

# Install from Anaconda to OpenAI Gym

이동민

삼성전자 서울대 공동연구소  
Jul 15, 2019

# Outline

---

- Anaconda
- Pytorch
- TensorboardX
- OpenAI Gym

# Anaconda

---



- 다양한 수학/과학, 시각화 라이브러리부터 머신러닝 라이브러리까지 쉽게 설치 가능하도록 지원하는 플랫폼
- 하나의 독립적인 환경을 만들어 라이브러리, 의존성 등을 쉽게 관리할 수 있음
- 윈도우, 맥, 리눅스 환경에서 모두 사용 가능
- <https://www.anaconda.com>

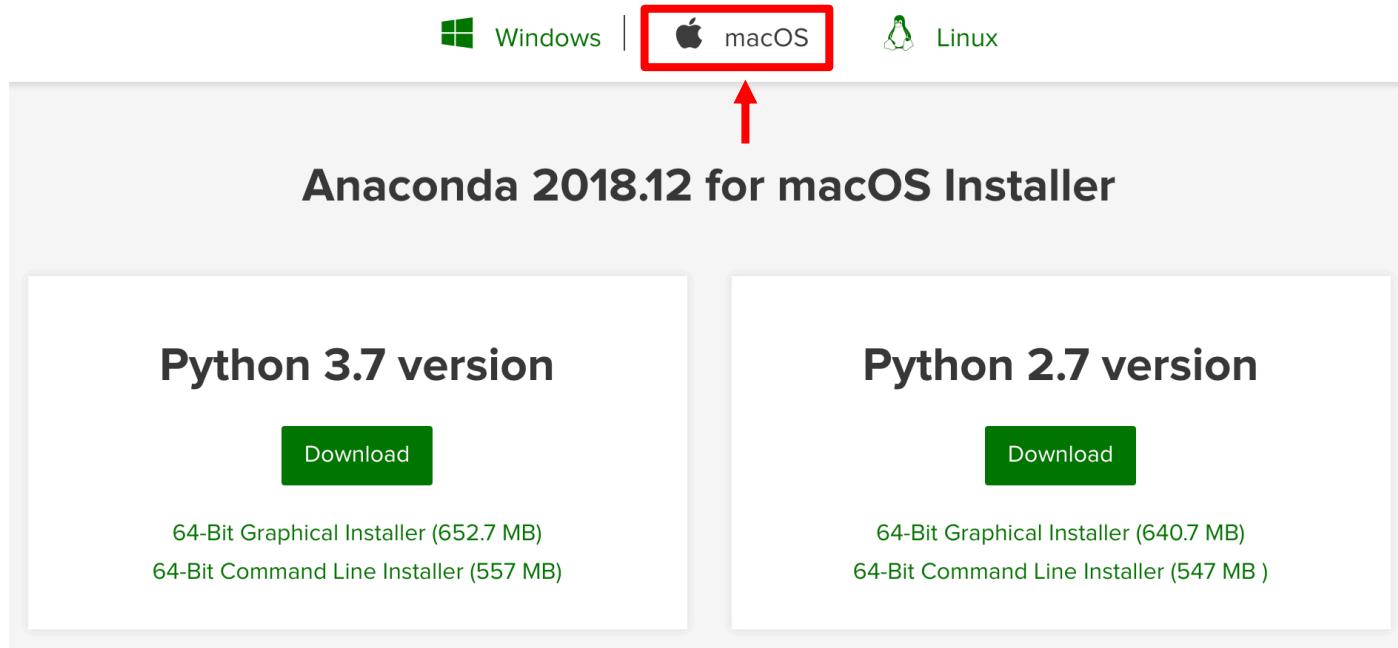
# Anaconda

- Anaconda 다운로드 (운영체제 및 CPU에 맞춰서 다운로드)

The image shows the official website for Anaconda. At the top, there is a navigation bar with links for Products, Why Anaconda?, Solutions, Resources, Company, Download, and a search icon. Below the navigation is a large banner for "AnacondaCON 2019" with a photo of two people smiling. The banner includes the date "April 3-5 | Austin, TX" and buttons for "Register Now" and "Learn More". A green bar at the bottom of the banner contains the text "Latest news: AnacondaCON 2019 Speaker Lineup and Agenda Released! Learn More>>". Below the banner, the text "The Enterprise Data Science Platform for..." is displayed. Three circular icons represent different user groups: "Data Scientists" (a magnifying glass over a chart), "IT Professionals" (a laptop with code), and "Business Leaders" (two hands shaking). The background features a light gray geometric pattern.

