

기계학습, 딥러닝

JRI Omni Studia (3/3) 2017.7.25

조남운

주제

- 기계학습 일반
- 딥 러닝 개관
- 보론: 웹에서 데이터 수집하기

준비물 (옵션)

- Python3 이 설치되어 있는 컴퓨터
 - Windows 10, bash 설치 권장
 - bash 설치: <http://sanghaklee.tistory.com/39>
 - Python3 설치: (bash상에서) \$ sudo apt-get install python3
- 인터넷 가능해야 함
- 따라해보기 위한 환경임. 필수요건이 아님!

들어가기에 앞서

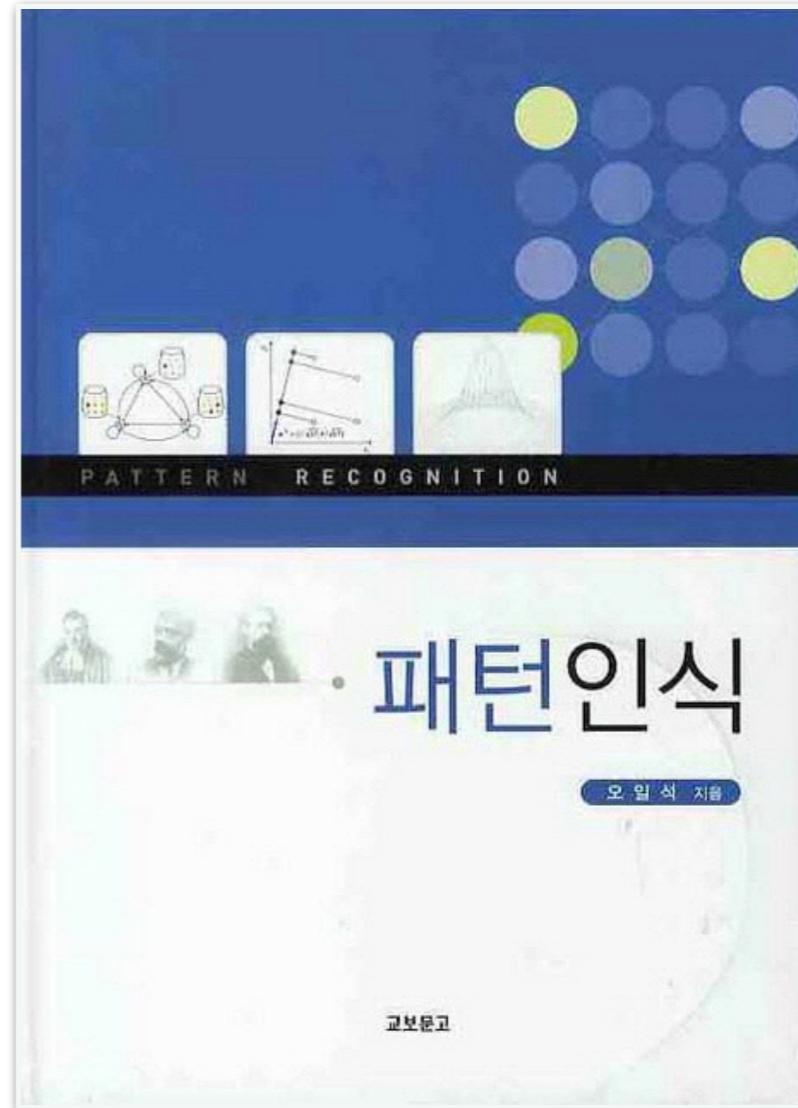
- 앞 두 주제와 달리, 이번 주제는 극히 최근에 공부하기 시작함
 - 탐색적 단계
- 본 슬라이드는 세 가지 서적에 의존
 - 사실상 이 슬라이드는 다음에 설명할 세 서적의 요약이라고 보아야 할 것임

참조 사이트

- 각 서적 홈페이지에 실습 자료가 수록되어 있음
- 본 슬라이드에 사용한 소스는 아래 사이트에서 다운로드하여 사용 가능
 - <https://github.com/z0nam/DeepLearning>
 - ** oTree (사회과학 실험 플랫폼) 강의록 사이트
 - <https://github.com/z0nam/oTreeBasic>

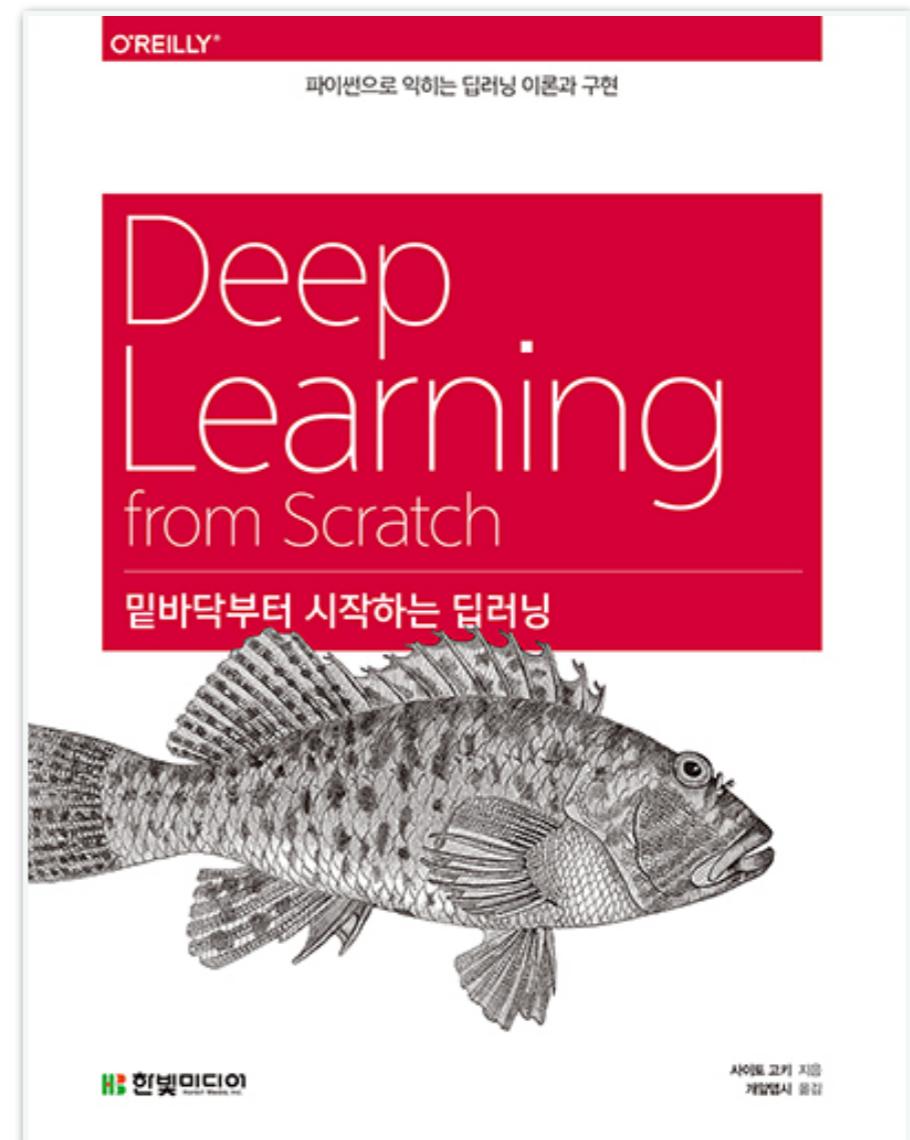
패턴인식

- 오일석 지음, 교보문고 2008
- 기계학습의 이론적 기초를 비교적 쉽고 친절하게 설명하고 있음
- 체계적으로 접근하려 할 경우 추천
 - 다만 최근의 인공지능 기술의 급격한 발전을 반영하지 못하고 있음은 감안해야 함



밑바닥부터 시작하는 딥러닝

- 사이토 고키 지음, 개앞맵시 옮김, 한빛미디어
- 딥러닝에 초점을 맞추어 기초이론을 설명하고 있음
- 당장의 응용은 어렵지만 딥러닝의 원리를 이해하는데 도움이 될 수 있음



머신러닝, 딥러닝 실전개발 입문

- 쿠지라 히코우즈쿠에 지음, 윤인성 옮김, 위키북스 2017
 - 기계학습, 딥러닝의 응용을 위한 예제 중심
 - 크롤링, 머신러닝, 딥러닝이 어떤 것인지 실제 실행해보고 싶은 경우 추천
 - 프로그래밍에 대한 약간의 사전지식 필요
 - 웹에서 데이터 수집 (스크레이핑)
 - 머신러닝
 - 딥러닝



본 슬라이드에 대해서

- 주제의 특성상 수학이나 프로그래밍을 아예 언급하지 않을 수는 없음
 - 이 부분에 대해서는 최소한으로 다루도록 노력 할 계획
- 슬라이드 분량이 다소 많음
 - 기술적 실습과 관련한 세부 내용 때문
 - 이와 관련한 슬라이드는 넘어가되, 향후 직접 검토해보고자 하는 분들을 위해 슬라이드로 남겨둠

기계학습 개관

목차

- 머신러닝에 대하여
 - 개관
 - 이론적 기초
- 실습
 - 머신러닝 프레임워크 scikit-learn
 - 머신러닝 실습: 손글씨 문자인식 (svm)

머신 러닝

- 컴퓨터로 학습을 구현하는 것
 - 예: 문자 인식, 음성 인식, ...
 - 딥러닝도 머신러닝의 일종
- 샘플 데이터를 통해 패턴 학습 --> 새로운 데이터에서 패턴을 분류
 - 예: 손글씨 인식 → 새로운 손글씨에서 글자를 추출

패턴

- 현상 속에서 반복적으로 나타나는 개념적 존재
- 패턴 인식: 사태 속에서 특정한 규칙을 지각하고 그것을 사태의 본성으로 이해하는 일
 - 사람(그리고 동물)에게는 쉽지만, 기계에는 (아직까지는) 어려운 일

Теория распознавания образов



(a) 누구인가?

인식 인식 인식
인식 신식 인식
인식 인식 인식
인식 인식 인식

(b) 무슨 글자인가?

(키, 몸무게, 높은 혈압, 낮은 혈압,
혈당, SGOT, SGPT)
=(169, 71, 130, 80, 94, 18, 26)

(c) 정상인가?

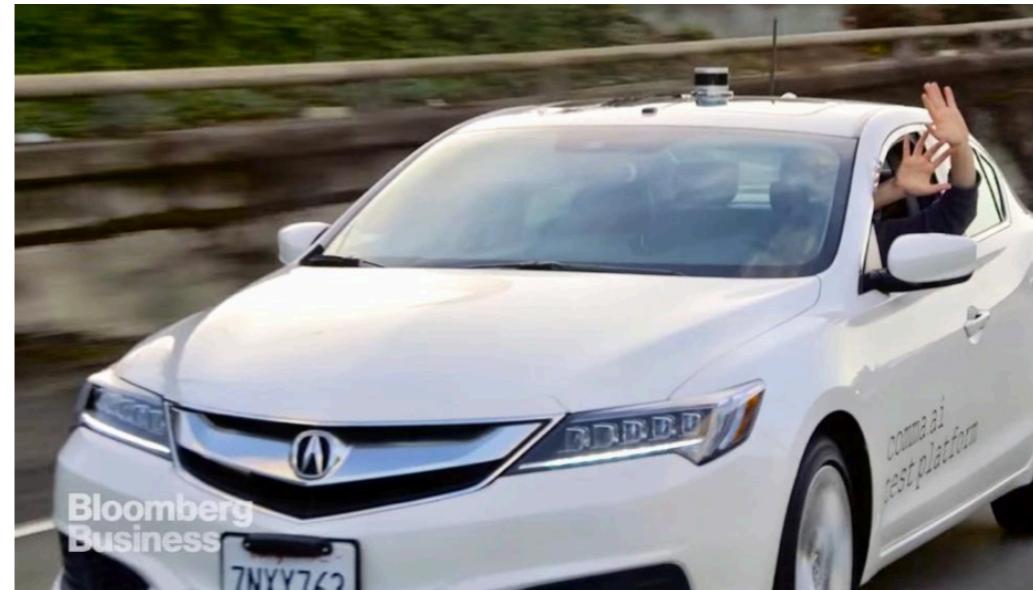
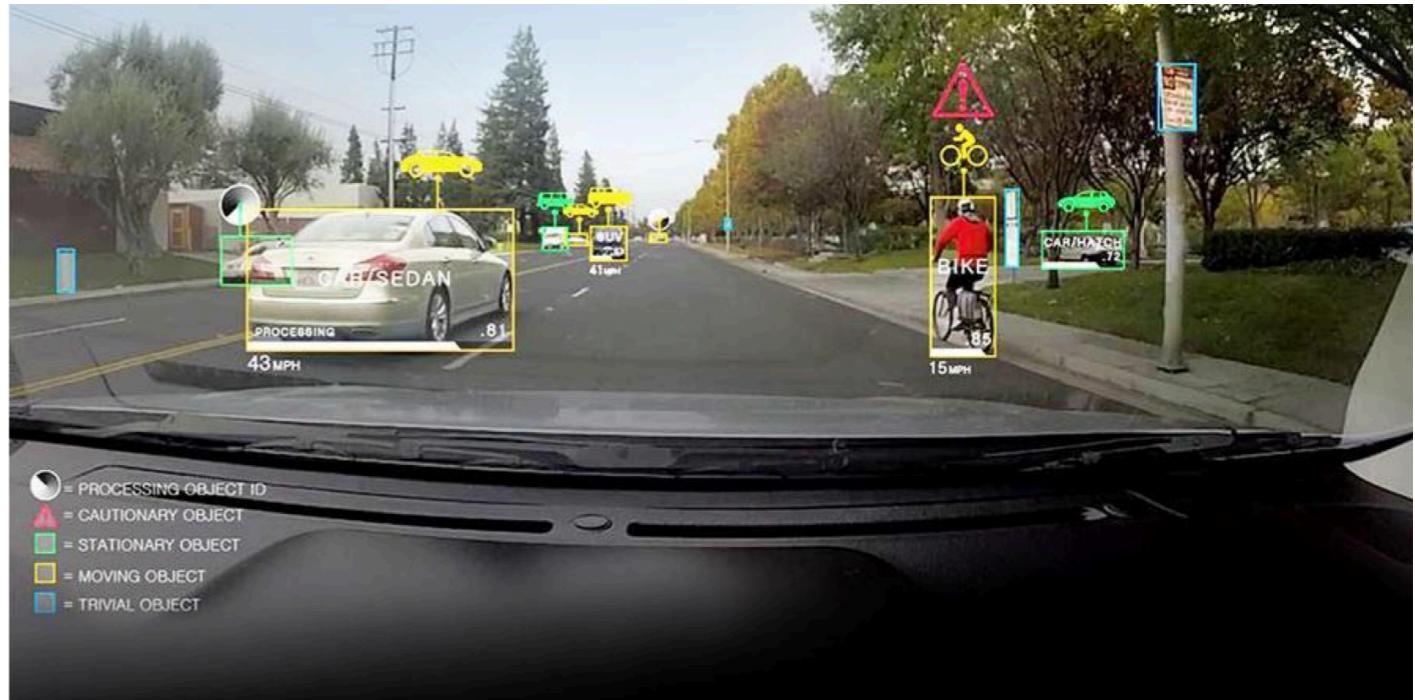
그림 1.1 인식 해보자.

출처: 패턴인식

패턴인식을 해봅시다!

기계의 패턴인식 사례

- 우편물 분류기
- 스마트폰 필기 입력기
- 지문 식별기
- 과속 단속기
- 청소로봇
- OCR
- 자동차 자율주행
- ...



https://www.slideshare.net/grigorysapunov/deep-learning-cases-text-and-image-processing?from_action=save

아내를 모자로 착각한 남자

- Oliver Sacks, The Man Who Mistook His Wife for a Hat (1985)
 - 심한 안면인식장애자의 에피소드
 - 얼굴을 구별하지 못하여 아내에게 큰 모자를 쓰워 식별
 - 인간에 있어서도 패턴 인식은 하드웨어(시각)의 문제가 아니라 소프트웨어(시각정보의 처리) 문제임



기계에게 질적인 일을 맡기기

- 컴퓨터는 수치연산만 할 수 있음
- 질적인 일:
 - 양적이지 않은 일
 - 분류하기, 특징 추출하기, ...
 - 양적인 일:
 - 계산적인 일
 - 계산의 연속으로 질적인 일을 해내는 것이 머신러닝의 목표

양적 변수와 질적 변수

- 컴퓨터는 숫자만을 다룰 수 있음: 숫자로 연결된 변수/상수만을 다룰 수 있음
 - 양적 변수: 숫자의 크기가 의미를 가짐
 - 예: 키, 몸무게, 나이
 - 질적으로 다른 양적 변수는 서로 더할 수 없음
 - 질적 변수: 숫자의 크기는 의미가 없음
 - 예: 성별코드, 학번
 - 연산 자체의 의미가 없음

질적으로 다른 숫자의 취급

- 질적으로 다른 것은 벡터에서 다른 차원으로 취급
 - 예: 빨강 = $(1, 0, 0, 0)$, 파랑 = $(0, 1, 0, 0)$
- 결국 기계학습은 특징을 나타내는 양으로 이루어진 벡터들의 공간에서 식별하고자 하는 것들을 잘 분리해낼 수 있는 일종의 구분선을 찾는 과정으로 볼 수 있음
 - 복잡한 함수로 이루어진 일종의 회귀 분석

머신 러닝의 구조

- 특징 추출: 데이터의 특징을 추출하여 벡터로 만드는 일
 - 예: 글자의 출현 빈도, 단어의 출현 빈도
- 학습: 벡터화된 학습 데이터를 통해 목적 함수의 매개 변수를 조절
- 평가: 학습의 성과를 평가
 - 학습 데이터 일부를 평가를 위한 데이터로 준비 함

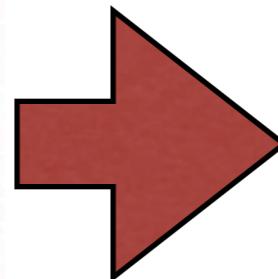
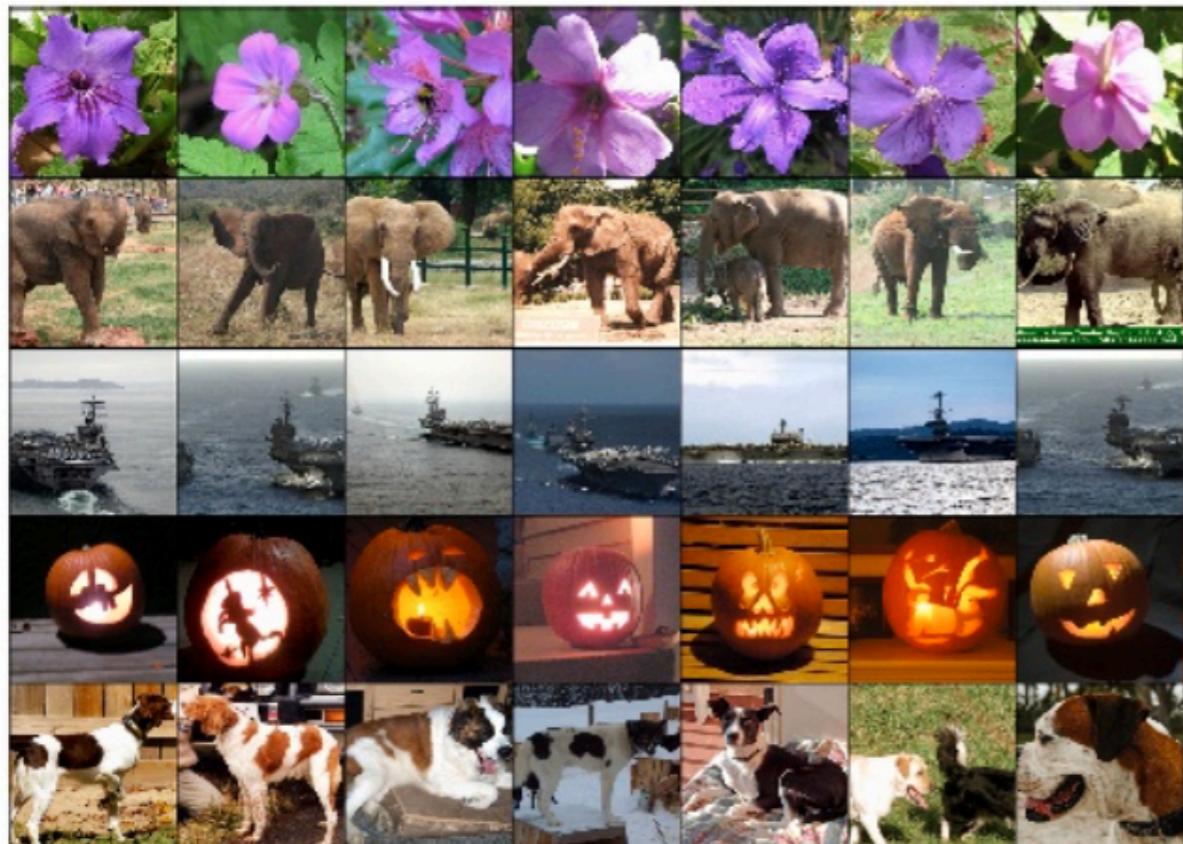
머신러닝의 종류

- 지도학습 (supervised learning)
- 자율학습 (unsupervised learning)
- 강화학습 (reinforcement learning)

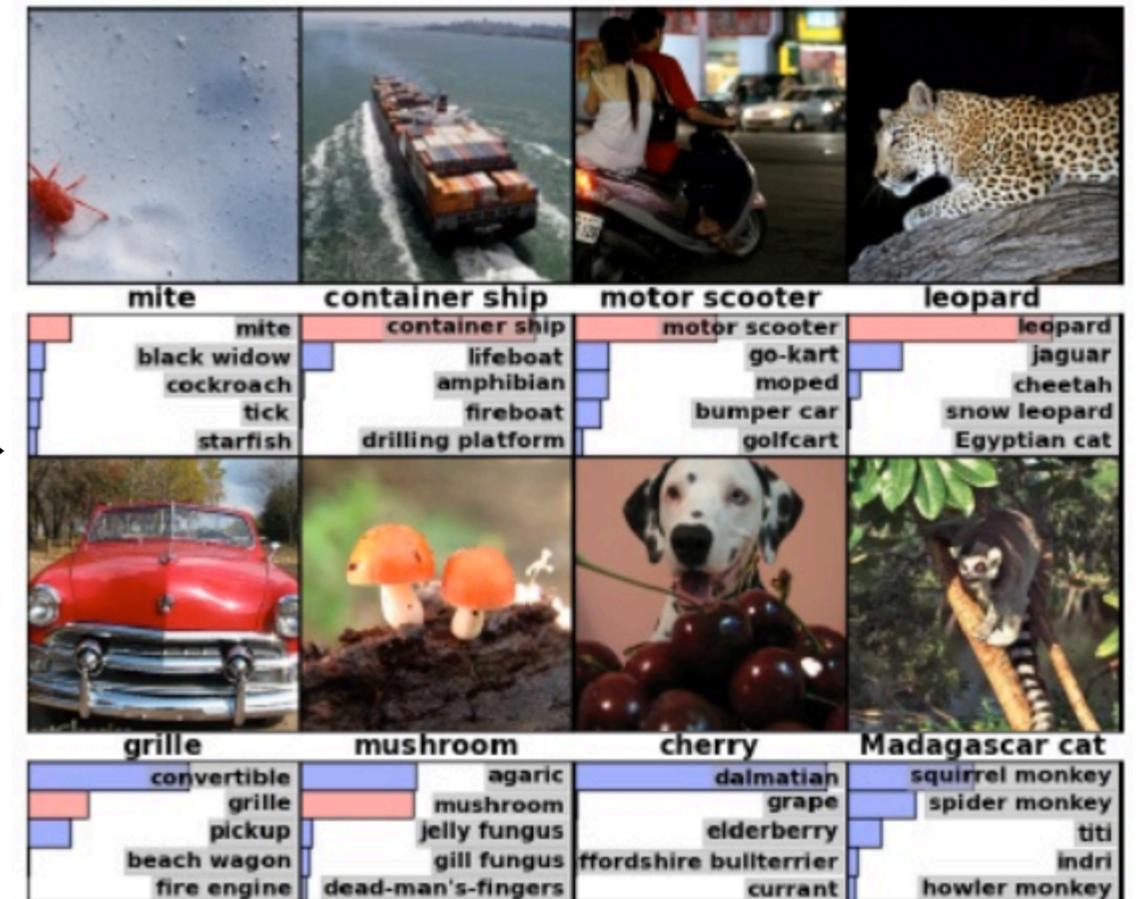
교사학습

- Supervised Learning
- 데이터와 함께 그 데이터의 정답 (예: 패턴분류)을 함께 학습
 - 예:
 - 글자 이미지 데이터와 그 글자의 종류,
 - 사물 이미지들과 그 사물의 이름

Supervised Learning



Pattern Recognition



<http://courses.cs.tamu.edu/choe/16spring/636/>



koush

@koush

Follow

i find it fascinating that image classifiers
can't tell the difference between fried
chicken and golden doodles



5:18 PM - 23 Jul 2017

38 Retweets 86 Likes

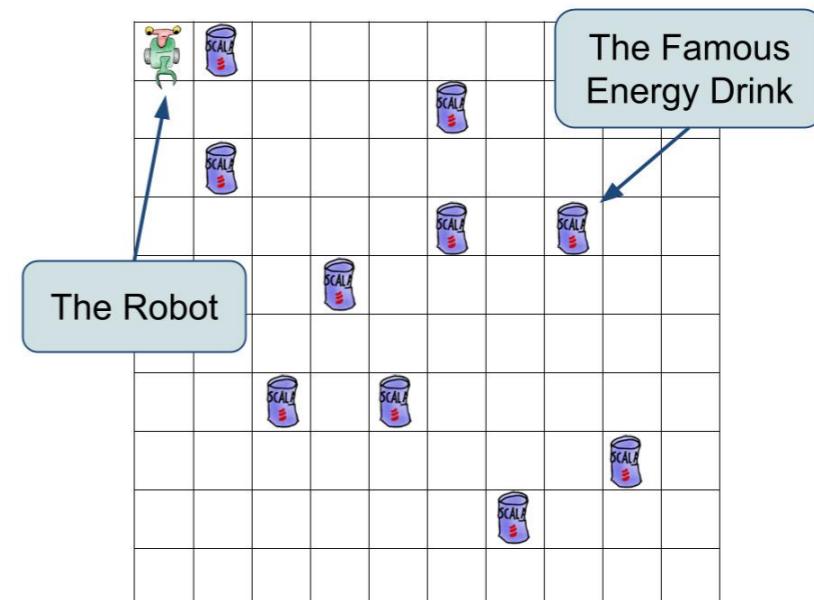


비교사학습

- Unsupervised Learning
- 사람도 패턴을 분류하기 어려운 본질적 구조를 확인할 때 사용
- 예
 - 클러스터 분석
 - 주성분 분석
 - 벡터 양자화
 - 자기 조직화

강화 학습

- Reinforcement Learning
- 행동 주체와 환경
 - ex) sugarscape, 로봇 미로찾기, 알파고 등
 - 행동을 수정해가면서 가장 보상 (성공함수 혹은 비용함수*(-1))을 크게 만들 행동 패턴을 찾아나가는 행위
 - 완전한 답을 제공해주는 것은 아님



