

[blog.csdn.net /weixin_41466575/article/details/113275000](https://blog.csdn.net/weixin_41466575/article/details/113275000)

Google colab使用之手把手教学_liuyishou-CSDN博客

4-5 minutes

原创

AI bro 2021-01-27 23:24:44



1200



收藏 12

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

目录

0 前言

1 极速开始

2 “持久化”使用方式

3 一些重要的使用经验

- 1、使用tensorflow第一版本
- 2、跑gpu
- 3、一个测试速度的官方Demo

0 前言

相信各位看到这篇文章，是因为自己电脑跑神经网络代码太慢了。这里我谈一下我个人经历。前几天我要跑一个tensorflow的项目——官方提供的一个28*28*3的手写数字生成项目。由于我没有配gpu，一个很简单的项目跑了很久都没成功。我开始了探索之路。

- 刚开始我尝试在自己电脑上配置gpu环境，网上配置gpu教程参差不齐，我试了很久也无法成功
- 经同学推荐入手了baidu AI Studio。发现不支持tensorflow框架，百度为了推广自己开源的paddle（和tensor框架类似），只允许在平台上跑paddle代码
- Google colab救了我。Google colab可以使用gpu跑，总体来说要比我电脑快了将近20倍，写此文章作为分享

以上介绍个人经历，为了给各位跳坑。接下来尽量干货分享。

使用Google colab要求：

- 能“翻墙”，且翻墙速度够快（才能给你一个更优的体验）
- 拥有Google邮箱账号（能翻墙的话可以自己注册）
- 如果以上两个条件不能满足，以下内容就没必要看了

1 极速开始

<https://colab.research.google.com/notebooks/welcome.ipynb> 点击进入页面如下：



点击文件->新建笔记本，如下图：



会要求登录Google账号，登录后会出现如下页面。在如下单元格内编辑代码，按shift+enter键就能正常运行。当然这里也是可以导入tensorflow和keras等神经网络常用库的。



这里科普一下:

- 这种急速开始的方式一旦关闭了这个页面，所有代码以及运行结果全部消失。这是个不持久的应用方法。
- Google colab其实是给你分配了一个linux云服务器（如下图所示）。当然linux服务器是随机分配的，一旦你离开页面结果自然无法保存。
- 所以说这种极速开始是没有意义的，以下给你介绍更“康”的使用方式。



2 “持久化”使用方式

1. `from google.colab import drive`
2. `drive.mount('/content/gdrive')`
3. `import os`
4. `os.chdir("/content/gdrive/MyDrive/Colab Notebooks")`

加入如上代码。如下图显示，下图显示要输入认证码，点击url链接，获取认证码，然后shift+enter按下。



google云盘：只要有google邮箱账号，就会有google云盘，就是和百度云盘类似的东西。这里解释下，以下内容要用。

上面代码解释：

- `drive.mount('/content/gdrive')`，表示将自己整个google云盘挂载到google随机给我们分配的linux服务器上
- 这样做的好处就是我们编写的代码都会存在google云盘，下次想用数据也好，源代码也好。只要挂载上就能使用上次或者上上次的代码或数据，这不就实现了持久化了（舒服了舒服了）。
- 你可以比对以下两图。 <https://drive.google.com/drive/my-drive> （这个是云盘链接）

The screenshot shows the Google Colab interface. On the left, the file explorer shows a directory structure: `gdrive` (parent), `MyDrive` (child), and `Colab Notebooks` (child of `MyDrive`). Below `Colab Notebooks` is `sample_data`. A red arrow points from the text "运行完那四行代码后就成功将整个google云盘挂载到了随机分配的linux服务器上" to the `gdrive` directory in the file explorer. Another red arrow points from the same text to the first code cell. The code cell contains the following code:

```
from google.colab import drive
drive.mount('/content/gdrive')
import os
os.chdir("/content/gdrive/MyDrive/Colab Notebooks")
```

Below the code cell, the output shows "Mounted at /content/gdrive". The second code cell contains:

```
[2] import tensorflow as tf
import keras as keras
print (tf.__version__)
2.4.0
```

At the bottom, there is a section for "云端硬盘" (Cloud Drive) with a search bar and a list of "我的云端硬盘" (My Cloud Drives). A red arrow points from the text "仔细观察这两个图片，Colab Notebooks是不是重复出现" to the `Colab Notebooks` directory in the file explorer and the `Colab Notebooks` directory in the "我的云端硬盘" section.

- 不要忘记还有一行代码。os.chdir("/content/gdrive/MyDrive/Colab Notebooks")
- 这句代码表示更改运行目录。emmmmmmmmm, 有点迷惑是不是? 看下图

The screenshot shows the Google Colab interface. On the left, the file explorer shows a directory structure: `gdrive` (parent), `MyDrive` (child), and `Colab Notebooks` (child of `MyDrive`). Below `Colab Notebooks` are `MNIST_data`, `resnet_model_20210127...`, `train`, and `transfer_model_20210127...`. A red arrow points from the text "我在train/cat/cat.0.jpg上有一张图片" to the `train` directory. Another red arrow points from the text "改了成了MNIST_data文件夹下的当前目录" to the `MNIST_data` directory. The code cell contains the following code:

```
[5] import os
os.chdir("/content/gdrive/MyDrive/Colab Notebooks/MNIST_data")
```

Below the code cell, the output shows "Directory already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount('/content/gdrive/MyDrive/Colab Notebooks/MNIST_data')". The sixth code cell contains:

```
[6] import os
os.chdir("/content/gdrive/MyDrive/Colab Notebooks/MNIST_data")
```

Below the code cell, the output shows "Directory already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount('/content/gdrive/MyDrive/Colab Notebooks/MNIST_data')". The seventh code cell contains:

```
import cv2 as cv
img = cv.imread("./train/cat/cat.0.jpg")
print (img.shape)
```

Below the code cell, the output shows "AttributeError: 'module' object has no attribute 'imread'". A red arrow points from the text "所以这个路径就访问不到" to the `./train/cat/cat.0.jpg` path in the code cell.

