x Dismiss

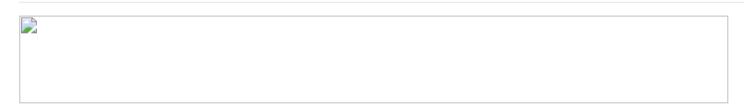
## Join the Stack Overflow Community

Stack Overflow is a community of 7.0 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

## Run multiple pre-trained Tensorflow nets at the same time



What I would like to do is to run multiple pre-trained Tensorflow nets at the same time. Because the names of some variables inside each net can be the same, the common solution is to use a name scope when I create a net. However, the problem is that I have trained these models and save the trained variables inside several checkpoint files. After I use a name scope when I create the net, I cannot load variables from the checkpoint files.

For example, I have trained an AlexNet and I would like to compare two sets of variables, one set is from the epoch 10 (saved in the file epoch\_10.ckpt) and another set is from the epoch 50 (saved in the file epoch\_50.ckpt). Because these two are exactly the same net, the names of variables inside are identical. I can create two nets by using

```
with tf.name_scope("net1"):
    net1 = CreateAlexNet()
with tf.name_scope("net2"):
    net2 = CreateAlexNet()
```

However, I cannot load the trained variables from .ckpt files because when I trained this net, I did not use a name scope. Even though I can set the name scope to "net1" when I train the net, this prevents me from loading the variables for net2.

I have tried:

```
with tf.name_scope("net1"):
    mySaver.restore(sess, 'epoch_10.ckpt')
with tf.name scope("net2"):
   mySaver.restore(sess, 'epoch_50.ckpt')
```

This does not work.

What is the best way to solve this problem?

tensorflow

asked Aug 26 '16 at 23:31



## 1 Answer

The easiest solution is to create different sessions that use separate graphs for each model:

```
# Build a graph containing `net1`.
with tf.Graph().as_default() as net1_graph:
  net1 = CreateAlexNet()
  saver1 = tf.train.Saver(...)
sess1 = tf.Session(graph=net1_graph)
saver1.restore(sess1, 'epoch 10.ckpt')
# Build a separate graph containing `net2`.
with tf.Graph().as_default() as net2_graph:
  net2 = CreateAlexNet()
  saver2 = tf.train.Saver(...)
```

```
sess2 = tf.Session(graph=net1_graph)
saver2.restore(sess2, 'epoch_50.ckpt')
```

If this doesn't work for some reason, and you have to use a single tf.session (e.g. because you want to combine results from the two network in another TensorFlow computation), the best solution is to:

- 1. Create the different networks in name scopes as you are already doing, and
- 2. Create separate tf.train.Saver instances for the two networks, with an additional argument to remap the variable names.

When constructing the savers, you can pass a dictionary as the var\_list argument, mapping the names of the variables in the checkpoint (i.e. without the name scope prefix) to the tf.Variable objects you've created in each model.

You can build the var\_list programmatically, and you should be able to do something like the following:

answered Aug 26 '16 at 23:53



Awesome answer! - denru Aug 27 '16 at 0:00

Stripping the prefix by using Istrip may cause incorrect results. Please use slicing instead. Other parts of the code work perfectly. Another question is that I found the name of a variable has a postfix like ":0", ":1". Do I need to get rid of this postfix before I store variables into a checkpoint file? — denru Aug 28 '16 at 0:30

Anyone tried this answer? I'm running into issues with the restore function not doing anything: stackoverflow.com/questions/41607144/... – TheCriticalImperitive Jan 12 at 11:47