TensorFlow - import meta graph and use variables from it

I'm training classification CNN using TensorFlow v0.12, and then want to create labels for new data using the trained model.

At the end of the training script, I added those lines of code:

```
saver = tf.train.Saver()
save_path = saver.save(sess,'/home/path/to/model/model.ckpt')
```

After the training completed, the files appearing in the folder are: 1. *checkpoint* ; 2. *model.ckpt.data-00000-of-00001* ; 3. *model.ckpt.index* ; 4. *model.ckpt.meta*

Then I tried to restore the model using the *.meta* file. Following this tutorial, I added the following line into my classification code:

```
saver=tf.train.import_meta_graph(savepath+'model.ckpt.meta') #line1
```

and then:

```
saver.restore(sess, save_path=savepath+'model.ckpt') #line2
```

Before that change, I needed to build the graph again, and then write (instead of line1):

```
saver = tf.train.Saver()
```

But, deleting the graph building, and using `line1` in order to restore it, raised an error. The error was that I used a variable from the graph inside my code, and the python didn't recognize it:

```
predictions = sess.run(y_conv, feed_dict={x: patches,keep_prob: 1.0})
```

The python didn't recognize the `y_conv` parameter. There is a way to restore the variables using the meta graph? if not, what os this restore helping, if I can't use variables from the original graph?

I know this question isn't so clear, but it was hard for me to express the problem in words. Sorry about it...

Thanks for answering, appreciate your help! Roi.

python variables tensorflow neural-network deep-learning

asked Feb 6 at 16:08

roishik
73 5

1 Answer

it is possible, don't worry. Assuming you don't want to touch the graph anymore, do something like this:

```
saver = tf.train.import_meta_graph('model/export/{}.meta'.format(model_name))
saver.restore(sess, 'model/export/{}'.format(model_name))
graph = tf.get_default_graph()
y_conv = graph.get_operation_by_name('y_conv').outputs[0]
predictions = sess.run(y_conv, feed_dict={x: patches,keep_prob: 1.0})
```

A preferred way would however be adding the ops into collections when you build the graph and then referring to them. So when you define the graph, you would add the line:

```
tf.add_to_collection("y_conv", y_conv)
```

And then after you import the metagraph and restore it, you would call:

```
y_conv = tf.get_collection("y_conv")[0]
```

It is actually explained in the documentation - the exact page you linked - but perhaps you missed it.

Btw, no need for the `.ckpt` extension, it might create some confusion as that is the old way of saving models.

answered Feb 6 at 17:24

Robert Lacok
713 2 10