算能-超分辨率重建模型迁移DEMO

本文以开源REAL-ESRGAN为例，向选手介绍本次竞赛题目完整的实现流程。

其中upscale.py，fix.py仅为示例演示，并非固定使用，选手可自行更改，pretrained_model中提供的预训练模型源码仓库如下：

REAL-ESRGAN: https://github.com/XPixelGroup/BasicSR/tree/master/basicsr

1 搭建本地开发环境

1.1 本地开发环境需求

- 开发主机：一台安装了Ubuntu18.04或Ubuntu 20.04的x86架构下的64位系统主机，运行内存8GB以上（小于8G可能会在量化时中断）。
- 算能docker镜像: sophgo/tpuc_dev:v2.2
- 算能sdk：tpu-mlir_v1.2.8-g32d7b3ec-20230802.tar.gz

1.2 安装docker

参考《官方教程》，若已经安装请跳过。

```
#安装docker
sudo apt-get install docker.io
sudo systemctl start docker
sudo systemctl enable docker

#创建docker用户组，之后docker命令可免root权限执行。（若已有docker组可忽略）
sudo groupadd docker

# 将当前用户加入docker组
sudo gpasswd -a ${USER} docker

# 重启docker服务
sudo service docker restart

# 切换当前会话到新group或重新登录重启X会话
newgrp docker
```

1.3 配置开发环境

```
#将压缩包解压到tpu-mlir
cd tpu-mlir_<date>_<hash>
mkdir tpu-mlir
tar zxvf tpu-mlir_v1.2.8-g32d7b3ec-20230802.tar.gz --strip-components=1 -C tpu-mlir

#创建docker容器并进入docker
cd tpu-mlir
#如果当前系统没有对应的镜像，会自动从docker hub上下载；
#此处将tpu-mlir的当前目录映射到docker内的/workspace目录
#myname只是举个名字的例子，请指定成自己想要的容器的名字
docker run --privileged --name myname -v $PWD:/workspace -it sophgo/tpuc_dev:v2.2

#初始化软件环境
cd /workspace/tpu-mlir
source ./envsetup.sh
```

至此，开发环境已经配置完成，可以开始模型迁移啦！

2 模型转换

首先选用一个开源模型，将预训练的模型转换成可以在算能TPU上运行的bmodel形式，本文以REAL-ESRGAN网络为例。

```
#模型转换命令
model_transform.py \
 --model_name r-esrgan \
 --input_shape [[1,3,200,200]] \
 --model_def r-esrgan4x+.pt \
 --mlir r-esrgan4x.mlir

#将mlir文件转换成f16的bmodel
model_deploy.py \
 --mlir r-esrgan4x.mlir \
 --quantize F16 \
 --chip bm1684x \
 --model resrgan4x.bmodel
```

更多模型编译案例可参考TPU-MLIR快速入门手册 。

3 模型推理

推理部分均在算能云开发空间内完成，云空间的使用与环境配置请参考CCF BDCI-算能竞赛云空间使用说明。

3.1 准备工作

将本地编译的模型、测试集及所需相关文件上传至云空间：

- 本地 bmodel

- 测试集地址：可从竞赛官网处下载获取
- 相关文件：https://github.com/sophgo/TPU-Coder-Cup/blob/main/CCF2023/npuengine.py

  https://github.com/sophgo/TPU-Coder-Cup/blob/main/CCF2023/sophon-0.4.8-py3-none-any.whl

  https://github.com/sophgo/TPU-Coder-Cup/tree/main/CCF2023/metrics

---

3.2 使用sail加载bmodel进行推理

选手可选择使用fp32、int8、fp16模式，最终只选择一种推理结果进行提交，推理结果命名为test.json。

加载fp16bmodel，生成超分辨率图像，并输出推理时间和niqe值：

```
python3 upscale.py \
--model_path models/resrgan4x.bmodel \
--input dataset/test \
--output results/test_fix \
--report results/test.json
```

其中，导入bmodel：

```
#npuengine.py中的EngineOV
self.model = sail.Engine(model_path, device_id, sail.IOMode.SYSIO)
```

计算重构图片的niqe值：

```
#upscale.py
output = cv2.imread(output_path)
with warnings.catch_warnings():
  warnings.simplefilter('ignore', category=RuntimeWarning)
  niqe_output = calculate_niqe(output, 0, input_order='HWC', convert_to='y')
```

输出结果文件所需的参数：

```
#upscale.py
#A榜结果
params = {"A": [{"model_size": model_size, "time_all": time_all, "runtime_avg":
format(runtime_avg, '.4f'),
                    "niqe_avg": format(niqe_avg, '.4f'), "images": result}]}
#B榜结果
params = {"B": [{"model_size": model_size, "time_all": time_all, "runtime_avg":
format(runtime_avg, '.4f'),
                    "niqe_avg": format(niqe_avg, '.4f'), "images": result}]}
```

至此，你已经成功生成了test.json文件，将该文件提交到CCFBDCI官方竞赛系统，看看自己的得分吧！

附录：

- TPU-MLIR学习资料：https://tpumlir.org/index.html
- TPU-MLIR开源仓库：https://github.com/sophgo/tpu-mlir
- TPU-MLIR学习视频：https://space.bilibili.com/1829795304/channel/collectiondetail?sid=734875
- TPU-MLIR入门手册：https://tpumlir.org/docs/quick_start/index.html