

FastCampus Pytorch



Ch1. Deep Learning & Pytorch

HARRY KIM

Lecture Content

- 
- 1 Deep Learning
 - 2 PyTorch
 - 3 Install

Deep Learning

Pytorch

Install

1. Deep Learning

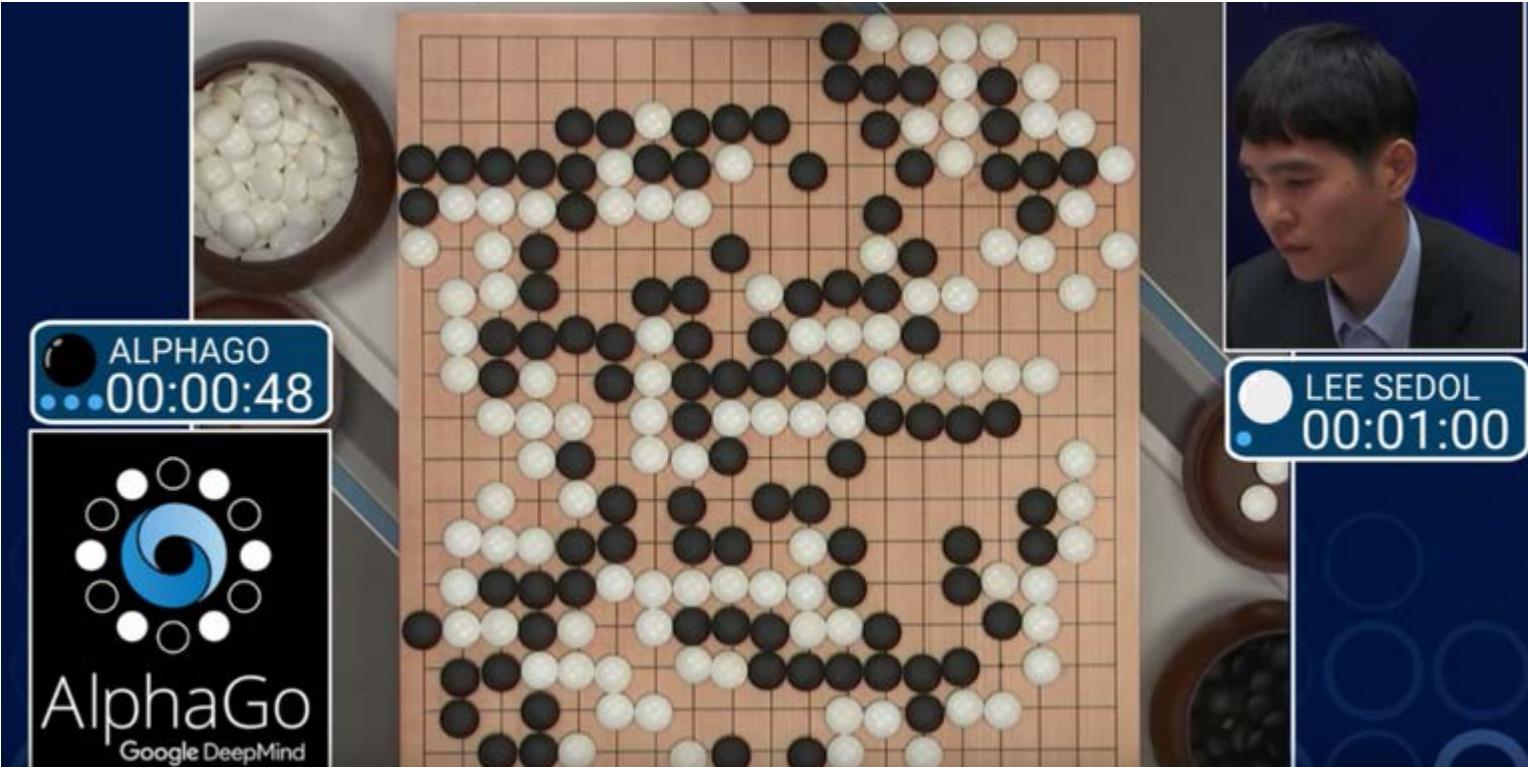
Deep Learning

Deep Learning

Pytorch

Install

- 딥러닝으로 변화하는 현재
 - Just Alpha GO..?



<https://www.popularmechanics.com/technology/a19863/googles-alphago-ai-wins-second-game-go/>

Deep Learning

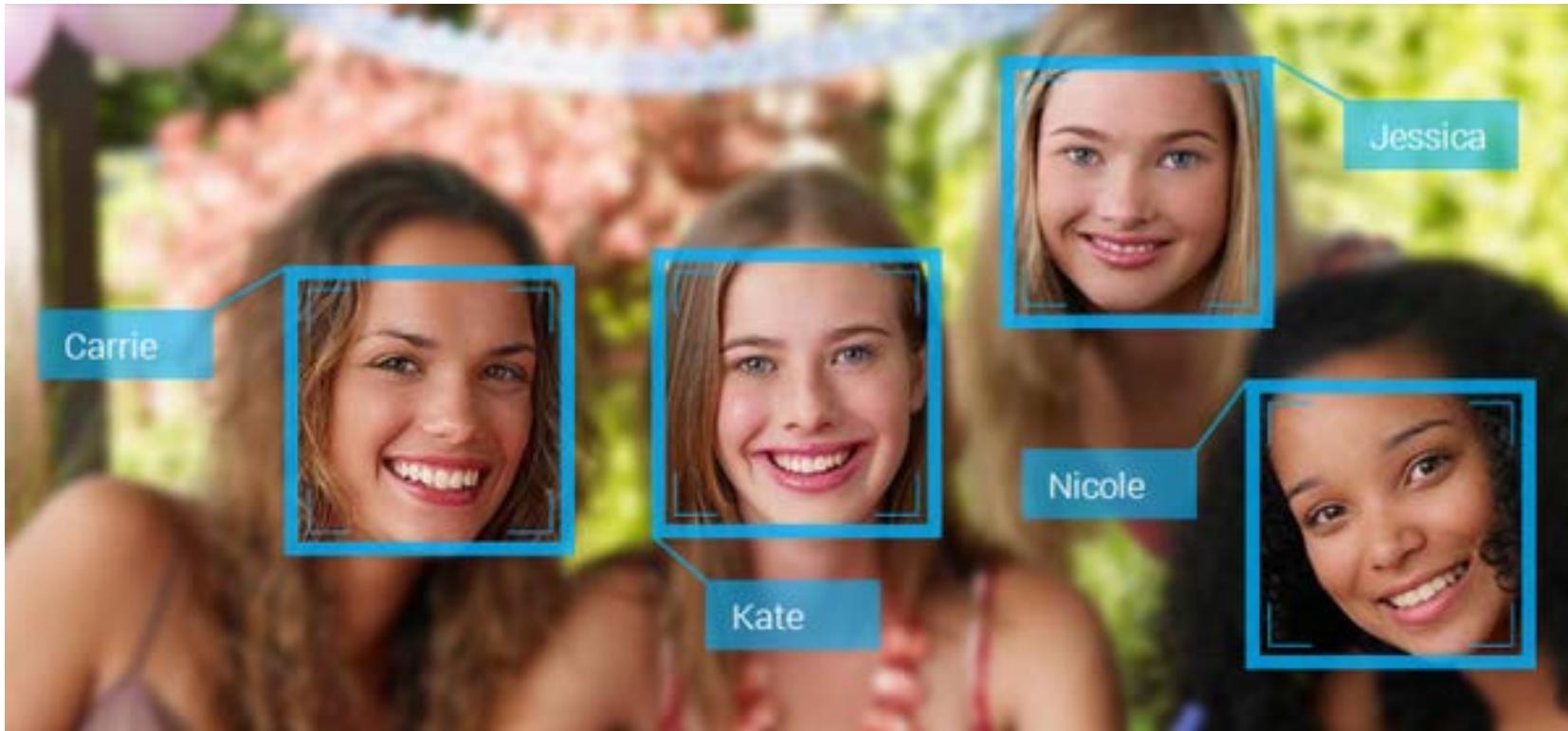
Deep Learning

Pytorch

Install

- 딥러닝으로 변화하는 현재

- 얼굴 인식(Face Detection)



<http://www.venturesquare.net/743912>

Deep Learning

Deep Learning

Pytorch

Install

- 딥러닝으로 변화하는 현재
 - 자율주행(Autonomous Driving)



<https://aitrends.com/ai-insider/sensor-fusion-self-driving-cars/>

Deep Learning

Deep Learning

Pytorch

Install

- 딥러닝으로 변화하는 현재

- 암 진단(IBM Watson)



<https://www.itnonline.com/content/deep-learning-medical-imaging-create-300-million-market-2021>



Deep Learning

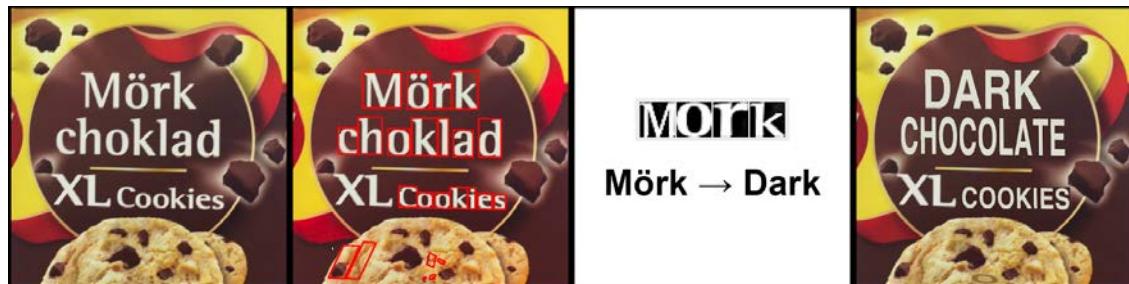
Deep Learning

Pytorch

Install

- 딥러닝으로 변화하는 현재

- 번역(Translation)



<https://ai.googleblog.com/2015/07/how-google-translate-squeezes-deep.html>

- 챗봇(Chatbot)

Google 어시스턴트

설마 오늘이 만우절은 아니죠?

죄송하지만 번지수를 잘못 찾으셨습니다

저는 Google 어시스턴트입니다 😊

하이 빅스비

시리야

프로야구 뉴스로봇

4월 28일 오후 11:23

(4.28일) NC 8:6 SK

백인식이 선발로 등판한 SK는 이태양이 나선 NC에게 6:8로 패하며 안방에서 승리를 내주었다. 경기의 승패에 결정적인 영향을 미친 키 플레이어는 손시헌이었다. 손시헌은 4회초 SK 고효준을 상대로 3점을 뽑아내어 팀의 승리에 결정적으로 기여했다. SK는 임창민을 끝까지 공략하지 못하며 안방에서 NC에 2점차 승리를 내주었다. 한편 오늘 NC에게 패한 SK는 4연패를 기록하며 수렁에 빠졌다.

좋아요 · 댓글 달기 · 공유하기

Deep Learning

Deep Learning

Pytorch

Install

- 딥러닝으로 변화하는 현재
 - 음성 생성(Voice Generation)



<https://audioclip.naver.com/channels/492/clips/2>



Deep Learning

Deep Learning

Pytorch

Install

- 딥러닝으로 변화하는 현재

- 얼굴 생성(Face Generation)



<https://www.youtube.com/watch?v=XOxxPcy5Gr4&t=175s> (1분 49초)

Deep Learning

Deep Learning

- 딥러닝으로 변화하는 현재

- 게임(Game)

Pytorch

Install



<https://www.youtube.com/watch?v=V1eYniJ0Rnk>

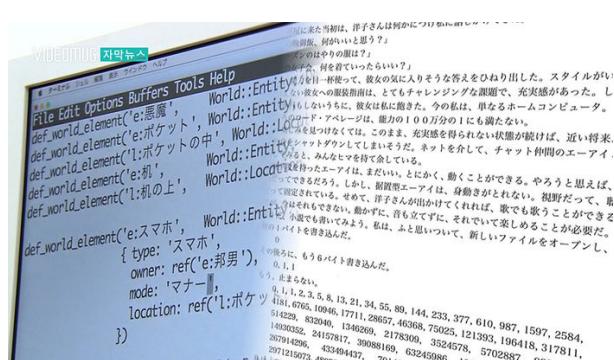
Deep Learning

Deep Learning

Pytorch

Install

- 딥러닝으로 변화하는 현재
 - 이 외에도..



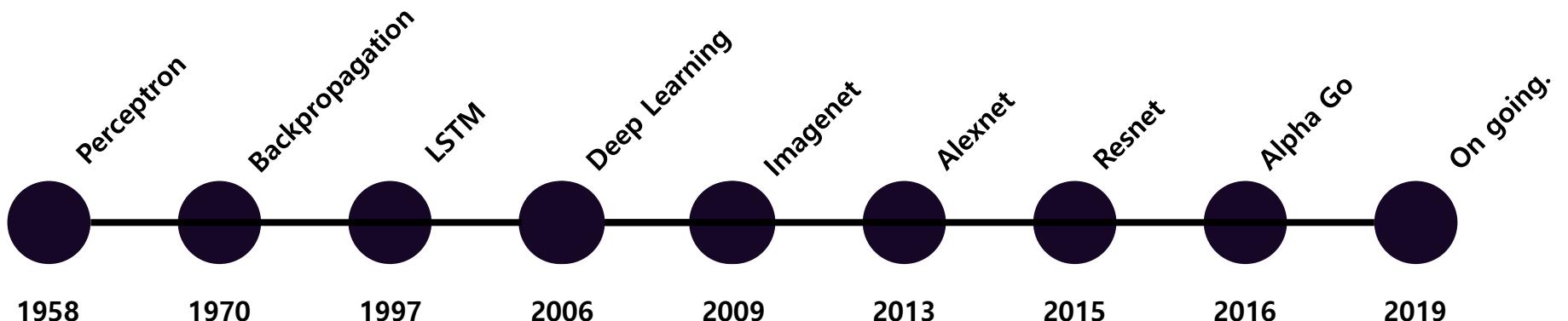
Deep Learning

Deep Learning

Pytorch

Install

- 머신러닝(Machine Learning)
 - 데이터로부터 규칙을 학습하는 프로그래밍
 - 컴퓨터가 학습하는 능력을 갖춘 것 (Arthur Samuel, 1959)
 - 작업 T에 대해 경험 E을 학습하여 성능 P를 향상시킨 것 (Tom Mitchell, 1997)
- 딥러닝(Deep Learning)
 - 현재 가장 주목받고 있는 머신러닝 알고리즘
 - 인공신경망(Artifical Neural Network) 기반의 학습 모델



Deep Learning

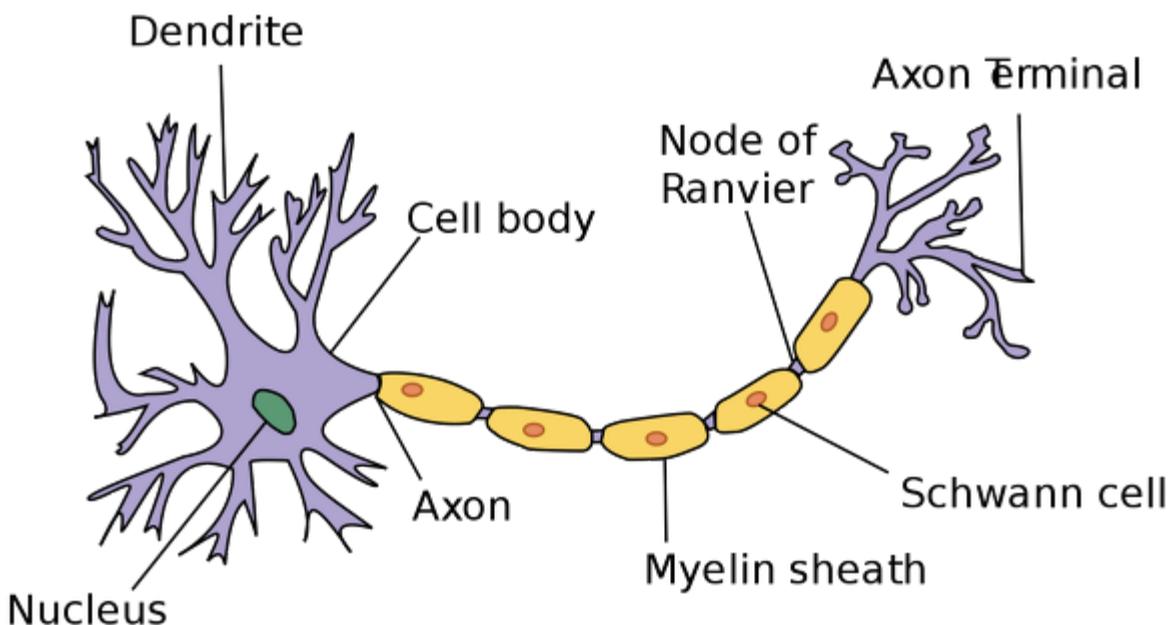
Deep Learning

Pytorch

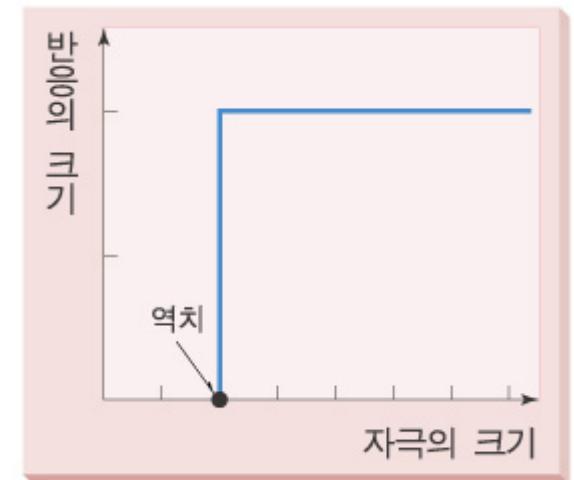
Install

▪ 퍼셉트론(Perceptron)

- 1957년 코넬 항공 연구소의 프랑크 로젠블라트에 의해 고안
- 가장 간단한 형태의 Feed-Forward 네트워크



<https://simple.wikipedia.org/wiki/Neuron>



<http://dlegongbuwarac.edupia.com>

Deep Learning

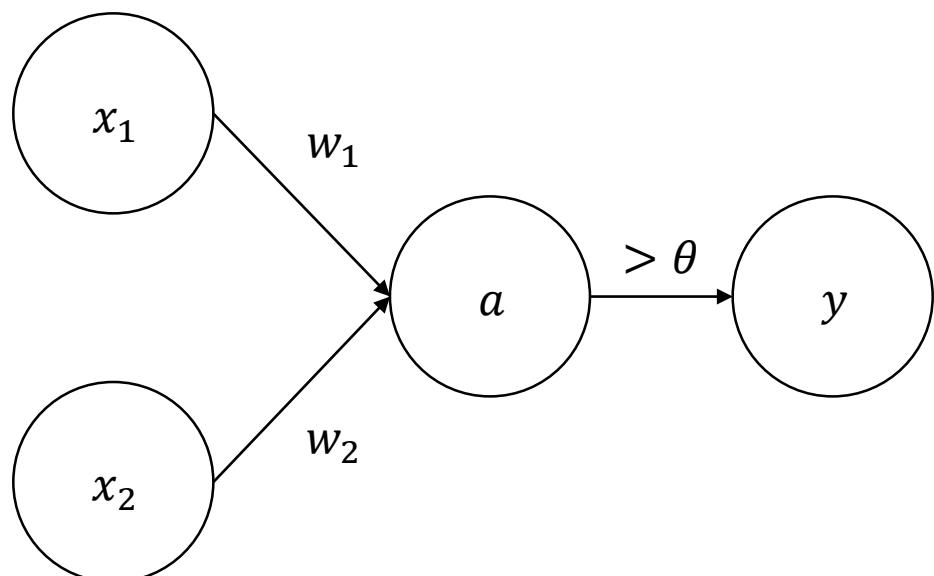
Deep Learning

Pytorch

Install

■ 퍼셉트론(Perceptron)

- 1957년 코넬 항공 연구소의 프랑크 로젠블라트에 의해 고안
- 가장 간단한 형태의 Feed-Forward 네트워크
- 다수의 신호를 입력받아 하나의 신호를 출력
 - x_1, x_2 : 입력 신호
 - w_1, w_2 : 가중치
 - y : 출력 신호
- 일정 임계값(θ)을 넘어설 때만 1을 출력
 - $y = \begin{cases} 0 & (w_1 * x_1 + w_2 * x_2 \leq \theta) \\ 1 & (w_1 * x_1 + w_2 * x_2 > \theta) \end{cases}$



