

Upgrade environment for NS block (Chinese version)

Author: [Tuo Wang \(王拓\)](#)

Contents

- [Overview](#)
 - [1. Problem](#)
 - [2. Test whether the local docker is available on the server](#)
- [Step-by-step Methods](#)
 - [1. Run train.py will report error: No such file or directory](#)
 - [2. Replace Python in the environment](#)
 - [3. Install CUSA 11.0, Pytorch 1.7.0](#)
 - [4. Modify *nvcc_args*](#)
 - [5. Modify *cxx_args*](#)
 - [6. Error reported: nvcc fatal](#)
 - [7. Error reported: ninja not installed](#)
 - [8. CMakeList.txt](#)

Overview

1. Problem

In PNS-Net ([Paper](#) | [Code](#)), it require this environment:

Mark 1 ↓

Our core design is built on CUDA OP with torchlib. Please ensure the base **CUDA toolkit version is 10.x (not at conda env)**, and then build the NS Block:

```
cd ./lib/PNS python setup.py build develop
```

```
# setup.py
from setuptools import setup
from torch.utils.cpp_extension import BuildExtension, CUDAExtension
from os.path import join

# project_root = './lib/PNS/PNS_Module'
project_root = 'PNS_Module'
sources = [join(project_root, file) for file in ['sa_ext.cpp',
                                                'sa.cu', 'reference.cpp']]

nvcc_args = [
    '-gencode', 'arch=compute_61,code=sm_61',
    '-gencode', 'arch=compute_70,code=sm_70',
```

```

        '-gencode', 'arch=compute_70,code=compute_70'
    ]
    cxx_args = ['-std=c++11']

    setup(
        name='self_cuda',
        ext_modules=[
            CUDAExtension('self_cuda_backend',
                          sources, extra_compile_args={'cxx': cxx_args, 'nvcc':
nvcc_args})
        ],
        cmdclass={
            'build_ext': BuildExtension
        })

```

Mark 1 ↑

But the server is like this:

NVIDIA-SMI 470.57.02				Driver Version: 470.57.02				CUDA Version: 11.4			
GPU Name				Persistence-M	Bus-Id	Disp.A	Memory-Usage	Volatile Uncorr. ECC	GPU-Util	Compute M.	MIG M.
Fan	Temp	Perf	Pwr:Usage/Cap								
0	NVIDIA	A100-PCI...	Off	00000000:C1:00.0	Off						
N/A	28C	P0	32W / 250W	0MiB / 40536MiB				0%	Default	Disabled	

2. Test whether the local docker is available on the server

Package the docker environment of local PNS-Net (py3.6-torch 1.1.0-cuda 10.1):

NVIDIA-SMI 410.48				Driver Version: 410.48			
-------------------	--	--	--	------------------------	--	--	--

Open the image container, and use the code **Mark 1**, it is found that the compilation process is no different from the local one. However, when running the code, **I find that does not call GPU, and the GPU occupancy rate is 0. There is no such program in NVIDIA SMI unless I use the bash command `ps -ax`**.

Upgrade the environment to a higher version of pytorch and CUDA (such as torch 1.7.0 and cuda11.0). The above code **Mark 1** can go running, but if I run the ***train.py***, it will report an error:

```

ImportError: libcublas.so.10.0: cannot open shared object file: No such
file or directory

```

After encountering various problems, the following scheme for upgrading the environment is finally formed.

Step-by-step Methods

1. Why compile related xxx.cpp files is successful when pytorch and CUDA have been upgraded but run **train.py** will report this error:

```
ImportError: libcublas.so.10.0: cannot open shared object file: No such file or directory
```

As I haven't learned how to use the `from torch.utils.cpp_extension import BuildExtension, CUDAExtension` to compile the C programs, I've been modifying pytorch and CUDA versions.

Finally, it was found that it was because of the variable `nvcc_args` and `cxx_Args` in **setup.py** does not match the upgraded CUDA version.

If you are interested, you can simply take a look at these articles ([pytorch通过torch.utils.cpp_extension构建CUDA/C++拓展](#) | [Pytorch cuda extension的例子有没有简单点的啊....?](#) | [pytorch 的 CUDA 编程 CUDAExtension](#)) to know what does the **setup.py** does roughly.

2. Replace Python in the environment (Operations that may be required.)

My environment is Python 3.6, if you meet some other errors, I suggest you change it to Python 3.6 and then try it again.

Change the python version by this: [docker容器内更换python版本以及pip版本](#)

If above bash command doesn't work, you can try this:

```
echo alias python=python3 >> ~/.bashrc
source ~/.bashrc
```

3. Install CUSA 11.0 , Pytorch 1.7.0 (The server driven environment is relatively high and is downward compatible)

You can directly pull the image from the docker hub or install it by yourself.

