

GPU资源使用指北

相较于需求，服务器的资源这段时间一直处在比较紧缺的状态，

由于TensorFlow等框架的特性（默认占用全部资源），经常出现GPU核心占用率不高但显存已满的情况，

既浪费了资源，也影响了使用，希望大家能更好地利用好资源

1. 选择使用的GPU核心「可选」
2. 限制可用的显存

```
In [ ]: # 规定程序使用的核心

import os

os.environ['CUDA_VISIBLE_DEVICES'] = "2,3"
```

可以先用nvidia-smi命令查看GPU使用情况，选择占用低的核心，

如下面这种情况，可以选择2，3核心

```
In [1]: !nvidia-smi
```

Wed Oct 3 14:44:00 2018

NVIDIA-SMI 384.125				Driver Version: 384.125			
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC		
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	
0	Tesla K80	Off	00000000:04:00.0	Off		0	
N/A	67C	P0	138W / 149W	10953MiB / 11439MiB	88%	Default	
1	Tesla K80	Off	00000000:05:00.0	Off		0	
N/A	49C	P0	88W / 149W	10953MiB / 11439MiB	71%	Default	
2	Tesla K80	Off	00000000:84:00.0	Off		0	
N/A	51C	P0	67W / 149W	4432MiB / 11439MiB	10%	Default	
3	Tesla K80	Off	00000000:85:00.0	Off		0	
N/A	39C	P0	71W / 149W	116MiB / 11439MiB	0%	Default	
Processes:							
GPU	PID	Type	Process name			GPU Memory Usage	
0	31294	C	/home/amax/anaconda3/bin/python3.6			10940MiB	
1	10421	C	/home/amax/anaconda3/bin/python3.6			10940MiB	
2	23586	C	python			4417MiB	
3	23586	C	python			99MiB	

```
In [ ]: # 限制显存占用
```

```
config = tf.ConfigProto()

# 开启TensorFlow显存占用按需求增长
# ! 请注意，程序运行完之后TensorFlow仍然不会释放显存，请手动关闭Kernel

config.gpu_options.allow_growth = True

# 设定每块GPU上使用的显存上限（0.4即40%）
config.gpu_options.per_process_gpu_memory_fraction = 0.4

tf.enable_eager_execution(config=config)
```

```
In [ ]: # 也可以在实例化Session时传入：
```

```
config = tf.ConfigProto()

config.gpu_options.allow_growth = True

config.gpu_options.per_process_gpu_memory_fraction = 0.4

with tf.Session(config=config) as sess:
    output = sess.run(...)
```

```
In [ ]:
```

保持程序稳定运行

(针对使用终端的情况)

之前好几次有同学反馈说程序被莫名Kill了，

这个真不是我们的锅，我们也很无奈

由于Linux的后台机制，当用户注销或者网络断开后，终端后收到挂断信号（SIGHUP），并向子进程广播SIGHUP信号，

子进程收到SIGHUP信号便会被关闭。

所以：请不要在终端直接使用 `python [file]` 运行程序

直接以 `python [file]` 运行程序时，一但设备休眠或者网络出现波动，服务器上的程序就会立刻被Kill

更何况，学校的锐捷认证是需要心跳包验证并且会定期下线的，所以，某种意义上，不存在可以保证程序稳定运行的网络环境

保证程序在终端下线后继续运行方法很多，推荐使用Tmux

服务器已经安装了Tmux，

- 直接运行tmux就可以新建session
- 按 `Ctrl + B`（Mac 为 `Control + B`），可以进入Tmux的命令模式
- 命令模式下，按 `D` 可以分离当前会话，运行后将会退出 tmux 进程，返回至上一级进程

这时，即使终端离线，session中运行的程序仍会保持运行

使用 `tmux a` 可以连接到之前分离的session

额外的命令

- 命令模式下：
 - `$` 重命名当前会话
 - `s` 选择会话列表
- 要在新建session时指定名称可以使用 `tmux new -s [name]`
- 要回到指定session，可以使用 `tmux a -t [name]`

关于Tmux的更多说明 (<https://leohxj.gitbooks.io/a-programmer-prepares/software/development/tmux.html>)

下载本文件 ([./Server_Usage.ipynb](#))