之前主要在linux上做deep leaning的train/inference。这次看下windows下怎么玩（我的笔记本电脑带GTX1650显卡）。在这个过程中遇到了好几个问题，转帖的这个文章和我配置相似，那些点完全也是我遇到的点。特贴下，存个档，感谢原作者分享。特别这一段：

from tensorflow.compat.v1 import ConfigProto

from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()

config.gpu\_options.allow\_growth = True

session = InteractiveSession(config=config)

实际测试我的segmentation的inference没问题。Train还是报内存不够，后把batch size改小了。能跑了，但是还是发现：[Tensorflow CUDA - CUPTI error: CUPTI could not be loaded or symbol could not be found](https://stackoverflow.com/questions/56860180/tensorflow-cuda-cupti-error-cupti-could-not-be-loaded-or-symbol-could-not-be)。

加个PATH:

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1\extras\CUPTI\lib64

还是有问题：

Unavailable: CUPTI error: CUPTI\_ERROR\_INSUFFICIENT\_PRIVILEGES

权限的问题：

[https://developer.nvidia.com/nvidia-development-tools-solutions-ERR\_NVGPUCTRPERM-permission-issue-performance-counters#SolnTag](https://developer.nvidia.com/nvidia-development-tools-solutions-ERR_NVGPUCTRPERM-permission-issue-performance-counters" \l "SolnTag)

另外：

os.environ['CUDA\_VISIBLE\_DEVICES'] = ‘0’

batch\_size = 1

好吧，终于解决了！半天时间没了。有空看下，Jetson nano的inference性能。

<https://blog.csdn.net/weixin_45023983/article/details/99178625?depth_1-utm_source=distribute.pc_relevant.none-task&utm_source=distribute.pc_relevant.none-task>

Win10+GTX1650显卡下安装Tensorflow-gpu1.14的踩坑过程及训练目标检测模型

原创JC\_BBX 最后发布于2019-08-11 19:18:56 阅读数 5173 收藏

展开

Win10+GTX1650显卡下安装Tensorflow-gpu1.14的踩坑过程及训练目标检测模型

作为一个刚接触深度学习的小白，因目标检测的任务需求，在网上查阅了大量前辈写的相关blog，学到了很多东西，最后自己搭建完环境训练了模型。因为搭配环境时踩了很多坑，我将内容整理一下，方便必要时再次查看。

安装tensorflow-gpu前python、cuda、cudnn、vs的版本需要一一对应，否则可能最后会报错。我装的比较新：

anaconda3

python3.7

cuda10.1

cudnn7.6

vs2019

tensorflow-gpu 1.14

安装tensorflow object detection API

1.去anaconda官网，下载完后自带python3.73

2.tensorflow分cpu版和gpu版，cup版相当容易，命令行pip install tensorflow即可，安装的是tensorflow1.14，gpu版相当麻烦，先跳过。

3.安装tensorflow object detection API，先去Github上下载https://github.com/tensorflow/models

然后安装配置Protobuf，下载地址https://github.com/protocolbuffers/protobuf/releases

具体的操作参考TensorFlow学习（四）之基于win10实现官方Object Detection API 步骤及出现的坑bug解决办法博主已经写的非常详细了，感谢博主，让我受益匪浅。

4.然后测试一下官方demo，在research\object\_detection文件夹下打开jupyter notebook，因为我的版本是tensorflow1.14，我会把下面的注释掉或者4前面加个1。

狗、人、风筝都出现了对应的框，测试成功。

搭建tensorflow-gpu环境

因为需要自己训练模型，cpu版的tensorflow可以训练，但是用它训练模型速度会慢很多，需要gpu加速。于是我卸载了cpu版的tensorflow。准备安装tensorflow-gpu。因为我电脑已经安装了最新版visual studio2019，和python3.73，因此只需要下载对应的cuda和cudnn。按照网上教程，我分别安装了CUDA10.1、10.0、9.0等多个版本和对应的cudnn，在命令行pip install tensorflow-gpu下载了安装了1.14版本，每次import tensorflow时都找不到指定模块，找了很多教程降版本也好，怎么怎么操作都无法解决。我的显卡型号是GTX1650。看了下显卡支持cuda的型号，发现支持的是CUDA10.2。CUDA10.2官网还没出。

然后去官网查看https://developer.nvidia.com/cuda-gpus貌似我显卡不在上面？

这让我这个新手一度怀疑该型号显卡不支持CUDA。但看到了一位前辈使用该显卡配置成功，又给我提供了相当关键的信息https://github.com/fo40225/tensorflow-windows-wheel

