

# ME490 자율주행시스템을 위한 프로그래밍 Programming for Autonomous System

카이스트 기계공학과

1

## Goal of this course

- Let students learn and experience how recent AI/Autonomous vehicle techniques are implemented in the mechanical system
- 학생들이 최근 주목받는 인공지능/자율주행 등 4차산업혁명 기술이 어떻게 기계공학과에서 다루는 시스템에 적용되는지를 배우고 실습하도록 하는 것
- Ultimately, students would know that the AI TOOLS are (just) tools that you could use for making machines autonomous
- 그리하여 학생들이 시스템을 발전시키는데 AI 관련 기술을 도구로 사용할 수 있다는 것을 알게 하는 것

2

# Feedback from Capstone 1

- Pros

- Challenging, fruitful curriculum
- 도전적이지만 많은 것을 배울 수 있었음
- Excellent TA support 조교들의 지원에 감사

- Cons

- Nonsystematic coding curriculum 소프트웨어에 대한 체계적 교육이 없어
- Time consuming coding and debugging 프로그래밍에 시간이 너무 많이 할애 되었음
- Rule changes 규정 등의 변경에 대한 불만

# Plans for Capstone 2

- Pros

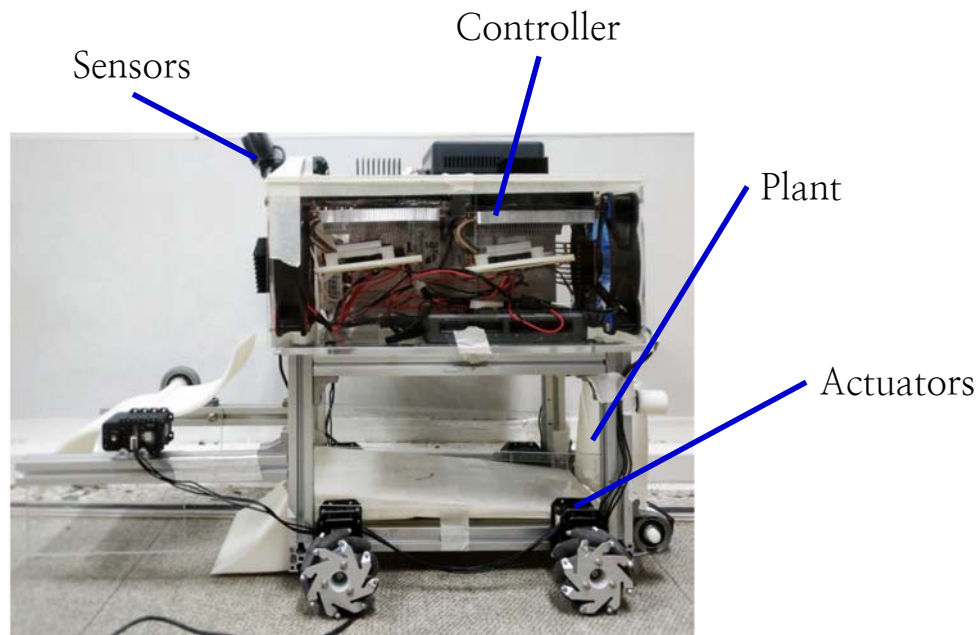
- Challenging, fruitful curriculum
- 도전적이지만 많은 것을 배울 수 있었음
- Excellent TA support 조교들의 지원에 감사

- Attempt to improve the Cons

- Programming for autonomous system ME491 창시구프로그래밍 교육 및 실습
- Rule setting: TA pre-runs (software wise)

The changes in rule (if any) would be introduced to make your life easy !

# Recall, your autonomous vehicle system



5

## Autonomous vehicle system

Controller

- Go forward
- With speed  $v$
- With direction  $\theta$ , combinations of  $n$ -wheels
- direct controller by X-box
- Previous 창시구 (시모제, 자동제어)
- New 창시구 (자율주행, 딥러닝)
- autonomous control: find out where & how to go and come back

• TOOLS for making system autonomous

“SLAM” “Machine learning”  
“Neural net CNN”  
“Reinforcement learning”  
“Deep learning”

6

# Programming for autonomous system

(Thanks to) Open source software

- Open-source software (OSS) is a type of computer software whose source code is released under a license in which [the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose.](#)
- Open-source software may be developed in a collaborative public manner.
- According to scientists who studied it, open-source software is a prominent example of open collaboration. [from wiki]

– ME490 Programming for autonomous system

– ME401 Capstone design 2

“SLAM” “Machine learning” CS376  
CE481 “Neural net CNN” CS470  
CS672 “Reinforcement learning”  
CS774 “Deep learning”

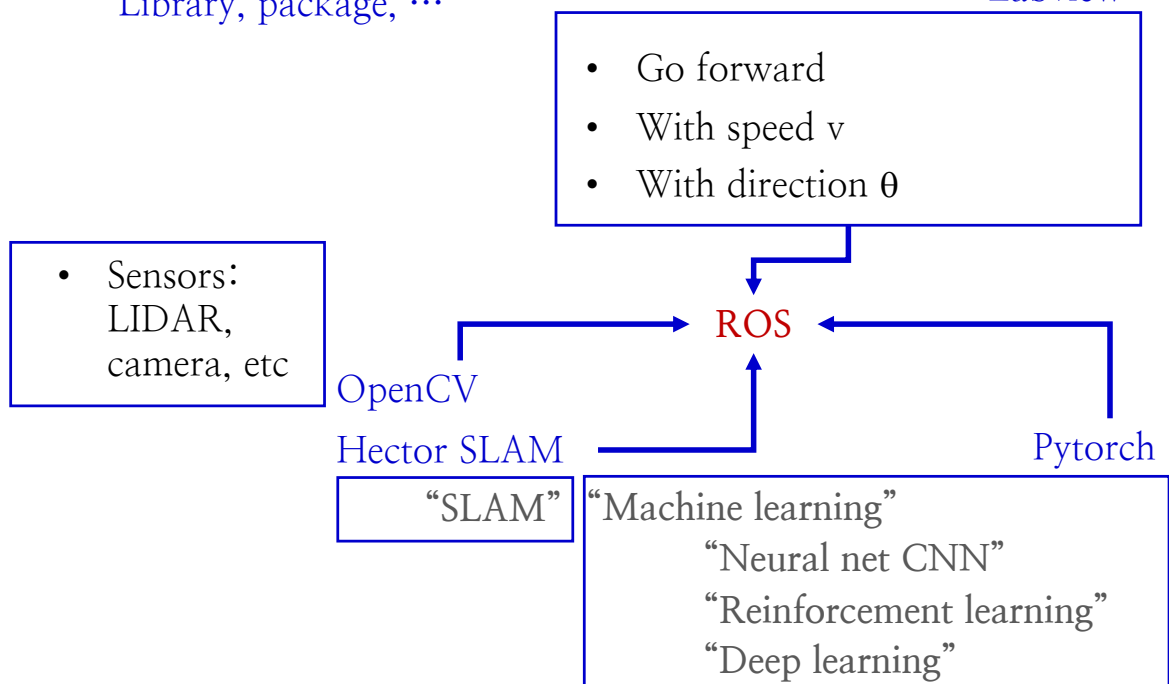
7

# Programming for autonomous system

Open source software

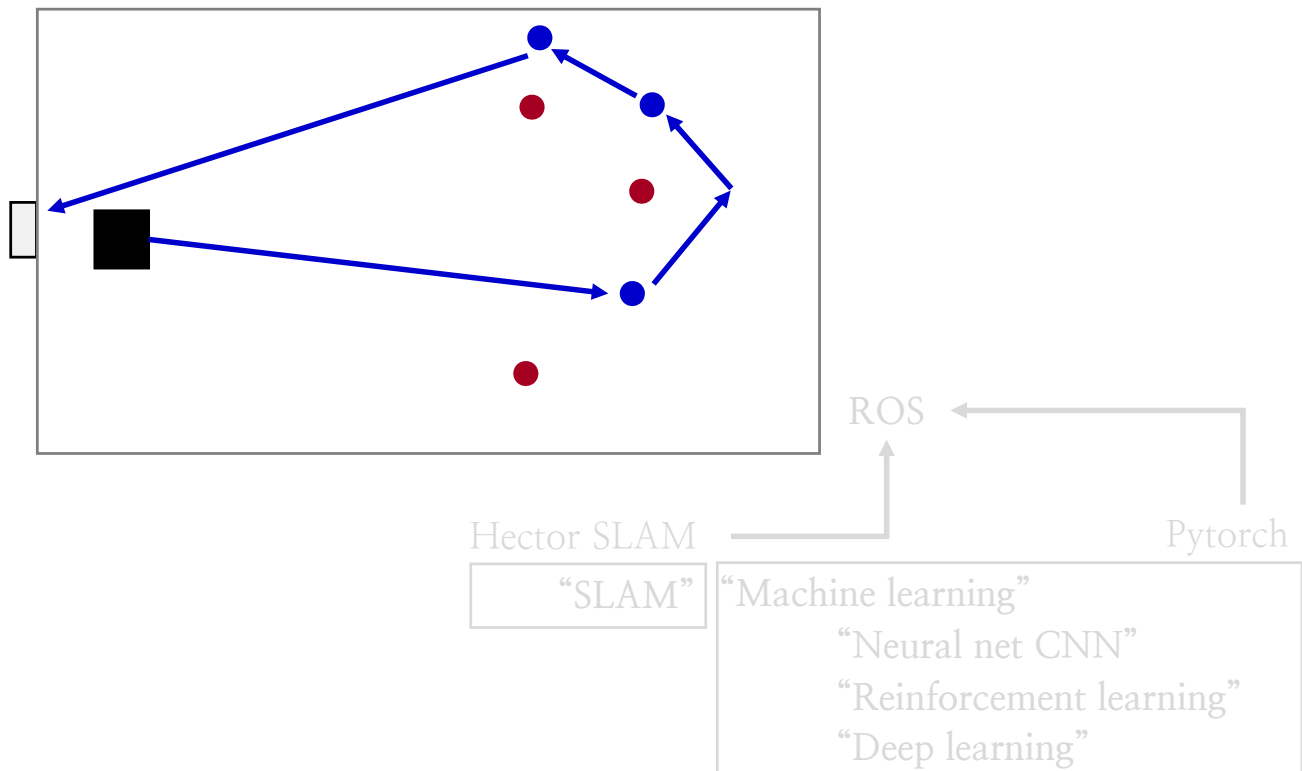
Library, package, ...

Labview

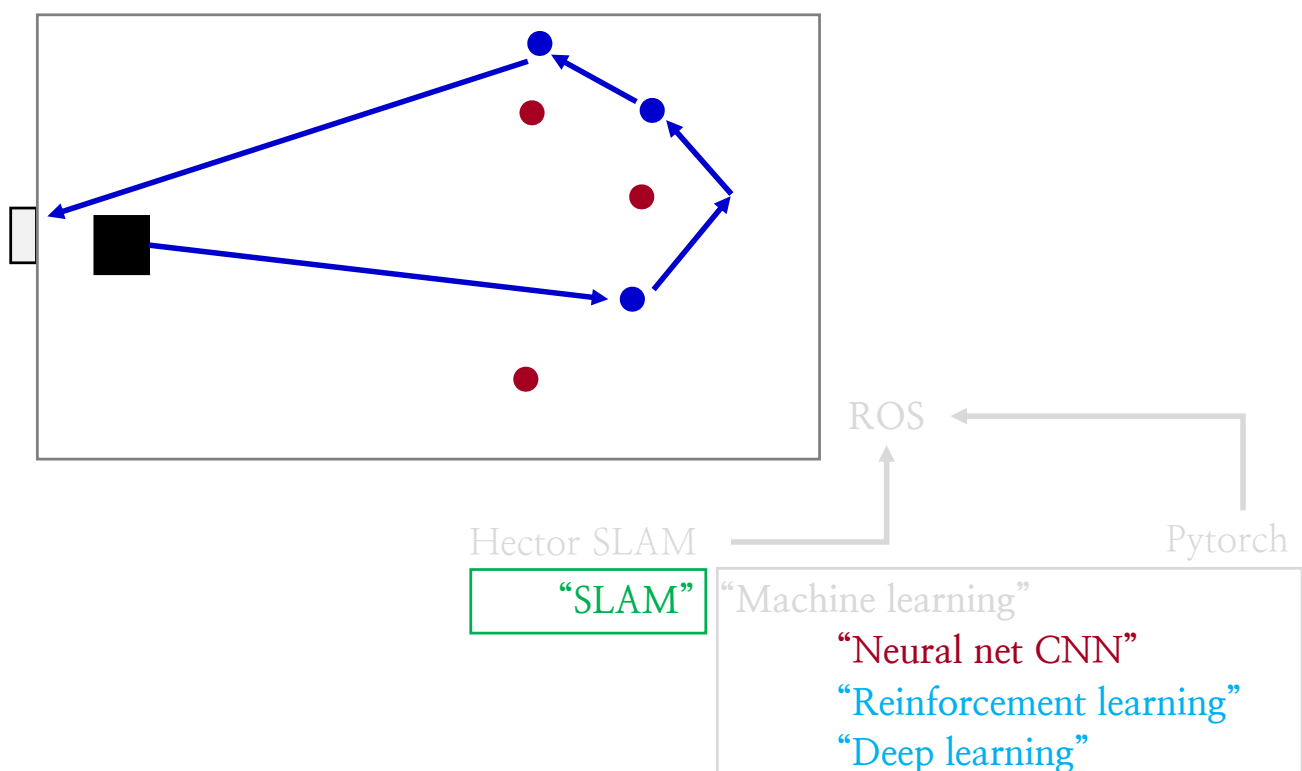


8

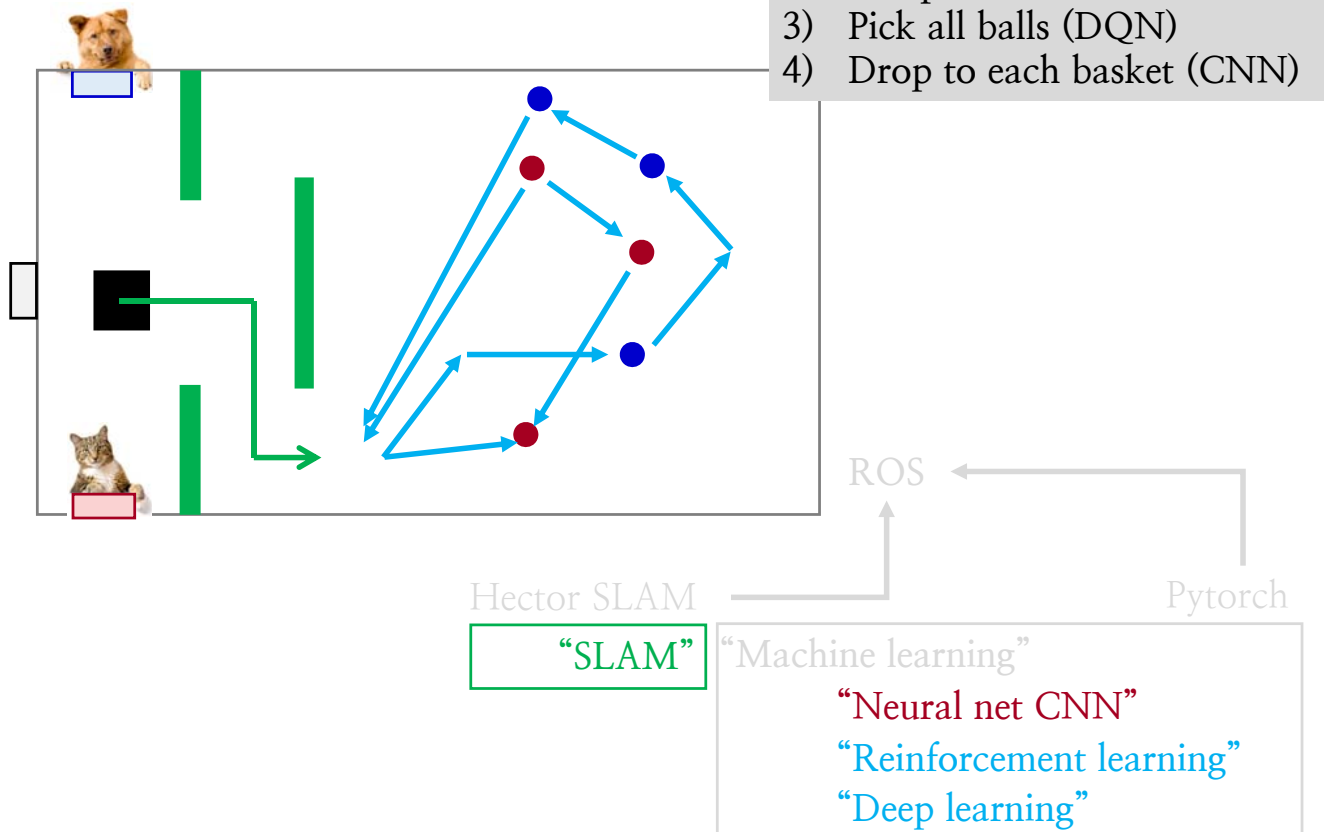
## Recall, final mission of Capstone 1



## Recall, final mission of Capstone 1



# Final mission (draft)



## Active learning (Education 4.0)

- 트랙 1: 플립드 러닝 방식 (교수님께서 사전 제작하신 동영상강의(필수)를 KLMS에서 학습 + 오프라인 상호협력학습)
- 트랙 2: 액티브 러닝 50% 이상 (수업시간 중 일방향 강의비중을 50%미만으로 줄이고, 토론/ 문제풀이/ 프로젝트 등의 다양한 학습활동(액티브 러닝) 으로 수업설계)
- Track 2: One-directional lecture: Active learning = 50:50
- 트랙 3: 액티브 러닝 100% (수업시간 중 일방향 강의 없이, 액티브러닝으로 수업설계)

	Mon	Tue	Wed	Thu
4:00-5:00	Class			
5:00-6:00	Practice			
6:00-7:00				
7:00-8:00				
8:00-9:00				

# Course schedule

W	Date	Class	Active learning
1	8/27	0. Course introduction	Programing structure: Python, PyTorch
2	9/3	1.0 Machine learning basic	Programing of MSE
3	9/10	1.1 Neural Network	Programing of MNIST or Dog/cat
4	9/17	1.2 Convolutional Neural Network	Quiz #1 Python & PyTorch Programing of CNN <i>9/21 Capstone 1st design review</i>
5	9/24-26	Choosuk	No Class
6	10/1	2.0 SLAM basic	Code review #2 due: Customized CNN Programing structure: ROS
7	10/8	2.1 SLAM package in ROS	Programing of Hector SLAM Code Assignment #3 due: Basic ROS
8	10/15-17	Midterm	No Class, No exam

13

W	Date	Class	Active learning
9	10/22	2.2 SLAM for Capstone project	Programing of SLAM & path planning <i>10/26 Capstone 2nd design review</i>
10	10/29	3.0 Reinforcement learning	Code review #4 due: Customized SLAM Programing of DQN
11	11/5	3.1 Deep Q-learning: environment	Programing of simulator / emulator
12	11/12	3.2 DQN: learning	Programing of Capstone-vehicle learning network
13	11/19	3.3. DQN: testing	Code review #5 due: DQN code Programing of Capstone-vehicle testing
14	11/26	Real system implementation	Capstone system testing <i>11/30 Capstone Final demo</i>
15	12/3		Code review #6 due: Total system code

14

# Course Syllabus

- Lecture
  - Class lecture & active learning: Mon 4:00~5:00 pm, ME building 2000
- Evaluation guideline
  - 1) Six code review reports : #1~#5 (10%, each), #6 (30% each)
  - 2) Attendance and class participation 20%
- Syllabus, weekly notices, forms, and lecture notes will be uploaded on the course web at KLMS, and github

15

## Programming structure & code review

- Data prep programming
- Architecture programming
  - Module programming
  - Inter-module control
- Interface programming
  - Interface for software modules
  - Interface for different devices
- Procedure
- Sample code(s) will be provided and you are supposed to modify them to fit your system
  - Code review of sample code
  - Code review of your own (customized) code

