

MNIST 手写数字分类对比实验指南

(MLP/CNN/MinGRU/Mamba/Transformer)

作者

2025 年 12 月 28 日

摘要

本文在统一的数据处理、训练与评估流程下，对比五种骨干网络（MLP、CNN、MinGRU、Mamba、Transformer）在 MNIST 及相关数据集上的分类性能，提供快速开始指南、调参建议、结果解读与扩展实验方向，并汇总输出文件与命令。

目录

1 项目简介	2
2 环境准备	2
3 快速开始	2
4 运行说明	2
5 配置与调参	2
5.1 主要参数	2
5.2 修改方式	3
6 输出与可视化	3
7 实验结果摘要	3
8 结果分析与解读	3
9 三数据集对比分析（统一坐标）	3
10 常见问题与解决	4
11 扩展实验建议	4
12 复现实验与对比	4
13 文件与命令汇总	4

1 项目简介

- 对比五种骨干网络在 MNIST 分类任务上的性能: `mlp`、`cnn`、`mingru`、`mamba`、`transformer`
- 统一数据处理、训练与评估流程; 输出训练/测试损失与准确率曲线以及指标汇总

2 环境准备

- 依赖: `torch`、`torchvision`、`numpy`、`matplotlib`
- 安装命令: `pip install -r classify/requirements.txt`
- 设备: 默认 CPU; 如需 GPU, 可在本地配置 CUDA 并修改代码中的设备

3 快速开始

- 运行基础对比 (MNIST): `python -m classify.main`
- 运行多数据集消融: `python -m classify.experiments`
- 结果位置: `classify/results/`
 - 单模型曲线: `{dataset}_{model}_{variant}_curves.png` (如 `mnist_cnn_base_curves.png`)
 - 横向对比 (同数据集不同模型): 测试准确率 `{dataset}_cross_model_test_acc.png`; 测试损失 `{dataset}_cross_model_test_loss.png`
 - 纵向消融(同模型不同结构/参数): 测试准确率 `{dataset}_{model}_ablation_test_acc.png`; 测试损失 `{dataset}_{model}_ablation_test_loss.png`
 - 指标汇总: `ablation_summary.json`

4 运行说明

- 训练与评估流程在 `classify/train.py` 中实现, `classify/main.py` 会顺序运行五种模型并保存结果
- 单模型运行示例(在交互式环境中调用):`from classify.train import train_one,train_one('cnn', epochs=3)`

5 配置与调参

5.1 主要参数

- `epochs`: 训练轮次 (默认 3, 建议 5-10 以获得更稳定结果)
- `batch_size`: 批次大小 (默认 64)
- `lr`: 学习率 (默认 `1e-3`)
- `device`: 设备 (`cpu` 或 `cuda`)

5.2 修改方式

- 批量运行时可编辑 `classify/train.py` 中 `run_all` 的调用参数
- 单模型运行时直接传参给 `train_one`

6 输出与可视化

每个模型训练结束会生成两类曲线（统一坐标： $\text{loss} \in [0, 1]$, $\text{acc} \in [0, 1]$ ）：

- 训练/测试损失随 epoch 变化：左图
- 训练/测试准确率随 epoch 变化：右图

对比图：横向（跨模型）与纵向（同模型不同参数/结构）均以“所有对比项画在一张图”输出，分别提供测试集损失与准确率两张图。

7 实验结果摘要

设置：`3 epoch, batch=64, Adam lr=1e-3, CPU`。

- `cnn`: Test Acc 约 0.990; Test Loss 约 0.032; 收敛速度快，稳定性好，训练/测试曲线贴合
- `mlp`: Test Acc 约 0.973; Test Loss 约 0.089; 表现稳健，收敛良好，但相对 CNN 略逊
- `mingru`: Test Acc 约 0.982; Test Loss 约 0.065; 序列建模对 MNIST 有增益，优于 MLP
- `transformer`: Test Acc 约 0.964; Test Loss 约 0.115; 简化 ViT，在更长训练与更大维度下可进一步提升
- `mamba`: Test Acc 约 0.928; Test Loss 约 0.253; 简化实现下短训较敏感，适当加深/增大状态维度可改善

8 结果分析与解读

- 收敛速度：`cnn` 与 `mingru` 收敛快且平滑；`mlp` 次之；`transformer` 与 `mamba` 需更长训练与更大容量
- 泛化能力：训练/测试曲线间隙小的模型更稳健，`cnn`、`mingru` 在当前设置下泛化较好
- 架构差异：`cnn` 在图像任务上具备局部建模与平移不变性优势；`mingru` 将图像按列序列化，门控机制帮助捕捉跨步依赖；`transformer` 依赖足够的 token 容量与层数，简化配置下性能一般；`mamba` 的状态空间机制强调长序列效率，图像上需更适配设计与更深层

9 三数据集对比分析（统一坐标）

- MNIST: `cnn` 收敛最快、准确率最高；`mingru` 次之；`mlp` 稳健；`transformer/mamba` 需更大容量及更长训练

- CIFAR-10: 彩色、复杂背景更凸显 `cnn` 的局部归纳偏置优势; 小型 `transformer` 随容量提升 (`embed/head/depth`) 有明显增益; `mlp` 由于缺少空间结构表现较弱
- Sequential MNIST (Permuted): 长序列依赖更能体现 MinGRU/Mamba/Transformer 差异; `cnn` 因失去二维先验表现较弱; `mlp` 亦逊色

10 常见问题与解决

- MNIST 下载失败: 检查网络或更换下载镜像; 确保 `torchvision` 版本满足要求
- 训练过慢: 减少 `epochs/batch_size`, 或在具备环境的情况下改为 `device='cuda'`
- 曲线不收敛: 降低学习率至 $5e-4/1e-4$, 增加 `epochs` 至 5–10
- 精度不达预期: `cnn` 增加通道宽度与层数, 在分类器处增加隐藏层; `transformer` 增大 `embed_dim`、`num_heads`、`depth`, 添加 `dropout`; `mamba` 增加 `state_size`, 堆叠多层 cell, 延长训练

11 扩展实验建议

- 增加训练轮次: 将 `epochs` 调整为 10–20, 观察曲线趋稳与最终指标提升
- 调优策略: 余弦退火或阶梯式学习率衰减; 在 `transformer` 中尝试 AdamW
- 架构扩展: `cnn` 双卷积块 (1->32->64)、批归一化、`Dropout(0.5)`; `mingru` 加大 `hidden_size` 至 256, 或双向扫描再融合; `mamba` 多层堆叠、提高 `state_size>=128`, 尝试可学习位置编码融合; `transformer patch_size=2` 增加 token 数量, `embed_dim=128~256`, `depth=4~6`

12 复现实验与对比

- 建议固定随机种子与数据加载参数, 以便不同架构可直接横向对比
- 在相同 `epochs/batch_size/lr` 条件下进行测试, 并统一评估指标与绘制曲线

13 文件与命令汇总

- 曲线文件: `classify/results/{dataset}_{model}_{variant}_curves.png`
- 对比图:`{dataset}_cross_model_test_acc.png`/`{dataset}_cross_model_test_loss.png`、`{dataset}_{mode}`
- 指标汇总: `classify/results/ablation_summary.json`
- 安装依赖: `pip install -r classify/requirements.txt`
- 运行全部: `python -m classify.main`