该文档主要说明关于深度学习模型训练保存与调用模型。

参考网址：<https://blog.csdn.net/lwplwf/article/details/62419087>

# 模型保存

模型保存，先要创建一个Saver对象：如

saver=tf.train.Saver()

在创建这个Saver对象的时候，有一个参数经常会用到，max\_to\_keep 参数，这个是用来设置保存模型的个数，默认为5，即 max\_to\_keep=5，保存最近的5个模型。如果想每训练一代（epoch)就想保存一次模型，则可以将 max\_to\_keep设置为None或者0，但是这样做除了多占用硬盘，并没有实际多大的用处，因此不推荐，如：

saver=tf.train.Saver(max\_to\_keep=0)

如果你只想保存最后一代的模型，则只需要将max\_to\_keep设置为1即可，即

saver=tf.train.Saver(max\_to\_keep=1)

创建完saver对象后，就可以保存训练好的模型了，如：

saver.save(sess,‘ckpt/mnist.ckpt',global\_step=step)

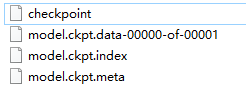
第二个参数设定保存的路径和名字，第三个参数将训练的次数作为后缀加入到模型名字中。

saver.save(sess, 'my-model', global\_step=0) ==> filename: 'my-model-0'

...

saver.save(sess, 'my-model', global\_step=1000) ==> filename: 'my-model-1000'





* checkpoint二进制文件，把变量名映射到对应的tensor值，简单理解就是权重等参数被保存到 .chkp.data 文件中，以字典的形式。文件保存了一个目录下所有的模型文件列表，这个文件是tf.train.Saver类自动生成且自动维护的。在 checkpoint文件中维护了由一个tf.train.Saver类持久化的所有TensorFlow模型文件的文件名。当某个保存的TensorFlow模型文件被删除时，这个模型所对应的文件名也会从checkpoint文件中删除。checkpoint中内容的格式为CheckpointState Protocol Buffer.
* model.ckpt.meta文件保存了TensorFlow计算图的结构，可以理解为神经网络的网络结构   
  TensorFlow通过元图（MetaGraph）来记录计算图中节点的信息以及运行计算图中节点所需要的元数据。TensorFlow中元图是由MetaGraphDef Protocol Buffer定义的。MetaGraphDef 中的内容构成了TensorFlow持久化时的第一个文件。保存MetaGraphDef 信息的文件默认以.meta为后缀名，文件model.ckpt.meta中存储的就是元图数据。
* model.ckpt文件保存了TensorFlow程序中每一个变量的取值，这个文件是通过SSTable格式存储的，可以大致理解为就是一个（key，value）列表。model.ckpt文件中列表的第一行描述了文件的元信息，比如在这个文件中存储的变量列表。列表剩下的每一行保存了一个变量的片段，变量片段的信息是通过SavedSlice Protocol Buffer定义的。SavedSlice类型中保存了变量的名称、当前片段的信息以及变量取值。TensorFlow提供了tf.train.NewCheckpointReader类来查看model.ckpt文件中保存的变量信息。

# 2.模型加载

用saver.restore()方法恢复变量：

saver.restore(sess,'ckpt.model\_checkpoint\_path')

sess：表示当前会话，之前保存的结果将被加载入这个会话；

ckpt.model\_checkpoint\_path：表示模型存储的位置，不需要提供模型的名字，它会去查看checkpoint文件。