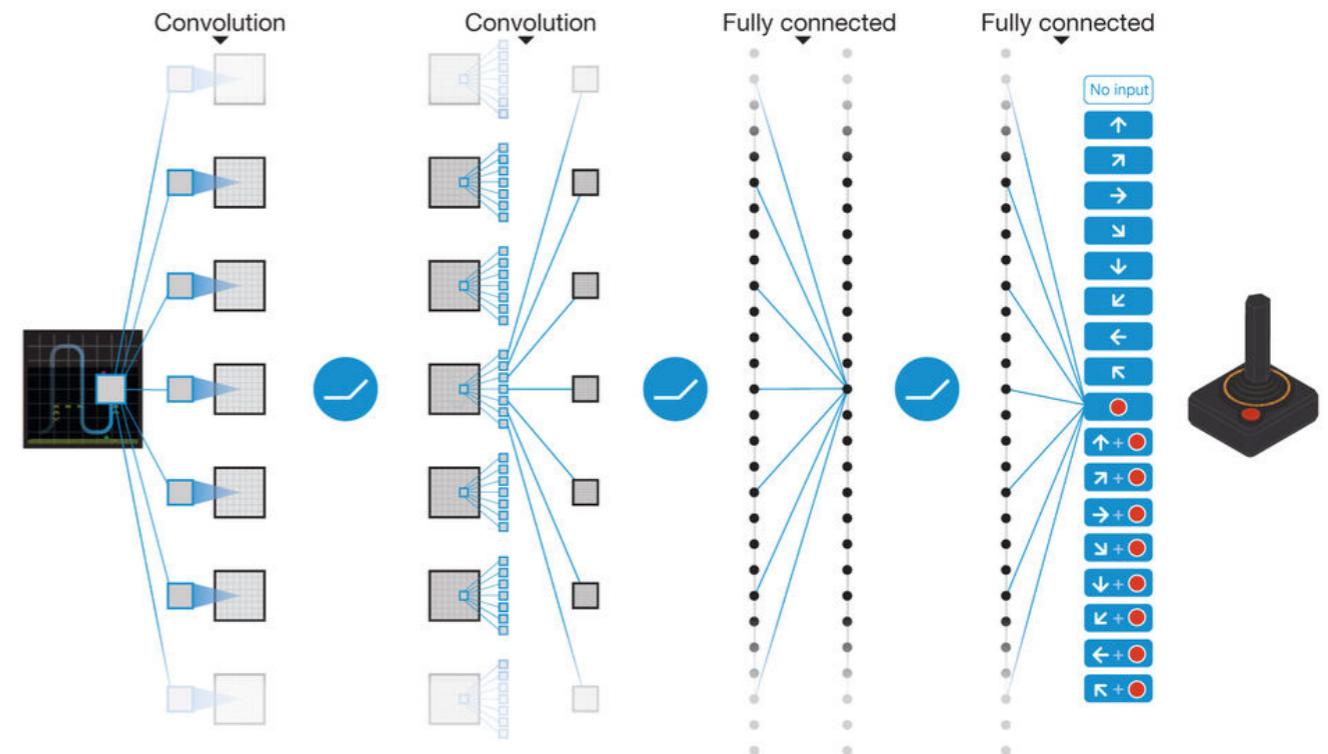
<http://blog.papauschek.com/wp-content/uploads/2014/02/Scala-Akka-Workshop-2.jpg>

머신러닝으로 할 수 있는 것

- 분류 (Classification)
- 그룹 나누기 (Clustering)
- 추천 (Recommendation)
- 회귀 (Regression)
- 차원 축소 (Dimension Reduction)

Deep Q-Network (DQN)

- 시각정보만을 제공
- 컴퓨터는 입력장치만을 사용 할 수 있음
- 게임점수를 보상함수로 사용
- 사전지식 없이 다양한 시도를 통해 높은 점수를 얻을 수 있는 패턴을 습득



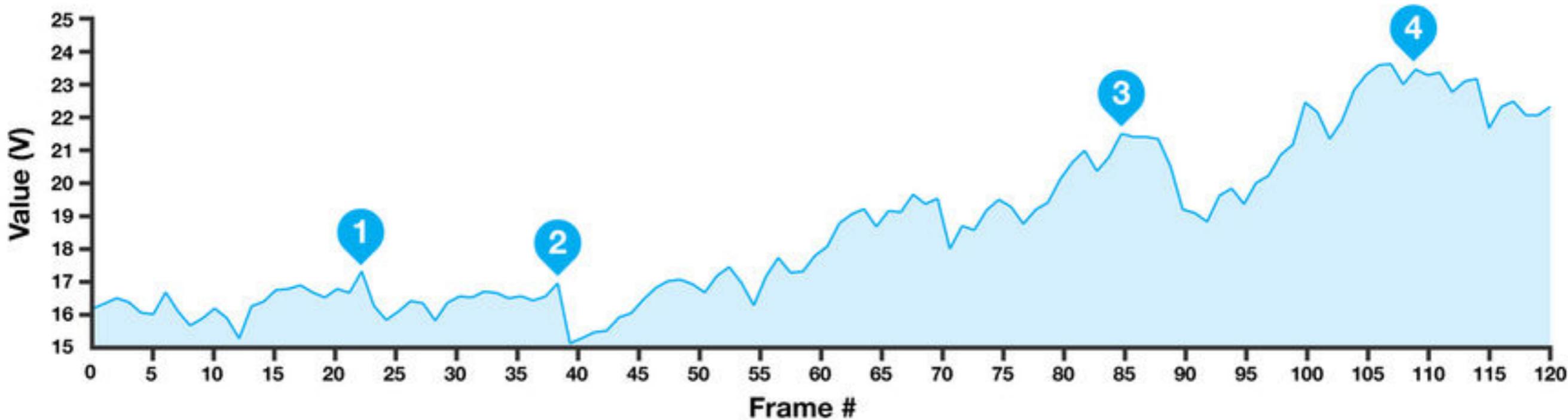
http://www.nature.com/nature/journal/v518/n7540/fig_tab/nature14236_F1.html

DQN의 예 (Video)

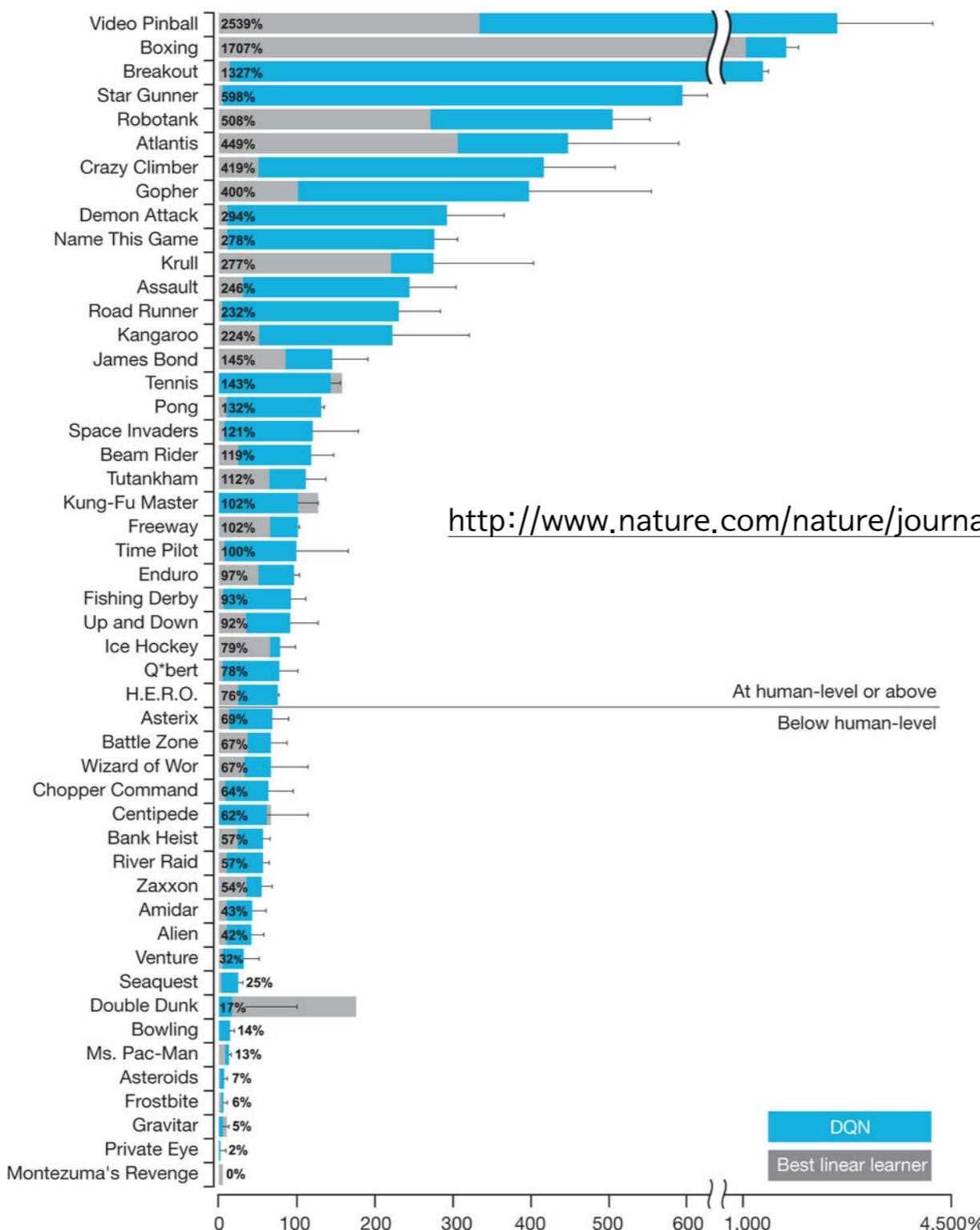


<https://youtu.be/V1eYniJ0Rnk>

a



Atari 2600 video game



기계학습: 이론

베이지언 확률론

- 기계학습의 기초가 되는 확률론
- 확률에 대한 경험주의적 접근방식
 - 통상적으로 배우는 확률론은 대체로 가능한 모든 사건의 확률 분포 (확률 변수)에 대한 이론
 - 사전적으로 규정된 확률 밀도 함수 (pdf)가 존재한다고 보는 관점
- 베이지언 추론: 관찰 (데이터)가 주어졌을 때 그것을 가장 잘 설명할 수 있는 분포를 추론

분별: 기초개념

$$\left. \begin{array}{l} P(\omega_1 | \mathbf{x}) > P(\omega_2 | \mathbf{x}) \text{이면, } \mathbf{x} \text{를 } \omega_1 \text{로 분류하고} \\ P(\omega_1 | \mathbf{x}) < P(\omega_2 | \mathbf{x}) \text{이면, } \mathbf{x} \text{를 } \omega_2 \text{로 분류하라.} \end{array} \right\} \quad (2.16)$$

- w_i : 분류기준 (추론해야 할 것)
- x : 주어진 값
- 이 조건부 확률은 베이지언 추론을 통해 훈련 데이터로부터 획득

$$P(\omega_i | \mathbf{x}) = \frac{p(\mathbf{x} | \omega_i)P(\omega_i)}{p(\mathbf{x})} = \frac{\text{우도 * 사전 확률}}{p(\mathbf{x})} \quad (2.17)$$

출처: 패턴인식

두 가지 학습 개념

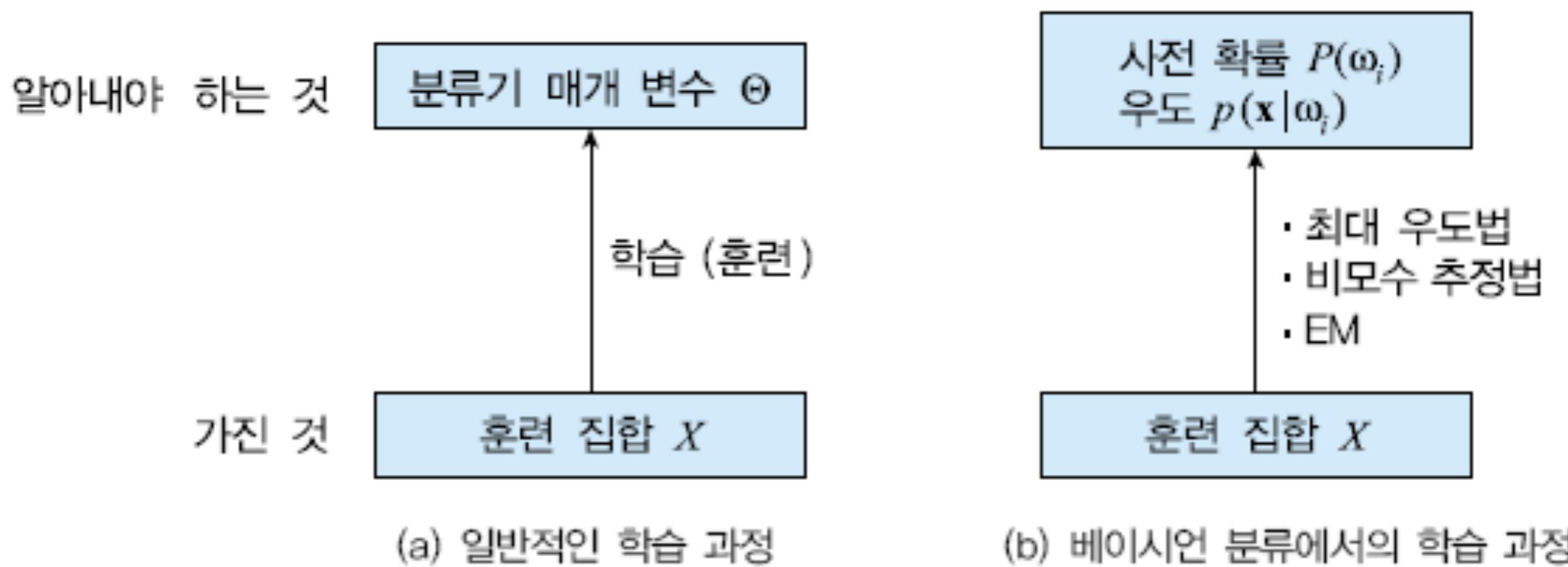


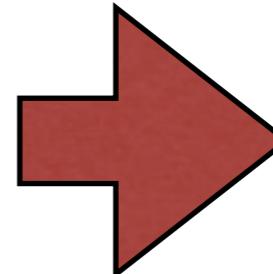
그림 3.1 일반적인 학습 과정과 베이시언 분류에서의 학습

출처: 패턴인식

기계학습(ML)에서의 확률 인식

- 컴퓨터에서 모든 것은 숫자
 - 질적인 분류마저도!
 - 모든 양적, 질적 특징을 숫자로 만드는 과정이 존재함:

6



0000**11**00
000**1**0000
00**1**00000
0**11**00000
11000**11**0
110000**11**
1100000**1**
11111110

기계학습 실습

scikit-learn

scikit-learn

- Python 머신러닝 프레임워크
 - 머신러닝에 필요한 클래스, 함수들을 제공
- \$ pip3 install -U scikit-learn
scipy matplotlib scikit-image
- \$ pip3 install -U pandas

nand-train.py

Train Data (#:4)

Preprocessing

- _sandbox/_ch04.ML/nand-train.py

Learning

Prediction

```
# NAND data  
  
nand_data = [  
    #P, Q, P NAND Q  
    [0,0,1],  
    [0,1,1],  
    [1,0,1],  
    [1,1,0]  
]  
  
# divide data and label for learning  
  
data = []  
label = []  
for row in nand_data:  
    p = row[0]  
    q = row[1]  
    r = row[2]  
    data.append([p,q])  
    label.append(r)  
  
# learning  
  
clf = svm.SVC()  
clf.fit(data,label)  
  
# prediction  
  
pre = clf.predict(data)  
print(" 예측결과:",pre)  
  
# display accuracy  
  
ok, total = 0,0  
for idx, answer in enumerate(label):  
    p = pre[idx]  
    if p == answer: ok +=1  
    total +=1  
print("정답률:",ok,"/",total,"=",ok/total)
```

Train Data

- P, Q : input (data)
- P NAND Q : output (label)
- 통상적으로 학습 데이터는 방대하기 때문에 이 부분은 별도의 데이터셋을 읽는 것으로 해결함
 - pandas

```
# NAND data
nand_data = [
    #P, Q, P NAND Q
    [0,0,1],
    [0,1,1],
    [1,0,1],
    [1,1,0]
]
```

Preprocessing

- 데이터를 변수로 할당하여 학습에 적합한 형태로 맞춤
 - normalisation
 - 자료형 일치

```
# divide data and label for learning
```

```
data = []
label = []
for row in nand_data:
    p = row[0]
    q = row[1]
    r = row[2]
    data.append([p,q])
    label.append(r)
```

Learning

- sklearn의 SVM 객체를 생성
 - 학습 모형
- fit() 으로 데이터와 레이블의 연결을 설명할 패턴을 찾음
 - fit의 첫번째 인자: 데이터(문제)
 - fit의 두번째 인자: 레이블(정답)
- 이 형태를 맞추기 위해 preprocessing을 한 것임

```
# learning
```

```
clf = svm.SVC()  
clf.fit(data,label)
```

Prediction

- data만 가지고 앞에서 fit한 모형에 대입하여 추론한 결과를 출력
- 통상적으로는 별도의 테스트 셋을 사용함

```
# prediction
```

```
pre = clf.predict(data)
print(" 예측결과:", pre)
```

정답률 계산

- 75%

```
# display accuracy

ok, total = 0,0
for idx, answer in enumerate(label):
    p = pre[idx]
    if p == answer: ok +=1
    total +=1
print("정답률:",ok,"/",total,"=",ok/total)
```

프레임워크의 사용

- pandas를 사용하여 작업 단순화
- metrics.accuracy_score 사용

```
import pandas as pd
from sklearn import svm, metrics

#NAND

nand_input = [
    [0,0,1],
    [0,1,1],
    [1,0,1],
    [1,1,0]
]

# preprocessing

nand_df = pd.DataFrame(nand_input)
nand_data = nand_df.ix[:,0:1] # data
nand_label = nand_df.ix[:,2] #label

# learning and prediction

clf = svm.SVC()
clf.fit(nand_data,nand_label)
pre = clf.predict(nand_data)

# accuracy

ac_score = metrics.accuracy_score(nand_label,pre)
print("정답률 = ",ac_score)
```

붓꽃 (iris) 품종 분류

- data:: (#150)
 - SepalLength (꽃받침 길이), SepalWidth (꽃받침 폭), PetalLength (꽃잎길이), PetalWidth (꽃잎폭) 으로
- label::
 - Iris-setosa, Iris-versicolor, Iris-virginica 품종을 분류
 - <https://github.com/pandas-dev/pandas/blob/master/pandas/tests/data/iris.csv>

1	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
2	5.1	3.5	1.4	0.2	Iris-setosa
3	4.9	3.0	1.4	0.2	Iris-setosa
4	4.7	3.2	1.3	0.2	Iris-setosa
5	4.6	3.1	1.5	0.2	Iris-setosa
6	5.0	3.6	1.4	0.2	Iris-setosa
7	5.4	3.9	1.7	0.4	Iris-setosa
8	4.6	3.4	1.4	0.3	Iris-setosa
9	5.0	3.4	1.5	0.2	Iris-setosa
10	4.4	2.9	1.4	0.2	Iris-setosa
11	4.9	3.1	1.5	0.1	Iris-setosa
12	5.4	3.7	1.5	0.2	Iris-setosa
13	4.8	3.4	1.6	0.2	Iris-setosa
14	4.8	3.0	1.4	0.1	Iris-setosa
15	4.3	3.0	1.1	0.1	Iris-setosa
16	5.8	4.0	1.2	0.2	Iris-setosa
17	5.7	4.4	1.5	0.4	Iris-setosa

iris-train2.py

- NAND-train.py와의 차이

- 데이터를 외부파일에서 읽음
- 훈련용, 테스트용 셋을 분리함

```
import pandas as pd
from sklearn import svm, metrics
from sklearn.model_selection import train_test_split

# reading data
csv = pd.read_csv("_data/iris.csv")

# preprocessing
csv_data = csv[["SepalLength", "SepalWidth", "PetalLength", "PetalWidth"]]
csv_label = csv[["Name"]]

# separating learning data and test data
train_data, test_data, train_label, test_label = train_test_split(csv_data, csv_label)

# learning and prediction
clf = svm.SVC()
clf.fit(train_data, train_label)
pre = clf.predict(test_data)

# getting accuracy
acc_score = metrics.accuracy_score(test_label, pre)
print("정답률: ", acc_score)
```

MNIST 손글씨 숫자 인식

MNIST database

- <http://yann.lecun.com/exdb/mnist/>
- train-images-idx3-ubyte.gz: training set images (9912422 bytes)
- train-labels-idx1-ubyte.gz: training set labels (28881 bytes)
- t10k-images-idx3-ubyte.gz: test set images (1648877 bytes)
- t10k-labels-idx1-ubyte.gz: test set labels (4542 bytes)

THE MNIST DATABASE of handwritten digits

[Yann LeCun](#), Courant Institute, NYU
[Corinna Cortes](#), Google Labs, New York
[Christopher J.C. Burges](#), Microsoft Research, Redmond

se of handwritten digits, available from this page, has a training set of 60,000 examples, i
t is a subset of a larger set available from NIST. The digits have been size-normalized and

e for people who want to try learning techniques and pattern recognition methods on real
efforts on preprocessing and formatting.

able on this site:

[.3-ubyte.gz](#): training set images (9912422 bytes)
[.1-ubyte.gz](#): training set labels (28881 bytes)
[.ubyte.gz](#): test set images (1648877 bytes)
[.ubyte.gz](#): test set labels (4542 bytes)

our browser may uncompress these files without telling you. If the files you download
y have been uncompressed by your browser. Simply rename them to remove the .gz exten
' application can't open your image files". These files are not in any standard image forma
ple) program to read them. The file format is described at the bottom of this page.



Data preprocessing

- download (mnist-download.py)
- uncompressing (mnist-download.py)
- binary to csv (mnist-tocsv.py)
- learning (mnist-train.py)

Deep Learning

목차

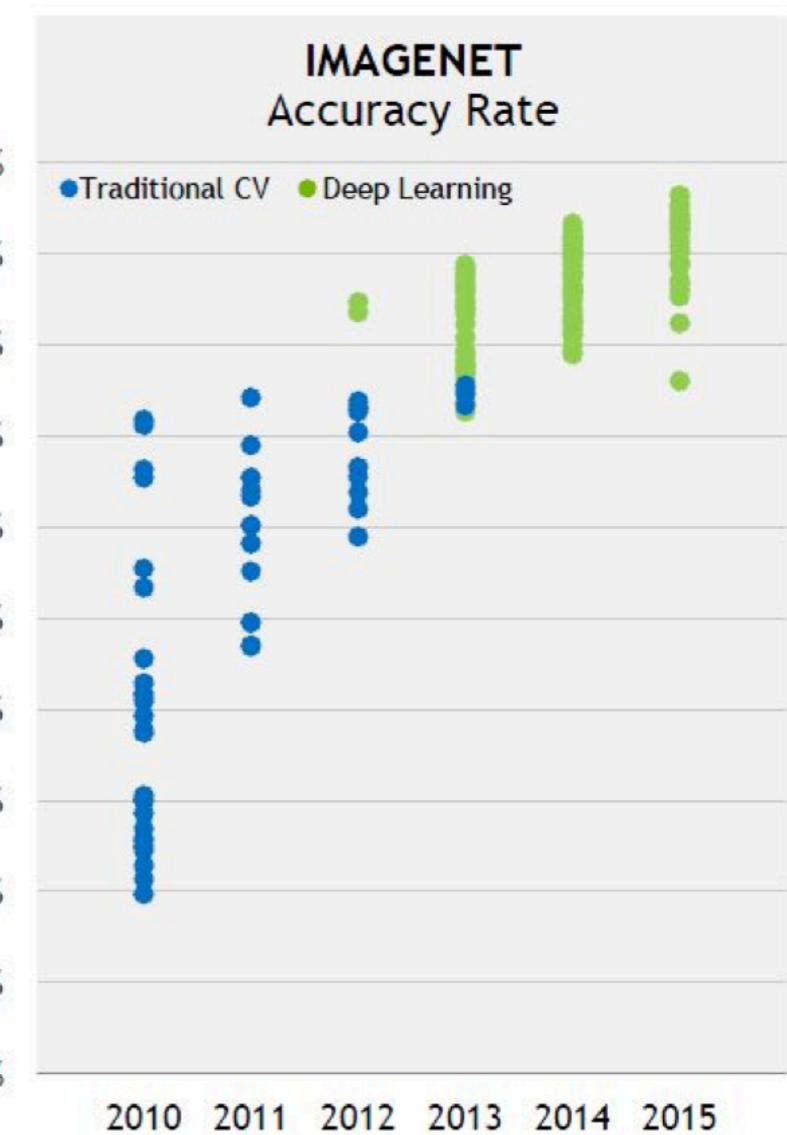
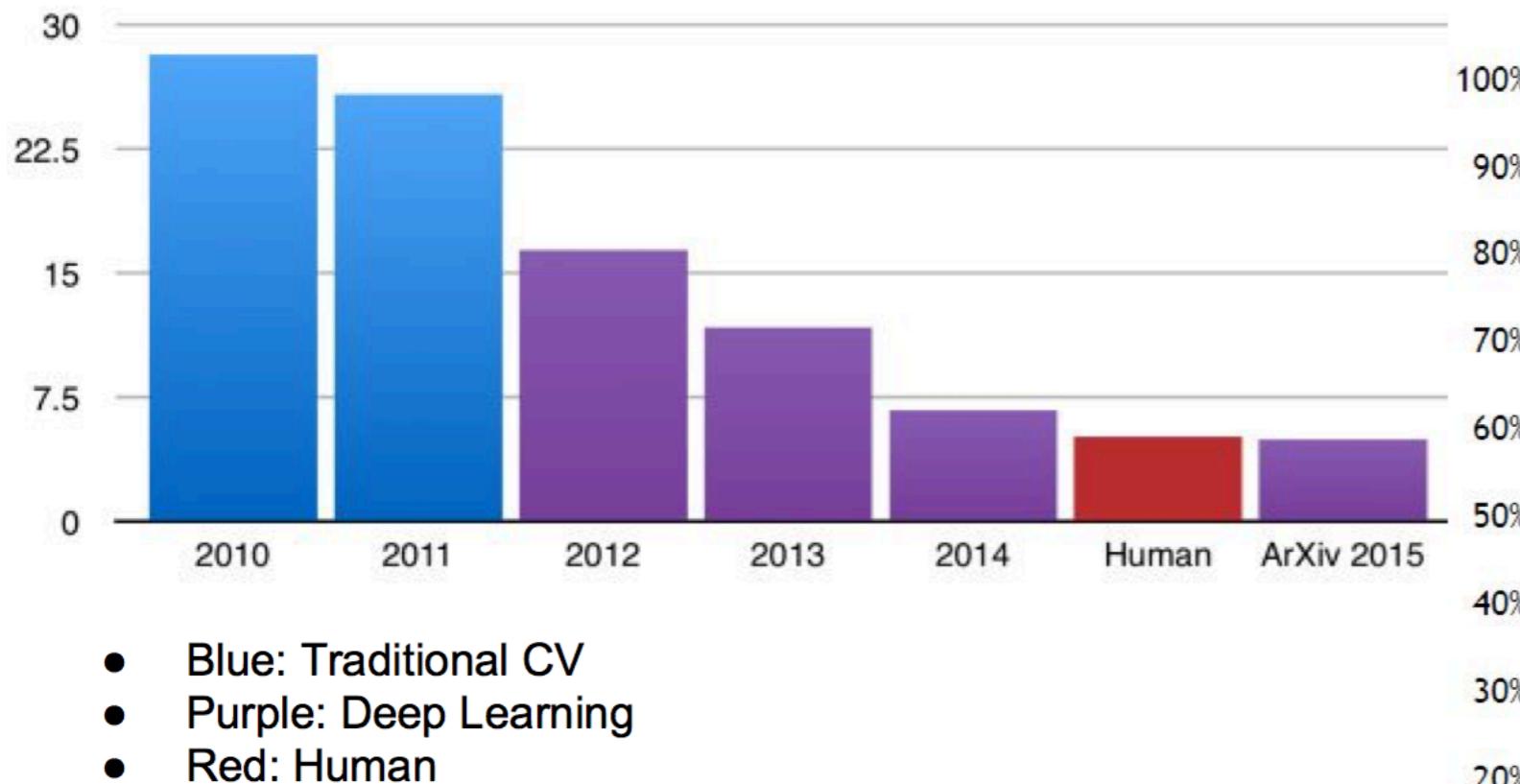
- 딥러닝 개요
 - 딥러닝이란?
 - 이론적 기초
- 실습
 - TensorFlow
 - TensorBoard
 - Keras