Deep Learning

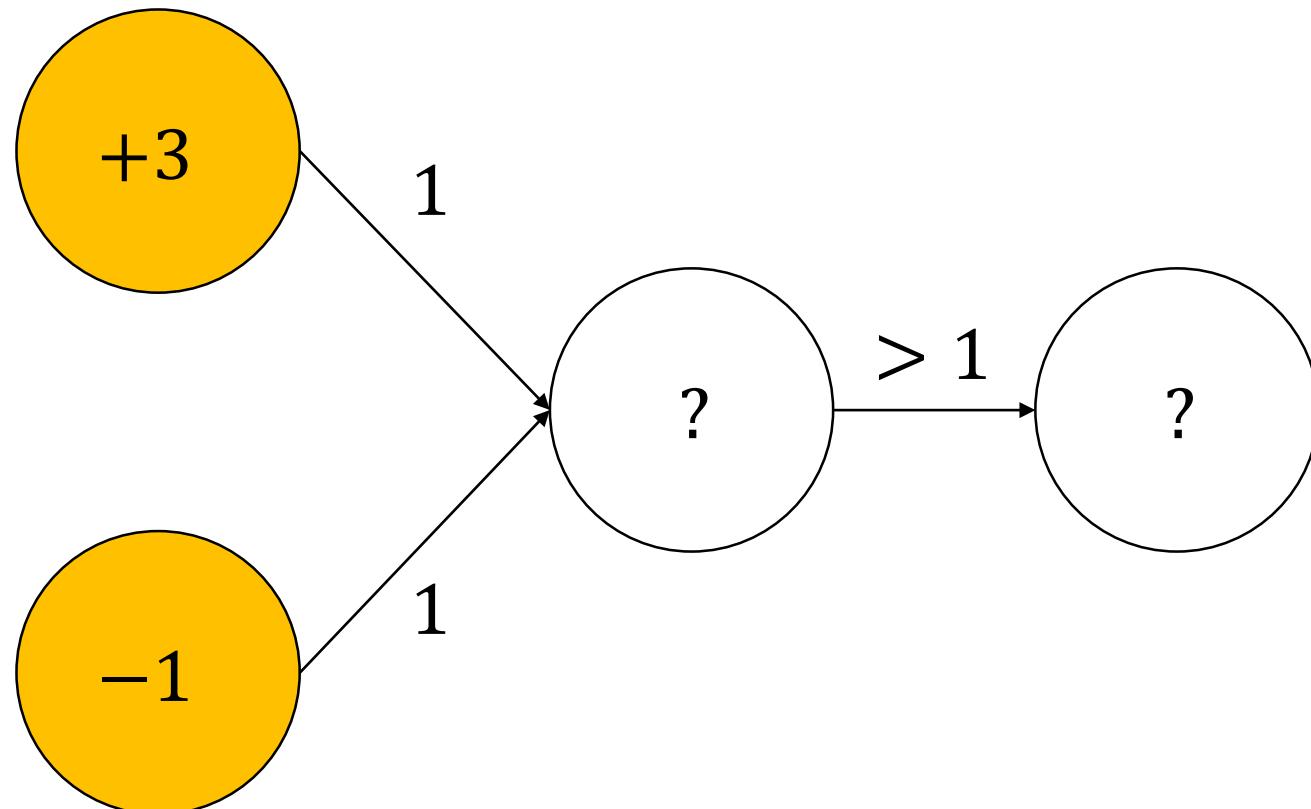
Deep Learning

- 퍼셉트론(Perceptron)

Pytorch

Install

예제1)



Deep Learning

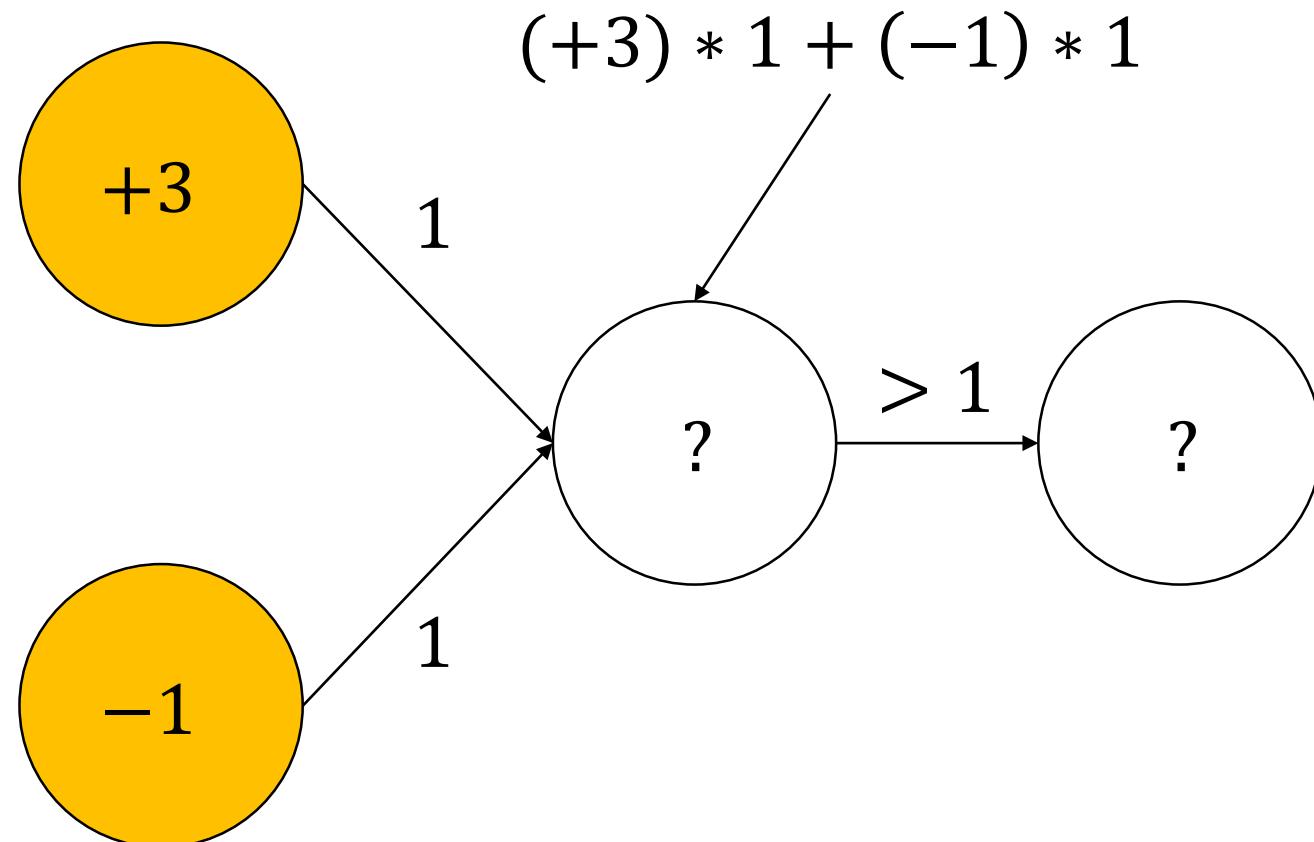
Deep Learning

- 퍼셉트론(Perceptron)

Pytorch

Install

예제1)



Deep Learning

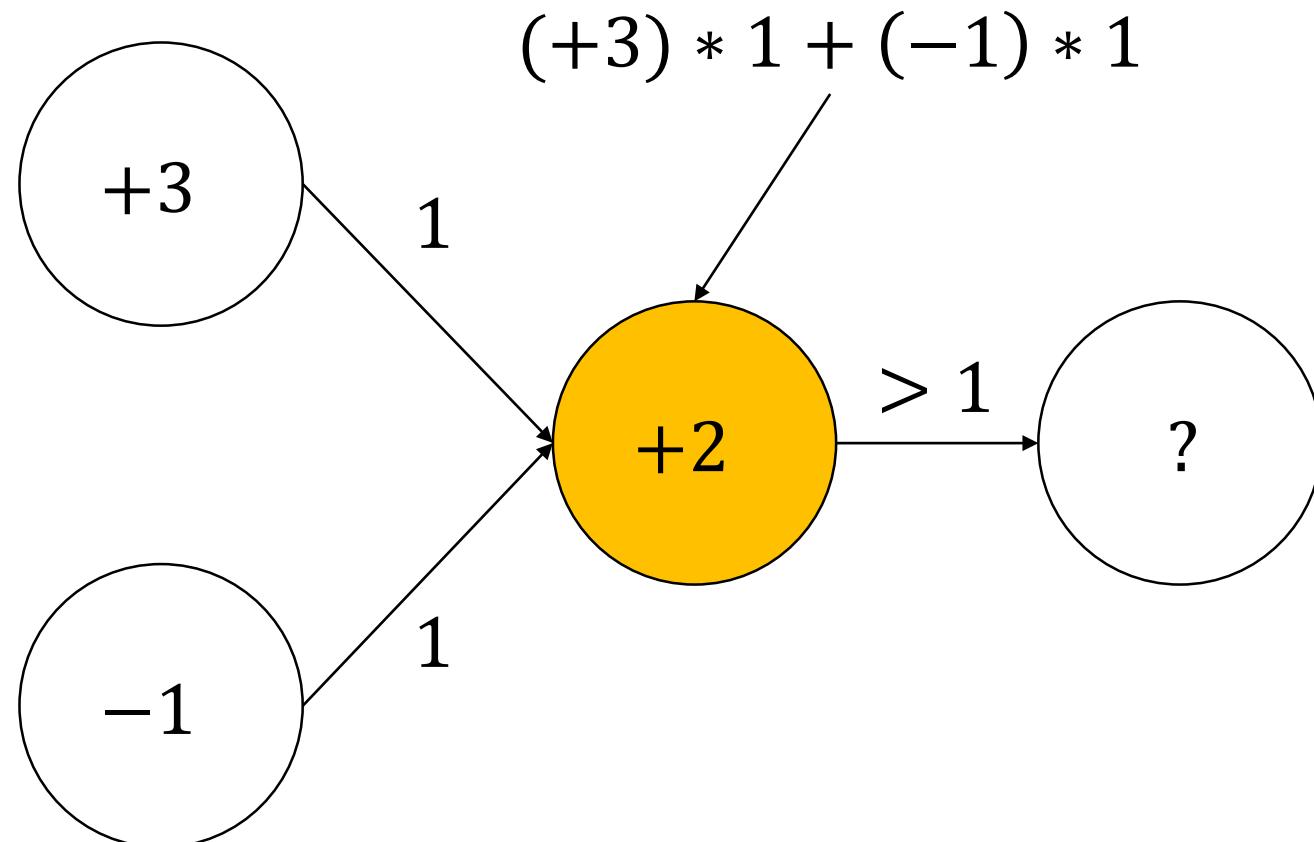
Deep Learning

- 퍼셉트론(Perceptron)

Pytorch

Install

예제1)



Deep Learning

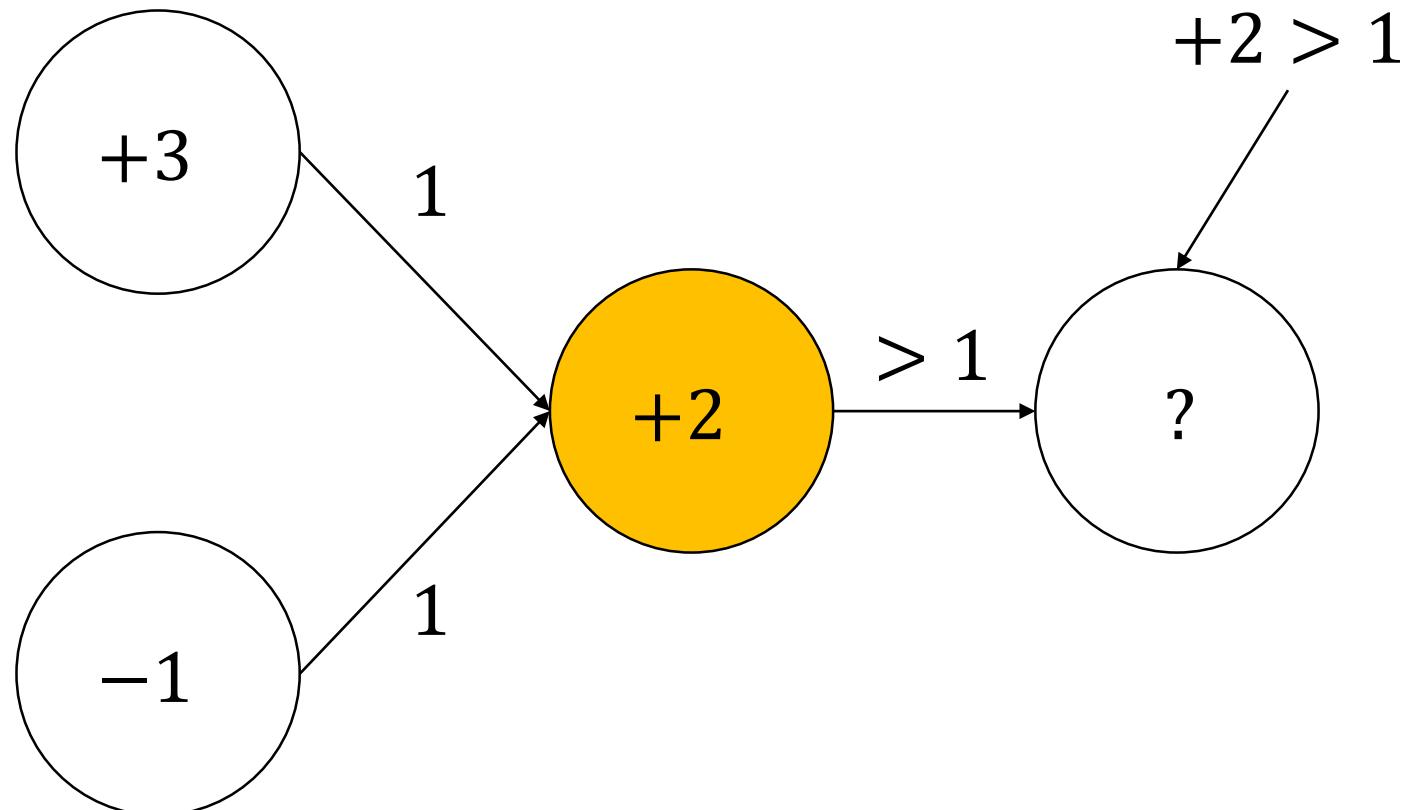
Deep Learning

- 퍼셉트론(Perceptron)

Pytorch

Install

예제1)



Deep Learning

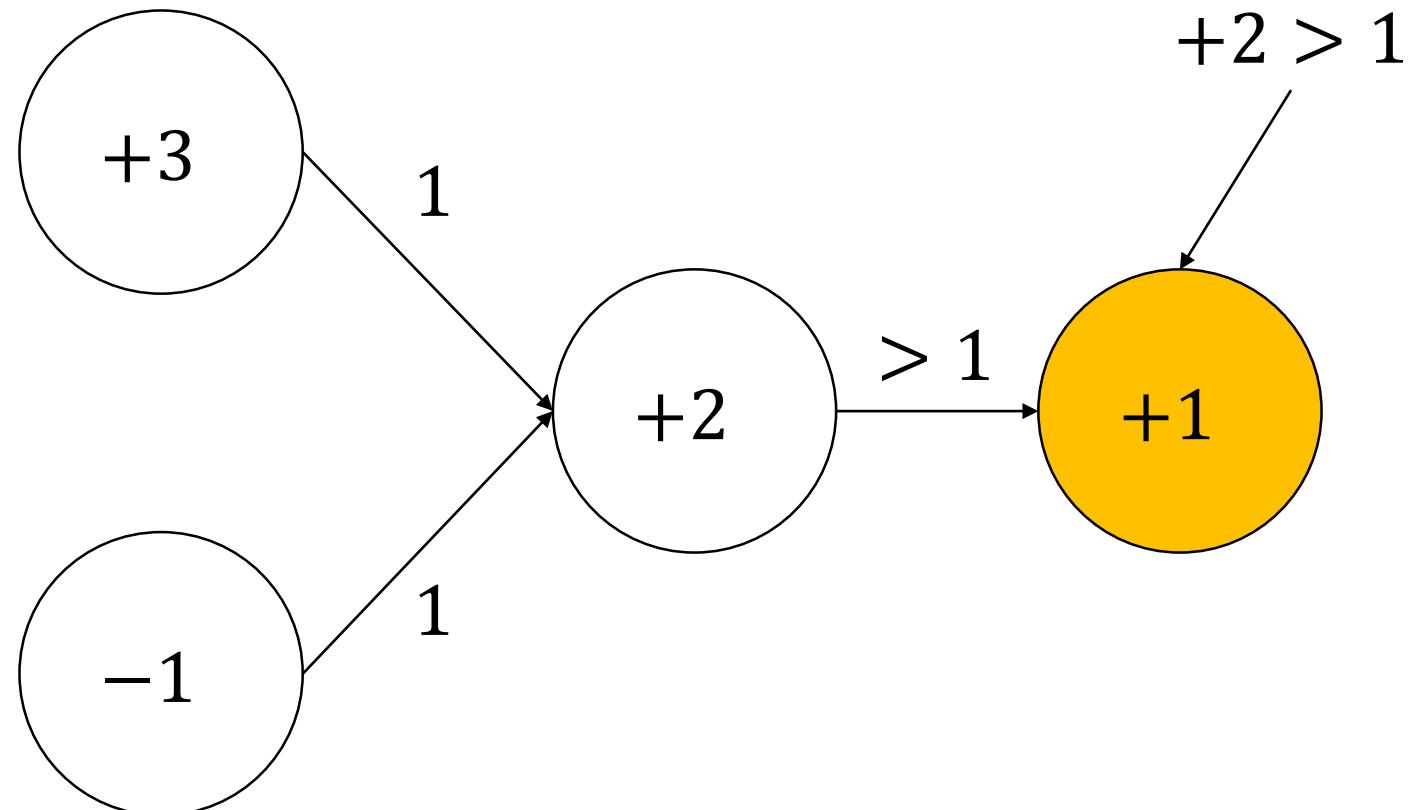
Deep Learning

- 퍼셉트론(Perceptron)

Pytorch

Install

예제1)



Deep Learning

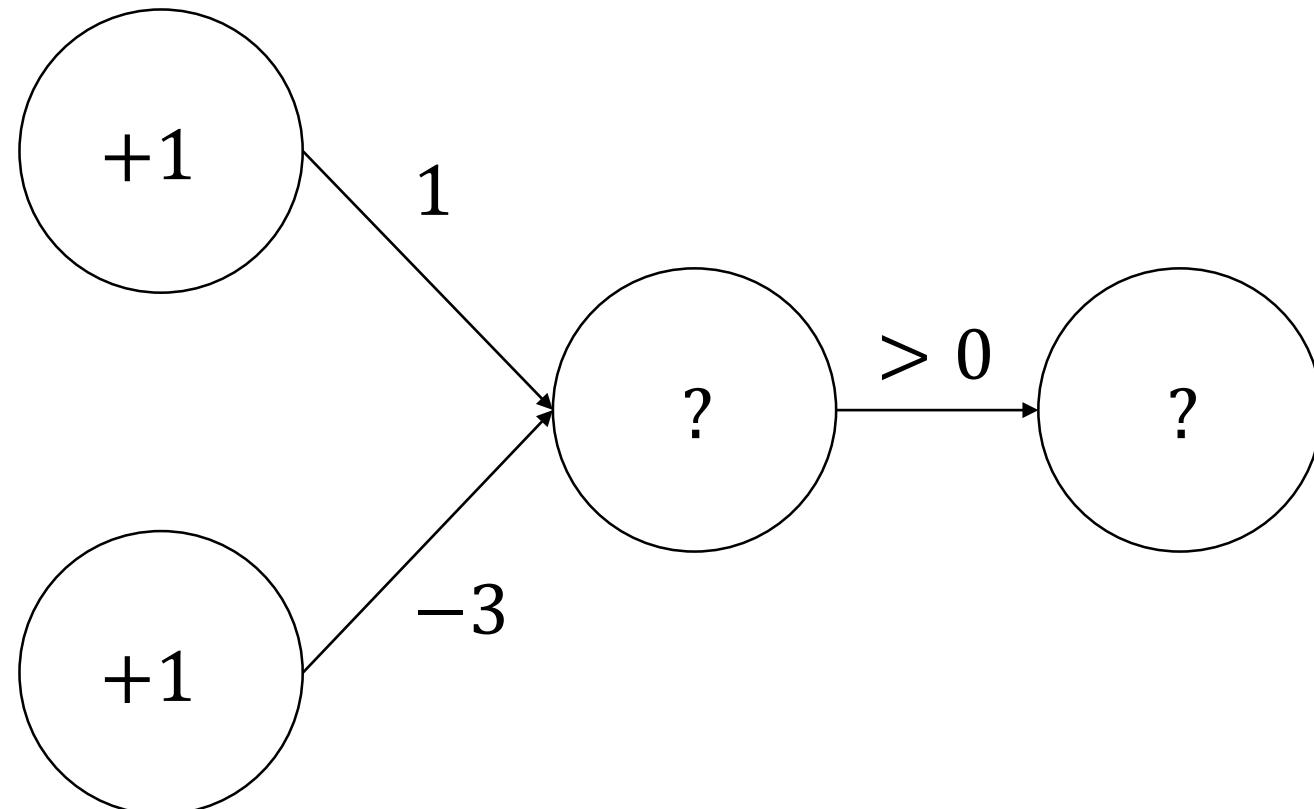
Deep Learning

- 퍼셉트론(Perceptron)

