寸为 28×28 。

特征转换：将 28×28 图像展平为 784 维向量（便于线性模型处理，因为 SVM 的数学逻辑是“对每个特征计算权重”，需要输入是“每个样本一行，每个特征一列”的矩阵格式（称为“样本 - 特征矩阵”）。 28×28 的二维图是“空间结构”，模型无法直接处理，展平后才能变成“特征向量”。）。

标签转换：将原始标签（8→0，9→1）转换为二分类标签。

标准化：使用 `StandardScaler` 对特征进行标准化（均值为 0，标准差为 1），避免特征尺度差异影响模型训练。

2. 初始化方法

权重初始化：线性模型的权重 w 采用随机正态分布，偏置 b 初始化为 0，避免初始值过大导致梯度爆炸。

SVM 初始化：线性核 $C=0.5$ ，高斯核 $\gamma=scale$ 。

3. 超参数选择

学习率：Hinge Loss 模型使用 0.0001，Cross-Entropy 模型使用 0.001（收敛更稳定）。

batch size：128（平衡训练效率与参数更新稳定性）。

训练轮次：150（使用早停机制，避免无效训练）。

早停耐心：15（连续 15 轮测试准确率无提升则停止训练，防止过拟合）。

4. 训练技巧

批量梯度下降（Mini-batch SGD）：随机打乱数据并分批次训练，平衡收敛速度与内存效率。

数值稳定性优化：Hinge Loss 模型中过滤空梯度样本，Cross-Entropy 模型中限制 sigmoid 输入范围，以避免指数溢出。

六、实验结果、分析及讨论

四组模型皆采用了测试集准确率、分类报告以及混淆矩阵三个维度衡量结果

LINEAR 的 SVM：

--- 线性核SVM ---

测试集准确率: 0.9713

分类报告:

	precision	recall	f1-score	support
0	0.97	0.97	0.97	974
1	0.97	0.97	0.97	1009
accuracy			0.97	1983
macro avg	0.97	0.97	0.97	1983
weighted avg	0.97	0.97	0.97	1983

混淆矩阵:

```
[[944  30]
 [ 27 982]]
```

rbfSVM:

--- 高斯核SVM ---

测试集准确率: 0.9839

分类报告:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	974
1	0.99	0.97	0.98	1009
accuracy			0.98	1983
macro avg	0.98	0.98	0.98	1983
weighted avg	0.98	0.98	0.98	1983

混淆矩阵:

```
[[969   5]
 [ 27 982]]
```

Hinge loss 线性分类模型:

```

--- Hinge Loss线性模型 ---
Epoch 10/100 - 损失: 0.0738 - 测试准确率: 0.9743
Epoch 20/100 - 损失: 0.0530 - 测试准确率: 0.9783
Epoch 30/100 - 损失: 0.0461 - 测试准确率: 0.9803
Epoch 40/100 - 损失: 0.0417 - 测试准确率: 0.9818
Epoch 50/100 - 损失: 0.0382 - 测试准确率: 0.9818
Epoch 60/100 - 损失: 0.0360 - 测试准确率: 0.9813
Epoch 70/100 - 损失: 0.0342 - 测试准确率: 0.9818
Epoch 80/100 - 损失: 0.0328 - 测试准确率: 0.9829
Epoch 90/100 - 损失: 0.0317 - 测试准确率: 0.9823
Epoch 100/100 - 损失: 0.0309 - 测试准确率: 0.9823
最终测试准确率: 0.9823
分类报告:

```

	precision	recall	f1-score	support
0	0.98	0.98	0.98	974
1	0.98	0.98	0.98	1009
accuracy			0.98	1983
macro avg	0.98	0.98	0.98	1983
weighted avg	0.98	0.98	0.98	1983

```

混淆矩阵:
[[957 17]
 [ 18 991]]

