Getting started with TensorBoard

Learn Deep Learning Denver 2017-09-14

Step 1: Tell TensorFlow your var names

- Step 1 is tell TensorFlow all your Python variable names!
- For example, instead of

-
  ```
  a = tf.placeholder(tf.float32)
  b = tf.placeholder(tf.float32)
  c = a + b
  ```
- Add names:

-
  ```
  a = tf.placeholder(tf.float32, name='a')
  b = tf.placeholder(tf.float32, name='b')
  c = tf.add(a,b, name='c')
  ```

Step 2: Make a log file directory

- Step 2 is to create a log file directory and point tf.Summary.FileWriter to it, along with your graph, so you can graph it out.
- You need to create the session first: sess = tf.Session()
- tf.summary.FileWriter(logdir, sess.graph)
- Additionally, you need to initialize the FileWriter, for some unknown reason, which you can do with the call to initialize all global variables: sess.run(tf.global_variables_initializer())
- Run your program to write the log files for TensorBoard.

Step 3: Run TensorBoard

- Step 3 is to run TensorBoard with your log file directory, and you should now be able to see a visualization of your computation graph:
- tensorboard --logdir=...your dir here…
- TensorBoard should respond with something like:
  TensorBoard 0.1.6 at http://yourmachine:6006
- Put the URL into your favorite browser, and presto!

Step 4: Group your stuff with scopes

- Step 4 is to group stuff with scopes. The way you do it is with the whacky python “with” keyword:
- with tf.name_scope(“ScopeName”):

Step 5: Make a scalar graph

- Step 5 is to make a graph of some scalar value; typically you run many iterations of the graph, such as for training your neural network, and you want to see a graph of some value, such as the cost function, on each iteration.
- Add summary nodes with tf.summary.scalar, e.g:
  ```
  cost_summary = tf.summary.scalar(cost)
  ```
- You can pass this to add_summary with sess.run():
  ```
  result = sess.run(cost_summary)
  summary_writer.add_summary(result, i) # i is iteration variable
  ```
- But, unless you’re graphing only one scalar, this won’t work...

Step 6: Graphing more than one thing

- Step 6 is to graph more than one thing by making a “merged” node.
- The way it works is, you create a new summary node that takes all the previous summary nodes as input. Then you ask this node to be calculated by passing it to sess.run(). Fortunately, you don’t have to do any of this merging yourself. All you have to do is call merge_all():
-
  ```
  merged = tf.summary.merge_all()
  result = sess.run(merged)
  summary_writer.add_summary(result, i)  # i is iteration variable
  ```

Step 7: Graphing more than one run

- TensorBoard can graph more than one run. The trick is to point each run to different log directories, then
- Point TensorBoard at a directory that’s the parent directory of all the subdirectories with all the different runs as the --logdir parameter
- The different runs will show up in the lower left on TensorBoard