

# Introduction to Artificial Intelligence



2019

Ha-Jin Yu, Dept. of Computer Science, University of Seoul  
서울시립대학교 컴퓨터과학부 유하진

HJYU@UOS.AC.KR



서울시립대학교  
UNIVERSITY OF SEOUL

# 강의 정보

- 에듀클래스 이용
  - <http://club.uos.ac.kr>
- Textbook:
  - 주교재:
    - 텐서플로로 시작하는 딥러닝, 출판사 제이펍, 역자명 진명조
  - 추천 참고 자료:
    - Youtube: Stanford CS230: Deep Learning | Autumn 2018
    - Deep Learning, MIT Press, Ian Goodfellow, Yoshua Bengio and Aaron Courville (<http://www.deeplearningbook.org>)
- 평가
  - 중간시험: 30%, 기말시험 30%
  - 과제: 30%
  - 출석 및 수업참여도 10%



# 연락처

- hjyu@uos.ac.kr
  - 제목이 [인공지능] 이름으로 시작하는 것만 중요 메일로 자동 분류되어 확인함
  - 예) 제목: 교수님 안녕하세요? → spam
  - 예) 제목: 문의드립니다. → spam
  - 예) 보낸 사람 이름이 없는 경우 → spam
- 010-2310-8720
  - 문자 메시지, 카카오톡 가능
- 정보기술관 306호 방문
  - 인공지능 연구 관련 문의 환영
  - 사전 이메일 또는 전화 예약 필요

하루 spam 100개  
제거에 필요한 시간 1시간  
ㅠㅠ

잡상인 ... ㅠㅠ



# 9월16일 휴강, 보강은 보강주간



GRAZ – AUSTRIA  
SEPTEMBER 15<sup>th</sup> – 19<sup>th</sup> 2019  
[WWW.INTERSPEECH2019.ORG](http://WWW.INTERSPEECH2019.ORG)



## »CROSSROADS OF SPEECH AND LANGUAGE«

» [www.interspeech2019.org](http://www.interspeech2019.org)

---

### CALL FOR PAPERS AND PROPOSALS FOR TUTORIALS, SPECIAL SESSIONS/CHALLENGES, AND SHOW & TELL

---

INTERSPEECH is the world's largest and most comprehensive conference on the science and technology of spoken language processing. INTERSPEECH conferences emphasize interdisciplinary approaches addressing all aspects of speech science and technology, ranging from basic theories to advanced applications. In addition to regular oral and poster sessions, INTERSPEECH 2019 will feature plenary talks by internationally renowned experts, tutorials, special sessions and challenges, show & tell sessions, and exhibits. A number of satellite events will also take place around INTERSPEECH 2019.

Original papers are solicited in, but not limited to, the following areas:

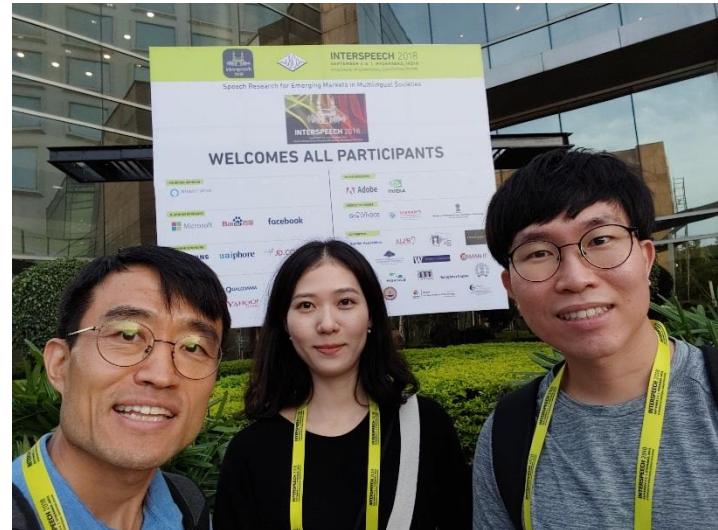
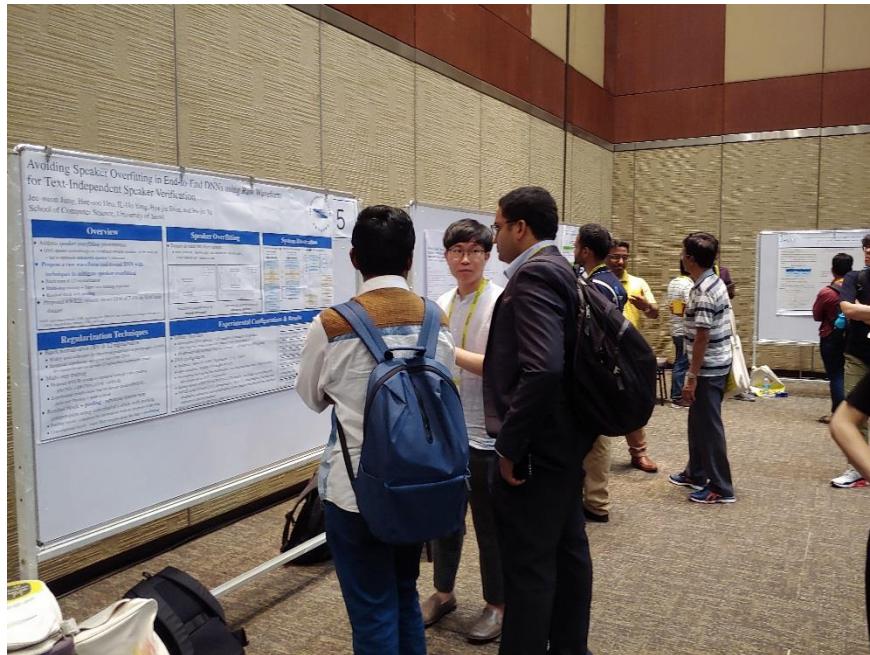
#### PAPER SUBMISSION

Papers intended for INTERSPEECH 2019 should be up to four pages of text. An optional fifth page could be used for references only. Paper submissions must conform to the format defined in the paper preparation guidelines and as detailed in the author's kit on the conference webpage. Please be aware that INTERSPEECH 2019 will use new templates and submissions will be accepted only in the new format. Submissions may also be accompanied by additional files such as multimedia files, to be included on the proceedings' USB drive. Authors must declare that their contributions are original and have not been submitted elsewhere for publication. Papers must be submitted via the online paper submission system. The working language of the