```

交叉熵函数线性分类模型:

```

--- Cross-Entropy Loss线性模型 ---
Epoch 10/100 - 损失: 0.1014 - 测试准确率: 0.9713
Epoch 20/100 - 损失: 0.0794 - 测试准确率: 0.9723
Epoch 30/100 - 损失: 0.0696 - 测试准确率: 0.9743
Epoch 40/100 - 损失: 0.0640 - 测试准确率: 0.9763
Epoch 50/100 - 损失: 0.0604 - 测试准确率: 0.9773
Epoch 60/100 - 损失: 0.0573 - 测试准确率: 0.9773
Epoch 70/100 - 损失: 0.0553 - 测试准确率: 0.9768
Epoch 80/100 - 损失: 0.0534 - 测试准确率: 0.9768
Epoch 90/100 - 损失: 0.0519 - 测试准确率: 0.9783
Epoch 100/100 - 损失: 0.0508 - 测试准确率: 0.9788
最终测试准确率: 0.9788
分类报告:

```

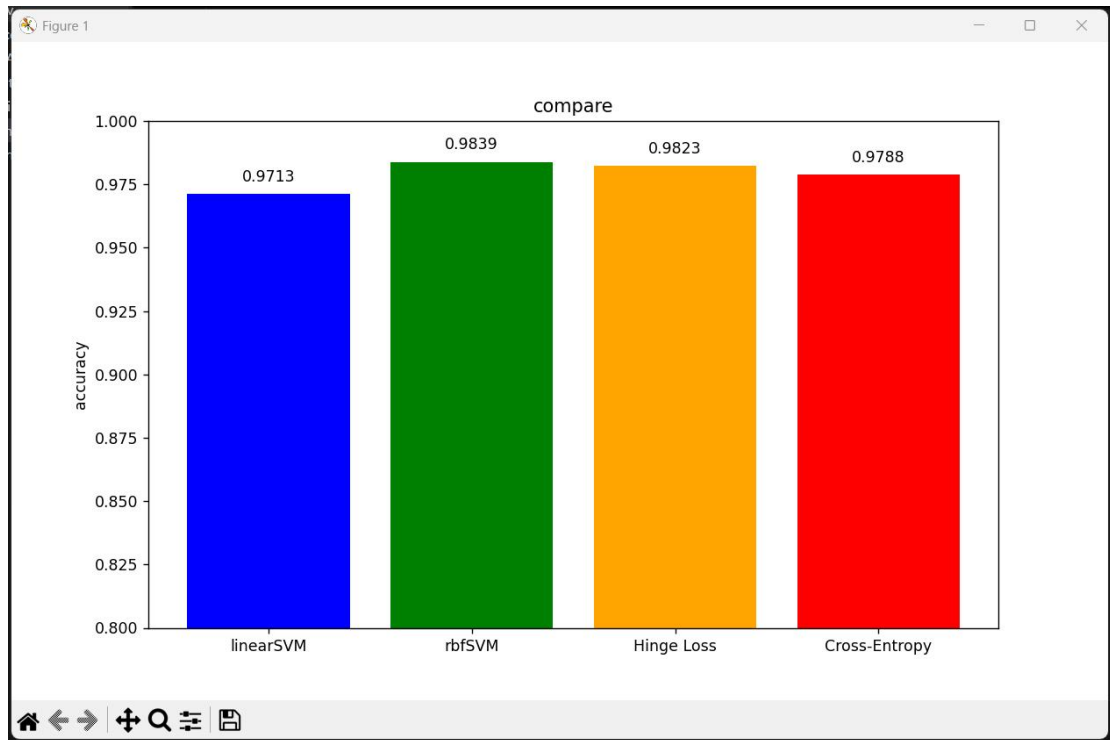
	precision	recall	f1-score	support
0	0.98	0.98	0.98	974
1	0.98	0.98	0.98	1009
accuracy			0.98	1983
macro avg	0.98	0.98	0.98	1983
weighted avg	0.98	0.98	0.98	1983

```

混淆矩阵:
[[953 21]
 [ 21 988]]

```

总比较:



分析：

- 1、核函数中高斯核函数最优，因为其能捕捉 8、9 的非线性特征
- 2、铰链损失函数比交叉熵函数更优，因为其更关注最大化边界