作业 1：线性模型和支持向量机

清华大学软件学院
机器学习，2025 年秋季学期

1 介绍

本次作业需提交两部分：说明文档（PDF 形式）和 Python 的源代码。请仔细阅读以下注意事项：

- 本次作业满分为 110 分，若总得分超过 100 分，则按 100 分计。
- 作业按小问逐点评分，请在说明文档中按题号清晰作答，便于助教批改。例如：

$$\textbf{2.2.1}\quad J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\big(h_\theta(x_i) - y_i\big)^2 + \cdots$$

- 除非特别说明，禁止直接调用机器学习开源库（如 `sklearn`、`pytorch` 等）。
- 请熟练使用 `numpy` 及其广播（broadcasting）机制。若用显式 `for` 循环实现本可向量化的矩阵运算，编程部分将不予计分。
- 代码文件中须保留与题目相关的全部实现。请通过清晰的变量命名与必要注释保证可读性；若可读性较差，将酌情扣分。
- 在 PDF 中说明完成作业过程中与他人交流或参考资料的具体情况：与他人交流需注明姓名（网络论坛可填用户名）；参考网络资料需附具体链接。
- 如使用大模型辅助作业完成，请在 PDF 中声明作业中你使用了大模型的部分和使用的方式。
- 严禁抄袭他人作业（含代码与文档）或公开自己的作业；一经查实，最高可扣至 −100 分（倒扣本次作业全部分值）。

2 线性模型与梯度下降（50pt）

在本题中，你将通过实现**梯度下降法（Gradient Descent）**来求解**岭回归（Ridge Regression）**问题。

2.1　特征归一化（4pt）

在实际任务中，若各维特征的量级差异较大，梯度下降的收敛会显著变慢；同时，在使用正则化时，量级较大的特征对正则项影响更强。因此需要进行特征归一化。常用做法是在**训练集**上对每个特征进行仿射变换，将其映射到区间 $[0,1]$；并对**测试集**施加与训练集一致的变换。

1. 补全函数 `split_data`，将数据集划分为训练集与测试集。
2. 补全函数 `feature_normalization`，实现特征归一化。

2.2　目标函数与梯度（10pt）

在**线性回归**（Linear Regression）中，我们考虑以下线性假设空间：

$$h_\theta : \mathbb{R}^d \to \mathbb{R}, \quad h_{\theta,b}(x) = \theta^T x + b,$$

其中 $\theta, x \in \mathbb{R}^d$，$b$ 为偏置项（bias）。

为方便推导与实现，我们通常在输入向量 x 的末尾添加一个恒为 1 的分量，以吸收偏置项 b。此时，模型可改写为：

$$h_\theta(x) = \theta^T x, \quad \text{其中}\theta, x \in \mathbb{R}^{d+1}.$$

我们希望找到合适的参数向量 θ，使得均方误差（Mean Squared Error, MSE）最小化：

$$J_{\text{MSE}}(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x_i) - y_i\right)^2,$$

其中训练样本为 $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m) \in \mathbb{R}^{d+1} \times \mathbb{R}$。

岭回归（Ridge Regression）是在线性回归的基础上引入 L_2 正则化项的模型。其目标函数定义为：

$$J(\theta) = J_{\text{MSE}}(\theta) + \lambda R(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x_i) - y_i\right)^2 + \lambda\theta^T\theta,$$

其中 $\lambda > 0$ 为正则化系数，用以控制模型的复杂度。λ 越大，模型参数受到的约束越强，从而能在一定程度上防止过拟合。

1. 将训练数据的特征记作

$$X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_m^T \end{pmatrix} \in \mathbb{R}^{m\times(d+1)}, \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m.$$

   请写出 $J(\theta)$ 的矩阵形式表达式。

2. 补全函数 `compute_regularized_square_loss`，在给定 θ 时计算 $J(\theta)$ 的数值。
3. 请写出 $J(\theta)$ 对 θ 的梯度（矩阵形式），并解释梯度的含义。
4. 补全函数 `compute_regularized_square_loss_gradient`，在给定 θ 时计算梯度 $\nabla J(\theta)$。

为了验证梯度计算是否正确，可以使用**数值梯度检验（numerical gradient checking）**。对于可导函数 $J(\theta)$，在某个方向 h 上的方向导数可由下式近似：

$$\frac{\partial J}{\partial h} \approx \frac{J(\theta + \varepsilon h) - J(\theta - \varepsilon h)}{2\varepsilon},$$