Deep Learning

- 머신러닝의 일종
- 퍼셉트론 (수학적으로 구현한 뉴런 모형)을 여러 층으로 연결한 인공신경망을 이용한 기계학습 기법
- cycle 없는 뉴럴 네트워크
 - 수리적 용이성
- “Deep Neural Network” (DNN): 3중 이상의 Hidden Layer

DL이 주목받는 이유

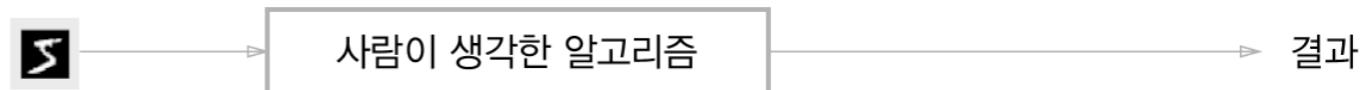
ILSVRC top-5 error on ImageNet



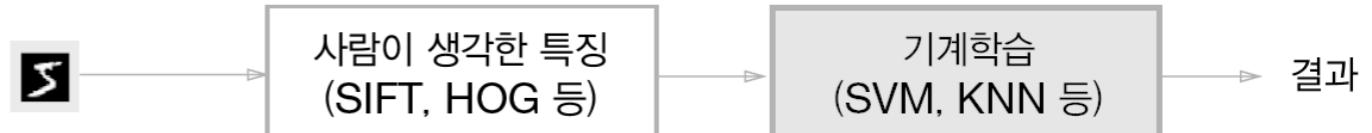
https://www.slideshare.net/grigorysapunov/deep-learning-cases-text-and-image-processing?from_action=save

DL이 주목받는 이유 2

- 전통적 ML은 데이터의 특징, 학습 방식 등은 연구자가 정해야 했음



- 회색영역: 인간이 개입하지 않는 영역



- DL은 태스크의 성격과 관계 없이 주어진 데이터 속에서 문제의 패턴을 발견하려 시도 함



- end-to-end 학습

사이트 고키 (2017)

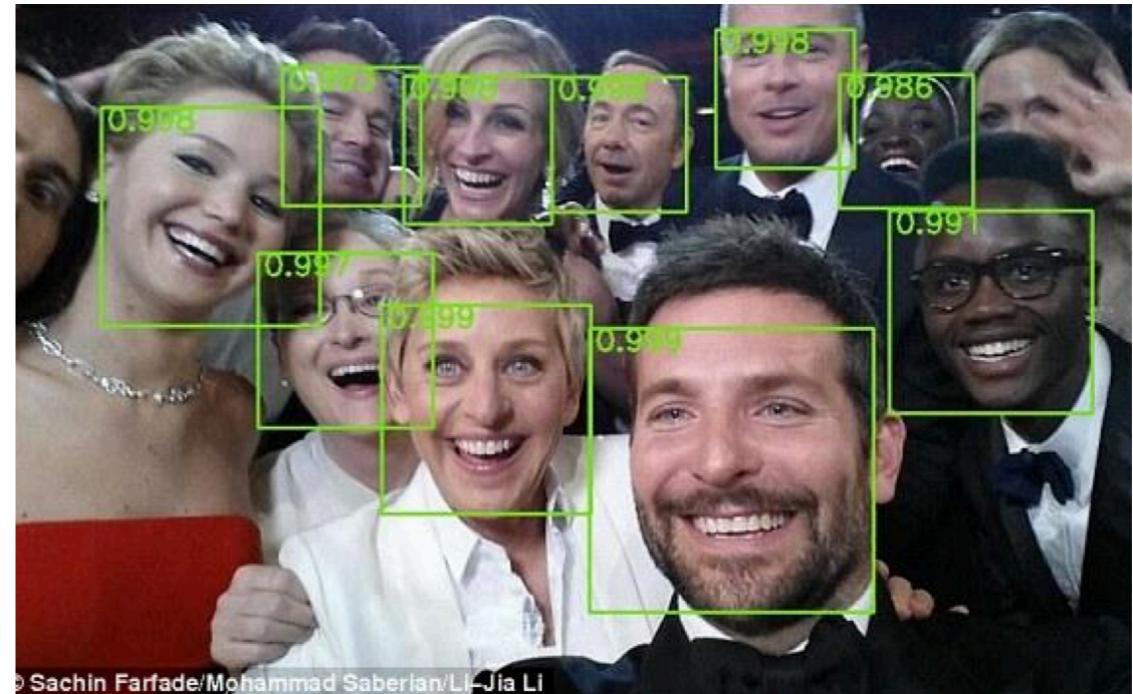
DL 응용사례

- Games with DQN (앞에서 다룸)
- 자율주행
- 사물검출
- 분할 segmentation
- Caption creation
- Image Creation

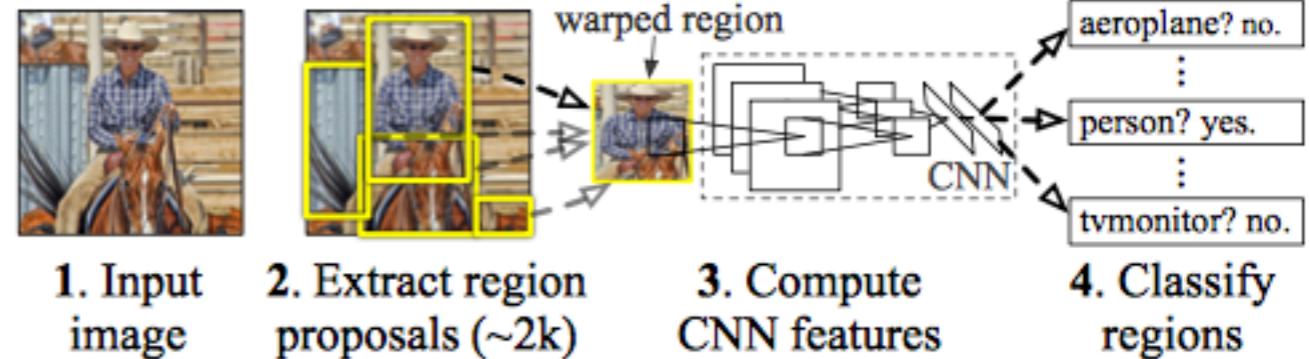
Object Detection

- 이미지 속에 담긴 사물의 위치와 종류를 알아내는 것
- 사물 인식보다 더 어려운 일
 - 이미지 내에서 사물의 범위와 위치를 알아내야 함
 - 여러 서로 다른 object들이 한 이미지에 존재할 수 있음
- Regions with Convolutional Neural Network (R-CNN)

- Girshick, R., Donahue, J., & Darrell, T. (2014).
- <https://arxiv.org/pdf/1311.2524.pdf>



R-CNN: Regions with CNN features



Segmentation

- <https://youtu.be/ZJMtDRbqH40>
- Fully Convolutional Network (FCN)
 - Shelhamer, E., Long, J., & Darrell, T. (2017).
 - https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

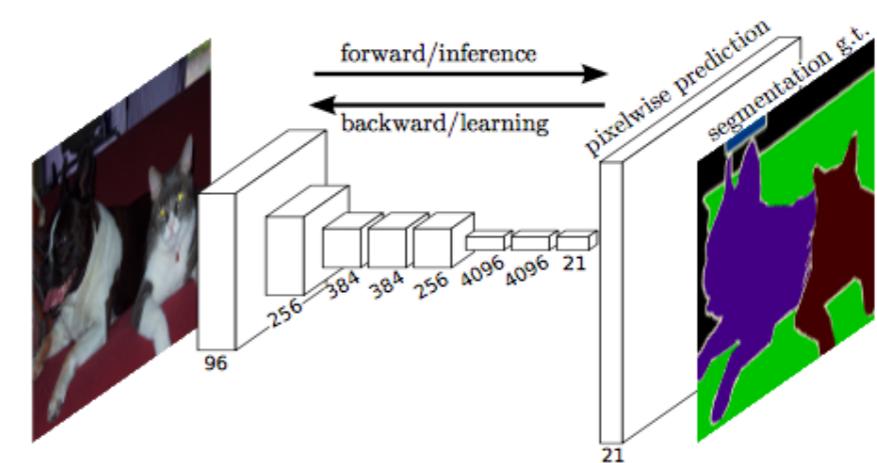
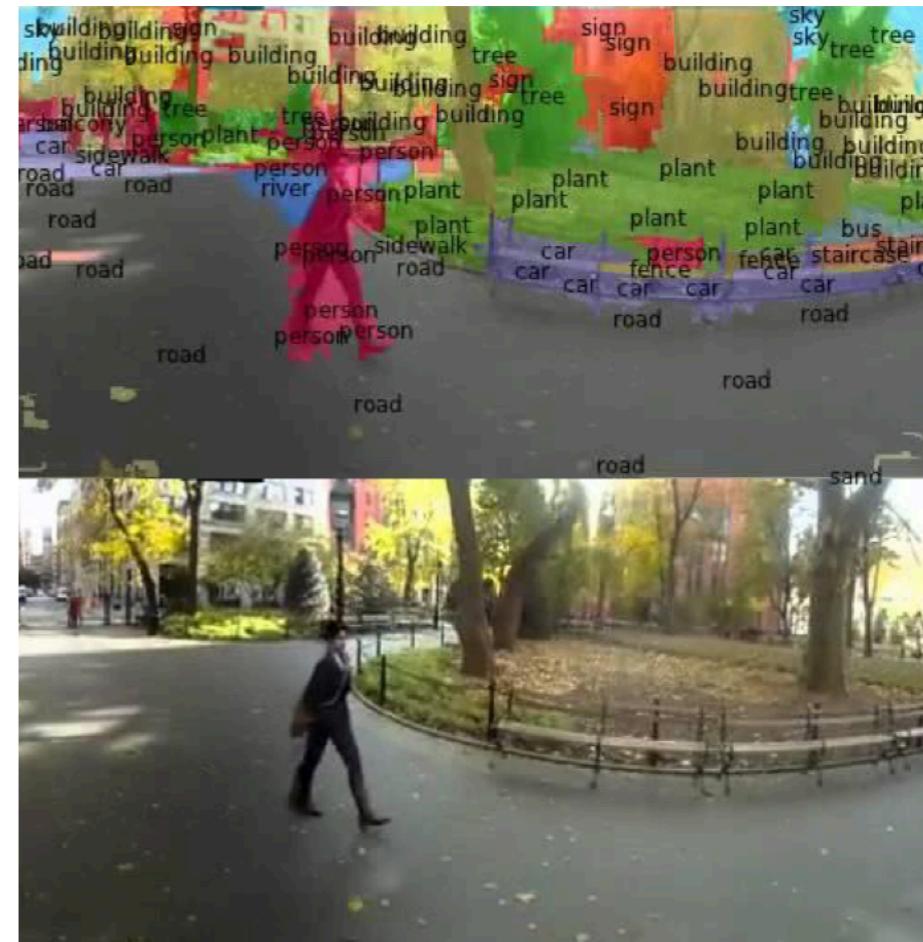
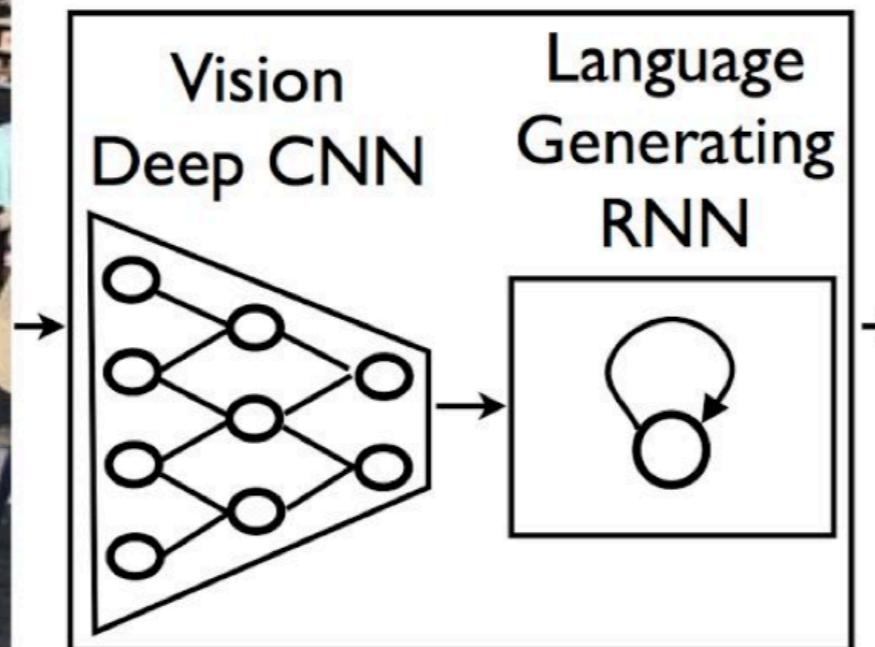


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

Caption Generation



A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

NIC (Neural Image Caption) = CNN + RNN

Vinyals et al. (2015)

<https://arxiv.org/abs/1411.4555>

Image Creation

- Deep Convolutional Generative Adversarial Networks (DCGAN)
- Radford, A., Metz, L., & Chintala, S. (2015).



Figure 3: Generated bedrooms after five epochs of training. under-fitting via repeated noise textures across multiple sam the beds.

DL: 이론적 기초

기본적 아이디어

- 뇌의 구조가 매우 간단한 뉴런의 매우 복잡한 얹힘으로 이루어짐을 발견
 - 인간 뇌의 뉴런수: 1000 억
 - 인간 뇌의 연결수(시냅스): 1000조 (평균 약 1만 link / 뉴런)
- 단일 뉴런의 구조는 매우 간단하고 논리적으로 작동함



<http://kr.brainworld.com/BrainEducation/487>

Model of Neuron

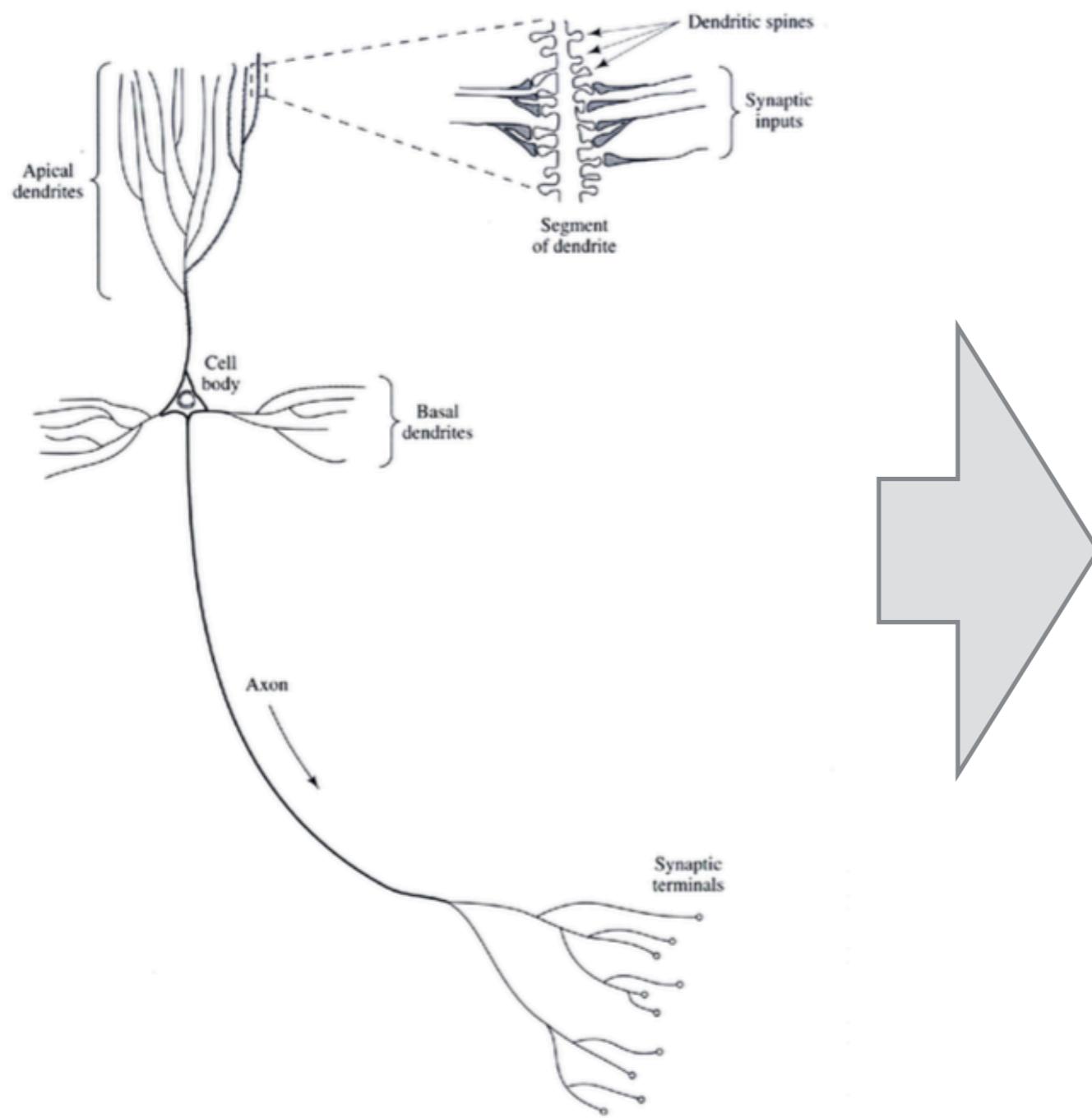
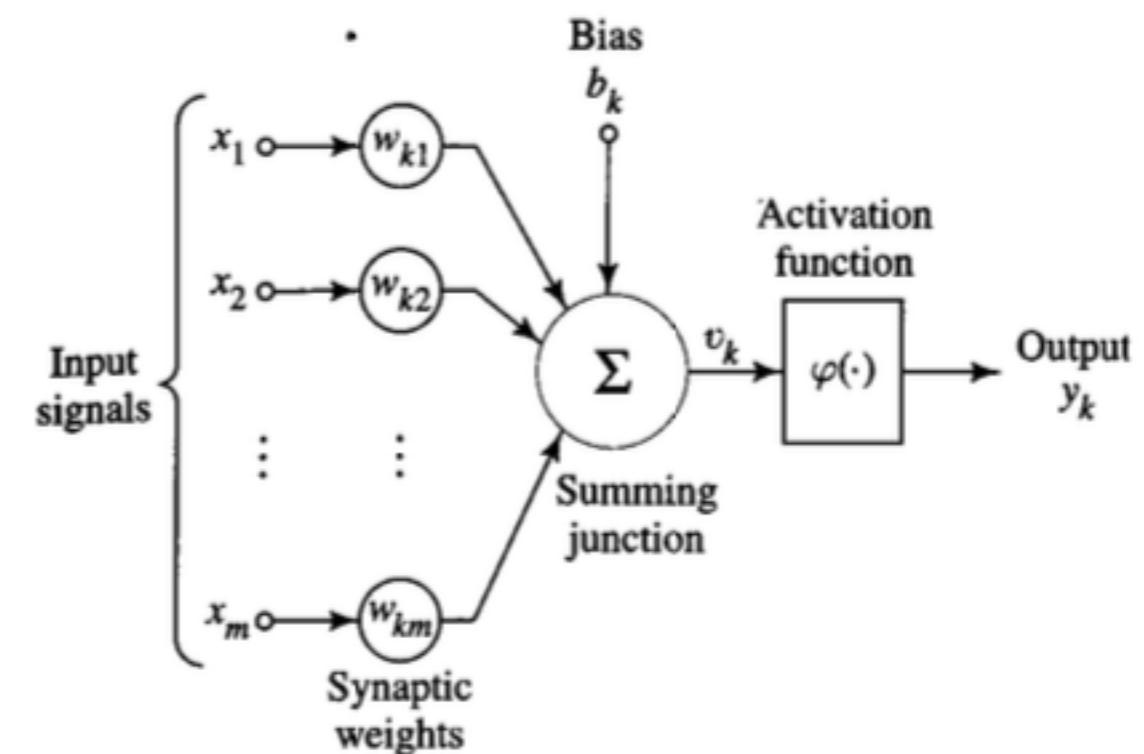


FIGURE 1.2 The pyramidal cell.



