黄世宇，清华大学贵系本科生，机器学习，深度学习
个人网站：http://huangshiyu.me

Tensorflow之Debug

参考资料:https://wookayin.github.io/tensorflow-talk-debugging

几种常用方法:

1.通过Session.run()获取变量的值

2.利用Tensorboard查看一些可视化统计

3.使用tf.Print()和tf.Assert()打印变量

4.使用Python的debug工具: ipdb, pudb

5.利用tf.py_func()向图中插入自定义的打印代码, tdb

6.使用官方debug工具: tfdbg

复习:

tensorflow是通过先建图再运行的方式进行运行,这就使得我们写在图建立过程中的输出语句在图运行的时候并不能得到执行,从而使得调试困难. 我们想在运行过程中,对训练的一些变量进行追踪和打印,对一些错误进行输出分析,下面介绍几种在tensorflow中进行debug的方法.

详细介绍:

1.通过Session.run()获取变量的值

一个简单的例子:

```
x = tf.placeholder(tf.float32)
y = tf.placeholder(tf.float32)
bias = tf.Variable(1.0)

y_pred = x ** 2 + bias     # x -> x^2 + bias
loss = (y - y_pred)**2     # l2 loss?

# Error: to compute loss, y is required as a dependency
print('Loss(x,y) = %.3f' % session.run(loss, {x: 3.0}))

# OK, print 1.000 = (3**2 + 1 - 9)**2
print('Loss(x,y) = %.3f' % session.run(loss, {x: 3.0, y: 9.0}))

# OK, print 10.000; for evaluating y_pred only, input to y is not required
print('pred_y(x) = %.3f' % session.run(y_pred, {x: 3.0}))

# OK, print 1.000 bias evaluates to 1.0
print('bias      = %.3f' % session.run(bias))
```

优点:简单易行

缺点:必须要在图中保持需要观察的变量,必须要进行一次完整的运行

比如我们要获取各个层的输出,会变得很繁琐:

搜索
谷歌搜索

随笔分类
Algorithm(14)
Android(1)
C(1)
C#(4)
C++(7)
Caffe(1)
Computational Biology & Bioinformatics(1)
Computer Vision(2)
Conference and Research(9)
Differential Geometry(10)
English(1)
Finance
Flask
Game Theory
GDB(1)
Git(4)
Google Cloud(4)
Introduction to Principle of Communication(14)

```
def alexnet(x):
    assert x.get_shape().as_list() == [224, 224, 3]
    conv1 = conv_2d(x, 96, 11, strides=4, activation='relu')
    pool1 = max_pool_2d(conv1, 3, strides=2)
    conv2 = conv_2d(pool1, 256, 5, activation='relu')
    pool2 = max_pool_2d(conv2, 3, strides=2)
    conv3 = conv_2d(pool2, 384, 3, activation='relu')
    conv4 = conv_2d(conv3, 384, 3, activation='relu')
    conv5 = conv_2d(conv4, 256, 3, activation='relu')
    pool5 = max_pool_2d(conv5, 3, strides=2)
    fc6 = fully_connected(pool5, 4096, activation='relu')
    fc7 = fully_connected(fc6, 4096, activation='relu')
    output = fully_connected(fc7, 1000, activation='softmax')
    return conv1, pool1, conv2, pool2, conv3, conv4, conv5, pool5, fc6, fc7

conv1, conv2, conv3, conv4, conv5, fc6, fc7, output = alexnet(images)  # ?!
```

```
_, loss_, conv1_, conv2_, conv3_, conv4_, conv5_, fc6_, fc7_ = session.run(
        [train_op, loss, conv1, conv2, conv3, conv4, conv5, fc6, fc7],
        feed_dict = {...})
```

可以通过词典或者新建一个类来改善这一情况:

