

APSS Training Guide 【APSS训练指南】

[2024.03.20]

本指南指导用户如何构建APSS并行搜索策略项目的训练过程。

本项目训练过程支持的后端设备为：

- Ascend([Help](#))
- GPU
- CPU

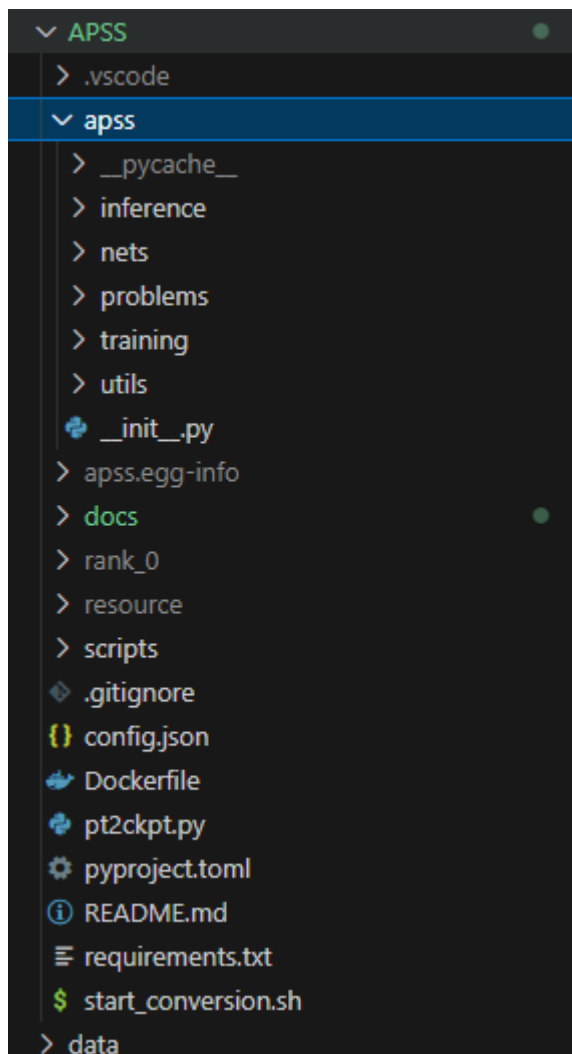
本指导文件以GPU环境和Ascend环境为例，所有代码在Python = 3.7; Mindspore = 2.2.0的Nvidia GPU V100和Ascend NPU 910B上通过测试。其中Ascend环境以90.90.93.240上的环境和代码为例。

目录

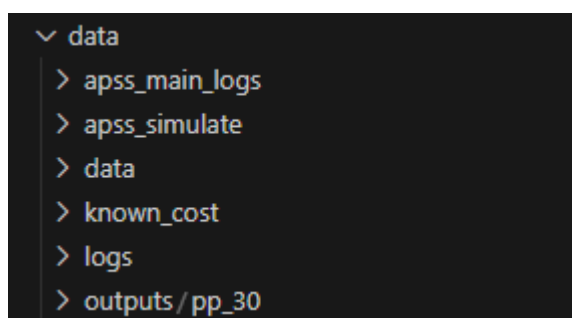
- [1. 项目清单](#)
- [2. 环境构建](#)
 - [GPU](#)
 - [Method 1: 使用Mindspore官方镜像并从源码构建](#)
 - [Method 2: 用我们已经构建好的镜像](#)
 - [Ascend](#)
- [程序运行](#)
- [训练原理](#)

1. 项目清单

- 代码包APSS (APSS.zip)
 - apss为项目源代码部分，inference表示推理部分代码，nets表示模型，problems表示我们抽象出来的拟训练问题，training表示训练部分代码，utils是一些工具类。
 - apss.egg-info为打包后测试使用pip安装apss包后的元数据信息，用户无需关心。
 - docs是一些说明文档，包括项目训练说明以及一些常见的问题。
 - resource是外部数据包/..data默认链接目录，主要包括日志、输出文件以及训练产生的ckpt，可由用户在config.json中配置。
 - scripts包含一些自动化处理的脚本。
 - config.json包含一些全局的配置，主要包括数据包目录配置、训练图启动模式，后端设备选择以及ID设置。
 - dockerfile为GPU训练环境镜像构建文件。
 - pyproject.toml为配置依赖启动文件。



