媒体与认知课程上机作业

一、项目背景

多模态学习（Multimodal Learning）是当前人工智能领域的研究热点，其中视觉与语言的融合任务，如图文匹配、图文检索、图文生成等，具有广泛的应用前景。

CLIP（Contrastive Language–Image Pretraining）是 OpenAI 提出的多模态预训练方法，利用图像-文本对比学习，使模型能自动将图像和文本嵌入到同一语义空间中，进而支持多种下游任务。

本项目旨在指导学生构建一个简化版本的 CLIP 模型，掌握对比学习的基本方法和图文模态对齐的核心原理。

二、项目目标

- 搭建一个具备图文对比学习能力的基础模型；
- 使用 PyTorch 实现图像编码器与文本编码器的双塔结构；
- 使用 InfoNCE 损失进行图文嵌入对齐训练；
- 分析模型在多模态语义对齐方面的效果及可视化结果。

三、技术路线与方法

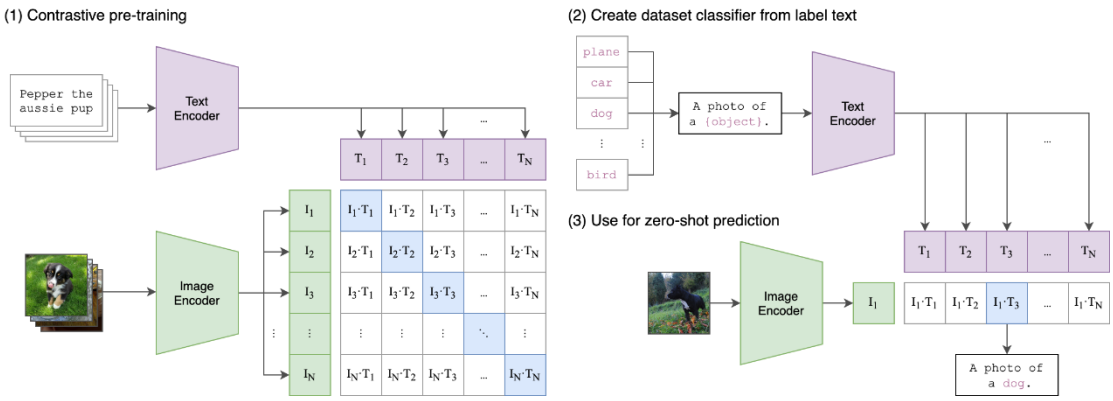
1. 数据集选择

使用 Flickr8k 数据集，数据格式为图像 + 文字描述（1-5 条）。数据集下载可通过清华云盘链接：

https://cloud.tsinghua.edu.cn/f/6e5dcf45eac345649665/

请将文件下载后放在\Flickr8k\images 目录下

2. 模型结构



a. 图像编码器【Task1】

- 使用 ResNet18 进行特征提取，输出特征向量；
- 添加线性层映射到共享语义空间。

b. 文本编码器【Task2】

- 使用 LSTM 或 Transformer 提取文本表示；
- 线性层将文本特征映射到相同维度的共享空间。

c. 对比损失【Task3】

使用 InfoNCE 损失函数对图文嵌入对进行正负对比学习，目标是：

- 同一图文对相似度最大
- 不同对之间的相似度最小

图像编码为z_i，文本编码为z_j

相似度用余弦相似度

$$sim(z_i, z_j) = \frac{z_i \cdot z_j}{|z_i|\,|z_j|}$$

则 InfoNCE 损失为（图像到文本方向）

$$\mathcal{L}_{i\text{mage}\to tx} = -\log \frac{\exp\left(\frac{\text{sim}(z_i^I, z_i^T)}{\tau}\right)}{\sum_{j=1}^{N} \exp\left(\frac{\text{sim}(z_i^I, z_j^T)}{\tau}\right) \exp\left(\frac{\text{sim}(z_i^I, z_j^T)}{\tau}\right)}$$

双向损失函数为：

$$\mathcal{L} = \frac{1}{2} \quad (\mathcal{L}_{i\text{mage}\to tx} + \mathcal{L}_{tx\to i\text{mage}})$$

请同学们完成代码中 ResNet、LSTM 模型（或者 Transformer 模型）的搭建以及 InfoNCE 损失函数的编写，即可开始训练

3. 训练流程

1. 图像 + 文本 输入模型
2. 提取图像特征 / 文本特征
3. 归一化后计算余弦相似度矩阵
4. 计算损失并反向传播
5. 每轮评估 Top-1 Accuracy / Recall@K 检索准确率，保存最优模型（此处需要同学们在 train.py 中修改）

