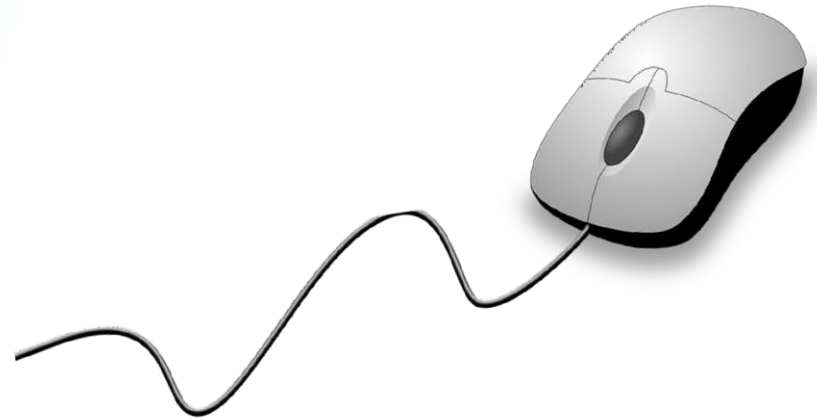


공개SW 솔루션 설치 & 활용 가이드

기타 > AI



# 제대로 배워보자

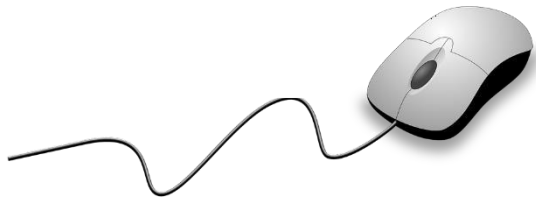
How to Use Open Source Software

---

Open Source Software Installation & Application Guide



오픈소스 소프트웨어 통합지원센터  
Open Source Software Support Center



# CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개
6. 활용예제
7. FAQ
8. 용어정리

# 1. 개요



<b>소개</b>	<ul style="list-style-type: none"> <li>TensorFlow™는 데이터 흐름 그래프를 사용하는 수치 연산용 오픈소스 소프트웨어 라이브러리</li> <li>그래프의 노드는 수학적 연산을 나타내며, 그래프의 변은 노드 간에 전달되는 다차원 데이터 배열(텐서)를 표현</li> </ul>		
<b>주요기능</b>	<ul style="list-style-type: none"> <li>유연한 아키텍처를 채택하여 단일 API를 통해 데스크 톱 서버 또는 휴대기기에 장착된 하나 이상의 CPU 또는 GPU에 연산을 배포 가능</li> </ul>		
<b>대분류</b>	<ul style="list-style-type: none"> <li>기타</li> </ul>	<b>소분류</b>	<ul style="list-style-type: none"> <li>AI</li> </ul>
<b>라이선스 형태</b>	<ul style="list-style-type: none"> <li>Apache 2.0</li> </ul>	<b>사전설치 솔루션</b>	<ul style="list-style-type: none"> <li>파이썬</li> </ul>
<b>실행 하드웨어</b>	<ul style="list-style-type: none"> <li>X86 , ARM 호환 프로세서,</li> <li>램 1GB 이상의 하드웨어</li> <li>500MB 이상의 디스크공간(최소 256MB)</li> </ul>	<b>버전</b>	<ul style="list-style-type: none"> <li>1.11 (2018년 10월 기준)</li> </ul>
<b>특징</b>	<ul style="list-style-type: none"> <li>오픈 소스 운영체제(라이선스 비용이 필요하지 않음)</li> <li>널리 알려진 인공지능 개발 및 운용 프레임 워크</li> </ul>		
<b>보안취약점</b>	<ul style="list-style-type: none"> <li>취약점 ID : CVE-2018-7577</li> <li>취약점 설명 : TensorFlow 체크 포인트 메타 파일은 Google의 <a href="https://github.com/google/snappy">https://github.com/google/snappy</a> 압축 / 압축 해제 라이브러리를 사용, TensorFlow에서 사용하는 스냅의 버전에는 memcpy-param-overlap 문제 발생</li> <li>대응방안 : 1.1.0 이후 패치된 버전 사용</li> <li>참고 경로 : <a href="https://github.com/tensorflow/tensorflow/blob/master/tensorflow/security/advisory/tfsa-2018-005.md">https://github.com/tensorflow/tensorflow/blob/master/tensorflow/security/advisory/tfsa-2018-005.md</a></li> </ul>		
<b>개발회사/커뮤니티</b>	<ul style="list-style-type: none"> <li>Google Brain Lab / <a href="https://ai.google/research/teams/brain">https://ai.google/research/teams/brain</a></li> </ul>		
<b>공식 홈페이지</b>	<ul style="list-style-type: none"> <li><a href="https://tensorflow.org">https://tensorflow.org</a></li> </ul>		



