tensorflow目标检测模型的压缩和ncnn转换

2017-08-24 | 深度学习 | 0 Comments

ncnn 是一个为手机端极致优化的高性能神经网络前向计算框架。ncnn 从设计之初深刻考虑手机端的部署和使用。无第三方依赖，跨平台，手机端 cpu 的速度快于目前所有已知的开源框架。支持 8bit 量化和半精度浮点存储，可导入 caffe和tensorflow 模型

编译ncnn

```
git clone https://github.com/Tencent/ncnn
sudo apt-get install libprotobuf-dev protobuf-compiler
cd ncnn
mkdir build && cd build
cmake ..
make -j
make install
```

进入 ncnn/build/tools 目录下，如下所示，我们可以看到已经生成了 caffe2ncnn 可ncnn2mem这两个可执行文件，这两个可执行文件的作用是将caffe模型生成ncnn 模型，并且对模型进行加密。在ncnn/build/tools/tensorflow下面也有tensorflow2ncnn，可以把tensorflow模型转化乘ncnn模型

tensorflow在arm上的安装

```
wget https://github.com/samjabrahams/tensorflow-on-raspberry-pi/releases/download/v0.11.0/tensorflow-0.11.0
sudo pip install tensorflow-0.11.0-cp27-none-linux_armv7l.whl
```

tensorflow的模型

ckpt, pb，meta等文件

1. the .ckpt file is the old version output of saver.save(sess), which is the equivalent of your .ckpt-data (see below)
2. the “checkpoint” file is only here to tell some TF functions which is the latest checkpoint file.
3. .ckpt-meta contains the metagraph, i.e. the structure of your computation graph, without the values of the variables (basically what you can see in tensorboard/graph).
4. .ckpt-data contains the values for all the variables, without the structure. To restore a model in python, you’ll usually use the meta and data files with (but you can also use the .pb file):

```
saver = tf.train.import_meta_graph(path_to_ckpt_meta)
saver.restore(sess, path_to_ckpt_data)
```

5. I don’t know exactly for .ckpt-index, I guess it’s some kind of index needed internally to map the two previous files correctly. Anyway it’s not really necessary usually, you can restore a model with only .ckpt-meta and .ckpt-data.

6. the .pb file can save your whole graph (meta + data). To load and use (but not train) a graph in c++ you’ll usually use it, created with freeze_graph, which creates the .pb file from the meta and data. Be careful, (at least in previous TF versions and for some people) the py function provided by freeze_graph did not work properly, so you’d have to use the script version. Tensorflow also provides a tf.train.Saver.to_proto() method, but I don’t know what it does exactly.

TF0.12以来的模型类型

Here’s my solution utilizing the V2 checkpoints introduced in TF 0.12.

There’s no need to convert all variables to constants or freeze the graph.

Just for clarity, a V2 checkpoint looks like this in my directory models:

```
checkpoint  # some information on the name of the files in the checkpoint
my-model.data-00000-of-00001  # the saved weights
my-model.index  # probably definition of data layout in the previous file
my-model.meta  # protobuf of the graph (nodes and topology info)
```

Python part (saving)

```
with tf.Session() as sess:
    tf.train.Saver(tf.trainable_variables()).save(sess, 'models/my-model')
```

If you create the Saver with tf.trainable_variables(), you can save yourself some headache and storage space. But maybe some more complicated models need all data to be saved, then remove this argument to Saver, just make sure you're creating the Saver after your graph is created. It is also very wise to give all variables/layers unique names, otherwise you can run in different problems.

Python part (inference)

```
with tf.Session() as sess:
    saver = tf.train.import_meta_graph('models/my-model.meta')
    saver.restore(sess, tf.train.latest_checkpoint('models/'))
    outputTensors = sess.run(outputOps, feed_dict=feedDict)
```

导出pb文件

```
python export_inference_graph.py \
  --alsologtostderr \
  --model_name=mobilenet_v1 \
  --image_size=224 \
  --output_file=/tmp/mobilenet_v1_224.pb
```

Exporting the Inference Graph

作者的pretrained model

frozen pb文件

```
python tensorflow/tensorflow/python/tools/freeze_graph.py \
--input_graph=models/slim/mobilenet_v1_224.pb \
--input_checkpoint=tmp_data/mobilenet_v1_1.0_224.ckpt \
--input_binary=true --output_graph=tmp_data/frozen_mobilenet.pb --output_node_names=mobilenetv1/Prediction
```

这里有个点在于怎么确定一个网络的output_node_names,参考

这个人自己搞的mobilenet

image_label

下载label数据

```
curl -L "https://storage.googleapis.com/download.tensorflow.org/models/inception_v3_2016_08_28_frozen.pb.tar.
  tar -C tensorflow/examples/label_image/data -xz
```

```
python tensorflow/tensorflow/examples/label_image/label_image.py --graph=tmp_data/inception_v3_2016_08_28_
```

reference

在树莓派上用TensorFlow玩深度学习

腾讯NCNN框架入门到应用

ncnn wiki

tensorflow模型的各种版本

tensorfow各种版本的模型

mobilenet的使用

retrain_mobilenet

MobileNet-SSD

从零开始码一个皮卡丘检测器-CNN目标检测入门教程

[Learning Note] Single Shot MultiBox Detector with Pytorch — Part 3

Building a Real-Time Object Recognition App with Tensorflow and OpenCV

How to train your own Object Detector with TensorFlow’s Object Detector API

Creating an Object Detection Application Using TensorFlow

Donate comment here

Donate

# deeplearning # tensorflow # 嵌入式 # 模型压缩

tensorflow训练-finetune-压缩模型

机器学习算法小结与对比

0条评论 DragonFive 1 登录

推荐 分享 评分最高

开始讨论…

通过以下方式登录

或注册一个 DISQUS 帐号 ?

姓名

来做第一个留言的人吧！

订阅 在您的网站上使用 Disqus添加 Disqus添加 隐私