

AI/ML 综合练习题集

2026 年 1 月 5 日

目录

I 数学基础练习题	2
1 计算题	2
1.1 矩阵运算	2
1.2 概率计算	2
1.3 最大似然估计	2
1.4 信息论	2
1.5 优化问题	3
2 概念题	3
2.1 线性代数	3
2.2 概率论	3
2.3 优化理论	3
2.4 信息论	4
2.5 图论	4
3 应用题	4
4 综合思考题	5
II 深度学习练习题	5
5 概念题	5
6 编程题	7

7 综合项目	8
7.1 项目：图像分类系统	8
7.2 项目：文本生成系统	8
7.3 项目：强化学习智能体	9
8 研究性作业	9
 III 大语言模型练习题	 10
9 概念题	10
10 编程题	10
11 综合项目	11
11.1 项目：构建智能问答系统	11
11.2 项目：代码生成助手	12
 IV 大语言模型先进技术练习题	 12
12 概念理解题	12
13 计算题	12
14 代码实现题	13
15 案例分析题	13
 V Python/NumPy/Pandas 练习题	 13
16 NumPy 综合例题	13
16.1 例题 1：数据预处理	13
16.2 例题 2：矩阵运算与线性代数	14
16.3 例题 3：统计分析与数据探索	14
16.4 例题 4：广播机制的实际应用	14
17 编程实践题	15
18 综合项目	15

Part I

数学基础练习题

1 计算题

1.1 矩阵运算

1. 给定矩阵 $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 和 $\mathbf{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$, 计算 \mathbf{AB} 、 \mathbf{A}^T 和 \mathbf{A}^{-1} (如果存在)。
2. 计算向量 $\mathbf{x} = [1, 2, 3]^T$ 的 L_1 、 L_2 和 L_∞ 范数。
3. 给定两个向量 $\mathbf{u} = [1, 0, 1]^T$ 和 $\mathbf{v} = [0, 1, 1]^T$, 计算它们的点积和余弦相似度。

1.2 概率计算

1. 抛一枚公平硬币 3 次, 计算恰好出现 2 次正面的概率。
2. 假设 $X \sim \mathcal{N}(0, 1)$, 计算 $P(-1 < X < 1)$ (可以使用标准正态分布表)。
3. 给定先验概率 $P(A) = 0.3$, $P(B|A) = 0.8$, $P(B|\neg A) = 0.2$, 使用贝叶斯定理计算 $P(A|B)$ 。

1.3 最大似然估计

1. 给定样本 $x_1 = 1, x_2 = 2, x_3 = 3$, 假设它们来自泊松分布 $P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$, 求参数 λ 的最大似然估计。
2. 给定样本 x_1, \dots, x_n 来自指数分布 $p(x) = \lambda e^{-\lambda x}$ ($x \geq 0$), 求参数 λ 的最大似然估计。

1.4 信息论

1. 计算伯努利分布 $P(X = 1) = p$, $P(X = 0) = 1 - p$ 的熵 $H(X)$, 并找出使熵最大的 p 值。
2. 给定联合分布:

		$X \setminus Y$	
		0	1
0	0	0.3	0.2
	1	0.1	0.4

计算 $H(X)$ 、 $H(Y)$ 、 $H(X|Y)$ 和 $I(X; Y)$ 。

1.5 优化问题

1. 使用梯度下降法求解 $f(x) = x^2 + 2x + 1$ 的最小值, 初始值 $x_0 = 0$, 学习率 $\eta = 0.1$, 迭代 5 次。
2. 使用拉格朗日乘数法求解约束优化问题:

$$\begin{aligned} \min \quad & x^2 + y^2 \\ \text{s.t.} \quad & x + y = 1 \end{aligned} \tag{1}$$

2 概念题

2.1 线性代数

1. 解释特征值和特征向量的几何意义。
2. 为什么在机器学习中经常使用矩阵的转置?
3. 解释为什么矩阵乘法不满足交换律, 但在神经网络中这通常不是问题。

2.2 概率论

1. 解释先验概率、似然函数和后验概率的区别和联系。
2. 为什么在机器学习中经常假设误差服从正态分布?
3. 最大似然估计和最大后验估计 (MAP) 的区别是什么?

