

주요하게 설치해야 할 항목은 3 가지이다.

1. CUDA
2. cuDNN
3. tensorflow-gpu

2018/11/30 현재 기준 tensorflow 홈페이지에서 안정판(stable)을 보면 1.2 로 되어있다. CUDA 툴킷은 10.0 까지 나왔지만 경험상 tf1.2 는 CUDA9.0 버전만 지원한다. cuDNN 은 버전 7 까지 나와 있지만 낮은 버전을 받을 필요 없이 cuda 버전에 맞는 cuDNN7 버전을 다운 받으면 된다.

## 1. CUDA 설치하기

### 1.1 준비

#### 1.1.1 GPU 확인

```
$ lspci |grep i nvidia
```

```
iis@iis-lab:~$ lspci | grep -i nvidia
01:00.0 VGA compatible controller: NVIDIA Corporation Device 1b80 (rev a1)
01:00.1 Audio device: NVIDIA Corporation Device 10f0 (rev a1)
```

미리 nvidia-smi (GPU 드라이버 겸 툴킷)을 설치하는 것이 낫다.

CUDA 를 설치하면서 만약 설치되어 있지 않으면 알아서 설치를 하는데 이 때 버전에 맞지 않는 드라이버가 간혹 깔리게 되면 다시 지우고 다시 설치해야 하는 번거로움이 있다.

```
$ nvidia-smi
```

명령어로 확인해 보고 설치되어 있지 않거나 설치하려고 하는 cuda 버전과 맞지 않은 드라이버가 깔려 있을 경우 제거하고 다시 깐다.

CUDA 버전	Driver
CUDA 9.2	396.xx
CUDA 9.1	387.xx
CUDA 9.0	384.xx
CUDA 8.0 (GA2)	375.xx
CUDA 8.0	367.4x
CUDA 7.5	352.xx
CUDA 7.0	346.xx

우리는 cuda-9.0 을 사용하기 때문에 384.xx 버전을 깔면 된다.

```
$ sudo apt-get install nvidia-384
```

### 1.1.2 OS 버전 확인

```
$ uname -m && cat /etc/*release
```

```
iis@iis-lab:~$ uname -m && cat /etc/*release
x86_64
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.3 LTS"
NAME="Ubuntu"
VERSION="16.04.3 LTS (Xenial Xerus)"
```

첫 줄에 x86\_64 이 intel cpu 64bit 라는 뜻이므로 이어서 다운 받을 때 해당 버전으로 다운 받으면 된다. 참고로 tensorflow gpu 버전은 32 비트를 지원하지 않는다. (2018 년 현재 기준)

### 1.1.3 gcc 확인

```
$ gcc --version
```

만약 에러가 난다면 gcc 를 먼저 다운 받아야 한다.

### 1.1.4 시스템 헤더 확인

설치된 커널에 알맞는 시스템 헤더가 설치되어 있어야 한다.

```
$ sudo apt-get install linux-headers-$(uname -r)
```

명령으로 헤더를 설치한다 (대부분 이미 설치되어 있다).

### 1.1.5 설치

설치 방식 선택: local or net

Local 은 전체 설치 파일은 다운 받아서 설치하는 방식이고 net 은 정보 파일만 받아서 인터넷을 통해 자동으로 다운받아 설치하는 방식이다. 현재 tensorflow1.2 는 9.0 을 지원하기 때문에 최신 버전 다운 링크는 <https://developer.nvidia.com/cuda-downloads> 이지만 CUDA9.0 다운로드 링크로 접속한다.

<https://developer.nvidia.com/cuda-90-download-archive> 혹시 나중에 tensorflow 가 CUDA 최신버전을 지원한다면 최신 버전 링크에서 다운 받으면 된다.

#### 1.1.5.1 net 방식

**Select Target Platform** ⓘ

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows	Linux	Mac OSX			
Architecture ⓘ	x86_64	ppc64le				
Distribution	Fedora	OpenSUSE	RHEL	CentOS	SLES	Ubuntu
Version	17.04	16.04				
Installer Type ⓘ	runfile (local)	deb (local)	deb (network)	cluster (local)		

**Download Installer for Linux Ubuntu 16.04 x86\_64**

The base installer is available for download below.

➤ **Base Installer**

**Download [2.8 KB]** ⬇️

Installation Instructions:

1. ``sudo dpkg -i cuda-repo-ubuntu1604_9.0.176-1_amd64.deb``
2. ``sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub``
3. ``sudo apt-get update``
4. ``sudo apt-get install cuda``

Other installation options are available in the form of meta-packages. For example, to install all the library packages, replace "cuda" with the "cuda-libraries-9-0" meta package. For more information on all the available meta packages click [here](#).

```
$ sudo dpkg -i cuda-repo-ubuntu1604_9.0.176-1_amd64.deb
```

```
$ sudo apt-key adv --fetch-keys
```

```
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install cuda-9-0
```

(링크에는 cuda 로 끝나지만 저대로 설치하면 10.0 이 설치된다. 따라서 설

치할 때 되도록이면 버전을 명시해 주는 것이 좋다.)

### 1.1.6 local 방식

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows	Linux	Mac OSX			
Architecture ⓘ	x86_64	ppc64le				
Distribution	Fedora	OpenSUSE	RHEL	CentOS	SLES	Ubuntu
Version	17.04	16.04				
Installer Type ⓘ	runfile (local)	deb (local)	deb (network)	cluster (local)		

### Download Installers for Linux Ubuntu 16.04 x86\_64

The base installer is available for download below.  
There are 4 patches available. These patches require the base installer to be installed first.

