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。
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

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。
