omimo / **create_hellotensor.py**
Last active 20 days ago

A simple example for saving a tensorflow model and preparing it for using on Android

create_hellotensor.py

```python
# Create a simple TF Graph
# By Omid Alemi - Jan 2017
# Works with TF <r1.0

import tensorflow as tf

I = tf.placeholder(tf.float32, shape=[None,3], name='I') # input
W = tf.Variable(tf.zeros_initializer(shape=[3,2]), dtype=tf.float32, name='W') # weights
b = tf.Variable(tf.zeros_initializer(shape=[2]), dtype=tf.float32, name='b') # biases
O = tf.nn.relu(tf.matmul(I, W) + b, name='O') # activation / output

saver = tf.train.Saver()
init_op = tf.global_variables_initializer()

with tf.Session() as sess:
  sess.run(init_op)

  # save the graph
  tf.train.write_graph(sess.graph_def, '.', 'hellotensor.pbtxt')

  # normally you would do some training here
  # we will just assign something to W
  sess.run(tf.assign(W, [[1, 2],[4,5],[7,8]]))
  sess.run(tf.assign(b, [1,1]))

  #save a checkpoint file, which will store the above assignment
  saver.save(sess, 'hellotensor.ckpt')

```

create_model_tf1.py

```python
# Create a simple TF Graph
# By Omid Alemi - Jan 2017
# Works with TF r1.0

import tensorflow as tf

I = tf.placeholder(tf.float32, shape=[None,3], name='I') # input
W = tf.Variable(tf.zeros(shape=[3,2]), dtype=tf.float32, name='W') # weights
b = tf.Variable(tf.zeros(shape=[2]), dtype=tf.float32, name='b') # biases
O = tf.nn.relu(tf.matmul(I, W) + b, name='O') # activation / output

saver = tf.train.Saver()
init_op = tf.global_variables_initializer()

with tf.Session() as sess:
  sess.run(init_op)

  # save the graph
  tf.train.write_graph(sess.graph_def, '.', 'tfdroid.pbtxt')

  # normally you would do some training here
```

```
  # but fornow we will just assign something to W
  sess.run(tf.assign(W, [[1, 2],[4,5],[7,8]]))
  sess.run(tf.assign(b, [1,1]))

  #save a checkpoint file, which will store the above assignment
  saver.save(sess, 'tfdroid.ckpt')

```

prep_model.py

```
# Preparing a TF model for usage in Android
# By Omid Alemi - Jan 2017
# Works with TF <r1.0

import sys
import tensorflow as tf
from tensorflow.python.tools import freeze_graph
from tensorflow.python.tools import optimize_for_inference_lib


MODEL_NAME = 'hellotensor'

# Freeze the graph

input_graph_path = MODEL_NAME+'.pbtxt'
checkpoint_path = './'+MODEL_NAME+'.ckpt'
input_saver_def_path = ""
input_binary = False
output_node_names = "O"
restore_op_name = "save/restore_all"
filename_tensor_name = "save/Const:0"
output_frozen_graph_name = 'frozen_'+MODEL_NAME+'.pb'
output_optimized_graph_name = 'optimized_'+MODEL_NAME+'.pb'
clear_devices = True


freeze_graph.freeze_graph(input_graph_path, input_saver_def_path,
                          input_binary, checkpoint_path, output_node_names,
                          restore_op_name, filename_tensor_name,
                          output_frozen_graph_name, clear_devices, "")



# Optimize for inference

input_graph_def = tf.GraphDef()
with tf.gfile.Open(output_frozen_graph_name, "r") as f:
    data = f.read()
    input_graph_def.ParseFromString(data)

output_graph_def = optimize_for_inference_lib.optimize_for_inference(
        input_graph_def,
        ["I"], # an array of the input node(s)
        ["O"], # an array of output nodes
        tf.float32.as_datatype_enum)


# Save the optimized graph

f = tf.gfile.FastGFile(output_optimized_graph_name, "w")
f.write(output_graph_def.SerializeToString())

# tf.train.write_graph(output_graph_def, './', output_optimized_graph_name)

```

prep_model_tf1.py

```python
# Preparing a TF model for usage in Android
# By Omid Alemi - Jan 2017
# Works with TF r1.0

import sys
import tensorflow as tf
from tensorflow.python.tools import freeze_graph
from tensorflow.python.tools import optimize_for_inference_lib


MODEL_NAME = 'tfdroid'

# Freeze the graph

input_graph_path = MODEL_NAME+'.pbtxt'
checkpoint_path = './'+MODEL_NAME+'.ckpt'
input_saver_def_path = ""
input_binary = False
output_node_names = "O"
restore_op_name = "save/restore_all"
filename_tensor_name = "save/Const:0"
output_frozen_graph_name = 'frozen_'+MODEL_NAME+'.pb'
output_optimized_graph_name = 'optimized_'+MODEL_NAME+'.pb'
clear_devices = True


freeze_graph.freeze_graph(input_graph_path, input_saver_def_path,
                          input_binary, checkpoint_path, output_node_names,
                          restore_op_name, filename_tensor_name,
                          output_frozen_graph_name, clear_devices, "")



# Optimize for inference

input_graph_def = tf.GraphDef()
with tf.gfile.Open(output_frozen_graph_name, "r") as f:
    data = f.read()
    input_graph_def.ParseFromString(data)

output_graph_def = optimize_for_inference_lib.optimize_for_inference(
        input_graph_def,
        ["I"], # an array of the input node(s)
        ["O"], # an array of output nodes
        tf.float32.as_datatype_enum)

# Save the optimized graph

f = tf.gfile.FastGFile(output_optimized_graph_name, "w")
f.write(output_graph_def.SerializeToString())

# tf.train.write_graph(output_graph_def, './', output_optimized_graph_name)
```