多模态Transformer推荐系统项目报告

本文档系统性介绍本项目的任务定义、数据处理流程、模型设计（含Transformer细节）、训练与损失函数、以及评估指标，并配以公式解释，帮助初学者理解整体方法论。

一、项目任务与数据

- 任务类型：基于用户交互序列的下一物品预测（Next-Item Prediction）。
- 数据来源：电商评论数据（包含用户对商品的交互、文本描述和商品图片）。
- 多模态特征：
  - 文本特征：使用 BERT 模型提取 `[CLS]` 位置的句子嵌入（维度 768）。
  - 图像特征：使用 ViT 模型提取 `[CLS]` 位置的图像嵌入（维度 768）。
- 序列形式：对每个用户，按时间排序其交互得到序列 `[item_1, item_2, ..., item_N]`。训练时在位置 t 用前 t-1 个物品的多模态特征作为输入，预测位置 t 的目标物品 `item_t`。

数据预处理要点

- 交互整理：将原始交互按 `user_id` 聚合，并按时间排序形成序列。
- 多模态嵌入：
  - 文本：`BERT(last_hidden_state)[:, 0, :]` 提取 `[CLS]` 表示。
  - 图像：`ViT(last_hidden_state)[:, 0, :]` 提取 `[CLS]` 表示。
- 物品ID映射：建立 `itemId → 索引` 的字典，用于训练分类（Embedding 查表）。
- 训练/测试划分：按用户序列切片得到训练样本与测试样本（例如 80% 训练，20% 测试）。
- 序列填充：批内各样本长度不一时，对输入序列和目标序列做右侧零向量（特征）与 `ignore_index`（标签）填充，同时保存每条样本的真实长度 `seq_len`。

二、模型设计（MmTransformer4Rec）

整体上，模型属于“两塔式”的变体：

- 塔一：用户序列编码塔（Transformer 编码器，融合文本与图像）。
- 塔二：物品嵌入表（`torch.nn.Embedding(n_items, hidden_dim)`）。

2.1 多模态特征融合（MmItemEncoder）

输入每个时间步的文本与图像嵌入，拼接后用 MLP 投影到隐藏维度：

- 输入 `x_t = concat(txt_t, img_t) ∈ R^{768+768}`
- 投影 `h_t = MLP(x_t) ∈ R^{hidden_dim}`

2.2 位置编码（PositionalEmbedding）

采用可学习或固定位置编码，将序列位置信息注入到 `h_t`：

- 令 `PE ∈ R^{max_len × hidden_dim}`，则对序列 `H ∈ R^{L × hidden_dim}`： $$ H' = H + PE[:L] $$

2.3 Transformer 编码器（多层）

每层包含 Multi-Head Attention（MHA）与前馈网络（FFN），并采用残差与归一化：

- 残差结构： $$ X_{attn} = \mathrm{RMSNorm}(X + \mathrm{MHA}(X, X, X; \text{mask})) $$ $$ X_{ffn} = \mathrm{RMSNorm}(X_{attn} + \mathrm{FFN}(X_{attn})) $$
- 注意力打分与 Softmax： $$ \text{scores} = \frac{QK^\top}{\sqrt{d_k}} $$ 对 **Key 方向**的 padding 位置使用掩码，将这些位置的 `scores` 填充为 $-\infty$，以确保 `softmax` 不分配概率到无效位置： $$ \text{scores}_{i,j} = \begin{cases} -\infty, & \text{若 } j \text{ 为 padding}\ \frac{Q_i \cdot K_j}{\sqrt{d_k}}, & \text{否则} \end{cases} $$ $$ \text{attn} = \mathrm{softmax}(\text{scores}) $$
- 前馈网络（ReLU/GELU）：两层线性层带 Dropout，提升表达能力。
- 归一化（RMSNorm）：对每个位置向量做均方根归一化，数值稳定： $$ \mathrm{RMSNorm}(x) = \frac{x}{\sqrt{\mathbb{E}[x^2] + \varepsilon}} \odot \gamma $$

2.4 得分计算（与物品嵌入表匹配）

Transformer 输出序列隐表示 `X ∈ R^{B × L × hidden_dim}`，与物品嵌入表 `E ∈ R^{n\_items × hidden_dim}` 做矩阵乘：

- 逐位置得分： $$ \text{logits}*{b,l,:} = *X*{b,l,:} \cdot E^\top $$
- 形状为 `(batch_size, seq_len, n_items)`，用于对每个位置进行分类。