16



## Ex) Programming structure & code review

```
%matplotlib inline
import gym
import math
import random
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from collections import namedtuple
from itertools import count
from PIL import Image
from tensorboardX import SummaryWriter

import torch
import torch.nn as nn
import torch.optim as optim
import torch.nn.functional as F
import torchvision.transforms as T

env = gym.make('CartPole-v0').unwrapped
writer = SummaryWriter()

# set up matplotlib
is_ipython = 'inline' in matplotlib.get_backend()
if is_ipython:
    from IPython import display
```

파이썬의 필요한 모듈을 임포트  
import necessary module

\*모듈(Module)은 파이썬 코드를 논리적으로 묶어서 관리하고 사용할 수 있도록 하는 것으로, 보통 하나의 파이썬 .py 파일이 하나의 모듈이 된다. 모듈 안에는 함수, 클래스, 혹은 변수들이 정의될 수 있으며, 실행 코드를 포함할 수도 있다

모듈안의 함수만 임포트  
import necessary function from module

\*하나의 모듈 안에는 여러 함수들이 존재할 수 있는데, 이 중 하나의 함수만을 불러 사용

머신 러닝 라이브러리인 파이토치 임포트  
import PyTorch, a machine learning library

시뮬레이션 라이브러리 gym에서 cartpole 시뮬레이션 환경을 불러서 env라는 클래스명으로 저장

학습 시키는 네트워크의 성능을 실시간으로 저장하여 관측하기 위해 tensorboardX의 함수 정의

17

## Ex) Programming structure & code review

```
def sample(self, batch_size):
    return random.sample(self.memory, batch_size)

def __len__(self):
    return len(self.memory)

class DQN(nn.Module):

    def __init__(self):
        super(DQN, self).__init__()
        self.conv1 = nn.Conv2d(3, 16, kernel_size=5, stride=2)
        self.bn1 = nn.BatchNorm2d(16)
        self.conv2 = nn.Conv2d(16, 32, kernel_size=5, stride=2)
        self.bn2 = nn.BatchNorm2d(32)
        self.conv3 = nn.Conv2d(32, 32, kernel_size=5, stride=2)
        self.bn3 = nn.BatchNorm2d(32)
        self.head = nn.Linear(448, 2)

    def forward(self, x):
        x = F.relu(self.bn1(self.conv1(x)))
        x = F.relu(self.bn2(self.conv2(x)))
        x = F.relu(self.bn3(self.conv3(x)))
        return self.head(x.view(x.size(0), -1))
```

batch 사이즈에 맞게 저장된 데이터 셋을 랜덤하게 선택해주는 함수

메모리에 할당된 사이즈 출력 함수

DQN\_모델을 정의하는 클래스

DQN 네트워크의 layer구조를 설정 여기서는 CNN을 사용

Forward step에서 넣어줄 activation function 정의

18

## Ex) Programming structure & code review

```
TARGET_UPDATE = 10
```

```
policy_net = DQN().to(device)
target_net = DQN().to(device)
target_net.load_state_dict(policy_net.state_dict())
target_net.eval()
```

```
optimizer = optim.RMSprop(policy_net.parameters())
memory = ReplayMemory(10000)
```

```
steps_done = 0
```

```
def select_action(state):
    global steps_done
    sample = random.random()
    eps_threshold = EPS_END + (EPS_START - EPS_END) * \
        math.exp(-1. * steps_done / EPS_DECAY)
    steps_done += 1
    if sample > eps_threshold:
        with torch.no_grad():
            return policy_net(state).max(1)[1].view(1, 1)
    else:
        return torch.tensor([random.randrange(2)], device=device, dtype=torch.long)
```

## Ex) Programming structure & code review

전체 프로그래밍의 기능적 모듈 등에 대한 분석

Also analyze the functional modules of main-simulator-test\_DQN code

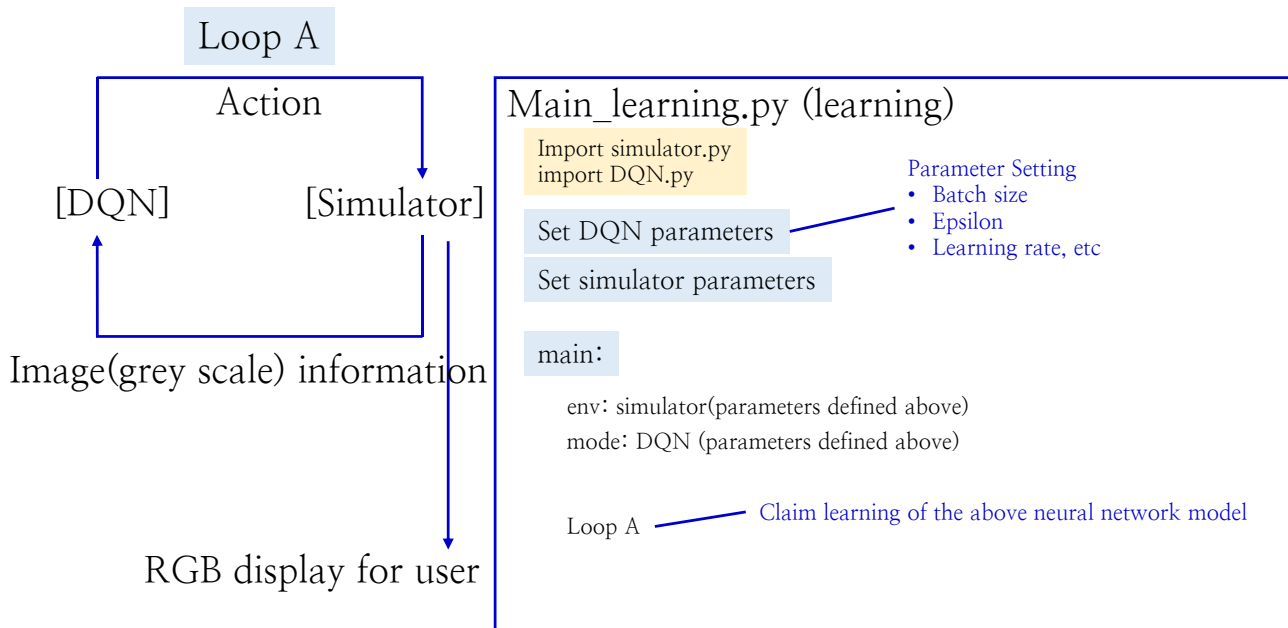
Main.py (learning)

Simulator.py

Test\_DQN.py

1850

- 각 모듈 코드의 개념, 구조, 세부 내용에 대한 리뷰 작성
- Review the concept, sub-structure of the code



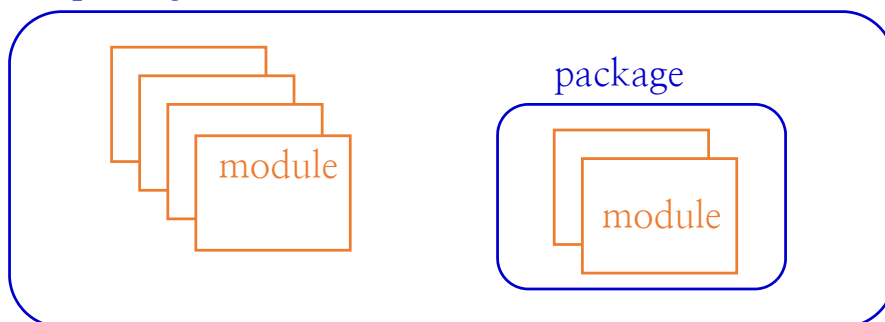
Q: Are you familiar to functions, modules, packages, class in Python?

21

1000

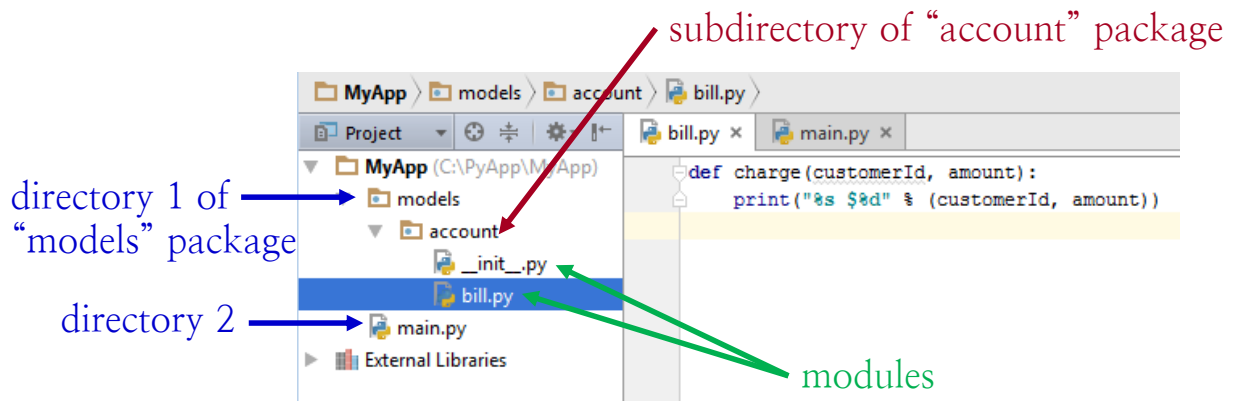
- 파이썬에서 모듈은 하나의 .py 파일을 가리키며, 패키지는 이러한 모듈들을 모은 컬렉션을 가리킴
- In Python, a module means a single .py file, and a package means a collection of these modules.
- 파이썬의 패키지는 하나의 디렉토리에 놓여진 모듈들의 집합을 가리키는데, 그 디렉토리에는 일반적으로 \_\_init\_\_.py 라는 패키지 초기화 파일이 존재
- A package in Python refers to a set of modules placed in a directory, which usually contains a package initialization file called \_\_init\_\_.py

package



# Python package

- 파이썬으로 큰 프로젝트를 수행하게 될 때, 모든 모듈을 한 디렉토리에 모아 두기 보다는 각 영역별로 디렉토리/서브디렉토리를 만들고 계층적인 카테고리 묶어서 패키지별로 관리하는 것이 편리하고 효율적이다.
- When you run a large project with Python, it's convenient and efficient to group all the modules in a hierarchical under the directories/subdirectories rather than a directory and manage them on a package-by-package basis



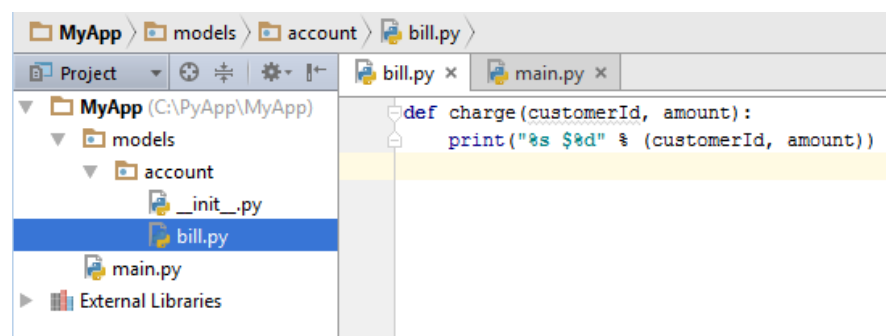
<http://pythonstudy.xyz/python/article/>

23

# Import Python package

- 패키지내 모듈을 import하기 위해서는 "import 패키지명.모듈명"과 같이 패키지명을 앞에 붙여 사용한다.
- To import the module in the package, use the package name prefixed with "import package name.module name".

```
1 | # 모듈 import  
2 | # import 패키지.모듈  
3 | import models.account.bill  
4 | models.account.bill.charge(1, 50)  
package name      module name      function name
```

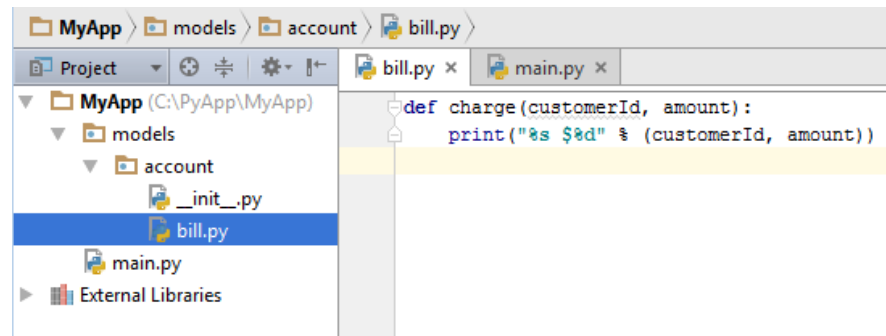


24

# Import Python package

- To import a module, try "from package import module"
- To import a function, try "from package.module import function"

```
1 # 모듈안의 모든 함수 import
2 # from 패키지명 import 모듈명
3 from models.account import bill      module name
4 bill.charge(1, 50)                  package name
5
6 # 특정 함수만 import
7 # from 패키지명.모듈명 import 함수명
8 from models.account.bill import charge  function name
9 charge(1, 50)                        package + module name
```



<http://pythonstudy.xyz/python/article/>

25

# Machine learning code...

```
import torch
class MyReLU(torch.autograd.Function):
    """
    XXXX
    def forward(ctx, input):
        """
        XXXX

    def backward(ctx, grad_output):
        """
        XXXX
        input, = ctx.saved_tensors
        grad_input = grad_output.clone()
        grad_input[input < 0] = 0
        return grad_input

dtype = torch.float
```

[<http://busterworld.tistory.com/64> [BW]]

26

# Python Class

- 클래스는 데이터와 그에 대한 처리를 하나로 정의한 것으로 함수와 유사 개념
- A class defines data and its processing as one group

```
class Bdb:
    """XXXXXXXXX

    count = 0

    def __init__(self, skip=None):
        self.skip = set(skip) if skip else None
        self.breaks = {}
        self.fncache = {}

    def canonic(self, filename):
        if filename == "<" + filename[1:-1] + ">":
            return filename
```

Description code

<http://www.dongwun.com/tc/142>

27

## Method: member of Python class

- 메서드(method): 클래스의 행위를 표현하는 것으로 클래스 내의 함수
- Method is a function in a class expressing the behavior of the class

```
class Bdb:
    """XXXXXXXXX

    count = 0

    def __init__(self, skip=None):
        self.skip = set(skip) if skip else None
        self.breaks = {}
        self.fncache = {}

    def canonic(self, filename):
        if filename == "<" + filename[1:-1] + ">":
            return filename
```

Method

<http://pythonstudy.xyz/python/article/>

28

# Class variables

- 클래스 변수 (class variable)은 메서드 밖에 존재하는 변수로 "클래스명.변수명" 으로 액세스 할 수 있다.
- Class variables is defined outside the method by "class.variable"

```
class Bdb:
```

```
    """XXXXXXXXX
```

```
    count = 0
```

← Class variable, accessed by 'Bdb.count'

```
    def __init__(self, skip=None):
        self.skip = set(skip) if skip else None
        self.breaks = {}
        self.fncache = {}
```

```
    def canonic(self, filename):
        if filename == "<" + filename[1:-1] + ">":
            return filename
```

# Instance variables

- 인스턴스 변수(Instance variable)는 메서드 안에서 사용되면서 "self.변수명" 처럼 사용된다.
- Instance variable is used in a method, such as "self.variable"

```
class Bdb:
```

```
    """XXXXXXXXX
```

```
    count = 0
```

```
    def __init__(self, skip=None):
```

Instance variable

```
        self.skip = set(skip) if skip else None
        self.breaks = {}
        self.fncache = {}
```

Class initializer that sets the  
- Instance variable  
- Initialize the object

```
    def canonic(self, filename):
        if filename == "<" + filename[1:-1] + ">":
            return filename
```

# Object in Python Class

- 클래스를 사용하기 위해 객체(Object)를 생성해야 하며 "객체변수명 = 클래스명()"과 같이 클래스명을 함수 호출하는 것처럼 사용하면 된다.
- To use a class, you first need to create an object from the class. To create an object in Python, use the class name as if you were calling a function like "object variable name = class name ()".

Class "Rectangle" is assigned to an object "r"

```
1 class Rectangle:
2     count = 0 # 클래스 변수
3
4     # 초기자(initializer)
5     def __init__(self, width, height):
6         # self.* : 인스턴스변수
7         self.width = width
8         self.height = height
9         Rectangle.count += 1
10
11     # 메서드
12     def calcArea(self):
13         area = self.width * self.height
14         return area
```

```
1 # 객체 생성
2 r = Rectangle(2, 3)
3
4 # 메서드 호출
5 area = r.calcArea()
6 print("area = ", area)
7
8 # 인스턴스 변수 액세스
9 r.width = 10
10 print("width = ", r.width)
11
12 # 클래스 변수 액세스
13 print(Rectangle.count)
14 print(r.count)
```

# Inheritance of Class

- (부모)클래스는 다른(자식) 클래스로 상속되며 상속 받기 위해서는 파생클래스(자식클래스)에서 클래스명 뒤에 베이스클래스(부모클래스) 이름을 괄호와 함께 넣어 주면 된다.
- To inherit a class, you can put the base class (parent class) name in parentheses in the derived class (child class), followed by the class name.

Dog, Duck are child class that inherits the functions from parent class "Animal"

```
1 class Animal:
2     def __init__(self, name):
3         self.name = name
4     def move(self):
5         print("move")
6     def speak(self):
7         pass
8
9 class Dog (Animal):
10     def speak(self):
11         print("bark")
12
13 class Duck (Animal):
14     def speak(self):
15         print("quack")
```



# Inheritance of Class

```
1 class Animal:
2     def __init__(self, name):
3         self.name = name
4     def move(self):
5         print("move")
6     def speak(self):
7         pass
8
9 class Dog (Animal):
10     def speak(self):
11         print("bark")
12
13 class Duck (Animal):
14     def speak(self):
15         print("quack")
```

자식클래스는 부모클래스의  
멤버들을 호출하거나 사용할  
수 있다

Child classes can call or  
use members of the parent  
class

```
1 dog = Dog("doggy") # 부모클래스의 생성자
2 n = dog.name # 부모클래스의 인스턴스변수
3 dog.move() # 부모클래스의 메서드
4 dog.speak() # 파생클래스의 멤버
```

<http://pythonstudy.xyz/python/article/>

33

## Active learning : week 1

- Python review
- Python module, package, class coding practice
- PyTorch review

# 1.0 Machine learning basic

35

## AI, Machine learning, Deep learning

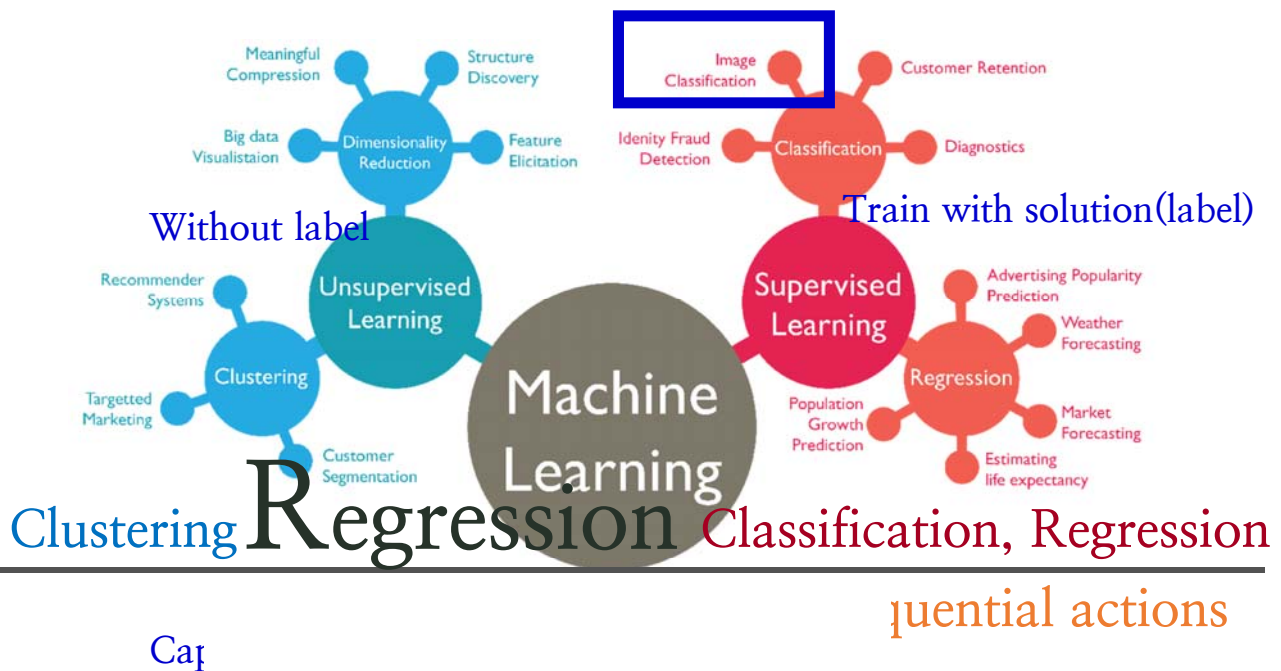
- Artificial Intelligence: technique which enables computers to mimic human behavior