퍼셉트론 Perception

- 신경망의 수학적 모형

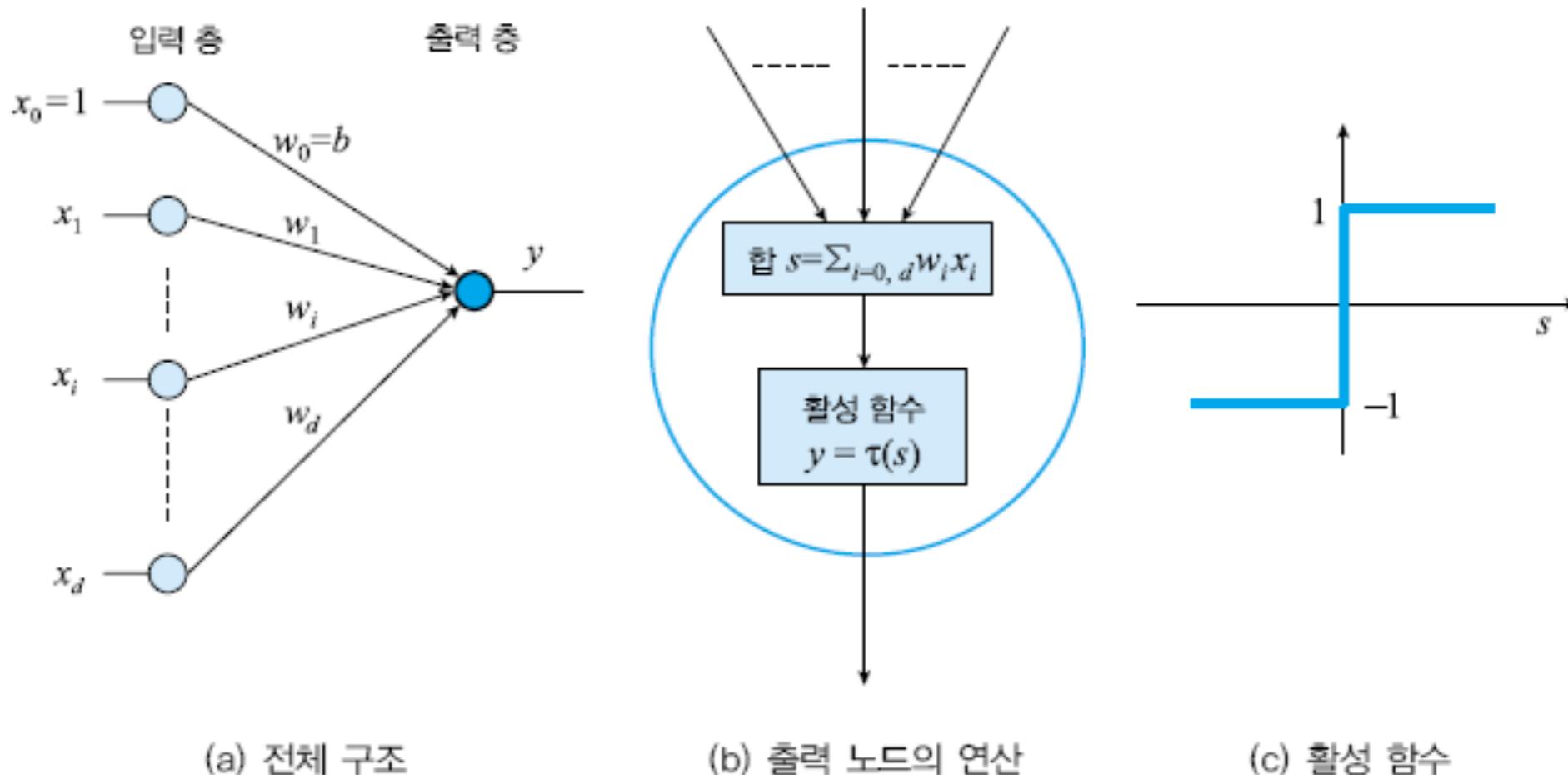


그림 4.2 퍼셉트론의 구조

출처: 패턴인식

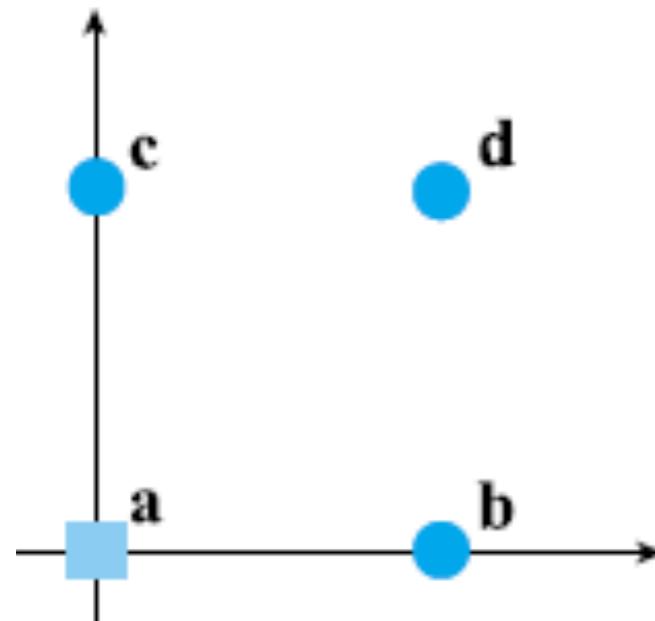
$$\mathbf{a} = (0,0)^T, t_a = -1$$

$$\mathbf{b} = (1,0)^T, t_b = 1$$

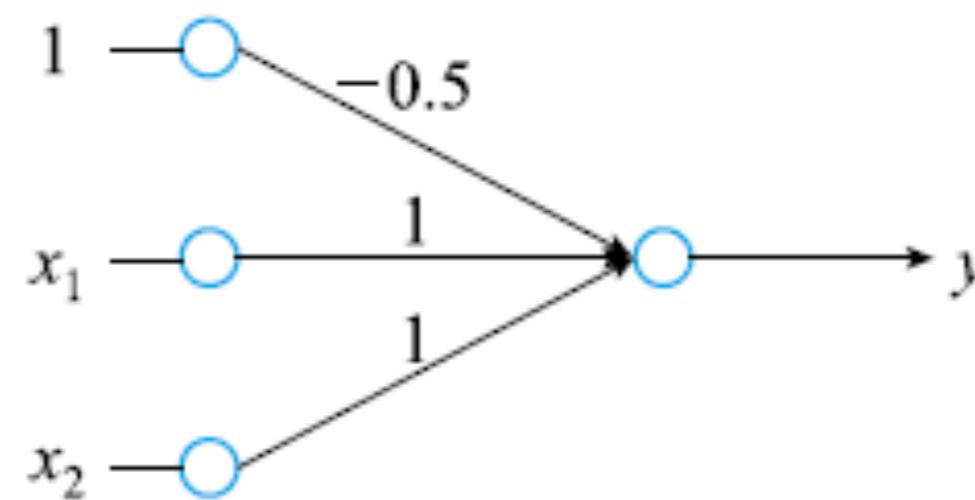
$$\mathbf{c} = (0,1)^T, t_c = 1$$

$$\mathbf{d} = (1,1)^T, t_d = 1$$

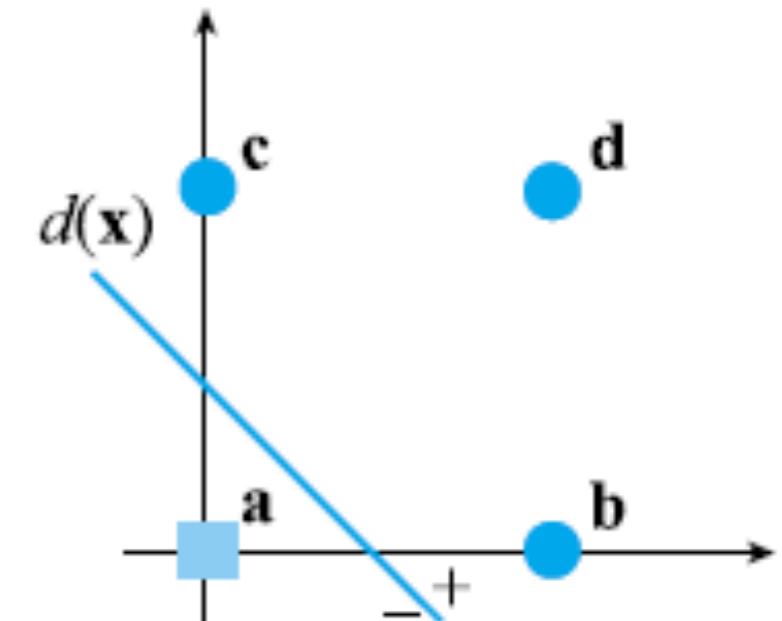
Simple Example: Logical OR



(a) OR 분류 문제



(b) OR 분류기로서 퍼셉트론

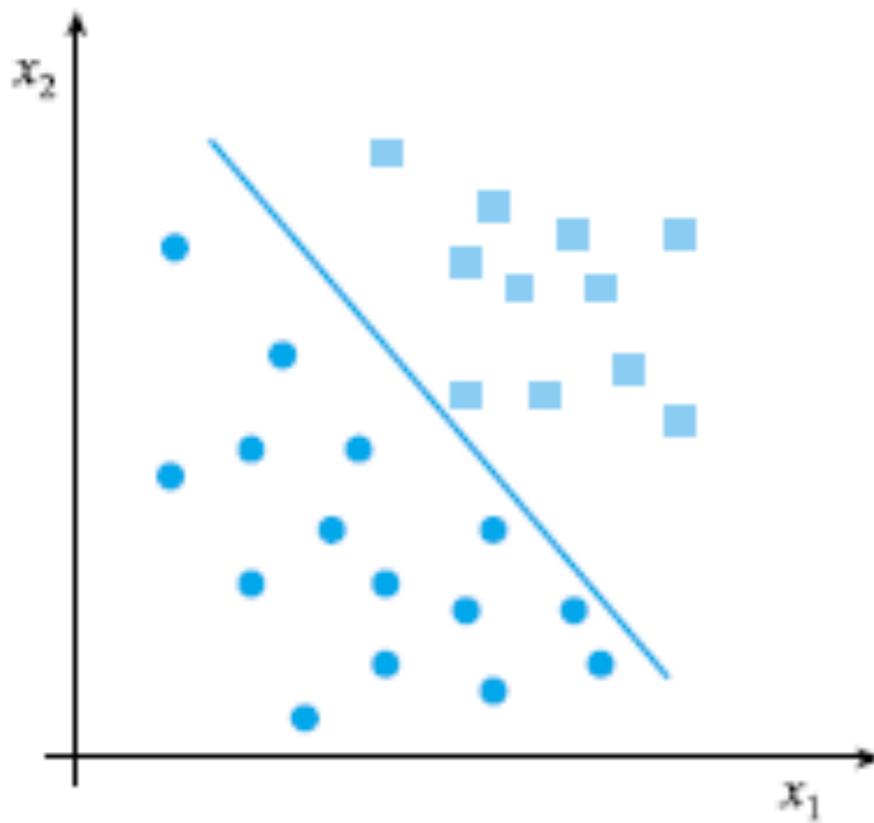


(c) 퍼셉트론은 선형 분류기

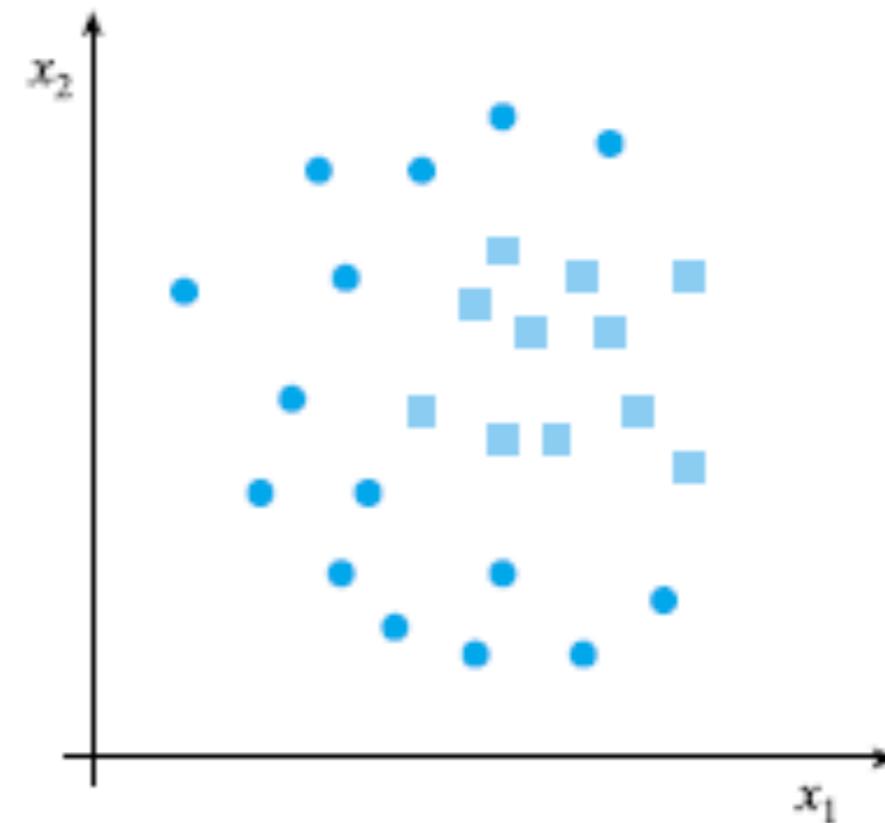
그림 4.3 퍼셉트론의 예

- a 는 -1 , b, c, d 는 $+1$ 로 분류할 w_1, w_2, w_3, b 를 찾는 문제
- 여기에서 $-1, +1$ 은 qualitative variable

선형 분리 불가능



(a) 선형 분리 가능



(b) 선형 분리 불가능

그림 4.5 선형 분리 가능과 불가능

출처: 패턴인식

현실 문제는 대체로 선형 분리 불가능: 단일 퍼셉트론으로 해결불가능

다층 퍼셉트론

MLP: Multi-Layer Perceptron

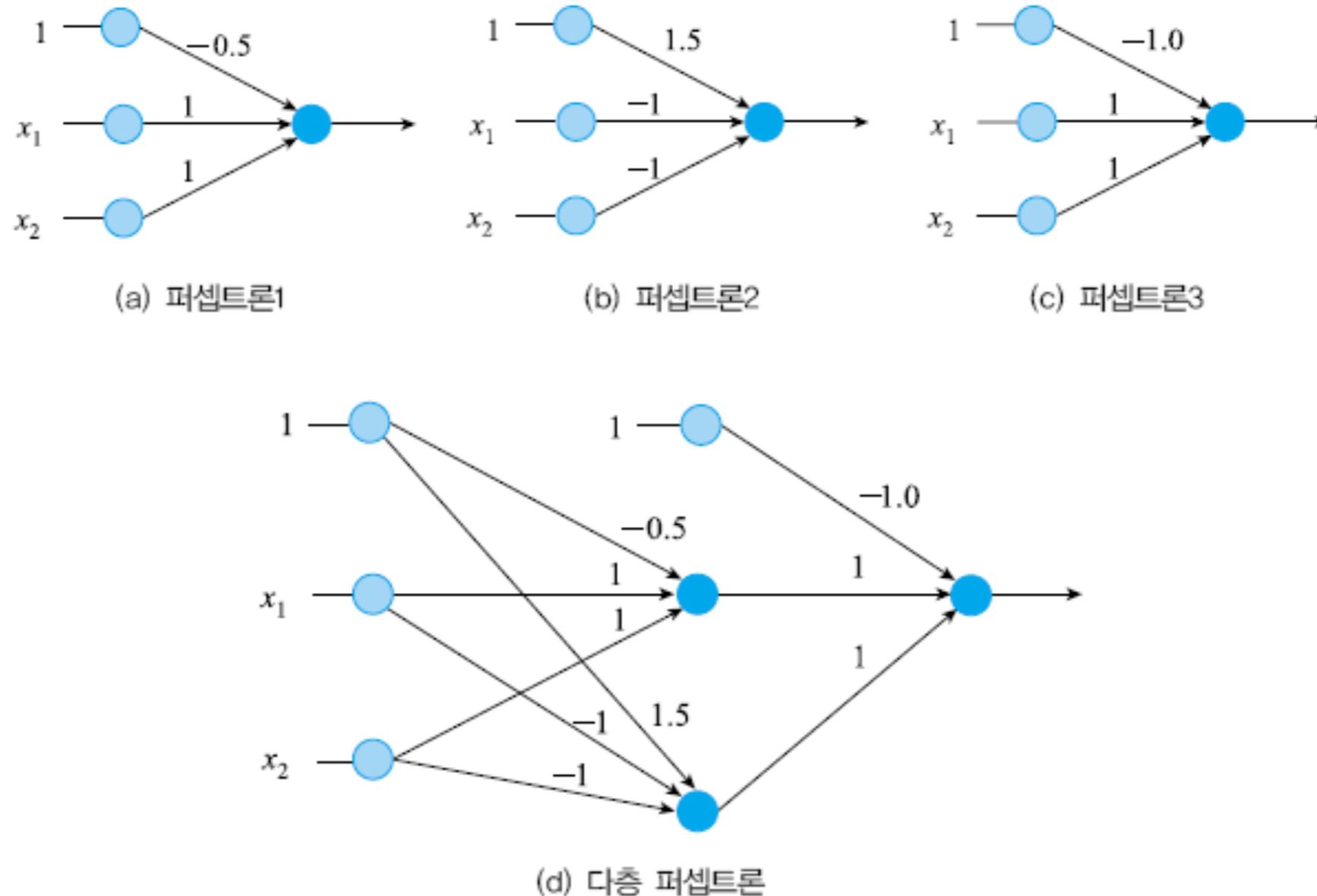
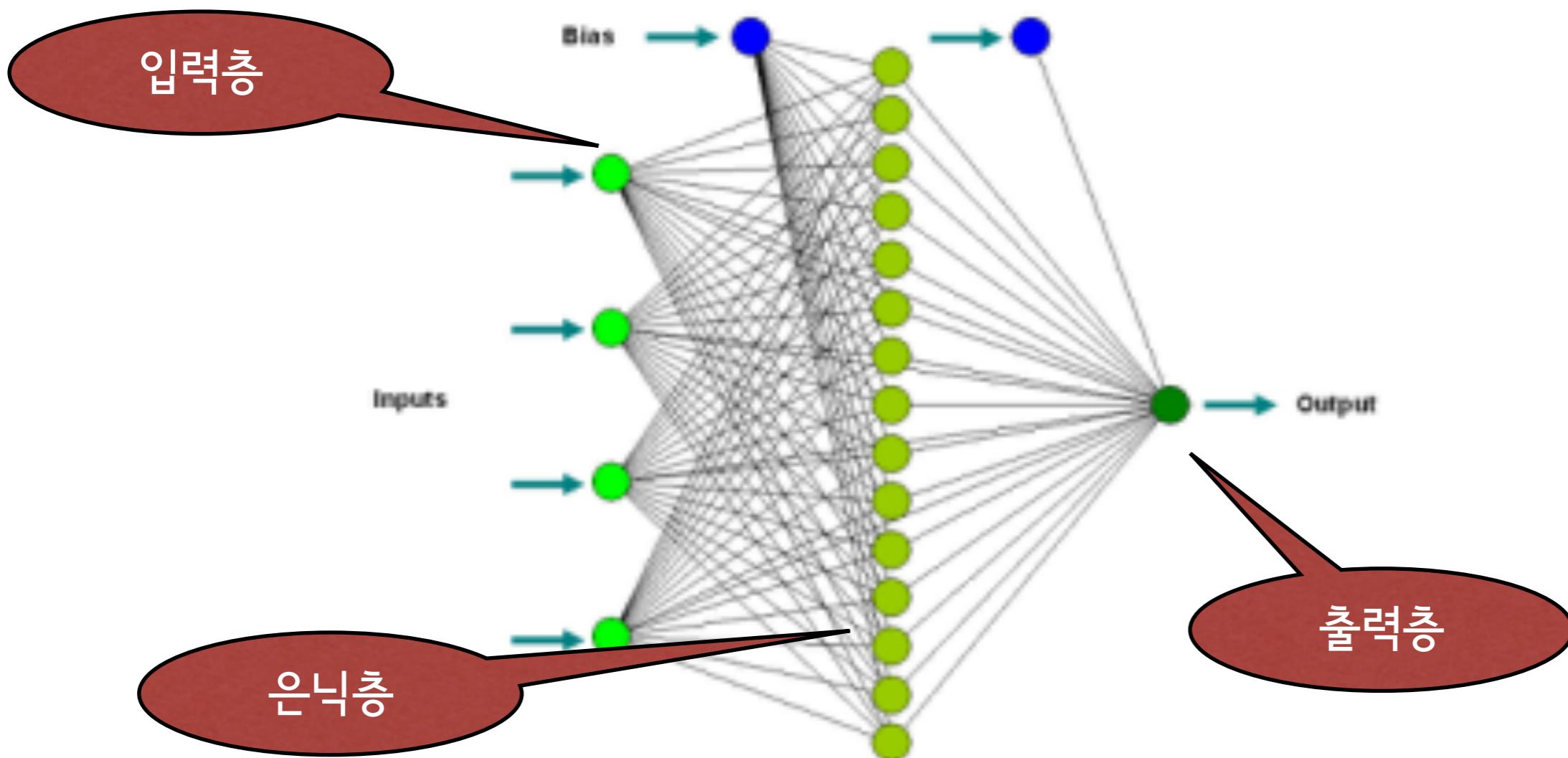


그림 4.8 세 개의 퍼셉트론과 이들을 연결하여 만든 다층 퍼셉트론

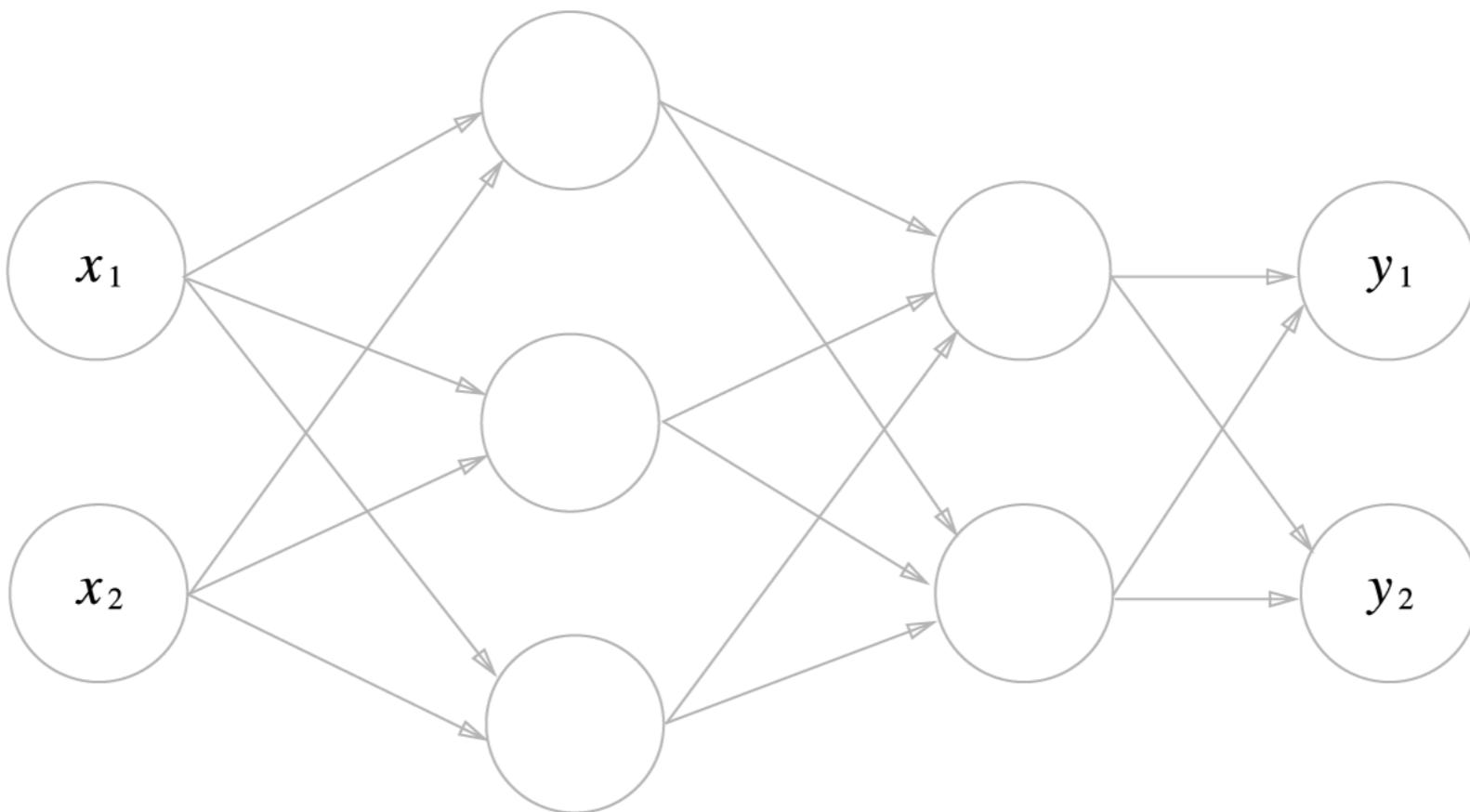
패턴 인식

신경망 Neural Networks



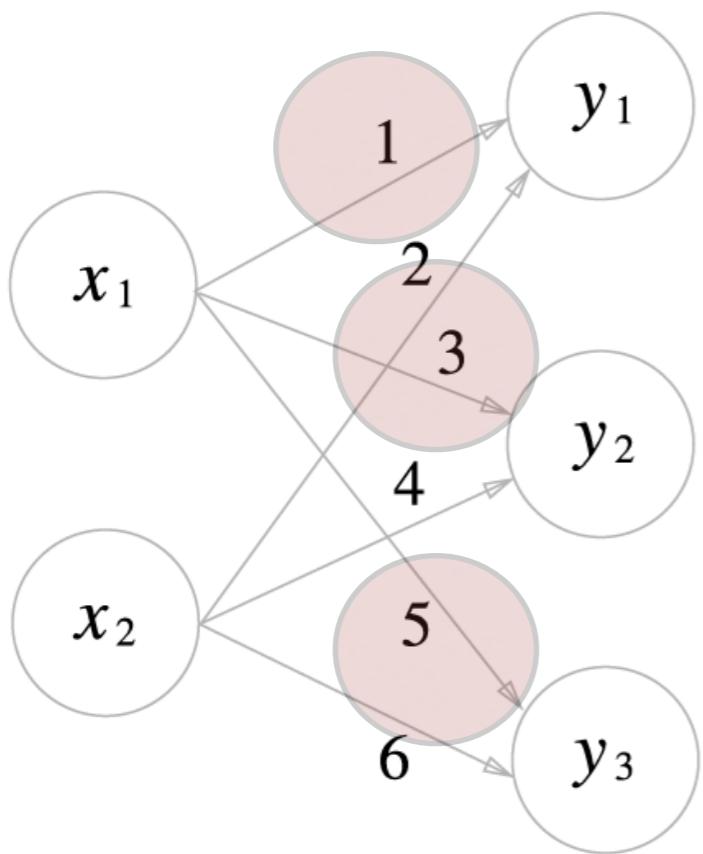
<http://neuromastersoftware.com/neural-network-theory-introduction/>

3층 신경망



사이토 고키 (2017)

신경망 \Rightarrow 선형대수

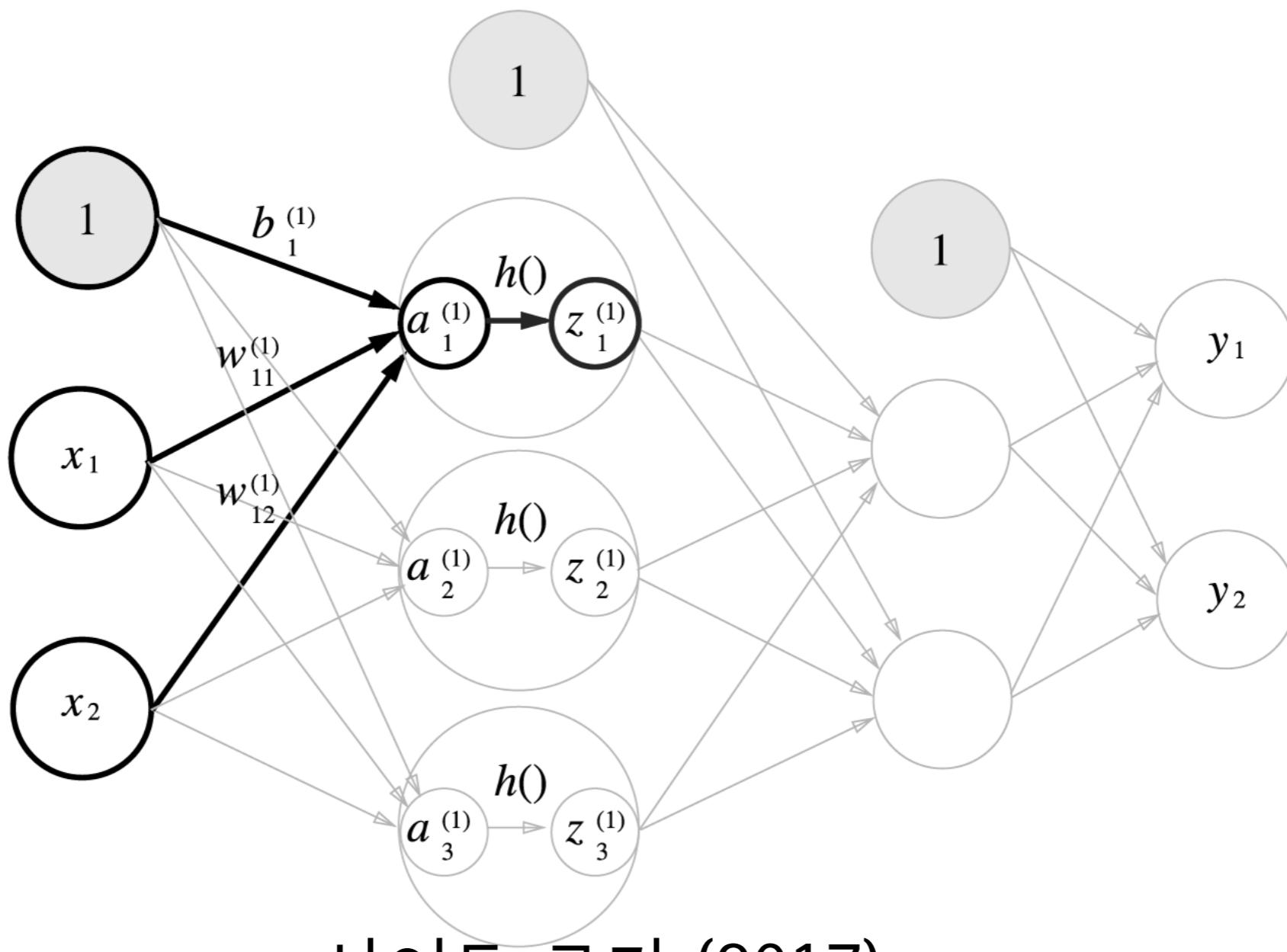


$$\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

$$\begin{array}{c} X \quad \quad W \quad = \quad Y \\ \boxed{2} \quad \quad \boxed{2 \times 3} \quad \quad \boxed{3} \\ \text{일치} \end{array}$$

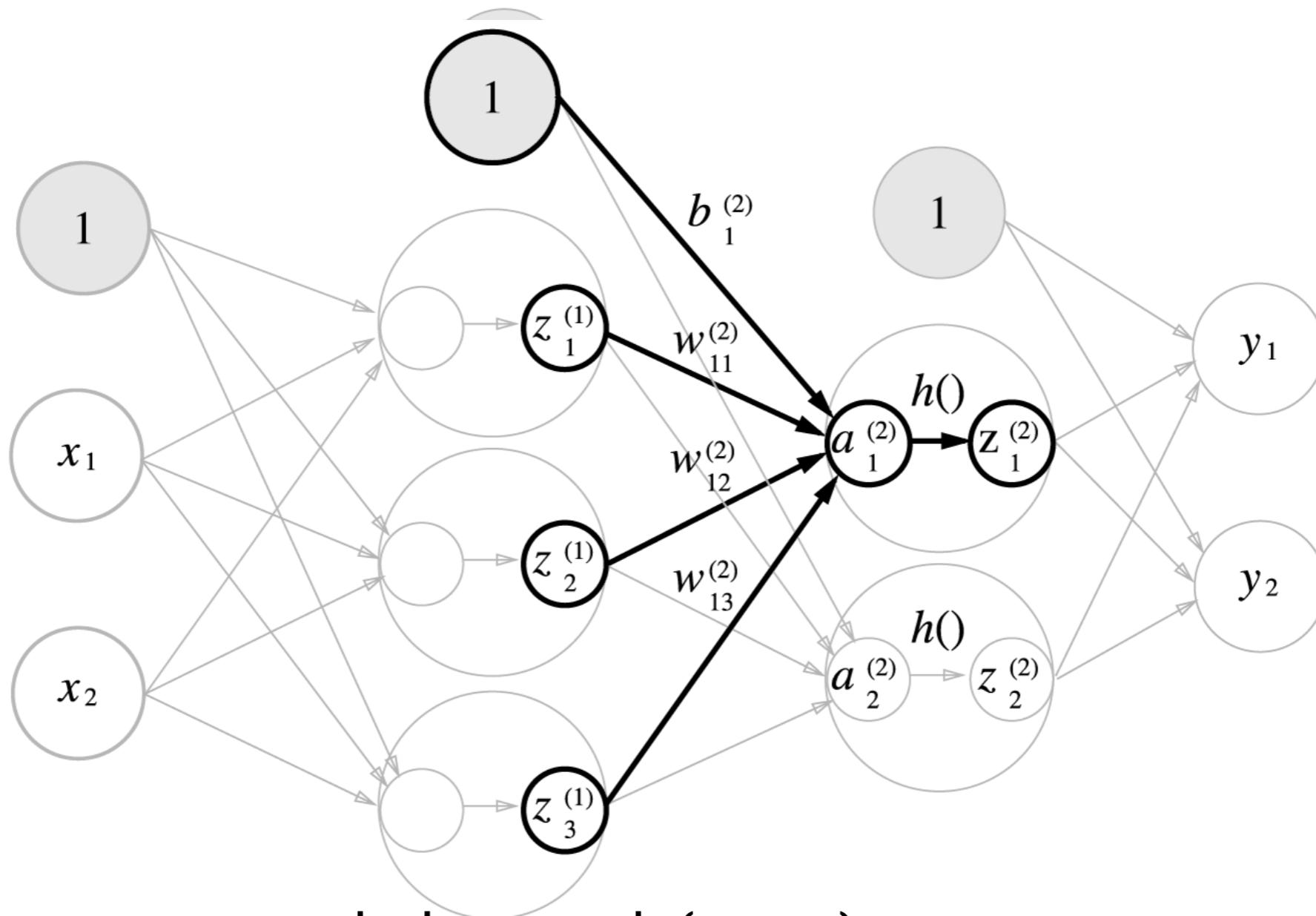
사이트 고키 (2017)