- 数据包data(APSS-graphmode.zip中与APSS同级)



- 单独开辟了占用空间较大的数据存储，并在构建程序运行环境时，分别将源代码文件和数据包内容同时映射或放入运行环境中，数据包由代码文件APSS中的/resource目录进行映射。即 APSS/resource --> data，无需用户感知。
- GPU环境的Docker镜像 (<https://hub.docker.com/repository/docker/cheny1m/apss-mindspore-gpu-cuda11.1/general>)
 - 使用Mindspore官方镜像后，使用pip进行源码安装。
 - 使用[dockerfile](#)构建我们已经打包好的容器或者从docker hub上拉取。
- Mindspore在Ascend环境下暂未使用Docker([Mindspore官方安装界面](#))，且所有训练测试均在杭研所 90.90.93.240和90.90.93.242服务器既定的Mindspore=2.2.0环境下完成，所以本项目不包含Ascend环境下的Docker。仅能通过源码安装。

2. 环境构建

Requirements:

- Python >= 3.7
- Mindspore >= 2.2.0 ([Help](#))

2.1. GPU

Method 1: 使用Mindspore官方镜像并从源码构建

启动容器：将源代码目录APSS（本例中为/data01/cym/MindSpore/APSS）和数据包目录data（本例中为/data01/cym/MindSpore/data）分别映射到容器内部的APSS目录（本例中为/root/APSS）和APSS/resource目录（本例中为/root/APSS/resource） 注意：如果数据包的容器映射目录不为默认的resource，请在config.json中修改RESOURCE_DIR的value为您定义的目录。

```
docker run -itd -v /dec/shm:/dev/shm -v /data01/cym/MindSpore/APSS:/root/APSS -v /data01/cym/MindSpore/data:/root/APSS/resource --name apss --runtime=nvidia swr.cn-south-1.myhuaweicloud.com/mindspore/mindspore-gpu-cuda11.1:2.2.0 /bin/bash
```

```
docker exec -it apss /bin/bash
```