- Machine learning: Subset of AI technique which uses statistical methods to enables machines to improve with experiences

- Deep learning: Subset of ML which make the computation of multi-layer neural networks feasible
- CNN
- RNN
- LSTM
- RL

# Types of Machine learning

Capstone design 2



[images from google]

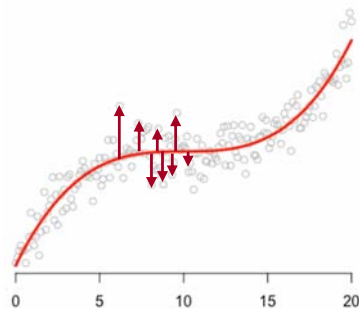
37

## Machine learning (= evolving Regression)

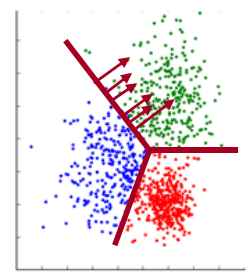
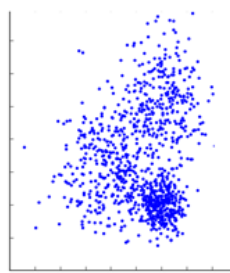
- A computer program is said to learn from **experience E** with respect to some class of **task T** and **performance measure P**, if its performance at task in **T**, as measured by **P**, improves with experience **E**. (Mitchell, 1997)
- **Task** : classification, regression, clustering
- **Performance (loss function)**: errors, distance
- **Experience**: data (labeled, unlabeled)



Rate of correct answers



Distance(=regression error)  
from model ( $y = f(x^3)$ )



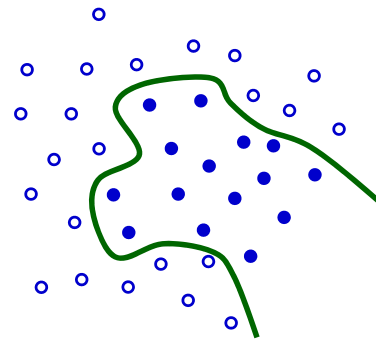
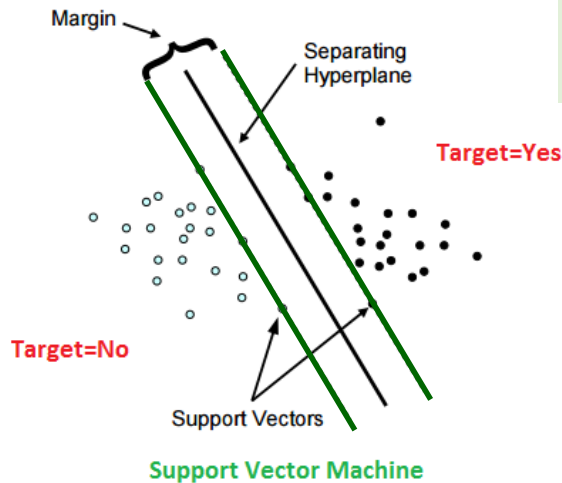
Distance from lines  
(Supporting Vector Machine)

# Classification algorithm (1 of many!)

- **Support vector machine (SVM) model** is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

Q: how to find supporting vectors?

Q: how to find a character(s) that separates the data for classification?



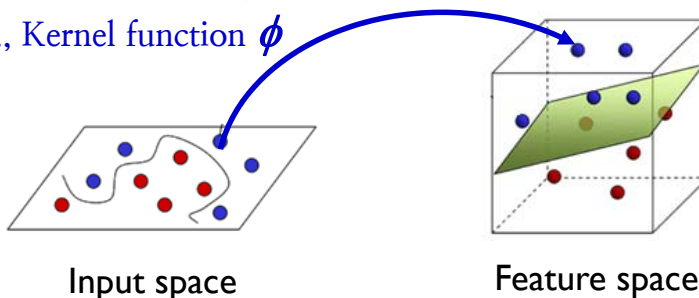
[images from google]

39

## Feature

- 머신러닝에서 판별하고자 하는 데이터에 대해서 판별의 근거로 활용할 수 있는 데이터의 특징들을 **feature**라고 한다
- In machine learning and pattern recognition, a **feature** is an individual measurable property or characteristic of a phenomenon being observed
- Choosing informative, discriminating and independent features is a crucial step for effective algorithms in machine learning
- **Kernel function** which enable the data in raw representation to operate in a high-dimensional, implicit "**feature**" space

Coordinate transfer,  
i.e., Kernel function  $\phi$

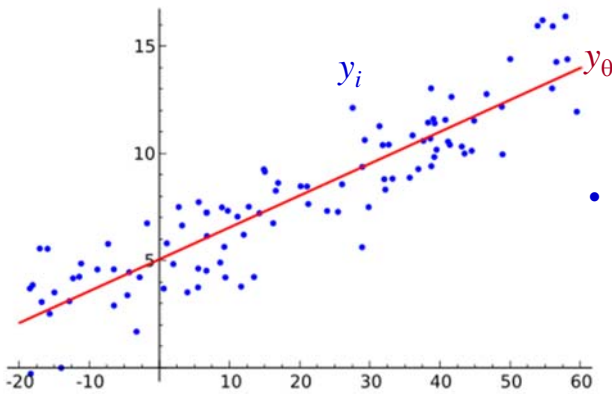


✓ Then find SVM = maximize the distance from separation lines(planes)

[images from google]

40

# Mathematical formulation of regression



- Model/feature (= regression function)

$$y_\theta(x_i) = \theta^T x_i + b$$

- Loss function (= cost function)

$$J(\theta) = \sum_{i=1} (y_\theta(x_i) - y_i)^2$$

- Mean Squared Error (MSE)

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2$$

- Optimization

$$\theta^* = \arg \min_{\theta} J(\theta)$$

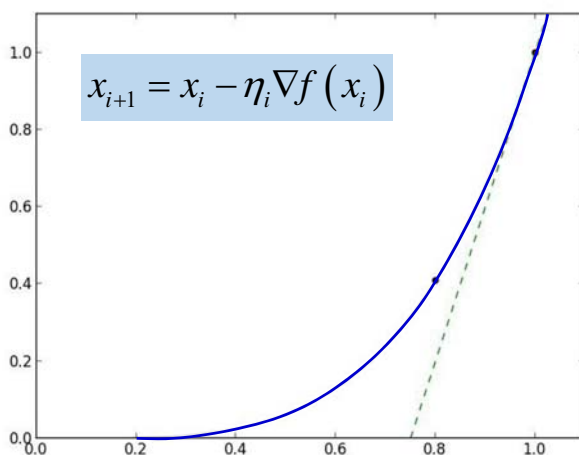
Q: How to solve this?

[images from google]

41

## Finding min MSE by Gradient Descent

- Gradient descent** is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum of a function using gradient descent, one **takes steps proportional to the negative of the gradient (or approximate gradient) of the function** at the current point iteratively. [wiki]



- 배치(batch)**: 모델 학습의 반복 1회, 즉 경사 업데이트 1회에 사용되는 예(데이터)의 집합.
- Batch: the set of examples used in one iteration (that is, one gradient update) of model training.

Q: If you have a VERY LARGE batch, even a single iteration may take a very long time to compute.

[images from google]

42

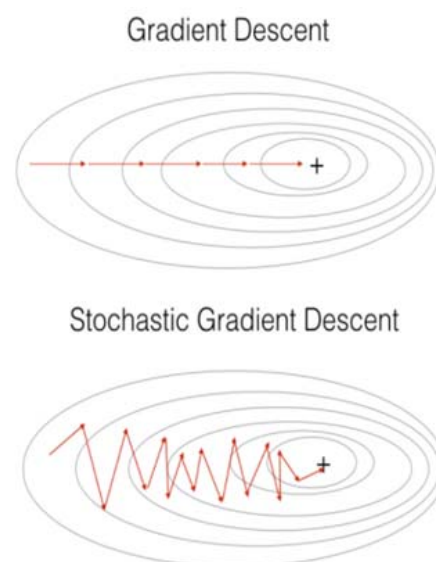
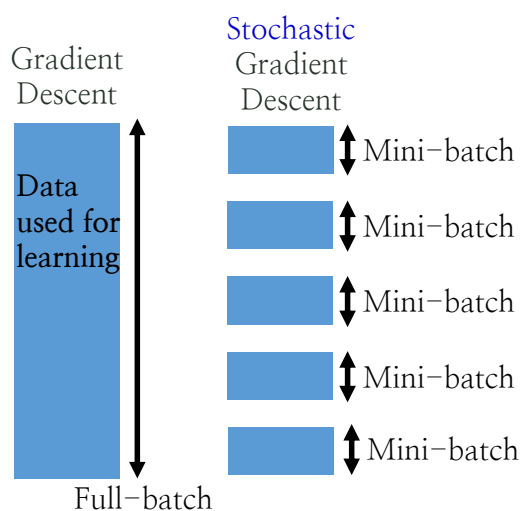
# Batch, Epoch, Iterations

- **Epoch** is an ENTIRE dataset passed forward through the neural network only ONCE
  - Since, one epoch is too big to feed to the computer at once we divide it in several smaller batches
  - Note that the optimization uses Gradient Descent which is an iterative process. So, updating the weights with single pass or one epoch is not enough
  - As the number of epochs increases, more number of times the weight are changed in the neural network and the curve goes from underfitting to optimal to overfitting curve
- **Batch Size**: total number of training examples present in a single batch
  - Larger batches = faster training vs. poor model quality for unseen data & computational power, so tune batch size
- **Iterations** is the number of batches needed to complete one epoch
  - Ex) 2000 training examples, Divide the dataset of 2000 examples into batches of 500 then it will take 4 iterations to complete 1 epoch

43

## Stochastic gradient descent (SGD)

- Stochastic Gradient Descent(SGD)
  - Use a mini-batch to calculate gradient
  - Randomly select the data set of mini-batch ('stochastic')
  - Requires many iterations to obtain reliable performance
  - Pros: Faster calculation
  - Cons: May cause nosy results



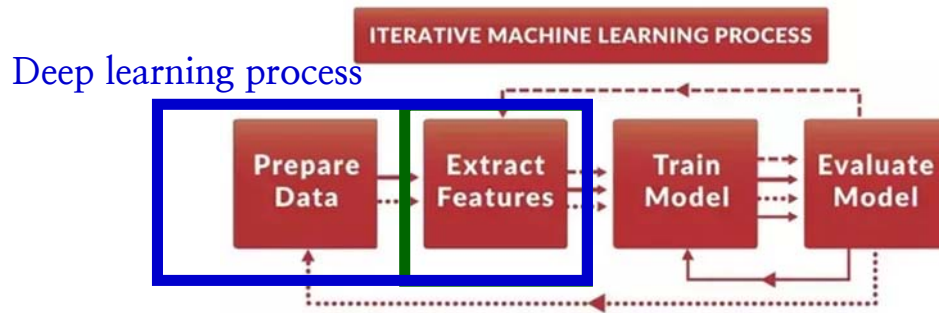
[images from google]

44

# Machine learning process

Greater number of features, or even features with non-linear characteristics will make the regression process infeasible: **Neural Net**

- 1) Select a parametric/nonparametric model (linear, kernel etc.)
- 2) Set a performance measurement (loss function)
- 3) Training data (optimizing model parameter)
- 4) Evaluate the final performance using test data

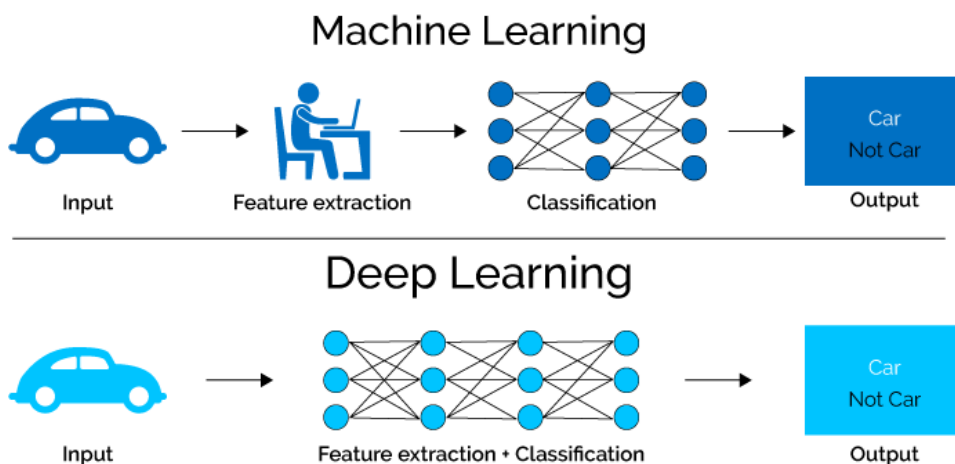


- **Deep learning** uses a cascade of **multiple layers** of nonlinear processing units for **feature extraction and transformation**.

[images from google]

45

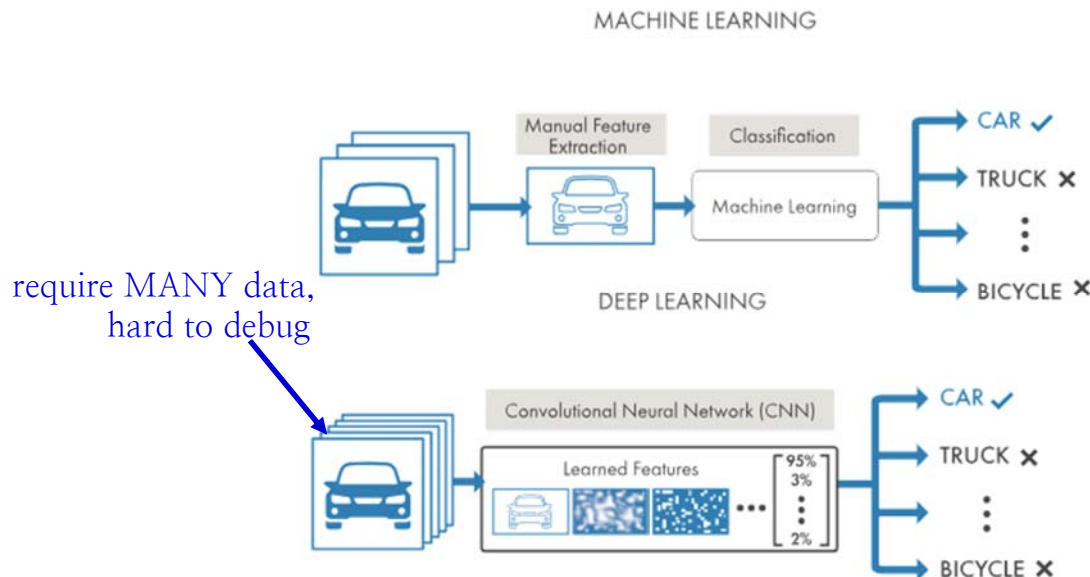
## Deep learning **Feature learning, end-to-end learning**





# Machine learning vs. deep learning

- In machine learning, you manually choose features and a classifier to sort images. With deep learning, feature extraction and modeling steps are automatic (so called, end-to-end learning).
- A key advantage of deep learning networks is that they often continue to improve as the size of your data increases.



<https://www.mathworks.com/discovery/deep-learning.html#withmatlab>

47

# Machine learning vs. deep learning

	Machine Learning	Deep Learning
Training dataset	Small	Large
Choose your own features	Yes	No
# of classifiers available	Many	Few
Training time	Short	Long



# In class review

RL: Reinforcement learning

AI

SVM: support vector machine

Feature

ML: Machine learning

Deep learning

Regression/clustering/classification

SGD: stochastic gradient descent

Kernel function

Supervised/unsupervised learning

NN: Neural net

Loss function

49

# Programming ML

- Regression
- Basic PyTorch programming

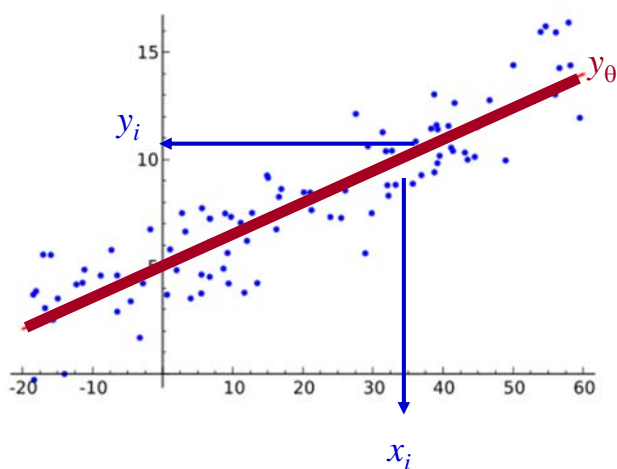
50

# 1.1 Neural Network

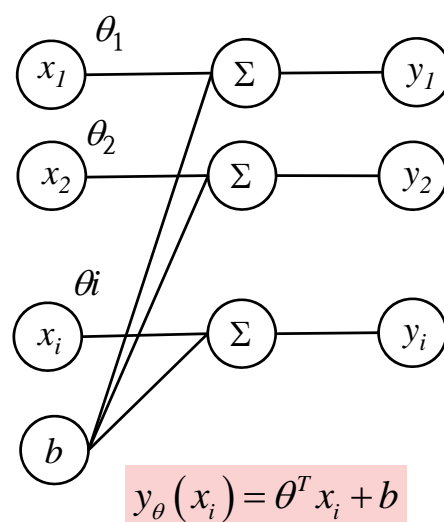
51

## Machine learning

- Try your own definition



- Loss function (= cost function)
- Mean Squared Error (MSE)



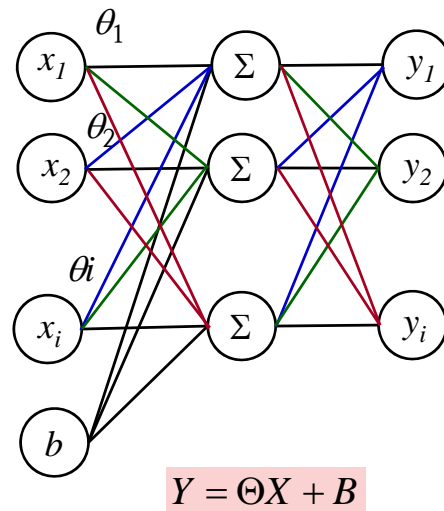
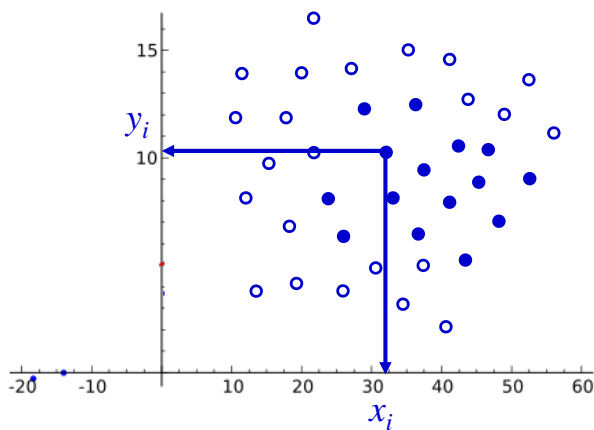
$$y_\theta(x_i) = \theta^T x_i + b$$

$$J(\theta) = \sum_{i=1} (y_\theta(x_i) - y_i)^2$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2$$