# 대학원 연구실

- 정보기술관 3층 323호 “지능형로봇 연구실”
- 딥러닝을 이용한 음성인식 등 자유로운 연구
- 학부생부터 가능
- 과제 참여로 학비, 생활비 지원
- 해외 학술대회 참석 전액 지원



# **Important conferences**

- NIPS, ICML, CVPR



# 강의 목표

- 기본적인 deep learning 개념 이해
- Tensorflow 기초 학습
- 새로운 deep learning 기술을 학습하기 위한 기초지식 학습

# Intelligence

- Reasoning
- Learning
- Language communication
- Planning
- Decision making
- Common-sense



# AI의 응용 분야

- 영상처리
  - 자율주행 자동차
  - 영상 탐색
- 음성/자연어 처리
  - Smart Speaker
  - “OK, Google”
- Big Data
  - Recommendation
- 보안
  - Speaker/face/finger print recognition
- 금융



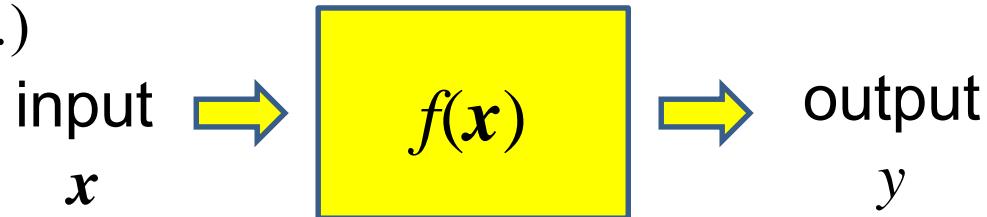
# AI의 역사

- A conference in 1956
  - John McCarthy (one of the organizers of the conference)
  - “Artificial intelligence”라는 용어 사용.
- 지식 기반 (knowledge-based systems)
  - Expert system
    - 특정 분야의 지식을 컴퓨터에 입력하여 그 분야의 전문가의 역할을 수행
- 1980~90년대
  - Multi-layer perceptron
- 2006~
  - Deep learning



# Pattern recognition

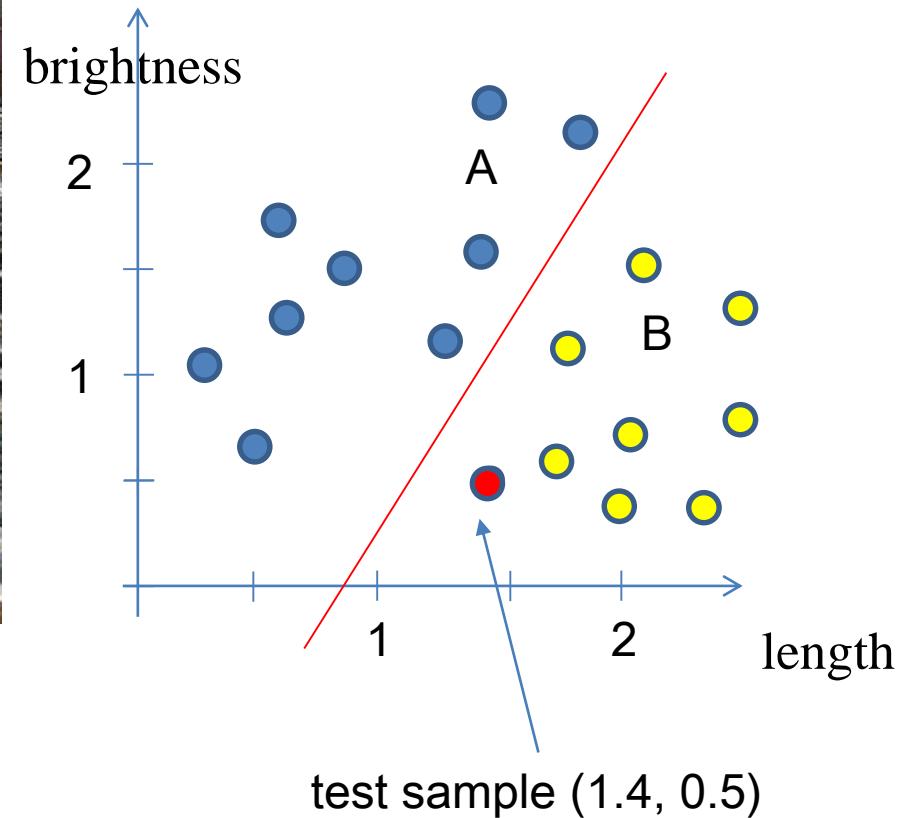
- Speech / speaker
- Image (face, characters, ...)
- Language
- Music
- ...



0 1 2 3 4 5 6 7 8 9

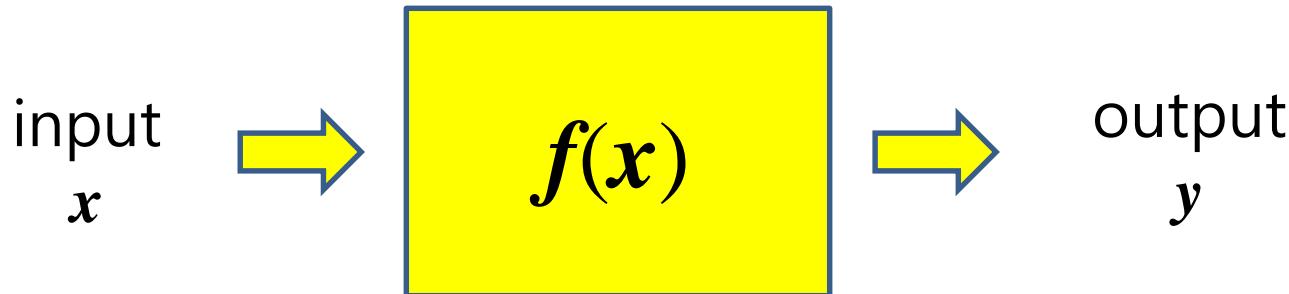
● ○ ○ ○ ○ ○ ○ ○ ○	0
○ ○ ○ ○ ● ○ ○ ○ ○	4
○ ● ○ ○ ○ ○ ○ ○ ○	1
○ ○ ○ ○ ○ ○ ○ ○ ○	9
○ ○ ● ○ ○ ○ ○ ○ ○	2
○ ● ○ ○ ○ ○ ○ ○ ○	1
○ ○ ○ ● ○ ○ ○ ○ ○	3
○ ● ○ ○ ○ ○ ○ ○ ○	1
○ ○ ○ ○ ● ○ ○ ○ ○	4
...	...