三、训练与损失函数

- 训练目标：位置对齐的交叉熵损失，输入 `[item_1, ..., item_{L-1}]`，目标为 `[item_2, ..., item_L]`。模型在每个位置`l` 预测 `item_{l+1}`。
- 交叉熵损失（含填充忽略）：
  - 对每条样本的有效长度 `seq_len`，只在有效位置计算损失；填充位置的目标标记为 `\text{ignore\_index} = -100`。
  - 平均损失：将逐位置损失按有效掩码求平均。
- 公式： 设第 `b` 条样本长度为 `L_b`，其有效位置集合为 `\mathcal{I}_b = \{0, 1, ..., L_b-1\}`，则批次损失为： $$ \mathcal{L} = \frac{1}{\sum_b L_b} \sum_{b} \sum_{l \in \mathcal{I}*b} *mathrm{CE}*(*text{logits}*{b,l,:},\ \text{target}_{b,l}) $$ 其中 `CE` 为交叉熵，填充位置不计入分母与分子。

四、评估方式与指标

评估在测试集上进行，输入整段序列，

- 用最后一个位置的隐状态 `X_{b, L_b-1, :}` 计算对所有物品的得分，得到该样本的 Top-K 推荐列表。
- 与真实下一物品进行比对，计算 HR@K 与 NDCG@K。

4.1 Hit Ratio（HR@K）

对于 Next-Item 任务（每样本只有 1 个目标），HR@K 与 Recall@K 等价：

- 定义：Top-K 列表命中目标的样本比例。

- 公式： $$ \mathrm{HR@K} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}{\text{target}_i \in \text{TopK}_i} $$

4.2 NDCG@K（归一化折扣累积增益）

- 对二元相关性（命中=1，否则=0），若目标在 Top-K 中位置为 `\text{rank}`（从 1 开始），则： $$ \mathrm{DCG@K}_i = \begin{cases} \frac{1}{\log_2(\text{rank}+1)}, & \text{若命中}\ 0, & \text{未命中} \end{cases} $$
- 理想 DCG（IDCG）为 1（目标排第一）。因此： $$ \mathrm{NDCG@K} = \frac{1}{N} \sum_{i=1}^N \mathrm{DCG@K}_i $$

4.3 Loss（测试集平均）

- 在测试集上同样按有效位置计算交叉熵损失，报告各批次的平均损失： $$ \mathrm{Test\ Loss} = \frac{1}{B} \sum_{b=1}^B \mathcal{L}_b $$

五、训练日志示例与效果解读

来自 `logs/train_YYYYMMDD_HHMMSS.log` 的片段（示例）：

- 初始损失（Epoch 0）逐步下降，Top-K 指标在训练过程中有一定提升；
- HR@10 通常高于 HR@5；NDCG@10 通常高于 NDCG@5，符合直觉（更大的 K 更易命中）。

示例（项目运行得到的某次结果）：

- Epoch 0 Test：Loss ≈ 2.15，HR@5 ≈ 0.29，HR@10 ≈ 0.46，NDCG@5 ≈ 0.18，NDCG@10 ≈ 0.23
- 随着训练进行，Loss 降低、NDCG 有所提升，表明模型学习到有用的序列模式与多模态信号。

六、实现细节与稳定性技巧

- **掩码只作用于 Key 方向**：在注意力里只屏蔽 Key 的 padding，避免某些 Query 行全为 $-\infty$ 导致 `softmax` 输出 NaN。
- **RMSNorm 加入 `\varepsilon`**：提升数值稳定性。
- **忽略索引（ignore_index）**：标签填充使用 `-100` 并在交叉熵中忽略。
- **梯度裁剪**：可选 `clip_grad_norm_` 避免梯度爆炸。
- **学习率调度**：如 `ReduceLROnPlateau` 或 `CosineAnnealingLR`，帮助进一步收敛。

七、总体流程图（文字版）

1. 加载并整理用户交互 → 形成时间序列
2. 提取文本 `[CLS]` 嵌入、提取图像 `[CLS]` 嵌入
3. 多模态融合投影到 `hidden_dim`
4. 加位置编码 → 送入 Transformer 编码器（多层）
5. 序列输出与物品嵌入表点积 → 得到每位置的 `logits`
6. 计算位置对齐交叉熵（忽略填充）进行训练
7. 测试时取序列最后位置，计算 Top-K → 评估 HR@K、NDCG@K

八、关键超参数（示例）

- `hidden_dim`: 512
- `n_heads`: 8
- `n_layers`: 2
- `ffn_dim`: 2048
- `txt_dim`: 768, `img_dim`: 768
- `max_seq_len`: 50
- `dropout`: 0.1
- `learning_rate`: 3e-4（可结合调度器与 warmup）

九、结论

该系统通过将文本与图像的 `[CLS]` 表示在时间维度上融合，并使用 Transformer 捕捉用户序列中的时序与语义关系，最终以分类的方式进行下一物品预测。评估以 HR@K 与 NDCG@K 为主，直观反映推荐命中与排序质量。对于初学者，这是一个完整的从数据到模型、从训练到评估的多模态推荐实践范式。