## 2. 기능요약



- TensorFlow는 구글 브레인 랩에서 만든 오픈 소스 인공지능 개발 프레임 워크임, 최초에 DistBelief로 2011년 내부에서 개발되어 구글 검색, 구글 번역, 구글 사진 서비스로 사용됨
- 2015년 11월 구글 브레인 랩(Google Brain Lab)의 제프 딘(Dean)은 DistBelief를 오픈 소스로 이름을 TensorFlow로 바꾸어 공개함

### 1<sup>st</sup> Generation : *DistBelief*



- *Dean et al. 2011*
- *Major Output Products*
  - *Inception (Image Categorization)*
  - *Google Search*
  - *Google Translate*
  - *Google Photos*

### 2<sup>nd</sup> Generation : *TensorFlow*

- *Dean et al. 2015 (November, 1<sup>st</sup>)*
- *Most of DistBelief users at Google have already switched to TensorFlow*

\* Source: Oriol Vinyals – Research Scientist at Google Brain



# 2. 기능요약



- TensorFlow의 주요 기능

주요기능	지원여부
32 / 64비트 OS	32 / 64 지원 ( x86 / x86_64 )
Pip 패키지	지원
Anaconda 설치	지원
Source 설치	지원
CPU 연산	지원 (Intel MKL 지원)
GPU 연산	지원 (Nvidia CUDA, AMD ROCm 지원)
TPU 연산	지원 (1.8이상)



# 3. 실행환경



- 하드웨어 제약이 거의 없음

Build Type	Artifacts
리눅스 CPU	pypi
리눅스 GPU	pypi
리눅스 XLA	source
MACOS	pypi
Windows CPU	pypi
Windows GPU	pypi
안드로이드	TensorFlow 1.12.0 지원
Raspberry Pi 0 and 1	Py2 Py3
Raspberry Pi 2 and 3	Py2 Py3
IBM s390x	TBA
IBM ppc64le CPU	TBA
IBM ppc64le GPU	Release
리눅스 CPU with Intel MKL DNN 파이썬2	1.11.0 py2.7
리눅스 CPU with Intel MKL DNN 파이썬3	1.11.0 py3.4, 1.11.0 py3.5, 1.11.0 py3.6

※ Official Builds



# 4. 설치 및 실행

## 세부 목차

1. 설치 준비
2. 설치 환경 준비
3. 시스템 부팅 및 설치 진입
4. 설치
5. 실행



# 4. 설치 및 실행



## 4.1 설치 준비(1/2)

- 우분투 저장소를 최신의 것으로 업데이트

```
[ubuntu@ubuntu:~]$ sudo apt-get update
[[sudo] password for ubuntu:
Hit:1 http://kr.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://kr.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://kr.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu xenial-security InRelease
Reading package lists... Done
ubuntu@ubuntu:~$ █
```

