

2026 专硕入组考察

Jiaye Group

说明

1. 本测试旨在了解大家的学术背景与工程能力，以便制定个性化的培养计划。
 2. 请在 **Track A (理论)** 和 **Track B (应用)** 中任选一个模块完成。
 3. **鼓励挑战：**如果你对两方面都有信心，可以尝试混合选做。
 4. **工具使用：**允许使用大语言模型辅助解题，但**必须自行检查答案的正确性与精准性**。大模型常在数学推导细节上出现幻觉，若出现此类逻辑谬误将严重扣分。可以与他人讨论题目，但禁止代做题目。
 5. **提交格式：**理论题请提供清晰的推导过程（PDF或手写扫描）；应用题请提供代码仓库链接及实验报告。
-

Track A：理论

一、必做题

1. 统计收敛与一致收敛

- (1) **依概率收敛 vs. 几乎处处收敛：**构造一个随机变量序列 $\{X_n\}$ ，使得 $X_n \xrightarrow{P} 0$ （依概率收敛到 0），但在 $n \rightarrow \infty$ 时 X_n 并不几乎处处收敛到 0。
- (2) **一致收敛 (Uniform Convergence)：**在机器学习中，我们为什么不仅关心单一假设 h 的经验风险收敛到真实风险（大数定律保证），而更关心一致收敛 ($\sup_{h \in \mathcal{H}} |R(h) - \hat{R}_S(h)| \rightarrow 0$)？如果一致收敛不成立，经验风险最小化 (ERM) 原则是否依然可靠？

2. 假设检验：两类错误 (Type I & Type II Errors)

- (1) **定义：**请严格定义假设检验中的**第一类错误**和**第二类错误**。
- (2) **权衡：**能否同时降低这两类错误的概率？请简述它们之间的制约关系。
- (3) **场景应用：**在癌症筛查模型中 (H_0 : 患者健康, H_1 : 患者患癌)，将癌症患者误诊为健康人属于哪一类错误？如果你是算法设计者，你会更倾向于控制哪一类错误？
- (4) **计算实例：**设观测数据 $X \sim N(\mu, 1)$ 。考虑假设检验问题： $H_0 : \mu = 0$ vs. $H_1 : \mu = 2$ 。设拒绝域为 $C = \{x \mid x > 1\}$ （即当观测值 $X > 1$ 时拒绝零假设）。请计算该检验的第一类错误概率 α 和第二类错误概率 β 。结果可用标准正态分布函数 $\Phi(\cdot)$ 表示。

二、选做题 (4选2)

模块 1：凸优化：平滑性与强凸性

考虑无约束优化问题 $\min_{x \in \mathbb{R}^n} f(x)$, 假设 $f(x)$ 是二阶可微的凸函数。

- (1) 概念辨析：请用数学语言严格定义函数的 **L -平滑 (L -smoothness)** 和 **μ -强凸 (μ -strong convexity)**。并说明 Hessian 矩阵 $\nabla^2 f(x)$ 在这两种性质下需满足的特征值条件。
- (2) 收敛性分析：对于梯度下降法 (Gradient Descent) $x_{t+1} = x_t - \eta \nabla f(x_t)$, 请证明：当 $f(x)$ 满足 L -平滑且步长 $\eta = 1/L$ 时，其收敛速率为 $O(1/T)$ 。若函数同时满足 μ -强凸，收敛速率会有何变化？

模块 2：统计学习理论：Rademacher Complexity

设假设空间 \mathcal{H} 是输入空间 \mathcal{X} 到 $\{-1, +1\}$ 的函数集合。给定样本集 $S = \{z_1, \dots, z_m\}$ 。

- (1) 经验 Rademacher 复杂度：写出经验 Rademacher 复杂度 $\hat{\mathfrak{R}}_S(\mathcal{H})$ 的定义式，并解释引入 Rademacher 变量 σ_i 的直观意义。它是在模拟什么？
- (2) 泛化误差界：已知对于二分类问题，以概率 $1 - \delta$, 对任意 $h \in \mathcal{H}$ 成立：

$$R(h) \leq \hat{R}_S(h) + 2\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

