

《人工智能导论》大作业

任务名称：不良内容图像检测

完成组号：5

小组人员：陈楠、瞿晨晔、池承运、李越凡

完成时间：2024.06.14

1. 任务目标

通过训练分类模型，在拥有一定鲁棒性的同时，实现区分正常图像与不良图像。

2. 具体内容

2.1 实施方案

本项目开发了一个基于PyTorch Lightning的暴力图像检测模型。使用预训练的ResNet-18模型，通过自定义数据模块 CustomDataModule 进行数据加载与预处理。模型定义在 model.py 中，采用交叉熵损失、Adam优化器，并计算准确率和AUROC。在 train.py 中设置训练器，进行模型训练，保存验证集损失最小的检查点。在 classify.py 中，从检查点加载已训练模型，并定义图像分类函数 classify，对输入张量进行分类。test.py 则加载检查点进行模型测试，评估其在测试集上的性能。

2.2 核心代码分析

1. 模型定义 (model.py)

ViolenceClassifier 类基于预训练的 ResNet18 构建。模型的全连接层被调整为输出两个类（不良与正常）。损失函数使用交叉熵，评价指标包括准确率（Accuracy）和多分类 AUROC。模型的优化器采用Adam，学习率设置为 $1e-3$ 。

2. 训练过程 (train.py)

训练脚本初始化数据模块 CustomDataModule 和模型 ViolenceClassifier，设置 GPU 设备及批量大小。训练器 (Trainer) 通过 ModelCheckpoint 回调监测验证集损失，保留性能最佳的模型。训练过程使用 GPU 加速，最大训练轮数为 40 轮。模型和数据模块的配置在训练过程中被用于拟合训练数据。

3. 分类功能 (classify.py)

分类功能模块加载最佳检查点的模型，用于图像分类。在推理时，使用训练器的 predict 方法进行批量预测。分类逻辑通过比较每个样本的分类得分确定其类别（0 表示正常，1 表示不良）。输出为每个图像的分类标签。

4. 测试过程 (test.py)

测试脚本用于评估模型在测试集上的性能。加载与训练过程相同的数据模块和模型检查点。测试阶段同样采用 GPU 加速，使用训练器的 `test` 方法在测试数据上运行模型，并计算相关性能指标（如测试集的损失和准确率）。

3. 工作总结

(1) 收获、心得

通过此次项目，我们收获了很多关于图像分类和深度学习的知识。我们学习了如何使用 `PyTorch Lightning` 进行高效的模型训练，并熟悉了 `ResNet` 等经典模型的应用。团队合作使我们能够相互学习，解决了许多技术难题。

在项目实施过程中，我们深刻理解了数据预处理和模型调优的重要性。通过反复实验和调参，我们提升了模型的准确率和鲁棒性。此外，我们还积累了丰富的代码实践经验，为以后的研究和工作打下了坚实的基础。

(2) 遇到问题及解决思路

在项目中，我们遇到了以下问题：

- 数据不足：为了增强模型的泛化能力，我们通过数据增强技术（如旋转、裁剪等）扩展了数据集。
- 模型过拟合：我们通过正则化技术和早停策略来防止模型过拟合。
- 不平衡数据：我们通过调整损失函数的权重和平衡采样策略，来应对不平衡数据带来的挑战。

4. 课程建议

- 增加实战项目的数量和多样性，帮助学生更好地理解理论知识并应用到实践中。
- 引入更多关于模型调优和性能评估的内容，提升学生在实际问题中的能力。
- 充分利用实践平台，在上面布置更多有关模型实践的题目。
- 在课程中采取团队合作，通过小组项目和讨论，让学生在交流中获得更多的学习机会。
- 提供云计算平台和高性能计算资源，支持学生尝试进行大规模数据处理和模型训练。