从源码构建：

```
cd ~/APSS
pip install -e .
```

```
upai@ubuntu-MS-7C73:/data01/cym/MindSpore$ docker run -itd -v /dec/shm:/dev/shm -v /data01/cym/MindSpore/APSS:/root/APSS -v /data01/cym/MindSpore/data:/root/APSS/resource --name apss --runtime=nvidia swr.cn-south-1.myhuaweicloud.com/mindspore/mindspore-gpu-cuda11.1:2.2.0 /bin/bash
20f8cb4c41f08fc948f961483ca8565c45c82fcb3cb96526216defe90d5c4c06
upai@ubuntu-MS-7C73:/data01/cym/MindSpore$ docker exec -it apss /bin/bash
root@20f8cb4c41f0:/# cd ~/APSS
root@20f8cb4c41f0:/~/APSS# pip install -e .
Looking in indexes: http://mirrors.aliyun.com/pypi/simple/
Obtaining file:///root/APSS
Installing build dependencies ... done
Checking if build backend supports build_editable ... done
Getting requirements to build editable ... done
Installing backend dependencies ... done
Preparing editable metadata (pyproject.toml) ... done
Requirement already satisfied: mindspore>=2.2.0 in /usr/local/python-3.7.5/lib/python3.7/site-packages (from apss==0.4.0) (2.2.0)
Collecting tensorboard-logger (from apss==0.4.0)
  Downloading http://mirrors.aliyun.com/pypi/packages/87/7a/ec0fd26dba69191f82eb8f38f5b401c124f45a207490a7ade6ea9717ecdb/tensorboard_logger-0.1.0-py3-none-any.whl (17 kB)
Requirement already satisfied: numpy in /usr/local/python-3.7.5/lib/python3.7/site-packages (from apss==0.4.0) (1.21.6)
Collecting tqdm (from apss==0.4.0)
  Downloading http://mirrors.aliyun.com/pypi/packages/2a/14/e75e52d521442e2fcc9f1df3c5e456aeadd034203d4797867980de558ab34/tqdm-4.66.2-py3-none-any.whl (78 kB)
78.3/78.3 kB 18.0 kB/s eta 0:00:00
Requirement already satisfied: protobuf>=3.13.0 in /usr/local/python-3.7.5/lib/python3.7/site-packages (from mindspore>=2.2.0->apss==0.4.0) (3.20.1)
Requirement already satisfied: asttokens>=2.0.4 in /usr/local/python-3.7.5/lib/python3.7/site-packages (from mindspore>=2.2.0->apss==0.4.0) (2.4.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/python-3.7.5/lib/python3.7/site-packages (from mindspore>=2.2.0->apss==0.4.0) (9.5.0)
Requirement already satisfied: scipy>=1.5.4 in /usr/local/python-3.7.5/lib/python3.7/site-packages (from mindspore>=2.2.0->apss==0.4.0) (1.7.3)
Requirement already satisfied: packaging>=20.0 in /usr/local/python-3.7.5/lib/python3.7/site-packages (from mindspore>=2.2.0->apss==0.4.0) (23.2)
Requirement already satisfied: psutil>=5.6.1 in /usr/local/python-3.7.5/lib/python3.7/site-packages (from mindspore>=2.2.0->apss==0.4.0) (5.9.6)
Requirement already satisfied: astunparse>=1.6.3 in /usr/local/python-3.7.5/lib/python3.7/site-packages (from mindspore>=2.2.0->apss==0.4.0) (1.6.3)
Requirement already satisfied: six in /usr/local/python-3.7.5/lib/python3.7/site-packages (from mindspore>=2.2.0->apss==0.4.0) (1.16.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/python-3.7.5/lib/python3.7/site-packages (from astunparse>=1.6.3->mindspore>=2.2.0->apss==0.4.0) (0.41.2)
Building wheels for collected packages: apss
  Building editable for apss (pyproject.toml) ... done
  Created wheel for apss: filename=apss-0.4.0-editable-py3-none-any.whl size=7109 sha256=7be514d50bd1ae332c2718d2e22f232ab38e027dd844e4a0f28363eccee8e8c2
  Stored in directory: /tmp/pip-ephem-wheel-cache-2nt78ujr/wheels/f3/26/a6/5323870c786336682397a13eb66951fb4f4b825e89bfd66479
Successfully built apss
Installing collected packages: tqdm, tensorboard-logger, apss
Successfully installed apss-0.4.0 tensorboard-logger-0.1.0 tqdm-4.66.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.3 -> 24.0
[notice] To update, run: pip install --upgrade pip
root@20f8cb4c41f0:/~/APSS#
```