词典:

```
def alexnet(x, net={}):
    assert x.get_shape().as_list() == [224, 224, 3]
    net['conv1'] = conv_2d(x, 96, 11, strides=4, activation='relu')
    net['pool1'] = max_pool_2d(net['conv1'], 3, strides=2)
    net['conv2'] = conv_2d(net['pool1'], 256, 5, activation='relu')
    net['pool2'] = max_pool_2d(net['conv2'], 3, strides=2)
    net['conv3'] = conv_2d(net['pool2'], 384, 3, activation='relu')
    net['conv4'] = conv_2d(net['conv3'], 384, 3, activation='relu')
    net['conv5'] = conv_2d(net['conv4'], 256, 3, activation='relu')
    net['pool5'] = max_pool_2d(net['conv5'], 3, strides=2)
    net['fc6'] = fully_connected(net['pool5'], 4096, activation='relu')
    net['fc7'] = fully_connected(net['fc6'], 4096, activation='relu')
    net['output'] = fully_connected(net['fc7'], 1000, activation='softmax')
    return net['output']

net = {}
output = alexnet(images, net)
# access intermediate layers like net['conv5'], net['fc7'], etc.
```

类:

```
class AlexNetModel():
    # ...
    def build_model(self, x):
        assert x.get_shape().as_list() == [224, 224, 3]
        self.conv1 = conv_2d(x, 96, 11, strides=4, activation='relu')
        self.pool1 = max_pool_2d(self.conv1, 3, strides=2)
        self.conv2 = conv_2d(self.pool1, 256, 5, activation='relu')
        self.pool2 = max_pool_2d(self.conv2, 3, strides=2)
        self.conv3 = conv_2d(self.pool2, 384, 3, activation='relu')
        self.conv4 = conv_2d(self.conv3, 384, 3, activation='relu')
        self.conv5 = conv_2d(self.conv4, 256, 3, activation='relu')
        self.pool5 = max_pool_2d(self.conv5, 3, strides=2)
        self.fc6 = fully_connected(self.pool5, 4096, activation='relu')
        self.fc7 = fully_connected(self.fc6, 4096, activation='relu')
        self.output = fully_connected(self.fc7, 1000, activation='softmax')
        return self.output

model = AlexNetModel()
output = model.build_model(images)
# access intermediate layers like self.conv5, self.fc7, etc.
```

2.利用Tensorboard查看一些可视化统计

3.使用tf.Print()和tf.Assert()打印变量

```
tf.Print(input, data, message=None,
         first_n=None, summarize=None, name=None)
```

一个简单例子:

```
def multilayer_perceptron(x):
    fc1 = layers.fully_connected(x, 256, activation_fn=tf.nn.relu)
    fc2 = layers.fully_connected(fc1, 256, activation_fn=tf.nn.relu)
    out = layers.fully_connected(fc2, 10, activation_fn=None)
    out = tf.Print(out, [tf.argmax(out, 1)],
                   'argmax(out) = ', summarize=20, first_n=7)
    return out
```

在前7次迭代中,它会输出预测的label,如下图所示:

```
I tensorflow/core/kernels/logging_ops.cc:79] argmax(out) = [6 6 6 4 4 6 4 4 6 6 4 0 6 4
I tensorflow/core/kernels/logging_ops.cc:79] argmax(out) = [6 6 0 0 3 6 4 3 6 6 3 4 4 4
I tensorflow/core/kernels/logging_ops.cc:79] argmax(out) = [3 4 0 6 6 6 0 7 3 0 6 7 3 6
I tensorflow/core/kernels/logging_ops.cc:79] argmax(out) = [6 1 0 0 0 3 3 7 0 8 1 2 0 9
I tensorflow/core/kernels/logging_ops.cc:79] argmax(out) = [6 0 0 9 0 4 9 9 0 8 2 7 3 9
I tensorflow/core/kernels/logging_ops.cc:79] argmax(out) = [6 0 1 1 9 0 8 3 0 9 9 0 2 6
I tensorflow/core/kernels/logging_ops.cc:79] argmax(out) = [3 6 9 8 3 9 1 0 1 1 9 3 2 3
[2016-06-03 00:11:08.661563] Epoch 00, Loss = 0.332199
```