其中 $\varepsilon > 0$ 为一个足够小的常数。

在实际操作中，可以依次取 $h = e_1, e_2, \ldots, e_{d+1}$（即各坐标方向的单位向量），计算每一维的近似梯度，并将它们拼接得到 $\nabla J(\theta)$ 的近似值。

代码中已提供函数 `grad_checker`，你可以利用该函数检验自己实现的梯度计算函数是否正确（存在近似计算，因此在一定误差范围内即为正常情况）。

2.3 梯度下降（10pt）

1. 在最小化 $J(\theta)$ 时，考虑从当前参数 θ 沿方向 $h \in \mathbb{R}^{d+1}$ 前进一步至 $\theta + \eta h$，其中 $\eta > 0$ 为步长。请用梯度写出目标函数值变化的近似表达式 $J(\theta + \eta h) - J(\theta)$，思考 h 为哪一前进方向时目标函数下降最快，并据此写出梯度下降中更新 θ 的表达式。
2. 程序中的 `main` 函数已加载数据、完成了训练集与测试集的划分以及特征归一化。请补全函数 `gradient_descent`，实现**梯度下降（Gradient Descent）**算法，使模型能在训练集上进行优化。
3. 选择合适的步长。固定正则化系数 $\lambda = 0$，从步长 $\eta = 0.1$ 开始，尝试多种固定步长（至少包括 $0.5, 0.1, 0.05, 0.01$），观察目标函数随训练迭代变化的曲线，记录不同步长下的收敛速度，并指出：
   - 哪个步长收敛最快；
   - 哪个步长会导致发散。

   请绘制目标函数 $J(\theta)$ 随迭代次数变化的曲线，并在图例中注明不同步长对应的曲线。

2.4 模型选择（8pt）

1. 我们可以通过**验证集**上的均方误差 $J_{\mathrm{MSE}}(\theta)$ 来选择合适的超参数。由于目前没有单独的验证集，请补全函数 `K_fold_split_data`，将训练集划分为 K 组（不妨令 $K = 5$），以便进行 K 折交叉验证。每一折中使用 $K - 1$ 份数据作为训练集，剩余 1 份作为验证集。
2. 补全函数 `K_fold_cross_validation`，实现 K 折交叉验证。请在不同超参数下运行模型，搜索最优超参数组合。搜索范围至少包括：

   $$\text{步长}\eta \in \{0.05, 0.04, 0.03, 0.02, 0.01\}, \quad \text{正则化系数}\lambda \in \{10^{-7}, 10^{-5}, 10^{-3}, 10^{-1}, 1, 10, 100\}.$$

   请用表格汇报不同超参数下模型在验证集上的均方误差，并报告最优超参数 (η^*, λ^*) 及其对应的测试集均方误差。

2.5 随机梯度下降（11pt）

当训练数据集规模非常大时，**批量梯度下降**（Batch Gradient Descent）每次更新参数都需要遍历全部样本，计算代价高、收敛速度慢。为提升效率，实际应用中通常采用**随机梯度下降算法**

（SGD, Stochastic Gradient Descent）。设第 i 个样本的平方误差为

$$f_i(\theta) = \big(h_\theta(x_i) - y_i\big)^2,$$

则总体目标函数为

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} f_i(\theta) + \lambda\,\theta^T\theta,$$

其中 $\lambda > 0$ 为正则化参数。

在 SGD 中，每一步仅使用一个**小批量**（mini-batch）样本集合 $\{(x_{i_k}, y_{i_k})\}_{k=1}^n$ 来近似整体目标。设 $n \ll m$，且索引 i_k 从 $\{1, 2, \ldots, m\}$ 中独立均匀采样，则对应的**小批量目标函数**为

$$J_{\mathrm{SGD}}(\theta) = \frac{1}{n}\sum_{k=1}^{n} f_{i_k}(\theta) + \lambda\,\theta^T\theta,$$

并使用 $\nabla J_{\mathrm{SGD}}(\theta)$ 作为 $\nabla J(\theta)$ 的随机近似来更新参数。

1. 请写出 $J_{\mathrm{SGD}}(\theta)$ 对应的梯度表达式 $\nabla J_{\mathrm{SGD}}(\theta)$。
2. 请证明随机梯度 $\nabla J_{\mathrm{SGD}}(\theta)$ 是 $\nabla J(\theta)$ 的**无偏估计**。即证明：

$$\mathbb{E}_{i_1, i_2, \ldots, i_n}\big[\nabla J_{\mathrm{SGD}}(\theta)\big] = \nabla J(\theta).$$