52

# Machine learning $\ni$ Neural network



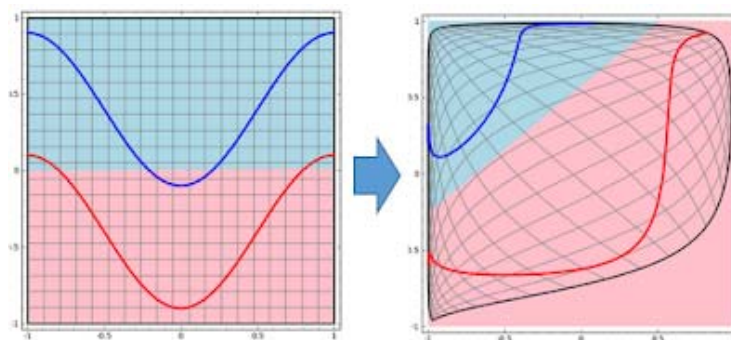
- Loss function (= cost function)
- Mean Squared Error (MSE)

$$J(\theta) = \sum_{i=1} (y_{\theta}(x_i) - y_i)^2$$
$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2$$

53

## Neural Network

- 신경망이란 데이터를 잘 구분하기 위해 데이터 공간들을 잘 왜곡해 (e.g. kernels) 선들을 긋고 (e.g. SVM) 선형 맞춤 (linear fitting)과 비선형 변환 (nonlinear transformation or activation)을 반복하여 구분결과가 더 잘 나오게 하는 과정(optimization)을 포함하는 구조라고 할 수 있다
- Neural networks are structures that are repeatedly built up with linear fitting and nonlinear transformation or activation to distinguish data

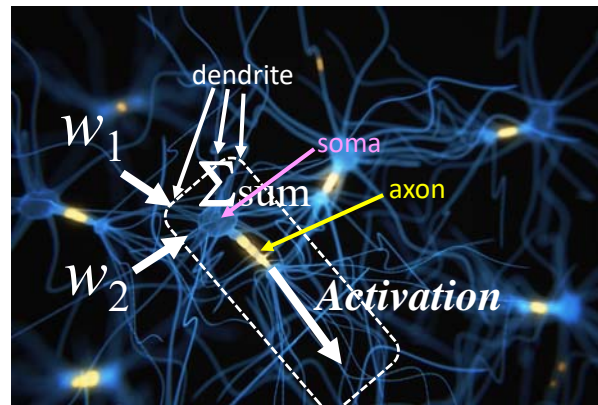


(사진출처: colah's blog)

- 파란선과 빨간선의 영역을 직선으로 구분한다고 하면 불완전한 구분이 되므로 (왼쪽), 공간을 왜곡하면 오른쪽과 같이 직선으로 구분가능하다
- 이처럼 인공신경망은 선 긋고, 구기고, 합하고를 반복하여 데이터를 처리한다

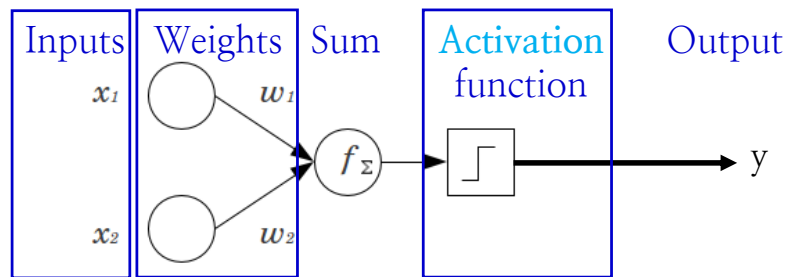
54

# Perceptron: a basic unit of neural network



## Activation function

- Step function
- Sigmoid(=logistic) function
- ReLU
- Softmax

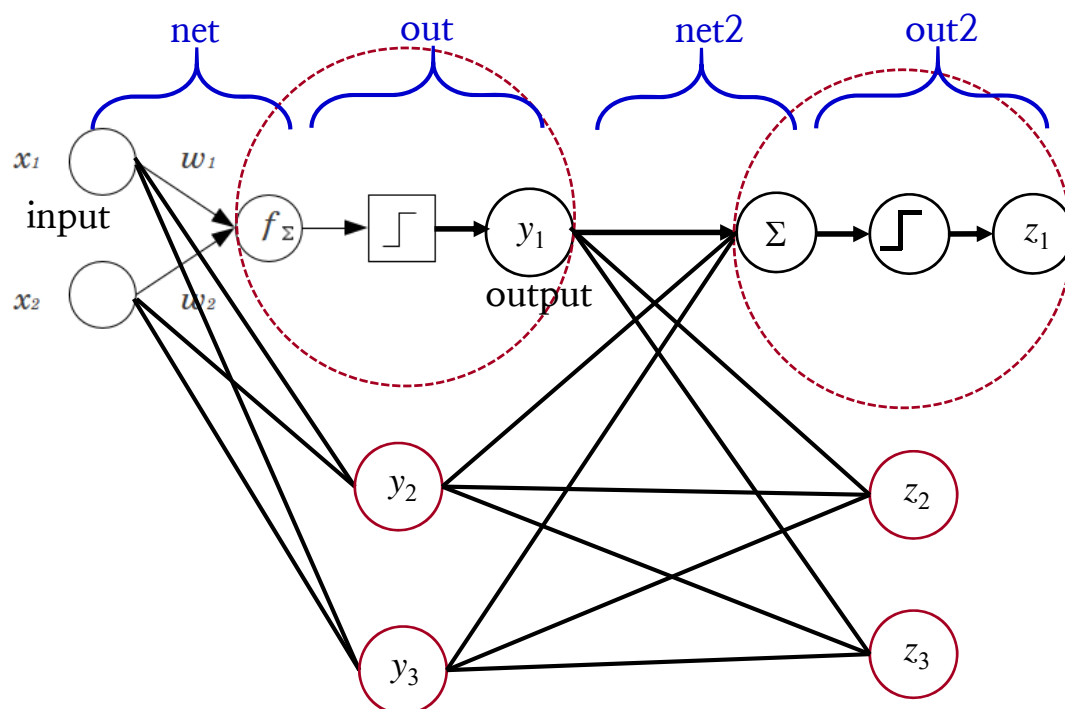


node So an artificial neuron simply calculates a "weighted sum" of its input and then decides whether it should be "fired" or not (= activation)

[images from google]

55

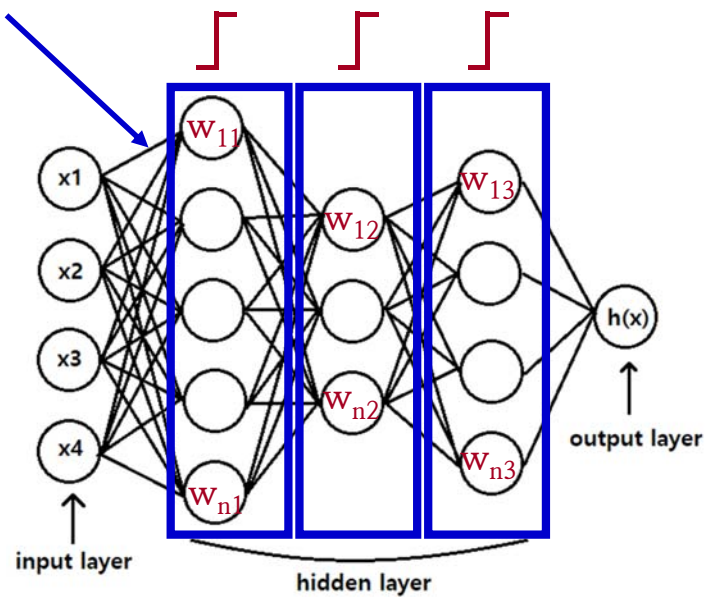
# Multilayer perceptron



56

# Multilayer perceptron

Fully connected



Q: how to find  $w$ 's?

A: Tune ' $w$ ' to minimize the error btw. the true outputs and the estimated outputs  $\equiv$  loss function

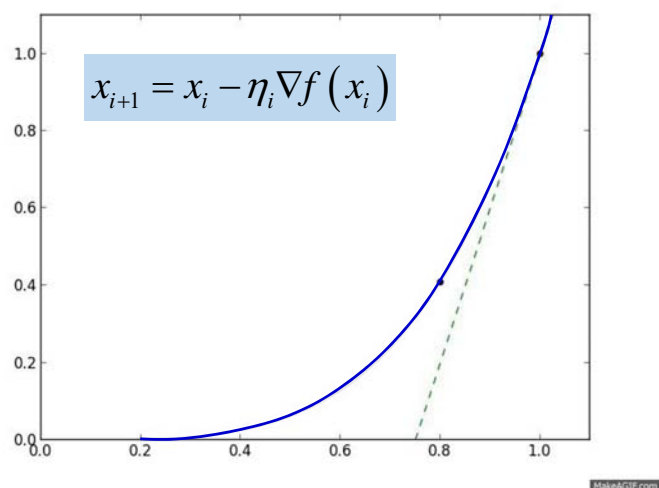
[images from google]

57

## Tune weight $w$ to minimize the loss function

- Recall, Gradient descent

Find  $x^*$  such that  $\min f(x^*)$



- Backpropagation

$$w' = w - \eta \nabla f(w)$$

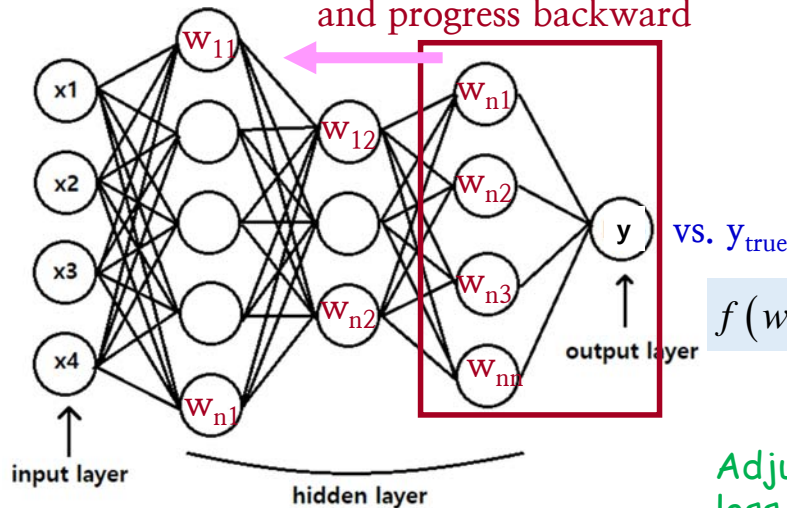
Q: What is the loss function  $f(w)$ ?

A: Error between the NN output & true values

58

# Backpropagation

➤ To tune 'w', starting from the outer most output, and progress backward



$$f(w) = f(y_{true} - y_{NN})^2$$

$$= f(y_{true} - Wx)^2$$

Adjust "w" to minimize loss function f

$$w' = w - \eta \nabla f(w)$$

Q: What is the loss function f(w)?

A: Error between the NN output & true values

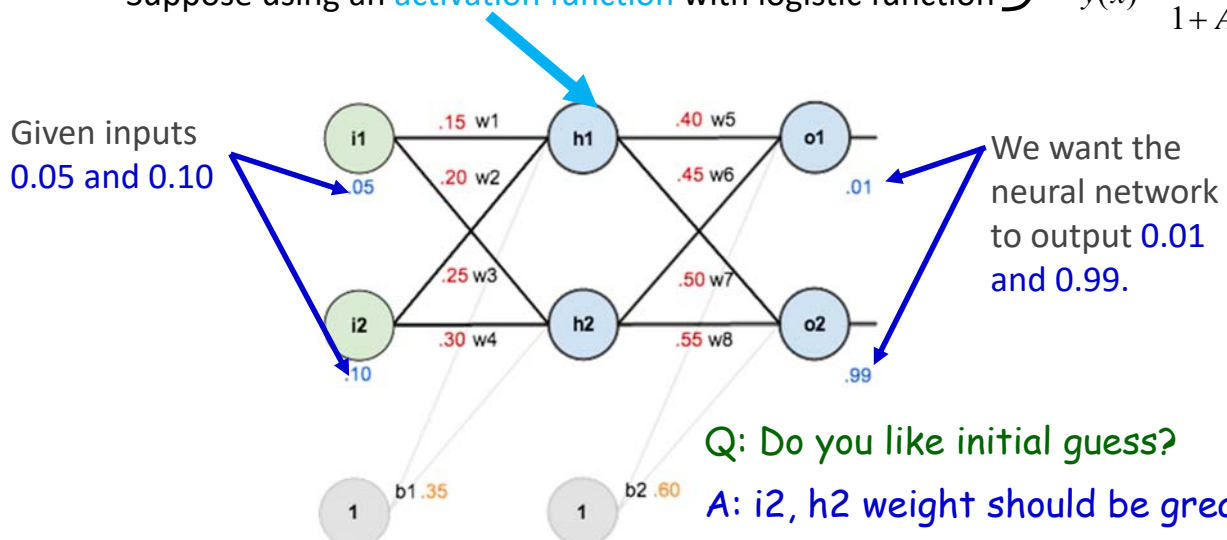
59

# Backpropagation

- Backpropagation is a method used to **calculate a gradient of the loss function** to **adjust the weights** of nodes in the network.

Ex) Here are the **initial weights**, the **biases**, and **training inputs/outputs**

Suppose using an **activation function** with logistic function  $y(x) = \frac{1}{1 + Ae^{Bx}}$

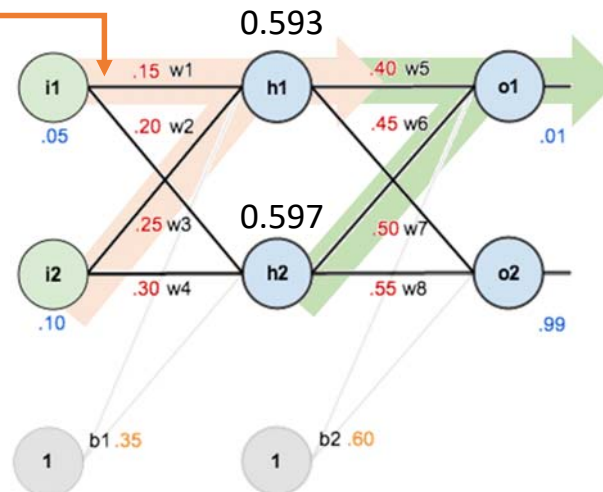


# Backpropagation

- Here's how we calculate the total **net** input for h1:

$$\text{net}_{h1} = w1 * i1 + w2 * i2 + b1 * 1 = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

- Using **logistic function** to get the output of h1:  $\text{out}_{h1} = \frac{1}{1 + e^{-\text{net}_{h1}}} = \frac{1}{1 + e^{-0.3775}} = 0.593$
- Carrying out the same process for h2 we get:  $\text{out}_{h2} = 0.597$



Repeat this process for the output layer neurons, using the output from the hidden layer neurons as inputs

+ (plus)

Activation function

<https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>

61

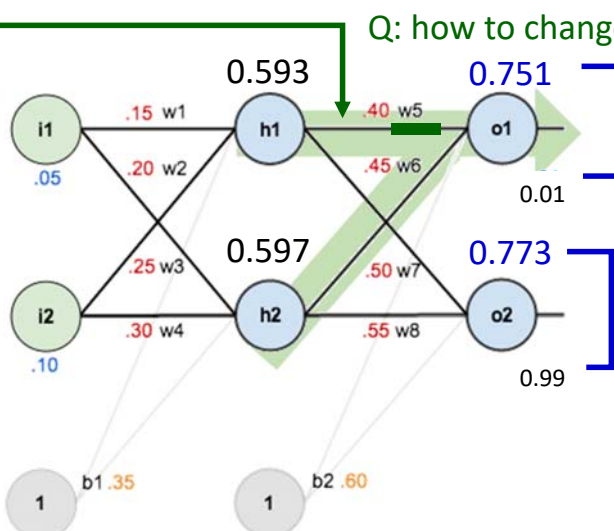
# Backpropagation

- Here's the output for o1:

$$\text{net}_{o1} = w5 * \text{out}_{h1} + w6 * \text{out}_{h2} + b2 * 1 = 0.4 * 0.593 + 0.45 * 0.597 + 0.6 * 1 = 1.106$$

$$\text{out}_{o1} = \frac{1}{1 + e^{-\text{net}_{o1}}} = \frac{1}{1 + e^{-1.106}} = 0.751$$

- Similarly,  $\text{out}_{o2} = 0.773$



- Q: how to change w5? • Then calculate the loss function

$$E_{total} = \sum \frac{1}{2} (\text{target} - \text{output})^2$$

$$E_{o1} = \frac{1}{2} (0.01 - \underline{0.751})^2 = 0.275$$

Q: what affects to  $\text{out}_{o1}$ ?

$$E_{o2} = 0.024$$

$$E_{total} = E_{o1} + E_{o2} = \underline{0.298}$$

Q: Which output component contributes to error more?

62



# Backward pass

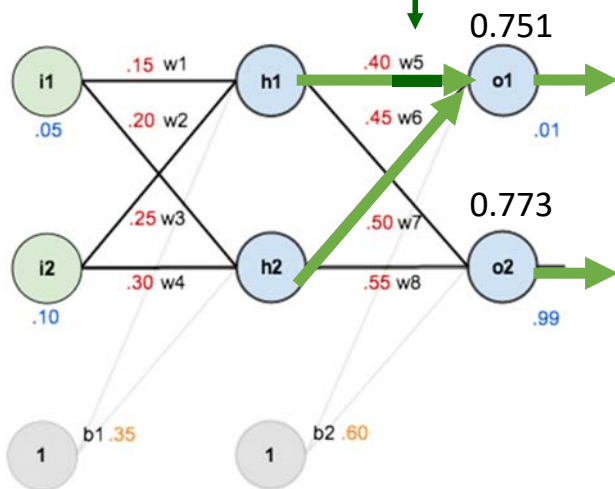
Q: We want to know how much a change in  $w_5$  affects the total error?

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o_1}} \cdot \frac{\partial out_{o_1}}{\partial net_{o_1}} \cdot \frac{\partial net_{o_1}}{\partial w_5}$$

$w_5$  affects to  $E_{total}$  through  $o_1$  by network + activation function

$$\frac{\partial E_{total}}{\partial out_{o_1}} = -(0.01 - o_1) = -(0.01 - 0.751) = 0.741$$

$$\frac{\partial out_{o_1}}{\partial net_{o_1}} = \frac{\partial}{\partial net_{o_1}} \left( \frac{1}{1 + e^{-net_{o_1}}} \right) = 0.187$$



$$= \left( 1 - \frac{1}{1 + e^{-net_{o_1}}} \right) \left( \frac{1}{1 + e^{-net_{o_1}}} \right) = (1 - 0.751) 0.751$$

$E_{total}$

$$E_{o_1} = \frac{1}{2} (0.01 - 0.751)^2 = 0.275$$

$$E_{o_2} = 0.024$$

$$E_{total} = E_{o_1} + E_{o_2} = 0.298$$

63

# Backward pass

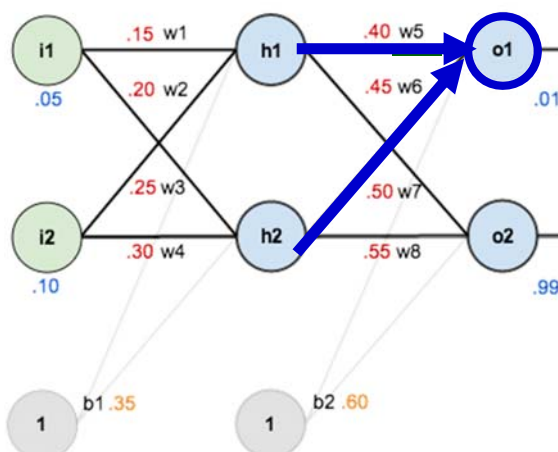
- Finally, how much does the total net input of  $o_1$  change with respect to  $w_5$ ?