신호전달



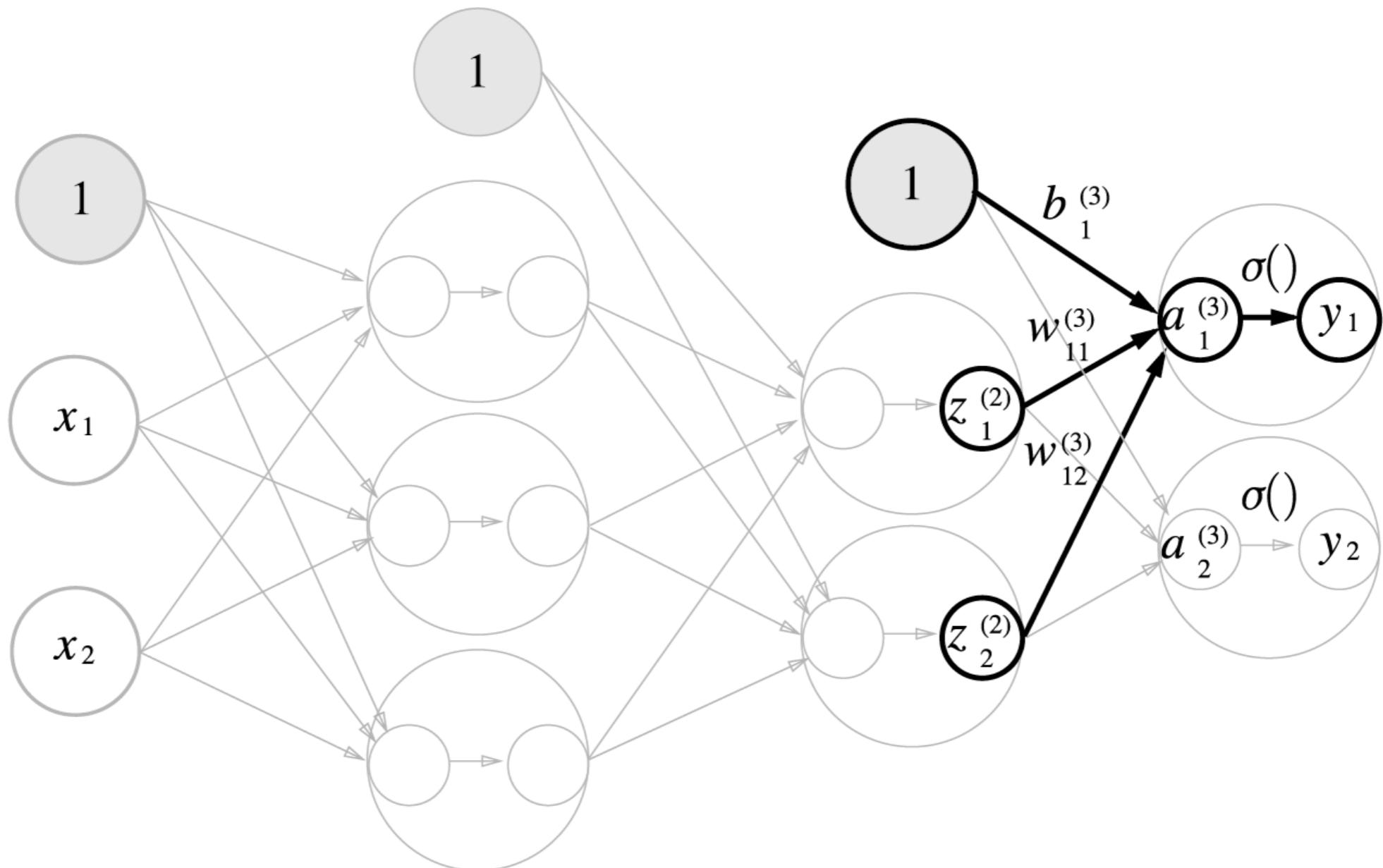
사이트 고키 (2017)

신호전달



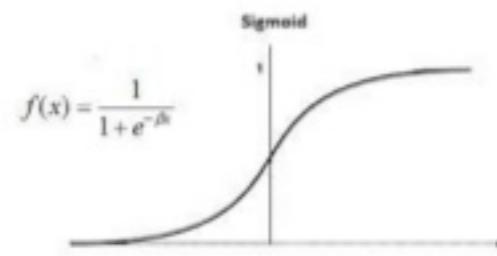
사이트 고키 (2017)

신호전달



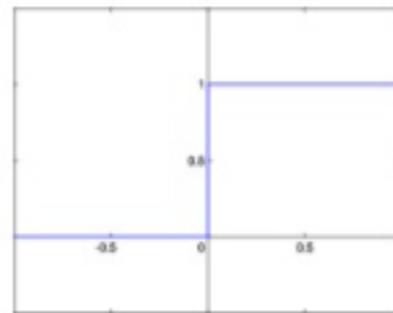
활성함수

Activation Functions



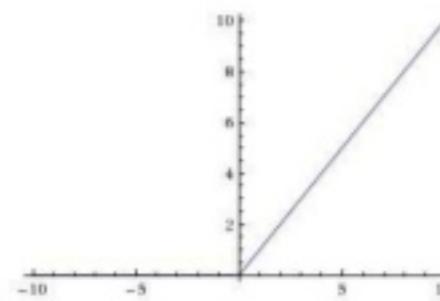
Sigmoid

http://www.saedsayad.com/artificial_neural_network.htm



Step

http://en.wikibooks.org/wiki/Artificial_Neural_Networks/Activation_Functions#Continuous_Log-Sigmoid_Function



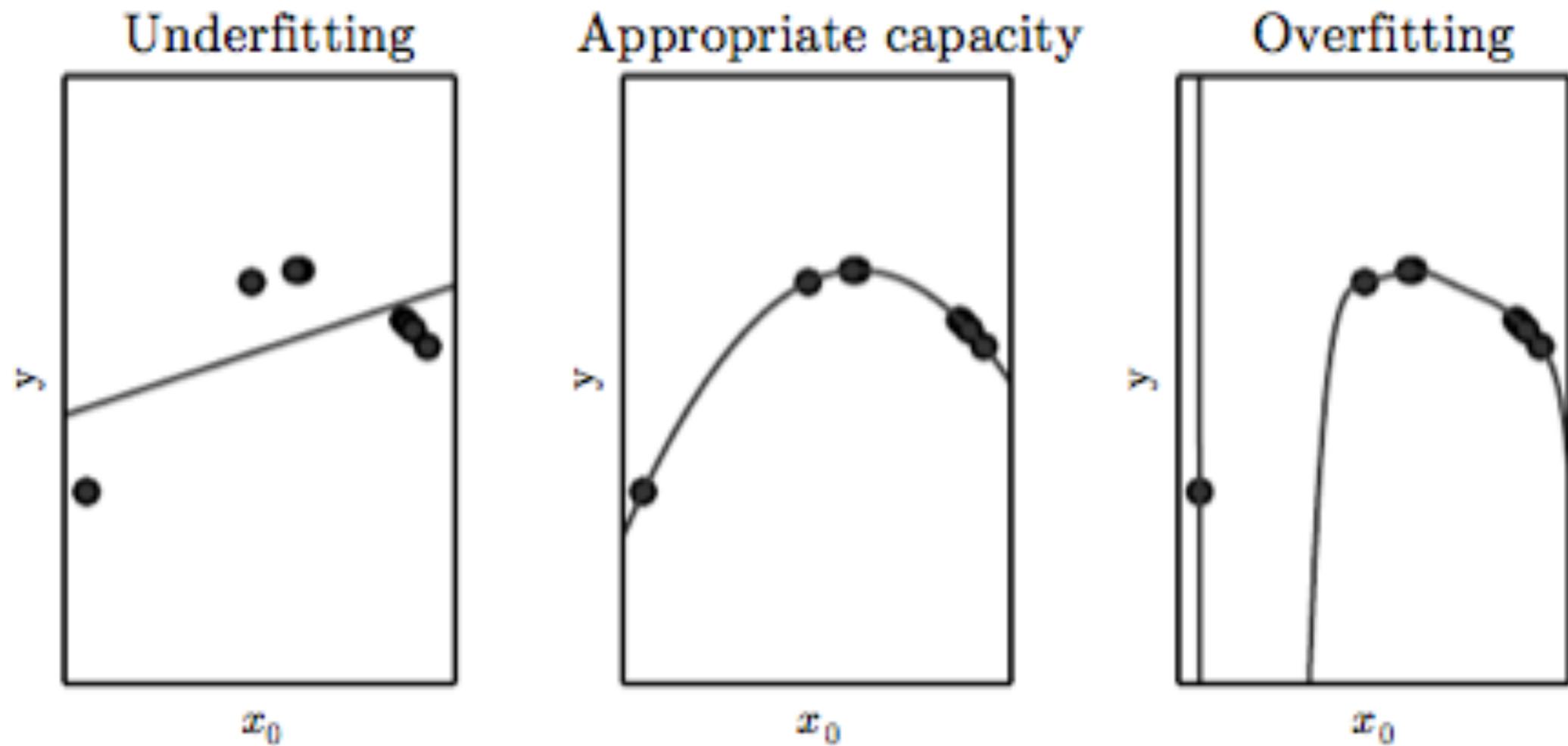
ReLU

<http://cs231n.github.io/neural-networks-1/>

And many others...

<https://image.slidesharecdn.com/usuconference-deeplearning-160418191119/95/introduction-to-deep-learning-7-638.jpg?cb=1461006739>

과적합 문제 Overfitting



Goodfellow, I., Bengio, Y., & Courville, A. (2016)

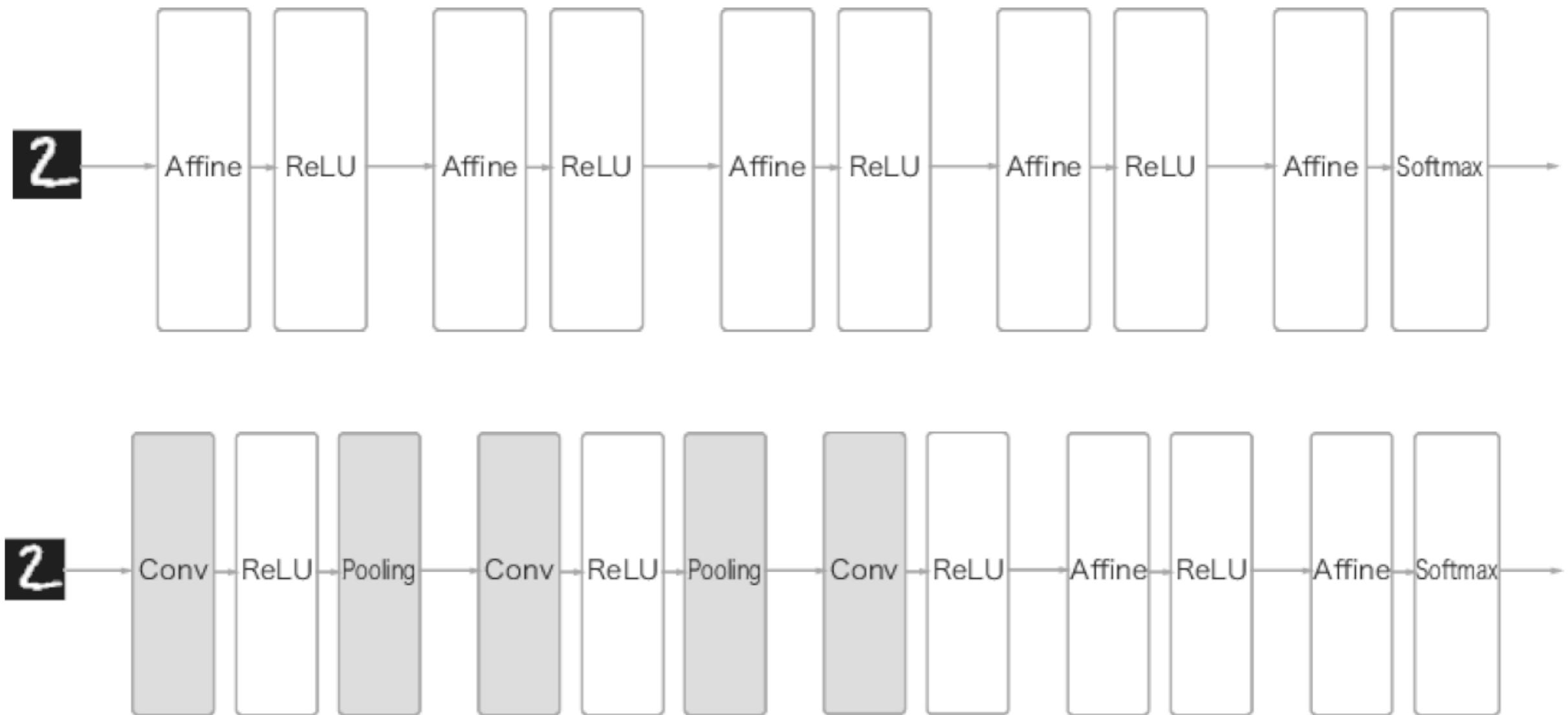
훈련데이터에만 지나치게 적응되어 그 외의 데이터에는
제대로 대응하지 못하는 상태

과적합에 대한 대응

- 가중치 감소 (overfit weight decay)
 - 학습 과정에서 큰 가중치에 큰 페널티를 부과
- 드롭아웃 (dropout)
 - 훈련때 무작위로 은닉층의 뉴런을 삭제
- 앙상블 학습 (ensemble learning)
 - 개별적으로 학습시킨 여러 모델의 결과를 평균내어 추론하는 방식

합성곱신경망

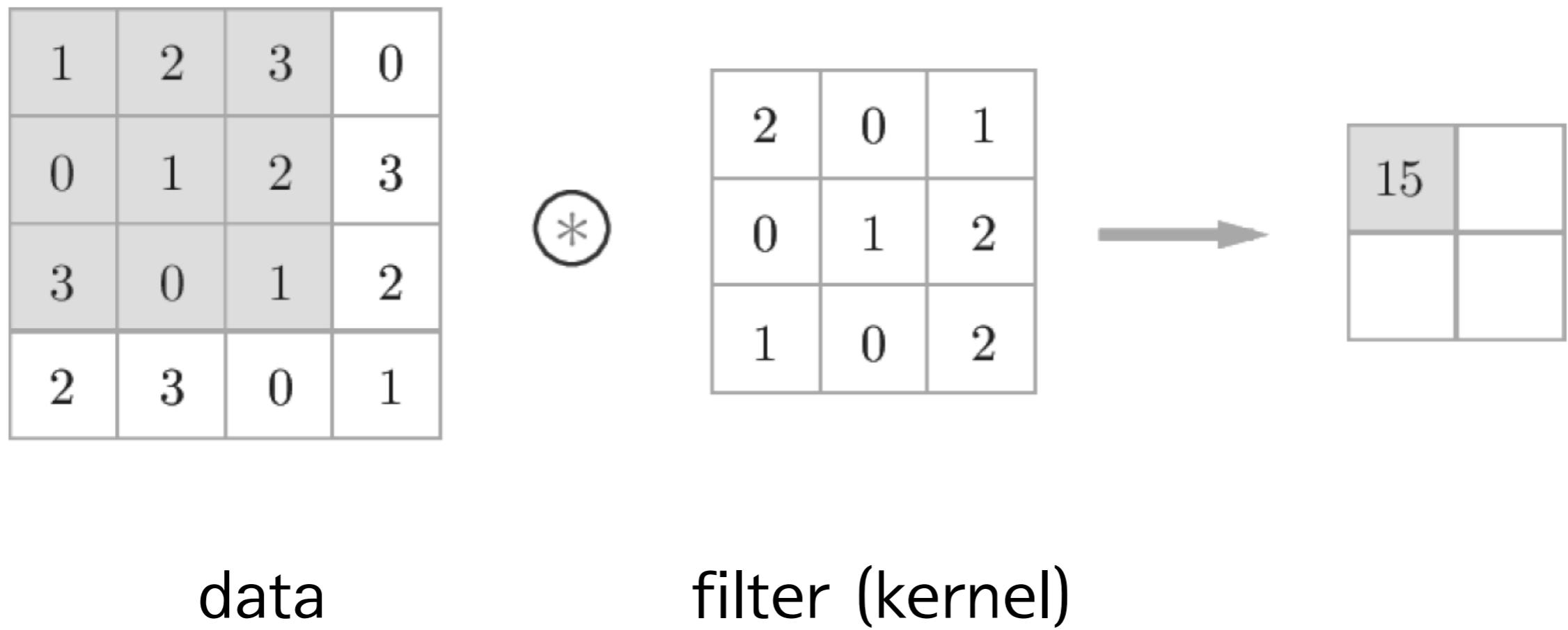
Convolutional Neural Network



CNN의 특징

- Affine layers는 데이터 형상 정보를 무시함
 - 28×28 pixel data를 784×1 pixel data로 취급
- CNN의 경우 형상을 유지하며, 데이터를 3차원 데이터로 전달함
 - 더 높은 인식능력을 보임

Convolution



사이트 고키 (2017)

Convolution

1	2	3	0
0	1	2	3
3	0	1	2
2	3	0	1

(*)

2	0	1
0	1	2
1	0	2



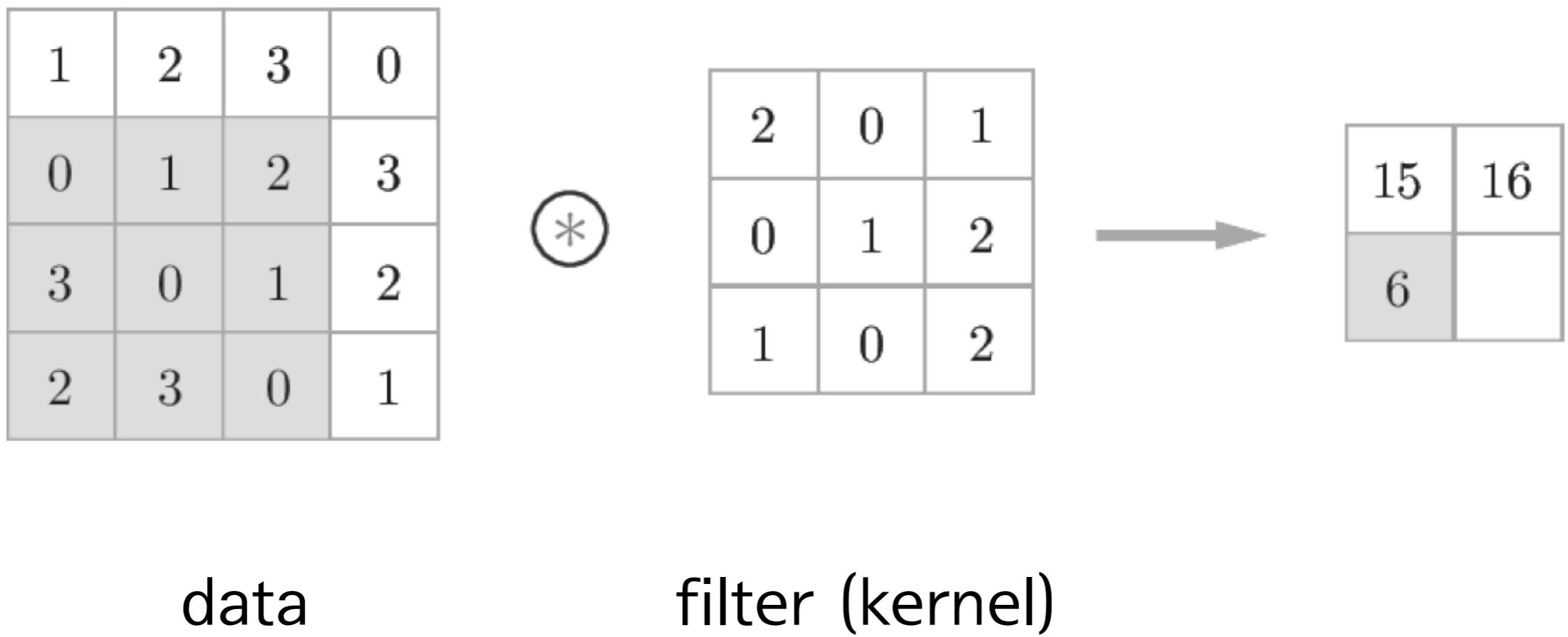
15	16

data

filter (kernel)

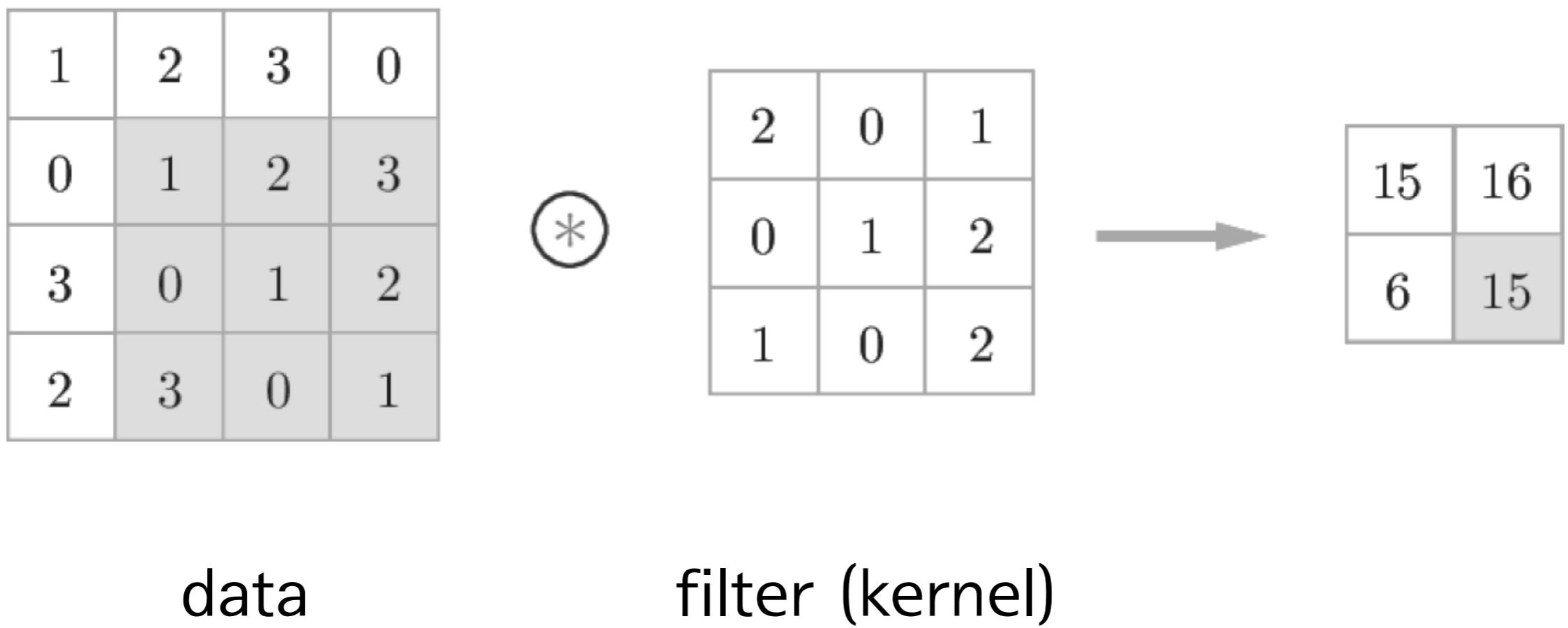
사이토 고키 (2017)

Convolution



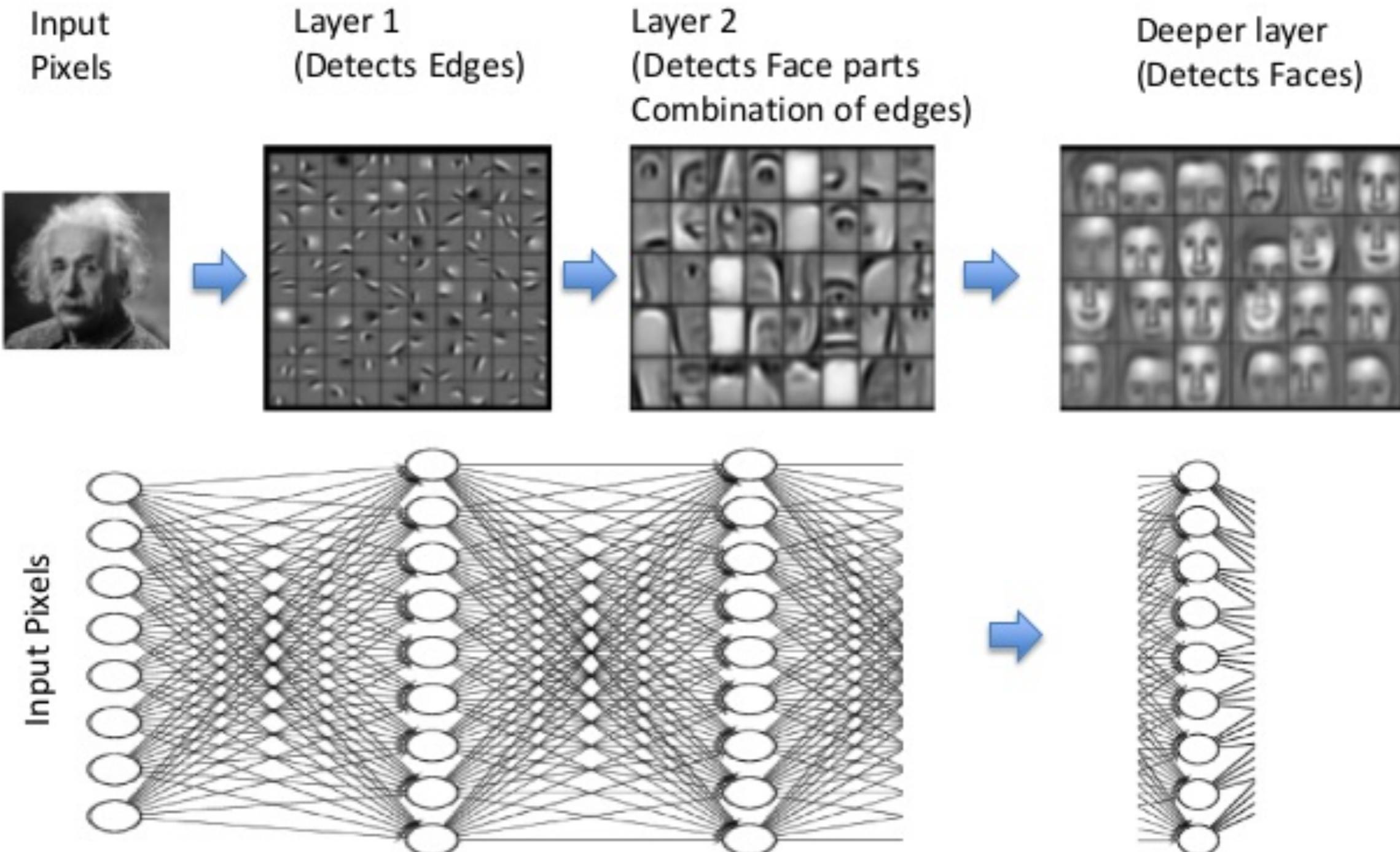
사이트 고키 (2017)