# Example: classifying fish



- Features : length, color, brightness, weight, ...

# Goal



- Goal: Input 에 맞는 output을 구하는 함수  $f(x)$ 를 찾는 것.

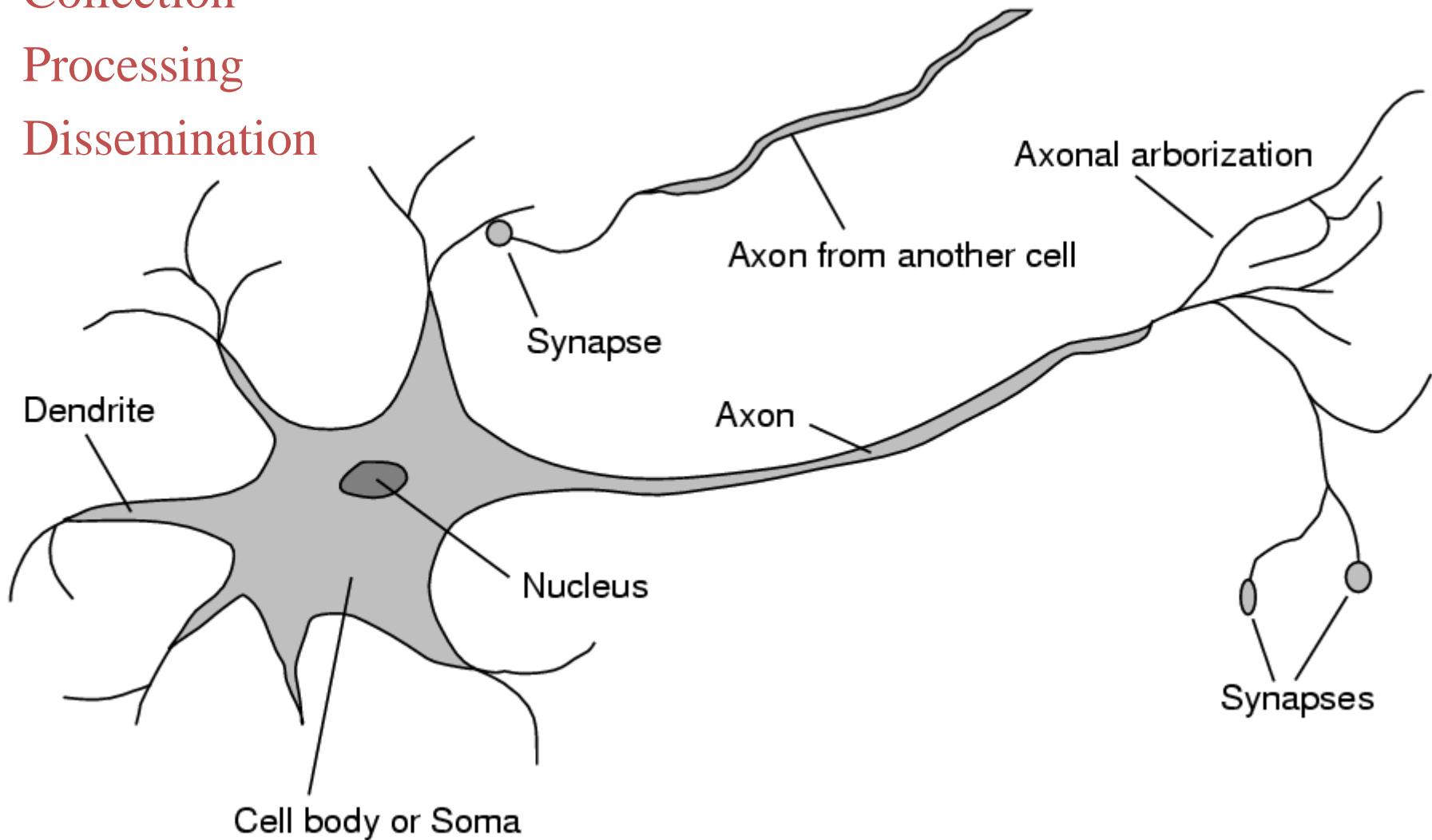
Classifier

- Inputs – feature values
  - 크기, 색, 강도, 위치, 경도, ...
- Outputs – different classes.
  - 종류, 글자, 음소, 단어, ...