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o_1}} \cdot \frac{\partial out_{o_1}}{\partial net_{o_1}} \cdot \frac{\partial net_{o_1}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial out_{o_1}} = -(0.01 - o_1) = -(0.01 - 0.751) = 0.741$$

$$\frac{\partial out_{o_1}}{\partial net_{o_1}} = \frac{\partial}{\partial net_{o_1}} \left( \frac{1}{1 + e^{-net_{o_1}}} \right) = 0.187$$

Recall,  $net_{o_1} = w_5 * out_{h_1} + w_6 * out_{h_2} + b_2 * 1$



$$\frac{\partial net_{o_1}}{\partial w_5} = out_{h_1} = 0.593$$

- Putting all together

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o_1}} \cdot \frac{\partial out_{o_1}}{\partial net_{o_1}} \cdot \frac{\partial net_{o_1}}{\partial w_5}$$

$$= 0.741 \cdot 0.187 \cdot 0.593 = 0.082$$

64



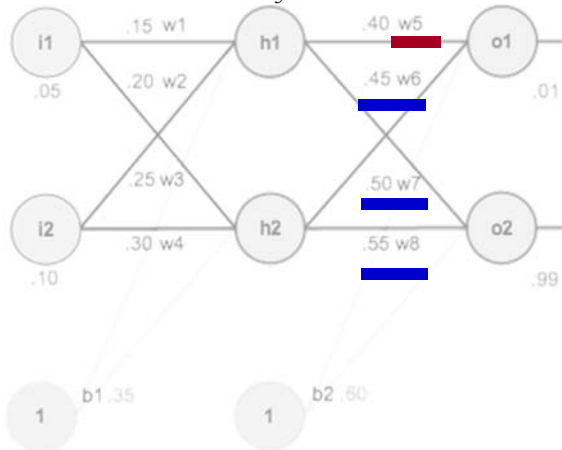
# Backward pass

- Update weight:** to decrease the error, we then subtract this value from the current weight (optionally multiplied by some learning rate, eta, which we'll set to 0.5):

$$w_5^+ = w_5 - \eta \cdot \frac{\partial E_{total}}{\partial w_5} = 0.4 - 0.5 \cdot 0.082 = 0.359$$

Reduce weight w5 to reduce error  
(= loss function)

$$\frac{\partial E_{total}}{\partial w_5} = 0.082$$



We can repeat this process to get the new weights w6, w7, and w8:

$$w_6^+ = 0.409$$

$$w_7^+ = 0.511$$

$$w_8^+ = 0.561$$

<https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>

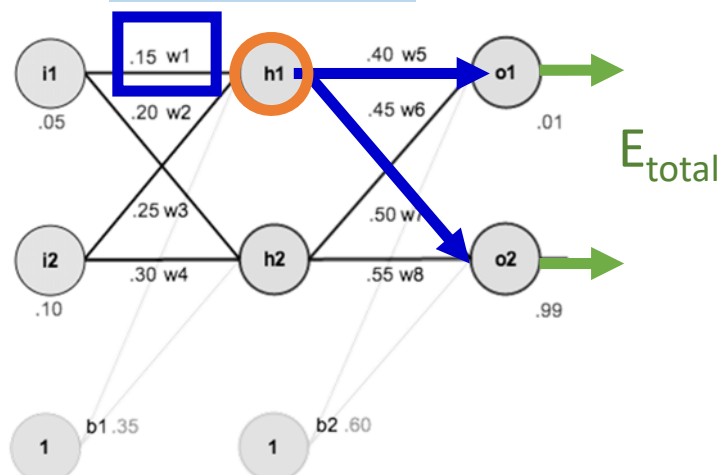
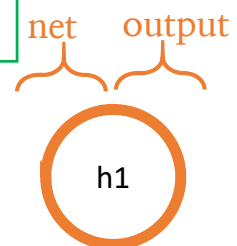
65

# Hidden layer

- Next, we'll continue the backwards pass by calculating new values for w1, w2, w3, w4.

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h_1}} \cdot \frac{\partial out_{h_1}}{\partial net_{h_1}} \cdot \frac{\partial net_{h_1}}{\partial w_1} \Rightarrow \frac{\partial E_{total}}{\partial w_1} = 0.00044$$

$$\frac{\partial E_{total}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial out_{h_1}} \cdot \frac{\partial E_{o_2}}{\partial out_{h_1}}$$



66

# Hidden layer

- We can now update all of our weights  $w_1, w_2, w_3, w_4$ .

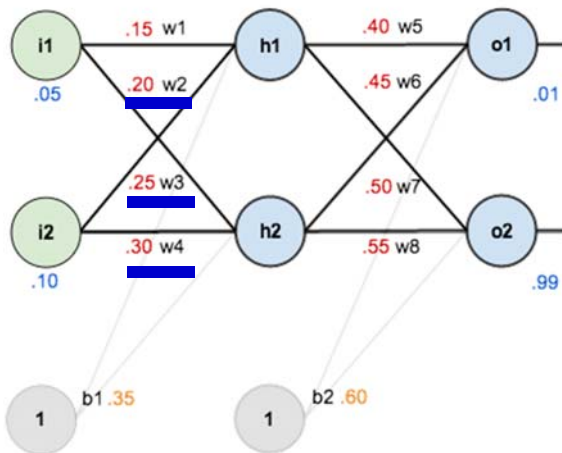
$$w_1^+ = w_1 - \eta \cdot \frac{\partial E_{total}}{\partial w_1} = 0.15 - 0.5 \cdot 0.00044 = 0.1497$$

$$w_2^+ = 0.1996$$

$$w_3^+ = 0.2498$$

$$w_4^+ = 0.2995$$

Reduce weight  $w_1$  (very slightly) to reduce error (= loss function)



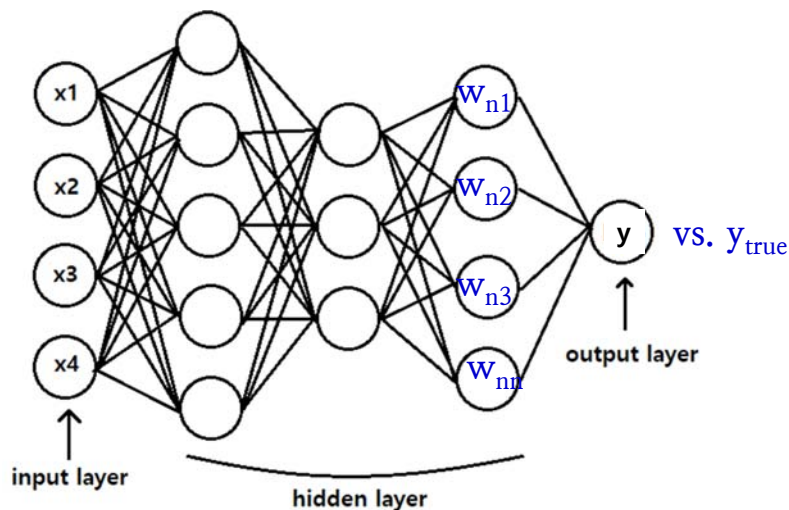
- Original network error  $E_{total} = 0.298$
- After the first round of backpropagation,  $E_{total} = 0.291$

It seems to be very small change, but after repeating this  $10^4$  times, for example, the error plummets to 0.0000351085.

<https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>

67

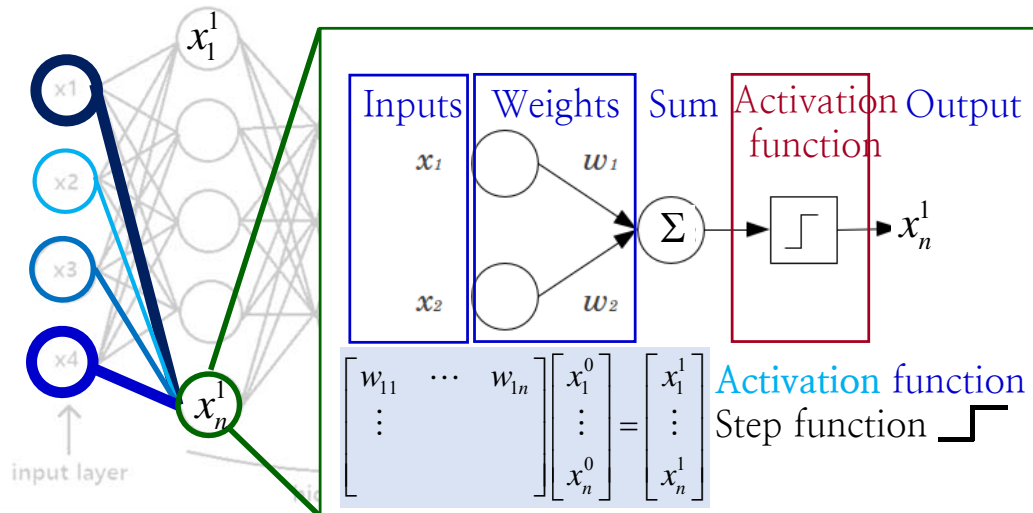
# Recall, Backpropagation



$$w' = w - \eta \nabla f(w)$$

- Derivative of output errors
- Output is a weighted sum of inputs with activation function

# Activation function in NN



$$w' = w - \eta \nabla f(w)$$

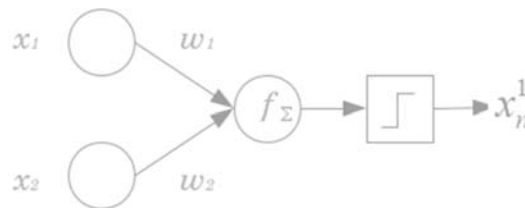
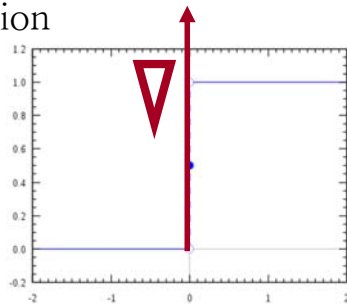
- Derivative of output errors
- Output is a **weighted sum of inputs** with **activation function**

Images from google

69

## Activation function

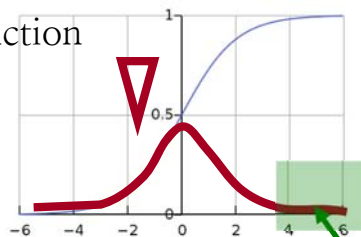
Step function



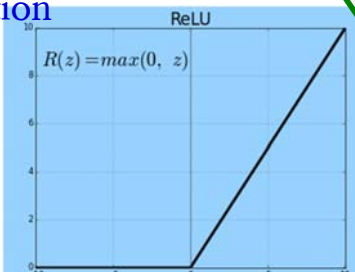
$$x_n^1 = \sigma(w_{1,1}x_1^0 + w_{1,2}x_2^0 + \dots + w_{1,n}x_n^0 + b_1)$$

Activation function, e.g. logistic

Logistic function



ReLU function



To update the weight  $w$ , take a derivative of  $f$ , i.e., activation function

$$w' = w - \eta \nabla f(w)$$

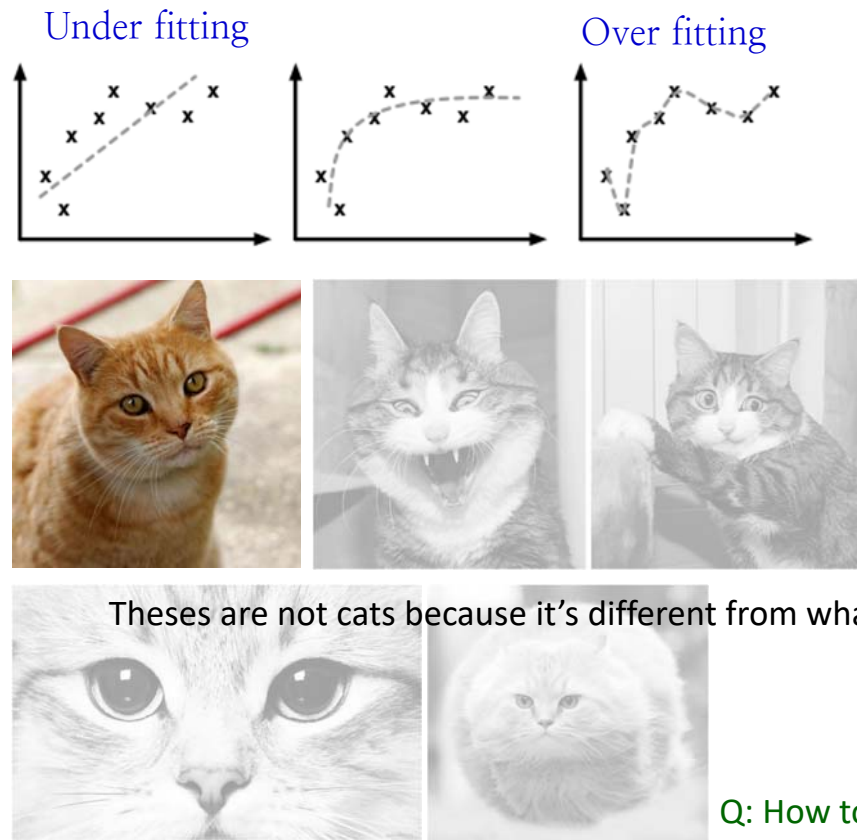
- Update of the weight ( $\sim$  proportional to the partial derivative  $\nabla$  of the error function) would not occur if the gradient will be vanishingly small. This may completely stop the neural network from further training.

$\Rightarrow$  'Vanishing gradient' problem

Images from google

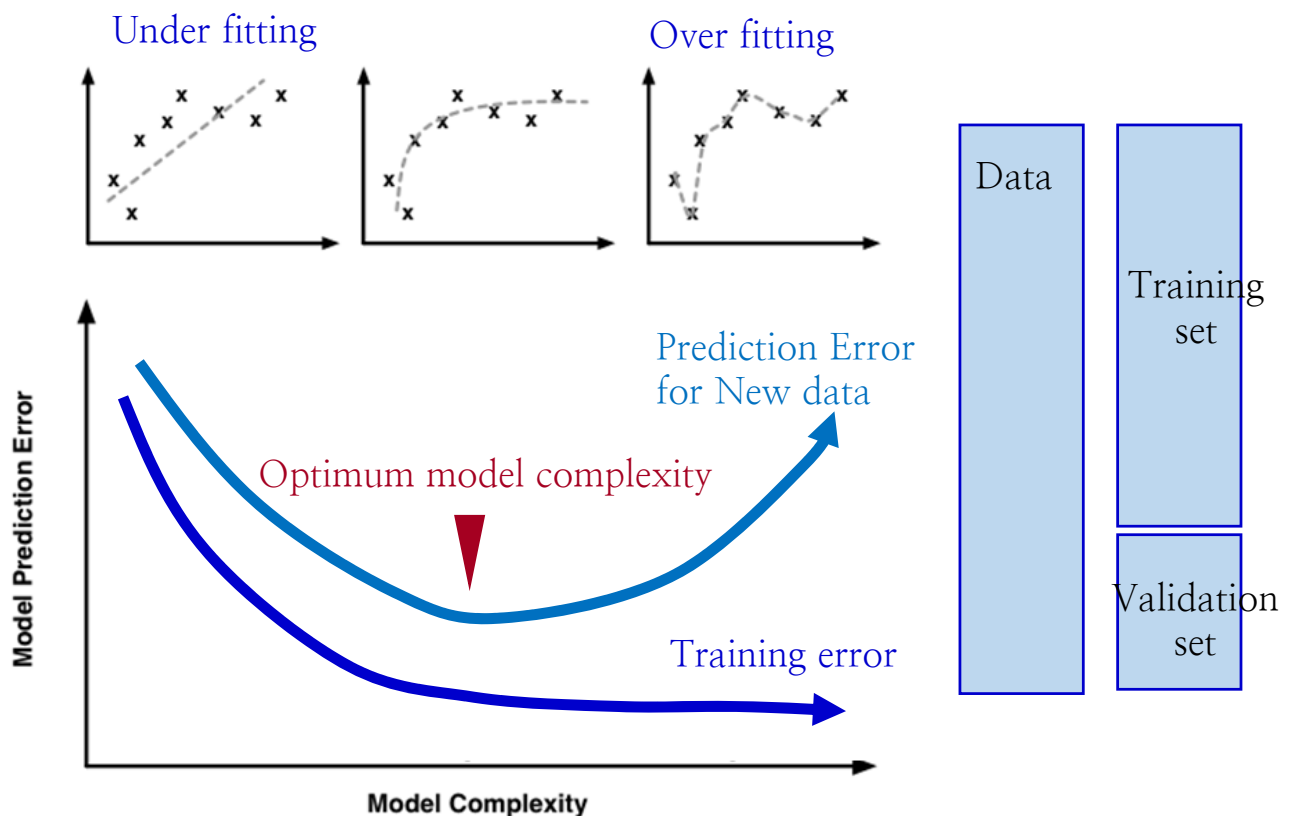
70

# Overfitting with training images



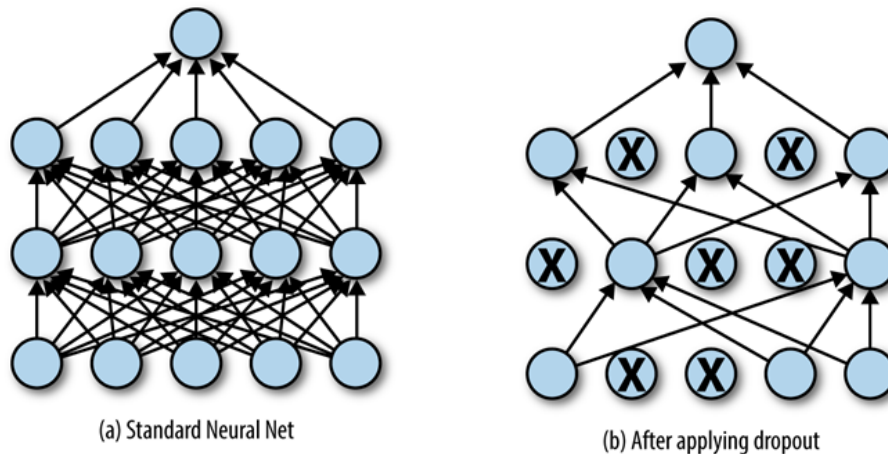
71

## Training & test (,validation)



# Dropout

- **Dropout** is a form of regularization that **randomly drops some proportion of the nodes** that feed into a fully connected layer.
- This helps prevent the net from relying on one node in the layer too much.
- Here, dropping a node means that its contribution to the corresponding activation function is set to 0. Since there is no activation contribution, the gradients for dropped nodes drop to zero as well.