Pytorch

Install

예제2)



Deep Learning

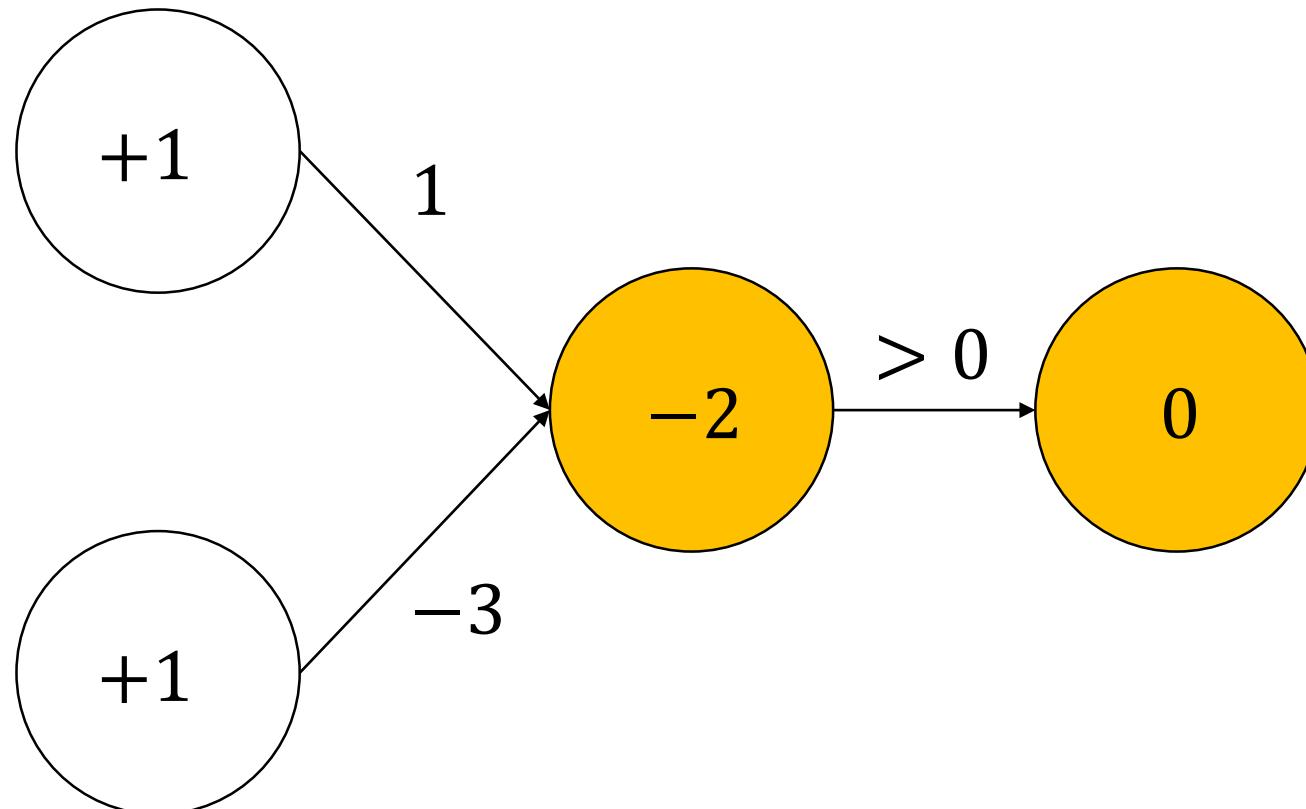
Deep Learning

- 퍼셉트론(Perceptron)

Pytorch

Install

예제2)



Deep Learning

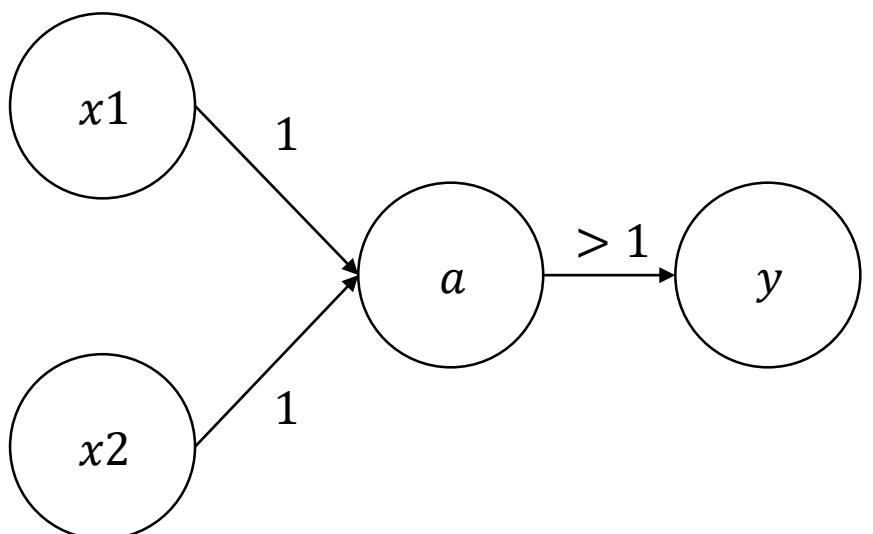
Deep Learning

- 퍼셉트론(Perceptron)

Pytorch

Install

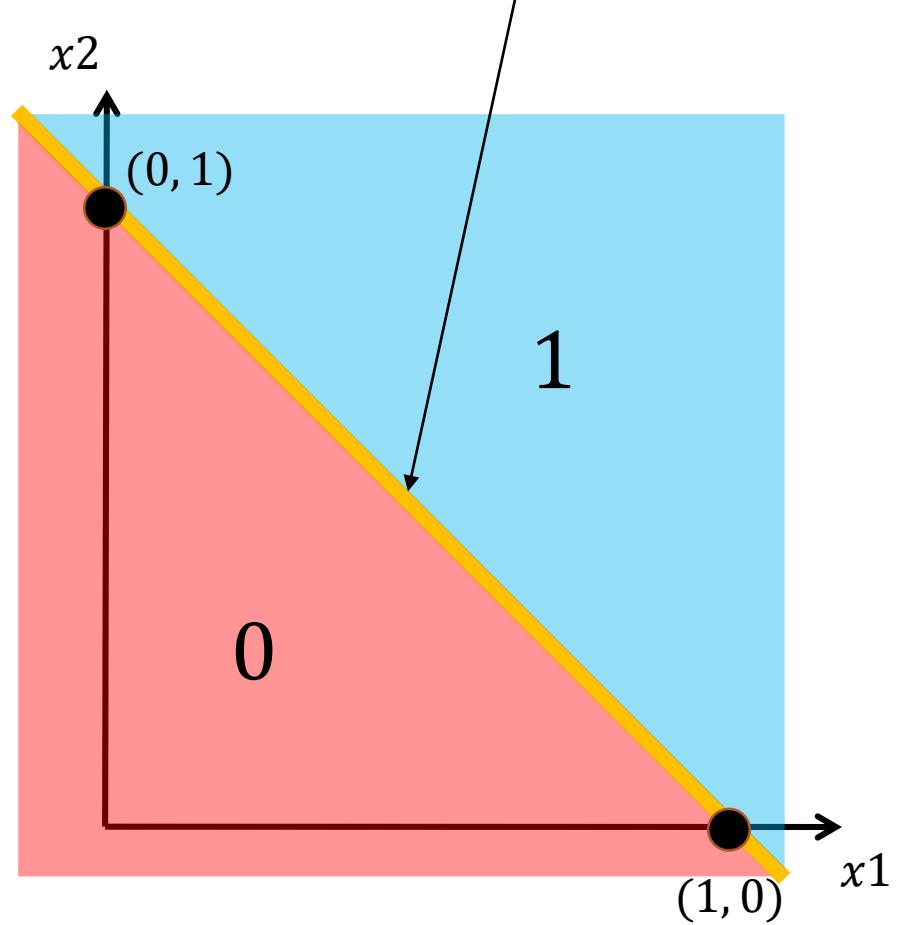
예제1) 퍼셉트론 = 선형분리!



$$1 * x_1 + 1 * x_2 = 1$$

$$\rightarrow x_2 = -x_1 + 1$$

$$x_2 = -x_1 + 1$$



Deep Learning

Deep Learning

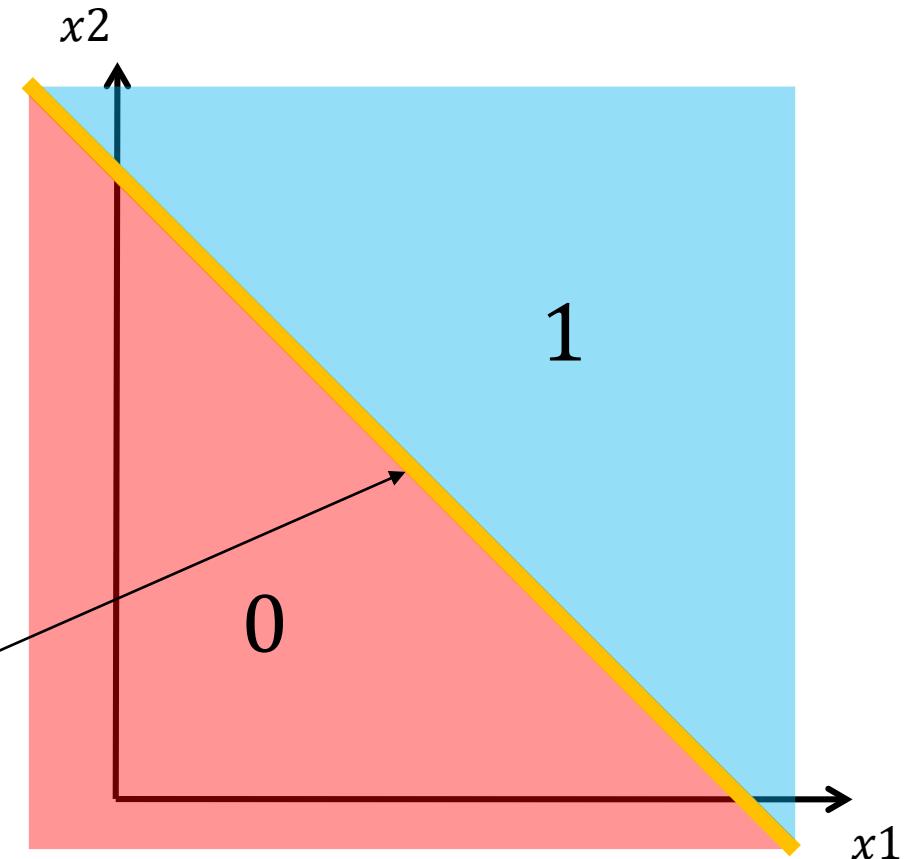
- 퍼셉트론(Perceptron)

Pytorch

Install

$$y = \begin{cases} 0 & (w_1 * x_1 + w_2 * x_2 \leq \theta) \\ 1 & (w_1 * x_1 + w_2 * x_2 > \theta) \end{cases}$$

$$\theta = w_1 * x_1 + w_2 * x_2$$



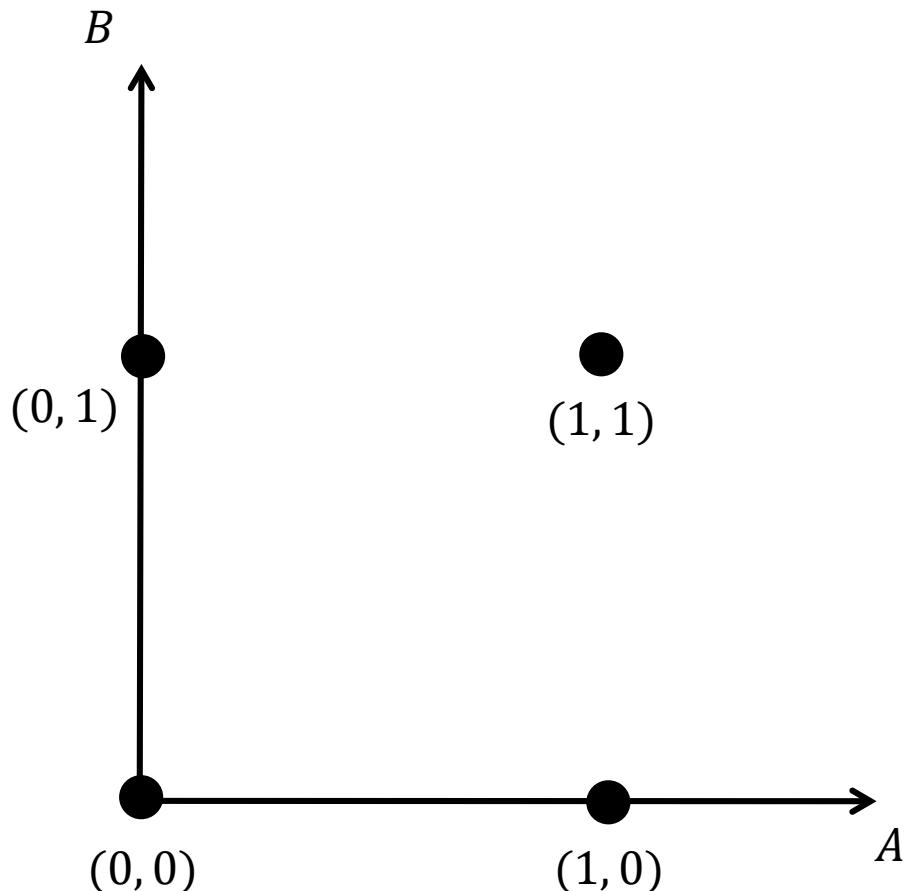
Deep Learning

Deep Learning

Pytorch

Install

- 퍼셉트론(Perceptron)
 - 다음과 같은 4개의 점을 분리하여 보자!



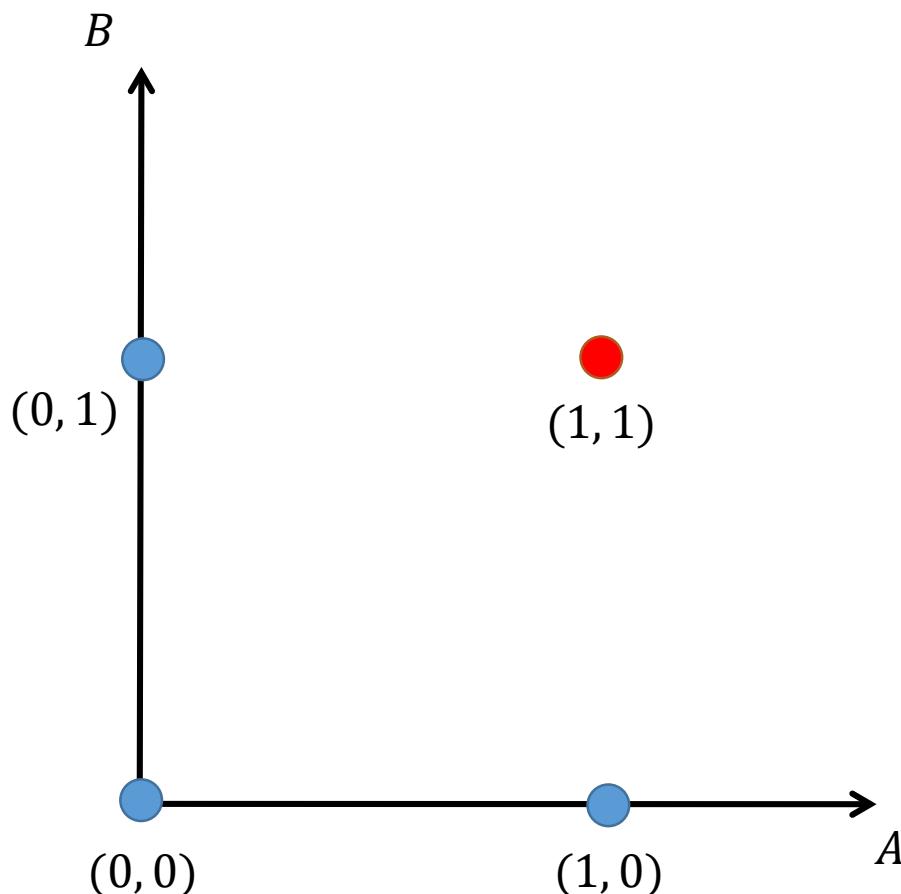
Deep Learning

Deep Learning

Pytorch

Install

- 퍼셉트론(Perceptron)
 - 다음과 같은 4개의 점을 분리하여 보자!



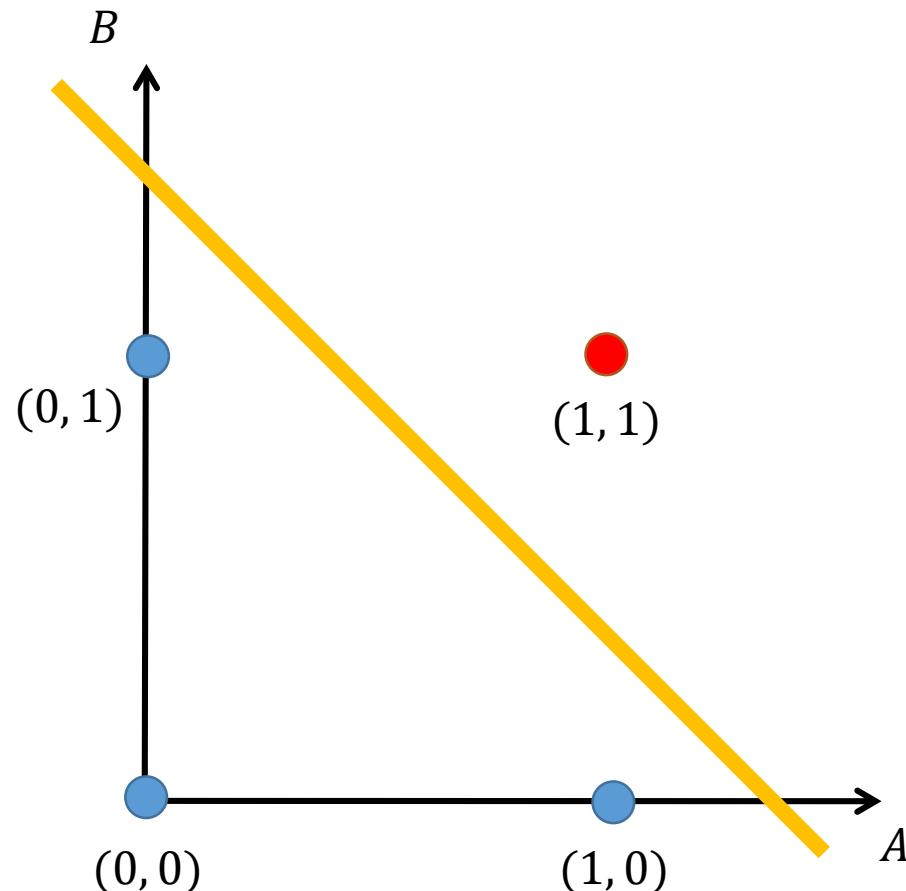
Deep Learning

Deep Learning

Pytorch

Install

- 퍼셉트론(Perceptron)
 - 다음과 같은 4개의 점을 분리하여 보자!