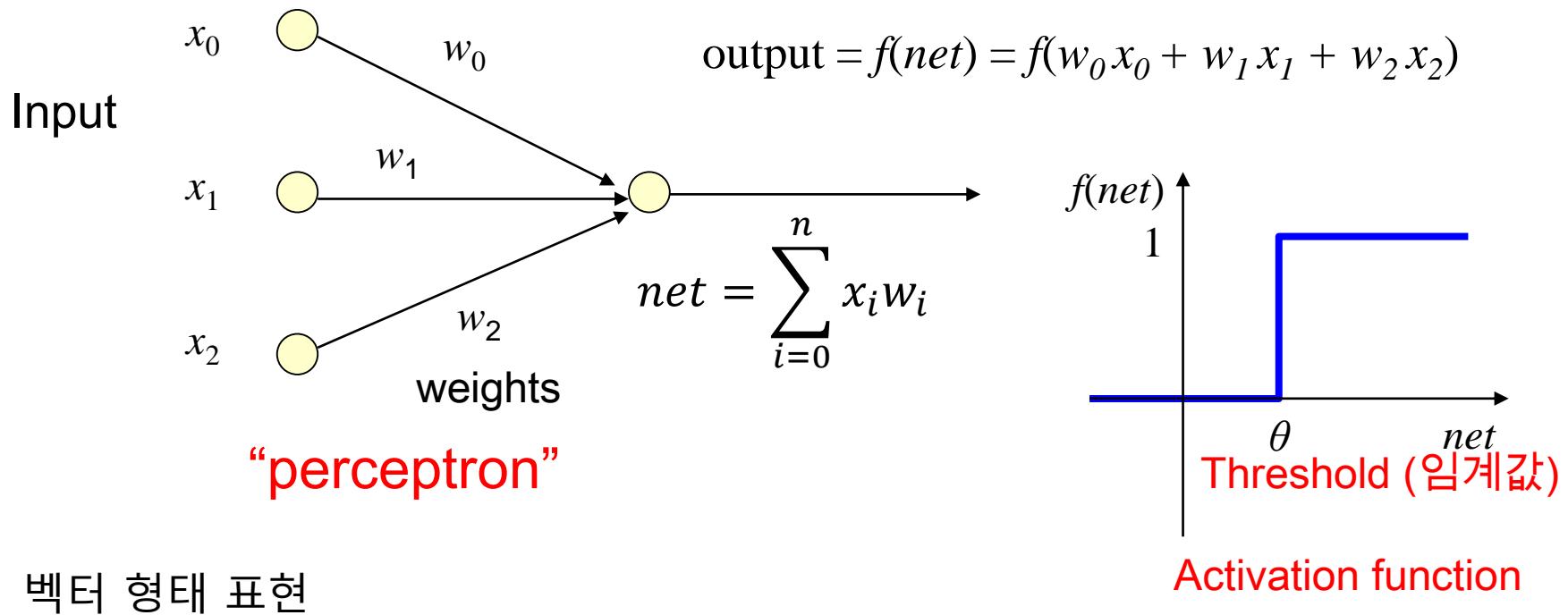


# Neuron : 뇌의 세포

- Collection
- Processing
- Dissemination



# A single network node (unit, neuron)



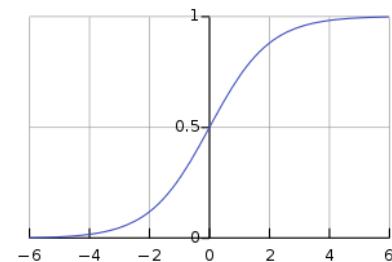
벡터 형태 표현

$$\mathbf{w} = [w_0, w_1, \dots, w_N]$$

$$\mathbf{x}^{(i)} = [x_0, x_1, \dots, x_N]$$

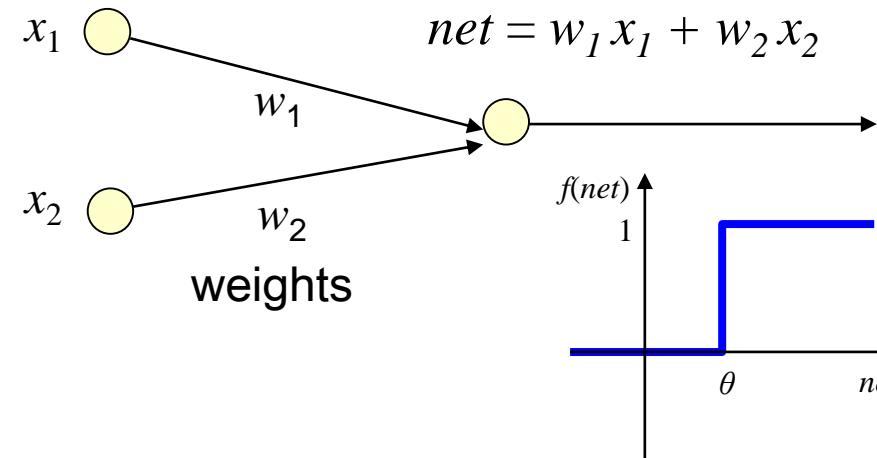
$$\text{net} = \mathbf{w}^T \mathbf{x} + \theta$$

$$f(\mathbf{x}^{(i)}; \mathbf{w}, \theta) = \text{network output}$$



Sigmoid function

# A single network node (unit, neuron)

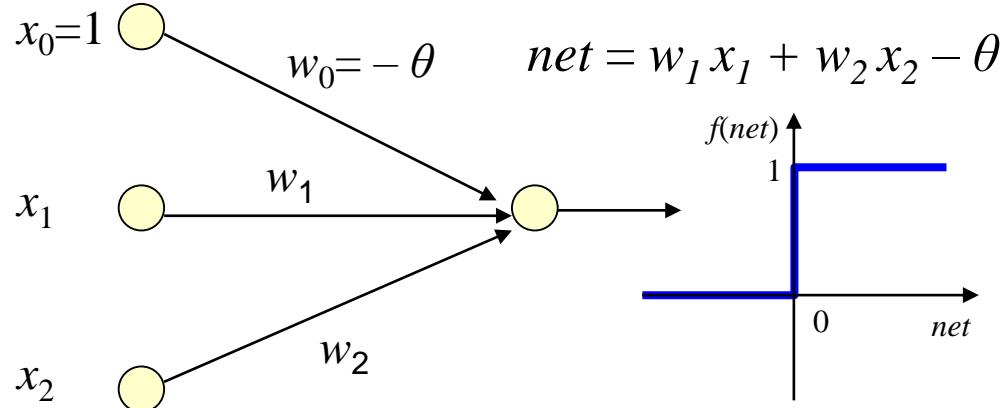


if  $net > \theta$  then  $output = 1$   
else  $output = 0$

if  $net - \theta > 0$  then  $output = 1$   
else  $output = 0$

if  $net + w_0 x_0 > 0$  then  $output = 1$   
else  $output = 0$

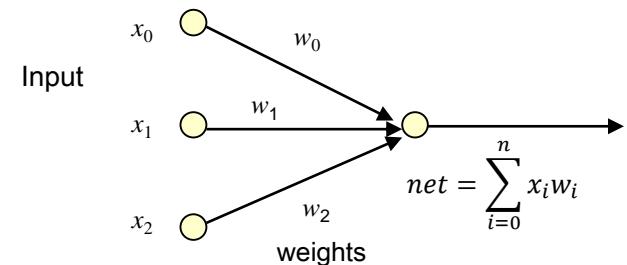
if  $net + w_0 > 0$  then  $output = 1$   
else  $output = 0$



# Neural networks 기본 구조

- Neurons

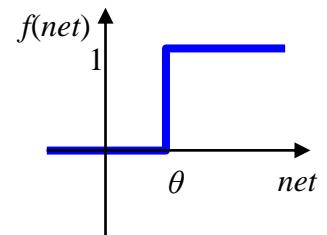
- Input signals  $x_i$
  - Weights  $w_i$ 
    - 정보를 학습, 기억하는 역할



- An activation level  $net$
  - An activation function (threshold function)
    - Activation level  $net$  threshold 보다 높으면 activate