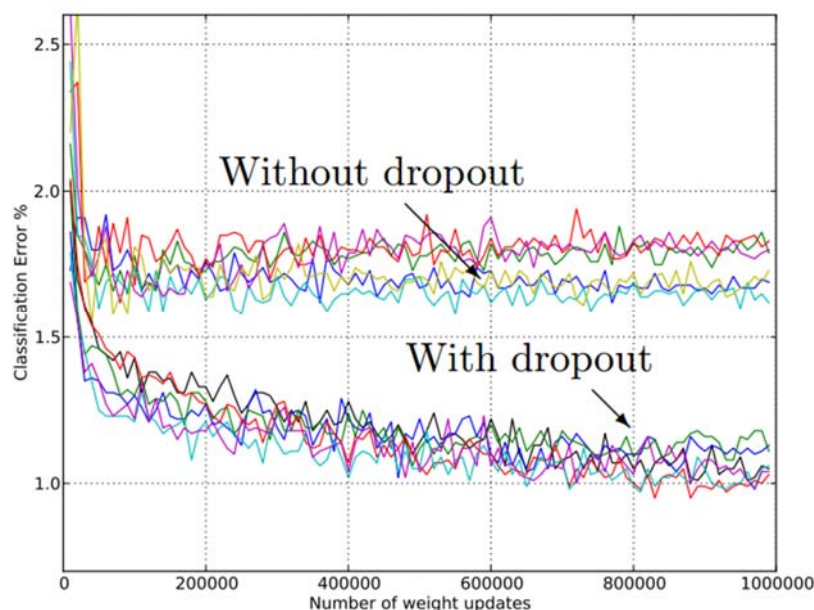


<https://www.safaribooksonline.com/library/view/tensorflow-for-deep/9781491980446/ch04.html>

73

# Dropout

- Below we have a classification error (Not including loss), observe that the test/validation error is smaller using dropout



[https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/dropout\\_layer.html](https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/dropout_layer.html)

74

# In class review

Backpropagation

Hidden layer

Neural Network (ANN)

Overfitting

Dropout

Softmax

ReLU

SGD: stochastic gradient descent

Fully connected

Vanishing gradient

Loss function

75

# Programing NN

Basic code structures & details

76

## 1.2 Convolutional Neural Network

77

### Convolution

- Convolve 감다; 감기다; 휘감다; 둘둘 말다[감다]; 빙빙 돌다
- Convolution
  - 대단히 복잡한[난해한] 것 ex) the bizarre convolutions of the story
  - (나선형의) 주름[구불구불한 것] the convolutions of the brain

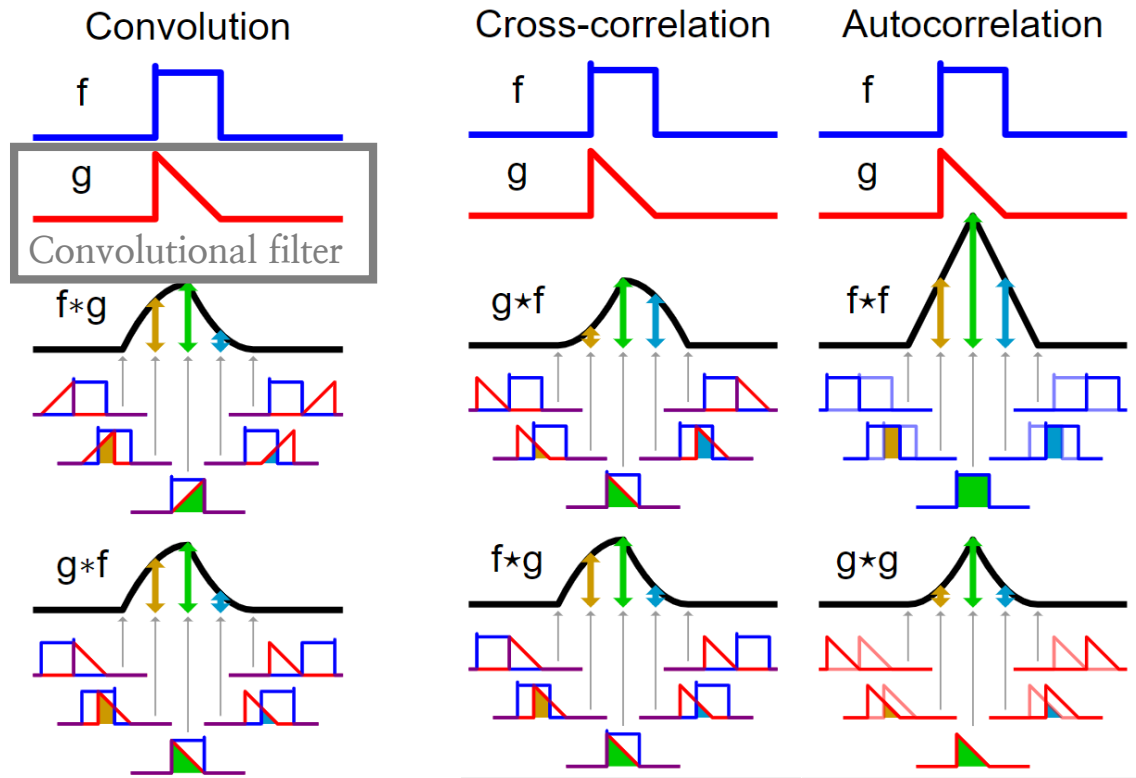
$$(f * g)(t) \equiv \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau = \int_{-\infty}^{\infty} f(t - \tau) g(\tau) d\tau$$

- 합성곱(convolution, 콘벌루션)은 하나의 함수와 또 다른 함수를 반전 이동한 값을 곱한 다음, 구간에 대해 적분하여 새로운 함수를 구하는 수학 연산자이다. a mathematical operation on two functions (f and g) to produce a third function that expresses how the shape of one is modified by the other

78



# Convolution

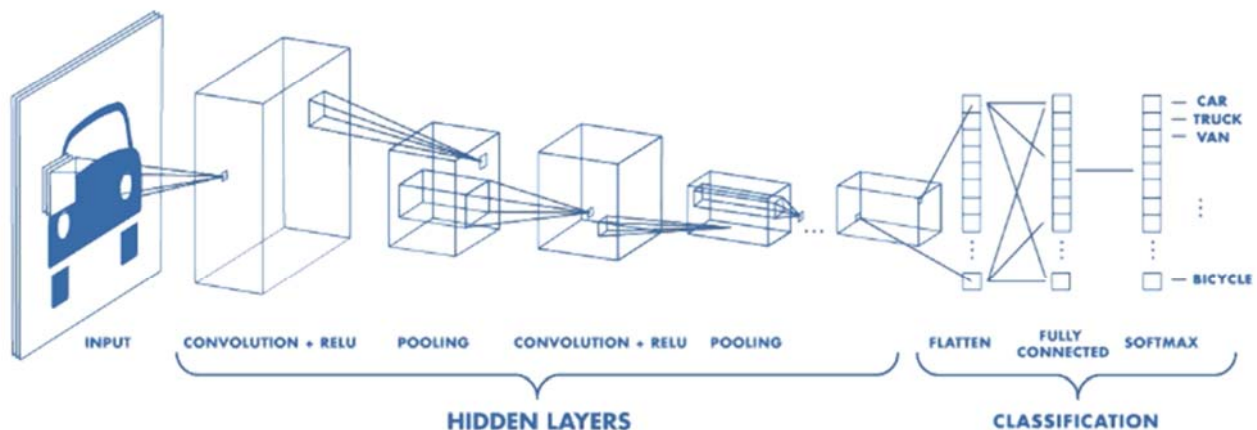


[Wikipedia]

79

## Convolutional neural network (CNN)

- 이미지 입력과 같은 경우 데이터량을 줄여 효율적으로 계산하기 위해 입력값에서 특징점(feature)을 추출하는 **필터(convolutional filter)**를 사용하여 **입력 데이터를 스캔하여 feature map (=activation map)을 생성하는 신경망**
- A neural network method that scans input data using a **convolutional filter that extracts feature points from the input values** to efficiently calculate the data amount in the case of image input.

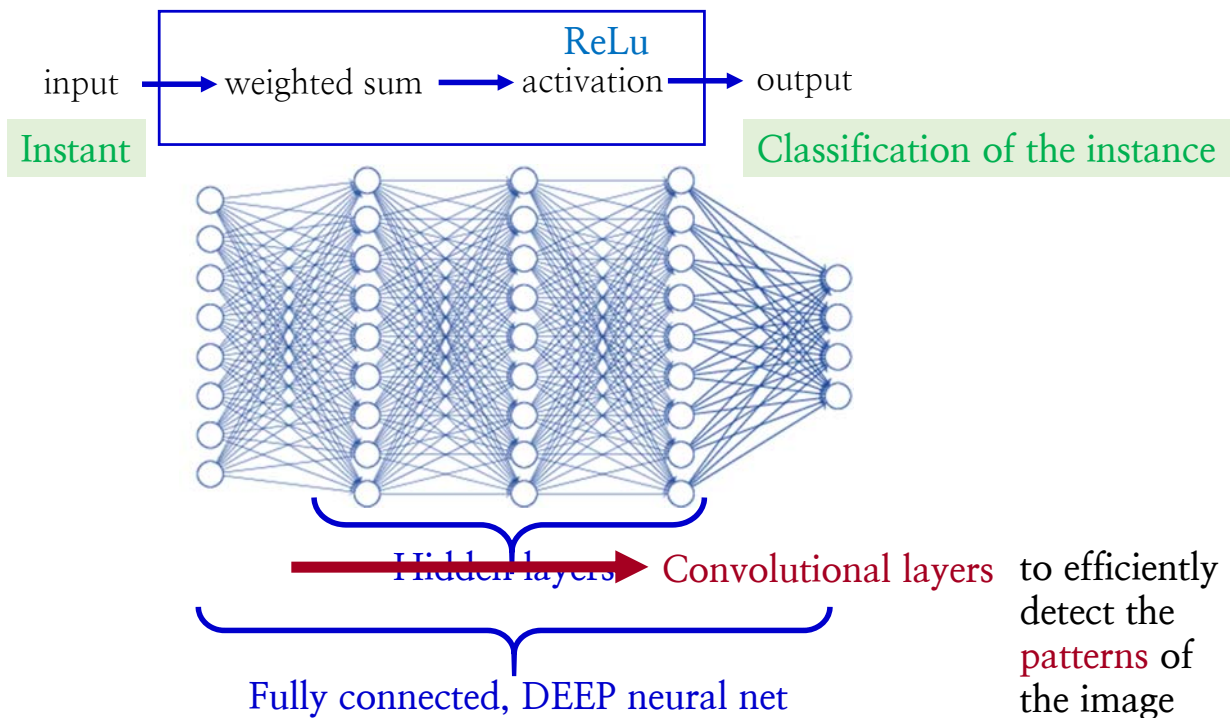


Images from google

80



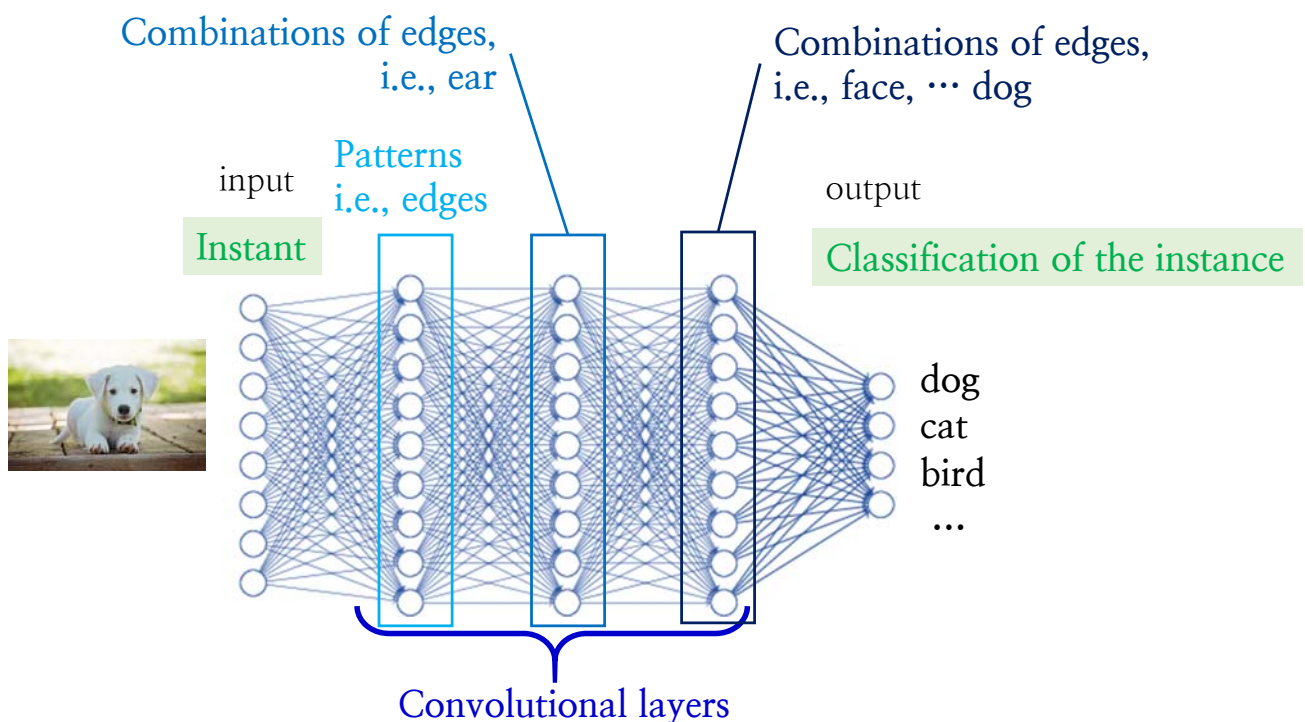
# Artificial NN, i.e., fully connected NN



Images from google

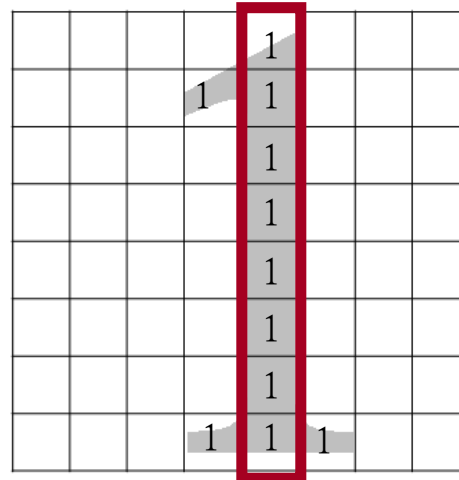
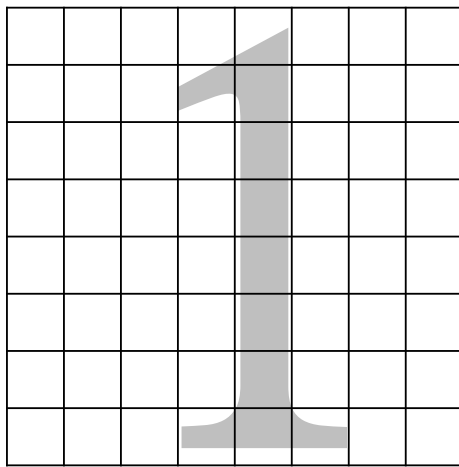
81

# Convolutional neural network



82

# Patterns of an image

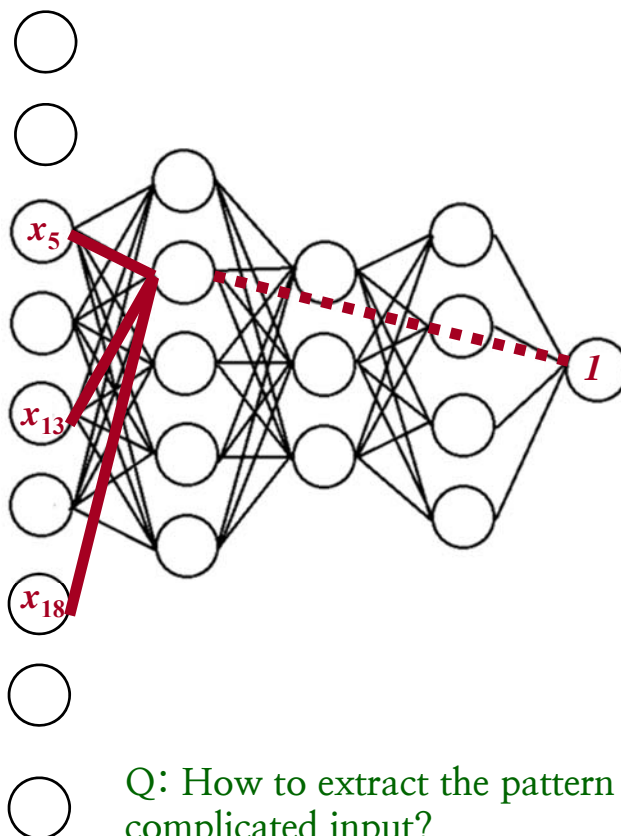
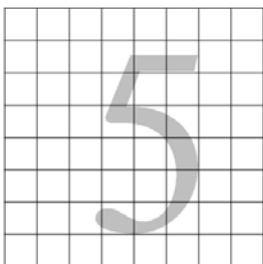


0
0
0
0
1
0
0
0
0
0
0
1
1
0
0
0
0
...

83

Patterns

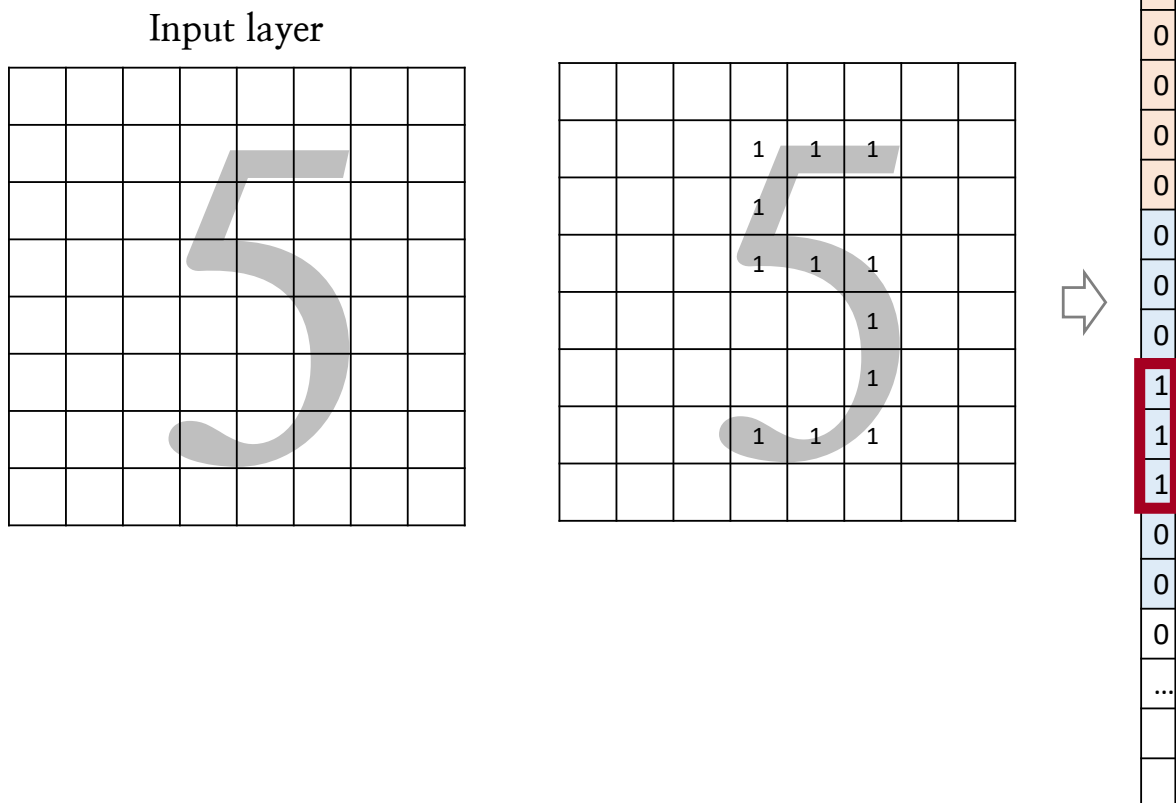
0
0
0
0
1
0
0
0
0
0
0
0
1
1
0
0
0
0
...



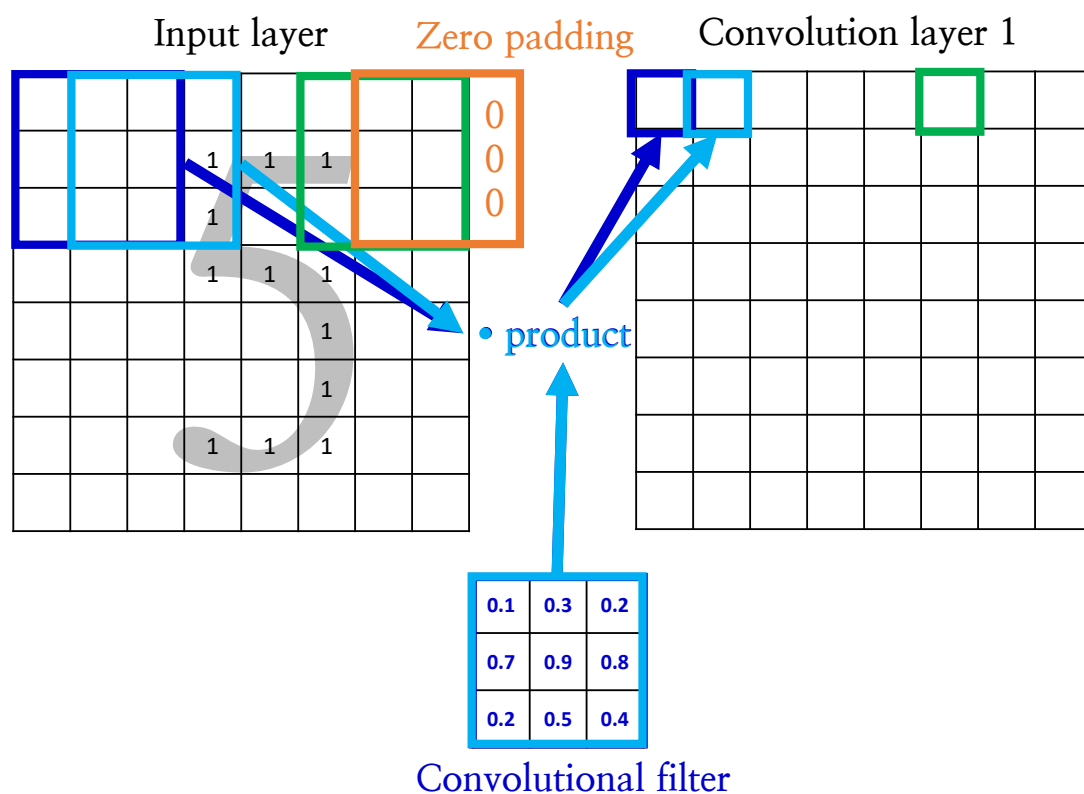
Q: How to extract the pattern of complicated input?