Deep Learning

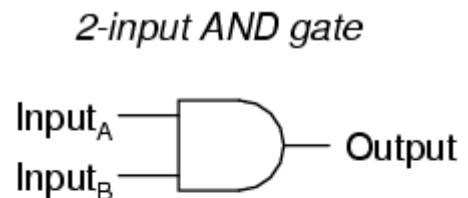
Deep Learning

- 퍼셉트론(Perceptron)

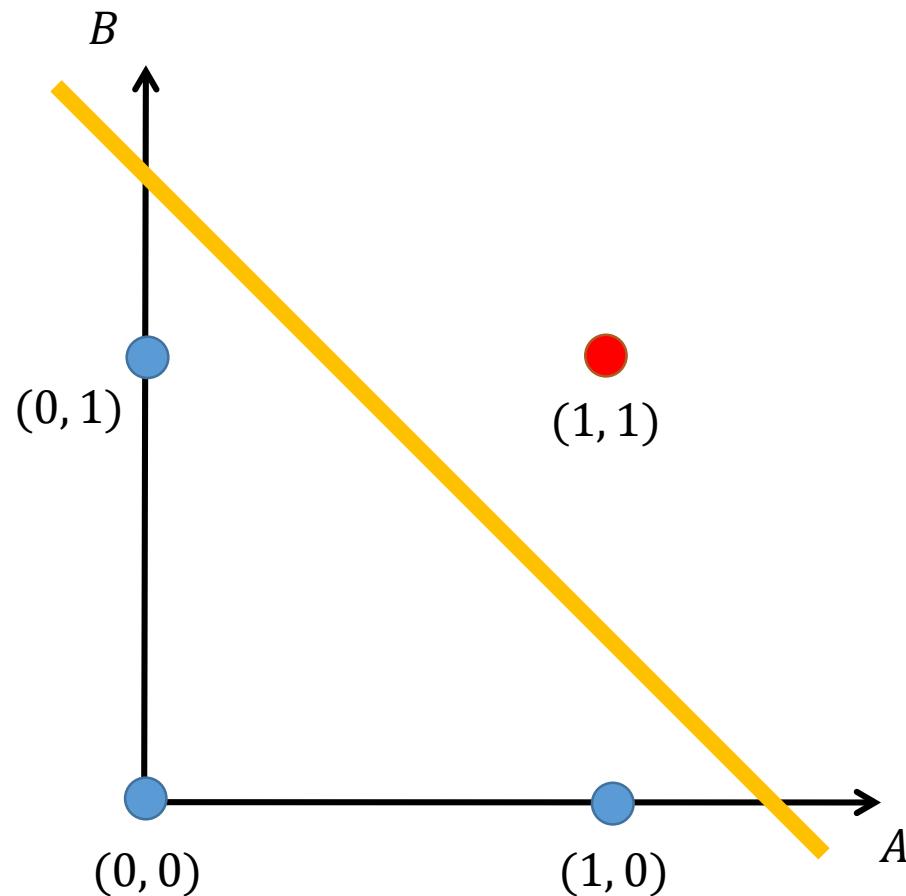
- AND 게이트

Pytorch

Install



A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



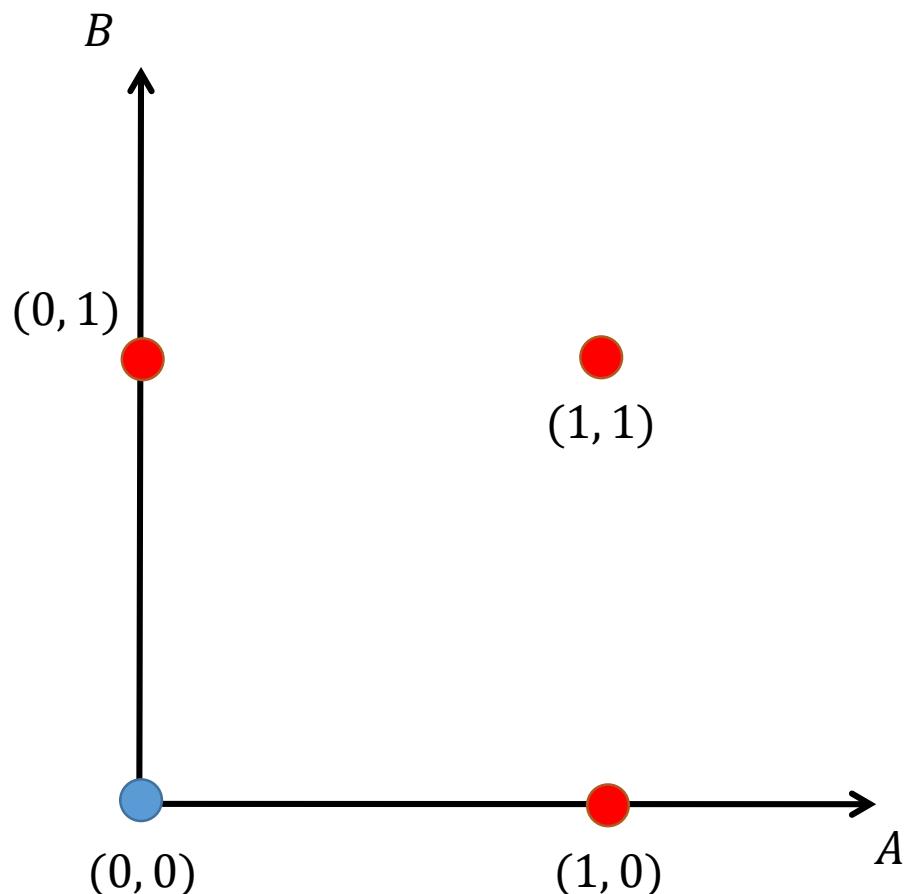
Deep Learning

Deep Learning

Pytorch

Install

- 퍼셉트론(Perceptron)
 - 다음과 같은 4개의 점을 분리하여 보자!



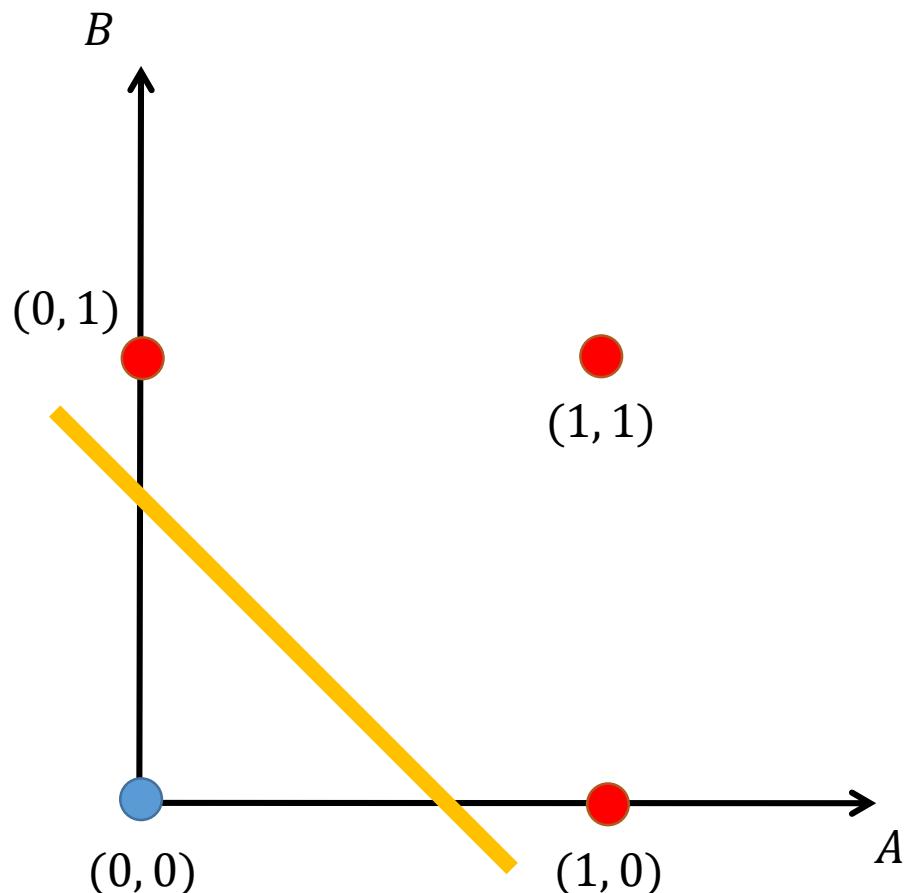
Deep Learning

Deep Learning

Pytorch

Install

- 퍼셉트론(Perceptron)
 - 다음과 같은 4개의 점을 분리하여 보자!



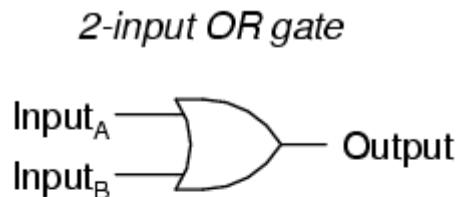
Deep Learning

Deep Learning

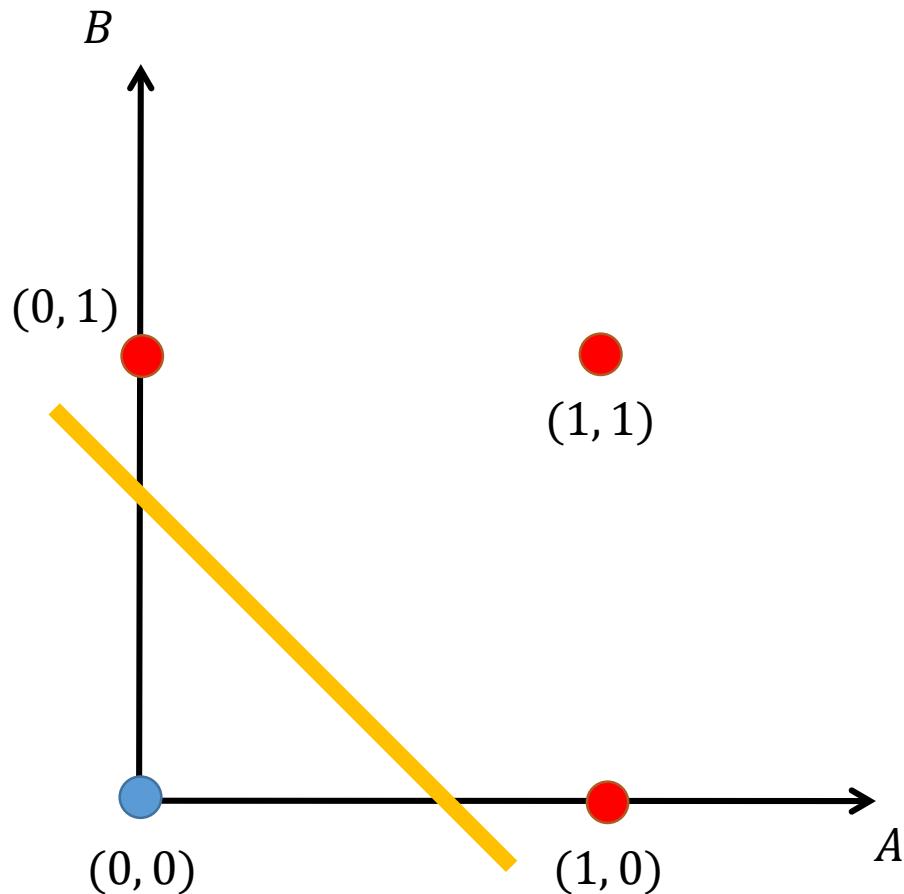
- 퍼셉트론(Perceptron)
 - OR 게이트

Pytorch

Install



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1
1	1	1



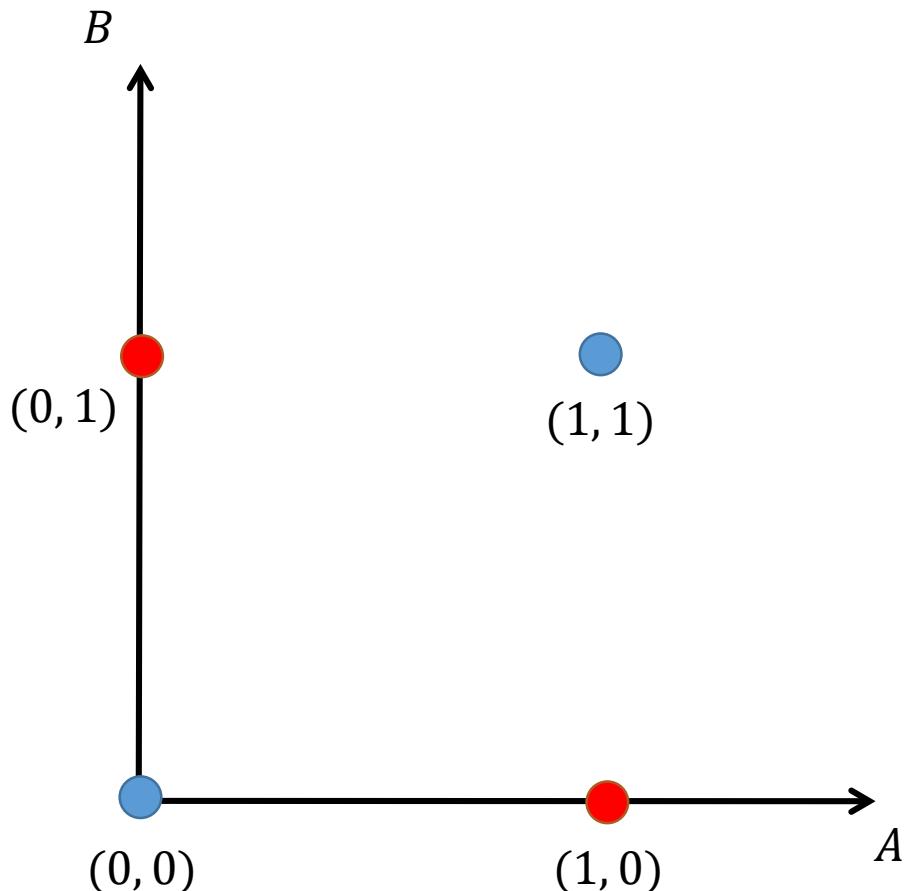
Deep Learning

Deep Learning

Pytorch

Install

- 퍼셉트론(Perceptron)
 - 다음과 같은 4개의 점을 분리하여 보자!



Deep Learning

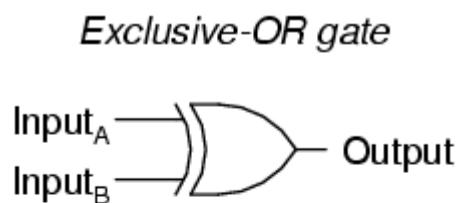
Deep Learning

- 퍼셉트론(Perceptron)

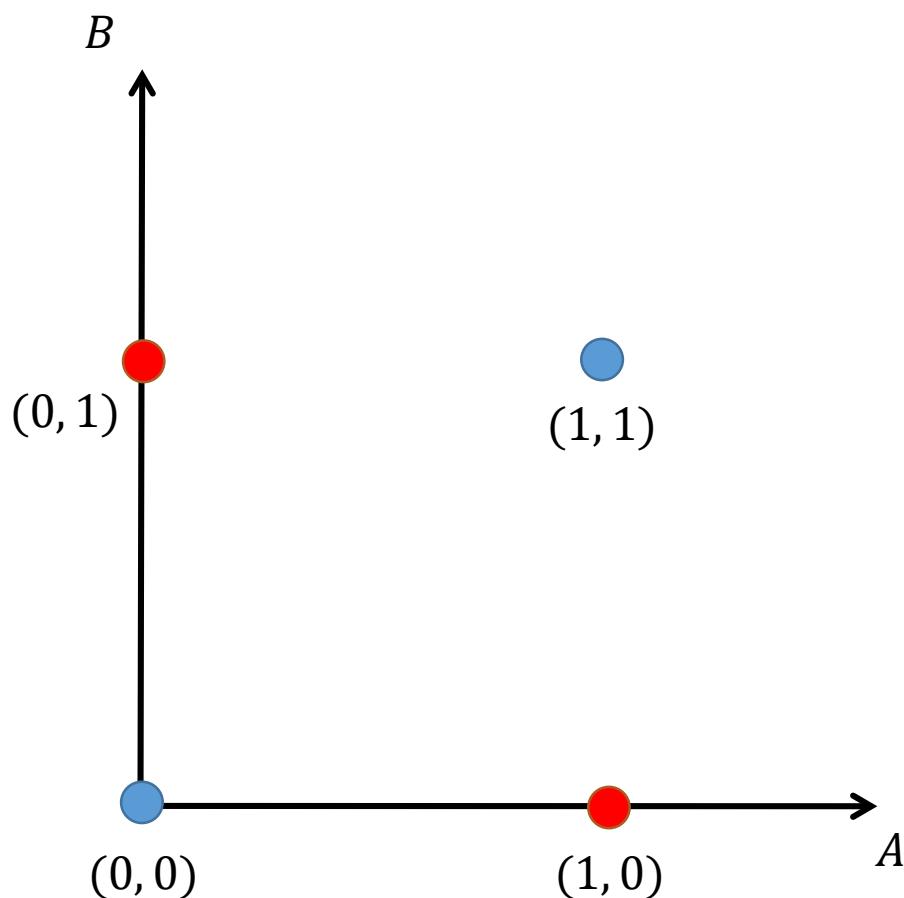
- XOR 게이트

Pytorch

Install



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



Deep Learning

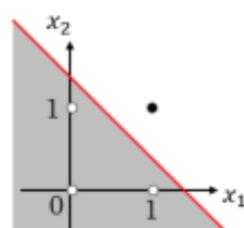
Deep Learning

Pytorch

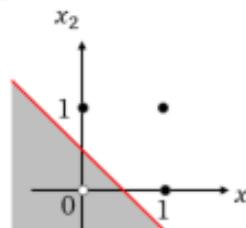
Install

■ 퍼셉트론(Perceptron)

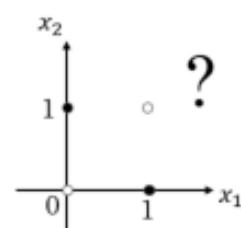
- 즉, 선형으로 구분할 수 없는 경우가 존재 (선형 분리 불가능)
- XOR 게이트



AND gate



OR gate



XOR gate

Input 1	Input 2	Output
1	1	0
1	0	1
0	1	1
0	0	0

https://inbi.ai/case_study.html

- 1개의 퍼셉트론(1 layer)으로는 XOR 게이트를 구현할 수 없음

Deep Learning

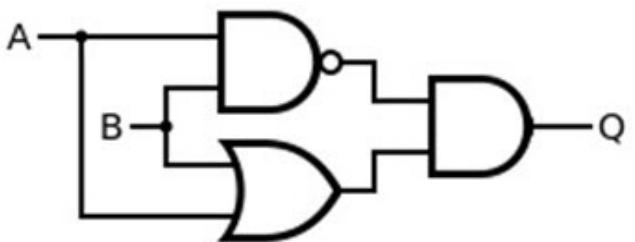
Deep Learning

Pytorch

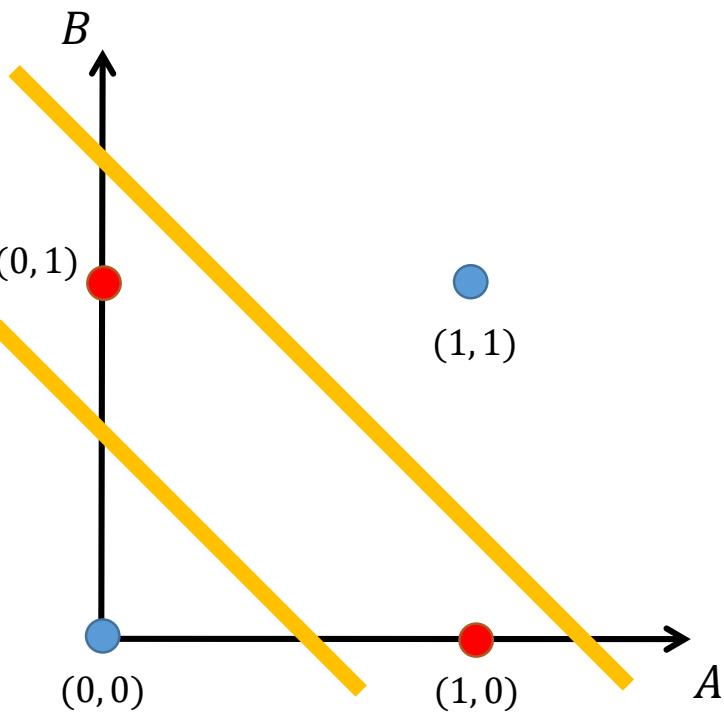
Install

■ 다층 퍼셉트론(Multilayer Perceptron)

- 따라서 여러 개의 퍼셉트론(Multi-layer perceptron)이 필요함
 - XOR은 2 Layer로 충분
 - $\text{XOR} = \text{AND}(\text{NAND}, \text{OR})$