- Global properties

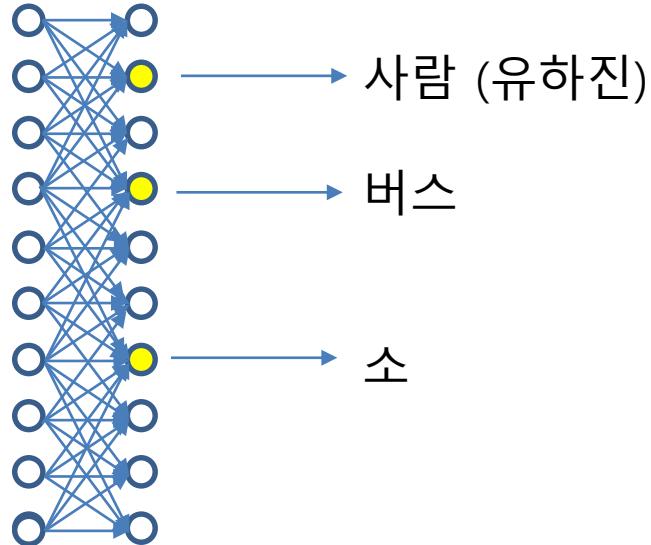
- The network topology
    - Neurons의 연결 구조
    - Feed forward, feed backward
  - The learning algorithm
  - The encoding scheme



# Neural network을 이용한 분류



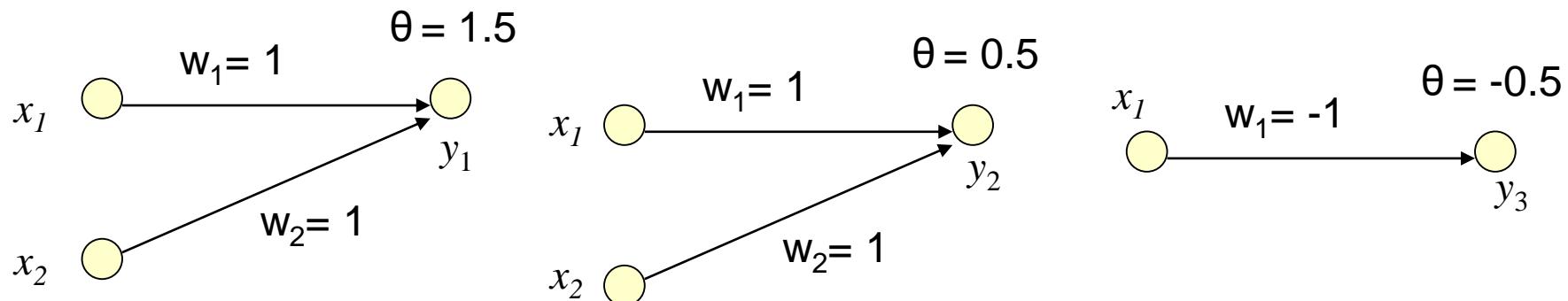
input



Quiz: 여기는 어느 나라?



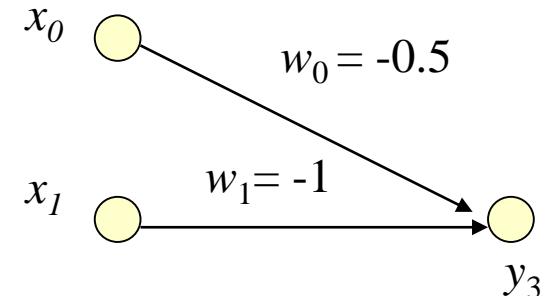
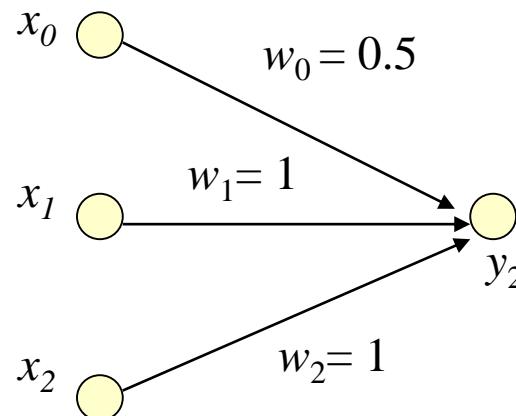
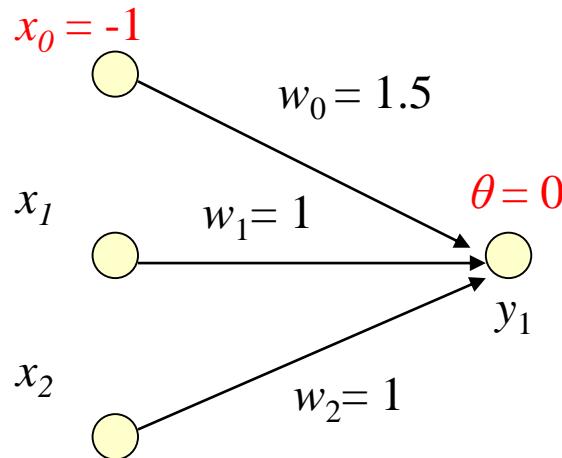
# Example: Logic Gates



input		$y_1$		$y_2$		$y_3$	
$x_1$	$x_2$	net	out	net	out	net	out
0	0						
0	1						
1	0						
1	1						



# Logic Gates ( $\theta=0$ 으로 고정하고, $w_0$ 이용)

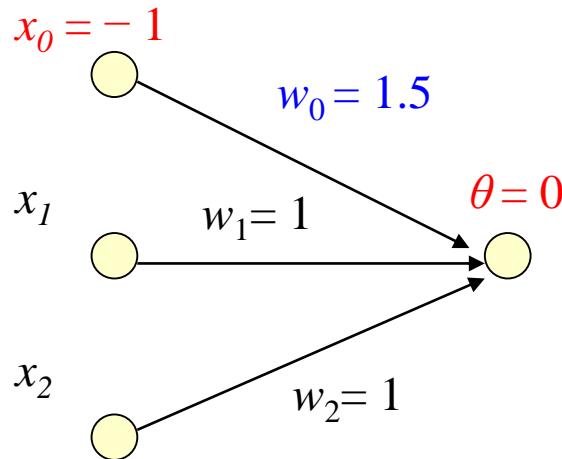


input		AND		OR		NOT	
$x_1$	$x_2$	net	out	net	out	net	out
0	0						
0	1						
1	0						
1	1						