84

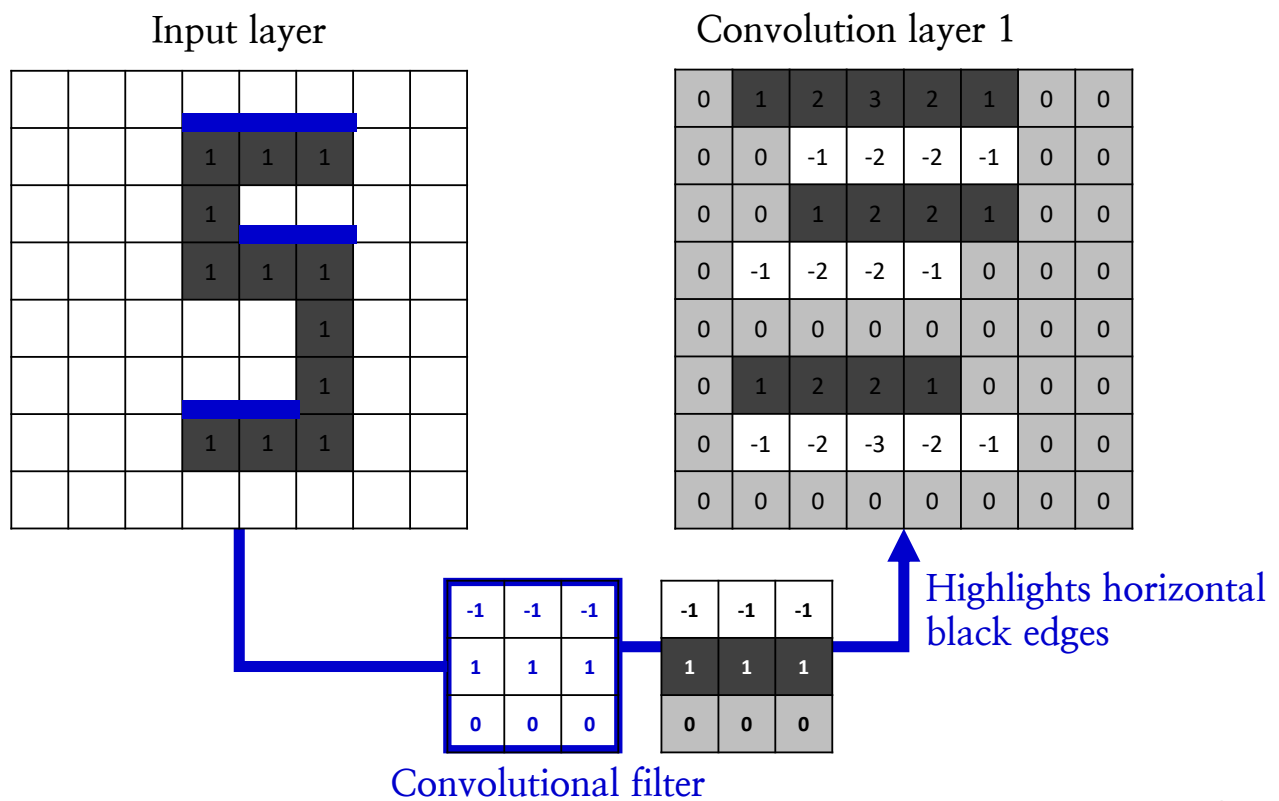
# Patterns of an image



# Convolution filter

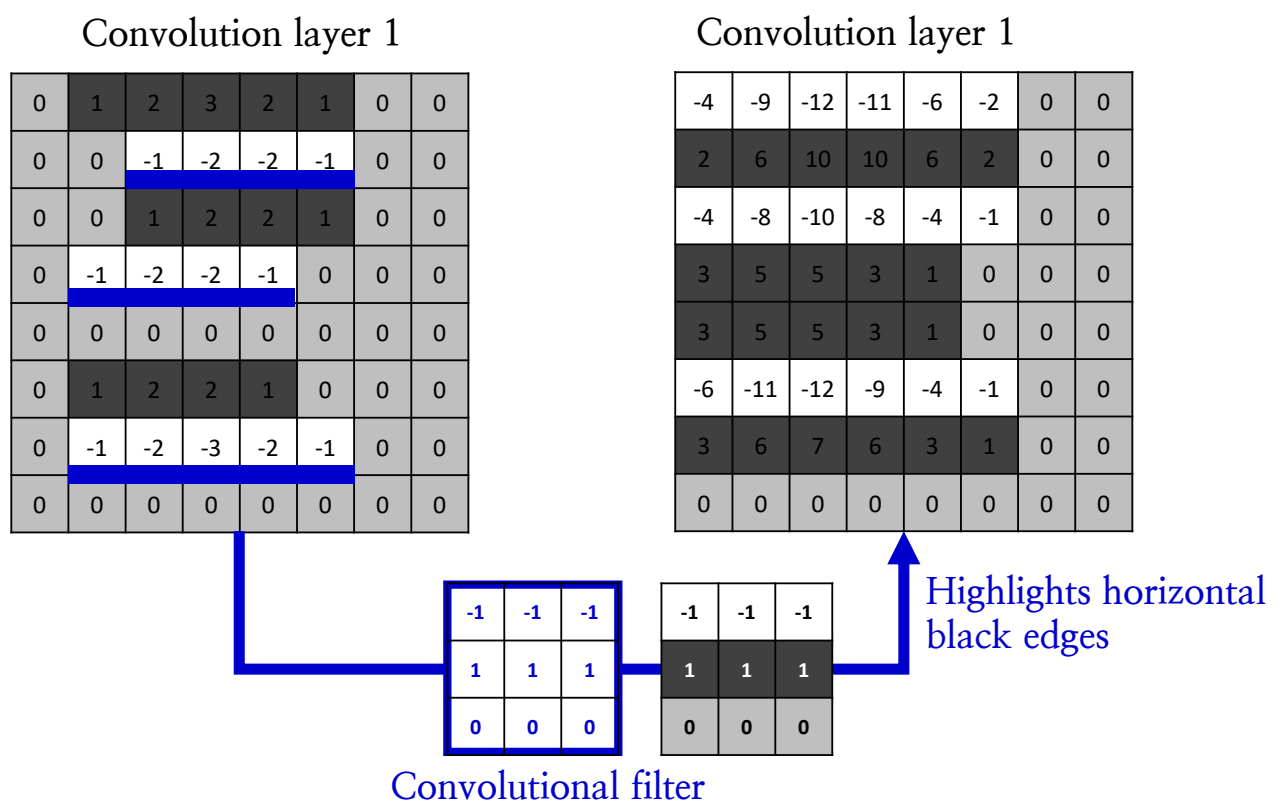


## Ex) Convolution filter detecting edges



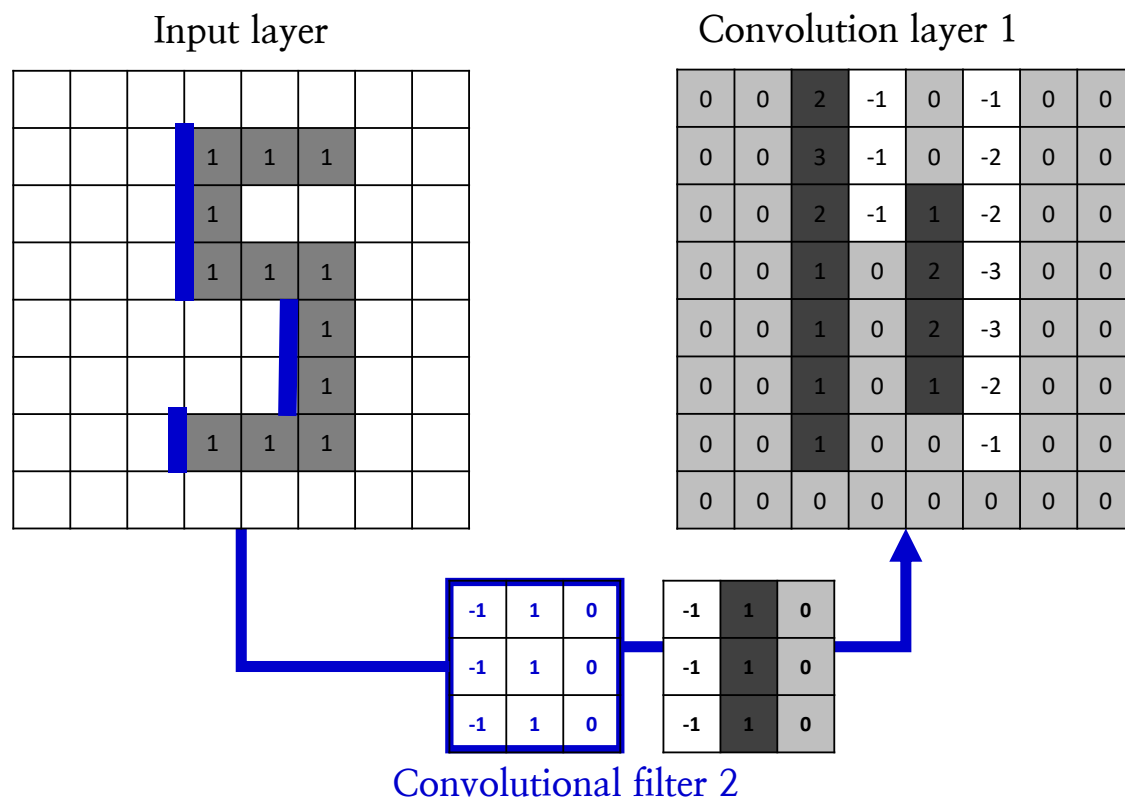
87

## Ex) Convolution filter detecting edges



88

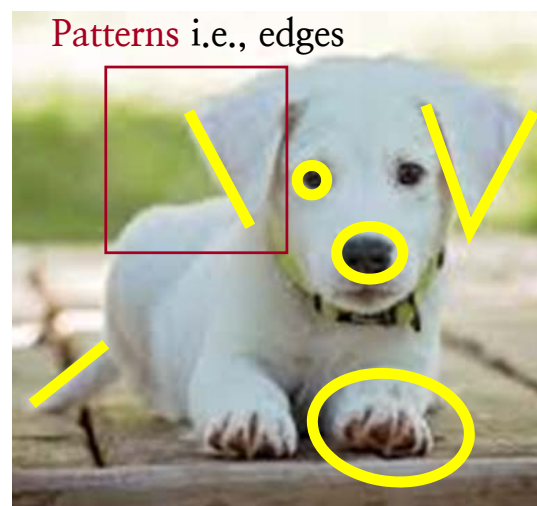
## Ex) Convolution filter detecting edges



89

## Patterns of the image

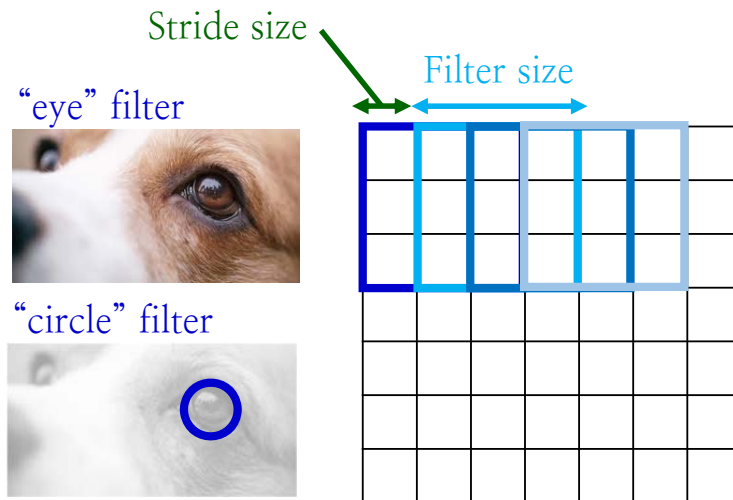
Apply multiple convolutional filters to extract feature map like edges



90

# Convolution filter

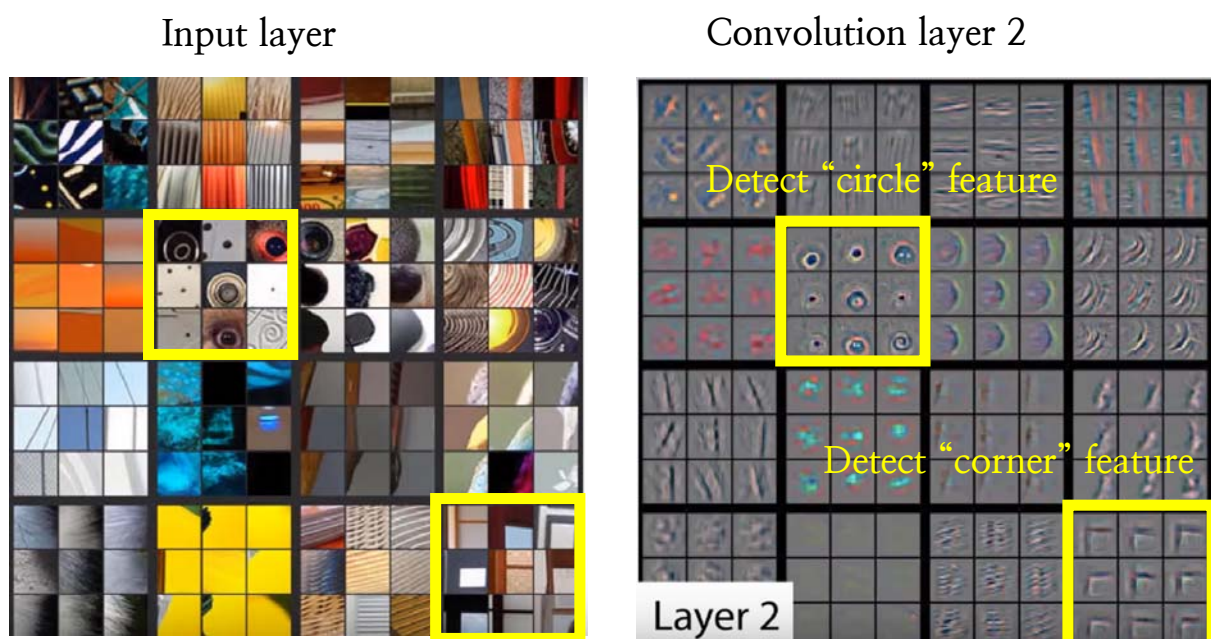
- 입력(이미지)으로 부터 여러가지 특징(예-테두리)을 추출하는 다중의 컨볼루션 필터를 적용
- Apply multiple convolutional filters to extract feature map like edges



Ex) 위의 이미지를 파란 사각형 윈도우 (컨볼루션필터)로 rolling 하며 추출  
Suppose you detect the image by viewing through a blue window

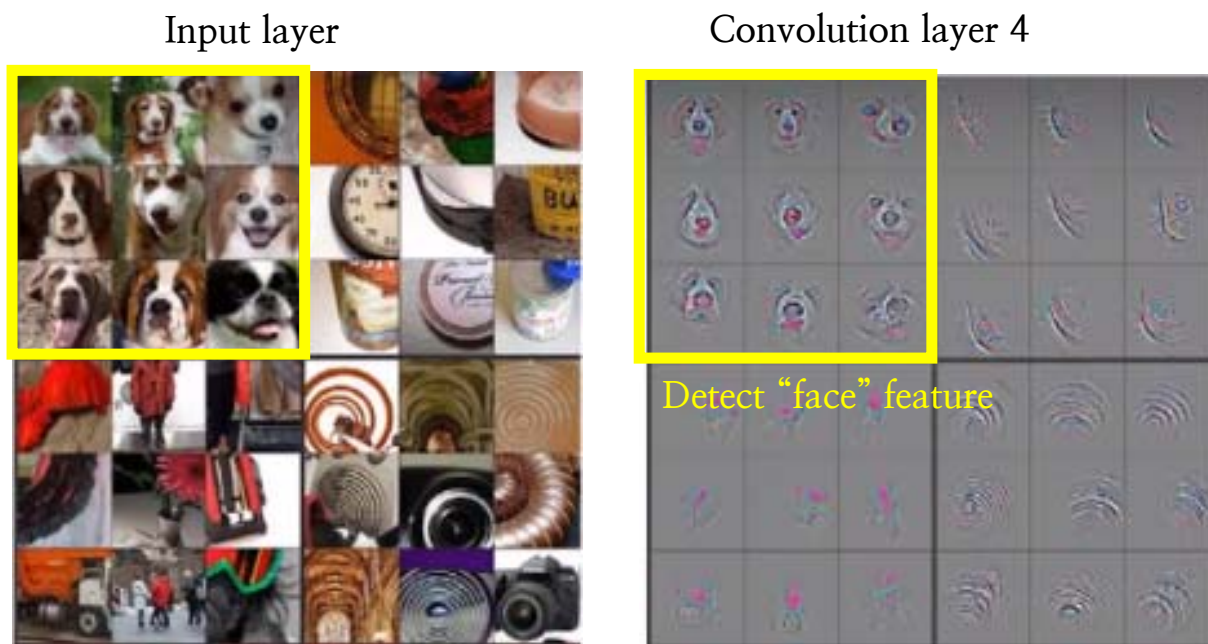
91

## Ex) Convolution filter detecting features





## Ex) Convolution filter detecting features

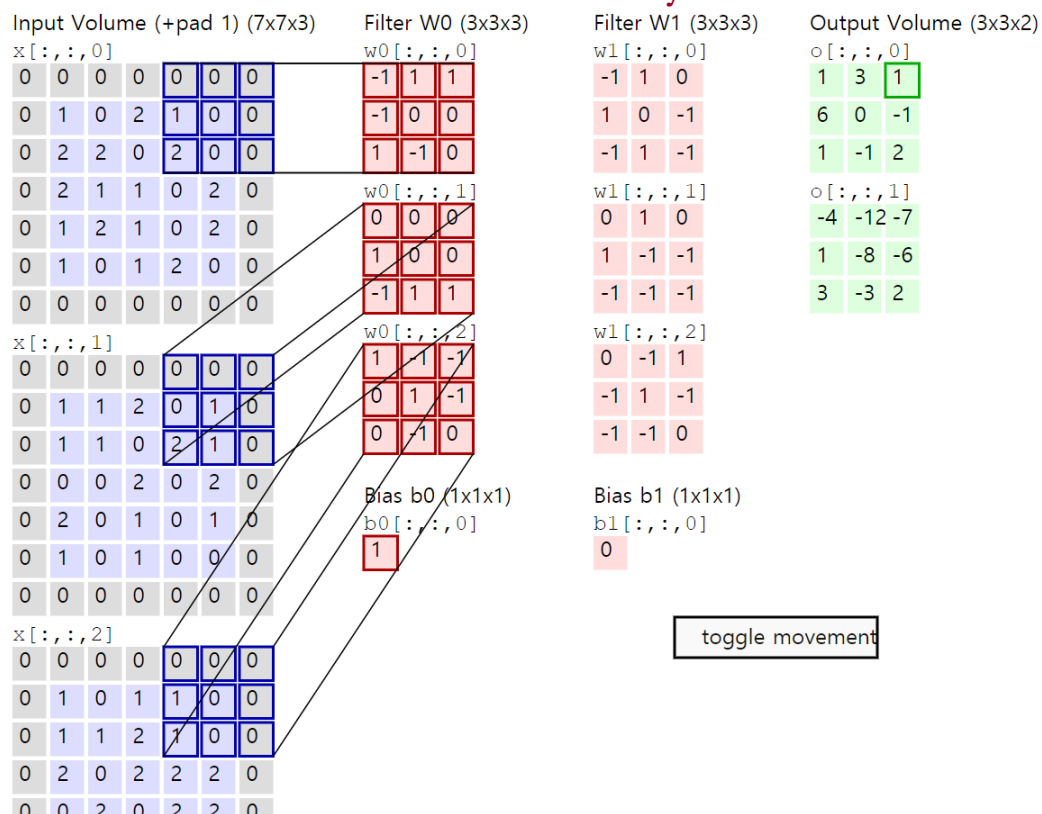


[https://www.youtube.com/playlist?list=PLZbbT5o\\_s2xq7LwI2y8\\_QtvuXZedL6tQU](https://www.youtube.com/playlist?list=PLZbbT5o_s2xq7LwI2y8_QtvuXZedL6tQU)