2.3 优化理论

1. 解释为什么梯度下降可能陷入局部最优, 而凸优化可以保证全局最优。
2. 学习率的选择对梯度下降有什么影响?
3. 解释拉格朗日乘数的物理意义或几何意义。

2.4 信息论

1. 解释熵、互信息和 KL 散度的直观含义。
2. 为什么 KL 散度不是真正的距离度量？
3. 在决策树中，为什么使用信息增益而不是直接使用熵？

2.5 图论

1. 解释邻接矩阵和邻接表的优缺点。
2. 为什么图神经网络需要特殊的消息传递机制，而不能直接使用传统的卷积？
3. 解释图注意力网络中注意力机制的作用。

3 应用题

1. PCA 实现：

- 给定一个数据矩阵，实现主成分分析（PCA）算法
- 使用特征值分解计算主成分
- 将数据投影到前两个主成分，可视化结果

2. 朴素贝叶斯分类器：

- 实现一个简单的朴素贝叶斯分类器
- 在文本分类任务上测试（如垃圾邮件检测）
- 分析先验概率对分类结果的影响

3. 梯度下降可视化：

- 实现梯度下降算法
- 在二维函数上可视化梯度下降的路径
- 比较不同学习率对收敛速度的影响

4. 信息增益计算：

- 实现信息熵和信息增益的计算
- 在简单的数据集上计算各特征的信息增益

- 解释为什么信息增益大的特征更适合用于分裂

5. 图的基本操作：

- 实现图的邻接矩阵和邻接表表示
- 实现图的遍历算法（深度优先搜索、广度优先搜索）
- 实现简单的图神经网络消息传递

4 综合思考题

1. 数学工具的综合应用：

- 选择一个机器学习算法（如逻辑回归、支持向量机、神经网络），分析其中用到了哪些数学工具。
- 解释这些数学工具在算法中各自的作用。
- 讨论如果缺少某个数学工具，算法会如何变化。

2. 数学与直觉：

- 选择一个数学概念（如熵、梯度、特征值），用非数学的语言解释其直观含义。
- 举一个生活中的例子说明这个概念。
- 解释这个概念在机器学习中的重要性。

3. 数学证明：

- 证明信息熵的非负性。
- 证明 KL 散度的非负性（使用 Jensen 不等式）。
- 证明梯度指向函数值增加最快的方向。

Part II

深度学习练习题

5 概念题

1. 梯度消失问题：

- 解释什么是梯度消失问题，为什么会出现？
- 列举至少三种缓解梯度消失问题的方法，并说明其原理。
- 为什么 ReLU 激活函数能够缓解梯度消失问题？

2. 卷积神经网络：

- 解释卷积操作中的局部连接和权重共享如何减少参数量。
- 比较最大池化和平均池化的优缺点。
- 为什么 CNN 适合处理图像数据？

3. 注意力机制：

- 解释自注意力和交叉注意力的区别。
- 为什么 Transformer 需要位置编码？
- 多头注意力的优势是什么？

4. 批量归一化：

- 解释批量归一化为什么能够加速训练。
- 批量归一化在训练和测试阶段的区别是什么？
- 为什么批量归一化具有正则化效果？

5. 强化学习：

- 解释强化学习与监督学习的区别。
- 什么是探索-利用权衡 (Exploration-Exploitation Trade-off)？
- DQN 中的经验回放和目标网络的作用是什么？

6. 损失函数：

- 比较均方误差 (MSE) 和平均绝对误差 (MAE) 的优缺点，各适用于什么场景？
- 为什么交叉熵损失函数与 softmax 激活函数配合使用效果最佳？
- 解释 Focal Loss 如何解决类别不平衡问题，其核心思想是什么？
- 在什么情况下应该使用 Huber Loss 而不是 MSE 或 MAE？
- 三元组损失函数在度量学习中的作用是什么？如何选择正负样本？

6 编程题

1. 实现多层感知机：

- 使用 PyTorch 或 TensorFlow 实现一个三层 MLP（输入层、隐藏层、输出层）
- 在 MNIST 数据集上训练模型
- 实现前向传播和反向传播（可以调用框架的自动微分功能）
- 尝试不同的激活函数（Sigmoid、tanh、ReLU）并比较效果

2. 实现简单的 CNN：

- 实现一个包含卷积层、池化层和全连接层的 CNN
- 在 CIFAR-10 数据集上训练模型
- 可视化卷积层的特征图，观察不同层学习到的特征

3. 实现 LSTM：

- 使用 PyTorch 或 TensorFlow 实现 LSTM 单元
- 在文本分类任务（如情感分析）上训练模型
- 比较 LSTM 和简单 RNN 的性能差异