   （提示：利用期望的线性性和样本独立同分布的性质，分别取每个样本梯度的期望并合并。）
3. 补全函数 `stochastic_grad_descent`，实现**随机梯度下降算法**。
4. 随机梯度下降具有较强的**噪声**，其训练曲线通常较为震荡，模型收敛速度与批大小密切相关。请固定 $\lambda = 0$，并根据第 2.3.3 或 2.4.2 小节的结果选定一个合适的步长 η。从批大小 1 开始，依次尝试多种不同的批大小（如 $1, 4, 8, 16, 32, \ldots$），观察并记录训练过程中的曲线变化。注意：由于小批量训练损失噪声较大，不能直接用其判断收敛，应当在**验证集**上计算**全批量损失**来评估模型是否收敛。开始训练前，请使用 `split_data` 重新划分训练集与验证集（不需要使用 `K_fold_split_data`）。

2.6 解析解（7pt）

1. 对于岭回归模型，我们可以直接推导出其解析解。请推导出岭回归模型的解析解表达式，并实现函数 `analytical_solution`。
2. 正则化往往可以有效避免过拟合。请用表格记录不同正则化系数 λ 的岭回归模型解析解在测试集上的均方误差，展现出正则化对于避免过拟合的有效性。
3. 请从计算时间、测试集均方误差等角度比较梯度下降类方法与解析解。思考：在当前任务中，机器学习优化方法是否优于解析解？请结合结果进行分析与说明。

3 Softmax 回归（10pt）

线性模型不仅可以用于回归任务，也可以扩展到**多分类**问题中。在这一题中，你将推导 **Softmax 回归（Softmax Regression）**模型的损失函数与梯度，并分析其性质。

设分类问题共有 K 个类别，输入样本为 $\mathbf{x} \in \mathbb{R}^n$，模型参数包括权重矩阵 $\mathbf{W} \in \mathbb{R}^{K\times n}$ 和偏置项 $\mathbf{b} \in \mathbb{R}^K$。模型的线性输出为：

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}.$$

为了将该线性输出映射为类别概率分布，我们使用 **Softmax 函数**：

$$\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_K]^T, \qquad \hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)}.$$

其中，$\hat{y}_i$ 表示样本 $\mathbf{x}$ 被模型预测为第 i 类的概率，满足 $\sum_{i=1}^K \hat{y}_i = 1$。

将样本的真实标签表示为**独热（one-hot）向量**形式：

$$\mathbf{y} = [y_1, y_2, \cdots, y_K]^T,$$

其中 $y_k = 1$ 表示样本真实属于第 k 类，其余分量 $y_i = 0$。Softmax 回归常采用**交叉熵损失函数（Cross-Entropy Loss）**作为优化目标：

$$\mathcal{L} = -\log \hat{y}_k = -\mathbf{y}^T \log \hat{\mathbf{y}}.$$

这一损失函数可理解为模型预测分布 $\hat{\mathbf{y}}$ 与真实分布 $\mathbf{y}$ 之间的负对数似然距离。

1. 将损失函数 $\mathcal{L}$ 视为 $\mathbf{z}$ 的函数。请推导 $\mathcal{L}$ 关于 $\mathbf{z}$ 的梯度 $\frac{\partial \mathcal{L}}{\partial \mathbf{z}}$。
2. 将损失函数 $\mathcal{L}$ 视为 $\mathbf{W}$ 和 $\mathbf{b}$ 的函数。请分别推导 $\mathcal{L}$ 关于 $\mathbf{W}$ 和 $\mathbf{b}$ 的梯度。
3. 设 $f(\mathbf{x})$ 是定义在 $\mathbb{R}^n$ 上的可二阶连续偏导的标量函数，其**海森矩阵**（Hessian Matrix）定义为：

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

   请写出 $\mathcal{L}$ 关于 $\mathbf{z}$ 的海森矩阵 $\mathbf{H} = \frac{\partial^2 \mathcal{L}}{\partial \mathbf{z}^2}$。
4. 请证明上述海森矩阵 $\mathbf{H}$ 是**半正定**的。这一结论说明 Softmax + 交叉熵损失函数在参数空间中是**凸的**（convex），从而具有唯一的全局最优解。

4 支持向量机（50pt）

在本题中，你将深入理解**支持向量机（Support Vector Machine, SVM）**的基本原理，包括硬间隔与软间隔支持向量机的推导，核方法的扩展应用，以及在真实文本数据上的分类实验。

