深度学习方法与实践实验二

2020年3月5日

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 董佩杰 | 学 号 | 2016012963 |
| 年级 | 2020级 | 指导老师 | 牛新 |

# 一、用Tensorflow2.1拟合余弦函数

**1. 题目说明：**

假设有函数, 其中a为学号倒数第5和第4位，b为学号最后两位（例如学号19020139对应)。首先从此函数中以相同步长（点与点之间在x轴上距离相同），在范围内，采样出2000个点，然后利用采样的2000个点作为特征点进行三次函数拟合(三次函数形式为 , 其中wi为可训练的权值，b为可训练的偏置值，x和y为输入的训练数据 ) 。

1. **具体要求：**
2. **要求使用TF2.x**
3. **用model.fit和自定义循环两种训练方法实现三次函数拟合的全部流程**
4. **分别使用回调函数和model.save模式保存拟合的模型**
5. 针对两种模型存储方式分别编写模型恢复程序分别，并同时绘制图像
6. 记录和打印保存前和恢复后的loss,并查看是否一致。

# 完成情况

**1.模型构建**

下图主要是数据构建以及模型定义：

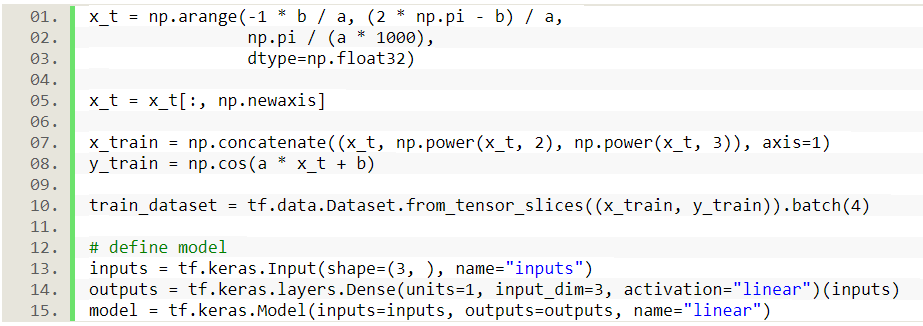


图 1 模型构建

**2.训练流程**

下图主要是模型的自定义训练以及测试流程：

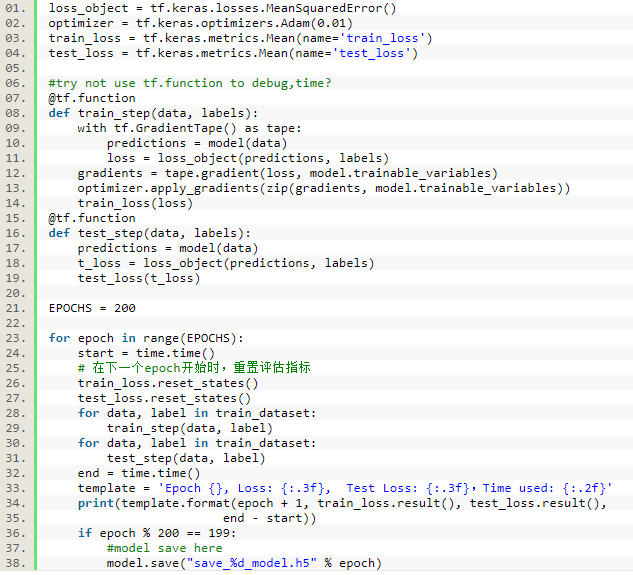


图 2 使用自定义循环

使用fit进行训练：

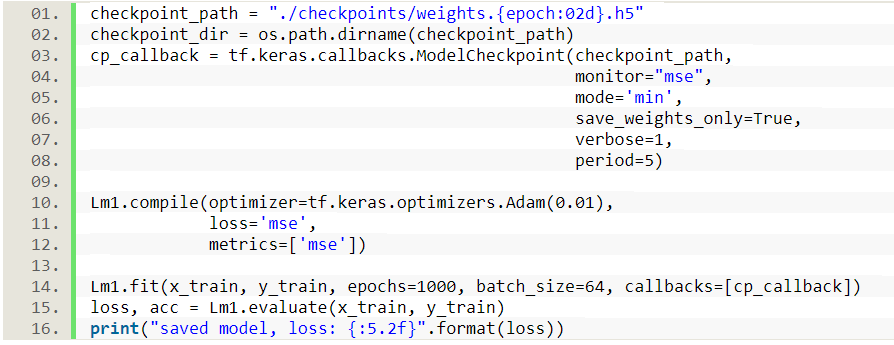
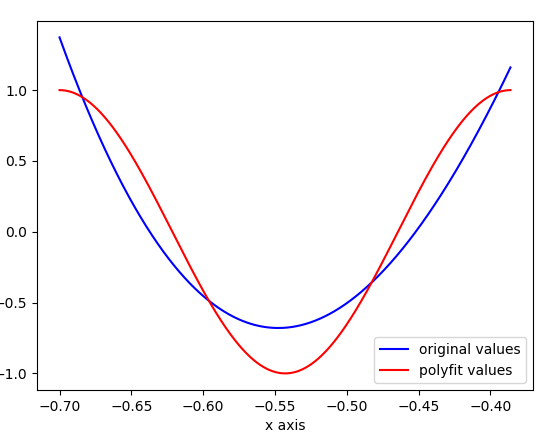


