🎯 Gradient Descent 完整教程

期中考试必考！从直觉到公式到手算

⚠️ **期中考试重点：会手算 Gradient Descent 的一次迭代（Quiz #1 真题！）**

1️⃣ 为什么需要 Gradient Descent?

问题：Normal Equation 的局限

Normal Equation（闭式解）：

$$\theta = (X^T X)^{-1} X^T y$$

问题：

- ❌ 需要计算矩阵的逆 → 当特征很多时（>1000），计算非常慢
- ❌ $X^T X$ 可能不可逆（行列式 = 0）
- ❌ 内存占用大（需要存储整个 $X^T X$ 矩阵）

Gradient Descent 的优势：

- ✅ 不需要求逆
- ✅ 可以处理百万级特征
- ✅ 内存友好（每次只用一小批数据）
- ✅ 可以用于非线性模型（Neural Networks）

2️⃣ 直觉理解：下山找最低点

⛰️ 比喻：雾天下山

场景：

- 你站在山上（随机位置）
- 目标：找到山谷（最低点）
- 问题：大雾天，看不到全景

策略：

1. 📍 感受脚下的坡度（哪个方向最陡）
2. 👣 朝最陡的下坡方向走一小步
3. 🔁 重复步骤 1-2，直到到达山谷

对应关系：

- **山的高度** = 损失函数 $J(\theta)$
- **你的位置** = 当前参数 θ
- **坡度** = 梯度 $\nabla J(\theta)$
- **步长** = Learning Rate α
- **山谷** = 最优参数（误差最小）

3️⃣ 数学定义

核心思想

从一个初始点开始，沿着**负梯度方向**（最陡的下坡方向）迭代更新参数，直到收敛。

> ### 更新公式（核心！）
>
> $$\theta_{new} = \theta_{old} - \alpha \nabla J(\theta_{old})$$
>
> **符号说明：**
>
> - θ = 参数向量（例如 [b, a] 或 [θ_0, θ_1, θ_2]）
> - α (alpha) = Learning Rate（学习率，控制步长）
> - ∇J(θ) = 梯度（Gradient，损失函数对参数的导数）
> - 减号 (-) = 往下坡走（而不是上坡）

为什么是"负"梯度?

> **梯度 ∇J(θ) 指向上坡（误差增大的方向）**
>
> **负梯度 -∇J(θ) 指向下坡（误差减小的方向）**
>
> 我们要让误差变小，所以要走**负梯度**方向！

4️⃣ Linear Regression 的 Gradient Descent

问题设定

给定数据点 $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$，找最佳的 a 和 b：

$$y = ax + b$$

损失函数（MSE - Mean Squared Error）

$$J(a,b) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

其中：

- $\hat{y}_i = ax_i + b$ （预测值）
- y_i = 真实值
- m = 数据点个数
- 前面的 1/2 是为了求导时抵消 2（数学技巧）

梯度推导（怎么来的）

对 a 求偏导：

$$\frac{\partial J}{\partial a} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i) \cdot x_i$$

推导过程：

1. $J(a, b) = (1/2m)\ \Sigma(ax_i + b - y_i)^2$
2. 对 a 求导，用链式法则：$2(ax_i + b - y_i) \cdot x_i$

3. 2 和 1/2 抵消，得到：(1/m) Σ($\hat{y}_i$ - y_i) · x_i

对 b 求偏导：

$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)$$

推导过程：

1. 对 b 求导：2(ax_i + b - y_i) · 1
2. 得到：(1/m) Σ($\hat{y}_i$ - y_i)

更新规则（必须记住！）

$$a_{new} = a_{old} - \alpha \cdot \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i) \cdot x_i$$

$$b_{new} = b_{old} - \alpha \cdot \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)$$

5️⃣ 手算步骤（期中考试必考！）

> ⚠️ **Quiz #1 真题：给定数据、初始参数、Learning Rate，计算一次 GD 迭代！**

标准步骤（背下来！）

1. ✅ **计算预测值 $\hat{y}_i$** $= a_{old} \times x_i + b_{old}$
2. ✅ **计算误差 e_i** $= \hat{y}_i - y_i$
3. ✅ **计算梯度 $\partial J/\partial a$** $= (1/m)\ \Sigma(e_i \times x_i)$
4. ✅ **计算梯度 $\partial J/\partial b$** $= (1/m)\ \Sigma(e_i)$
5. ✅ **更新参数 a** $= a_{old} - \alpha \times \partial J/\partial a$
6. ✅ **更新参数 b** $= b_{old} - \alpha \times \partial J/\partial b$