4.1 硬间隔支持向量机（12pt）

给定一个线性可分的数据集 $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m) \in \mathbb{R}^d \times \{-1, +1\}$，硬间隔支持向量机旨在寻找一个能够**完全正确分类**样本的线性超平面：

$$w^T x + b = 0, \quad \text{其中} w \in \mathbb{R}^d,\ b \in \mathbb{R}.$$

使得：

$$y_i(w^T x_i + b) > 0, \quad 1 \le i \le m.$$

也就是说，所有标记为 $y = 1$ 的样本位于超平面的一侧，而标记为 $y = -1$ 的样本位于另一侧。SVM 不仅要求样本可分，还希望找到**间隔（margin）最大**的分类超平面，以提高模型的泛化能力。

1. 间隔（margin）定义为两类样本点分别与分类超平面的最近距离之和，可表示为

$$\max_{w \in \mathbb{R}^d,\, b \in \mathbb{R}} \gamma \quad \text{s.t.} \quad y_i \frac{w^T x_i + b}{\|w\|_2} \ \ge\ \gamma, \ \ 1 \le i \le m,$$

   其中 $\dfrac{w^T x_i + b}{\|w\|_2}$ 为点 x_i 到超平面 $w^T x + b = 0$ 的**有向距离**。请说明上述问题**等价于**下面的带约束二次优化问题：

$$\begin{aligned} \min_{w \in \mathbb{R}^d,\, b \in \mathbb{R}} \quad & \frac{1}{2}\|w\|_2^2 \\ \text{s.t.} \quad & y_i\left(w^T x_i + b\right) \ \ge\ 1, \quad 1 \le i \le m. \end{aligned} \tag{1}$$

   设拉格朗日乘子 $\mu_i \ge 0$ $(1 \le i \le m)$ 对应约束 $y_i(w^T x_i + b) \ge 1$，则问题 (1) 的拉格朗日函数为

$$L(w, b, \mu) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{m} \mu_i \big[y_i(w^T x_i + b) - 1\big]. \tag{2}$$

2. 请根据 (1)–(2) 写出硬间隔 SVM 的 KKT 条件，包含：原始可行性、对偶可行性、互补松弛条件以及驻点条件。
3. 证明满足 KKT 条件的最优解 w 一定可表示为训练样本的线性组合：

$$w = \sum_{i=1}^{m} \alpha_i\, y_i\, x_i, \quad \alpha_i = \mu_i \ge 0.$$

   若某 $\alpha_i \ne 0$，称对应样本 x_i 为**支持向量**。请进一步说明：所有支持向量均位于分隔边界

$$w^T x + b = \pm 1$$

   上。

4.2 软间隔支持向量机（10pt）

线性可分是一种理想化的假设，然而在真实的数据集中，由于噪声、标注误差或样本分布复杂等原因，往往**无法保证所有样本均被正确分类**。为此，支持向量机（SVM）在实际中会引入**软间隔（Soft Margin）**思想，允许部分样本点违反分类约束，但通过惩罚项控制违约程度，从而在"间隔最大化"与"误差最小化"之间取得平衡。

软间隔 SVM 的优化问题可表示为：

$$
\begin{aligned}
\min_{w\in\mathbb{R}^d,\, b\in\mathbb{R},\, \xi\in\mathbb{R}^m} \quad & \frac{\lambda}{2}\|w\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i^p \\
\text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \\
& \xi_i \geq 0, \quad 1 \leq i \leq m,
\end{aligned} \tag{3}
$$

其中，ξ_i 为**松弛变量（slack variable）**，用于度量第 i 个样本对约束的违反程度；参数 $p \geq 1$ 控制对违约样本的惩罚力度，$\lambda > 0$ 为正则化系数，平衡模型复杂度与分类误差。

1. 请写出 $p \geq 1$ 时该优化问题的**拉格朗日函数（Lagrangian）**。
2. 当 $p = 1$ 时，求解该问题的**对偶形式（Dual Form）**。
3. 在 $p = 1$ 时，原始优化问题可改写为如下形式：

$$
\min_{w\in\mathbb{R}^d,\, b\in\mathbb{R}} \frac{\lambda}{2}\|w\|^2 + \frac{1}{m}\sum_{i=1}^{m}\max\{0,\, 1 - y_i(w^T x_i + b)\}. \tag{4}
$$

