

运行配置说明

学号姓名：201250104 苏致成

运行前配置

GPU选型

华为云并无众多显卡可选，因此选择平台 *AutoDL*。

AutoDL

- 1. *2080Ti*：首先选择了 *2080Ti*，只有11G显存。显存过小，导致程序异常结束。
- 2. *Tesla p40*：其次选择 *Tesla p40*。时长共37小时，已完成但可惜并未保存。
- 3. *V100 – SXM2 – 32GB*：最后因为需要同时有大显存且支持 *Cuda10*，因此选择了 *V100 – SXM2 – 32GB*。

GPU型号简介

附GPU型号简介						目录
型号	显存	单精 (FP32)	半精 (FP16)	详细参 数	说明	一. 选择CPU 二. 选择GPU 三. 选择内存 附GPU型号简介
Tesla P40	24GB	11.76 T	11.76 T	查看	比较老的Pascal架构GPU，对于cuda11.x之前且对大显存有需求的算法是非常不错的选择	
TITAN Xp	12GB	12.15 T	12.15 T	查看	比较老的Pascal架构GPU，用作入门比较合适	
1080 Ti	11GB	11.34 T	11.34 T	查看	和TITANXp同时代的卡，同样适合入门，但是11GB的显存偶尔会比较尴尬	
2080Ti	11GB	13.45 T	53.8 T	查看	图灵架构GPU，性能还不错，老一代型号中比较适合做混合精度计算的GPU。性价比高	
V100	16/32GB	15.7 T	125 T	查看	老一代专业计算卡皇，半精性能高适合做混合精度计算	
3060	12GB	12.74 T	约24T	查看	如果1080Ti的显存正好尴尬了，3060是不错的选择，适合新手。需要使用cuda11.x	
A4000	16GB	19.17 T	约76T	查看	显存和算力都比较均衡，适合进阶过程使用。需要使用cuda11.x	
3080Ti	12GB	34.10 T	约70T	查看	性能钢炮，如果对显存要求不高则是非常合适的选择。需要使用cuda11.x	

选择镜像

选择 *Cuda 10.0* 版本，*python 3.7* 版本，*pytorch 1.1.0* 版本。

镜像	PyTorch 1.1.0	Python 3.7(ubuntu18.04)	Cuda 10.0
GPU	Tesla P40 * 1 显存: 24GB		
CPU	12核 Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz 内存: 60GB		
默认硬盘	系统盘: 20 GB	数据盘: 免费:50GB SSD 付费:0GB	
附加磁盘	无		
端口映射	无		
网络	上行宽带: 10MB/s 下行宽带: 10MB/s		
费用	¥0.78/时 ¥0.82/时		

运行流程

1. 使用 `xshell` 进行连接服务器，注意需要将端口号后置，[查看官方文档](#)。具体为 `ssh root@region-1.autodl.com 44562`。
2. `pip install -r requirements.txt`，也可以手动 `pip install requirements.txt` 中的文件。
`conda install` 也可以执行类似任务。
3. 手动在 `src` 目录下 `git clone bashmagic` 的[项目](#)。
4. 下载 `en_core_web_sm`（而不是 `en!`，因为 `en` 过期了），当然输 `en` 也可以，系统会自动替换为 `en_core_web_sm`，并需要手动修改对应代码。使用指令：`python -m spacy download en`。
5. 处理数据集：`sh preprocess.sh`，此命令脚本处理 `FB15k - 237, WN18RR, FB15k - 237 - attr` 和 `kinship`。
6. 示例指令：
 - 使用 `ConvTransE` 模型与 `FB15k - 237` 数据集，指令如下：

```
CUDA_VISIBLE_DEVICES=0 python main.py model ConvTransE init_emb_size 100
dropout_rate 0.4 channels 50 lr 0.001 kernel_size 3 dataset FB15k-237 process True
```

- 使用 `SACN` 模型与 `FB15k - 237` 数据集，指令如下：

```
CUDA_VISIBLE_DEVICES=0 python main.py model SACN dataset FB15k-237 process True
```

- 可以在 `src.spodernet.spodernet.utils.global_config.py` 中调节参数，或者在命令行中指定参数。并且对不同的数据集，需要分别调参。

其他问题

1. 加载学术资源慢，使用代理[加速](#)。
2. 查看显卡，输入 `nvidia-smi`。

```
root@container-e84811a152-f63865cc:~/SACN/SACN# nvidia-smi
Sat Oct 15 09:43:37 2022

+-----+
| NVIDIA-SMI 510.47.03      Driver Version: 510.47.03      CUDA Version: 11.6     |
+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M. |
+-----+-----+
|  0  Tesla V100-SXM2...    On      | 00000000:1A:00:0 Off  |      0MiB / 32768MiB |      0%      Default |
| N/A   37C    P0      38W / 300W | 0MiB / 32768MiB |           | N/A |
+-----+-----+

+-----+
| Processes: |
| GPU   GI    CI          PID    Type   Process name                      GPU Memory |
| ID   ID     ID              |                 | Usage     |
+-----+-----+
| No running processes found |
+-----+
```

3. 若想缩短时间, 可以更改 *main.py* 中循环次数 (如将 *epoch* 调为 300) 。当然结果会受一定影响。