※ 사전 준비 사항 : 노트북 or 데스크탑, 우분투 리눅스, 네트워크





# 4. 설치 및 실행



## 4.1 설치 준비(2/2)

- 리눅스 기준 배포판(우분투 16.04LTS)으로 설치
- 파이썬과 TensorFlow에 필요한 패키지들을 설치

```
[ubuntu@ubuntu:~]$ sudo apt-get install -y \  
> git \  
> gcc \  
> make \  
> openssl \  
> libssl-dev \  
> libbz2-dev \  
> libreadline-dev \  
> libsqlite3-dev
```



# 4. 설치 및 실행



## 4.2 설치 환경 준비(1/6)

- 파이썬 개발 환경을 준비
- Github에서 파이썬 개발 환경(pyenv) 를 받아서 설치

```
ubuntu@ubuntu:~$ git clone https://github.com/yyuu/pyenv.git ~/.pyenv
'/home/ubuntu/.pyenv'에 복제합니다...
remote: Counting objects: 15972, done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 15972 (delta 2), reused 12 (delta 2), pack-reused 15954
오브젝트를 받는 중: 100% (15972/15972), 3.05 MiB | 603.00 KiB/s, 완료.
델타를 알아내는 중: 100% (10870/10870), 완료.
연결을 확인하는 중입니다... 완료.
ubuntu@ubuntu:~$ echo 'export PYENV_ROOT="$HOME/.pyenv" ' >> ~/.bashrc
ubuntu@ubuntu:~$ echo 'export PATH="$PYENV_ROOT/bin:$PATH" ' >> ~/.bashrc
ubuntu@ubuntu:~$ echo 'eval "$(pyenv init -)" ' >> ~/.bashrc
ubuntu@ubuntu:~$ source ~/.bashrc
ubuntu@ubuntu:~$ exec $SHELL -l
ubuntu@ubuntu:~$ █
```



# 4. 설치 및 실행



## 4.2 설치 환경 준비(2/6)

- Github에서 파이썬 개발 환경(pyenv)에 필요한 가상 파이썬 환경을 받아서 설치

```
[ubuntu@ubuntu:~]$ git clone https://github.com/yyuu/pyenv-virtualenv.git \
> ~/.pyenv/plugins/pyenv-virtualenv
'/home/ubuntu/.pyenv/plugins/pyenv-virtualenv'에 복제합니다...
remote: Counting objects: 1943, done.
remote: Total 1943 (delta 0), reused 0 (delta 0), pack-reused 1943
오브젝트를 받는 중: 100% (1943/1943), 547.68 KiB | 390.00 KiB/s, 완료.
델타를 알아내는 중: 100% (1332/1332), 완료.
연결을 확인하는 중입니다... 완료.
ubuntu@ubuntu:~$ echo 'eval "$(pyenv virtualenv-init -)" ' >> ~/.bashrc
ubuntu@ubuntu:~$ exec $SHELL -l
```



# 4. 설치 및 실행



## 4.2 설치 환경 준비(3/6)

- 파이썬 개발 환경에 필요한 파이썬 3.6.x 버전을 설치

```
[ubuntu@ubuntu:~]$ pyenv install 3.6.6
Downloading Python-3.6.6.tar.xz...
-> https://www.python.org/ftp/python/3.6.6/Python-3.6.6.tar.xz
Installing Python-3.6.6...
Installed Python-3.6.6 to /home/ubuntu/.pyenv/versions/3.6.6
```



# 4. 설치 및 실행



## 4.2 설치 환경 준비(4/6)

- 파이썬 개발 환경에 필요한 파이썬 3.6.x 버전을 tensorflow라고 가상 파이썬 환경으로 설정

```
[ubuntu@ubuntu:~]$ pyenv virtualenv 3.6.6 tensorflow ]
Looking in links: /tmp/tmp_uk979jz
Requirement already satisfied: setuptools in /home/ubuntu/.pyenv/versions/3.6.6/
envs/tensorflow/lib/python3.6/site-packages (39.0.1)
Requirement already satisfied: pip in /home/ubuntu/.pyenv/versions/3.6.6/envs/te
nsorflow/lib/python3.6/site-packages (10.0.1)
ubuntu@ubuntu:~$
```