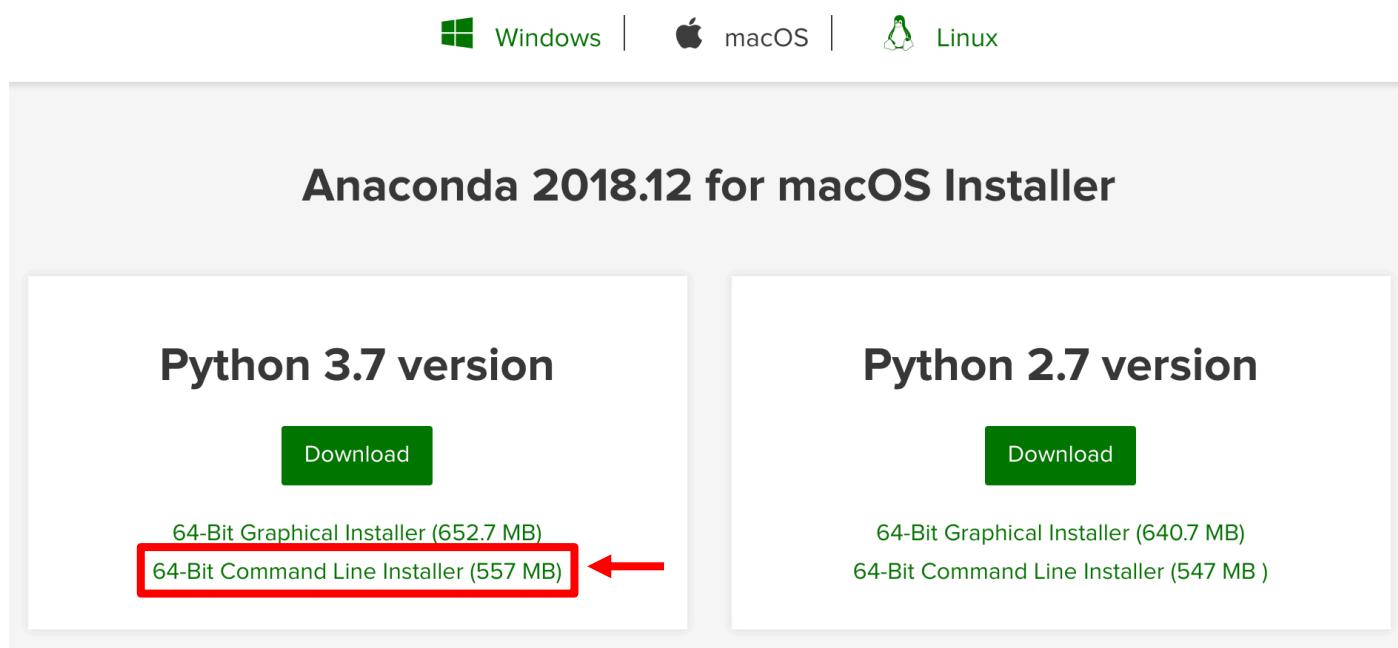
# Anaconda

- Anaconda 다운로드 (운영체제 및 CPU에 맞춰서 다운로드)



# Anaconda

- Anaconda 다운로드 (운영체제 및 CPU에 맞춰서 다운로드)



# Anaconda

---

- bash를 이용하여 Command Line Installer 다운로드 받기
  - bash ~/Downloads/Anaconda3-2018.12-MacOSX-x86\_64.sh
  - License terms 나오면 yes or no 중에서 yes
  - VSCode 설치 나오면 yes or no 중에서 yes
- 만약 터미널로 bash가 아닌 zsh를 사용중이라면, Anaconda의 bin 디렉토리를 PATH에 추가하기
  - vim .zshrc
  - export PATH="/Users/username/anaconda3/bin:\$PATH" 추가
    - :wq로 저장하고 나오기
  - source ~/.zshrc 로 적용하기



# Anaconda

- Anaconda version 확인
  - conda --version
- 새로운 환경 생성
  - conda create -n env\_name python=3.6
  - y/n 나오면 y

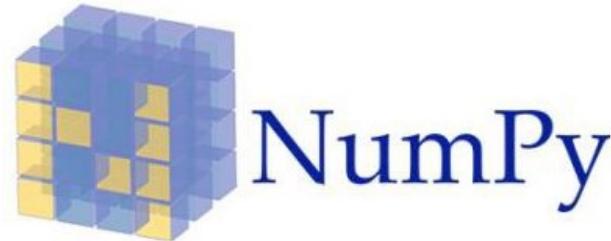
새로 만들 환경의 이름  
자유롭게 설정 가능

새로 만들 환경의 파이썬 버전  
실습에서는 3.6 사용
- 가상환경 목록보기
  - conda env list
- 가상환경 활성화 / 비활성화 하기
  - 활성화 - conda activate env\_name
  - 비활성화 - conda deactivate