6️⃣ 完整例子（Quiz #1 真题！）

题目

数据：

x	y
1	3
2	5

初始参数： $a = 0, b = 0$

Learning Rate： η (eta) = 0.1

任务： 计算一次 Gradient Descent 迭代

步骤 1：计算预测值

$\hat{y}_1 = a \times x_1 + b = 0 \times 1 + 0 = \mathbf{0}$

$\hat{y}_2 = a \times x_2 + b = 0 \times 2 + 0 = \mathbf{0}$

步骤 2：计算误差

$e_1 = \hat{y}_1 - y_1 = 0 - 3 = \mathbf{-3}$

$e_2 = \hat{y}_2 - y_2 = 0 - 5 =$ **-5**

步骤 3：计算梯度（对 a）

$$\frac{\partial J}{\partial a} = \frac{1}{m}\sum(e_i \times x_i)$$

$$= \frac{1}{2}[(e_1 \times x_1) + (e_2 \times x_2)]$$

$$= \frac{1}{2}[(-3 \times 1) + (-5 \times 2)]$$

$$= \frac{1}{2}[-3 - 10]$$

$$= \frac{1}{2} \times (-13) = \boxed{-6.5}$$

步骤 4：计算梯度（对 b）

$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum e_i$$

$$= \frac{1}{2}[e_1 + e_2]$$

$$= \frac{1}{2}[(-3) + (-5)]$$

$$= \frac{1}{2} \times (-8) = \boxed{-4}$$

步骤 5：更新参数 a

$$a_{new} = a_{old} - \alpha \times \frac{\partial J}{\partial a}$$

$$= 0 - 0.1 \times (-6.5)$$

$$= 0 + 0.65 = \boxed{0.65}$$

步骤 6：更新参数 b

$$b_{new} = b_{old} - \alpha \times \frac{\partial J}{\partial b}$$

$$= 0 - 0.1 \times (-4)$$

$$= 0 + 0.4 = \boxed{0.4}$$

最终答案

一次迭代后的新参数：

$$a = 0.65, \quad b = 0.4$$

新的模型：

$$y = 0.65x + 0.4$$

💡 **重复这个过程多次（100-1000次），最终会收敛到：**

a ≈ 2, b ≈ 1 （真实最优解）

7️⃣ Learning Rate (α) 的影响

α 大小	效果	问题
α 太大 (如 α = 10)	步子太大	❌ 可能跳过最优点 ❌ 震荡或发散
α 合适 (如 α = 0.1)	稳步下降	✅ 顺利收敛
α 太小 (如 α = 0.001)	步子太小	❌ 收敛非常慢 ❌ 需要更多迭代

⚠️ **α 是一个 Hyperparameter（超参数），需要人工选择或调优！**

8️⃣ 三种变体（Midterm 可能考）

Batch Gradient Descent（批量梯度下降）

每次用**全部数据**计算梯度

- ✅ 稳定
- ❌ 慢（数据量大时）

Stochastic Gradient Descent (SGD)（随机梯度下降）

每次只用**1个数据点**计算梯度

- ✅ 快
- ❌ 抖动大（不稳定）

Mini-Batch Gradient Descent（小批量梯度下降）

每次用**一小批数据**（如 32、64、128 个）

- ✅ 折中方案（最常用）
- ✅ 速度快 + 相对稳定

9️⃣ 考试常见题型

题型 1：手算一次迭代（必考！）

给定数据、初始参数、Learning Rate，要求：

1. 计算预测值 $\hat{y}$
2. 计算误差 e
3. 计算梯度 $\partial J/\partial a$, $\partial J/\partial b$
4. 更新参数 a_{new}, b_{new}

题型 2：多变量 Linear Regression

$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

需要计算 3 个梯度：$\partial J/\partial \theta_0$, $\partial J/\partial \theta_1$, $\partial J/\partial \theta_2$

题型 3：选择题 - 概念理解

- Q: 什么时候用 GD 而不是 Normal Equation?
  A: 特征很多（>1000）时
- Q: Learning Rate 太大会怎样?
  A: 可能不收敛（Overshoot）

- Q: 梯度方向是上坡还是下坡?

  A: 上坡（所以要加负号）

📝 考试速查表

必须记住的公式

更新规则：

$$\theta_{new} = \theta_{old} - \alpha \nabla J(\theta)$$

Linear Regression 梯度（简单形式）：

$$\frac{\partial J}{\partial a} = \frac{1}{m} \sum (e_i \times x_i)$$
$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum e_i$$

其中 $e_i = \hat{y}_i - y_i$

手算步骤（背诵！）

1. 算 ŷ（预测）
2. 算 e（误差）
3. 算梯度（∂J/∂a, ∂J/∂b）
4. 更新参数（减去 α × 梯度）

常见错误

- ❌ 误差算反了：应该是 ŷ - y，不是 y - ŷ
- ❌ 忘记除以 m（数据点个数）
- ❌ 更新时加号变减号（应该是减）
- ❌ Learning Rate 忘记乘

- ❌ 梯度对 a 的计算忘记乘 x_i

🎯 练习题

练习：手算一次 GD 迭代

数据：

x	y
0	1
1	3

初始参数： $a = 1, b = 0$

Learning Rate： $\alpha = 0.1$

计算：

1. $\hat{y}_1 = ?$, $\hat{y}_2 = ?$
2. $e_1 = ?$, $e_2 = ?$
3. $\partial J/\partial a = ?$
4. $\partial J/\partial b = ?$
5. $a_{new} = ?$
6. $b_{new} = ?$

在此处写下你的计算过程...

💡 下一步

做完这个练习后，你应该：

1. ✅ 理解 Gradient Descent 的直觉（下山找最低点）
2. ✅ 知道公式怎么来的（推导梯度）
3. ✅ 会手算一次迭代（6 个步骤）
4. ✅ 理解 Learning Rate 的作用

接下来复习：

- L2 Regularization（Ridge）
- Overfitting vs Underfitting
- Classification Metrics
- Fuzzy Logic