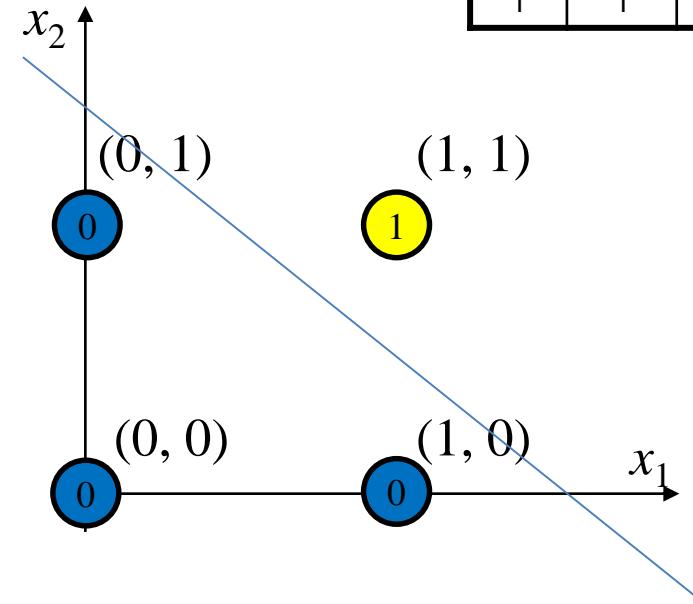
# What is perceptron?

- AND gate



$$\begin{aligned} \text{net} &= w_0 x_0 + w_1 x_1 + w_2 x_2 \\ &= -1.5 + x_1 + x_2 = 0 \end{aligned}$$

$$x_2 = -x_1 + 1.5$$

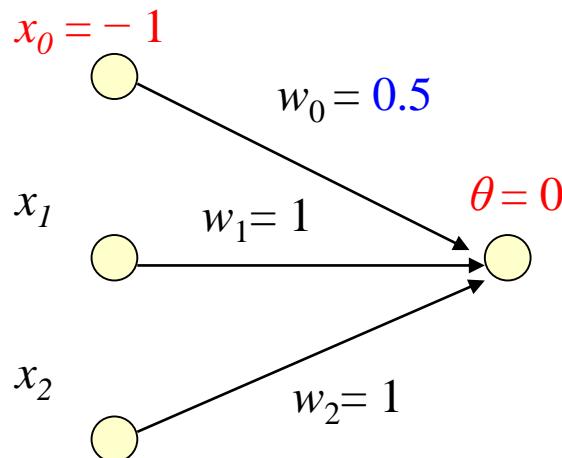


input		output
$x_1$	$x_2$	
0	0	0
0	1	0
1	0	0
1	1	1



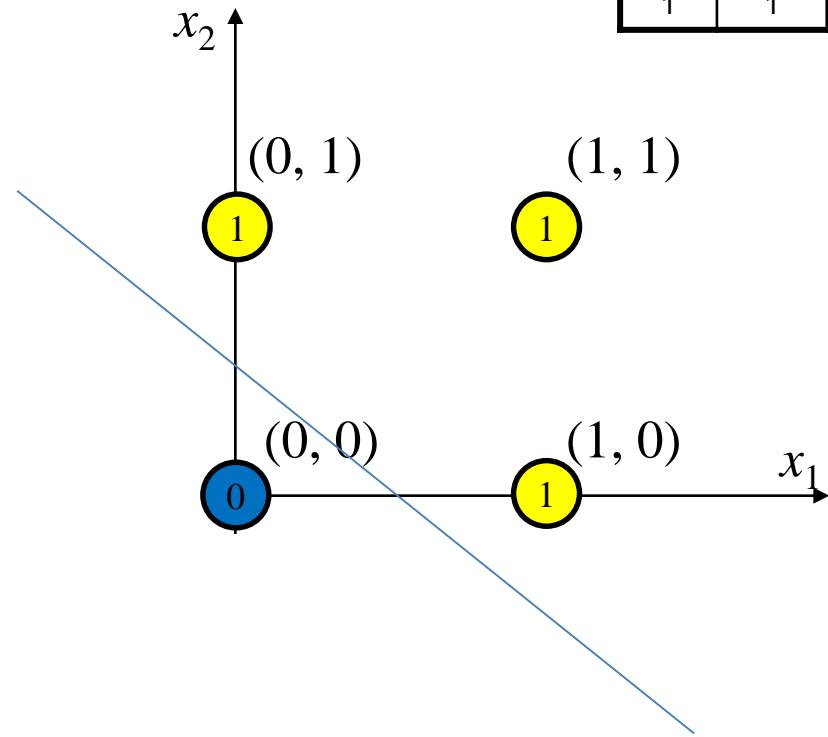
# What is perceptron?

- OR gate



$$\begin{aligned} \text{net} &= w_0 x_0 + w_1 x_1 + w_2 x_2 \\ &= -0.5 + x_1 + x_2 = 0 \end{aligned}$$

$$x_2 = -x_1 + 0.5$$



input		output
$x_1$	$x_2$	
0	0	0
0	1	1
1	0	1
1	1	1



# Decision function

- A line that separates the two classes

$$x_2 = 1.5x_1 + 0.5$$

- Decision function:

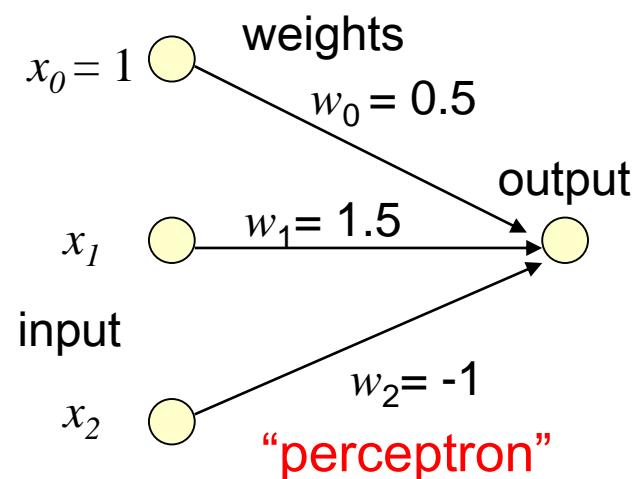
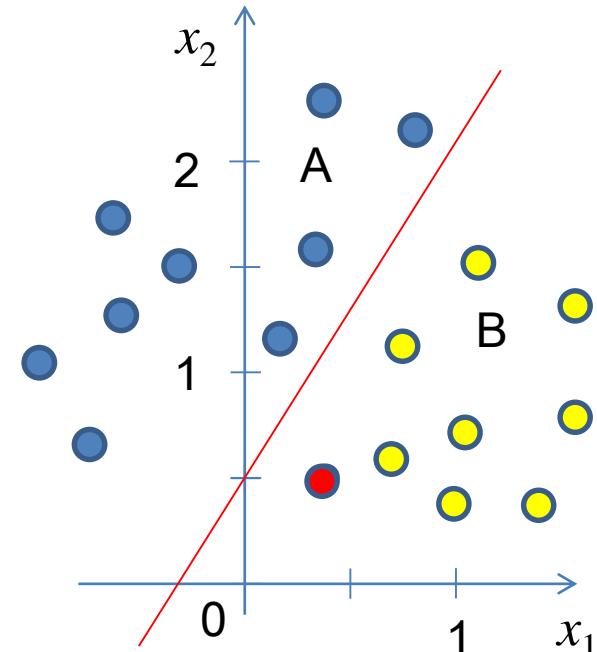
$$f(x_1, x_2) = -x_2 + 1.5x_1 + 0.5$$