**CORE**  
Control + Optimization Research Lab

# Numpy



NumPy

- Numerical Python
- 파이썬의 고성능 과학 계산용 패키지
- Vector와 Matrix와 같은 Array 연산의 표준
- Anaconda를 설치하면 자동적으로 설치되어 있음



**CORE**  
Control + Optimization Research Lab

# Pytorch

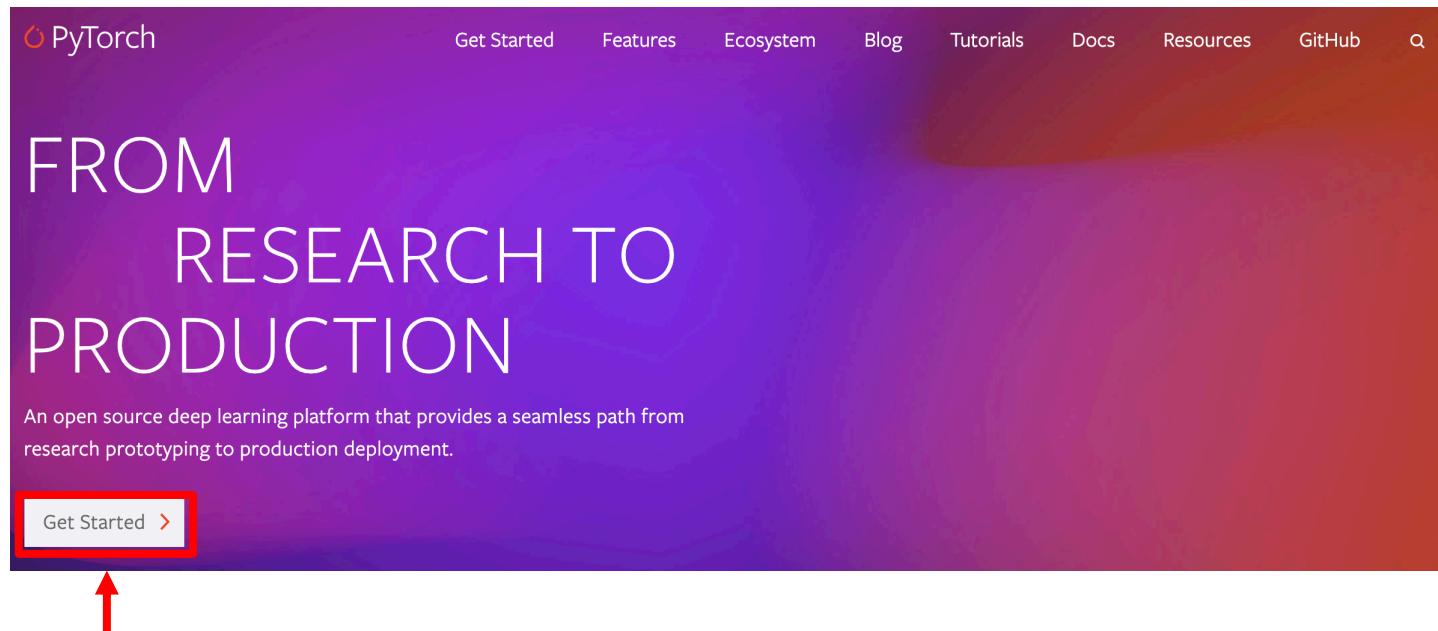
---



- Numpy를 대체하고 GPU를 쉽게 사용할 수 있음
- 최고의 유연성과 속도를 제공하는 딥러닝 연구 플랫폼
- 이해와 디버깅이 쉽고 직관적이며, 간결한 코드(Pythonic code)로 구성
- Define-by-Run 방식을 기반으로 한 실시간 결과값을 시각화
- 파이썬 라이브러리(Numpy, Matplotlib 등)와 높은 호환성을 가짐

# Pytorch

- Pytorch 설치
  - <https://pytorch.org>



# Pytorch

- Pytorch 설치
  - <https://pytorch.org>

## START LOCALLY

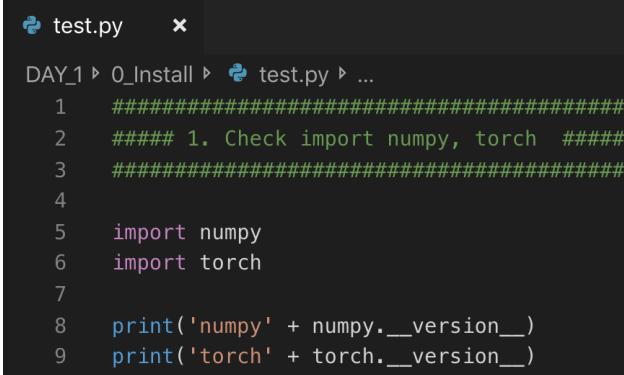
Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch 1.0. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.0 builds that are generated nightly. Please ensure that you have met the prerequisites below (e.g., numpy), depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also [install previous versions of PyTorch](#). Note that LibTorch is only available for C++.