Convolution



사이트 고키 (2017)

필터 (혹은 커널)



Stride

1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1
2	3	0	1	2	3	0
1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1

⊗

2	0	1
0	1	2
1	0	2



15		

스트라이드 : 2

1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1
2	3	0	1	2	3	0
1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1

⊗

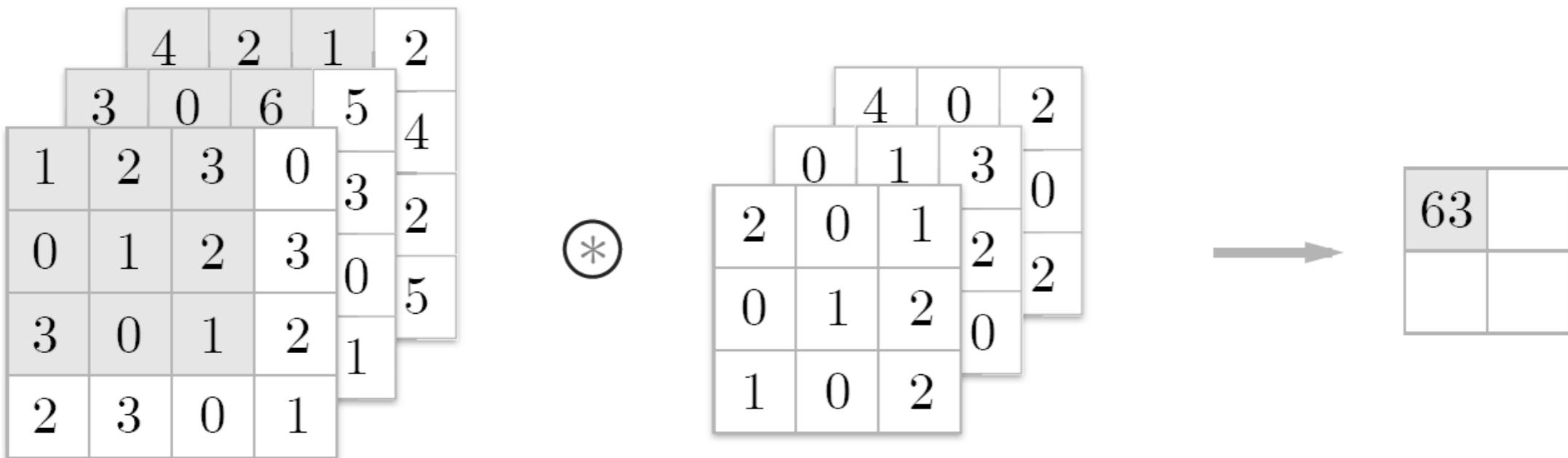
2	0	1
0	1	2
1	0	2



15	17	

사이토 고키 (2017)

n 차원으로의 확대

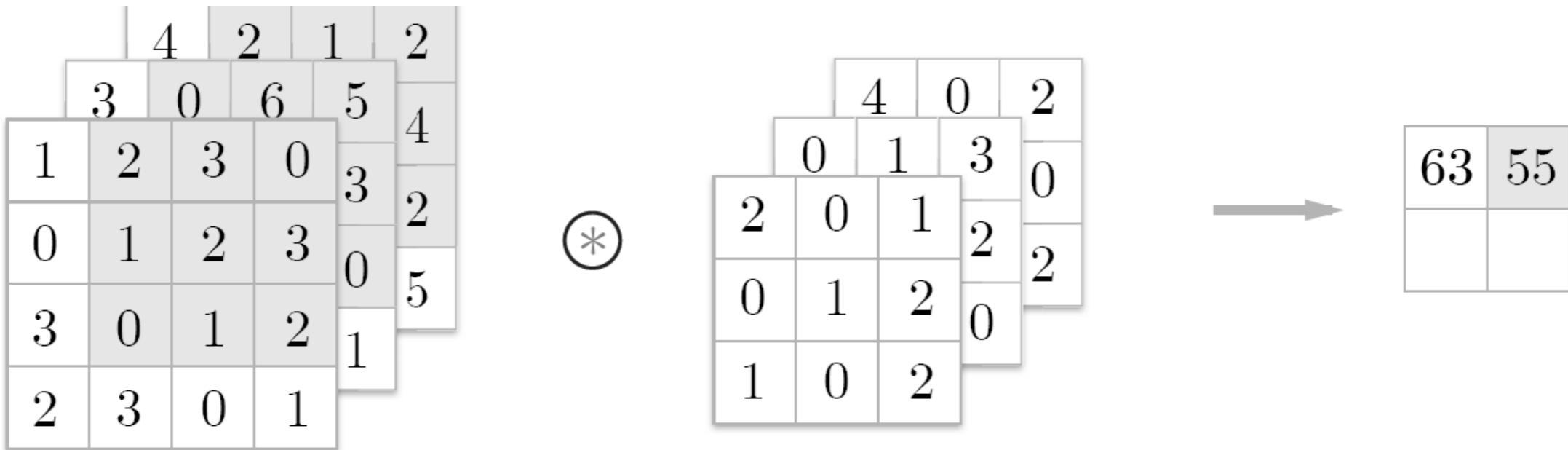


사이트 고키 (2017)

그래픽 이미지는 3채널: R, G, B
여기에 가로, 세로 위치정보까지 모두 3차원 정보

유한차원(n 차원)으로 확장 가능: tensor

n 차원으로의 확대

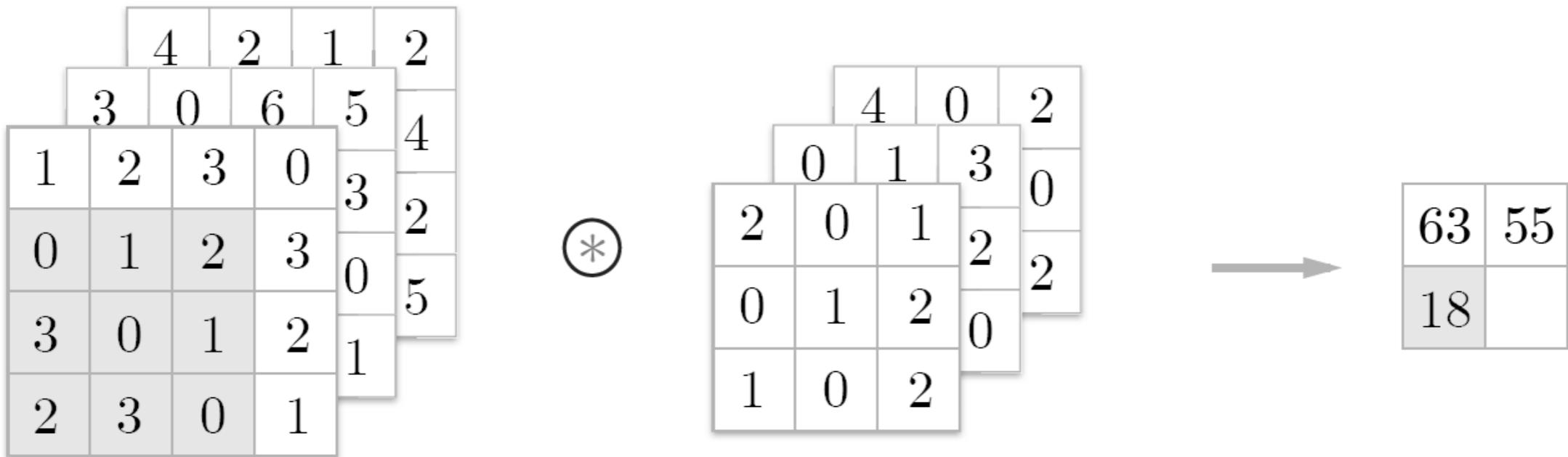


사이트 고키 (2017)

그래픽 이미지는 3채널: R, G, B
여기에 가로, 세로 위치정보까지 모두 3차원 정보

유한차원(n 차원)으로 확장 가능: tensor

n 차원으로의 확대

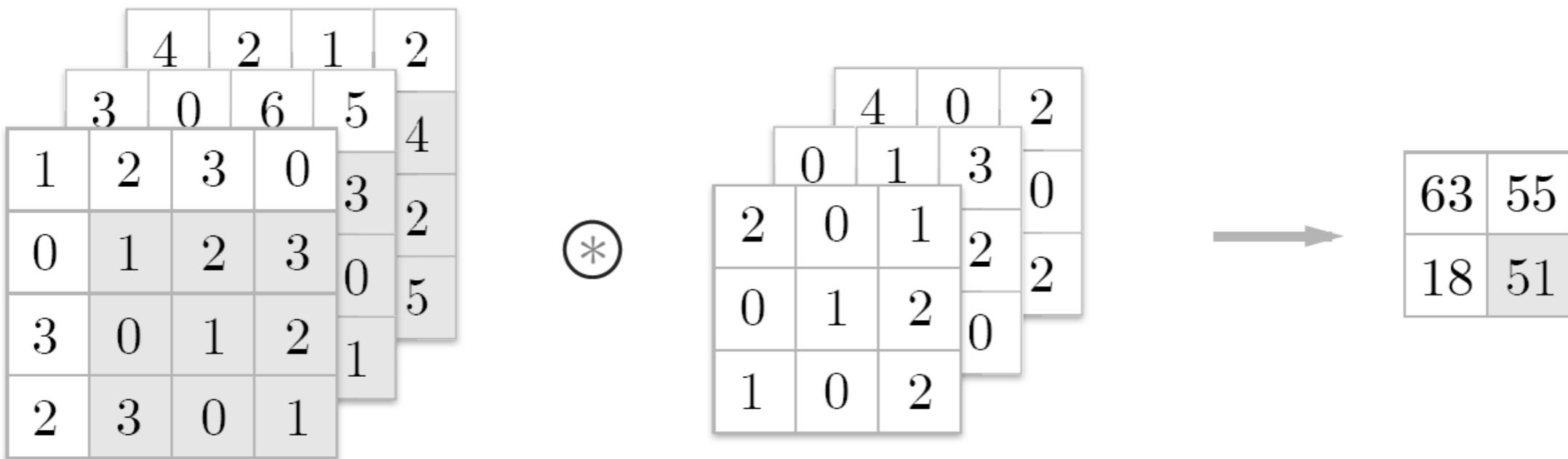


사이트 고키 (2017)

그래픽 이미지는 3채널: R, G, B
여기에 가로, 세로 위치정보까지 모두 3차원 정보

유한차원(n 차원)으로 확장 가능: tensor

n차원으로의 확대

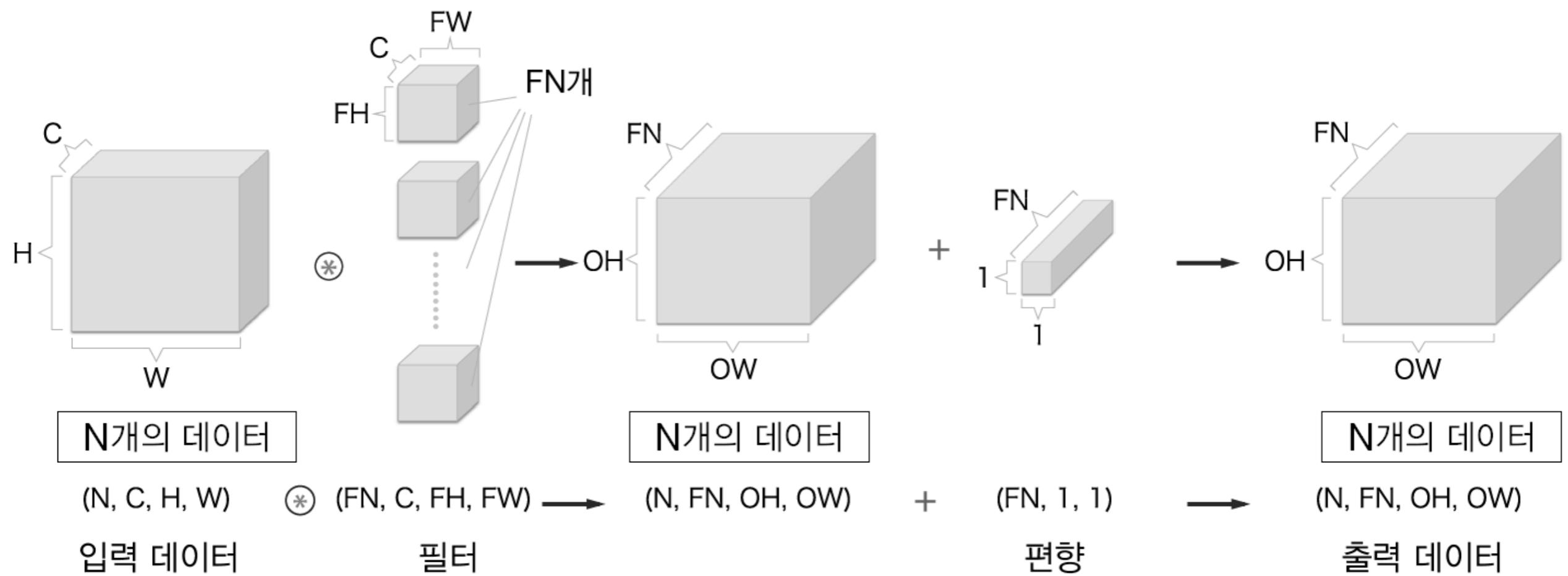


사이트 고키 (2017)

그래픽 이미지는 3채널: R, G, B
여기에 가로, 세로 위치정보까지 모두 3차원 정보

유한차원(n차원)으로 확장 가능: tensor

Convolution Process



사이토 고키 (2017)

Pooling Layer (or Downsampling Layer)

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3
4	

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3
4	2

Max Pooling with Stride 2

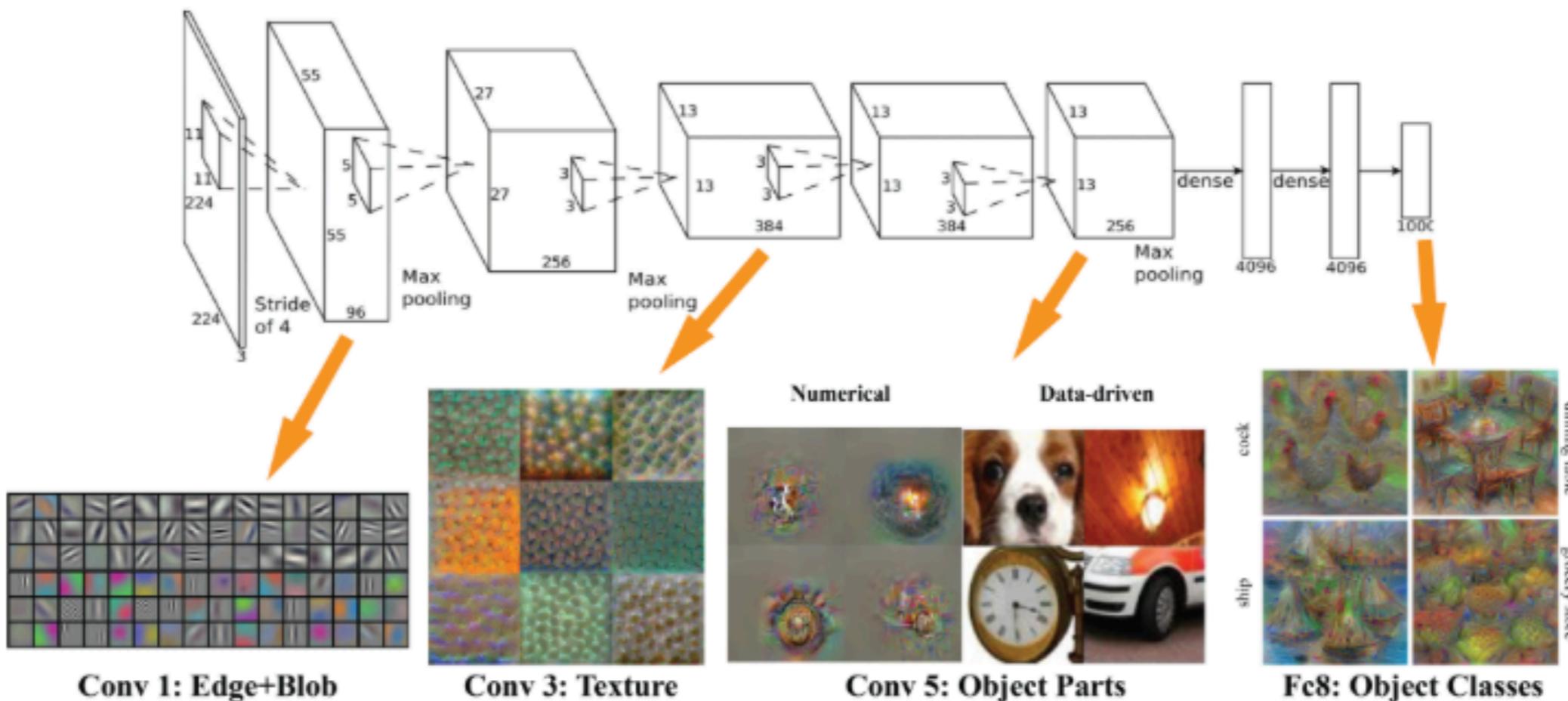
사소한 입력 변화를 흡수 (노이즈에 robust하게 만듦)

사이토 고키 (2017)

CNN 중첩에 따른 정보변화

Goal: Find an image that optimize the activation of a single neuron [Erhan et al., Simonyan et al., Zhou et al.]

a. *Different Layers:* ([AlexNet](#)) We visualize Conv1,3,5 neurons learned from ImageNet dataset. With the increasing layer depth, neurons are learned to recognize from simple edge/blob and texture pattern to complex object parts and class. (For Conv 5, we retrieve real images for comparison with Zhou et al.)

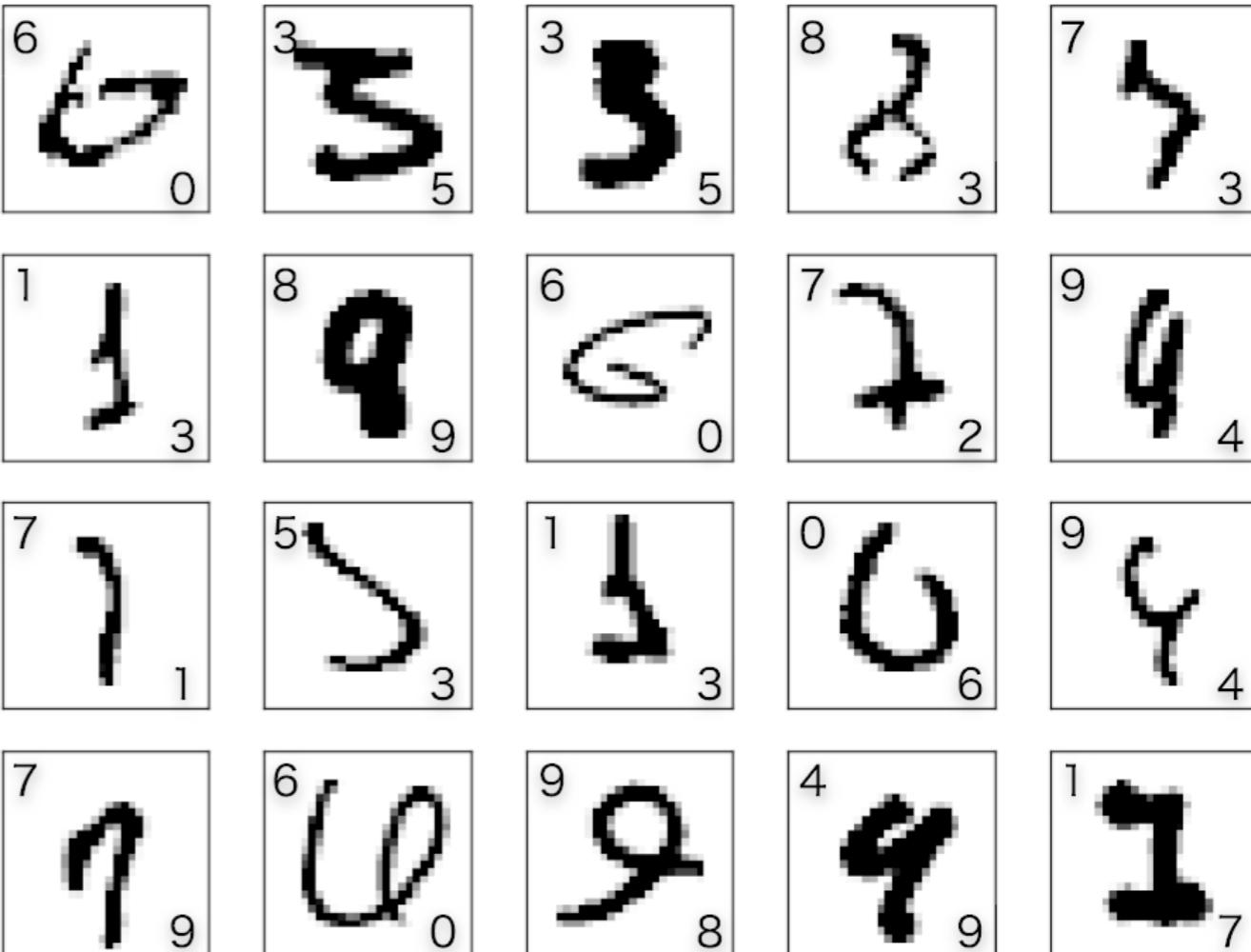


계층이 깊어질수록 더 복잡하고 추상화된 정보가 추출됨

DL과 관련한 추가적 이야기

MNIST DL 인식에서 인식 실패한 이미지들

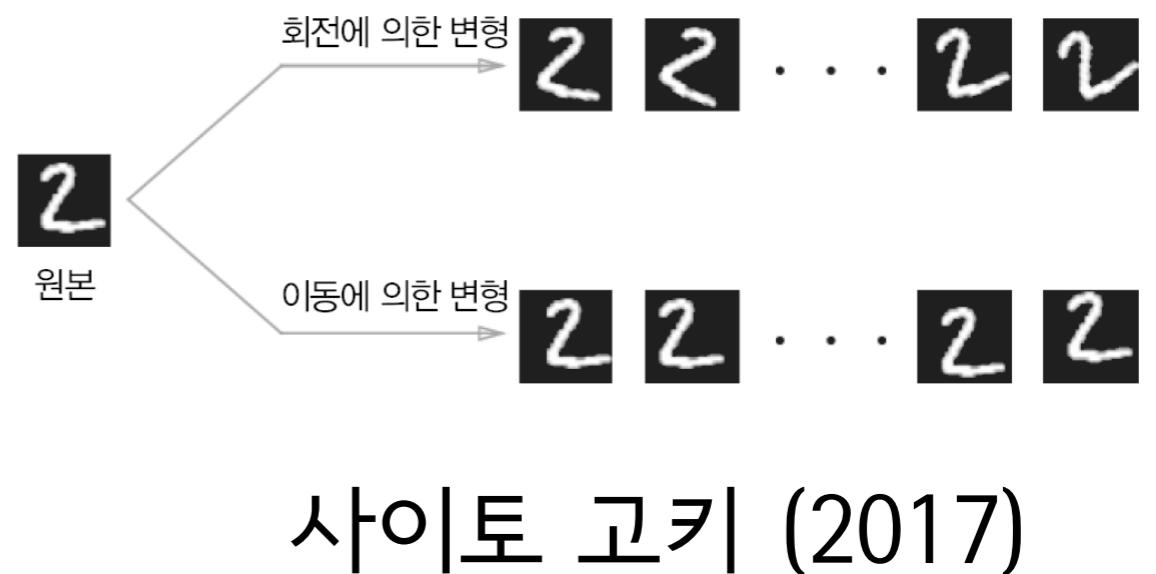
- 인간이 이해하기에도 어려운 손글씨였음
- 인간과 유사한 패턴의 인식 오류
- 인간의 인식을 기계로 구현 한 것이라는 측면에서 보았을 때에는 Good News일 수 있음



사이트 고키 (2017)

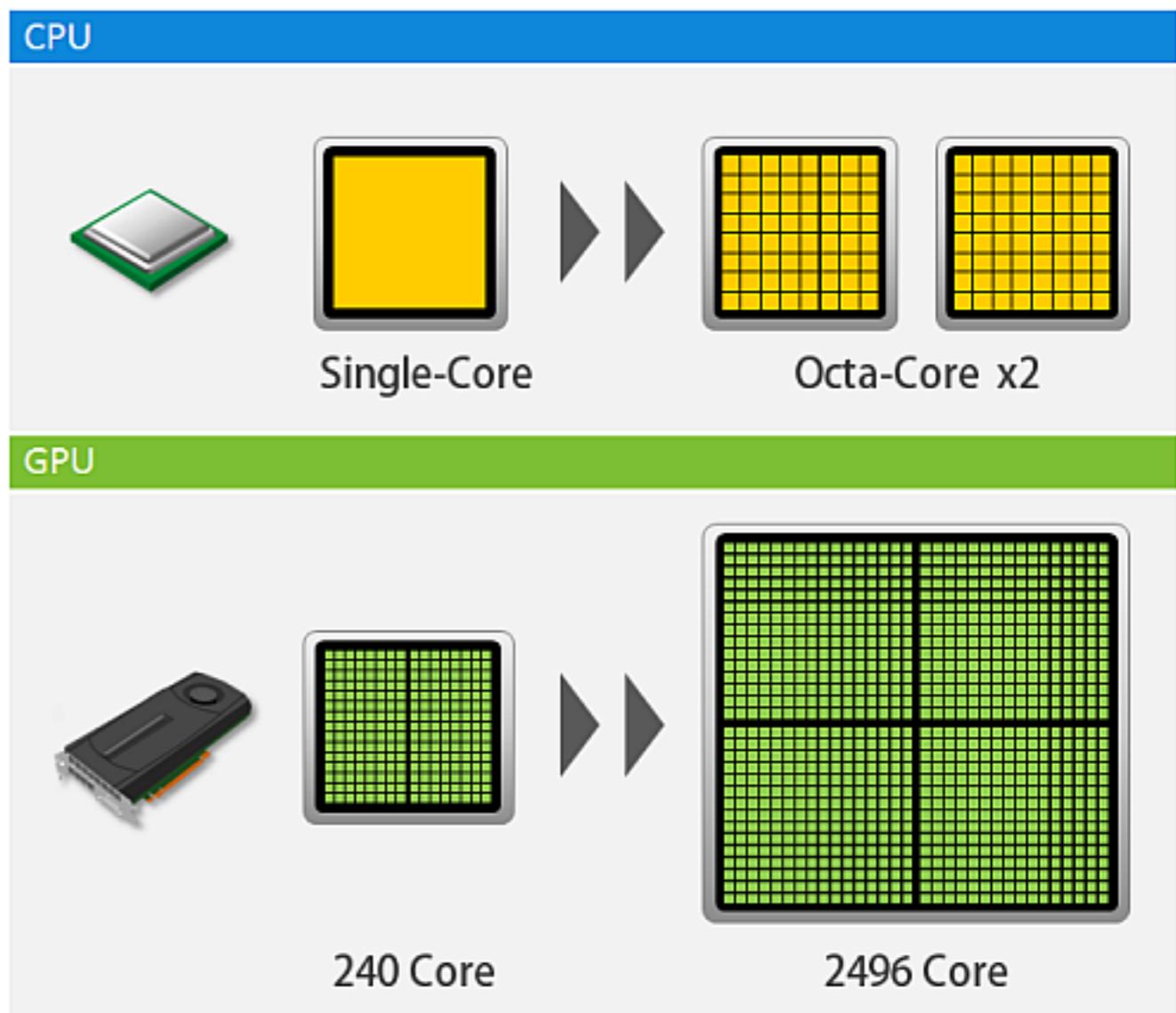
Data Augmentation

- 차원의 저주 (Dimension Curse)
 - 태스크가 복잡해질수록
⇒ 데이터 차원 증가 ⇒
필요한 데이터수 폭증
- 훈련 이미지를 변형하여 훈련 데이터를 추가
 - 회전, 이동, 잘라내기
crop, 뒤집기|flip, 노이즈
추가 등
 - 학습 퍼포먼스 증가함