4. 结果分析与可视化【Task4】

- 列出模型训练结束后的性能指标：Top-1 Accuracy / Recall@K
- 请编写文本 → 图像 Top-K 检索示例的可视化代码，例如：

- 结合模型训练结果和可视化分析的情况，尝试分析模型在文图检索任务上的表现，比如对文本描述或者图像类型是否存在偏好，预测结果好/不好可能的原因是什么，哪些方向可以进行改进等等
- T-SNE 可视化嵌入空间可视化（选做）

5. 可尝试改进的方向（建议）【Task5】

模型结构优化：

- 尝试不同的网络作为图像编码器，或者使用预训练模型

- 对文本编码器使用 BERT 结构，探索预训练文本模型在多模态任务中的表现。

数据增强与正则化：

- 对图像进行各种数据增强（旋转、缩放、色彩变换等），提升模型鲁棒性。
- 对文本进行同义词替换、随机删除等方式，增强文本描述的多样性。

损失函数与学习率调节：

- 结合多种损失函数探索对比学习效果。
- 采用动态学习率调整策略优化训练过程。

同学们可探索更多可改进的方式来提高最终性能指标

四、附录环境配置

1. 安装 Conda
   Anaconda 是一个专门为科学计算设计的 Python 发行版，它提供了统一的环境管理功能，支持 Linux、Mac 和 Windows 平台。内置许多科学计算和数据分析的 Python 库。

   Miniconda 是 Anaconda 的一个轻量级版本，它默认只包括 Python 和 Conda。用户可以通过 Conda 或 Pip 安装所需的其他库。推荐使用 Miniconda 来构建项目，因为它允许用户在创建新环境时按需添加必要的依赖。

2. 创建虚拟环境
   创建名字为 py38，python 版本为 3.8 的虚拟环境指令：

   conda create --name py38 python=3.8

   激活该虚拟环境：conda activate py38

3. 安装 pytorch，下面以 torch 版本为 2.0.1 为例
   GPU 版本安装指令：

   conda install pytorch==2.0.1 torchvision==0.15.2 torchaudio==2.0.2 -c pytorch

   CPU 版本安装指令：

   conda install pytorch==2.0.1 torchvision==0.15.2 torchaudio==2.0.2 cpuonly -c pytorch

4. 安装作业环境所需依赖
   pip install -r requirements.txt

五、作业提交要求

本次作业，分两个阶段提交：

- 2025.5.16 前完成基础部分（Task1 ~ Task4）的实现和实验结果。请将实验报告与代码文件打包压缩（不要包含数据集文件），统一命名为‘姓名_学号_上机作业中期结果.zip’
- 2025.6.1 前完成改进部分，并完成最终报告（在之前报告基础上进行改进部分的补充）和代码。请将实验报告与代码文件打包压缩（不要包含数据集文件），统一命名为‘姓名_学号_上机作业.zip’

六、实验报告撰写要求

实验报告应包括以下几个方面的内容，要求语言清晰、结构合理，重点突出对实验过程与结果的理解与分析：

1. 整体方案理解
   简要说明实验的任务目标与整体流程，包括模型结构、关键模块的设计意图与功能理解，展示你对所实现方法的全面把握。

2. 实验过程描述
   介绍你在实验中所做的关键尝试与改进，包括模型设计上的调整、参数选择、训练策略优化等。建议结合代码细节说明你的思路。

3. 实验结果分析
   展示主要实验结果，并对其进行分析和比较。可以包括模型性能指标（如准确率、损失曲线等）、不同方案下的表现差异等，体现你对结果的思考。

4. 可视化展示（如使用）
   建议用图表、曲线或示例输出等方式直观展示实验效果，加深对模型行为的理解。例如：训练过程的 loss 曲线、样本嵌入可视化、预测结果与真实标签的对比等。

5. 总结与反思
   总结实验中取得的效果与存在的不足，思考可能的改进方向或未来的扩展思路。

写作建议：不追求华丽辞藻，关键是“把你做了什么、为什么这么做、效果如何”讲清楚。真实反映你的实验过程和理解深度即可。

参考文献

[1]Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PmLR, 2021: 8748-8763.