请简述上述不等式中三项分别代表什么？并回答：如果我们将假设空间 \mathcal{H} 扩大（例如增加神经网络的层数），这三项中哪一项会显著变大？这体现了什么 Trade-off？

- (3) 关联性：McDiarmid 不等式在证明上述泛化界时起到了关键作用，请简述其核心思想。

模块 3：近端梯度下降 (Proximal Gradient Descent)

针对 Lasso 问题 $\min_w \frac{1}{2} \|y - Xw\|_2^2 + \lambda \|w\|_1$:

该目标函数不可导。请推导针对 ℓ_1 范数的近端算子 (Proximal Operator) $\text{prox}_{\lambda\|\cdot\|_1}(v)$ 的闭式解。

模块 4：RKHS (再生核希尔伯特空间)

请叙述 表示定理 (Representer Theorem) 的内容。它如何将一个在无限维函数空间中的优化问题转化为有限维参数的优化问题？

Track B：应用

一、必做题 (Compulsory)

1. 手写核心组件

请不使用 PyTorch/TensorFlow 中的高级 API (如 `nn.MultiheadAttention`, `nn.BatchNorm2d`)，仅使用张量操作 (Tensor operations, 如 `matmul`, `transpose`, `mean`, `var`) 实现以下模块之一 (二选一)：

- **选项 A: Multi-Head Self-Attention**

- 实现一个类 `MyMultiHeadAttention`。
- **输入:** Batch Size, Sequence Length, Embedding Dimension。
- **要求:** 包含 Q, K, V 的线性投影, Scaled Dot-Product Attention 计算, Mask 处理 (可选) , 以及最终的输出投影。
- **思考:** 为什么除以 $\sqrt{d_k}$ 是必须的? 如果不除, 当维度 d_k 很大时, Softmax 的梯度会发生什么变化?

- **选项 B: Batch Normalization**

- 实现一个类 `MyBatchNorm2d`。
- **要求:** 必须正确处理 `training` 和 `eval` 两种模式的区别。
- 在训练时使用当前 Batch 的均值方差, 同时更新全局的 `running_mean` 和 `running_var`。
- 在推理时直接使用全局统计量。

2. 残差连接与梯度流

- 搭建一个深层网络, 并自行寻找数据集。
- **实验 A:** 不使用 Residual Connection 进行训练。
- **实验 B:** 使用 Residual Connection 进行训练。
- **可视化:** 请记录并画出训练初期 (前 5 个 Epoch) 第一层和最后一层梯度的分布直方图。
- **分析:** 通过对梯度分布, 解释 ResNet 是如何缓解梯度消失/退化问题的。

二、选做题 (4选2)

模块 1：位置编码

Transformer 原始论文使用了正弦/余弦位置编码。目前的 LLM 常用 RoPE。

请简述 RoPE 的核心思想, 并写出一段伪代码或 PyTorch 代码演示它是如何通过旋转操作将相对位置信息注入到 Attention 矩阵中的。

模块 2：混合精度训练

在训练大模型时, 我们常用 FP16 或 BF16。

- 请解释什么是 Loss Scaling? 为什么在 FP16 训练中需要它?
- 使用 PyTorch 的 `torch.cuda.amp` 改写一段标准的训练循环代码, 使其支持混合精度训练。

模块 3：VAE 的重参数化技巧

VAE 中，我们需要从分布 $N(\mu, \sigma^2)$ 中采样隐变量 z 。

直接采样 $z \sim N(\mu, \sigma^2)$ 会导致梯度无法反向传播。请写出重参数化技巧的数学公式，并用代码片段展示如何在 PyTorch 中实现这一采样过程以保持计算图的连通性。

模块 4：LLM Agent 与工具调用

假设你有一个大语言模型接口 `llm_generate(prompt)` 和两个外部工具函数：`get_weather(city)` 和 `calculator(expression)`。

- 请设计一段 System Prompt，使模型在面对复杂指令（如“上海比北京热吗？”）时，能够遵循 ReAct (Reasoning + Acting) 模式，输出思考过程和工具调用指令。
- 请写一段 Python 代码（或伪代码），实现 Agent 的核心循环逻辑：用户输入 → LLM 解析 → 判断是否调用工具 → 执行工具 → 结果回填 LLM → 生成最终回复。