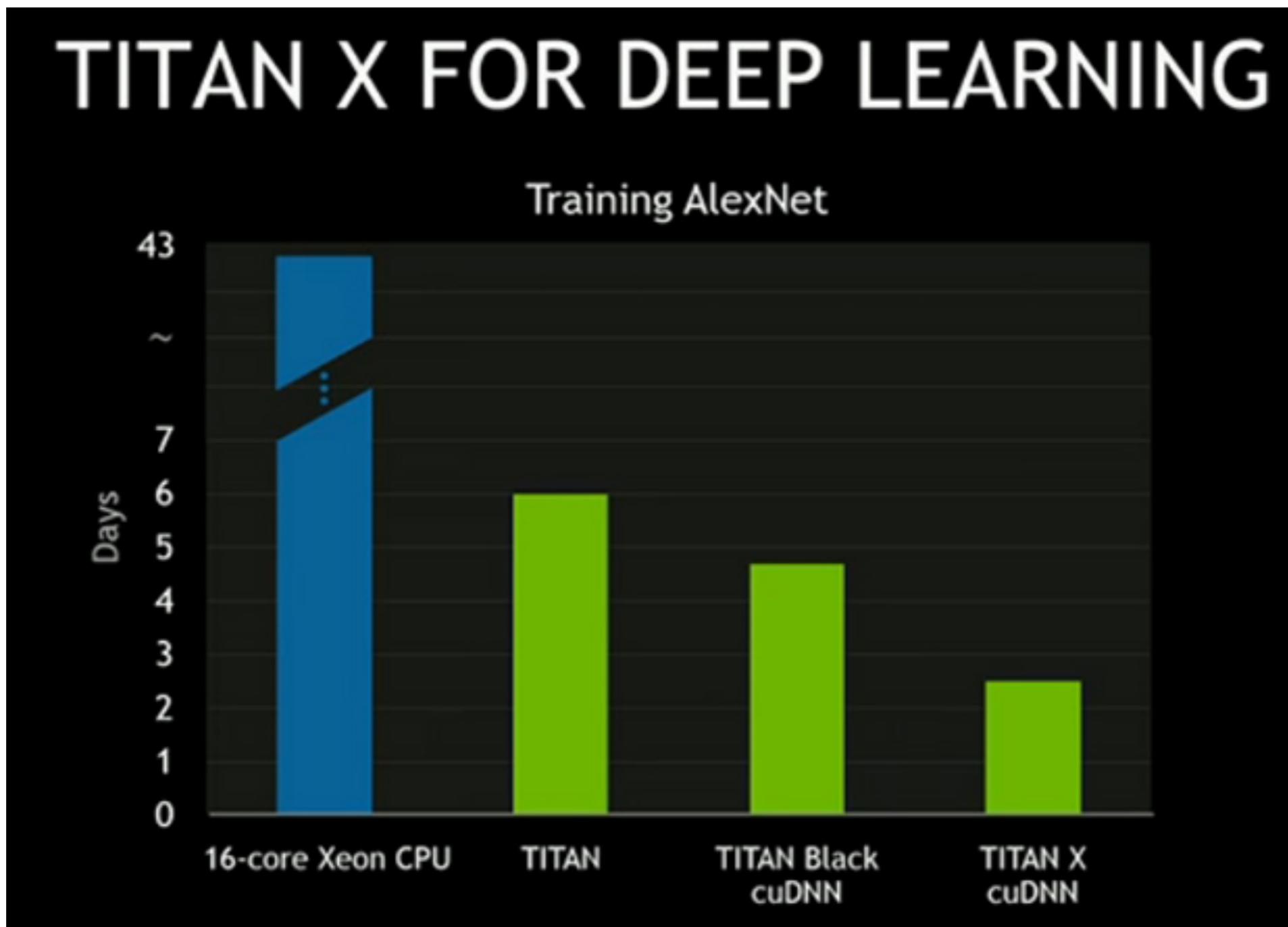
GPGPU

- General Purpose Computing on Graphics Processing Units
- 딥러닝을 위해서는 대규모의 연산이 필요
- GPU 사용시 동일 학습에 필요한 시간을 대폭 단축할 수 있음.



http://www.hpc.co.jp/gpu_solution.html

AlexNet Training Time: CPU versus GPU



DL 실습

TensorFlow

- Open Source Library for deep learning
 - <https://github.com/tensorflow/tensorflow>
 - <https://www.tensorflow.org>
 - \$ sudo easy_install —upgrade pip
 - \$ pip3 install tensorflow

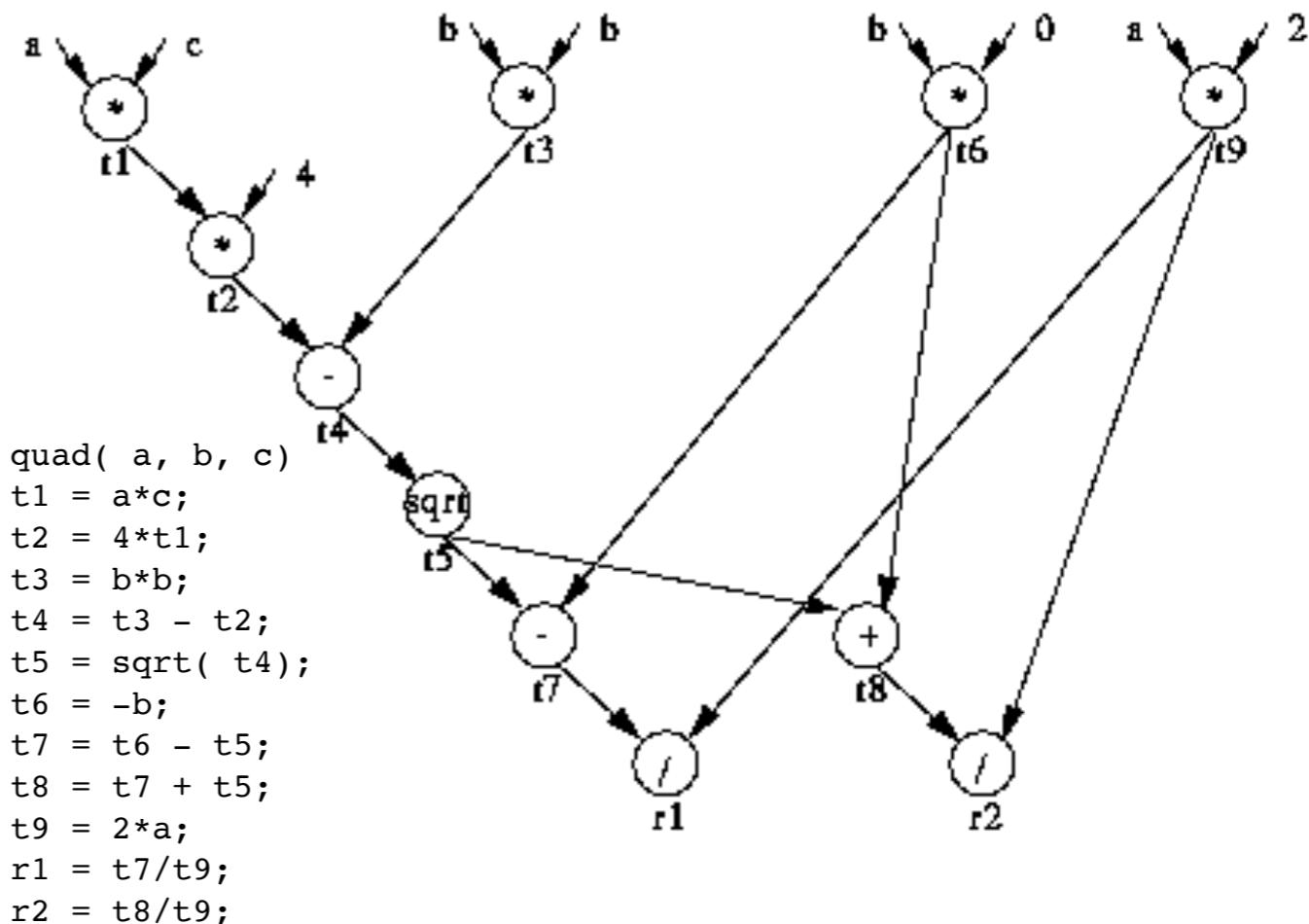
Simple Calculation

- $a+b$ 는 덧셈의 결과가 아니라 데이터 플로우 그래프라는 객체
- 텐서플로우 세션 (session)으로 데이터 플로우 그래프 객체를 실행하는 것

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> sess.run(hello)
'Hello, TensorFlow!'
>>> a = tf.constant(10)
>>> b = tf.constant(32)
>>> sess.run(a+b)
42
>>>
```

Data-Flow Graphs (DFG)

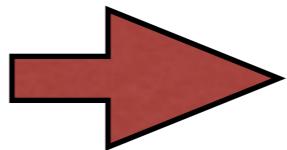
- 연산들 사이에서의 자료의 존성을 그래프로 표현한 것
 - 예) 근의 공식
 - t_2 는 t_1 에 의존적: t_2 는 t_1 이 계산된 후에만 계산 가능
 - t_3 은 t_1, t_2 와 무관하게 계산 가능



<http://bears.ece.ucsb.edu/research-info/DP/dfg.html>

Variable

- All seems to be DFG
- 텐서플로에서는 변수를 학습해야 할 파라미터들로 사용



```
import tensorflow as tf

# define constants
a = tf.constant(120, name="a")
b = tf.constant(140, name="b")
c = tf.constant(199, name="c")

# define variable
x = tf.Variable(0, name="x")

# define data flow graph

calc_op = a + b + c
assign_op = tf.assign(x, calc_op)

# run session

sess = tf.Session()
sess.run(assign_op)

# output

print(sess.run(x))
```

Placeholder

- Similar to Array?

```
import tensorflow as tf

# define placeholder
a = tf.placeholder(tf.int32, [None])

# define some operation
b = tf.constant(2)
x2_op = a * b

# session start
sess = tf.Session()

# run & output
r1 = sess.run(x2_op, feed_dict = {a:[1,2,3]})  
print(r1)

r2 = sess.run(x2_op, feed_dict = {a:[10,100,130,200,4002312]})  
print(r2)
```

```
[2 4 6]
[          20        200        260        400  8004624]
```

DL example

- height, weight, 그리고 비만도 (마름, 보통, 비만) 데이터를 학습시킨뒤, 5000개의 테스트셋으로 정확도를 판별
- data format: csv
- Procedure:
 - data process
 - making DFGs
 - define model learning
 - session run

```
import pandas as pd # to read csv
import numpy as np
import tensorflow as tf

# data read

csv = pd.read_csv("../_mainText_srcs/ch5/bmi.csv")

# data normalizing

csv[ "height" ] /= 200
csv[ "weight" ] /= 100

# label to array

bmi_class = { "thin": [ 1,0,0 ], "normal": [ 0,1,0 ], "fat": [ 0,
csv[ "label_pat" ] = csv[ "label" ].apply(lambda x: np.array(bm

# test set

test_csv = csv[15000:20000]
test_pat = test_csv[ [ "weight", "height" ] ]
test_ans = list(test_csv[ "label_pat" ])

# making DFG

#1. declaring placeholder

x = tf.placeholder(tf.float32, [None,2]) # height, weight
y = tf.placeholder(tf.float32, [None,3]) # pattern
```

softmax regression

- softmax regression

```
#1. declaring placeholder
```

```
x = tf.placeholder(tf.float32, [None, 2]) # height, width  
y = tf.placeholder(tf.float32, [None, 3]) # pattern
```

```
#2. declaring variables
```

```
w = tf.Variable(tf.zeros([2, 3])) # weight  
b = tf.Variable(tf.zeros([3])) # bias
```

```
#3. defining softmax regression
```

```
y_hat = tf.nn.softmax(tf.matmul(x, w) + b)
```

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

$$y = \sigma(W \bullet \mathbf{z} + b)$$

learning

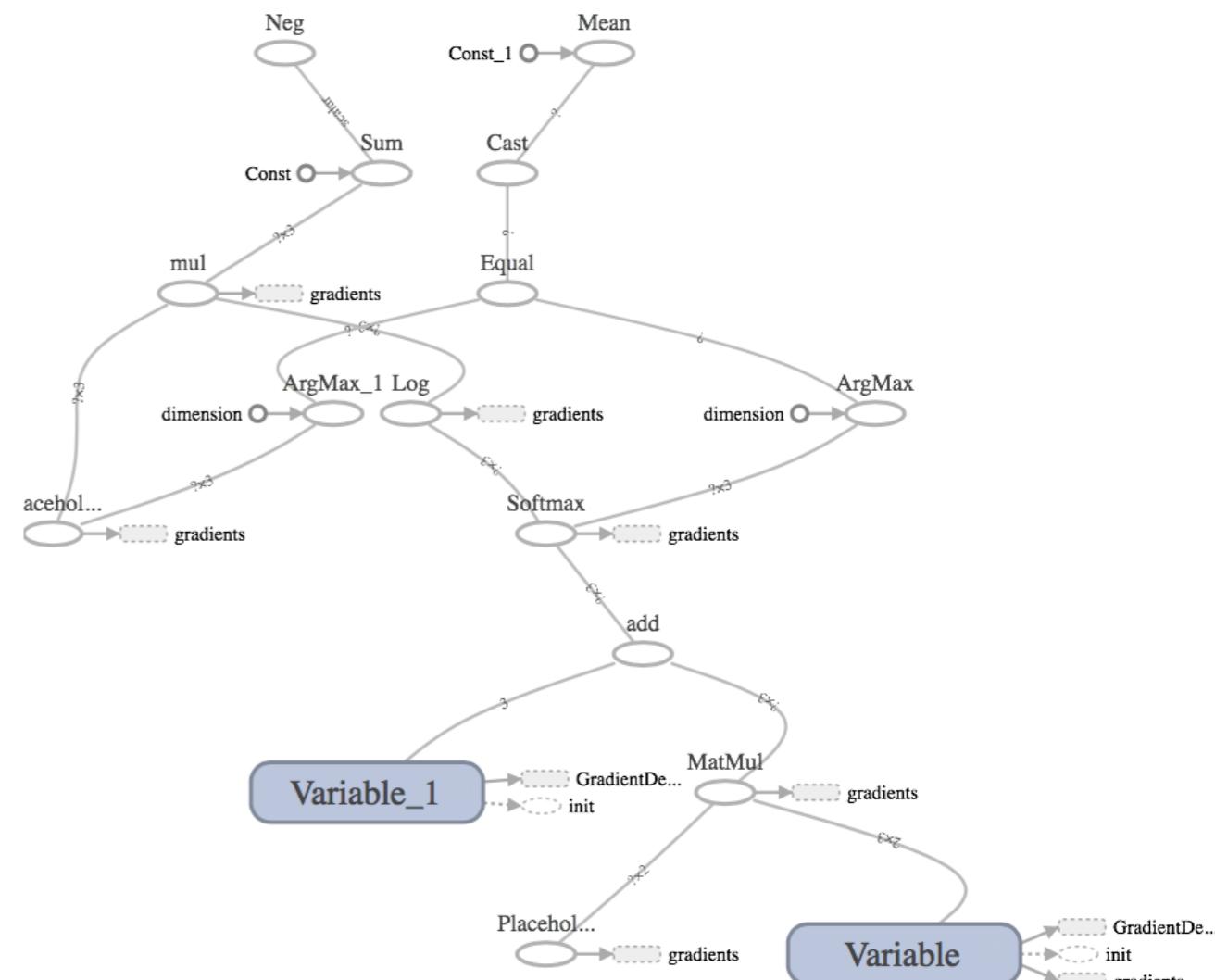
- train: cost function을 정의내리고 cost를 극소화하는 W, b 를 찾는 것
 - OLS의 cost function: square sum of errors
 - 여기에서는 entropy로 정의
- y : real value
- y_{hat} : predicted value

```
# model learning
```

```
cross_entropy = -tf.reduce_sum(y*tf.log(y_hat))
optimizer = tf.train.GradientDescentOptimizer(0.01)
train = optimizer.minimize(cross_entropy)
```

TensorBoard

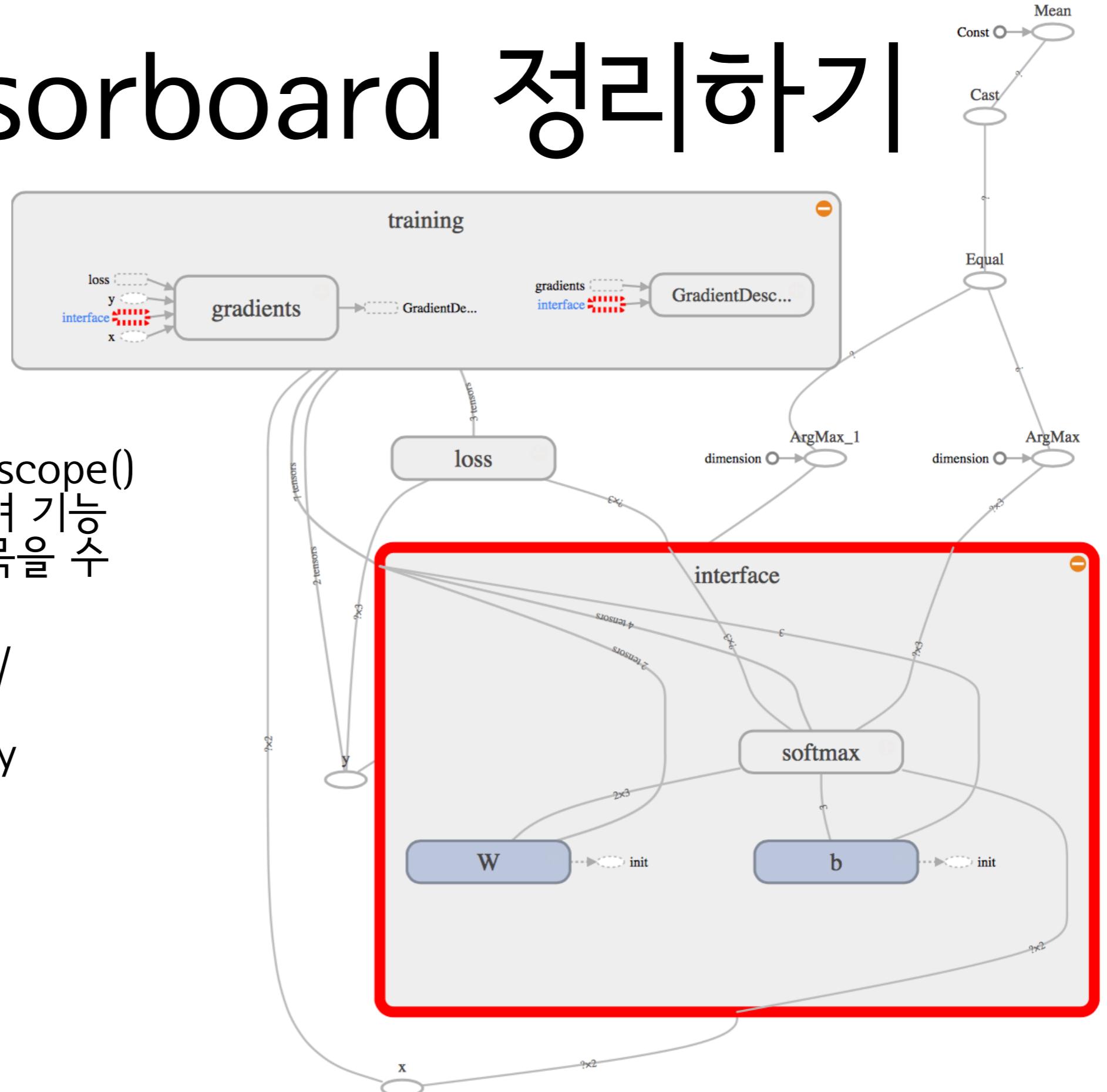
- 데이터 플로우 시각화 도구
- train 실행시
SummaryWriter를 병행 실행
- 첫번째 인자 디렉토리에 정 보 저장
- tensorboard 명령어로 실행
→ 브라우저상에서 확인
- tf 1.0 이후 부터는
tf.train.SummaryWriter 가 아니라
tf.summary.FileWriter임.



_sandbox/_ch05/bmi_tb.py

Tensorboard 정리하기

- `tb.name_scope()`를 사용하여 기능 모듈별로 묶을 수 있음
- `_sandbox/_ch05/bmi_tb2.py`



Keras

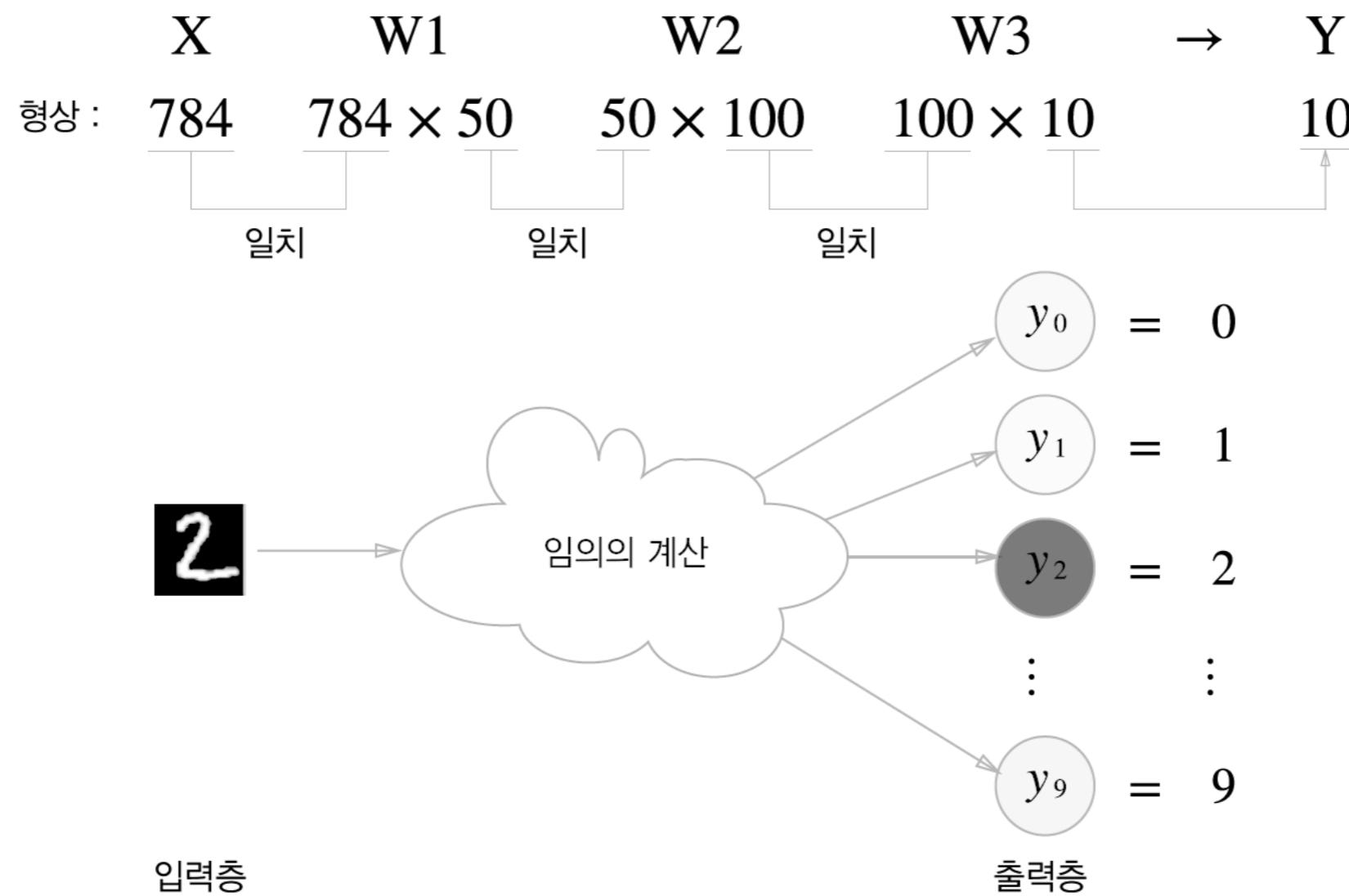
- <https://keras.io/>
- Deep Learning library for Theano and TensorFlow
 - \$ pip3 install keras
 - ~/.keras/keras.json # 설정파일
- 좀 더 모델에 집중할 수 있도록 모듈화

```
{  
    "image_dim_ordering": "tf",  
    "epsilon": 1e-07,  
    "floatx": "float32",  
    "backend": "tensorflow",  
}
```

MNIST 손글씨 인식

IMAGE:
 $28 \times 28 = 784$ pixels

7	2	1	0	4	1	4	9	5	9
0	6	9	0	1	5	9	7	3	4



ex) keras-handwriting.py

- _sandbox/_ch05/keras-handwriting.py
- MNIST 손글씨 데이터 사용
 - 훈련 데이터 #60000
 - 테스트 데이터 #10000
 - <http://yann.lecun.com/exdb/mnist/>
- 핵심은 model
 - 각 층을 add로 추가 → compile → fit

```
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation
from keras.optimizers import Adam
from keras.utils import np_utils

# reading mnist data
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# data postprocessing
X_train = X_train.reshape(60000,784).astype('float32')
X_test = X_test.reshape(10000,784).astype('float')
X_train /= 255
X_test /= 255

# converting label data to array type
y_train = np_utils.to_categorical(y_train,10)
y_test = np_utils.to_categorical(y_test,10)

# defining model structure
model = Sequential()
model.add(Dense(512, input_shape=(784,)))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(Dense(10))
model.add(Activation('softmax'))

# making model
model.compile(
    loss='categorical_crossentropy',
    optimizer=Adam(),
    metrics=['accuracy'],
)

# training
hist = model.fit(X_train, y_train)

# testing
score = model.evaluate(X_test,y_test,verbose=1)
print('loss=',score[0])
print('accuracy=',score[1])
```

Curse of Dimensionality

- 특성량의 차원수에 비해 학습 데이터가 한정적일 경우 목표하고자 하는 성능을 내지 못하는 상황
- 학습된 데이터 분류는 해내지만 새로운 데이터에 대해서는 분류를 잘 해내지 못하는 경우
- 학습 데이터가 한정적일 경우 문제가 복잡할수록 문제가 됨

보론: 웹에서 데이터 추출 (크롤링, 스크레이핑)

* 이 부분은 서울대학교 사회학과 강동현 님의 도움을 크게 받았음을 밝힙니다.

웹으로부터 데이터 수집

NAVER 금융

종목명·펀드명·환율명·원자재명 입력

통합검색

그리고

1

메일

추천종목

금융 홈 국내증시 해외증시 시장지표 펀드 투자전략 뉴스 MY금융 추천종목

· [금융감독원] 전자공시시스템 · [한국거래소] 공매도 종합 포털 · [한국은행] 100대 통계지표 · [네이버뉴스] 그래픽으로 보는 경제 · IPO정보 오픈

세계 주요증시 현황

다우산업	21,580.07 ▼31.71 -0.15%	나스닥종합	6,387.75 ▼2.25 -0.04%
	21,622.54 21,590.26 21,557.97 21,525.69 21,493.41		21,580.07 21,571.38 21,562.69 21,554.00 21,545.21
상해종합	3,250.60 ▲12.617 +0.39%	S&P 500	2,472.54 ▼0.91 -0.04%
	3,238.00 3,248.67 3,250.60 3,252.00 3,254.33		2,474.29 2,471.77 2,469.26 2,466.75 2,464.23
프랑스	5,121.65 ▲3.99 +0.08%	항셍	26,846.83 ▲140.74 +0.53%
	5,127.04 5,127.04 5,114.78 5,102.53		26,846.83 26,846.83 26,846.83 26,846.83 26,846.83
니케이 225	19,975.67 ▼124.08 -0.62%	독일	12,195.01 ▼45.05 -0.37%
	19,975.67 19,963.67 19,951.67 19,939.67 19,927.67		12,195.01 12,195.01 12,195.01 12,195.01 12,195.01
영국	7,384.38 ▼68.53 -0.92%		

주요뉴스

더보기

최근 조회

MY STOCK

최근 조회 종목이 없습니다.

해외 주요지수

현지시간기준

더보기

해외 인기검색

1. 항셍 차이나기업(H) +0.31%
2. 상해종합 +0.39%
3. 나스닥 종합 -0.04%
4. 인도 SENSEX +0.68%
5. 독일 DAX30 -0.41%

투자 가이드

- 해외증시 거래시간

국가명	지수명	현재가	전일대비	등락률	등락률 그래프	시간
미국	다우 산업	21,580.07	▼ 31.71	-0.15%		07.21 16:35
미국	다우 운송	9,471.27	▼ 11.82	-0.12%		07.21 16:35
미국	나스닥 종합	6,387.75	▼ 2.25	-0.04%		07.21 16:01
미국	나스닥 100	5,921.53	▲ 0.30	+0.01%		07.21 16:01
미국	S&P 500	2,472.54	▼ 0.91	-0.04%		07.21 16:35