# 4. 설치 및 실행



## 4.2 설치 환경 준비(5/6)

- 실습에 사용할 폴더를 만듦 (~/.tensorflow)  
\$ mkdir -p ~/.tensorflow  
\$ cd ~/.tensorflow
- Pyenv를 사용해서 실습 폴더에 사용할 파이썬 버전을 고정  
\$ pyenv local tensorflow

```
[ubuntu@ubuntu:~]$ mkdir -p ~/.tensorflow  
[ubuntu@ubuntu:~]$ cd ~/.tensorflow/  
[ubuntu@ubuntu:~/.tensorflow]$ pyenv local tensorflow  
(tensorflow) ubuntu@ubuntu:~/.tensorflow$
```



# 4. 설치 및 실행



## 4.2 설치 환경 준비(6/6)

- Pypi 를 최신의 것으로 업데이트

```
(tensorflow) ubuntu@ubuntu:~/tensorflow$ pip install -U pip setuptools  
Collecting pip  
  Downloading https://files.pythonhosted.org/packages/c2/d7/90f34cb0d83a6c5631cf71dfe64cc1054598c843a92b400e55675cc2ac37/pip-18.1-py2.py3-none-any.whl (1.3MB)  
    100% |████████████████████████████████████████| 1.3MB 12.9MB/s  
Collecting setuptools  
  Downloading https://files.pythonhosted.org/packages/82/a1/ba6fb41367b375f5cb653d1317d8ca263c636cff6566e2da1b0da716069d/setuptools-40.5.0-py2.py3-none-any.whl (569kB)  
    100% |████████████████████████████████████████| 573kB 1.6MB/s  
Installing collected packages: pip, setuptools  
  Found existing installation: pip 10.0.1  
    Uninstalling pip-10.0.1:  
      Successfully uninstalled pip-10.0.1  
  Found existing installation: setuptools 39.0.1  
    Uninstalling setuptools-39.0.1:  
      Successfully uninstalled setuptools-39.0.1  
Successfully installed pip-18.1 setuptools-40.5.0  
(tensorflow) ubuntu@ubuntu:~/tensorflow$
```



# 4. 설치 및 실행



## 4.3 설치

- TensorFlow 를 pip로 설치

```
(tensorflow) ubuntu@ubuntu:~/tensorflow$ pip install tensorflow
Collecting tensorflow
  Downloading https://files.pythonhosted.org/packages/ce/d5/38cd4543401708e64c9e
e6afa664b936860f4630dd93a49ab863f9998cd2/tensorflow-1.11.0-cp36-cp36m-manylinux1
_x86_64.whl (63.0MB)
  100% |████████████████████████████████████████████████████████████████████████████████| 63.0MB 798kB/s
Collecting tensorboard<1.12.0,>=1.11.0 (from tensorflow)
  Downloading https://files.pythonhosted.org/packages/9b/2f/4d788919b1feef04624d
63ed6ea45a49d1d1c834199ec50716edb5d310f4/tensorboard-1.11.0-py3-none-any.whl (3.
0MB)
  100% |████████████████████████████████████████████████████████████████████████████████| 3.0MB 3.3MB/s
Collecting six>=1.10.0 (from tensorflow)
  Downloading https://files.pythonhosted.org/packages/67/4b/141a581104b1f6397bfa
78ac9d43d8ad29a7ca43ea90a2d863fe3056e86a/six-1.11.0-py2.py3-none-any.whl
Collecting grpcio>=1.8.6 (from tensorflow)
  Downloading https://files.pythonhosted.org/packages/c3/4c/0a7c55764ac3013ca7a5
e9638ee7b161488c0611afc2be465452987a3ccc/grpcio-1.16.0-cp36-cp36m-manylinux1_x86
_64.whl (9.7MB)
  100% |████████████████████████████████████████████████████████████████████████████████| 9.7MB 4.5MB/s
```