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



Deep Learning

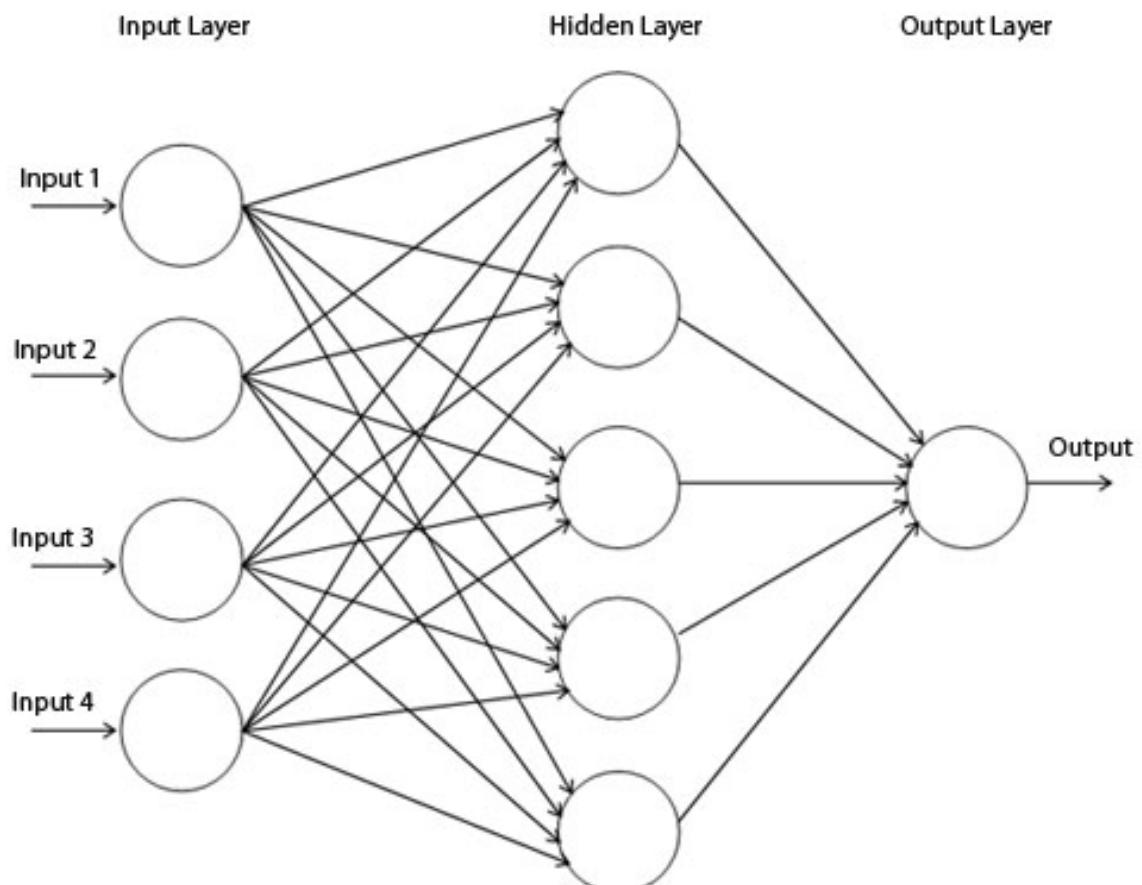
Deep Learning

Pytorch

Install

- 신경망(Neural Network)/다층 퍼셉트론(Multilayer Perceptron)

- 입력층(Input Layer)
- 은닉층(Hidden Layer)
- 출력층(Output Layer)



Deep Learning

Deep Learning

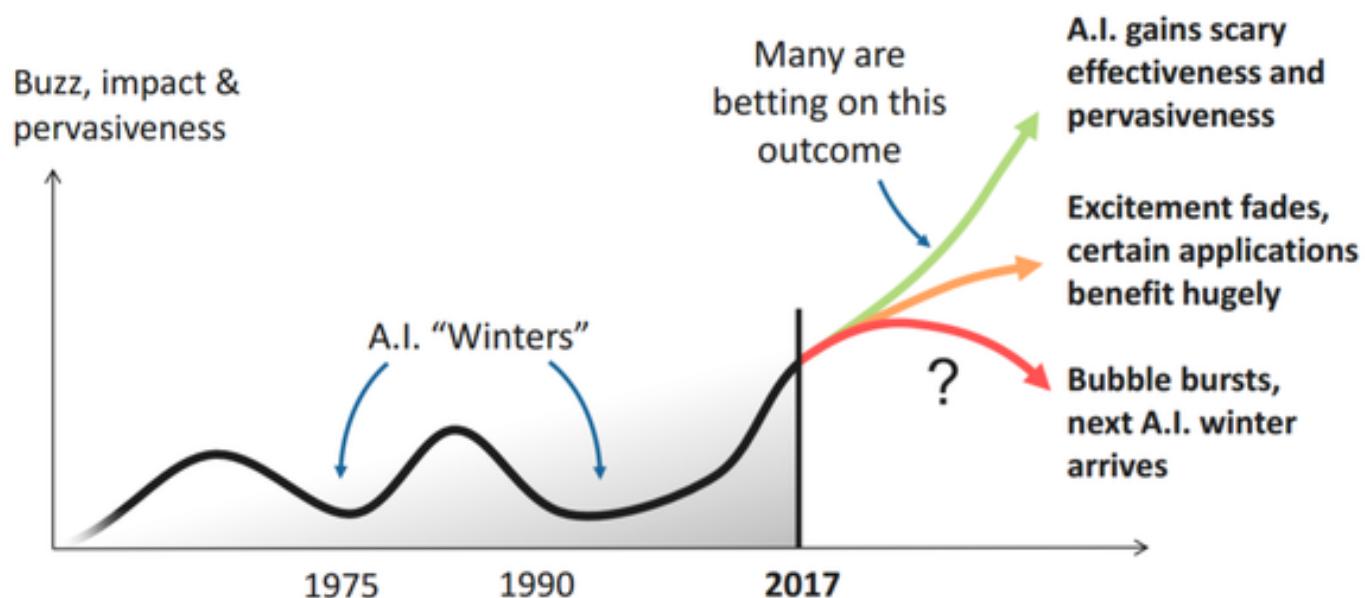
Pytorch

Install

■ 다층 퍼셉트론(Multilayer Perceptron)

- 문제는 “어떻게 학습시킬 것인가?”에 대한 고민
- 이는 곧 AI Winter로 이어짐

AI is enjoying significant hype and investment



<https://www.quora.com/What-caused-the-AI-winter-and-what-were-the-early-warning-signs-Given-the-state-of-AI-today-is-there-likely-to-be-another-period-of-low-interest-in-the-field-What-bottlenecks-would-be-the-cause-of-that>

Deep Learning

Deep Learning

Pytorch

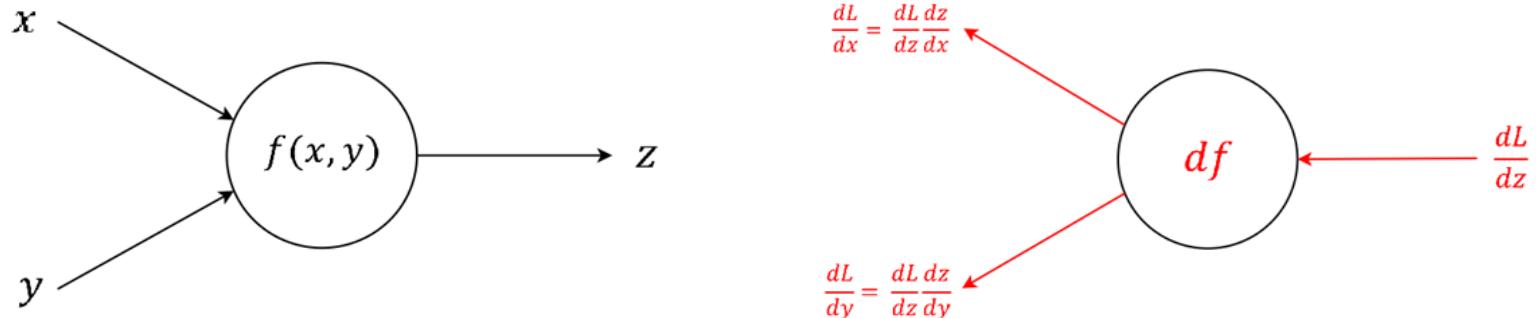
Install

- 역전파(Backpropagation)

- 1970년 다층 퍼셉트론의 학습을 위한 역전파(Backpropagation) 방법이 등장

Forwardpass

Backwardpass



- 동시에 1997년 CNN, LSTM 등의 알고리즘도 제안됨

Deep Learning

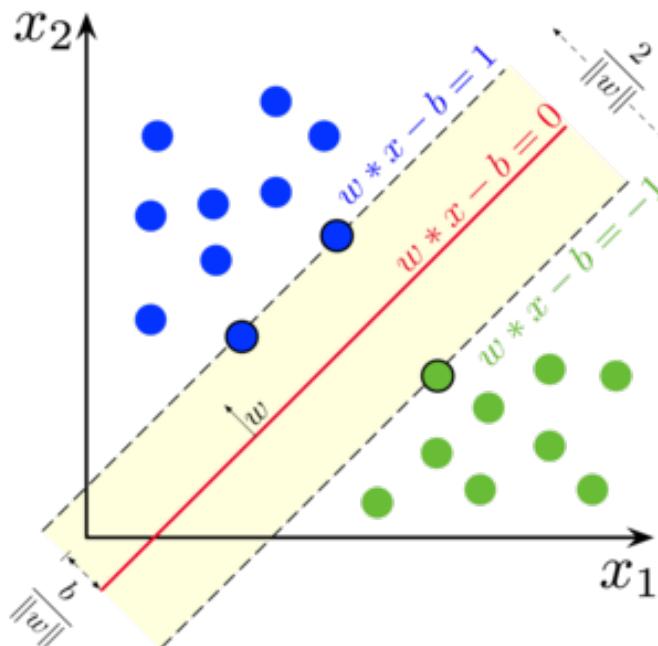
Deep Learning

Pytorch

Install

■ 역전파(Backpropagation)

- 하지만, 다음과 같은 문제로 연구 퇴보
 - Kernel Machine(SVM 등)의 등장과 성공
 - 하드웨어의 성능 부족 : GPU의 부재 등
 - 데이터의 부족



https://en.wikipedia.org/wiki/Support-vector_machine

Deep Learning

Deep Learning

Pytorch

Install

■ Imagenet

- 그러면 중 2009년 Imagenet Dataset0 등장함
 - 총 1.6천만개의 데이터
 - 100,000개의 Class로 이루어진 이미지들
 - 응급차, 달마시안, 이집트고양이 등등
- Imagenet Large Scale Visual Recognition Challenge (ILSVRC)
 - 1백만개의 이미지
 - 1000개의 Class

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



Deep Learning

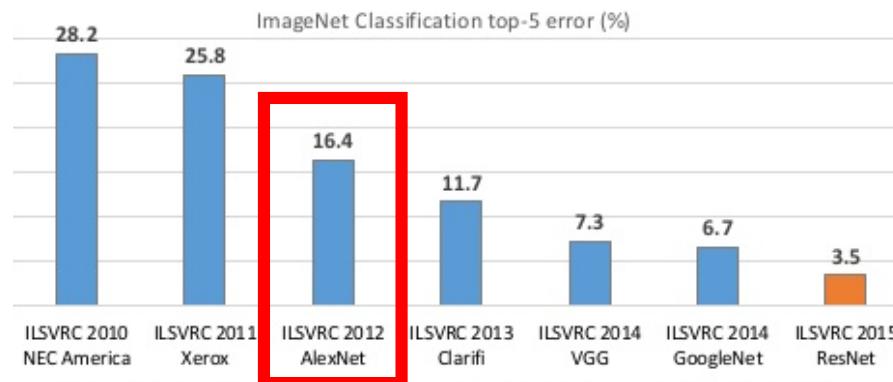
Deep Learning

Pytorch

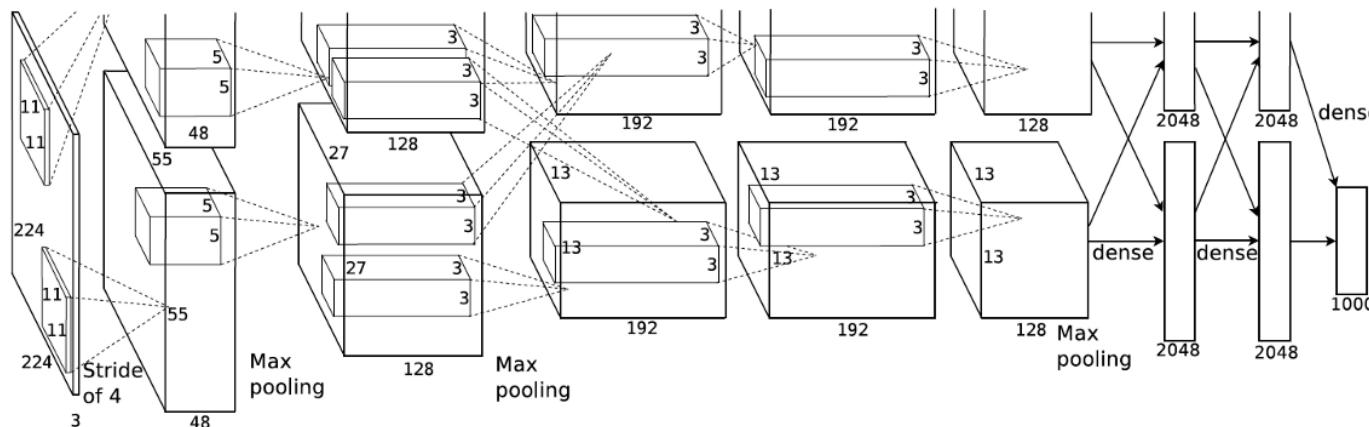
Install

AlexNet

- 2012년 이후 딥러닝 방식이 ILSVRC 최우수 팀으로 선정됨



AlexNet



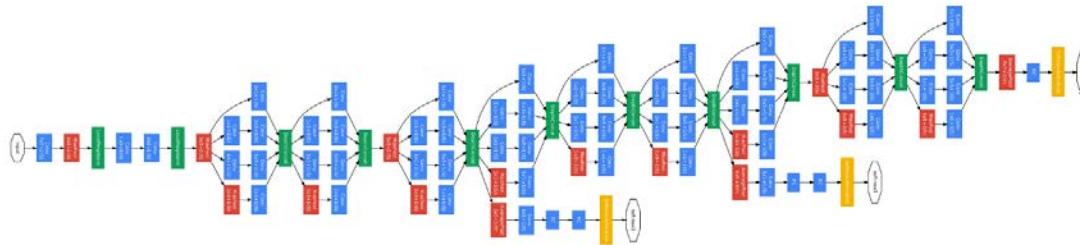
Deep Learning

Deep Learning

- GoogLeNet

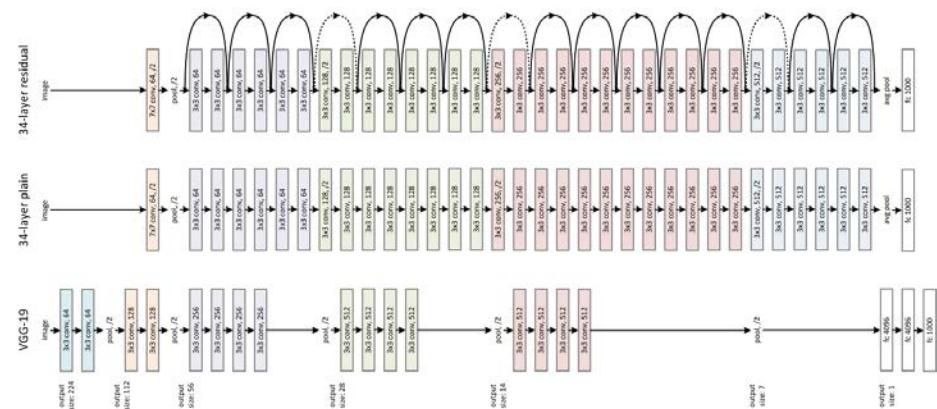
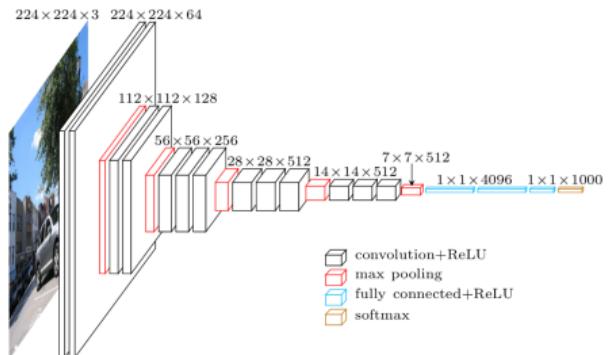
Pytorch

Install



- Resnet

- 그 이후 VGG/GoogLeNet 등 많은 아키텍쳐 시도로 ResNet이 3.5%까지 오류율을 낮춤
- 이 결과는 일반적인 인간의 인식 능력을 뛰어넘음



Deep Learning

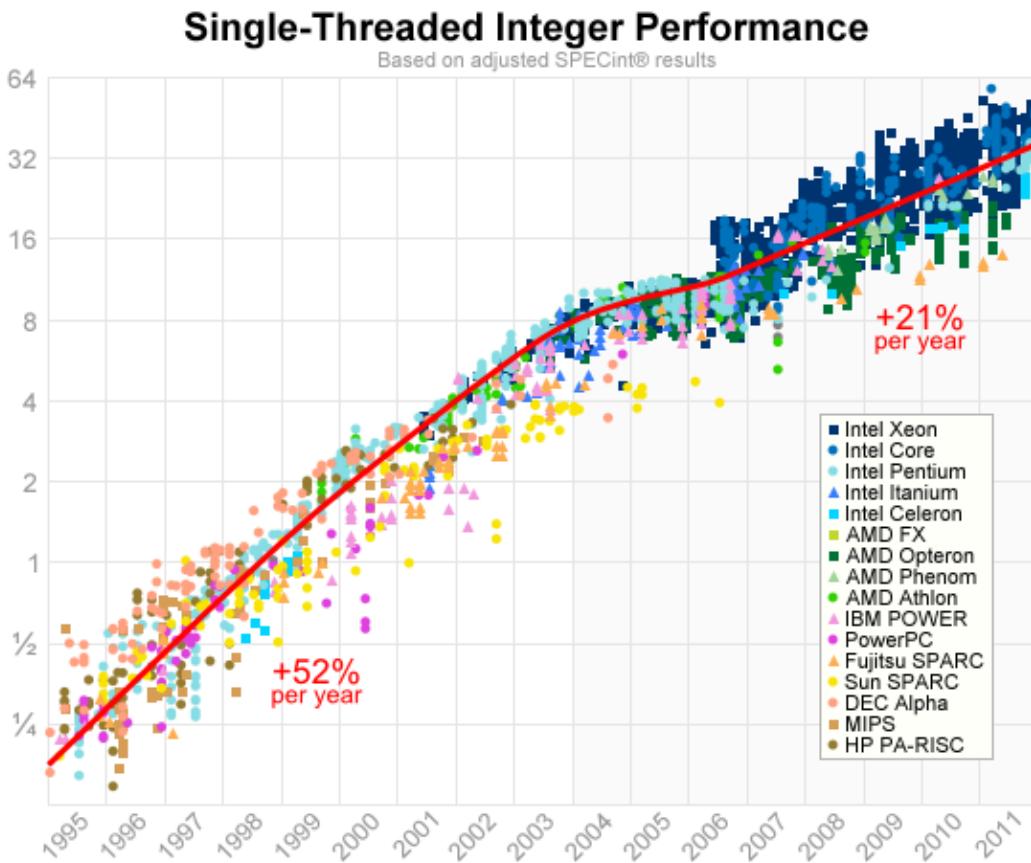
Deep Learning

- 어떻게 가능했는가?

1. 하드웨어의 성능 개선

Pytorch

Install

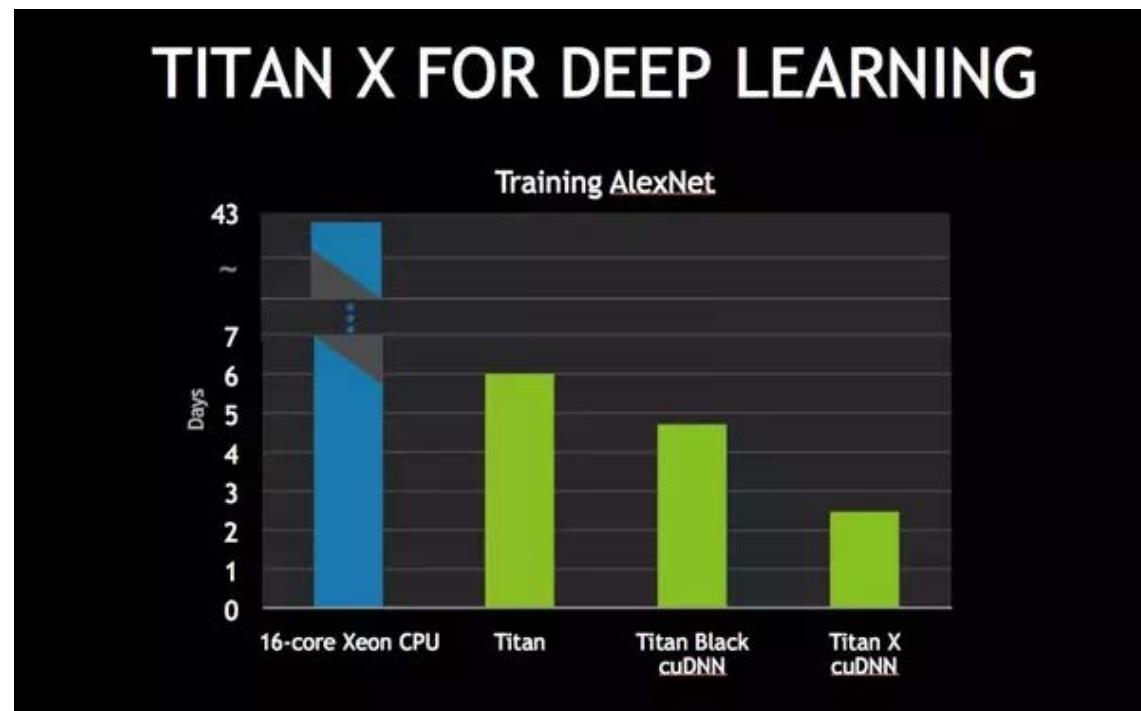


Deep Learning

Deep Learning

Pytorch

Install



■ 어떻게 가능했는가?