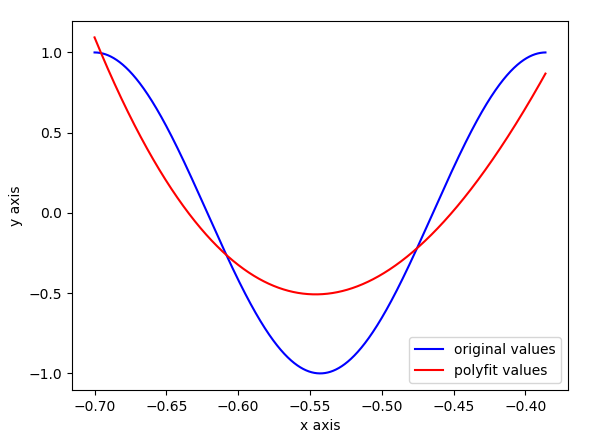
图 3 使用fit训练截图

**3.训练结果**

以下结果是用自定义循环方式，epoch=10000的结果，最终loss=0.05：

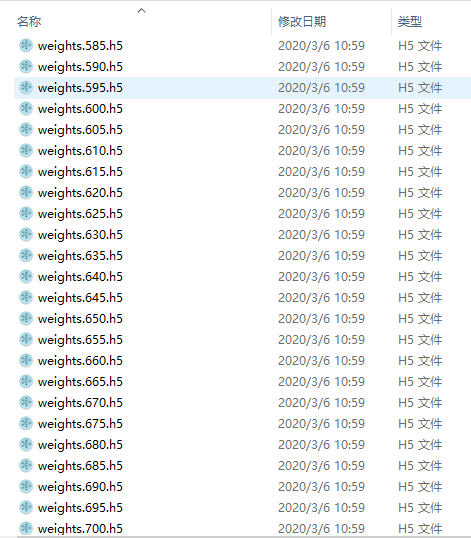


下图结果是使用fit方法进行拟合的结果，epoch=1000, 最终loss=0.09:

