Tensorflow的神经网络框架是模型化且可视化，所有对于局部结构的命名，设计与保存需要掌握。

对于变量的获取主要通过

Tf.get\_variable()和tf.Variable()

空间名有

Tf.variable\_scope()和tf.name\_scope().

Tf.name\_scope()对tf.get\_variable()不起作用，它只是起到命名模块空间的作用。

Tf.Variable()创建的变量不存在重复问题，系统自动在名字动手脚。

主要是tf.variable\_scope()和tf.get\_variable()的配合使用

在tf.variable\_scope()空间里命名的tf.get\_variable()变量为局部变量，当属性reuse=False，tf.get\_variable()只能创建新变量，

当属性reuse=True，tf.get\_variable()只能获取之前创建的变量

With tf.variable\_scope(“log”,resuse=True)

保存：  
saver = tf.train.Saver()

Saver.save(sess,”/path/to/model.ckpt”)

Saver.restore(sess,”/path/to/model.ckpt”)

直接加在持久画图

Saver=

tf.train.import\_meta­­\_graph(“/path/to/model.ckpt/model.ckpt.meta”)

saver.restore(sess,”/path/to/model.ckpt”)

获取保存的变量，如果被加载的名字与加载变量的名不同，需要知道被加载变量的名

Saver = tf.train.Saver({ “v1”:v1,”v2”:v2 })

Tensorflow 通过dictionary（字典）将模型保存时的变量和需要加载的变量联系起来

目的：方便使用变量的滑动平均值

p114

Ema = tf.train.ExponentialMovingAverage(0.99)

Saver = tf.train.Saver(ema.variables\_to\_restore())

Saver.restore(sess,”/path/tomodel.ckpt”)

将变量及其取值通过常量的方式保存

Graph\_def = tf.get\_default\_graph().as\_graph\_def()

Output\_graph\_def= graph\_util.convert\_veriables\_to\_constants(sess,graph\_def, [‘add’])

With tf.gfile.Gfile(“/path/to/combined\_model.pb”,”wb”) as f:

f.writer(output\_graph\_def.SerializeToString()

读取保存的模型文件

With gfile.FastGFile(“/path/to/combined\_model.pb”,”rb”) as f:

graph\_def = tf.Graphdef()

graph\_def.ParseFromString(f.read())