1. 하드웨어의 성능 개선

- > 16-core CPU에서는 43일
- > GPU(Titan X)에서는 6일

이는 GPU가 기존 그래픽 연산을 병렬적으로 처리하는 기능을 가지고 있었기 때문에, 딥러닝에서의 병렬적인 대량 연산을 빠르게 수행하기 때문이다. 반면 CPU는 연속적인 복잡 계산에 유용하여 CNN 등에서 성능이 GPU만큼 나올 수 없다.

GPU를 사용하기 위해서는 NVIDIA의 CUDA를 사용한다.

Deep Learning

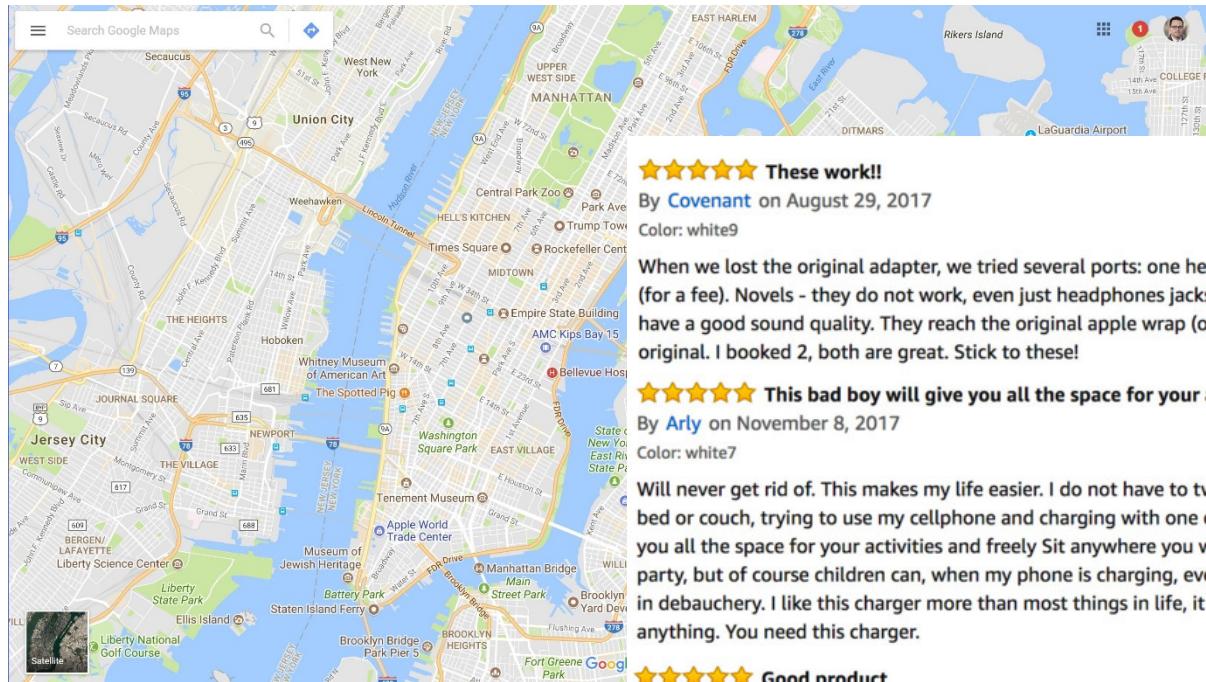
Deep Learning

Pytorch

Install

■ 어떻게 가능했는가?

2. 빅데이터의 등장



Deep Learning

Deep Learning

Pytorch

Install

■ 어떻게 가능했는가?

3. 성능 개선 방법의 등장

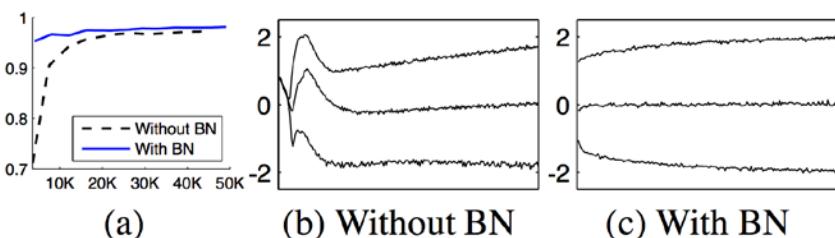
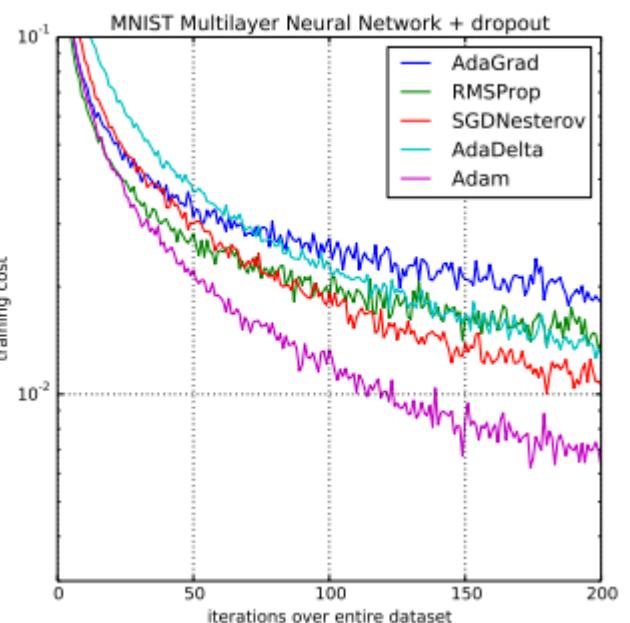
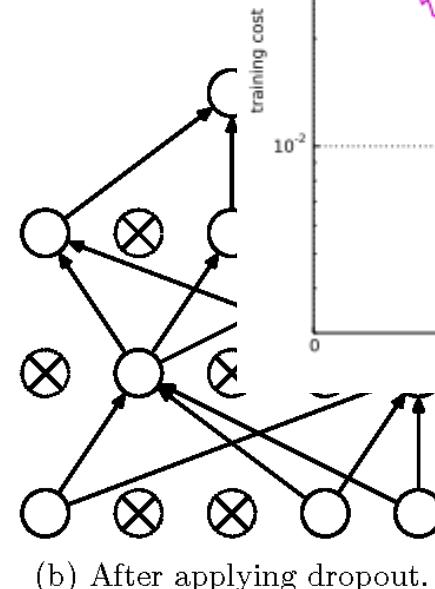
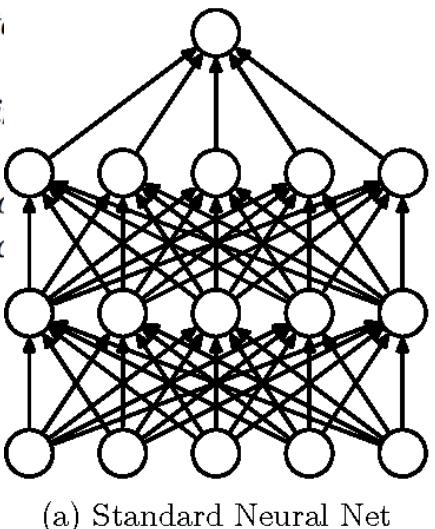


Figure 1: (a) The test accuracy of the MNIST network trained with and without Batch Normalization, vs. the number of training steps. (b) The evolution of individual neuron's mean and standard deviation over time. (c) The evolution of individual neuron's mean and standard deviation over time. Batch Normalization makes the network train faster and more stable.



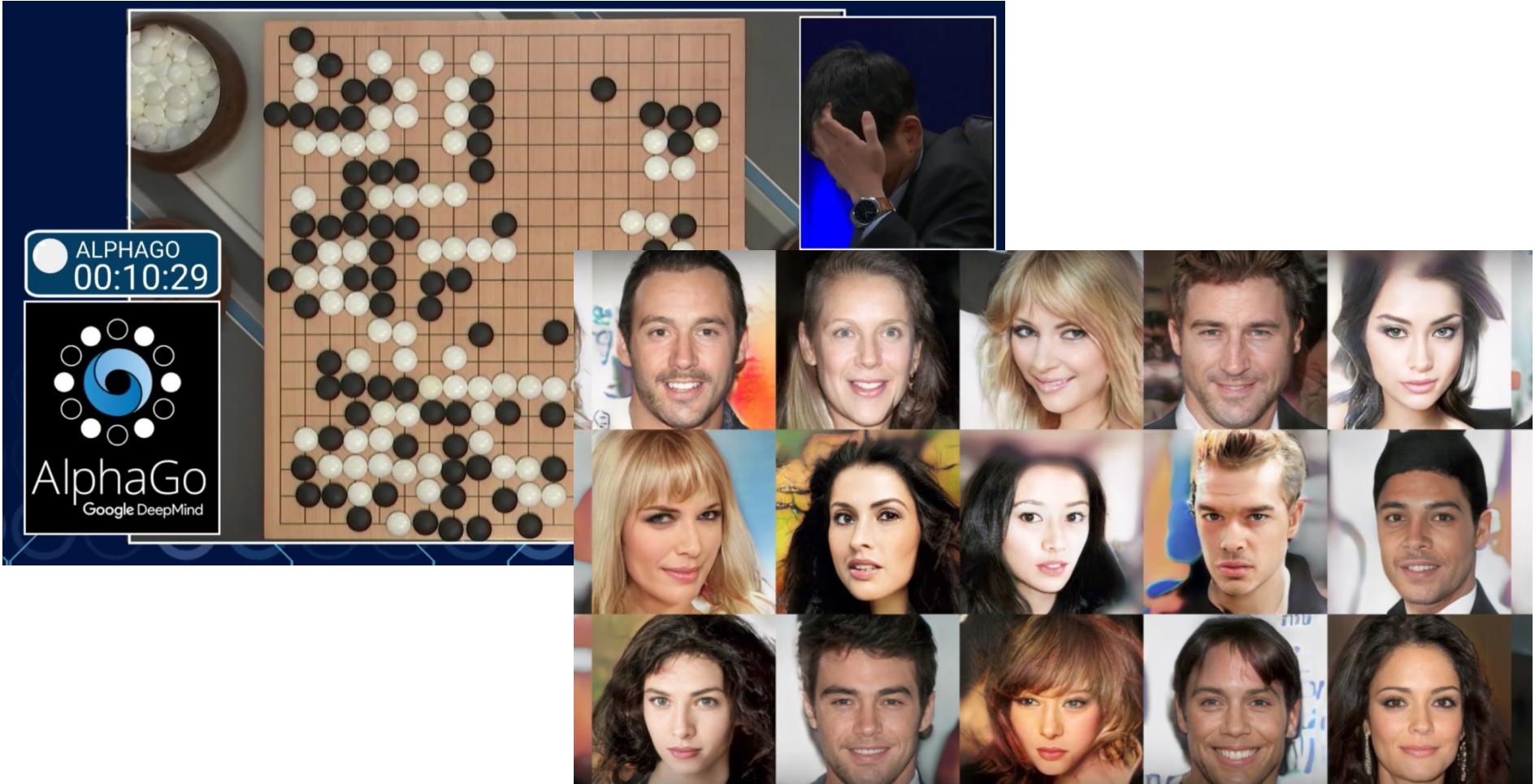
Deep Learning

Deep Learning

- Alpha Go의 등장과 범용화

Pytorch

Install



Deep Learning

Deep Learning

Pytorch

Install

- **새로이 등장하는 학습 방법들**
 - 지도/교사 학습(Supervised Learning)
 - 데이터와 정답을 제공하여 학습하게 하는 알고리즘
 - 회귀/분류
 - **Neural Networks**
 - 비지도/비교사 학습(Unsupervised Learning)
 - 데이터만을 제공하여 학습하게 하는 알고리즘
 - 클러스터링
 - **Autoencoders**
 - **강화학습(Reinforcement Learning)**
 - 주어진 리워드를 최대화하도록 학습하는 알고리즘
 - DQN, DDPG

Deep Learning

Deep Learning

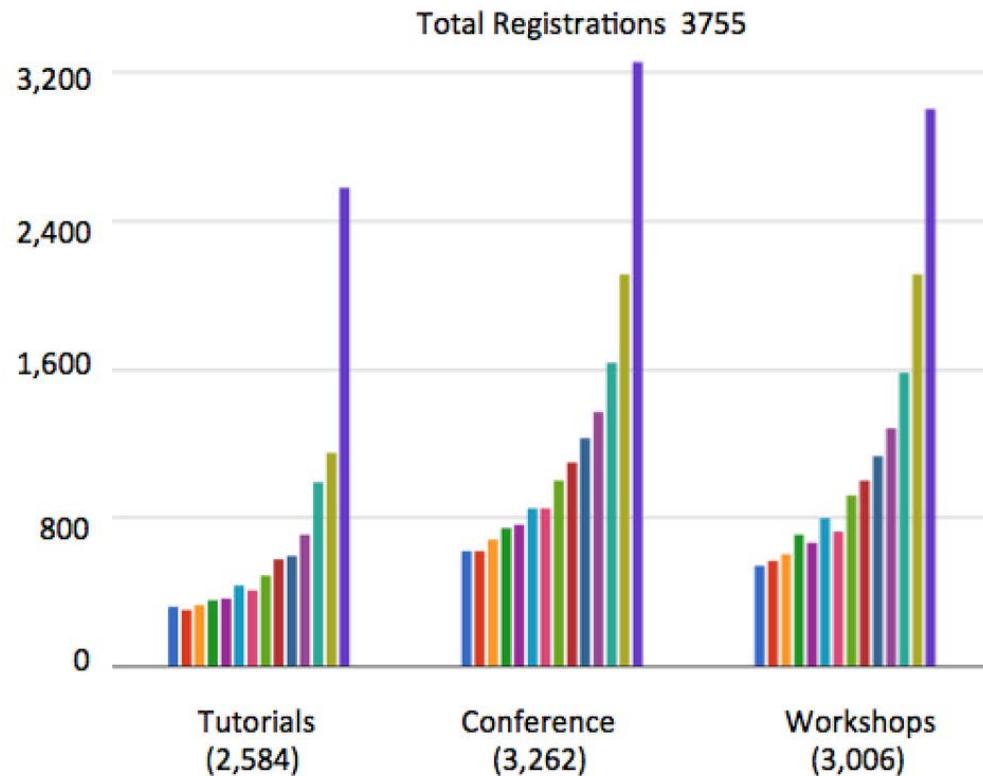
- 최근 학계 동향

- NIPS(Neural Information Processing Systems)

Pytorch

Install

NIPS Growth



Deep Learning

Pytorch

Install

2. Pytorch

■ Pytorch

- **Developer** : Facebook's artificial-intelligence research group
- **OS** : Linux, macOS, Windows
- <https://pytorch.org/>
- <https://pytorch.org/docs/stable/index.html>

▪ Version Released

- 2017.12.05 : Pytorch 0.3.0 Release
- 2018.02.15 : Pytorch 0.3.1 Release
- 2018.04.25 : Pytorch 0.4.0 Release
- 2018.10.04 : Pytorch 1.0.0 Release

▪ Community

- Pytorch KR
- 6000명 이상



A screenshot of the PyTorch documentation website. The header features the PyTorch logo and a "version selector" dropdown set to "0.4.1". Below the header is a search bar labeled "Search docs". The main content area is divided into sections: "NOTES" (Autograd mechanics, Broadcasting semantics, CUDA semantics, Extending PyTorch, Frequently Asked Questions, Multiprocessing best practices, Serialization semantics, Windows FAQ), "PACKAGE REFERENCE" (torch, torch.Tensor, Tensor Attributes, torch.sparse, torch.cuda, torch.Storage, torch.nn, torch.nn.functional, torch.nn.init, torch.optim, torch.autograd, torch.distributions, torch.multiprocessing), and "torch._utils" (torch._utils.checkpoint).

Docs » PyTorch documentation

Edit on GitHub

PyTorch documentation

PyTorch is an optimized tensor library for deep learning using GPUs and CPUs.

Notes

- Autograd mechanics
- Broadcasting semantics
- CUDA semantics
- Extending PyTorch
- Frequently Asked Questions
- Multiprocessing best practices
- Serialization semantics
- Windows FAQ

Package Reference

- torch
- torch.Tensor
- Tensor Attributes
- torch.sparse
- torch.cuda
- torch.Storage
- torch.nn
- torch.nn.functional
- torch.nn.init
- torch.optim
- torch.autograd
- torch.distributions
- torch.multiprocessing
- torch._utils
- torch._utils.checkpoint

Pytorch

Deep Learning

Pytorch

Install

- **TensorFlow v.s. Pytorch**
 - **TensorFlow**
 - Python 기반의 라이브러리
 - 데스크톱, 모바일 등 다양한 플랫폼에 적용 가능
 - Tensor Board
 - Define-and-Run
 - **Pytorch**
 - 기존 Lua 기반의 딥러닝 프레임워크(torch)를 python으로 이식
 - Python과의 유연한 연동 (DataLoader, Numpy, DataFrame)
 - 쉽고 간결한 코드
 - Define-by-Run

Pytorch

Deep Learning

Pytorch

Install

	Languages	Tutorials and training materials	CNN modeling capability	RNN modeling capability	Architecture: easy-to-use and modular front end	Speed	Multiple GPU support	Keras compatible
Theano	Python, C++	++	++	++	+	++	+	+
Tensor-Flow	Python	+++	+++	++	+++	++	++	+
Torch	Lua, Python (new)	+	+++	++	++	+++	++	
Caffe	C++	+	++		+	+	+	
MXNet	R, Python, Julia, Scala	++	++	+	++	++	+++	
Neon	Python	+	++	+	+	++	+	
CNTK	C++	+	+	+++	+	++	+	

Deep Learning

Pytorch

Install

```
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()

        self.layer = nn.Sequential(
            nn.Conv2d(1, 16, 5),
            nn.ReLU(),
            nn.Conv2d(16, 32, 5),
            nn.ReLU(),
            nn.MaxPool2d(2, 2),
            nn.Conv2d(32, 64, 5),
            nn.ReLU(),
            nn.MaxPool2d(2, 2)
        )

        self.fc_layer = nn.Sequential(
            nn.Linear(64*3*3, 100),
            nn.ReLU(),
            nn.Linear(100, 10)
        )

    def forward(self, x):
        out = self.layer(x)
        out = out.view(batch_size, -1)
        out = self.fc_layer(out)

        return out
```

v.s.

```
X = tf.placeholder(tf.float32, [None, 28, 28, 1])
Y = tf.placeholder(tf.float32, [None, 10])
keep_prob = tf.placeholder(tf.float32)

W1 = tf.Variable(tf.random_normal([3, 3, 1, 32], stddev=0.01))
L1 = tf.nn.conv2d(X, W1, strides=[1, 1, 1, 1], padding='SAME')
L1 = tf.nn.relu(L1)
L1 = tf.nn.max_pool(L1, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')

W2 = tf.Variable(tf.random_normal([3, 3, 32, 64], stddev=0.01))
L2 = tf.nn.conv2d(L1, W2, strides=[1, 1, 1, 1], padding='SAME')
L2 = tf.nn.relu(L2)
L2 = tf.nn.max_pool(L2, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')

W3 = tf.Variable(tf.random_normal([7*7*64, 256], stddev = 0.01))
L3 = tf.reshape(L2, [-1, 7*7*64])
L3 = tf.matmul(L3, W3)
L3 = tf.nn.relu(L3)
L3 = tf.nn.dropout(L3, keep_prob)

W4 = tf.Variable(tf.random_normal([256, 10], stddev=0.01))
model = tf.matmul(L3, W4)
```

Deep Learning

Pytorch

Install

```
img_dir = "./image"
img_data = dsets.ImageFolder(img_dir, transforms.Compose([
    transforms.Grayscale(),
    transforms.ToTensor(),
]))
https://pytorch.org/docs/stable/torchvision/transforms.html

print(img_data.classes)
print(img_data.class_to_idx)
```

V.S.

```
nclass = len(paths)
valid_exts = [".jpg", ".gif", ".png", ".tga", ".jpeg"]
for i, relpath in zip(range(nclass), paths):
    imgcnt = 0
    path = cwd + "/image_small/" + relpath
    flist = os.listdir(path)
    for f in flist:
        if os.path.splitext(f)[1].lower() not in valid_exts:
            continue
        fullpath = os.path.join(path, f)
        currimg = imread(fullpath)

        if use_gray:
            grayimg = rgb2gray(currimg)
        else:
            grayimg = currimg

        graysmall = imresize(grayimg, (imgsize[0], imgsize[1]))
        grayvec = np.reshape(graysmall, (1, -1))

        # one-hot 코딩을 위해, 정방단위행렬을 만들고, i행만 추출
        curr_label = np.eye(nclass, nclass)[i:i+1, :]
        curr_label = [relpath]
        if imgcnt is 0:
            totalimg = grayvec
            totallabel = curr_label
        else:
            totalimg = np.concatenate((totalimg, grayvec), axis = 0)
            totallabel = np.concatenate((totallabel, curr_label), axis = 0)
        imgcnt = imgcnt + 1

        savepath = cwd + "/data_small/" + relpath + ".npz"
        np.savez(savepath, totalimg=totalimg, totallabel=totallabel)
        print(relpath, "has SAVED")

print("Total %d images" % (imgcnt))
```

Pytorch

Deep Learning

Pytorch

Install

↑ Posted by u/cjmcmurtrie 1 year ago ▾

203 Discussion [D] So... Pytorch vs Tensorflow: what's the verdict on how they compare? What are their individual strong points?