4. 实现注意力机制：

- 实现自注意力机制
- 实现多头注意力机制
- 在序列到序列任务（如机器翻译）中应用注意力机制

5. 实现批量归一化：

- 实现批量归一化层（包括训练和测试模式）
- 在深度网络中应用批量归一化，观察训练速度和最终性能的变化
- 比较使用和不使用批量归一化的训练曲线

6. 实现 Dropout：

- 实现 Dropout 层（包括训练和测试模式）
- 在过拟合的模型上应用 Dropout，观察正则化效果
- 可视化 Dropout 对模型权重分布的影响

7. 实现不同的损失函数：

- 实现 MSE、MAE 和 Huber Loss，比较它们在回归任务上的表现
- 实现交叉熵损失函数，在分类任务上验证其效果
- 实现 Focal Loss，在类别不平衡的数据集上比较其与标准交叉熵的差异
- 实现三元组损失函数，在图像检索任务中应用
- 尝试组合多个损失函数（如内容损失 + 风格损失），观察效果

7 综合项目

7.1 项目：图像分类系统

- 使用深度学习框架（PyTorch 或 TensorFlow）构建一个完整的图像分类系统
- 实现数据加载、预处理、模型定义、训练和评估的完整流程
- 尝试不同的网络架构（MLP、CNN、ResNet）
- 应用不同的优化技术（批量归一化、Dropout、学习率调度）
- 使用数据增强技术提高模型泛化能力
- 可视化训练过程（损失曲线、准确率曲线）
- 分析模型的错误案例，提出改进方案

7.2 项目：文本生成系统

- 使用 RNN/LSTM/GRU 或 Transformer 构建文本生成模型
- 在文本数据集（如小说、新闻）上训练模型
- 实现不同的采样策略（贪婪搜索、随机采样、束搜索）
- 评估生成文本的质量（困惑度、BLEU 分数等）
- 尝试不同的超参数设置，观察对生成质量的影响

7.3 项目：强化学习智能体

- 使用深度强化学习 (DQN 或策略梯度方法) 训练游戏智能体
- 在 OpenAI Gym 环境 (如 CartPole、Atari 游戏) 中训练
- 实现经验回放和目标网络 (对于 DQN)
- 可视化训练过程和智能体的行为
- 分析不同超参数 (学习率、折扣因子等) 对性能的影响

8 研究性作业

1. **文献阅读:** 阅读以下经典论文, 总结其主要贡献:

- LeCun et al. (1998). "Gradient-based learning applied to document recognition"
- Krizhevsky et al. (2012). "ImageNet classification with deep convolutional neural networks"
- He et al. (2016). "Deep residual learning for image recognition"
- Vaswani et al. (2017). "Attention is all you need"
- Devlin et al. (2018). "BERT: Pre-training of deep bidirectional transformers for language understanding"

2. **技术调研:** 选择一个深度学习应用领域 (如医疗影像、自动驾驶、金融风控), 调研该领域的最新进展, 包括:

- 主要技术方法
- 面临的挑战
- 最新研究成果
- 实际应用案例

3. **实验设计:** 设计一个实验来验证某个深度学习技术 (如批量归一化、残差连接) 的有效性:

- 明确研究问题
- 设计对比实验

- 确定评估指标
- 分析实验结果

Part III

大语言模型练习题

9 概念题

1. Transformer 架构:

- 解释自注意力和交叉注意力的区别。
- 为什么 Transformer 需要位置编码?
- 多头注意力的优势是什么?

2. 预训练与微调:

- 比较自回归语言建模和掩码语言建模的优缺点。
- 解释 Few-shot Learning 和 In-context Learning 的区别。
- LoRA 如何实现参数高效微调?

3. 提示工程:

- Chain-of-Thought 如何提高模型的推理能力?
- Few-shot Prompting 的工作原理是什么?
- 如何设计有效的提示?

4. RAG:

- RAG 相比直接生成有什么优势?
- 如何构建高效的检索系统?
- RAG 如何解决大模型的幻觉问题?

10 编程题

1. 实现 Transformer 编码器:

- 使用 PyTorch 实现完整的 Transformer 编码器
- 包含位置编码、多头注意力、前馈网络
- 在文本分类任务上测试

2. 实现注意力机制：

- 实现缩放点积注意力
- 实现多头注意力
- 可视化注意力权重

3. 实现 RAG 系统：

- 构建文档向量索引
- 实现检索功能
- 基于检索结果生成答案

4. 提示工程实践：

- 设计 Few-shot Prompting 提示
- 实现 Chain-of-Thought 提示
- 比较不同提示策略的效果

5. 模型微调：

- 使用 Hugging Face Transformers 微调预训练模型
- 实现 LoRA 微调
- 比较全参数微调和 LoRA 的效果

11 综合项目

11.1 项目：构建智能问答系统

- 使用预训练的大语言模型
- 实现 RAG 增强生成
- 构建知识库和检索系统
- 设计用户界面
- 评估系统性能

11.2 项目：代码生成助手

- 在代码数据集上微调模型
- 实现代码生成功能
- 添加代码补全功能
- 实现代码审查功能

Part IV

大语言模型先进技术练习题

12 概念理解题

1. 解释 LoRA 的数学原理，说明为什么低秩分解能够有效。
2. 比较 QLoRA 和 LoRA 的区别，说明量化的作用。
3. 解释 Flash Attention 如何减少内存占用。
4. 说明 BLEU 分数的简短惩罚的作用。
5. 解释 ROUGE-L 和 ROUGE-N 的区别。