class B if  $f(x_1, x_2) \geq 0$

class A if  $f(x_1, x_2) < 0$

Example) a new sample ( 0.4, 0.5 )

- $f(0.4, 0.5) = .05 + (1.5 \times 0.4) + 0.5 = 0.6$   
→ class B
- Decision function → **neuron**

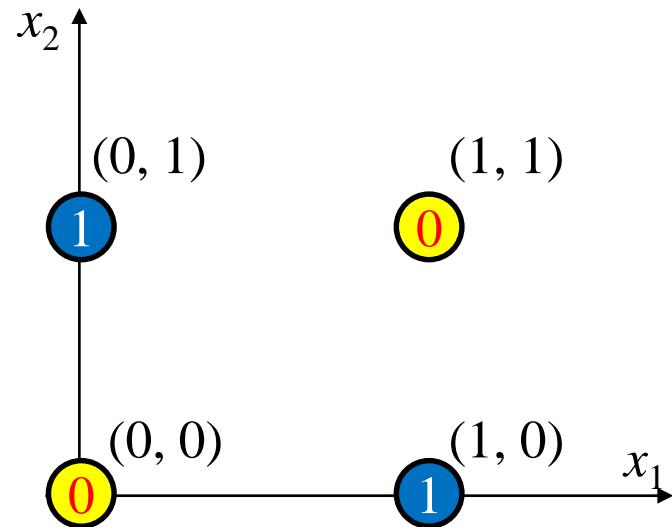


# Limitations of the perceptron model

- Not linearly separable.

Input (x)	Output (y)
(0, 0)	0
(0, 1)	1
(1, 0)	1
(1, 1)	0

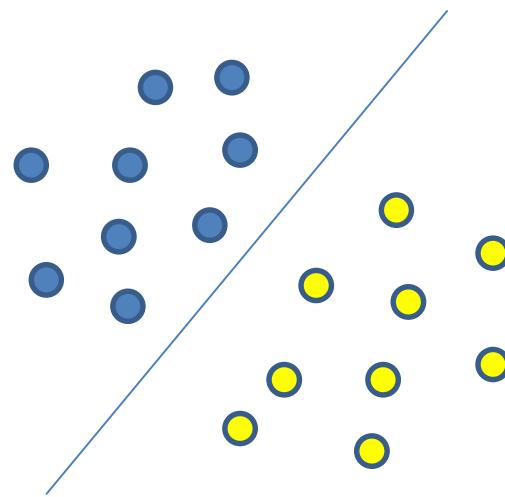
The truth table for exclusive-or.



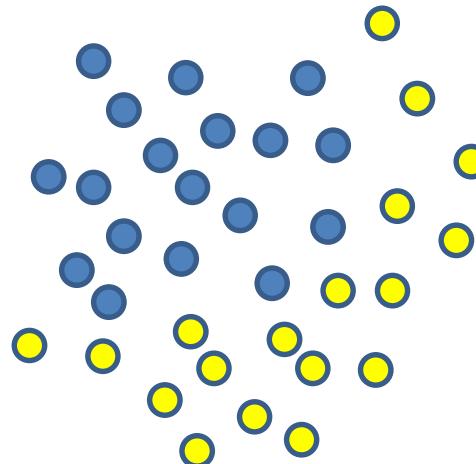
# Linear and non-linear problems

- Classification task

- **Linear** problem : 하나의 직선이나 평면, 초평면으로 모든 sample 의 class를 분리할 수 있을 때  
↔ **Non-linear** problem



**linear** problem

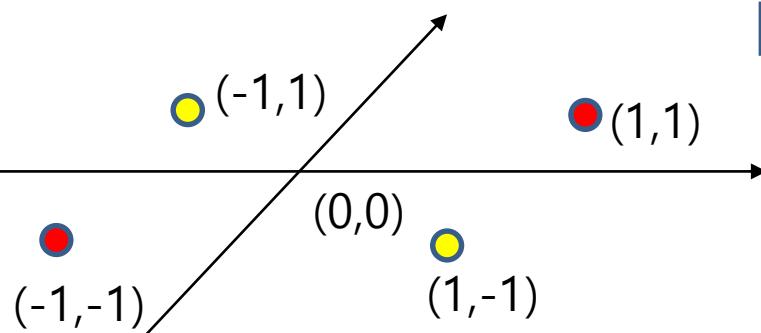


**non-linear** problem

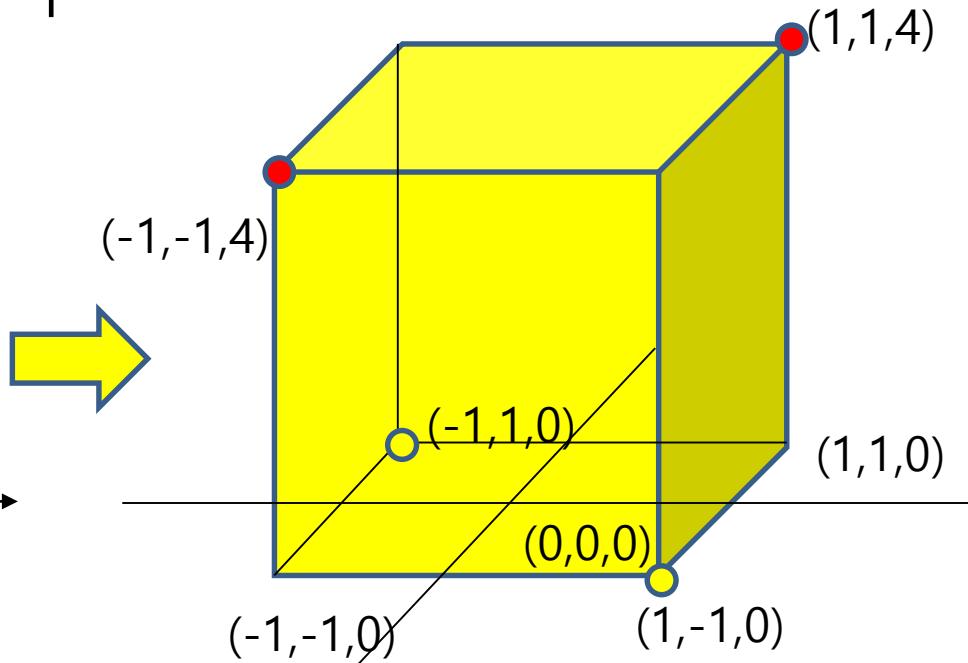
# Solving non-linear problem with a neural network