PyTorch Build	Stable (1.0)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python 2.7	Python 3.5	Python 3.6	Python 3.7
CUDA	8.0	9.0	10.0	None
Run this Command:	<pre>conda install pytorch torchvision -c pytorch # MacOS Binaries dont support CUDA, install from source if CUDA is needed</pre>			



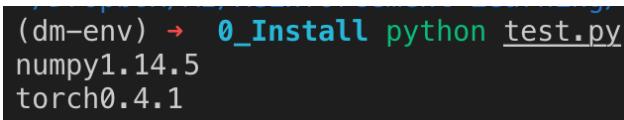
# Pytorch

- Pytorch 설치
  - conda install pytorch torchvision -c pytorch
- Numpy와 Pytorch가 제대로 설치 되었는지 확인
  - VSCode에서 DAY\_1 -> 0\_install -> test.py 선택



```
test.py *  
DAY_1 > 0_Install > test.py > ...  
1 #####  
2 ##### 1. Check import numpy, torch #####  
3 #####  
4  
5 import numpy  
6 import torch  
7  
8 print('numpy' + numpy.__version__)  
9 print('torch' + torch.__version__)
```

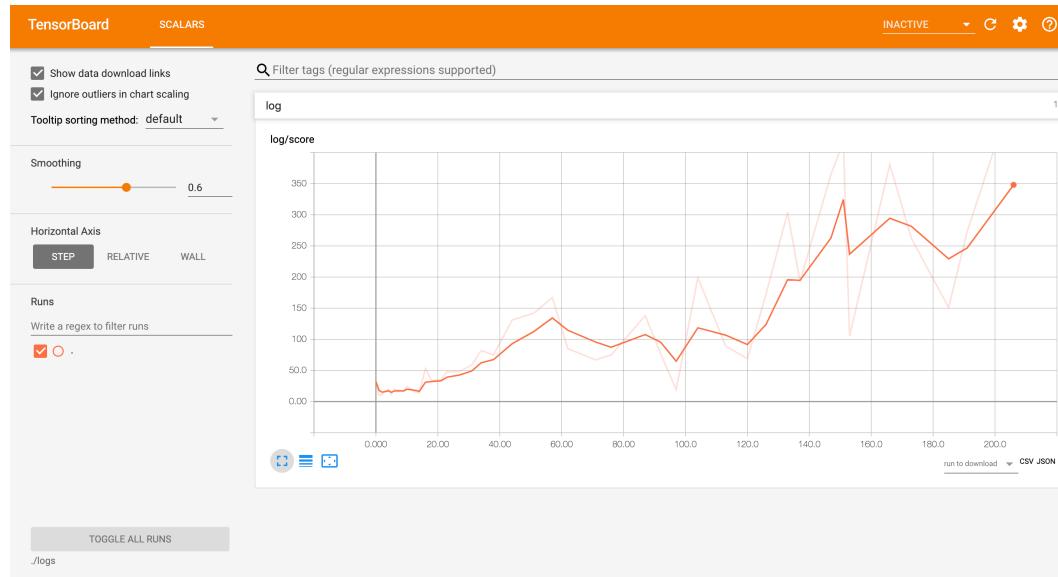
- Ctrl + shift + ~ 로 터미널창 띄우기
- conda activate env\_name
- 터미널에서 python test.py 입력하여 확인하기



```
(dm-env) → 0_Install python test.py  
numpy1.14.5  
torch0.4.1
```



# TensorboardX



- Tensorflow의 Tensorboard처럼 Tensorflow 없이도 tensor들의 flow를 시각적으로 볼 수 있도록 만든 라이브러리 → 주로 Pytorch에서 많이 사용
- 강화학습 알고리즘 구현 시, episode마다 reward가 얼마나 나오는지 보기 쉽게 알 수 있음
- <https://tensorboardx.readthedocs.io/en/latest/tutorial.html>

