

AI学习笔记--Tensorflow--TPU 使用

张量处理单元 (TPU) 是一种定制化的 ASIC 芯片, 它由谷歌从头设计, 并专门用于机器学习工作负载。TPU 为谷歌的主要产品提供了计算支持, 包括翻译、照片、搜索助理和 Gmail 等。Cloud TPU 将 TPU 作为可扩展的云计算资源, 并为所有在 Google Cloud 上运行尖端 ML 模型的开发者与数据科学家提供计算资源。

TPU作为 Google 定制的 TensorFlow 作业芯片, 在运算加速方面具有很好的优势。首先可以打开官网查看更多信息。

<https://coral.ai>

官方提供的例子包括了:

- Minigo 阿帕奇狗, 围棋训练
- Smart Bird feeder 记录鸟类喂食场景, 需要对应的 board 和 camera
- Embedded Teachable Machine 机器学习例子
- Teachable Sorter
- Image classification 1000个 label 的图形识别
- object detection 运动目标识别
- Pose estimation 动作特征识别, 需要 board 和 camera

我们以最简单的 image classification 来作为示例, 了解 TPU 整个使用的方法。首先, TPU是一块 USB 外挂的设备。是需要使用 USB 驱动的, 这个驱动主要在 Google.Cloud 中, 作为国内大部分用户都无力去下载, 可以使用 VPN 加上终端代理的方式去下载。

```
# 添加软件源
→ AI echo "deb https://packages.cloud.google.com/apt coral-edgetpu-stable
main" | sudo tee /etc/apt/sources.list.d/coral-edgetpu.list
→ AI proxychains curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
# 更新软件源目录
→ AI proxychains sudo apt-get update
```

然后下载 TPU 的运行环境和软件工具:

```
→ AI proxychains sudo apt-get install libedgetpu1-std
```

TPU 默认使用的环境是 USB 3.0, 并且官方推荐的 USB 接口是 3.0 口, 可以在 lsusb 中查看对应设备是否已经接入到 PC:

```
→ AI lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 006: ID 174f:2410 Syntek
Bus 001 Device 005: ID 8087:0a2a Intel Corp.
Bus 001 Device 004: ID 0bda:0129 Realtek Semiconductor Corp. RTS5129 Card
Reader Controller
Bus 001 Device 003: ID 138a:0011 Validity Sensors, Inc. VFS5011 Fingerprint
Reader
#TPU 设备
```

```
Bus 001 Device 002: ID 1a6e:089a Global Unichip Corp.
```

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

如果在未 install 工具环境的情况下，这个识别应该是 Google。注：不安装工具并不影响实际的加速运行效果。

可选的环境还有一个 maximum operating frequency，应该是可以配置时钟片选信号用的工具，默认是 2 倍时钟信号。可以参考上文例子：

```
→ AI proxychains sudo apt-get install libedgetpu1-max
```

然后，在 install 一个 TensorFlow lite 的运行时库，保证 TPU 运行时的环境。注意：TPU 不具备终端训练模型能力，至少 2.0 版本没有，本运行环境是 TensorFlow-Lite 的 runtime 环境。

这里为了省事，建议配置 Python 3.5 以上版本，并且每个版本都有一个对应的 whl 作为依赖 API。需要 install 的统一在下面打了：

```
→ AI pip3 install --no-binary=:all: --force-reinstall numpy
→ AI pip3 install interpreter_wrapper
→ AI pip3 install tflite_runtime
→ AI pip3 install tensorflow
→ AI pip3 install tqdm
→ AI pip3 install absl-py
→ AI pip3 install virtualenv virtualenvwrapper
→ AI pip3 install /home/genesis/share/工具/tflite_runtime-1.14.0-cp35-cp35m-linux_x86_64.whl #Google 服务器拉的
```

Edge tpu api:

```
cd ~/
wget https://dl.google.com/coral/edgetpu_api/edgetpu_api_latest.tar.gz -O
edgetpu_api.tar.gz --trust-server-names
tar xzf edgetpu_api.tar.gz
cd edgetpu_api
bash ./install.sh
```

上述例子是 Minigo 项目所需要的依赖和其余没有 Dev board、camera 的独立项目所需要的依赖库。接着就可以使用第一个简单的例子了。首先在 GitHub 服务器上拉对应的代码：

```
mkdir coral    cd coral
git clone https://github.com/google-coral/tflite.git
```

然后，会找到 example 下游一个 classification 的项目，项目主要结构包括了 label、model、input (image) 这三个子目录。需要执行的是对环境的初始化操作，大部分都写在了 shell 中。运行一下即可：

```
cd tflite/python/examples/classification
bash install_requirements.sh
```

最后，执行相应的 Python 文件，导入对应的模型和图片：

```
python3 classify_image.py \  
--model models/mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite \  
--labels models/inat_bird_labels.txt \  
--input images/parrot.jpg
```

有兴趣的朋友可以把 label 对应的文字描述去掉，或者删掉Ara macao 这一项目，就会发现返回的键值对。



running 结果:

```
INFO: Initialized TensorFlow Lite runtime.
```

```
----INFERENCE TIME----
```

```
Note: The first inference on Edge TPU is slow because it includes loading
```

```
the model into Edge TPU memory.
11.8ms
3.0ms
2.8ms
2.9ms
2.9ms
-----RESULTS-----
Ara macao (Scarlet Macaw): 0.76562
```

如果想要自己体会 TPU 对逻辑运算加速的效果，可以使用本例中另一个模型来对比一下实验结果：

```
→ classification git:(master) x python3 classify_image.py \
--model models/mobilenet_v2_1.0_224_inat_bird_quant.tflite \
--labels models/inat_bird_labels.txt \
--input images/parrot.jpg
INFO: Initialized TensorFlow Lite runtime.
----INFERENCE TIME----
Note: The first inference on Edge TPU is slow because it includes loading
the model into Edge TPU memory.
104.3ms
70.8ms
52.3ms
52.1ms
52.2ms
-----RESULTS-----
Ara macao (Scarlet Macaw11111): 0.78906
→ classification git:(master) x python3 classify_image.py \
--model models/mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite \
--labels models/inat_bird_labels.txt \
--input images/parrot.jpg
INFO: Initialized TensorFlow Lite runtime.
----INFERENCE TIME----
Note: The first inference on Edge TPU is slow because it includes loading
the model into Edge TPU memory.
96.9ms
10.5ms
10.4ms
10.6ms
10.5ms
-----RESULTS-----
Ara macao (Scarlet Macaw11111): 0.78516
```

以上例子只是基础，在 TensorFlow-Lite 的运行环境中，可以不需要 edage 的工具，似乎是环境包含了这个软件部分。