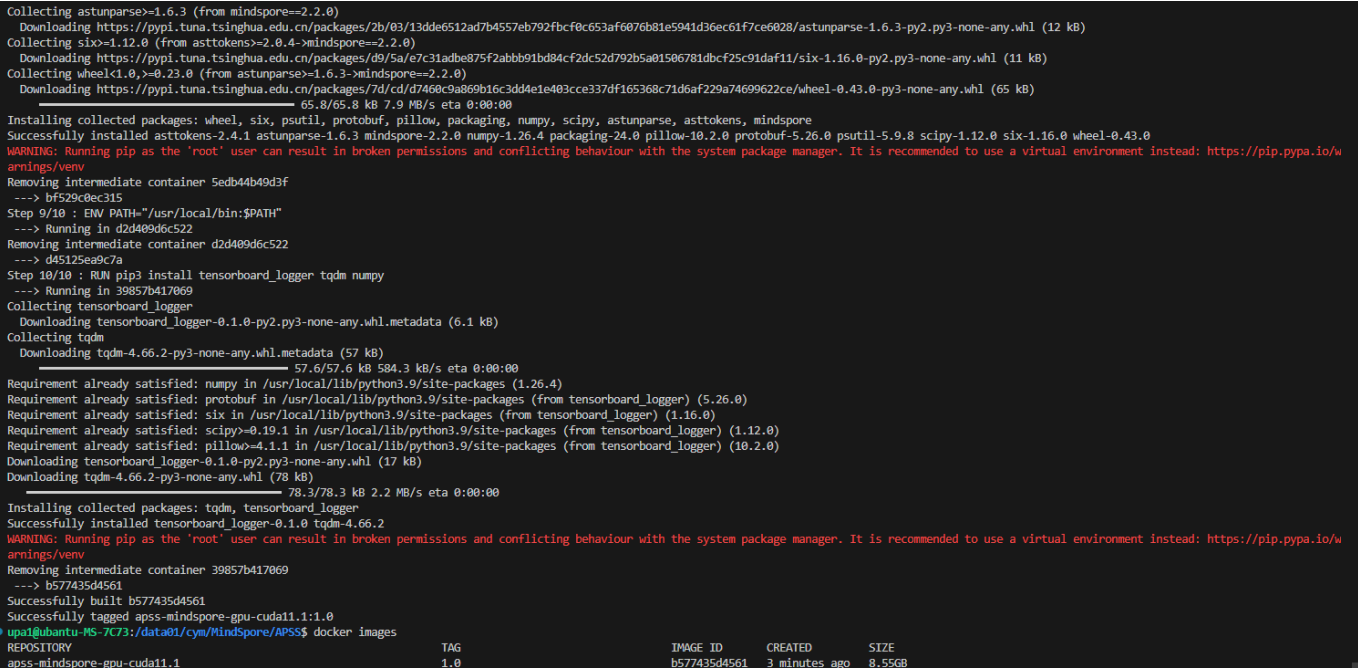
Method 2: 使用我们已经构建好的镜像

[可选1]拉取镜像

```
docker push chenylm/apss-mindspore-gpu-cuda11.1:1.0
```

[可选2]或者通过dockerfile构建镜像

```
cd APSS
docker build -t apss-mindspore-gpu-cuda11.1:1.0 .
```



获得镜像后,启动容器（代码映射解释见[方法1 启动容器](#)）：如采用通过dockerfile构建镜像，请将下述命令中的chenylm/apss-mindspore-gpu-cuda11.1:1.0替换为apss-mindspore-gpu-cuda11.1:1.0

```
docker run -itd -v /dev/shm:/dev/shm -v /home/upa1/cym/MindSpore/APSS:/root/APSS -v /home/upa1/cym/MindSpore/data:/root/APSS/resource --name apss --runtime=nvidia chenylm/apss-mindspore-gpu-cuda11.1:1.0 /bin/bash

docker exec -it apss /bin/bash
cd ~/APSS
```

2.2 Ascend

进入已有的Mindspore环境后从源码构建即可：

```
# from 90.90.93.240
cd /home/xby
source xby_env.sh r2_3
```

```
cd ~/APSS
pip install -e .
```

3. 程序运行

3.1 设置运行环境的context

本步骤主要设置运行时的目标设备和模式，默认目标设备为`Ascend`，默认运行模式为`PYNATIVE_MODE`。如需查看详情和修改目标设备及运行模式，请在`config.json`中修改。

- "DEVICE_TARGET": 设置运行设备。支持[Ascend],[GPU],[CPU].
- "CONTEXT_MODE": 设置运行环境context的mode,在[0]: (GRAPH_MODE)和[1]: (PYNATIVE_MODE)中选择。

```
APSS > {} config.json > ...
1  {
2      "RESOURCE_DIR": "resource/",
3      "RESOURCE_DIR_DESC": "指定数据文件的目录路径。",
4
5      "DEVICE_TARGET": "GPU",
6      "DEVICE_TARGET_DESC": "设置后端设备。支持[Ascend],[GPU],[CPU]。",
7
8      "DEVICE_ID": 0,
9      "DEVICE_ID_DESC": "设置具体运行的后端设备编号，取值应在[0,最大卡数-1]",
10
11     "CONTEXT_MODE": 1,
12     "CONTEXT_MODE_DESC": "设置运行环境context的mode,在[0](GRAPH_MODE)和[1](PYNATIVE_MODE)中选择。"
13 }
14
```