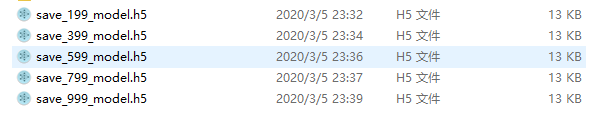


****4.使用回调函数和model.save模式保存拟合的模型****

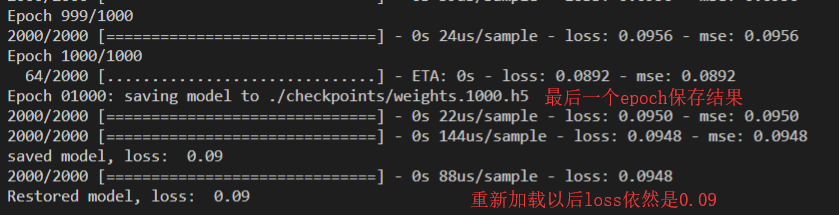
通过回调函数保存的结果如下：



通过model.save保存结果如下

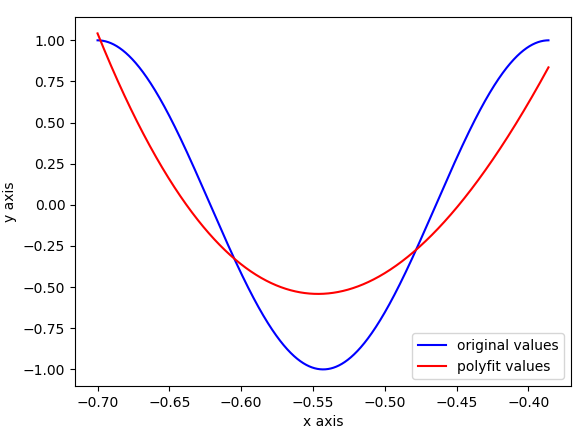


**5.记录和打印保存前和恢复后的loss,并查看是否一致。**

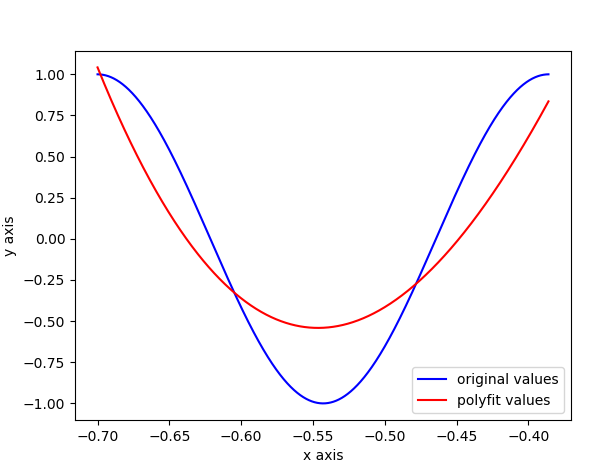


**6.绘制图像结果**

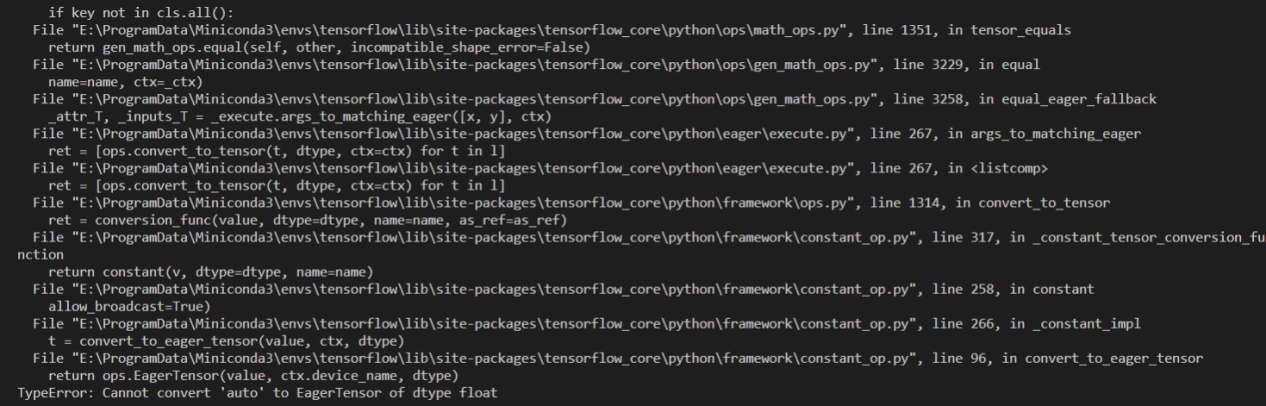
最后一个epoch结果：



Restore以后结果：



尝试并报告loss\_object = tf.keras.losses.MeanSquaredError()改为 tf.keras.losses.MeanSquaredError（去括号）的效果：

