



Python 3.6.9로 전환하기

☰ 선택	
🕒 최종 편집 일시	@2025년 11월 27일 오후 3:50
👤 최종 편집자	🐼 임규빈 서울 미래자동차공학과

1. 원래 깔려있던 python 3.6.9 사용하기

```
# 1. Pyenv 환경에서 기본 시스템으로 전환
pyenv global system

# 2. 버전 확인(Python 3.6.9 필요!!!)
python3 --version

# 3. 필수 시스템 패키지 설치
sudo apt-get update
sudo apt-get install python3-pip

# 4. Pip 업그레이드 (Python 3.6 호환 마지막 버전)
python3 -m pip install --upgrade "pip<22.0" "setuptools<59.7.0" "protobuf<4" Cython
```

2. 딥러닝 라이브러리 설치(NVIDIA Wheels)

pip install 로 받으면 CPU를 사용하는 버전이 깔리기 때문에 NVIDIA가 제공하는 전용 파일을 받아야 함.

<https://forums.developer.nvidia.com/t/pytorch-for-jetson/72048>

```
# 1. PyTorch 1.10.0 (GPU) 다운로드 & 설치
wget https://nvidia.box.com/shared/static/fjtbno0vpo676a25cgvuqc1wty0fkkg6.whl -O torch-1.10.0-cp36-cp36m-linux_aarch64.whl

python3 -m pip install torch-1.10.0-cp36-cp36m-linux_aarch64.whl
```

```
# 2. ONNX Runtime 1.11.0 (GPU) 다운로드 & 설치
# (링크 만료 시 1.10.0 사용)
wget https://nvidia.box.com/shared/static/b18zgrudrbf06id1q4j600j56d474581.whl -O onnxruntime_gpu-1.11.0-cp36-cp36m-linux_aarch64.whl

python3 -m pip install onnxruntime_gpu-1.11.0-cp36-cp36m-linux_aarch64.whl
```

3. 기타 라이브러리 설치

```
#1. Core dumped 예방 => Illegal Instruction 에러 방지 (OpenBLAS 설정)
# (이건 영구 적용을 위해 .bashrc에도 넣었습니다)
echo 'export OPENBLAS_CORETYPE=ARMV8' >> ~/.bashrc
source ~/.bashrc

# (선택. 아리온 174는 안함) Scipy와 sklearn 설치 중 메모리 부족 방지 위해 가상메
모리(스왑 메모리) 늘리기
sudo fallocate -l 4G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile

# (스왑메모리 설정 이후, 파일을 모두 설치한 후 아래 실행 필요)
# 스왑 파일 삭제 (설치 후 디스크 확보)
sudo swapoff /swapfile
sudo rm /swapfile

# 2. Scipy, sklearn 등등 설치(안전 빌드)
python3 -m pip install "scipy==1.5.4" "scikit-learn==0.24.2" "pandas==1.1.5" --no-cache-dir
python3 -m pip install "matplotlib==3.3.4" "numpy==1.19.5" "seaborn==0.11.2"

# 3. pytorch_lightning 설치
# 의존 라이브러리 먼저 설치
python3 -m pip install "torchmetrics==0.6.0" "pydeprecate==0.3.1" "future
```

```
>=0.17.1"
python3 -m pip install "pytorch-lightning==1.5.10"
```

4. pycuda 설치

```
# 1. NVCC 컴파일러 경로 설정 (필수)
echo 'export PATH=/usr/local/cuda/bin:$PATH' >> ~/.bashrc
echo 'export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH' >> ~/.bashrc
source ~/.bashrc

# 2. PyCUDA 설치
python3 -m pip install pycuda
```

5. 검토

```
# 1. torch, cuda 설정 확인
# 환경변수 설정 (습관처럼 해주세요)
export OPENBLAS_CORETYPE=ARMV8

# 확인 코드 실행
python3 -c "import pytorch_lightning as pl; import torch; print('PL Version:', pl.__version__); print('CUDA Available:', torch.cuda.is_available())"
```



1. 아래 코드 실행(반드시 다른 모든 프로그램 종료 후 실행할 것!!! vscode 창 1개가 최대.)

```
cnn_122_onnx_trt_fp_32_TX2.py
```

2. /artifact 폴더에 .engine, .pt, .onnx 파일 생성 확인
3. 아래 코드 실행해서 검증

```
test.py
```

추가 유용한 프로그램

- jtop: 메모리 및 gpu 사용량을 알 수 있음.

```
sudo pip3 install jetson-stats
```

⇒ restart 필요. 터미널에 jtop 입력해서 실행

- firefox

```
sudo apt-get update
```

```
sudo apt-get install firefox -y
```

```
firefox &
```