Cat_Dog_Classification

1. 项目简介

基于MindsSpore框架实现了一个简单的二分类猫狗分类器，使用ResNet50网络进行训练，ONNX进行模型部署，并通过PyQt5进行简单的UI界面展示

2. 环境配置

本项目基于Linux环境进行训练得到权重文件并导出ONNX，在Win11进行模型部署和测试。将环境配置文件编写为Shell脚本，即：env.sh文件，通过如下命令进行环境部署

```
bash env.sh
```

ONNX文件（放于项目根目录下）：

链接：https://pan.baidu.com/s/17xlq1hValsjosvCEy0CgJw?pwd=2qpo
提取码：2qpo

CheckPoint模型权重文件（放于项目根目录下）：

链接：https://pan.baidu.com/s/1Q2Fhbb3dcSG9CQT0DVWsNw?pwd=0qnr

提取码：0qnr

数据集（放于DataSet/PetImages目录下）：

Kaggle猫狗数据集

3. 目录介绍

```
|-- cnn101.py
    |-- cnn152.py
    |-- cnn50.py
    |-- env.sh
    |-- export.py
    |-- file_list.txt
    |-- infer.py
    |-- onnx_infer.py
```

```
    |-- process_data.py
    |-- README.md
    |-- resnet.py
    |-- resnet101.py
    |-- resnet152.py
    |-- train.py
    |-- train_256.py
    |-- train_adam.py
    |-- train_attention.py
    |-- train_cnn.sh
    |-- train_continue.py
    |-- train_test.py
    |-- train_transfer.py
    |-- val.py
    |-- dataset
    |   |-- PetImages
    |       |-- clean.py
    |       |-- partition.py
    |       |-- Cat
    |       |-- Dog
    |-- log
    |   |-- file_list.txt
    |   |-- file_path.py
    |   |-- cnn101_lr0.1_bs64
    |   |   |-- train.log
    |   |-- cnn152_lr0.1_bs64
    |   |   |-- train.log
    |   |-- cnn50_lr0.1_bs64
    |   |   |-- train.log
    |   |-- resnet101_lr0.1_bs64
    |   |   |-- train.log
    |   |-- resnet152_lr0.1_bs64
    |   |   |-- train.log
    |   |-- resnet50_attention_lr0.1_bs256
    |   |   |-- train.log
    |   |-- resnet50_lr0.001_bs64
    |   |   |-- train.log
    |   |-- resnet50_lr0.001_opt-adam_bs64
    |   |   |-- train.log
    |   |-- resnet50_lr0.01_bs256
    |   |   |-- train.log
    |   |-- resnet50_lr0.01_bs64
    |   |   |-- train.log
    |   |-- resnet50_lr0.01_opt-adam_bs64
    |   |   |-- train.log
    |   |-- resnet50_lr0.1_bs256
    |   |   |-- train.log
    |   |-- resnet50_lr0.1_bs64
    |       |-- train.log
    |-- model_utils
    |   |-- config.py
    |   |-- config
```

```
|   |   |-- resnet101_imagenet2012_config.yaml
|   |   |-- resnet152_imagenet2012_config.yaml
|   |   |-- resnet18_cifar10_config.yaml
|   |   |-- resnet18_cifar10_config_gpu.yaml
|   |   |-- resnet18_imagenet2012_config.yaml
|   |   |-- resnet18_imagenet2012_config_gpu.yaml
|   |   |-- resnet34_imagenet2012_config.yaml
|   |   |-- resnet50_cifar10_config.yaml
|   |   |-- resnet50_imagenet2012_Ascend_Thor_config.yaml
|   |   |-- resnet50_imagenet2012_Boost_config.yaml
|   |   |-- resnet50_imagenet2012_config.yaml
|   |   |-- resnet50_imagenet2012_GPU_Thor_config.yaml
|   |   |-- resnet_benchmark_GPU.yaml
|   |   |-- se-resnet50_imagenet2012_config.yaml
|   |-- __pycache__
|       |-- config.cpython-37.pyc
|       |-- config.cpython-39.pyc
|-- output
|-- plot_log
|   |-- file_list.txt
|   |-- plot_log.py
|   |-- csv_data
|   |   |-- cnn101_lr0.1_bs64
|   |   |   |-- training_data.csv
|   |   |-- cnn152_lr0.1_bs64
|   |   |   |-- training_data.csv
|   |   |-- cnn50_lr0.1_bs64
|   |   |   |-- training_data.csv
|   |   |-- resnet101_lr0.1_bs64
|   |   |   |-- training_data.csv
|   |   |-- resnet152_lr0.1_bs64
|   |   |   |-- training_data.csv
|   |   |-- resnet50_attention_lr0.1_bs256
|   |   |   |-- training_data.csv
|   |   |-- resnet50_lr0.001_bs64
|   |   |   |-- training_data.csv
|   |   |-- resnet50_lr0.001_opt-adam_bs64
|   |   |   |-- training_data.csv
|   |   |-- resnet50_lr0.01_bs256
|   |   |   |-- training_data.csv
|   |   |-- resnet50_lr0.01_bs64
|   |   |   |-- training_data.csv
|   |   |-- resnet50_lr0.01_opt-adam_bs64
|   |   |   |-- training_data.csv
|   |   |-- resnet50_lr0.1_bs256
|   |   |   |-- training_data.csv
|   |   |-- resnet50_lr0.1_bs64
|   |       |-- training_data.csv
|   |-- img
|       |-- cnn101_lr0.1_bs64
|       |   |-- training_validation_plot.png
|       |-- cnn152_lr0.1_bs64
```

```
|   |   |-- training_validation_plot.png
|   |-- cnn50_lr0.1_bs64
|   |   |-- training_validation_plot.png
|   |-- resnet101_lr0.1_bs64
|   |   |-- training_validation_plot.png
|   |-- resnet152_lr0.1_bs64
|   |   |-- training_validation_plot.png
|   |-- resnet50_attention_lr0.1_bs256
|   |   |-- training_validation_plot.png
|   |-- resnet50_lr0.001_bs64
|   |   |-- training_validation_plot.png
|   |-- resnet50_lr0.001_opt-adam_bs64
|   |   |-- training_validation_plot.png
|   |-- resnet50_lr0.01_bs256
|   |   |-- training_validation_plot.png
|   |-- resnet50_lr0.01_bs64
|   |   |-- training_validation_plot.png
|   |-- resnet50_lr0.01_opt-adam_bs64
|   |   |-- training_validation_plot.png
|   |-- resnet50_lr0.1_bs256
|   |   |-- training_validation_plot.png
|   |-- resnet50_lr0.1_bs64
|       |-- training_validation_plot.png
|-- __pycache__
    |-- resnet.cpython-39.pyc
```