# 4. 설치 및 실행



## 4.4 실행

- TensorFlow 를 파이썬 콘솔에서 실행

```
[(tensorflow) ubuntu@ubuntu:~/tensorflow$ python ]
Python 3.6.6 (default, Oct 30 2018, 17:12:54)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
[>>> import tensorflow ]
[>>> print(tensorflow.__version__) ]
1.11.0
>>> █
```



# 5. 기능소개

## 세부 목차

1. 연산 정의하기
2. 세션 정의하기
3. 세션 실행하기
4. 세션 결과출력하기



# 5. 기능소개



## 5.1 연산 정의하기

- import tensorflow 로 TensorFlow 라이브러리를 불러옴
- tensorflow 를 tf로 정의
- tf.constant() 를 사용해서 x1, x2에 상수를 정의
- tf.multiply(x1,x2) 를 사용해서 x1과 x2 의 결과를 result에 대입

```
>>> import tensorflow as tf
>>> x1 = tf.constant(5)
>>> x2 = tf.constant(6)
>>> result = tf.multiply(x1,x2)
>>> print(result)
Tensor("Mul:0", shape=(), dtype=int32)
>>>
```

```
]
]
]
]
]
```



# 5. 기능소개



## 5.2 세션 정의하기

- print(result) 를 하면 출력결과가 다른 값으로 나타남  
이것은 tensorflow는 session() 을 사용해서 연산그래프를 정의하고 실행하기 때문임
- sess=tf.Session() 을 연산그래프를 정의

```
[>>> import tensorflow as tf  
[>>> x1 = tf.constant(5)  
[>>> x2 = tf.constant(6)  
[>>> result = tf.multiply(x1,x2)  
[>>> sess = tf.Session()
```



# 5. 기능소개



## 5.3 세션 실행하기

- `sess=tf.Session()` 을 연산그래프를 정의했으므로 `sess.run()` 로 세션을 실행

```
[>>> import tensorflow as tf  
[>>> x1 = tf.constant(5)  
[>>> x2 = tf.constant(6)  
[>>> result = tf.multiply(x1,x2)  
[>>> sess = tf.Session()
```

```
[>>> print(sess.run(result))  
30  
>>> █
```



# 5. 기능소개



## 5.4 세션 결과 출력하기

- sess=tf.Session() 을 연산그래프를 정의했음
- sess.run() 로 세션을 실행
- print()로 연산 결과를 출력함

```
[>>> import tensorflow as tf  
[>>> x1 = tf.constant(5)  
[>>> x2 = tf.constant(6)  
[>>> result = tf.multiply(x1,x2)  
[>>> sess = tf.Session()
```

```
[>>> print(sess.run(result))  
30  
>>> █
```



# 6. 활용예제

## 세부 목차



1. 숫자 인식하기
2. 숫자 인식을 위한 데이터 셋을 불러오자.
3. 학습 모델을 정의하자.
4. 모델을 학습하자.
5. 학습 모델을 평가하자.



# 6. 활용예제



## 6.1 숫자 인식하기

- Tensorflow 1.11 이상에서는 keras 라이브러리를 포함
- keras 를 사용해서 숫자 인식
- 다음의 코드를 파이썬 콘솔에 붙여 넣고 실행

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation=tf.nn.relu),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```





# 6. 활용예제



## 6.2 숫자 인식을 위한 데이터 셋을 불러오자.

- 숫자 인식을 위한 데이터 셋을 불러옴

```
>>> import tensorflow as tf
>>> mnist = tf.keras.datasets.mnist
>>>
>>> (x_train, y_train), (x_test, y_test) = mnist.load_data()
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dataset
s/mnist.npz
11493376/11490434 [=====] - 1s 0us/step
>>> x_train, x_test = x_train / 255.0, x_test / 255.0
```