3.2 一步执行训练

```
python -m apss.training.apss_run --graph_size 30 --num_split 15 --rebuild_data
```

- `graph_size` , `num_split` 分别代表了问题的层数大小和需要执行pipeline划分的数量，两个命令行参数共同描述了所训练问题的大小，可根据需求动态调整。目前`graph_size`取值范围为[8,18,25,30,42,54,102], `num_split`取值范围为[1,3,7,15,31,63]。
- `rebuild_data` 表示是否在执行训练前，从Data Synthesizer中生成训练数据，默认建议开启。如果需要从`ckpt`中接续训练或无需改变之前生成的训练数据直接禁用`--rebuild_data`参数即可。生成的训练数据可在数据包data即`/resource`映射目录下找到。

```

'problem': 'pp',
'resume': None,
'run_name': 'pp_30_16_20240319T153800',
'run_test': False,
'save_dir': 'outputs/pp_30/pp_30_16_20240319T153800',
'seed': 1234,
'shrink_size': None,
'tanh_clipping': 10.0,
'use_cuda': True,
'val_dataset': 'data/pp/pp_30_16_validation_seed1234.pkl',
'val_size': 10000)
device: GPU
mode: 1
Create ThLogger to resource/logs/pp_30/pp_30_16_20240319T153800
problem is: <class 'apss.problems.pp.problem_pp_pp'>
The model has been initialized!
selected rollout...
Evaluating baseline model on evaluation dataset
100% | 18/10 [00:02<00:00, 4.42it/s]
rollout initialization complete!
Start train epoch 0, lr=1e-04 for run pp_30_16_20240319T153800
Evaluating baseline on dataset...
100% | 1250/1250 [01:49<00:00, 11.48it/s]
train batch baseline name: <apss.training.reinforce_baselines_pp.RolloutBaselinePP object at 0x7f5cd31a1450>
| 0/2500 [00:00<, ?it/s]
loss: 23.504658, 0
el_dp cost: 9.927049, hl cost: 12.373417, eval hl cost: 12.919736
pi: [2, 1, 1, 2, 2, 1, 2, 1, 4, 4, 1, 4, 2, 1, 1, 1]
baseline pi: [11, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1]
1% | 18/2500 [00:09<16:11, 2.56it/s]
Sampled bad values, resampling!
1% | 24/2500 [00:11<15:55, 2.59it/s]
Sampled bad values, resampling!
1% | 26/2500 [00:12<15:55, 2.59it/s]
Sampled bad values, resampling!
1% | 36/2500 [00:16<15:49, 2.60it/s]
Sampled bad values, resampling!
2% | 62/2500 [00:26<15:47, 2.59it/s]
```

执行上述代码会执行apss的训练。每个模型训练默认训练100个epoch，每个epoch训练1,280,000条数据，batch_size为512。如需微调这些超参，请在[options.py](#)中调整。

3.3 模型参数保存

- 执行训练后，本次运行的参数文件及.ckpt文件将保存在数据包的/output文件夹下，日志保存在数据包的/log文件夹下，可以通过tensorboard_logger在浏览器中实时查看训练过程及其数据。
- 当前已经训练好的可用于推理的权重保存在resource/outputs/pp_30下不带时间戳的文件夹中。

```

> apss
> apss.egg-info
> docs
▼ resource
  > apss_main_logs
  > apss_simulate
  > data
  > known_cost
  > logs
  ▼ outputs / pp_30
    > pp_30_2
    > pp_30_4
    > pp_30_8
    > pp_30_16
    > pp_30_16_20240314T101140
    > pp_30_16_20240319T152858
    > pp_30_16_20240319T153446
    > pp_30_16_20240319T153800
  {} args.json
```

训练原理