缺点:函数过于简单,没法在满足一定条件下进行输出,或者隔多少次迭代才进行一次输出.

```
tf.Assert(condition, data, summarize=None, name=None)
```

这个函数在condition等于False的时候进行输出data的值,并且抛出异常,summarize决定输出多少变量值

使用方法:

```
def multilayer_perceptron(x):
    fc1 = layers.fully_connected(x, 256, activation_fn=tf.nn.relu, scope='fc1'
    fc2 = layers.fully_connected(fc1, 256, activation_fn=tf.nn.relu, scope='fc
    out = layers.fully_connected(fc2, 10, activation_fn=None, scope='out')
    # let's ensure that all the outputs in `out` are positive
    assert_op = tf.Assert(tf.reduce_all(out > 0), [out], name='assert_out_posi
    with tf.control_dependencies([assert_op]):
        out = tf.identity(out, name='out')
    return out
```

或者可以:

```
    # ... same as above ...
    out = tf.with_dependencies([assert_op], out)
    return out
```

或者把所有的Assert操作加入collection:

```
def multilayer_perceptron(x):
    fc1 = layers.fully_connected(x, 256, activation_fn=tf.nn.relu, scope='fc1'
    fc2 = layers.fully_connected(fc1, 256, activation_fn=tf.nn.relu, scope='fc
    out = layers.fully_connected(fc2, 10, activation_fn=None, scope='out')
    tf.add_to_collection('Asserts',
        tf.Assert(tf.reduce_all(out > 0), [out], name='assert_out_gt_0')
    )
    return out

# merge all assertion ops from the collection
assert_op = tf.group(*tf.get_collection('Asserts'))
```

```
... = session.run([train_op, assert_op], feed_dict={...})
```

官方也提供了一些assert相关的函数: https://www.tensorflow.org/versions/r0.12/api_docs/python/check_ops.html#asserts-and-boolean-checks

```
tf.assert_negative(x, data=None, summarize=None, name=None)
tf.assert_positive(x, data=None, summarize=None, name=None)
tf.assert_proper_iterable(values)
tf.assert_non_negative(x, data=None, summarize=None, name=None)
tf.assert_non_positive(x, data=None, summarize=None, name=None)
tf.assert_equal(x, y, data=None, summarize=None, name=None)
tf.assert_integer(x, data=None, summarize=None, name=None)
tf.assert_less(x, y, data=None, summarize=None, name=None)
tf.assert_less_equal(x, y, data=None, summarize=None, name=None)
tf.assert_rank(x, rank, data=None, summarize=None, name=None)
tf.assert_rank_at_least(x, rank, data=None, summarize=None, name=None)
tf.assert_type(tensor, tf_type)
tf.is_non_decreasing(x, name=None)
tf.is_numeric_tensor(tensor)
tf.is_strictly_increasing(x, name=None)
```

4.使用Python的debug工具: ipdb, pudb

5.利用tf.py_func()向图中插入自定义的打印代码

```
tf.py_func(func, inp, Tout, stateful=True, name=None)
```

简单例子:

```
def my_func(x):
    # x will be a numpy array with the contents of the placeholder below
    return np.sinh(x)

inp = tf.placeholder(tf.float32, [...])
y = py_func(my_func, [inp], [tf.float32])
```