(1) CUDA is usually installed together during the installation of pytorch:

[Pytorch Official Website:](#)

v1.7.0

Conda

OSX

```
# conda
conda install pytorch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0 -c pytorch
```

Linux and Windows

```
# CUDA 9.2
conda install pytorch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0 cudatoolkit=9.2 -c pytorch

# CUDA 10.1
conda install pytorch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0 cudatoolkit=10.1 -c pytorch

# CUDA 10.2
conda install pytorch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0 cudatoolkit=10.2 -c pytorch

# CUDA 11.0
conda install pytorch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0 cudatoolkit=11.0 -c pytorch

# CPU Only
conda install pytorch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0 cpuonly -c pytorch
```

Wheel

OSX

```
pip install torch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0
```

Linux and Windows

```
# CUDA 11.0
pip install torch==1.7.0+cu110 torchvision==0.8.0+cu110 torchaudio==0.7.0 -f https://download.pytorch.org/wheels/torch_cuda110.html

# CUDA 10.2
pip install torch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0 -f https://download.pytorch.org/wheels/torch_cuda102.html

# CUDA 10.1
pip install torch==1.7.0+cu101 torchvision==0.8.0+cu101 torchaudio==0.7.0 -f https://download.pytorch.org/wheels/torch_cuda101.html

# CUDA 9.2
pip install torch==1.7.0+cu92 torchvision==0.8.0+cu92 torchaudio==0.7.0 -f https://download.pytorch.org/wheels/torch_cuda92.html

# CPU only
pip install torch==1.7.0+cpu torchvision==0.8.0+cpu torchaudio==0.7.0 -f https://download.pytorch.org/wheels/torch_cpu.html
```

Personally, I think the command of `pip install` is too slow and the connection may break, so I recommend you download the installation package in the official website ([torch and torchvision](#) |

[torchaudio](#)) and then upload it to the server.

(2) Or install CUDA by yourself: [服务器docker安装多版本cuda](#)

4. Modify *nvcc_args*

According to this article ([各种 NVIDIA 架构所匹配的 arch 和 gencode](#)), I modify the code in **setup.py**:

```
nvcc_args = [
    '-gencode', 'arch=compute_61,code=sm_61',
    '-gencode', 'arch=compute_70,code=sm_70',
    '-gencode', 'arch=compute_70,code=compute_70'
]
```

to this:

```
nvcc_args = [
    '-gencode', 'arch=compute_52,code=sm_52',
    '-gencode', 'arch=compute_60,code=sm_60',
    '-gencode', 'arch=compute_61,code=sm_61',
    '-gencode', 'arch=compute_70,code=sm_70',
    '-gencode', 'arch=compute_75,code=sm_75',
    '-gencode', 'arch=compute_80,code=sm_80',
    '-gencode', 'arch=compute_80,code=compute_80'
]
```

5. Modify *cxx_args*

Modify the code in **setup.py**:

```
cxx_args = ['-std=c++11']
```

to this:

```
cxx_args = ['-std=c++14']
```

6. Error reported: nvcc fatal : A single input file is required for a non-link phase when an outputfile is specified:

```
CMake Warning (dev) in ad-census_generated_main.cu.o.cmake:
  Syntax Warning in cmake code at

    /home/xxx/software/AD-Census/build/CMakeFiles/ad-census.dir/ad-
```

```
census_generated_main.cu.o.cmake:79:198
```

Argument not separated from preceding token by whitespace.
This warning is **for** project developers. Use `-Wno-dev` to suppress it.

nvcc fatal : A single input file is required **for** a non-link phase when an outputfile is specified

CMake Error at ad-census_generated_main.cu.o.cmake:206 (message):

```
Error generating
/home/xxx/software/AD-Census/build/CMakeFiles/ad-census.dir/./ad-
census_generated_main.cu.o
```

Add this code in the begining of the **CMakeLists.txt** ([The tutorial link](#)):

```
get_directory_property(dir_defs DIRECTORY ${CMAKE_SOURCE_DIR}
COMPILE_DEFINITIONS)
set(vtk_flags)
foreach(it ${dir_defs})
    if(it MATCHES "vtk*")
        list(APPEND vtk_flags ${it})
    endif()
endforeach()
```

7. Error reported: ninja not installed

```
pip install ninja
```

8. CMakeList.txt

Modify this code:

```
set_property(TARGET SA PROPERTY CXX_STANDARD 11)
```

to this:

```
set_property(TARGET SA PROPERTY CXX_STANDARD 14)
```