93

## Ex) Convolution with layered input

### Two convolution layers

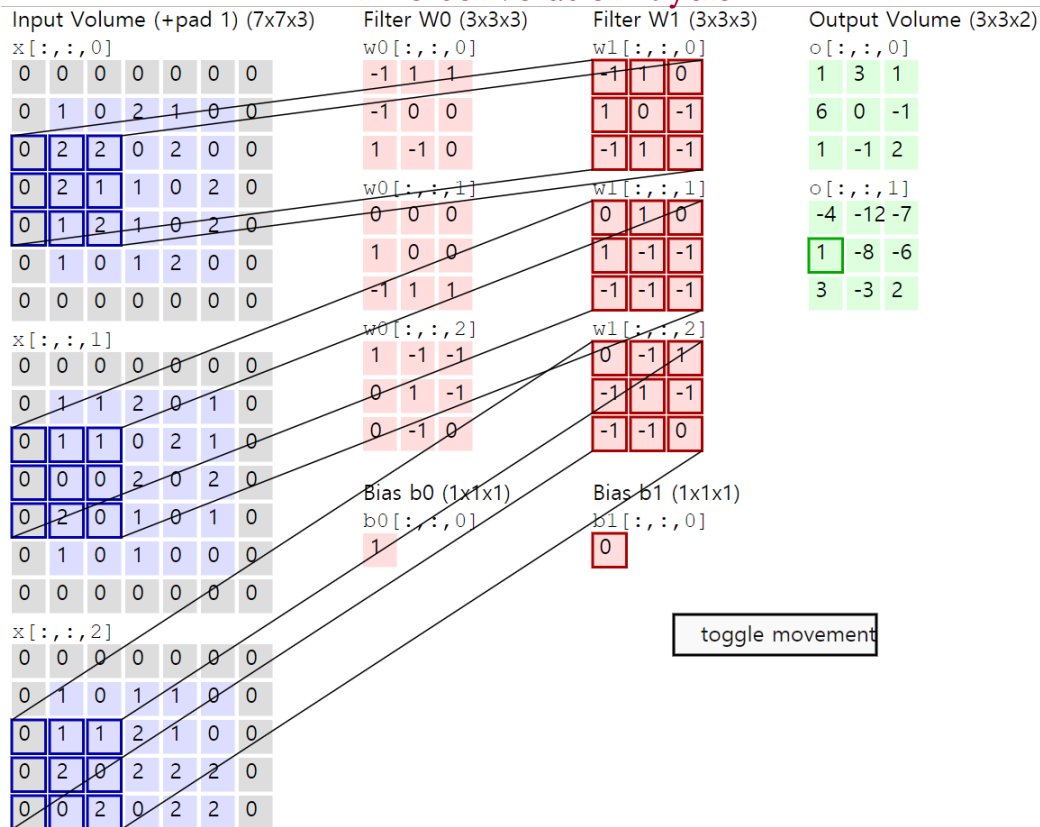


<http://cs231n.github.io/convolutional-networks/>

94

# Ex) Convolution with layered input

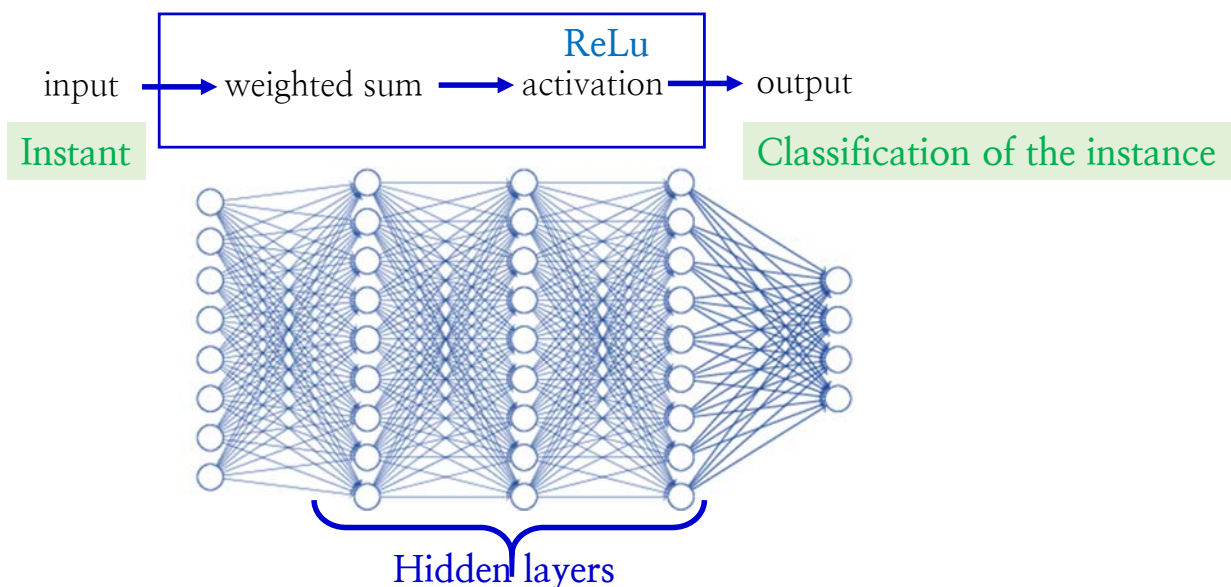
## Two convolution layers



<http://cs231n.github.io/convolutional-networks/>

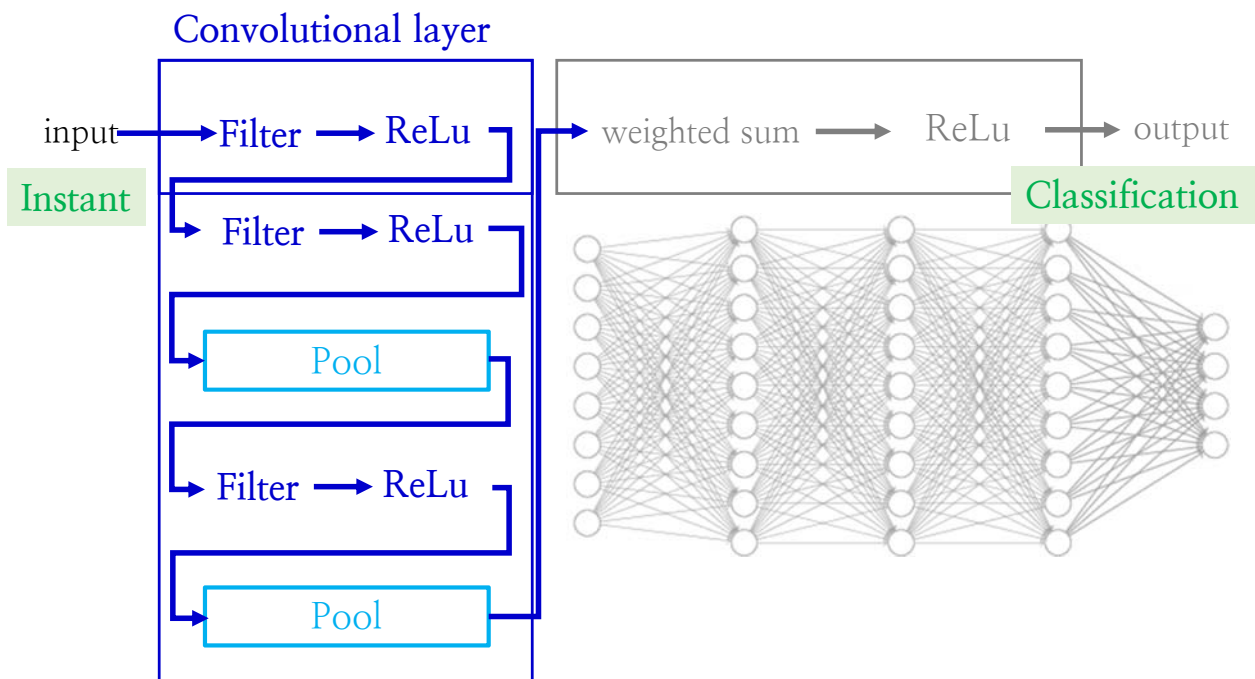
95

# Recall, Artificial NN



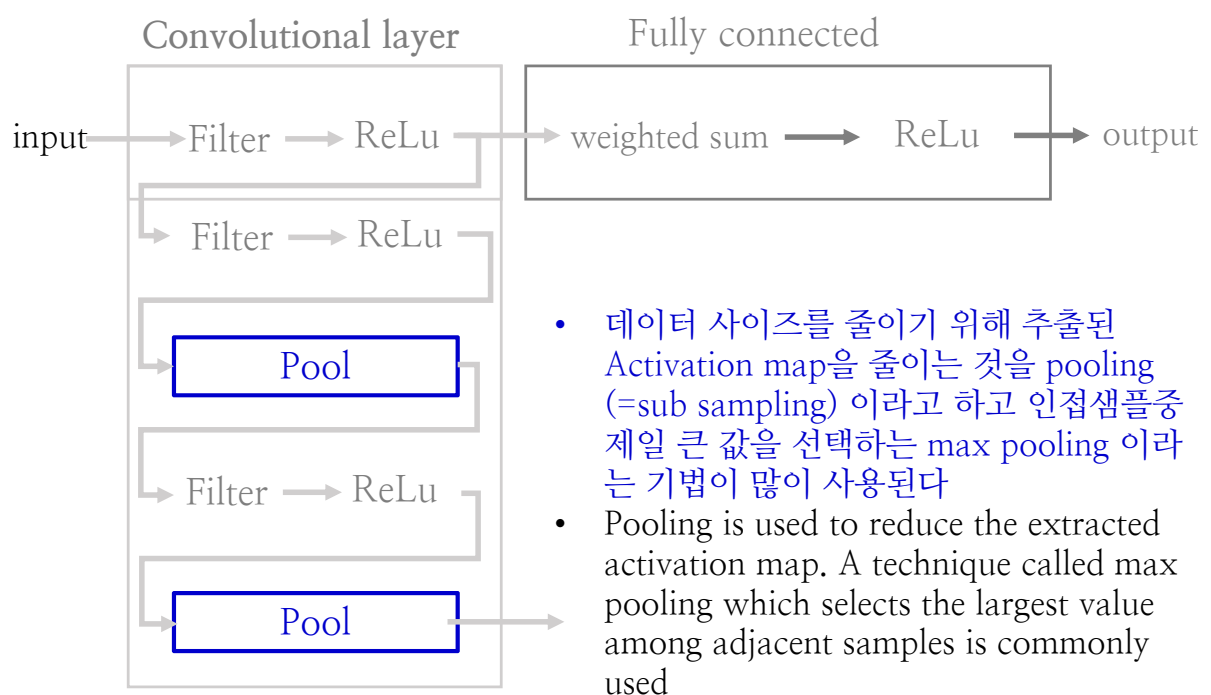


# Convolutional neural network



97

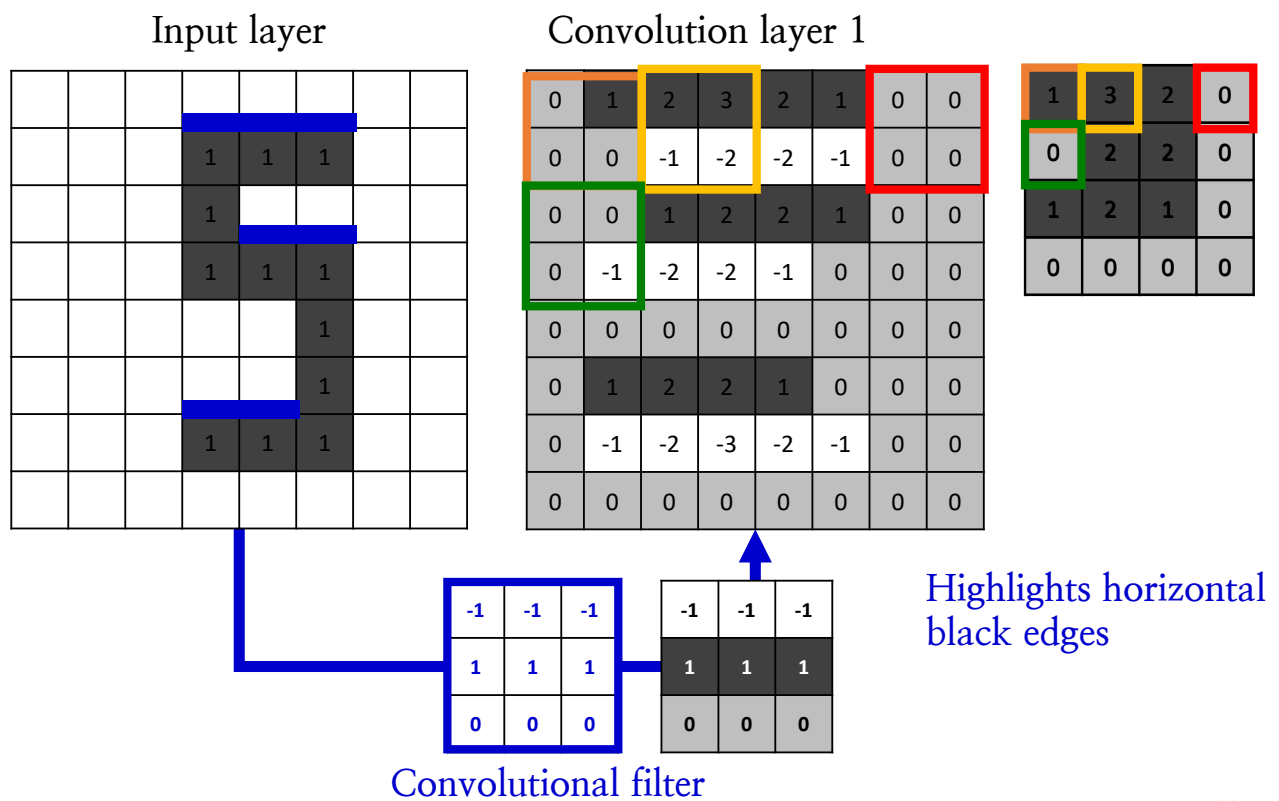
# Convolutional neural network



98

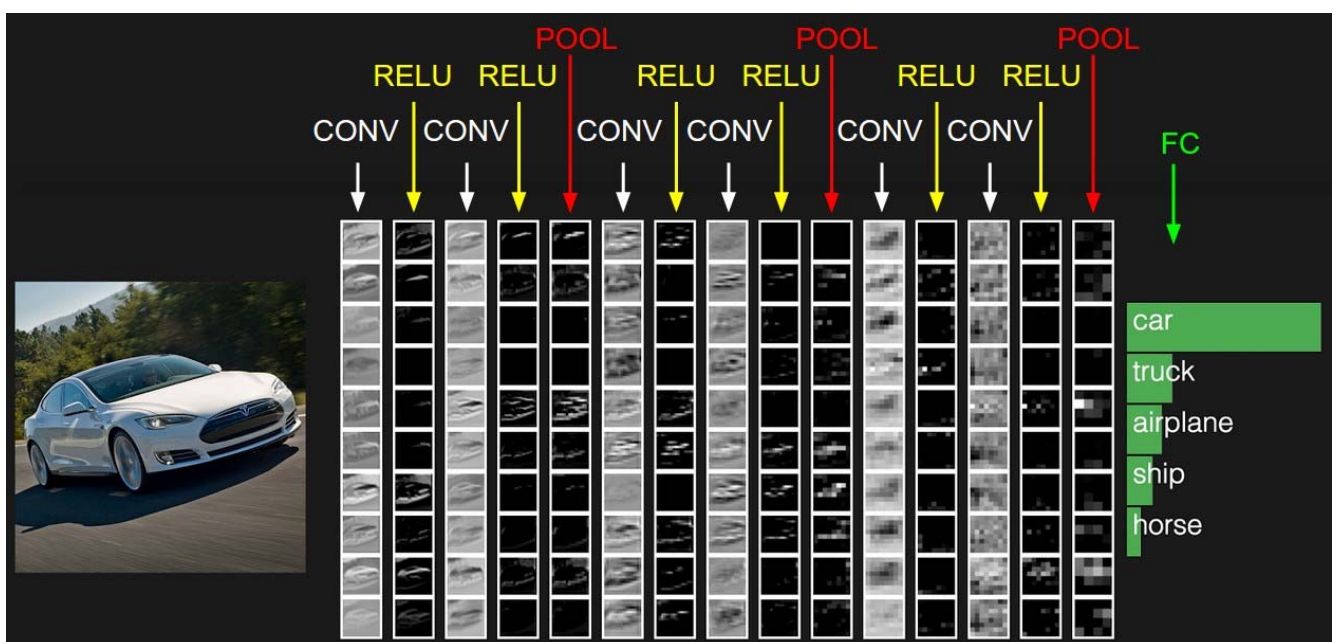
# Max pooling example

Filter size = 2  
Stride size = 2



99

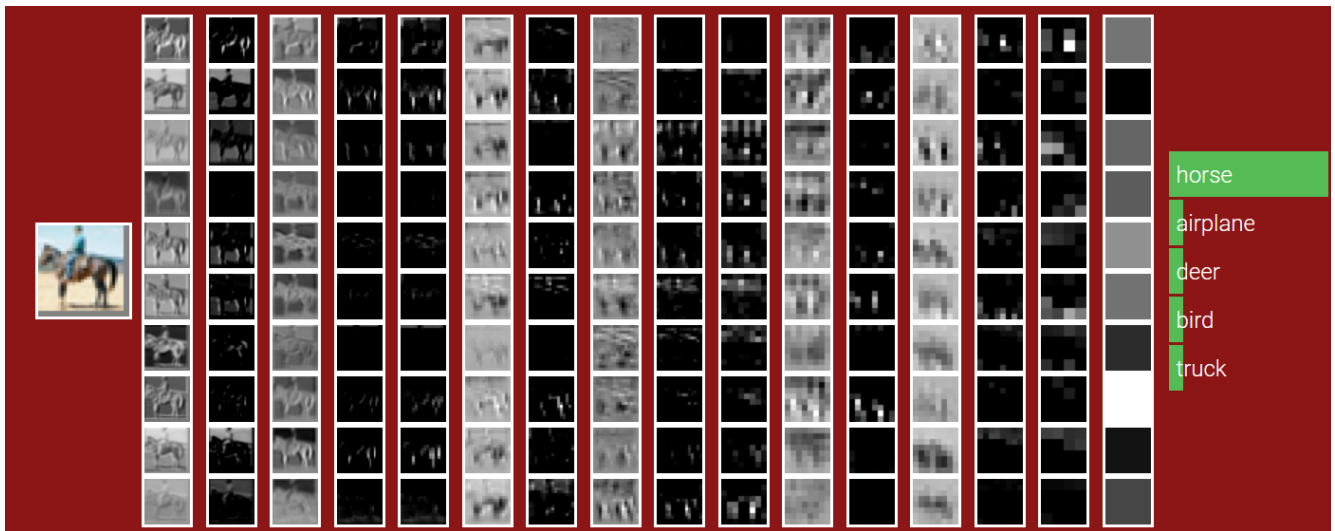
# CNN example



추천: <https://www.slideshare.net/yongho/ss-79607172>, 자습해도 모르겠던 딥러닝, 머릿속에 인스톨 시켜드립니다, 하용호

# CNN example

<http://cs231n.stanford.edu/>



101

# CNN example

<http://cs231n.stanford.edu/>



102

# Programing CNN

[import & setting parameters]

[Visualize a few images]

[Train the model]

```
loss.backward()  
optimizer.step()
```

[Visualize the model prediction]

[Fine tune the convolution net]

```
optim.SGD(model_ft.parameters(), lr = 0.001, momentum = 0.9)
```

[Train & evaluate]