1. tensorflow2中模型的保存

（一）概述

Tensorflow2中提供了 tf.train,Checkpoint这个变量保存与恢复类，可以使用save()和restore()方法将 Tensorflow中所有包含CheckPointable State的对象进行保存和恢复。

如，优化器tf.keras.optimizer，变量tf.Variable，层tf.keras.Layer和模型tf.keras.Model实例都可以被保存。

（二）Checkpoint对象的保存

要保存CheckPoint对象，首先需要声明一个Checkpoint，tf.train.Checkpoint()接受的初始化参数是一系列的键值对，键名任意取，值为需要保存的对象。这里的键名非常重要，在恢复模型时，需要使用同样的键名。

checkpoint = tf.train.Checkpoint(model=model)

在模型训练完成后，对模型进行保存，使用save函数保存，参数为保存的目录名。

checkpoint.save(path)

如：

checkpoint.save(‘./save/model.ckpt’)

save目录下就会生成checkpoint，model.ckpt-1.index，model.ckpt-1.data-00000-of-00001的三个文件。

save()方法能够运行多次，每次运行后都能得到一个.index文件和一个.data文件，文件名的序号依次增加。

1. Checkpoint对象的恢复

（一）恢复模型

当我们需要为某个模型加载之前已经训练好的参数时，需要再次实例化一个checkpoint，键名需要和保存的键名一致，再调用restore()方法恢复模型参数，restore()方法的参数为之前保存的文件的目录+编号，如”./save/model.ckpt-1”，跟之前保存的目录名相比添加了编号1，即第一个保存的文件。

如：

#待恢复参数的同一模型

model\_to\_be\_restored = MyModel()

#新建checkpoint

checkpoint = tf.train.Checkpoint(model=model\_to\_be\_restored)

checkpoint.restore(path\_index)

（二）恢复最近的一个Checkpoint

使用tf.train.latest\_checkpoint(path)返回目录下最近一次checkpoint的文件名。

如：

checkpoint.restore(tf.train.latest\_checkpoint(“./save”))

三．保存与恢复变量代码模板

先实例化checkpoint，继而训练，保存模型，再在其他模型中恢复保存的参数。

train.py

model = MyModel()

checkpoint = tf.train.CheckPoint((myModel=model)

#训练模型

checkpoint.save(“./save/model.ckpt”)

test.py

model = MyModel()

checkpoint = tf.train.CheckPoint(myModel=model)

checkpoint.restore(tf.train.latest\_checkpoint(“./save”))

#模型使用

1. 管理checkpoint

当checkpoint保存的数量过多时，我们可能只想保存最后几个checkpoint，或者希望使用其他的编号方式，这些都要通过checkpointManager来实现。

checkpoint = tf.train.Checkpoint(model=model)

manager = tf.train.CheckpointManager(checkpoint, directory=”./save”, checkpoint\_name=’model.ckpt’, max\_to\_keep=k)

directory 参数为文件保存的路径， checkpoint\_name 为文件名前缀（不提供则默认为

ckpt ）， max\_to\_keep 为保留的Checkpoint数目。

然后使用manager,save()保存即可，如需要修改编号的形式，则在保存时添加checkpoint\_number参数，如：将编号设置为100

manager.save(checkpoint\_number=100)