➤ Base Installer

Download (1.2 GB) 

Installation Instructions:

1. ``sudo dpkg -i cuda-repo-ubuntu1604-9-0-local_9.0.176-1_amd64.deb``
2. ``sudo apt-key add /var/cuda-repo-<version>/7fa2af80.pub``
3. ``sudo apt-get update``
4. ``sudo apt-get install cuda``

Other installation options are available in the form of meta-packages. For example, to install all the library packages, replace "cuda" with the "cuda-libraries-9-0" meta package. For more information on all the available meta packages click [here](#).

```
$ sudo dpkg -i cuda-repo-ubuntu1604-9-0-local_9.0.176-1_amd64.deb
$ sudo apt-key add /var/cuda-repo-<version>/7fa2af80.pub
$ sudo apt-get update
$ sudo apt-get install cuda-9-0
```

#### 1.1.7 환경변수 설정

전체 사용자가 다 사용하게 하기 위해서 home 에 있는 .bashrc 말고 /etc/profile 을 수정해 준다.

```
$ cd /etc/
$ sudo vi profile
```

```

# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "$PS1" ]; then
  if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
unset i
fi

export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/cuda9.0/lib64:/usr/local/cuda-9.0/extras/CUPTI/lib64"
export CUDA_HOME=/usr/local/cuda-9.0
export PATH=/usr/local/cuda-9.0/bin:$PATH

```

마지막에 두 줄을 추가한다.

```

export CUDA_HOME=/usr/local/cuda-9.0
export PATH=/usr/local/cuda-9.0/bin:$PATH

```

파일을 닫아주고

```
$ source /etc/profile
```

로 실행 시켜주고 cuDNN 설치를 위해 재부팅한다.

```
$ nvcc --version
```

명령어로 cuda 설치 버전을 확인할 수 있다.

```
$ nvidia-smi
```

명령어로도 GPU 상태를 확인 할 수있다.

제거:

```
$ sudo apt-get purge nvidia-*
```

## 2. cuDNN 설치

### 2.1 설치

링크: <https://developer.nvidia.com/rdp/cudnn-download>

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

☒ I Agree To the Terms of the cuDNN Software License Agreement

Note: Please refer to the [Installation Guide](#) for release prerequisites, including supported GPU architectures and compute capability

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the [Deep Learning SDK Docur](#)

[Download cuDNN v7.4.1 \[Nov 8, 2018\], for CUDA 10.0](#)

[Download cuDNN v7.4.1 \[Nov 8, 2018\], for CUDA 9.2](#)

[Download cuDNN v7.4.1 \[Nov 8, 2018\], for CUDA 9.0](#)

## Library for Windows, Mac, Linux and Ubuntu (x86\_64 architecture)

[cuDNN Library for Windows 7](#)

[cuDNN Library for Windows 10](#)

[cuDNN Library for Linux](#)

[cuDNN Runtime Library for Ubuntu16.04 \[Deb\]](#)

[cuDNN Developer Library for Ubuntu16.04 \[Deb\]](#)

[cuDNN Code Samples and User Guide for Ubuntu16.04 \[Deb\]](#)

[cuDNN Runtime Library for Ubuntu14.04 \[Deb\]](#)

[cuDNN Developer Library for Ubuntu14.04 \[Deb\]](#)

[cuDNN Code Samples and User Guide for Ubuntu14.04 \[Deb\]](#)

위 세 파일을 다운받는다.

다운받은 폴더로 이동하여 아래 명령을 입력한다.

```
$sudo dpkg -i libcudnn7_7.0.3.11-1+cuda9.0_amd64.deb
$sudo dpkg -i libcudnn7-dev_7.0.3.11-1+cuda9.0_amd64.deb
$sudo dpkg -i libcudnn7-doc_7.0.3.11-1+cuda9.0_amd64.deb
```

위에서 CUDA 를 설치할 때 환경 변수를 설정했던 방법대로 /etc/profile 에

```
export
```

```
LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/cuda9.0/lib64:/usr/local/cuda/extras/CUPTI/lib64
```

를 추가하고 source 하고 재부팅 해준다.

## 2.2 설치 확인 테스트

설치된 cudnn 테스트 코드를 홈으로 복사한다.

```
$cp -r /usr/src/cudnn_samples_v7/ $HOME
```

mnist 데이터를 사용한 테스트를 위해 복사된 폴더 내 mnist 폴더로 이동한다.

```
$ cd $HOME/cudnn_samples_v7/mnistCUDNN
```

컴파일 한다.

```
$make clean && make
```

실행한다.

```
$ ./mnistCUDNN
```

결과 마지막에

`Test passed!` 라는 문구가 뜨면 설치 성공한 것이다.

3. Tensorflow 설치 (<https://www.tensorflow.org/install/> )

```
$ sudo pip install tensorflow-gpu
```

확인:

명령창서 python 을 열고 tensorflow 를 임포트 한다.

```
>>> import tensorflow as tf
```

```
>>> sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
```

이 때 뜨는 메시지 마지막에 `gpu:X` 이렇게 끝나면 GPU 에서 잘 돌아가고 있는 것이다. CPU 에서 돌아가면

`cpu:0` 으로 끝난다.