对照表格，我找到了相匹配的版本。

1.CUDA和cuDNN安装

点击CUDA下载10.1版本，我这好像不管下载在哪最后都在C:\Program Files\NVIDIA GPU Computing Toolkit文件夹下。安装好后我去添加环境变量，貌似已经自动添加好了。

点击cuDnn下载CUDA10.1对应的版本，为7.60版本。这个是深度神经网络的GPU加速库。下载需要注册，可能稍微费时一点。

解压后把cuDNN中bin，include，lib文件夹下的文件对应的复制到C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1中相对应文件夹即可。

2.tensorflow-gpu安装

找到之前Github上的那个查版本对应的项目，下载**tensorflow-windows-wheel/1.14.0/py37/GPU/**下的whl文件，有sse2和avx2的，对照表格两者对应的compute capability不同，我的GPU显示计算能力7.5，可能应该选avx2吧。但我好奇的下载了sse2的。我也不知道是否有必然的联系…反正最后没什么错。

下载完成后在对应下载的文件夹下打开命令行

pip install tensorflow\_gpu-1.14.0-cp37-cp37m-win\_amd64.whl

之后输入import tensorflow as tf好像不再报没找到模块的错误了，太好了。在python环境中输入

import tensorflow as tf

tf.Session()

会显示一些关于我的GPU信息，算是成功了。

3.再次测试官方demo

这次安装完成了tensorflow-gpu1.14，基本上那些blog都是没有问题的，但是不知道为什么我在测试时出现了bug

调用卷积算法失败，可能是因为cuDNN没有初始化？这个bug不太好找，出现这类问题的人不多，有人说可能是tensorflow的版本过高，他们降到1.8版本就解决了这个问题，但python3.7不支持tensorflow1.8。一旦降版本可能要从头再来。后来在https://github.com/tensorflow/tensorflow/issues/24828找到了一个解决方法。在代码中前加上

from tensorflow.compat.v1 import ConfigProto

from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()

config.gpu\_options.allow\_growth = True

session = InteractiveSession(config=config)

这段意思应该是给GPU分配动态内存

训练模型

训练模性的步骤一般是：

1.准备大量所需的图像，并分成训练集和测试集，比例我选的是7：3

2.用LabelImg软件标注图像中所需识别的目标，生成xml格式文件。该软件开源在Github上

3.用脚本将xml格式文件转化成csv格式

4.再用python脚本将csv转化成tensorflow可以读取的record格式

5.设置配置文件，在object\_detection\samples\configs下有很多种类可供选择，但需修改部分代码

6.创建pbtxt格式的文本文件，需要写上检测的类别

7.启动object\_detection文件夹下自带的model\_main.py开始训练

8.运行自带的python export\_inference\_graph.py将model.ckpt文件，生成模型

9.利用该模型进行训练

这方面主要参考了这篇博文Tensorflow object detection API 搭建属于自己的物体识别模型（2）——训练并使用自己的模型这位博主非常用心，讲的非常详细。甚至出了教学视频，使我很快掌握了模型训练的方法。

稍有不同的是我上文中提到我出现了无法调用卷积的bug，所以我在model\_main.py中加入了GPU内存分配的代码。

另外，由于项目的局限性，数据集只有收集到数百张，若是从头开始训练，数据集较少，大概率无法收敛。我下载了一些训练好的网络做fineturn。由于最近刚看完SSD算法的论文，对它感情比较深，于是在Tensorflow detection model zoo

下载了ssd\_inception\_v2\_coco模型。用ssd算法在inception v2网络训练coco数据集生成的模型。

config格式的配置文件在samples文件夹下找到对应的模型。需要修改下几个地方：

将类别改为具体定义的种类

fineturn部分地址修改为刚下载模型的model.ckpt文件所在地

修改地址到对应的record文件和pbtxt文本文件中

然后可以开始按照上面的步骤开始训练了。以下是训练界面，显示训练步数，损失函数等。每隔一定步数会保存一次model.ckpt

打开可视化界面，下图是step=13K时部分截取界面。

最后进行预测时，效果还是不错的，几乎可以正确识别大部分异常处，关于项目的预测结果图就不放了，放一张使用以上同样方法训练出的模型预测图，数据集是网上直接下载来的。

可以看到框的位置和类别判断的较为准确，但是后面的人被车门挡住后却无法识别出来 。从总体来看效果还是不错的。

以上是我这个初学者所遇到的坑，感谢之前的博主们所做的贡献，让我可以事半功倍。第一次写blog，如有错误，请多多担待，谢谢。

————————————————

版权声明：本文为CSDN博主「JC\_BBX」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/weixin\_45023983/article/details/99178625