```

<li class="on">
<dl>
<dt>
class="point_up">
class="dt"><a href="/world/sise.nhn?symbol=DJI@DJI" onclick="clickcr(this,'mit.dowt','','','event')"><span class="blind">다우 산업</span></a></dt>
<dd>
class="point_status"><strong>14,578.54</strong><em>52.38</em><span><span>+</span>0.36%</span><span class="blind">상승</span></dd>
<dd>
class="graph"><a href="/world/sise.nhn?symbol=DJI@DJI" onclick="clickcr(this,'mit.dowc','','','event')"></a></dd>
<dd>
class="date"><span class="date"><em>2013.03.28</em>&ampnbsp16:36</em> 기준</span></dd>

```

다우 산업: 21,580.07

2-1 로그인 필요한 사이트 에서 다운받기

- 1. http protocol

- 기본적으로는 이용자가 url에 접근하려 할 때 서버에 요청하면 서버가 응답해주는 구조. 같은 URL에 여러 번 접근해도 같은 데이터를 준다는 의미에서 stateless 통신이라고 이야기 함. 조금 구체적으로는 어떤 데이터를 유저가 가져갔는지에 대한 state를 저장하지 않음.

- 2. 쿠키 (Cookie)

- 과거의 정보를 방문자 컴퓨터가 일시적으로 http 기반으로 저장하는 것을 의미함. (헨젤과 그레텔이 지나온 길 표시하기 위해 쿠키 떨어뜨린 데서 유래.) 1개 쿠키에 저장할 수 있는 데이터 크기는 4096 byte로 제한. (방문자 컴퓨터 메모리에 저장)

- 3. 세션 (Session)

- 쿠키와는 다르게 웹 서버에서 정보가 저장이 됨. 서버 쪽에서 발행한 세션id를 key로 삼아 통신을 진행하게 됨. Stateless 통신과 구별하여 stateful 통신이라고 지칭.

2-1 로그인 필요한 사이트 에서 다운받기

- requests 모듈
- urllib.request보다는 외부 모듈인 requests를 사용한다. (pip 이용하여 설치할 것 pip3 install requests)
- 예제는 한빛출판 네트워크를 활용해서 제시함.
 - <http://www.hanbit.co.kr/member/login.html>
 - <http://www.hanbit.co.kr/myhanbit/myhanbit.html>

Selenium

- 자바스크립트를 많이 활용하는 웹사이트의 경우는 requests 모듈로 대처가 불가능함. (Ajax로 데이터를 나중에 가져오는 경우)
- 이런 경우는 원격 조작이 필요한데 이 때 사용하는 것이 Selenium. 해당 모듈을 통해 chrome이나 firefox 같은 웹브라우저에 접근 가능
- pip로 인스톨 할 것.

웹 브라우저 원격 조작에 사용하는 Selenium

- 화면 없이 명령줄에서 사용할 수 있는 웹 브라우져.
사파리와 같은 엔진 사용.
- <http://phantomjs.org/download.html> 가서 다운
로드 하는게 편함.

PhantomJS

웹 브라우저 원격 조작에 사용하는 Selenium

The screenshot shows the official website for PhantomJS. At the top, there's a dark header bar with the "PhantomJS" logo (a blue bell icon) on the left and links for "SOURCE CODE", "DOCUMENTATION", "API", "EXAMPLES", and "FAQ" on the right. Below the header is a yellow banner containing the text: "Please take a moment to [improve this document](#) with anything that could be useful to other developers, we'd love to see it."

Documentation

Get Started

- [Download](#)
- [Build](#)
- [Releases](#)
- [Release Names](#)
- [REPL](#)

Learn

- [Quick Start](#)
- [Headless Testing](#)

Download

Note There is no need to ask when a binary package for a given platform will be ready. The packagers are fully aware of every release and they give their best effort to make the binaries available.

Windows

Download [phantomjs-2.1.1-windows.zip](#) (17.4 MB) and extract (unzip) the content.

The executable `phantomjs.exe` is ready to use.

윈도우의 경우 zip 파일 받아서 압축 해제 시키면 exe 파일 생성 됨.

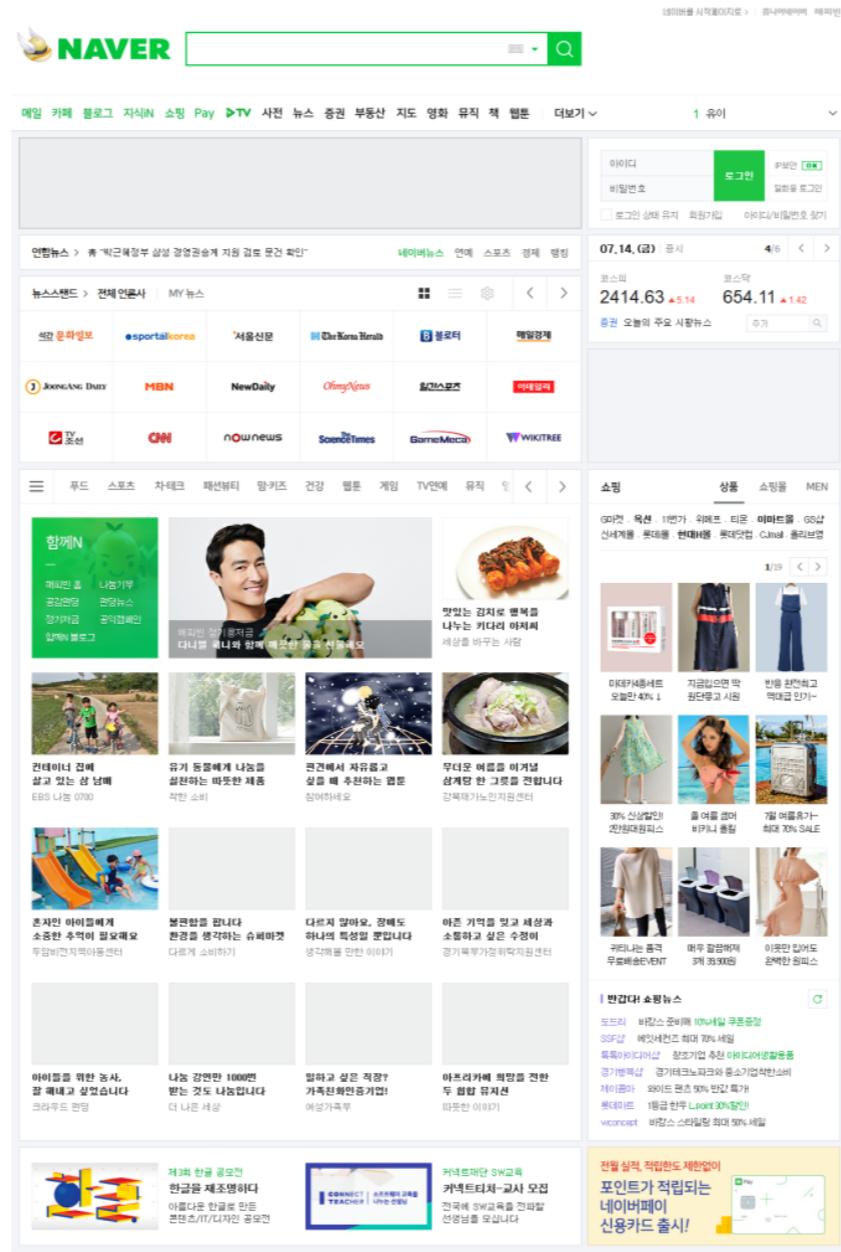
Selenium → phantomJS

```
In [9]: from selenium import webdriver  
  
url = "http://www.naver.com"  
phantom_path = "C:\\Users\\sncc\\Downloads\\phantomjs-2.1.1-windows\\bin\\phantomjs.exe"  
browser = webdriver.PhantomJS(phantom_path)  
browser.implicitly_wait(3)  
browser.get(url)  
browser.save_screenshot("naver.png")  
print('saved')  
browser.quit()
```

saved

Selenium 내 webdriver class를 가지고 온 뒤, phantomJS에 접근하는 방식.

Naver.png



naver.png

2-2 웹브라우저를 이용한 스크래이핑

네이버 쇼핑 구매목록 스크래핑하기

The screenshot shows the Naver Pay homepage with a green header bar. The header includes the NAVER logo, a search bar with '네이버쇼핑', and user information like '장바구니' and 'gambug'. Below the header, there's a navigation bar with tabs: 결제내역, 포인트, 송금, 선물함, 이벤트·쿠폰, and 구매평. The main content area has a sidebar on the left with a user profile for 'gambug님' showing '네이버페이 포인트 2,721원', and links for 기본 설정, 알림수신 설정, 카드 관리, 계좌 관리, 비밀번호, and 배송지 관리. The main content area displays a summary of recent purchases: 결제확인/완료 0, 배송중/완료 0, 취소/반품/교환 0. It also features a section titled '쇼핑 짐·Q&A를 쇼핑MY에서 확인하세요' with a link to '찜한상품'.

최근 6개월 동안의 내역이 없습니다.

<https://order.pay.naver.com/home?tabMenu=SHOPPING>

2-2 웹브라우저를 이용한 스크래이핑

네이버 쇼핑 구매목록 스크래핑하기

```
In [16]: from selenium import webdriver
phantom_path = "C:\Users\sncc\Downloads\phantomjs-2.1.1-windows\bin\phantomjs.exe"

def naver_shopping_lst(user_id, pw):
    browser = webdriver.PhantomJS(phantom_path)
    browser.implicitly_wait(3)
    url_login = "https://nid.naver.com/nidlogin.login"
    browser.get(url_login)
    print("로그인 페이지 접근")

    id_element = browser.find_element_by_id('id') #id 요소로 element 추출
    id_element.clear() #아이디 입력란 clear
    id_element.send_keys(user_id)
    pw_element = browser.find_element_by_id("pw")
    pw_element.clear()
    pw_element.send_keys(pw)

    form = browser.find_element_by_css_selector("input.btn_global[type=submit]")
    form.submit()
    print("로그인 버튼 클릭")

    browser.get("http://order.pay.naver.com/home?tabMenu=SHOPPING")

    products = browser.find_elements_by_css_selector(".p_info span")
    print(products)
```

```
로그인 페이지 접근
로그인 버튼 클릭
[]
```

2-1 로그인 필요한 사이트 에서 다운받기

① 안전하지 않음 | www.hanbit.co.kr/member/login.html

HOME 한빛미디어 한빛아카데미 한빛비즈 한빛라이프 한빛에듀 리얼타임 한빛정보교과서

로그인 회원가입

BRAND Channel.H STORE SUPP

로그인 | 아이디 찾기 | 비밀번호 찾기 | 회원가입 | 이용약관 | 개인정보취급방침

아이디

비밀번호

로그인

아이디 저장

아이디 찾기 비밀번호 찾기 회원가입

저자 직강으로만 준비했
한빛 온라인
자세히 보기

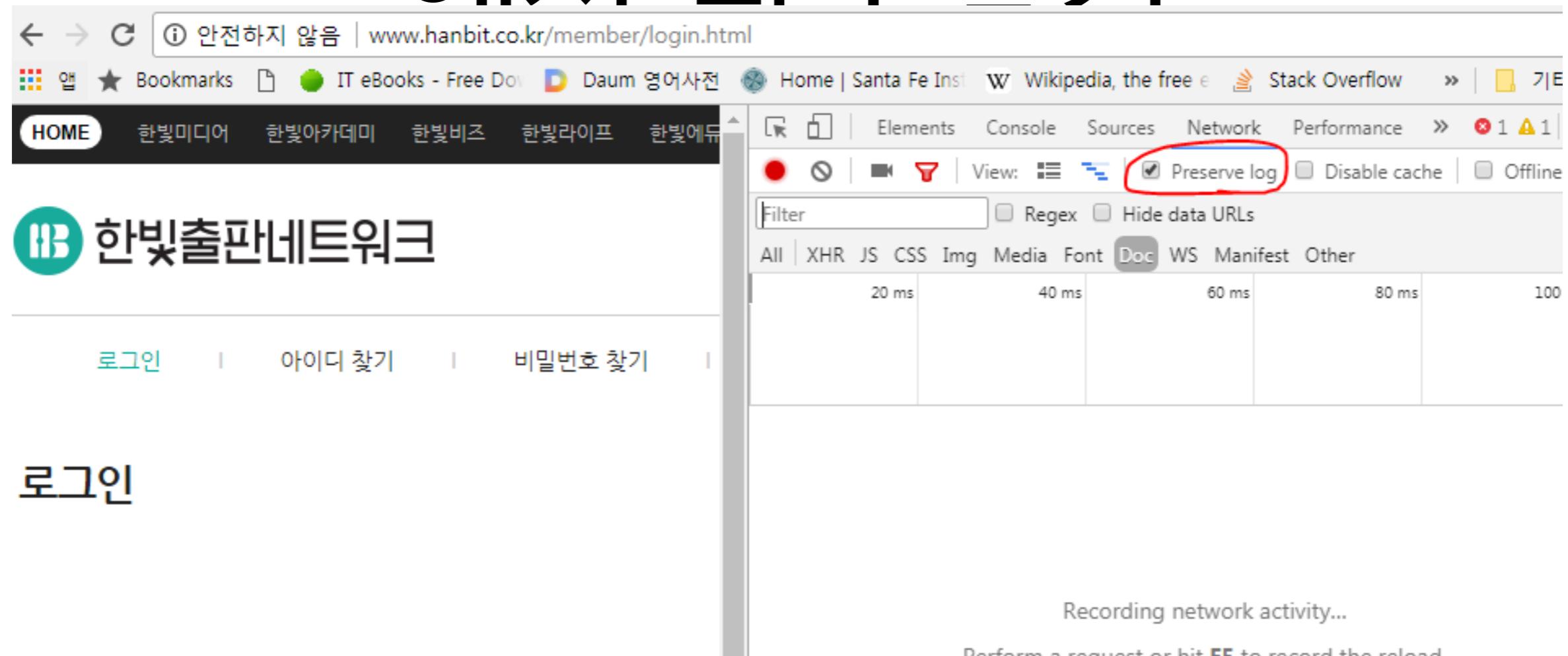
[http://www.hanbit.co.kr/
member/login.html](http://www.hanbit.co.kr/member/login.html)

2-1 로그인 필요한 사이트 에서 다운받기

The screenshot shows a web browser window with the URL www.hanbit.co.kr/myhanbit/myhanbit.html. The page header includes the Hanbit logo and navigation links like HOME, 한빛미디어, 한빛아카데미, 한빛비즈, 한빛라이프, 한빛에듀, 리얼타임, 한빛정보교과서, and 로그아웃. Below the header, there are two main sections: 'My Book' and 'My eBook'. The 'My Book' section features a circular badge labeled 'Family' and text indicating the user is a 'Family' member. The 'My eBook' section displays a box showing 2,000 points and 0 won. A sidebar on the right shows recent purchases.

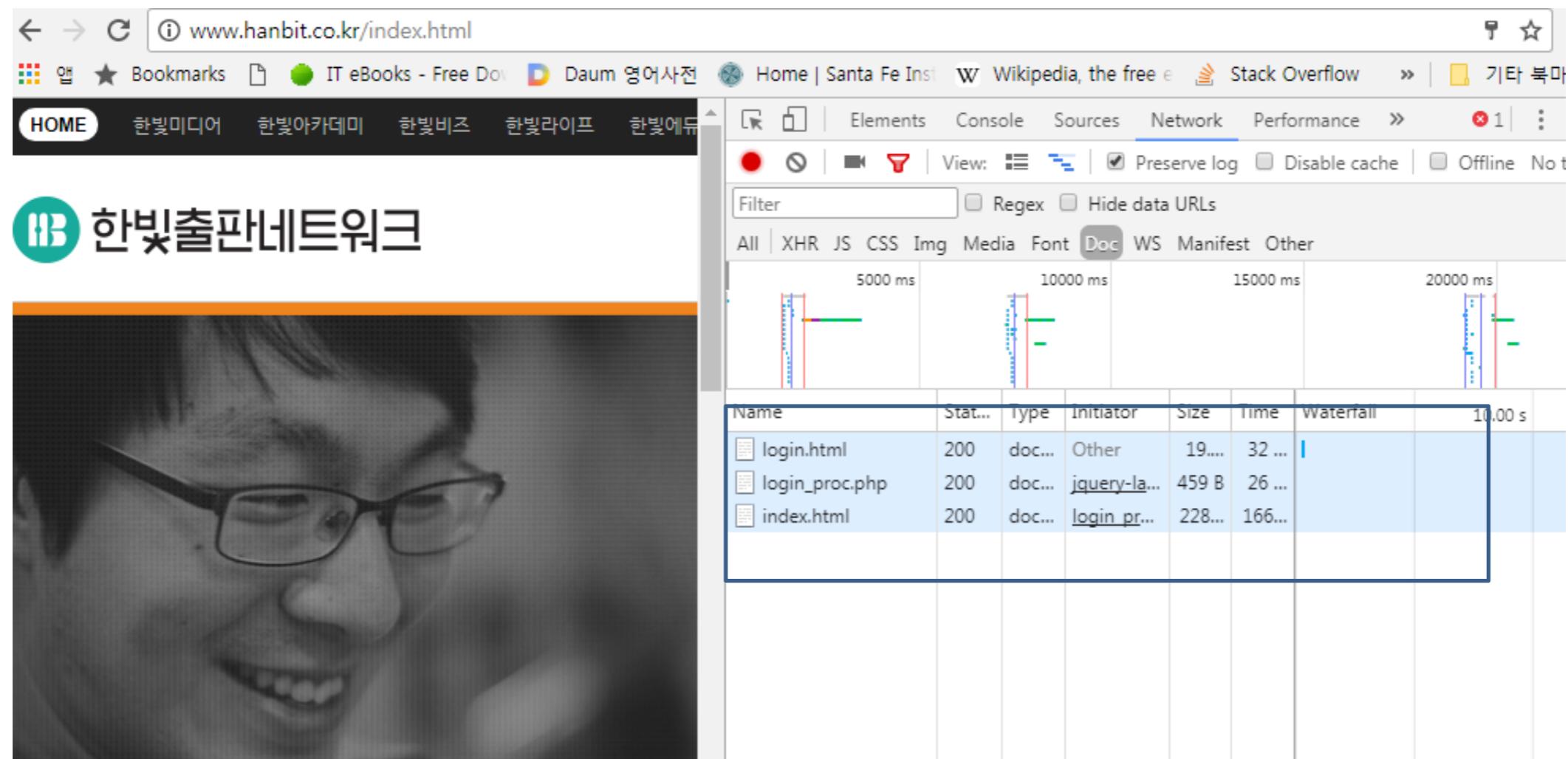
[http://www.hanbit.co.kr/
myhanbit/myhanbit.html](http://www.hanbit.co.kr/myhanbit/myhanbit.html)

2-1 로그인 필요한 사이트 에서 다우바기



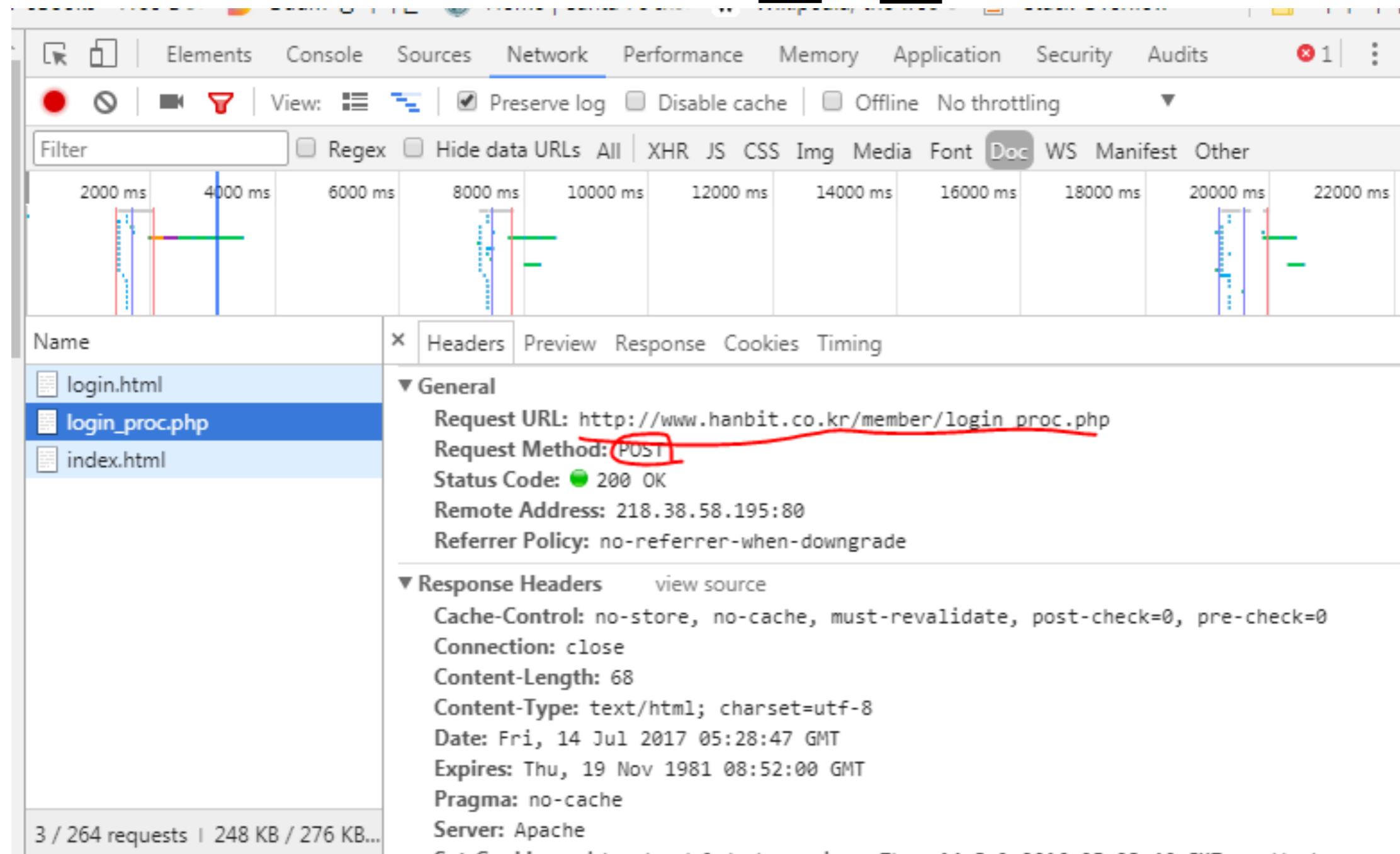
페이지 검사에서 preserve log 체크. 그 이후 로그인 해보면?

2-1 로그인 필요한 사이트 에서 다운받기



이렇게 바뀜. 그 중에서 login_proc.php 클릭

2-1 로그인 필요한 사이트 에서 다운받기



Request url 쪽으로 post를 전달했다는 이야기.

2-1 로그인 필요한 사이트 에서 다운받기

- 마일리지와 e코인 가져오기

```
In [1]: import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin

In [2]: def scrape_hanbit_data(user, pw, url_login, page_url):
    session = requests.session()
    login_info = {"m_id":user, "m_passwd":pw}
    res = session.post(url_login, data=login_info)
    res.raise_for_status()
    res = session.get(page_url)
    res.raise_for_status()

    soup = BeautifulSoup(res.text, "html.parser")
    mileage = soup.select_one(".mileage_section1 span").get_text()
    ecoin = soup.select_one(".mileage_section2 span").get_text()
    print("마일리지: " + mileage)
    print("이코인: " + ecoin)

In [3]: login_page = "http://www.hanbit.co.kr/member/login_proc.php"
my_page = "http://www.hanbit.co.kr/myhanbit/myhanbit.html"

In [4]: scrape_hanbit_data('gambug', 'testtest1', login_page, my_page)
```

마일리지: 2,000
이코인: 0

Application Programming Interface (API)

- API: Application Programming Interface. 외부에서 프로그램의 기능을 호출할 수 있게 만든 것. XML이나 JSON 형식으로 요청에 응답해주는 경우가 많음.
- 어차피 크롤링 표적이 될 거니까 서버 쪽 부담을 줄여버리자는 아이디어 + 자사 사이트 검색 기능을 활성화 (옥션, 11번가, amazon 등)
- 앱 개발시에 API를 쓰는 경우 언제든지 사라질 수 있다고 생각하는 것이 정신건강에 이로움. 큰 회사에서 만들었다고 오래가고, 개인이 만들었다고 유지보수 잘 안되고 사라져 버리는 것도 아님. 구글 같은 경우 수요가 적으면 그냥 없애버리는 경우도 비일비재 함. “없어지면 어떻게 하지?” 정도를 항상 고려할 것.

Twitter API

- <https://dev.twitter.com/overview/api>
- 트위터 데이터를 직접 제공
받을 수 있는 채널 제공

The screenshot shows the Twitter Developer Documentation API Overview page. The header features the Twitter logo and the text "Developers". Below the header, the title "Twitter Developer Documentation" is displayed in large white font on a blue background. A navigation bar below the title includes "Docs" and "API Overview". On the left side, there is a dark sidebar with the heading "Products & Services" and a list of links: Best practices, API overview, Upcoming changes to Tweets, Object: Tweets, Object: Users, Object: Entities, Object: Entities in Objects, Object: Places, Twitter IDs, Connecting to Twitter API using TLS, and Using cursors to navigate.

API Overview

Here are some resources that will help you understand the basics of all our APIs. If you haven't already, make sure that you have familiarized yourself with the [Twitter Developer Policy](#). Check out the [OAuth](#) section to learn more about how we do authentication and authorization.

API Objects

There are four main "objects" that you'll encounter in the API: [Tweets](#), [Users](#), [Entities](#) (see also [Entities in Objects](#)), and [Places](#). See the anatomy of these objects, and learn about properties like [Twitter IDs](#) or [Place Attributes](#) to know what to expect.

Connecting to and navigating around the APIs

Check out the best practices around [Connecting to Twitter API using TLS](#), [Using cursors to navigate collections](#) and [Error Codes & Responses](#) to learn how to most effectively interact with the Twitter APIs.

Twitter Libraries

수고하셨습니다!