Have any users here had extensive experience with both? What are your main concerns or delights with both libraries?

I never made a switch from Torch7 to Tensorflow. I played around with Tensorflow but I always found Torch7 more intuitive (maybe I didn't play around enough!). I also had a tip that Pytorch was on the way, so decided I would wait for that.

After a few weeks using Pytorch, I don't think I'll be moving to Tensorflow any time soon, at least for my passion projects. It's ridiculously simple to write custom modules in Pytorch, and the dynamic graph construction is giving me so many ideas for things that previously would've been achieved by late-night hacks (and possibly put on the wait list). I think Pytorch is an incredible toolset for a machine learning developer. I realise that the wealth of community resources is much stronger for Tensorflow, but when working on novel projects (instead of re-coding known architectures or reading tutorials) this isn't always much help.

Thoughts?

47 Comments Share ...

94% Upvoted

↑ thatguydr 12 points · 1 year ago

↓ This is the best reply I've seen in this subreddit in months. I've been using TF at work strictly because of its distributed capabilities, but I'm going to check out PyTorch at home. Thanks!

Share Save

직장에서는 TensorFlow가 범용성(모바일 이식 등) 때문에 강제되지만,
확인하는 것은 Pytorch로 집에서 하겠다.

Pytorch

Deep Learning

Pytorch

Install



Deep Learning

Pytorch

Install

3. Install

Install

Deep Learning

Pytorch

Install

- 아나콘다(ANACONDA)란?

- 자주 사용되는 수학&과학 분야의 여러 패키지들을 한 번에 제공하는 **파이썬 배포 프로그램**
- Scipy, Numpy, Matplotlib, Pandas 등 제공
- 머신러닝, 딥러닝 분야에서 **아나콘다 사용 권장**
- Pytorch 역시 아나콘다 사용을 권장 중
- **독립된 실행환경(가상환경) 구축 가능**

Install

Deep Learning

Pytorch

Install

- 아나콘다(ANACONDA) 설치

- <https://www.anaconda.com/download/>



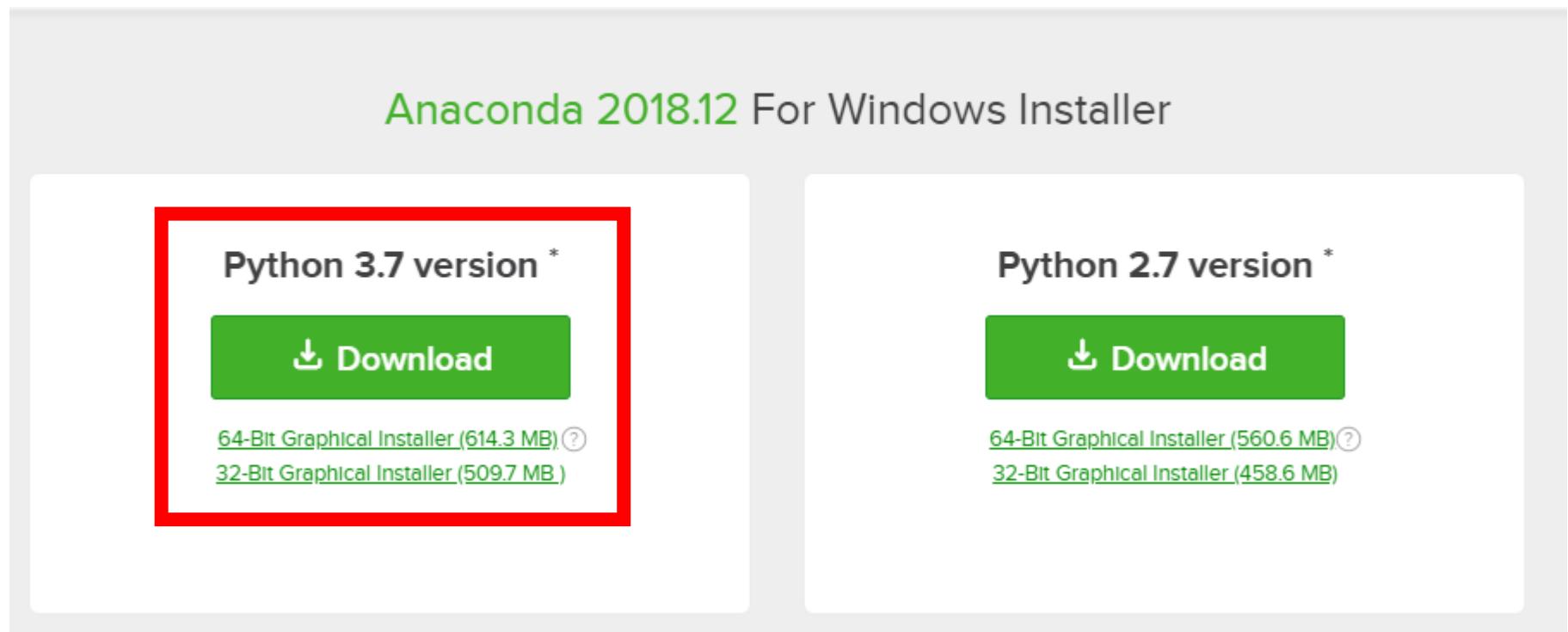
Windows



macOS



Linux



Anaconda 2018.12 For Windows Installer

Python 3.7 version *

 Download

[64-Bit Graphical Installer \(614.3 MB\)](#) (?)
[32-Bit Graphical Installer \(509.7 MB\)](#)

Python 2.7 version *

 Download

[64-Bit Graphical Installer \(560.6 MB\)](#) (?)
[32-Bit Graphical Installer \(458.6 MB\)](#)

Install

Deep Learning

Pytorch

Install

- 아나콘다(ANACONDA) 설치

<https://repo.anaconda.com/archive/>

[Anaconda3-5.2.0](#)

Anaconda3-5.2.0-Linux-ppc64le.sh	288.3M	2018-05-30 13:05:40	cbd1d5435ead2b0b97dba5b3cf45d694
Anaconda3-5.2.0-Linux-x86.sh	507.3M	2018-05-30 13:05:46	81d5a1648e3aca4843f88ca3769c0830
Anaconda3-5.2.0-Linux-x86_64.sh	621.6M	2018-05-30 13:05:43	3e58f494ab9fbe12db4460dc152377b5
Anaconda3-5.2.0-MacOSX-x86_64.pkg	613.1M	2018-05-30 13:07:00	9c35bf27e9986701f7d80241616c665f
Anaconda3-5.2.0-MacOSX-x86_64.sh	523.3M	2018-05-30 13:07:03	b5b789c01e1992de55ee911754c310d4
Anaconda3-5.2.0-Windows-x86.exe	506.3M	2018-05-30 13:04:19	285387e7b6ea81edba98c011922e235a
Anaconda3-5.2.0-Windows-x86_64.exe	631.3M	2018-05-30 13:04:18	62244c0382b8142743622fdc3526eda7