# TensorboardX

---

- Tensorflow + TensorboardX 설치

- pip install tensorflow
- pip install tensorboardX
- 설치가 잘 되었는지 확인

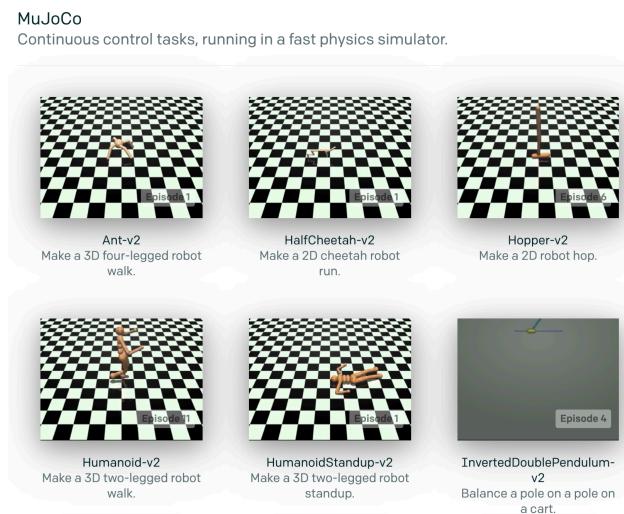
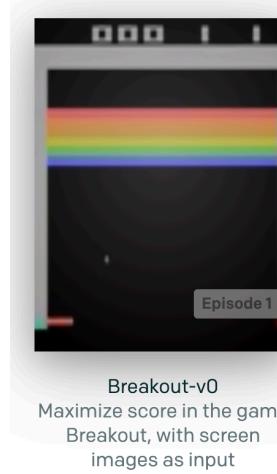
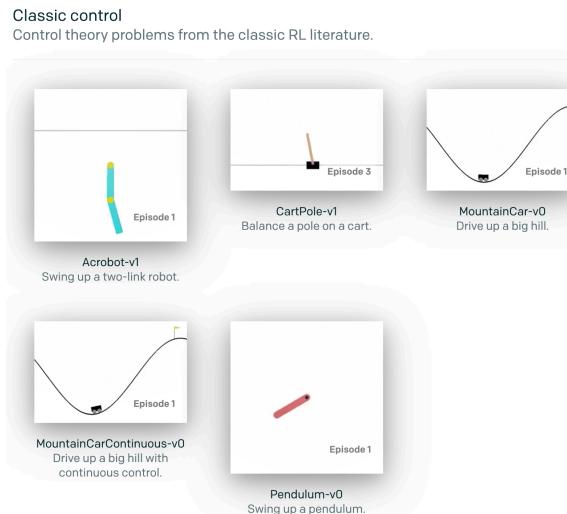
```
from tensorboardX import SummaryWriter  
writer = SummaryWriter('./logs')
```

- logs라는 디렉토리와 events.out.tfevents라는 파일이 만들어지면 성공

# OpenAI Gym



- OpenAI에서 간단한 게임들을 통해 강화학습을 테스트 할 수 있는 Gym이라는 환경을 제공
- Tensorflow나 Pytorch와 같은 수치 계산 라이브러리와도 호환됨



**CORE**  
Control + Optimization Research Lab

# OpenAI Gym

- Gym 설치
  - pip install gym
- Gym이 제대로 설치 되었는지 확인

```
import gym

env = gym.make('CartPole-v1')

for episode in range(10000):
    done = False
    state = env.reset()

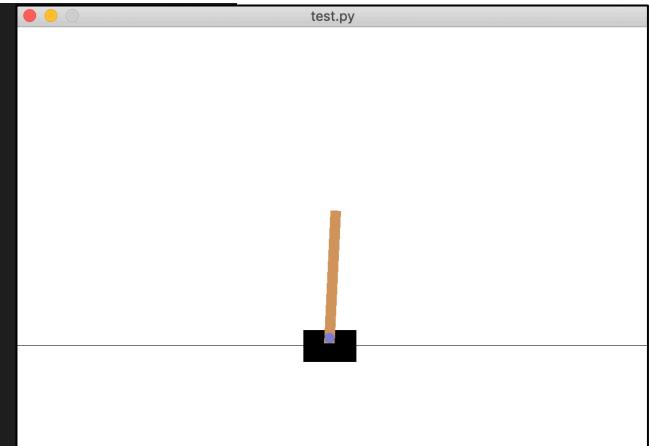
    while not done:
        env.render()

        action = env.action_space.sample()
        next_state, reward, done, _ = env.step(action)

        print('state: {} | action: {} | next_state: {} | reward: {} | done: {}'.format(
            state, action, next_state, reward, done))

        state = next_state

    if done:
        break
```



# Thank you



**CORE**  
Control + Optimization Research Lab