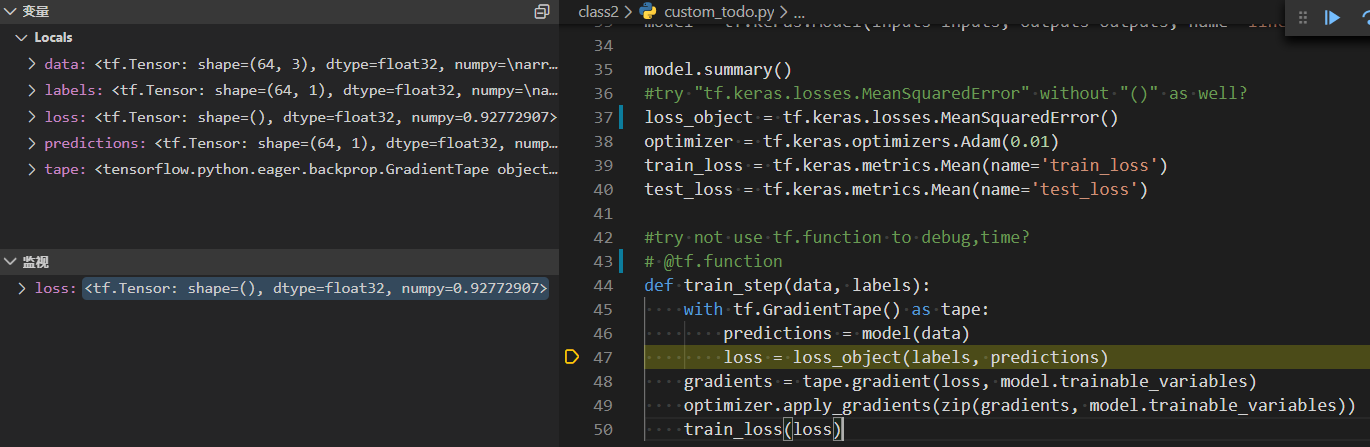


以上报错的原因，如果去掉括号，生成的loss\_object就是一个类，而不是一个实例化的对象。

**7.tf.function功能**

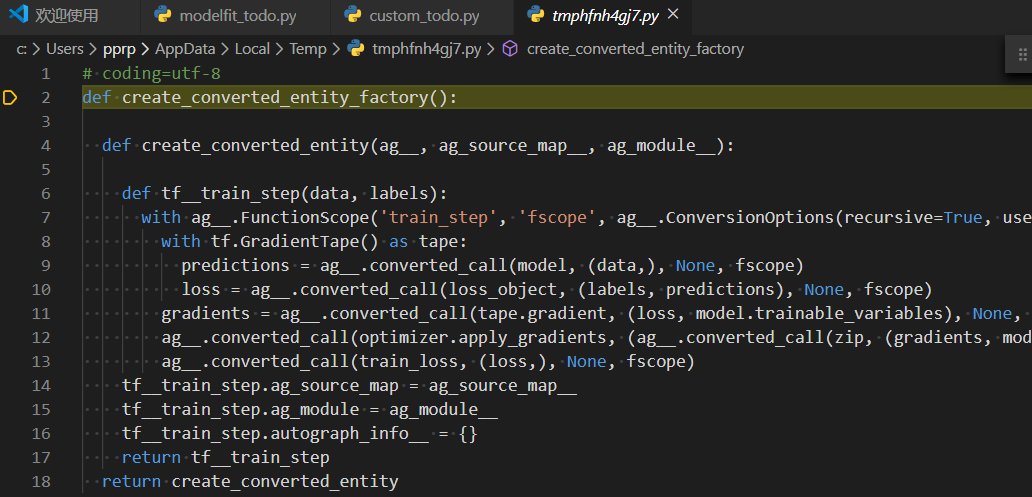
自定义循环添加[@tf](https://github.com/tf" \o "@tf).function和不添加的效果，可调试否？运行时间？

不添加效果如下：



可以进行debug并查看变量内容。

添加效果如下：



如果开启tf.function，那么debug会进入一个新的临时文件，还发现只能进入这个临时文件一次，之后就无法进入那个临时文件。

速度上，开启tf.function速度要快很多，同样的batch size=64:

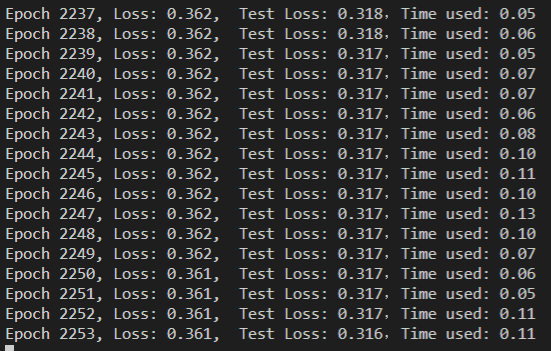


图 4 开启tf.function

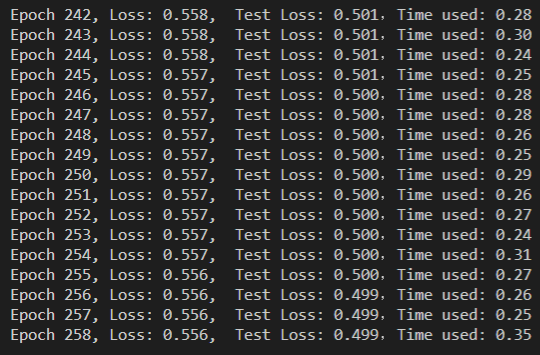


图 5 不开启tf.function

一开始不理解，既然有了动态图，为何还要使用静态图。现在看来，应该是现在关闭tf.function的时候进行debug，利用动态图灵活的特点进行调试。然后在稳定以后，可以开启静态图，加速训练过程。

**8.发现**

相同的optimizer和loss的情况下，使用两种方法相同迭代次数结果不一样。Fit方法能够更快的收敛，使用fit方法只需要用1000epoch就能达到与自定义循环10000epoch差不多的结果。