13 计算题

1. 困惑度计算：

- 给定序列长度为 100，平均对数概率为 -2.5，计算困惑度
- 手算并编写代码验证

2. BLEU 计算：

- 参考: "the cat sat on the mat"
- 候选: "a cat sat on mat"
- 计算 1-gram 到 4-gram 的精确度、简短惩罚和 BLEU 分数
- 手算并编写代码验证

3. ROUGE 计算:

- 参考: "the cat is on the mat"
- 候选: "cat mat"
- 计算 ROUGE-1、ROUGE-2 和 ROUGE-L
- 手算并编写代码验证

14 代码实现题

1. 实现一个完整的 LoRA 训练脚本，包括数据加载、模型配置、训练循环和评估。
2. 实现 BLEU 分数计算函数，支持多个参考文本。
3. 实现 ROUGE 分数计算函数，包括 ROUGE-N 和 ROUGE-L。
4. 实现困惑度计算函数，支持批量计算。

15 案例分析题

1. 给定一个文本生成任务，设计完整的评估方案，包括选择合适的评估指标、实现评估代码、分析结果。
2. 分析 LoRA 在不同任务上的效果，比较不同 rank 设置的影响。
3. 设计一个模型部署方案，包括模型压缩、服务化部署和性能优化。

Part V

Python/NumPy/Pandas 练习题

16 NumPy 综合例题

16.1 例题 1：数据预处理

给定一个包含 1000 个样本、20 个特征的数据集，完成以下任务：

1. 计算每个特征的均值和标准差
2. 进行 Z-score 标准化（均值为 0，标准差为 1）

3. 检测并处理异常值（使用 3 原则）
4. 将数据重塑为适合输入神经网络的形状

16.2 例题 2：矩阵运算与线性代数

给定矩阵 $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ 和向量 $\mathbf{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ ，完成以下任务：

1. 计算矩阵乘法 \mathbf{Ab}
2. 计算 \mathbf{A} 的转置 \mathbf{A}^T
3. 计算 $\mathbf{A}^T \mathbf{A}$ 的特征值和特征向量
4. 对 \mathbf{A} 进行 SVD 分解

16.3 例题 3：统计分析与数据探索

给定一个包含学生成绩的数据集，完成以下分析：

1. 计算各科目的平均分、标准差、最高分、最低分
2. 找出总分最高的学生
3. 计算各科目的及格率（60 分以上）
4. 找出所有科目都及格的学生

16.4 例题 4：广播机制的实际应用

实现批量归一化（Batch Normalization）操作，这是深度学习中常用的技术。

要求：

1. 给定一个批次的数据 $\mathbf{X} \in \mathbb{R}^{B \times D}$ (B 个样本, D 个特征)
2. 对每个特征进行归一化： $\hat{x}_i = \frac{x_i - \mu}{\sigma}$
3. 应用缩放和平移： $y_i = \gamma \hat{x}_i + \beta$

17 编程实践题

1. 数组操作:

- 创建一个 5×5 的随机数组
- 提取对角线元素
- 计算每行的均值和标准差
- 找出最大值和最小值的位置

2. 广播机制应用:

- 给定一个 100×10 的特征矩阵
- 使用广播机制对每个特征进行标准化
- 验证标准化后的均值和标准差

3. 线性代数运算:

- 实现矩阵乘法的多种方法
- 计算矩阵的逆和行列式
- 实现特征值分解和 SVD 分解

4. 统计分析:

- 生成模拟数据
- 计算各种统计量 (均值、方差、中位数、百分位数)
- 检测和处理异常值

18 综合项目

项目：数据预处理流水线

- 实现一个完整的数据预处理流水线
- 包括数据加载、清洗、特征工程、标准化等步骤
- 使用 NumPy 进行高效的数值计算
- 可视化数据分布和预处理效果
- 评估预处理对后续机器学习任务的影响

通过完成以上练习题，读者可以全面掌握 AI/ML 领域的核心知识，从数学基础到深度学习，从大语言模型到实际应用，建立完整的知识体系。