大作业-实验文档

潘林朝

实验内容

使用华为提出的MindSpore深度学习框架,完成面向猫狗分类任务的网络训练和验证,并转换为相应的格式,部署到边缘设备上。

环境搭建(windows)

安装Anaconda

打开搜索引擎,搜索**清华源镜像网站**,搜索镜像 anaconda,然后选择 archive 文件夹,选择合适的版本(访问<u>下载地址</u>),下载安装包后,选择相应的路径进行安装。

提示窗口的环境变量可以不添加。

配置Mindspore(CPU)环境

点击win+s 搜索 Anaconda Prompt (Anaconda3), 打开该终端窗口。关闭电脑的网络代理, 比如 Clash For Windows 等软件。在终端输入下面的命令完成Mindspore环境创建。

```
conda create -n MindSpore python==3.7.5
conda activate MindSpore

# 后面使用pip安装python包,需要pip换源
# open C:/Users/your-user-name/pip/pip.ini
# past the content below
[global] index-url =
https://mirrors.huaweicloud.com/repository/pypi/simple
trusted-host = mirrors.huaweicloud.com timeout = 120
```

```
# alternatively, you can switch to tsinghua site
[global] index-url =
https://pypi.tuna.tsinghua.edu.cn/simple

# run this command in Anaconda Prompt (Anaconda3) with
MindSpore activated
pip install mindspore==1.5.2

# check if successfully installed
python -c "import mindspore; mindspore.run_check()"
```

可选:

在pycharm添加该环境作为当前项目的解释器。打开**File-**>**Settings->Python Interpreter** ,可点击当前环境的设置,点击**Add** 。

学习MindSpore

具体可参考华为官方提供的<u>tutorial</u>。

Tensor

与TensorFlow类似,为了构建模型的计算图,需要使用Tensor这种基本数据结构。

- 声明和使用: Tensor(np.zeros([1, 2, 3]), ms.float32)
- 属性: default_input, name, requires_grad,
 layerwise_parallel
- 常见操作: asnumpy(), size(), dim(), dtype, shape,set_dtype()

Operation

算子可以理解为计算图如何连接向量,即计算图上的一个节点Node。f分类为与Array相关,与数据计算相关,网络类,控制类,其它。示例代码:

```
data1 = Tensor(np.array([[0, 1], [2,
1]]).astype(np.int32))
data2 = Tensor(np.array([[0, 1], [2,
1]]).astype(np.int32))
op = P.Concat()
output = op((data1, data2))

cos = P.Cos()
input_x = Tensor(np.array([0.24, 0.83, 0.31, 0.09]),
mindspore.float32)
output = cos(input_x)

input_x = Tensor(np.array([-1, 2, -3, 2, -1]),
mindspore.float16)
relu = nn.ReLU()
relu(input_x)
```

Cell

感觉上与Pytorch的nn.Module类似,继承Cell后,___init___初始化该计算的相关属性,construct 定义该计算如何执行,与Pytorch的forward 类似。

常用模块

- mindspore.dataset
- mindspore.common

- mindspore.context
- mindspore.nn
- mindspore.ops
- mindspore.train

开发流程

- 1. 数据处理
 - 数据加载/数据增强
- 2. 模型定义
 - 定义网络/损失函数/优化器
- 3. 模型训练
 - Loss监控/验证/保存Checkpoint
- 4. 模型推理
 - 推理/部署

通过LetNet5分类MNIST数据集,详见相应的开发示例数程。

猫狗分类实验流程(实验主要内容)

详见华为官方提供的教程。

下载实验所需的工程文件和数据集

- 工程文件链接
- 开源数据集<u>链接</u>

或者群里下载提供的完整压缩包。

注意: 代码路径需要是英文。

安装必要的包

在MindSporePetClassification/code 下,打开Anaconda Prompt (Anaconda3),并激活创建的MindSpore 环境,运行pip install -r requirements.txt 命令以安装必要的包。

```
conda activate MindSpore
pip install -r requirements.txt
# easydict used in src/config.py
pip install easydict
```

数据预处理

在 MindSporePetClassification 路径下,终端运行下面的命令以预处理数据集(注意对应的数据集名称)。

```
python .\code\preprocessing_dataset.py
.\kagglecatsanddogs_3367a.zip
```

运行后,数据集以 9:1 比例划分训练集和验证集,该比例可在 preprocessing_dataset.py 文件中更改,默认设置 split_dataset 的 eval_split=0.1。

训练模型

解压后的文件缺少 mobilenet V2.ckpt ,也就是保存了预训练 MobileNet V2的参数文件,可点击链接下载,然后将下载的文件保存到 MindSporePetClassification 路径下,并修改文件名为 mobilenet V2.ckpt。

在 MindSporePetClassification 路径下,终端运行下面的命令以训练模型。

出现六张图片的窗口表示预训练模型的预测结果,关闭该窗口后,在训练过程中会再次出现该窗口,表明fine-tuning后的预测结果,后关闭窗口即可继续进行模型训练。

最后会得到mobilenetv2.mindir的MindlR模型文件。

转换模型格式

因为部署到边缘设备需要的是ms格式模型,所以需要使用MindSpore Lite 转换上面的mobilenetv2.mindir 为ms模型。

进入 MindSporePetClassification/converter 文件夹中,打开cmd终端,运行下面的命令完成转换。

```
.\converter_lite.exe --fmk=MINDIR --
modelFile=../mobilenetv2.mindir --outputFile=pet
```

运行成功后,在MindSporePetClassification/converter 文件夹中会有一个pet.ms 文件。

converter的版本需要对应mindspore的版本,如果不符合,则进行替换,访问链接,查找对应的操作系统,硬件架构和版本,下载包含converter工具的压缩包,提取并替换上述文件夹内的converter。

下载app进行模型部署

APP的地址<u>下载地址</u>,安装至手机后,连接电脑,传输上一步导出的 pet.ms 文件到手机的 PetClassification 文件夹。

手机设置安装应用的读写手机存储的权限为【始终允许】,在android12的小米手机上,手机存放该模型的路径是/内部存

储/Android/data/com.mindspore.classificationforpet/files/

,建议使用手机连接电脑进行文件传输。

注意:这里的APP仅适用于安卓手机。

需要在实验报告中截图展示相应的识别结果。

必需/建议完成的实验内容

- 1. 分别选择10张猫和狗的图片,使用网络进行预测,统计预测准确率,或者可视化预测结果;
 - 查看train.py文件中的get_samples_from_eval_dataset 函数;
 - 设置相应的10张猫/狗图片路径,替换test_list,修改utils.py的predict_from_net中subplot的参数(2,3,i+1)为(2,10,i+1);
- 2. 比较使用fine-tuning和未使用fine-tuning的模型预测性能,说明现象;
 - 默认使用fine-tuning, 注释load_ckpt语句即不使用预训练模型;
- 3. 修改网络训练参数/网络结构参数,探索这些参数对模型性能的影响;
 - 在train.py文件中的 set_config 函数, 例如batch_size/epoch_size/warmup_epochs;
 - models.py文件中, 186行的early stop的参数 5;
 - 默认使用mobilenetv2架构,尝试替换为v3?