- 不过好像这行代码没什么太大用，愿意加就加上吧

3 一些重要的使用经验

1. 打开了colab就默认加载tensorflow 2.x的版本, 如果你想用1版本的tensorflow。按如下步骤:

- 先点击重新启动代码执行程序, 意味着清空变量

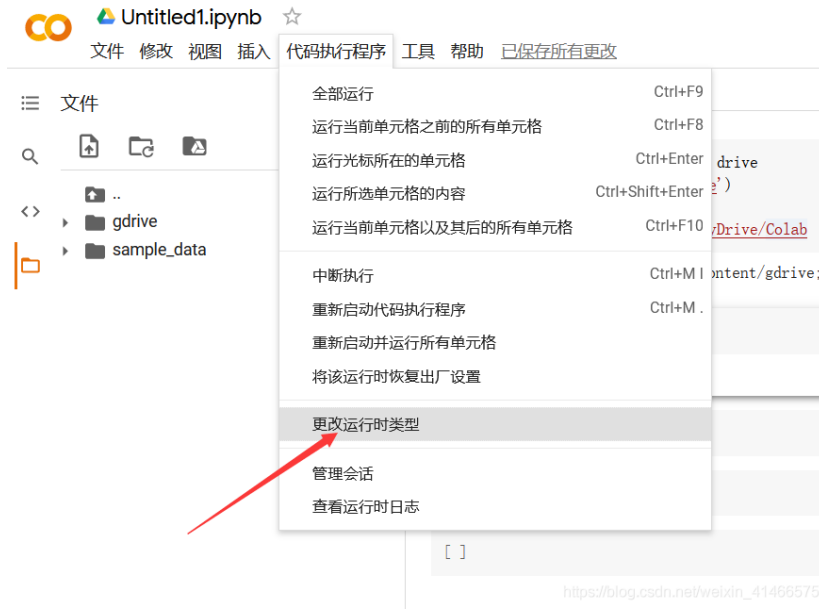


- 运行 `%tensorflow_version 1.x`, 出现 TensorFlow 1.x selected. 即成功切换到 tensorflow 1版本。一定得先清空变量, 否则运行`%tensorflow_version 1.x`, 就没有用



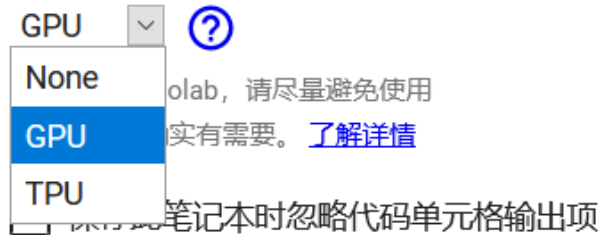
2、使用gpu运行代码。按如下步骤:

- 如下图所示操作即可。运行python代码就会默认使用gpu来运行。下面有个测试速度的官方demo, 可以看看。



笔记本设置

硬件加速器



取消

保存

- 最好使用tpu, tpu比gpu速度快。详见 [CPU, GPU, TPU的区别](#)

3、一个测试速度的官方Demo

```
1. %tensorflow_version 1.x
2. import tensorflow as tf
3. import timeit
4.
5. device_name = tf.test.gpu_device_name()
6. if device_name != '/device:GPU:0':
7. print(
```

```
8. '\n\nThis error most likely means that this notebook is not '
9. 'configured to use a GPU. Change this in Notebook Settings via
   the '
10. 'command palette (cmd/ctrl-shift-P) or the Edit menu.\n\n')
11. raise SystemError('GPU device not found')
12.
13. def cpu():
14.     with tf.device('/cpu:0'):
15.         random_image_cpu = tf.random.normal((100, 100, 100, 3))
16.         net_cpu = tf.keras.layers.Conv2D(32, 7)(random_image_cpu)
17.         return tf.math.reduce_sum(net_cpu)
18.
19. def gpu():
20.     with tf.device('/device:GPU:0'):
21.         random_image_gpu = tf.random.normal((100, 100, 100, 3))
22.         net_gpu = tf.keras.layers.Conv2D(32, 7)(random_image_gpu)
23.         return tf.math.reduce_sum(net_gpu)
24.
25.
26. cpu()
27. gpu()
28.
29.
30. print('Time (s) to convolve 32x7x7x3 filter over random
       100x100x100x3 images ')
31. '(batch x height x width x channel). Sum of ten runs.')
32. print('CPU (s):')
```

```
33. cpu_time = timeit.timeit('cpu()', number=10, setup="from __main__
    import cpu")

34. print(cpu_time)

35. print('GPU (s):')

36. gpu_time = timeit.timeit('gpu()', number=10, setup="from __main__
    import gpu")

37. print(gpu_time)

38. print('GPU speedup over CPU: {}'.format(int(cpu_time/gpu_time)))
```

结果如下:

```
> TensorFlow 1.x selected.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/python/ops/resource_variable_ops.py:1630: calling BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_variable_ops)
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
Time (s) to execute 70x70x70 filter conv kernel 100x100x100x3 images (batch = kernel = width = height) 500 of size 1000
```

如有疑问，以下评论区留言。力所能及，必答之。