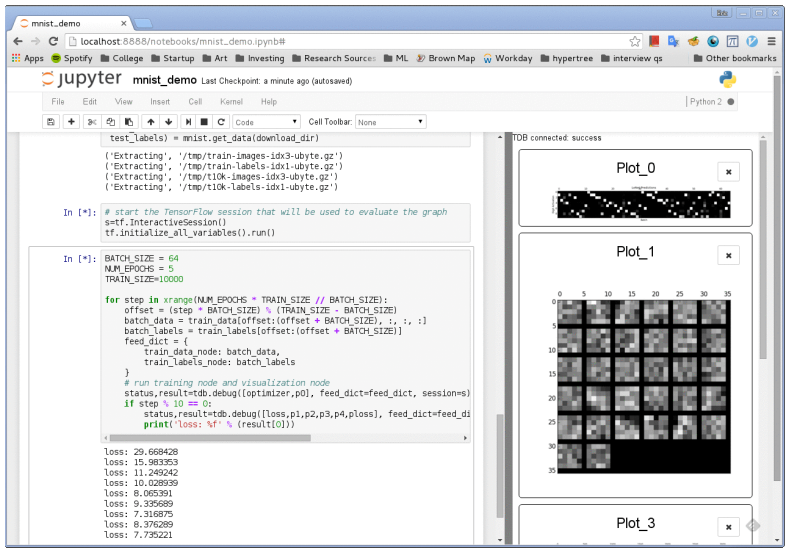
另一个例子:

```
def multilayer_perceptron(x):
    fc1 = layers.fully_connected(x, 256, activation_fn=tf.nn.relu, scope='fc1')
    fc2 = layers.fully_connected(fc1, 256, activation_fn=tf.nn.relu, scope='fc2')
    out = layers.fully_connected(fc2, 10, activation_fn=None, scope='out')

    def _debug_print_func(fc1_val, fc2_val):
        print 'FC1 : {}, FC2 : {}'.format(fc1_val.shape, fc2_val.shape)
        print 'min, max of FC2 = {}, {}'.format(fc2_val.min(), fc2_val.max())
        return False

    debug_print_op = tf.py_func(_debug_print_func, [fc1, fc2], [tf.bool])
    with tf.control_dependencies(debug_print_op):
        out = tf.identity(out, name='out')
    return out
```

一个第三方debug工具: https://github.com/ericjang/tdb



6.使用官方debug工具: tfdbg

黄世宇/Albert's Personal Page：http://huangshiyu.me/

分类: Tensorflow

好文要顶 关注我 收藏该文

Shiyu_Huang
关注 - 0
粉丝 - 1
+加关注

0 0

« 上一篇：sklearn数字图像识别(RBM+logistic)

» 下一篇：Tensorflow之快速加载MNIST数据集

posted @ 2017-04-17 10:52 Shiyu_Huang 阅读(27) 评论(0) 编辑 收藏

的发展(持续更新)(485)
3. Unity之屏幕画线(462)
4. Unity之读取本地图片(418)
5. 查看Tensorflow版本(363)
6. 京东阅读PDF导出(319)
7. Python之数组拼接，组合，连接(240)
8. VAE(Variational Autoencoder)的原理(227)
9. Linux之apt-get无sudo权限安装软件(190)
10. Tensorflow如何选择GPU(177)
11. pip依赖安装与记录(128)
12. 重尾分布(Heavy-tailed Distribution)(92)
13. 种类并查集(88)
14. Tensorflow之计算tensor平均值(78)
15. 受限玻尔兹曼机（RBM, Restricted Boltzmann machines）和深度信念网络（DBN, Deep Belief Networks）(74)
16. A Survey of Shape Feature Extraction Techniques中文翻译(73)
17. 新装的Ubuntu在Nvidia显卡上分辨率不对(58)
18. Macbook之设置Finder显示文件完整路径(57)
19. 2017机器学习相关会议时间(54)
20. Matplotlib之无GUI时的解决办法(51)

刷新评论 刷新页面 返回顶部

最新IT新闻:
· 魅蓝E2专用Hi Case趣闪保护壳：15种闪光灯样式
· 1299元起，魅族发布了魅蓝E2 ：闪光灯天线一体设计，颜值高吗？
· 癌症 为啥一发现就是晚期？无奈……
· 格力电器拟向股东派发108亿分红 董明珠税前分得近八千万
· Instagram用户增长率创新高 用户总数达7亿
» 更多新闻...

最新知识库文章:
· 唱吧DevOps的落地，微服务CI/CD的范本技术解读
· 程序员，如何从平庸走向理想？
· 我为什么鼓励工程师写blog
· 怎么轻松学习JavaScript
· 如何打好前端游击战
» 更多知识库文章...