4. 代码说明

DataSet:

`clean.py` 为数据清洗脚本，保证图片格式为JPEG格式，删除其他非法格式

`partition.py` 为数据集划分脚本，9:1划分训练集和验证集

`pachong.py` 为爬虫程序，在Win11环境下的个人PC中，通过添加Chrome驱动，修改查询词，即可将对应的图片下载到本地

Train:

`train.py` 为训练脚本，定义超参，其他脚本文件类似，不过是修改了部分参数

`export.py` 为导出脚本，将训练好的best.ckpt文件导出ONNX用于下一步的模型部署和UI展示

Log:

`该部分为模型训练过程的日志保存文件夹`

Output:

`该部分为模型权重文件保存文件夹`

Plot_Log：

`该部分为模型训练的Loss和Acc数据处理和可视化部分，通过运行plot_log.py创建img文件夹和csv文件夹，保存可视化结果和对应的数据`

Inner:

`infer.py` 可进行推理，但没有UI界面

部署+UI：

`onnx_infer.py` 该部分最好在Win个人PC中运行（本人在服务器中没有root权限，环境没法配置），需要下载onnxruntime、PyQt5

```
pip install onnxruntime
pip install PyQt5
```

UI界面展示：

Demo:

见Demo文件夹

5. 项目总结

本项目比较基础的实现了一个有UI界面的二分类ResNet网络分类器，主要在训练网络过程中尝试了改进网络结构、调整超参、进行图像增强、加入Attention机制等方式，同时熟悉了ONNX部署模型，后续可作为Web后端做成一个完整的项目，并通过uni-app打包实现多平台部署。

水平有限，能力不足，欢迎issue、fork和star，给个免费的小星星！