   其中 $\max\{0, 1 - y_i(w^T x_i + b)\}$ 即为**合页损失（Hinge Loss）**。

   在实际求解中，除了使用对偶方法（如 SMO 算法）外，也可以采用**次梯度下降（Subgradient Descent）**进行优化。

   请证明：当我们定义单样本损失函数

$$
J_i(w, b) = \frac{\lambda}{2}\|w\|^2 + \max\{0,\, 1 - y_i(w^T x_i + b)\},
$$

   其关于参数 w 和 b 的次梯度可分别表示为：

$$
\partial J_i|_w = \begin{cases} \lambda w - y_i x_i, & \text{若} y_i(w^T x_i + b) < 1, \\ \lambda w, & \text{若} y_i(w^T x_i + b) \geq 1, \end{cases} \qquad \partial J_i|_b = \begin{cases} -y_i, & \text{若} y_i(w^T x_i + b) < 1, \\ 0, & \text{若} y_i(w^T x_i + b) \geq 1. \end{cases}
$$

4.3 核方法（8pt）

在实际应用中，许多分类问题是**非线性可分**的，即在原始特征空间中无法通过线性超平面进行良好划分。为了解决这类问题，可以引入**核技巧（Kernel Trick）**：将输入数据 $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T \in \mathcal{X}$ 通过**非线性映射** $\boldsymbol{\Phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_n(\boldsymbol{x}))^T$ 投影到高维特征空间中，再在该空间中寻找线性分类超平面。定义由基函数诱导的核函数为：$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\Phi}(\boldsymbol{x}_1) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_2)$，从而可以在不显式计算高维映射的情况下，用核函数直接替代点积操作，大幅降低计算复杂度。

1. 设对称函数 $k(\boldsymbol{x}, \boldsymbol{x}') = \cos\angle(\boldsymbol{x}, \boldsymbol{x}')$ 定义在 $\mathbb{R}^n \times \mathbb{R}^n$ 上，其中 $\angle(\boldsymbol{x}, \boldsymbol{x}')$ 表示向量 $\boldsymbol{x}$ 与 $\boldsymbol{x}'$ 的夹角。请证明由该函数构成的核矩阵 $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j}$ 对称半正定；并写出其对应的基函数映射 $\boldsymbol{\Phi}(\boldsymbol{x})$，使得 $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\Phi}(\boldsymbol{x}_1) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_2)$。

2. 若已知非线性映射 $\mathbf{\Phi}$ 及其对应的核函数 $k(\cdot,\cdot)$，在上一节的软间隔支持向量机（取 $p=1$）中使用该核函数，请写出其**对偶问题形式**。并说明：对于任意测试样本 $\boldsymbol{x}_{\text{test}}$，分类结果可通过核函数计算为

$$f(\boldsymbol{x}_{\text{test}}) = \operatorname{sign}\left(\sum_{i=1}^{m} \alpha_i y_i \, k(\boldsymbol{x}_i, \boldsymbol{x}_{\text{test}}) + b\right),$$

其中 α_i 为对偶变量，$\boldsymbol{x}_i$ 为支持向量。

4.4 情绪检测（20pt）

本题将使用**线性软间隔支持向量机（SVM）**完成情绪分类任务，类别包括：**开心（joy）**与**伤心（sadness）**。我们已在 `start_code.py` 中提供了数据加载与预处理示例（你也可以自行实现）。请基于所学内容完成以下任务：

1. 在函数 `linear_svm_subgrad_descent` 中实现 SVM 的**随机次梯度下降**算法，并在情绪检测数据集上进行训练。（提示：可参考第 2.5 节中关于随机梯度下降的实现思路。）
2. 调整超参数（例如批大小、正则化系数 λ、步长及其衰减策略等），观察训练完成后模型在**训练集**与**验证集**上的准确率变化。通过绘制曲线或表格记录不同超参数设置下的表现，并对结果进行简要分析。（提示：不要求穷举所有超参数组合，只需如实记录你的调参过程与发现。）
3. 在函数 `kernel_svm_subgrad_descent` 中实现**基于核函数的非线性 SVM**，例如使用线性核或高斯核。通过合理调整超参数，比较模型在测试集上的准确率表现。请分析并说明核函数的引入是否提高了模型性能，以及可能的原因。（6pt）
4. 计算并汇报最终 SVM 模型在**验证集**上的分类表现，包括：
   - 准确率（Accuracy）
   - F1 值（F1-Score）
   - 混淆矩阵（Confusion Matrix）
5. 写出**逻辑斯特回归（Logistic Regression）**的目标函数与梯度的矩阵形式，并实现基于**随机梯度下降**的训练算法。报告模型的验证集准确率，并对比 SVM 与逻辑斯特回归在本任务中的表现。（提示：可以复用 SVM 中的大部分代码甚至超参数）