# 6. 활용예제



## 6.3 학습 모델을 정의하자.

- 숫자 인식을 위한 학습 모델을 정의

```
>>> model = tf.keras.models.Sequential([
...     tf.keras.layers.Flatten(),
...     tf.keras.layers.Dense(512, activation=tf.nn.relu),
...     tf.keras.layers.Dropout(0.2),
...     tf.keras.layers.Dense(10, activation=tf.nn.softmax)
... ])
```

```
>>> model.compile(optimizer='adam',
...               loss='sparse_categorical_crossentropy',
...               metrics=['accuracy'])
>>>
```



# 6. 활용예제



## 6.4 모델을 학습하자.

- 숫자 인식을 데이터와 함께 학습 모델로 학습함

```
>>> model.fit(x_train, y_train, epochs=5)
Epoch 1/5
60000/60000 [=====] - 10s 166us/step - loss: 0.2008 - a
cc: 0.9410
Epoch 2/5
60000/60000 [=====] - 10s 173us/step - loss: 0.0809 - a
cc: 0.9749
Epoch 3/5
60000/60000 [=====] - 10s 173us/step - loss: 0.0530 - a
cc: 0.9837
Epoch 4/5
60000/60000 [=====] - 11s 178us/step - loss: 0.0380 - a
cc: 0.9879
Epoch 5/5
60000/60000 [=====] - 11s 190us/step - loss: 0.0278 - a
cc: 0.9907
<tensorflow.python.keras.callbacks.History object at 0x7f79840a30b8>
```



# 6. 활용예제



## 6.5 학습 모델을 평가하자.

- 숫자 인식을 위한 학습된 모델을 평가함
- 97% 정확도를 확인 가능

```
[>>> model.evaluate(x_test, y_test)
10000/10000 [=====] - 0s 31us/step
[0.07255643702821107, 0.9797]
>>>
```





**Q** TensorFlow 사용자는 어떻게 사용할 수 있을까요?

**A** TensorFlow로 데이터와 학습 모델을 개발하면 추론 모델을 저장할 수 있습니다. 추론 모델을 모바일기기, 임베디드기기 등에 내장해서 사용자들이 인공지능 서비스를 이용할 수 있습니다.

**Q** 리눅스 이외에 다른 운영체제(윈도우)도 지원하나요?

**A** 현재 TensorFlow는 기본 개발 환경이 파이썬이 동작하는 환경이라면 어떤 곳에서도 동작합니다. 윈도우즈는 물론이며, 맥오에스, 라즈베리 파이, 그리고 안드로이드에서도 TensorFlow를 활용할 수 있습니다.



# 8. 용어정리



용어	설명
TensorFlow	구글에서 만든 인공지능 프레임 워크
Keras	구글에서 만든 인공지능 프레임 워크
GPL	GNU General Public License의 약자. 라이선스의 종류중 하나. 소스의 취득, 수정, 배포, 공개가 자유로우나 최종 배포시 GPL라이선스를 따라야 함. 수정 배포 시 자체개발한 소스도 공개해야 함
Apache 2.0	아파치 소프트웨어 재단에서 자체적으로 만든 소프트웨어에 대한 라이선스 규정, 아파치 2.0 라이선스는 누구나 해당 소프트웨어에서 파생된 프로그램을 제작할 수 있으며 저작권을 양도, 전송할 수 있는 라이선스 규정을 의미
파이썬	<b>파이썬(Python)</b> 은 1991년 프로그래머인 귀도 반 로섬(Guido van Rossum)이 발표한 고급 프로그래밍 언어로, 플랫폼 독립적이며인터프리터식, 객체 지향적, 동적 타이핑(dynamically typed) 대화형 언어
pyenv	파이썬을 개발을 도와주는 소프트웨어로 가상 파이썬 환경을 지원
우분투	리눅스 운영체제로 레드햇과 더불어 가장 많이 사용



# Open Source Software Installation & Application Guide



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.