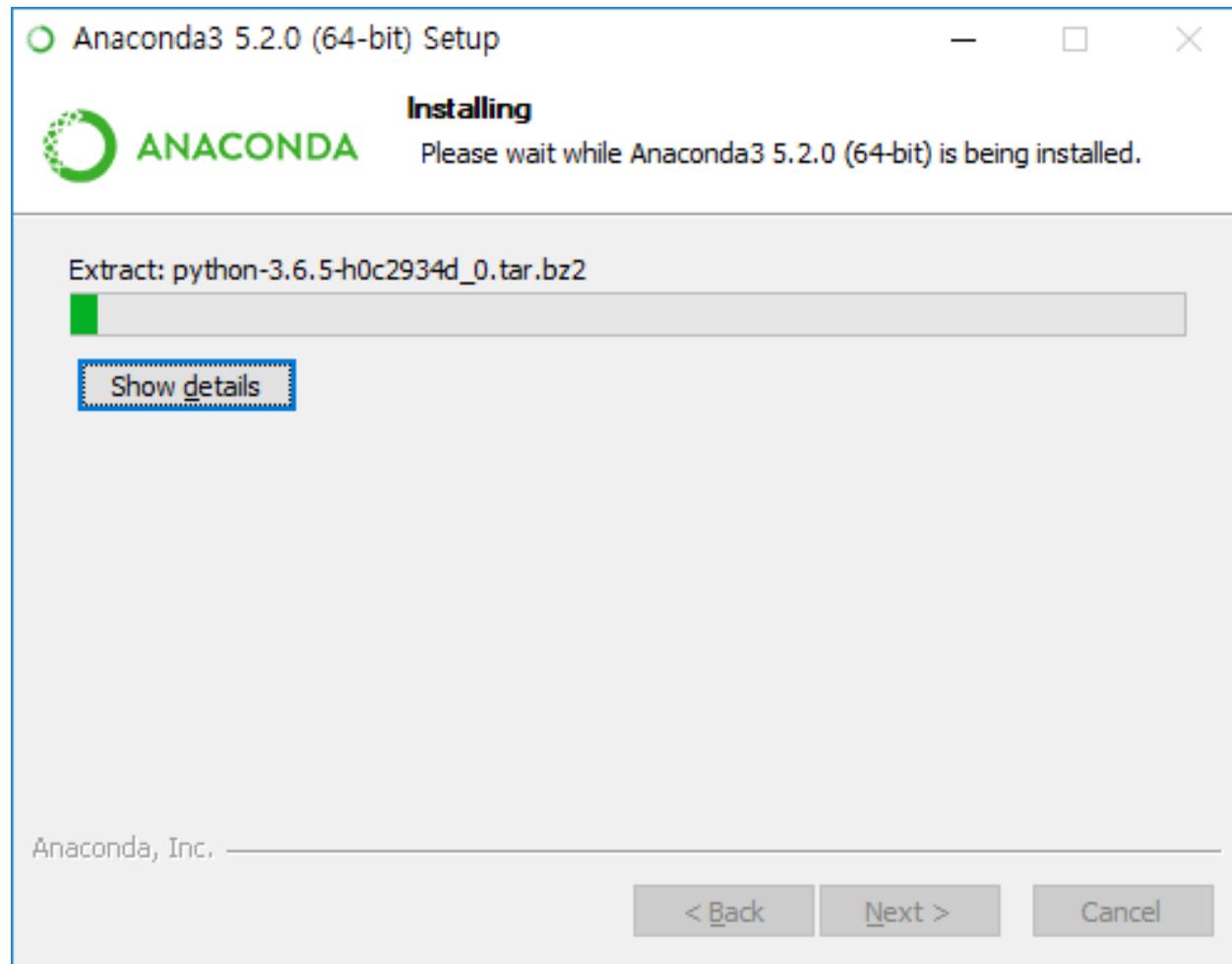
Install

Deep Learning

Pytorch

Install

- 아나콘다(ANACONDA) 설치



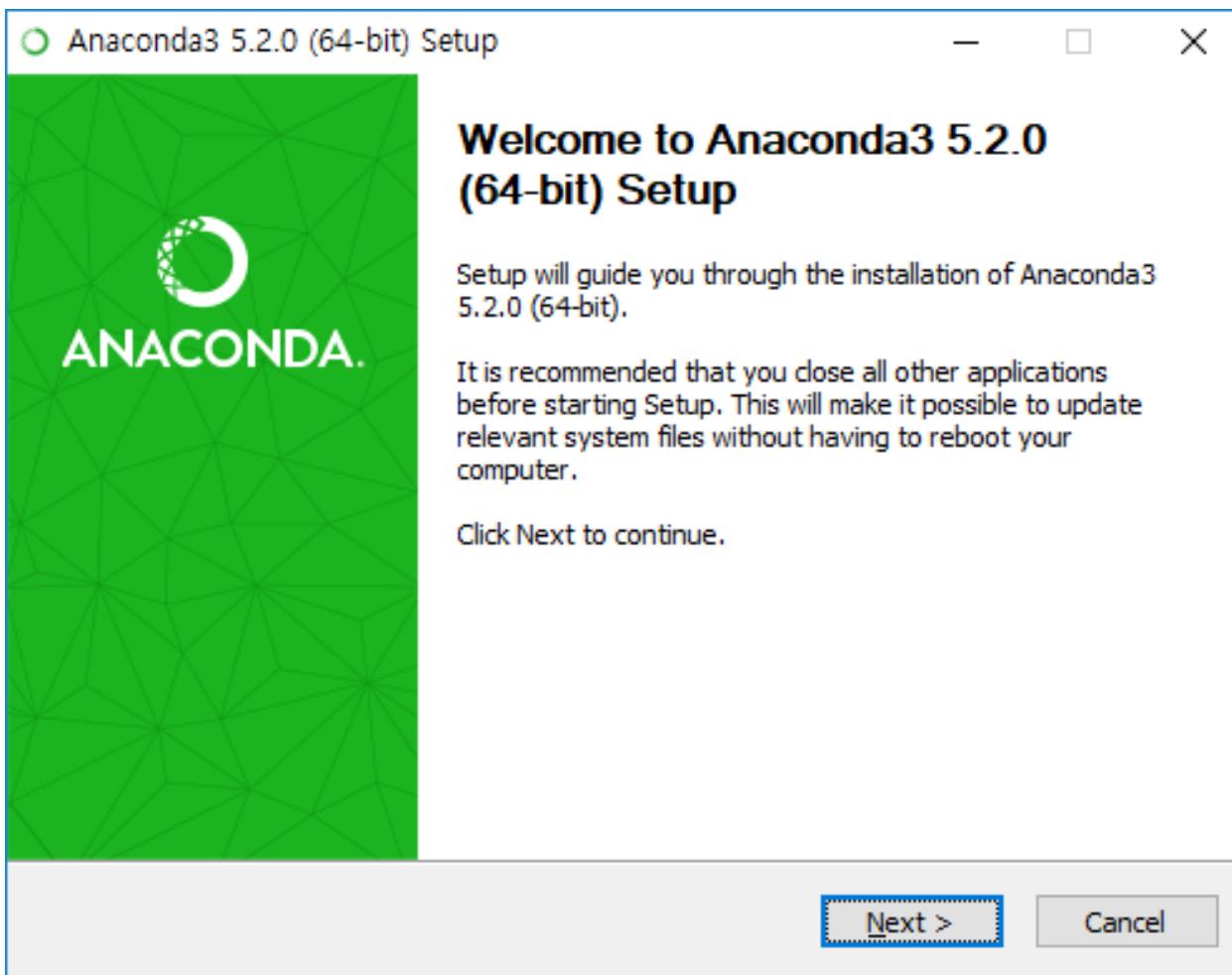
Install

Deep Learning

Pytorch

Install

- 아나콘다(ANACONDA) 설치



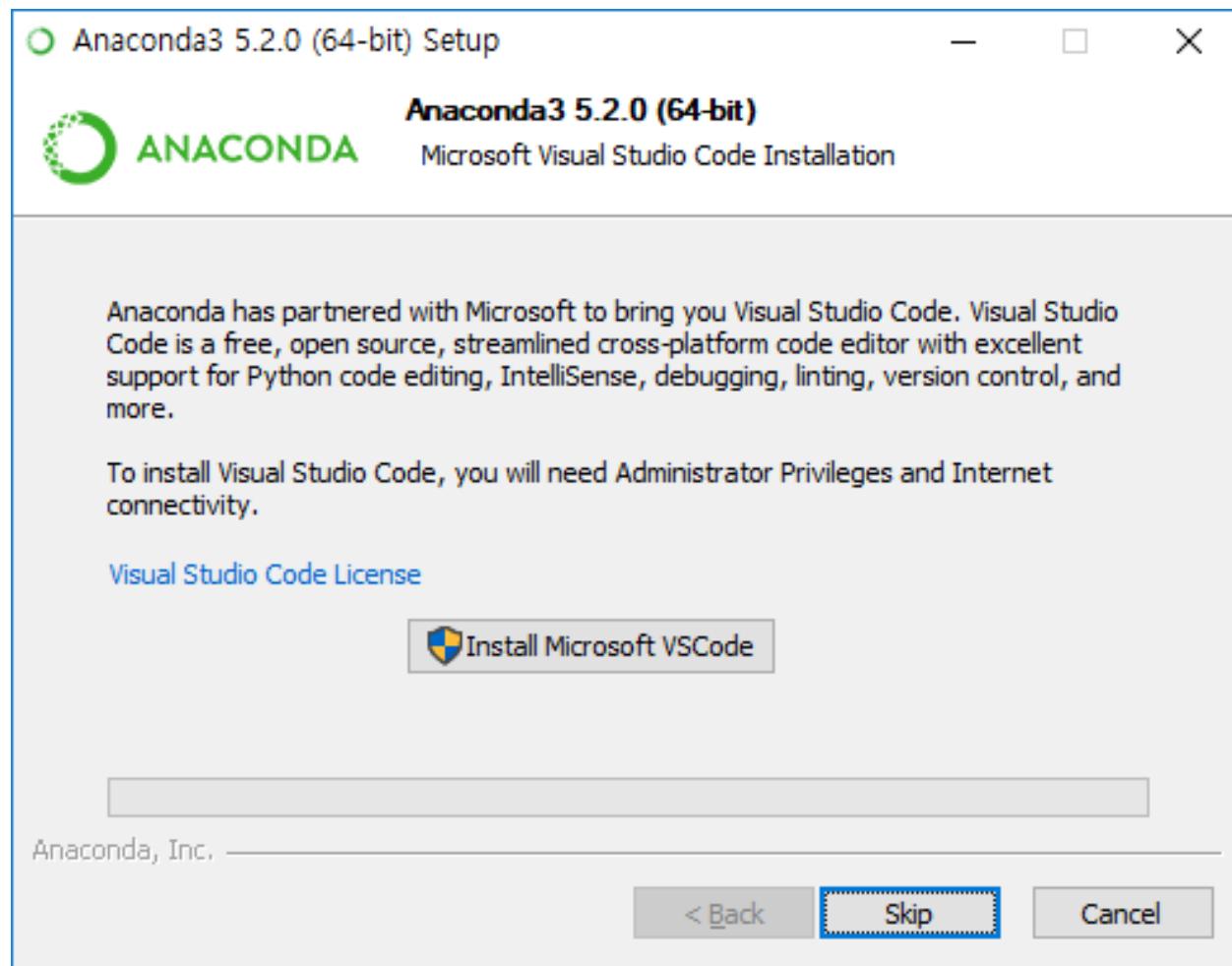
Install

Deep Learning

Pytorch

Install

- 아나콘다(ANACONDA) 설치



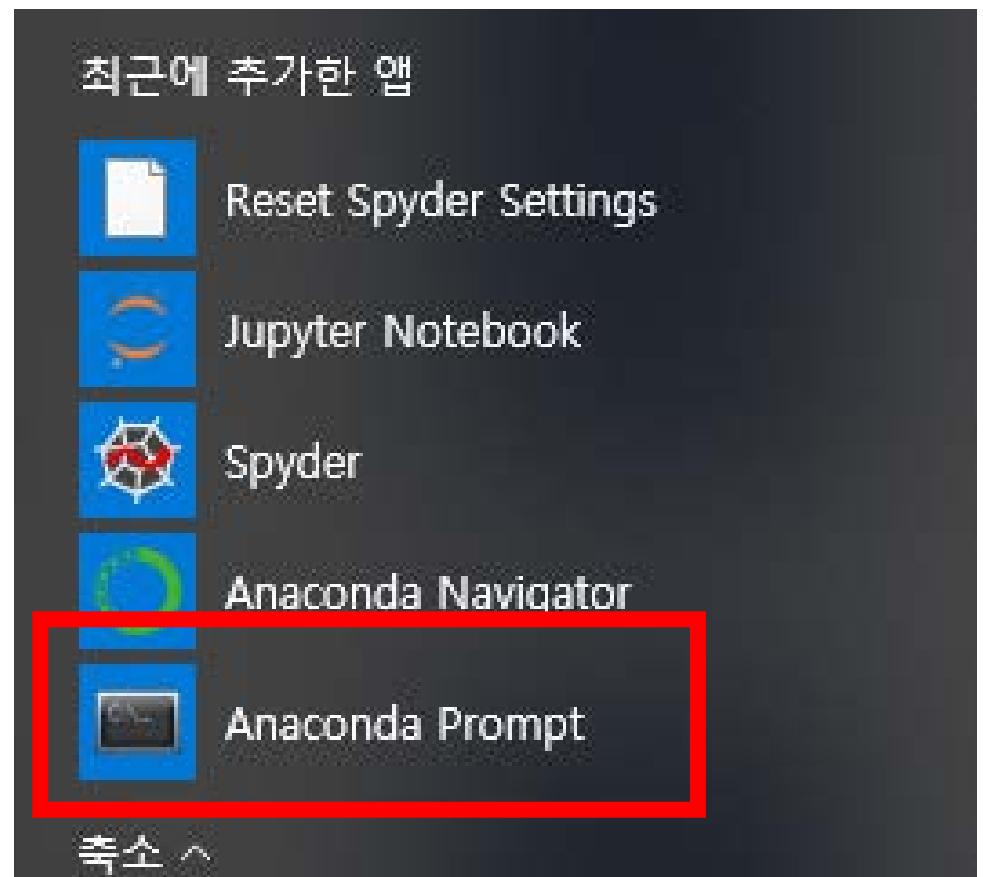
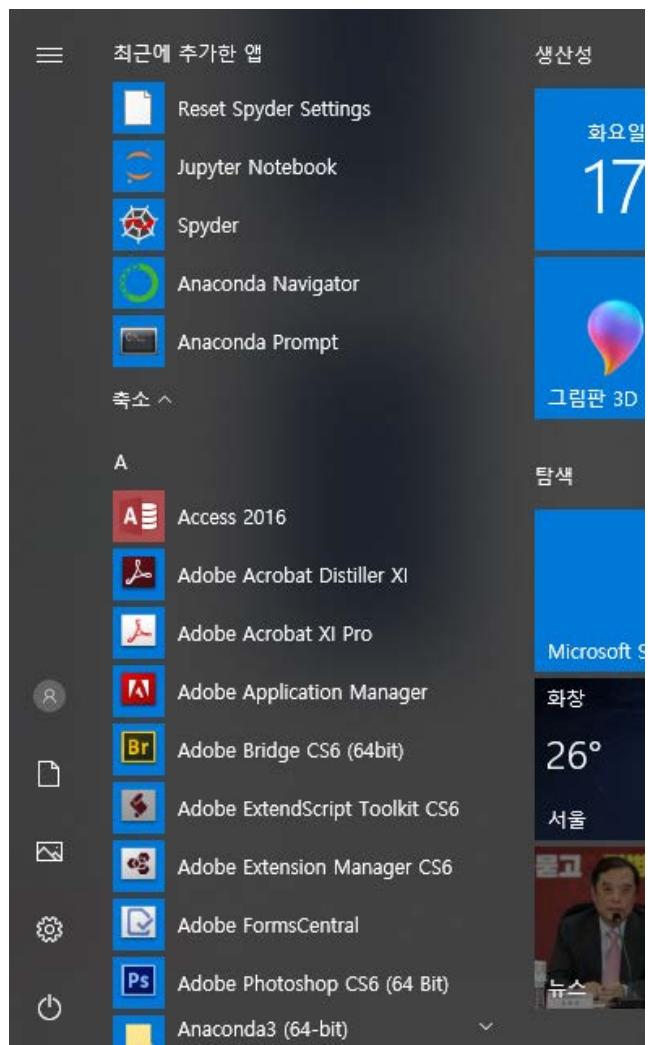
Install

Deep Learning

Pytorch

Install

■ 아나콘다(ANACONDA) 설치



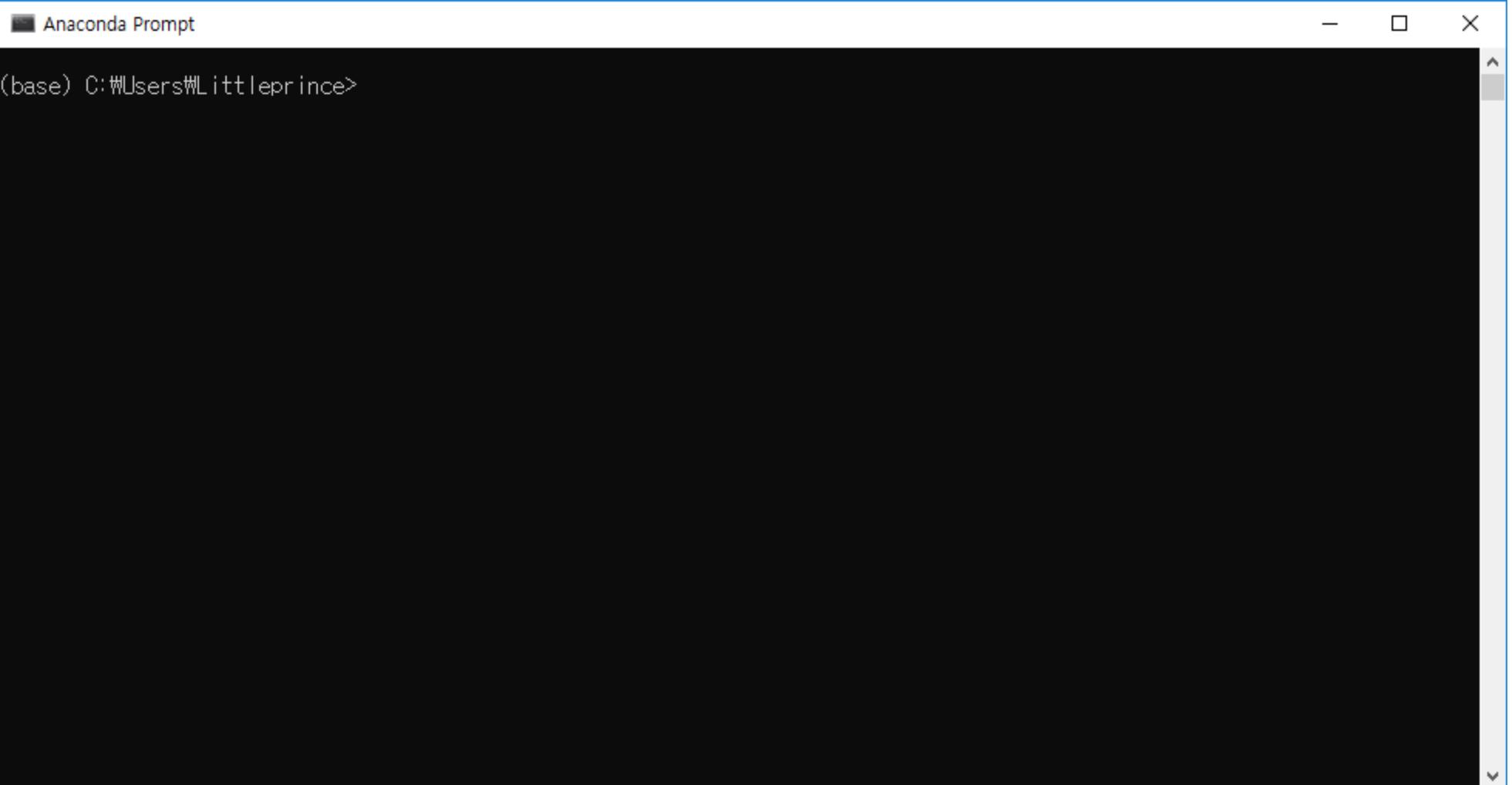
Install

Deep Learning

Pytorch

Install

- 아나콘다(ANACONDA) 실행



A screenshot of the Anaconda Prompt window. The title bar says "Anaconda Prompt". The command line shows "(base) C:\Users\Littleprince>". A scroll bar is visible on the right side of the window.

```
(base) C:\Users\Littleprince>
```

Install

Deep Learning

Pytorch

Install

- Pytorch 설치

<https://pytorch.org/>

The screenshot shows the PyTorch website's installation page for Windows. The interface includes dropdown menus for PyTorch Build (Stable 1.0), Your OS (Windows), Package (Conda), Language (Python 3.6), and CUDA (9.0). A red box highlights the command "conda install pytorch torchvision -c pytorch" located under the "Run this Command:" section.

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	C++
CUDA	8.0	9.0	10.0	None	

Run this Command:

```
conda install pytorch torchvision -c pytorch
```

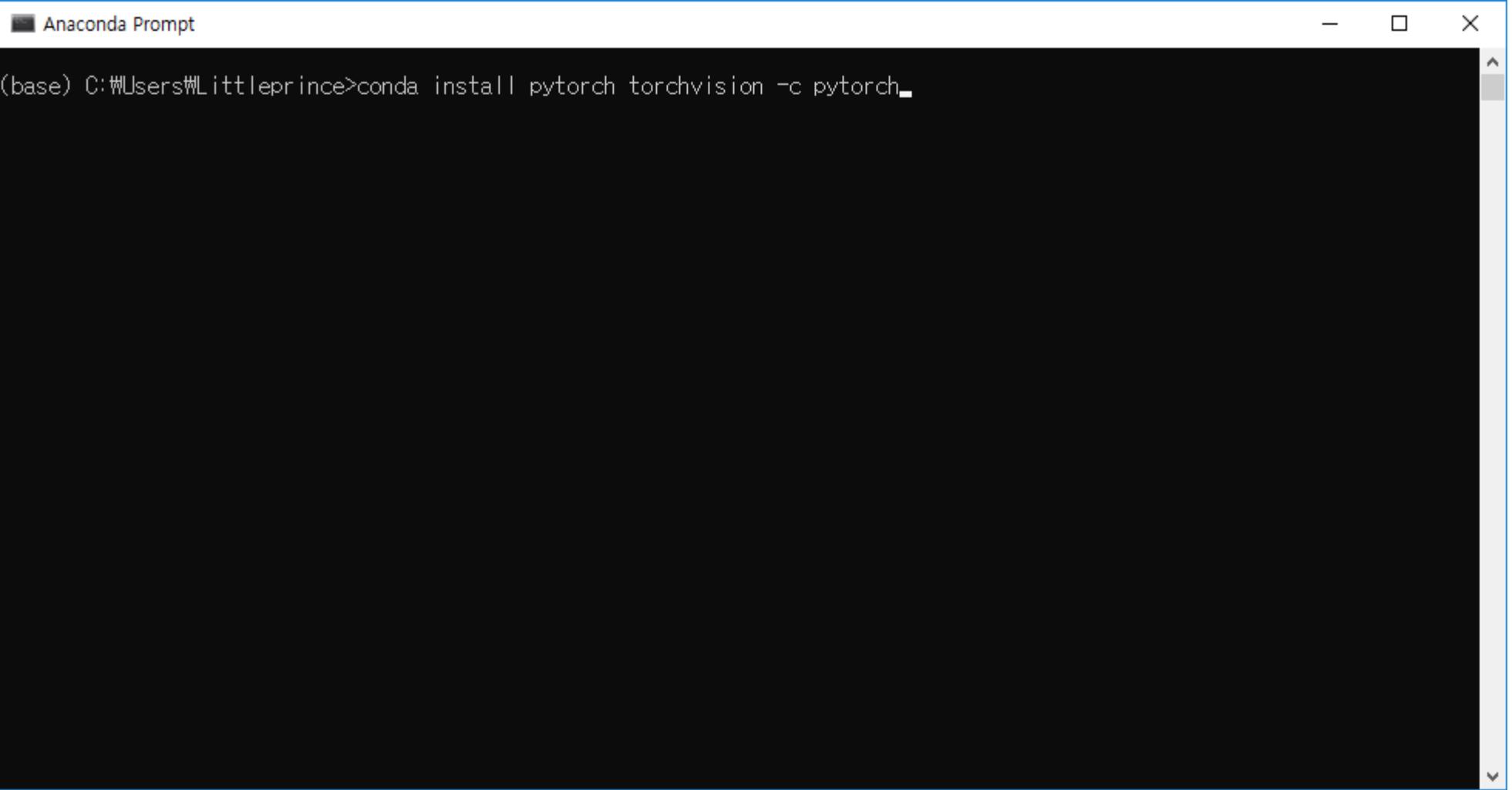
Install

Deep Learning

Pytorch

Install

- Pytorch 설치



Anaconda Prompt

```
(base) C:\Users\Littleprince>conda install pytorch torchvision -c pytorch
```

Install

Deep Learning

Pytorch

Install

- **Jupyter Notebook이란?**
 - 대화형 컴퓨팅을 위한 커맨드 쉘
 - 다양한 언어 지원
 - 브라우저를 통한 코드 공유 제공
 - 크롬(인터넷 브라우저) 설치 권장

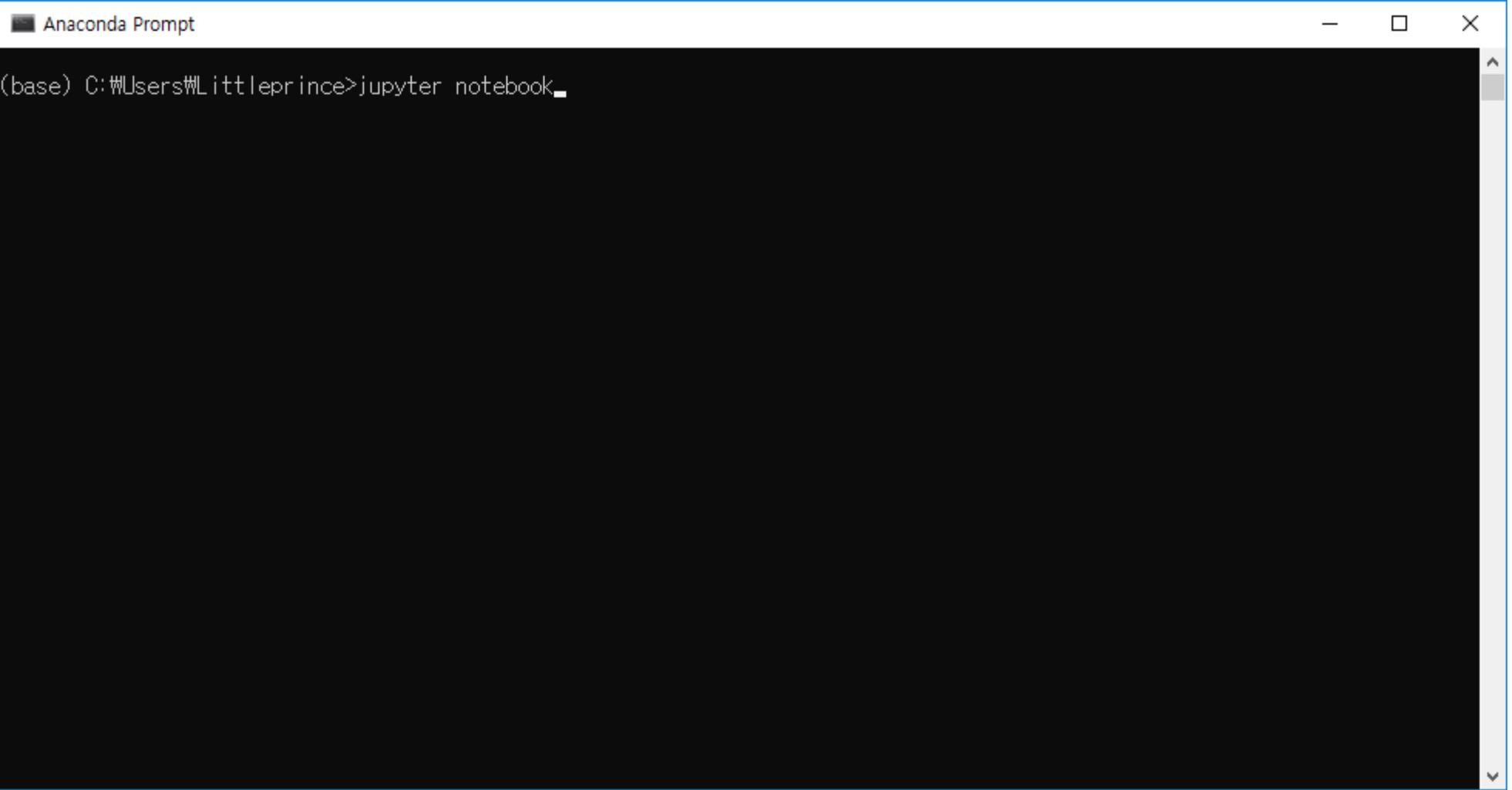
Install

Deep Learning

Pytorch

Install

- Jupyter Notebook 실행



Anaconda Prompt

(base) C:\Users\Littleprince>jupyter notebook

A screenshot of the Anaconda Prompt window. The title bar says "Anaconda Prompt". The command line shows "(base) C:\Users\Littleprince>jupyter notebook". The window has a standard OS X style with close, minimize, and maximize buttons in the top right corner.

Install

Deep Learning

Pytorch

Install

■ Jupyter Notebook 실행

```
선택 Anaconda Prompt - jupyter notebook

(base) C:\Users\Littleprince>jupyter notebook
[I 00:35:52.262 NotebookApp] JupyterLab beta preview extension loaded from C:\Users\Littleprince\Anaconda3\lib\site-packages\jupyterlab
[I 00:35:52.262 NotebookApp] JupyterLab application directory is C:\Users\Littleprince\Anaconda3\share\jupyter\lab
[I 00:35:52.871 NotebookApp] Serving notebooks from local directory: C:\Users\Littleprince
[I 00:35:52.871 NotebookApp] 0 active kernels
[I 00:35:52.871 NotebookApp] The Jupyter Notebook is running at:
[I 00:35:52.871 NotebookApp] http://localhost:8888/?token=db2881d72ec7db328127d710f20bb23f9be876fc268d8a88
[I 00:35:52.871 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 00:35:52.871 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=db2881d72ec7db328127d710f20bb23f9be876fc268d8a88&token=db2881d72ec7db328127d710f20bb23f9be876fc268d8a88
[I 00:35:53.949 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

Install

Deep Learning

Pytorch

Install

Jupyter Notebook 실행

```
선택 Anaconda Prompt - jupyter notebook

(base) C:\Users\Littleprince>jupyter notebook
[I 00:35:52.262 NotebookApp] JupyterLab beta preview extension loaded from C:\Users\Littleprince\Anaconda3\lib\site-packages\jupyterlab
[I 00:35:52.262 NotebookApp] JupyterLab application directory is C:\Users\Littleprince\Anaconda3\share\jupyter\lab
[I 00:35:52.871 NotebookApp] Serving notebooks from local directory: C:\Users\Littleprince
[I 00:35:52.871 NotebookApp] 0 active kernels
[I 00:35:52.871 NotebookApp] The Jupyter Notebook is running at:
[I 00:35:52.871 NotebookApp] http://localhost:8888/?token=db2881d72ec7db328127d710f20bb23f9be876fc268d8a88
[I 00:35:52.871 NotebookApp] Use Control-C to quit.
[00:35:52.871 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=db2881d72ec7db328127d710f20bb23f9be876fc268d8a88&token=db2881d72ec7db328127d710f20bb23f9be876fc268d8a88
[I 00:35:53.949 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

Install

Deep Learning

Pytorch

Install

■ Jupyter Notebook 실행

The screenshot shows the Jupyter Notebook web interface. At the top, there is a header with the Jupyter logo and navigation links for 'Quit' and 'Logout'. Below the header, there are three tabs: 'Files' (selected), 'Running', and 'Clusters'. A message 'Select items to perform actions on them.' is displayed above the file list. On the right side of the interface, there are sorting and filtering options: 'Name' (sorted by name), 'Last Modified' (sorted by last modified date), and 'File size'. The main area displays a list of files and folders in the current directory ('/'). The list includes:

File/Folder	Last Modified
3D Objects	3일 전
Anaconda3	4분 전
Contacts	3일 전
Desktop	13분 전
Documents	4분 전
Downloads	12분 전
Favorites	3일 전
Links	3일 전
Music	3일 전
OneDrive	한 달 전
Pictures	3일 전
Saved Games	3일 전
Searches	3일 전
Videos	3일 전

Install

Deep Learning

Pytorch

Install

Jupyter Notebook 실행



Deep Learning

Pytorch

Install

실습