1. 직선의 개수를 늘인다.
2. Input vectors 의 차원을 높인다

Augmenting input  $(x_1+x_2)^2$



하나의 직선으로 분리 불가능  
(non-linear)

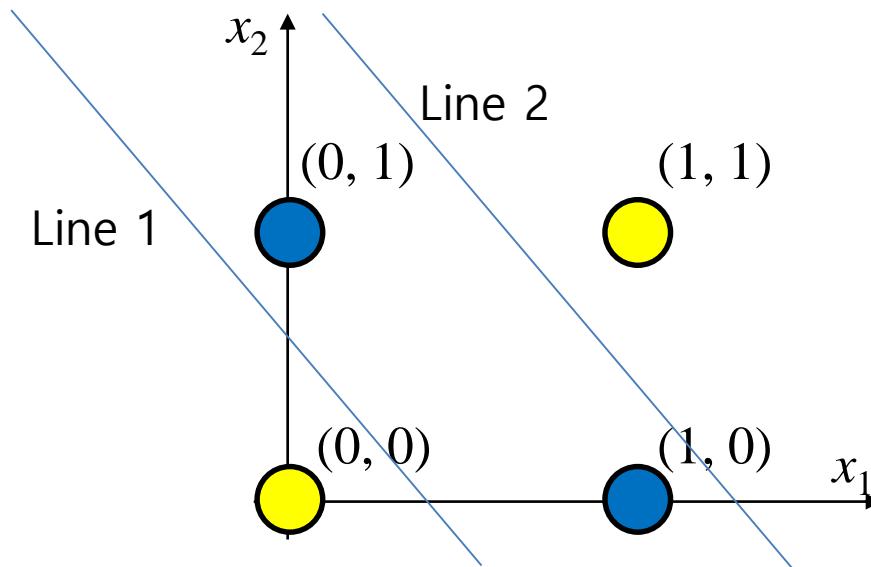


평면으로 분리 가능  
(linear)



# Solving non-linear problem with a neural network

- Solution : uses **two lines** to separate the data
- Two lines → two nodes → multilayered perceptron
- Require
  - Two hidden nodes, both connected to two input nodes to represent the two lines
  - A third node to combine the outputs from the two hidden (middle) nodes.

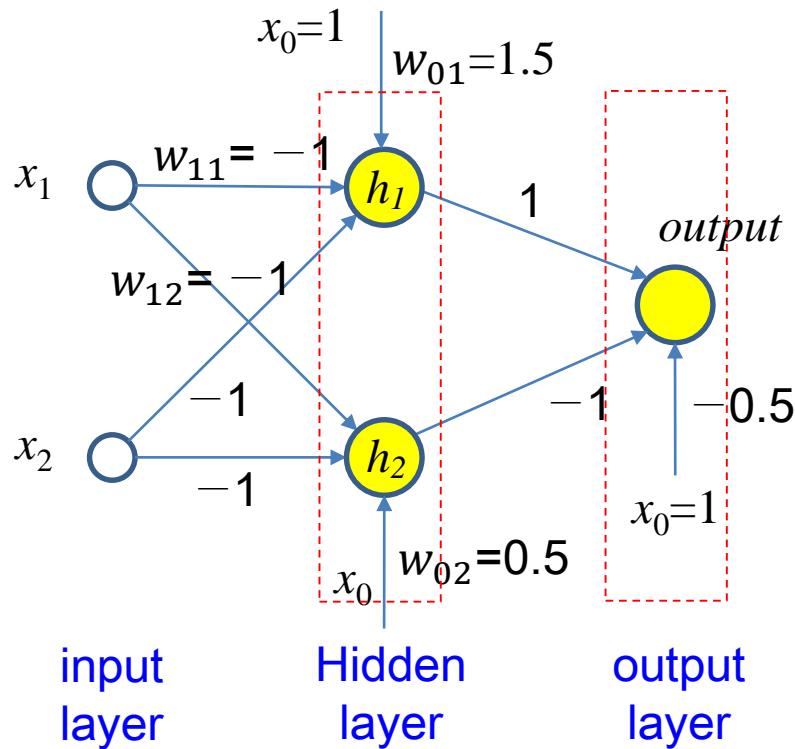


The XOR problem solved with two separating lines.



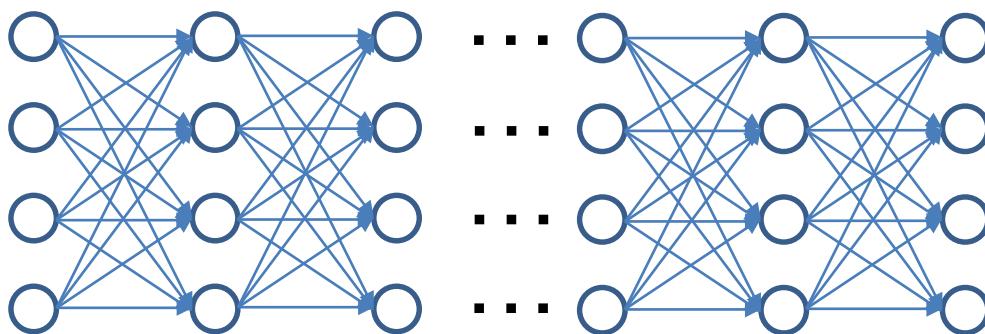
# Network used to model the XOR relation

Two-layered perceptron



$x_1$	$x_2$	$net_1$	$h_1$	$net_2$	$h_2$	$net_3$	output
0	0						
1	0						
0	1						
1	1						

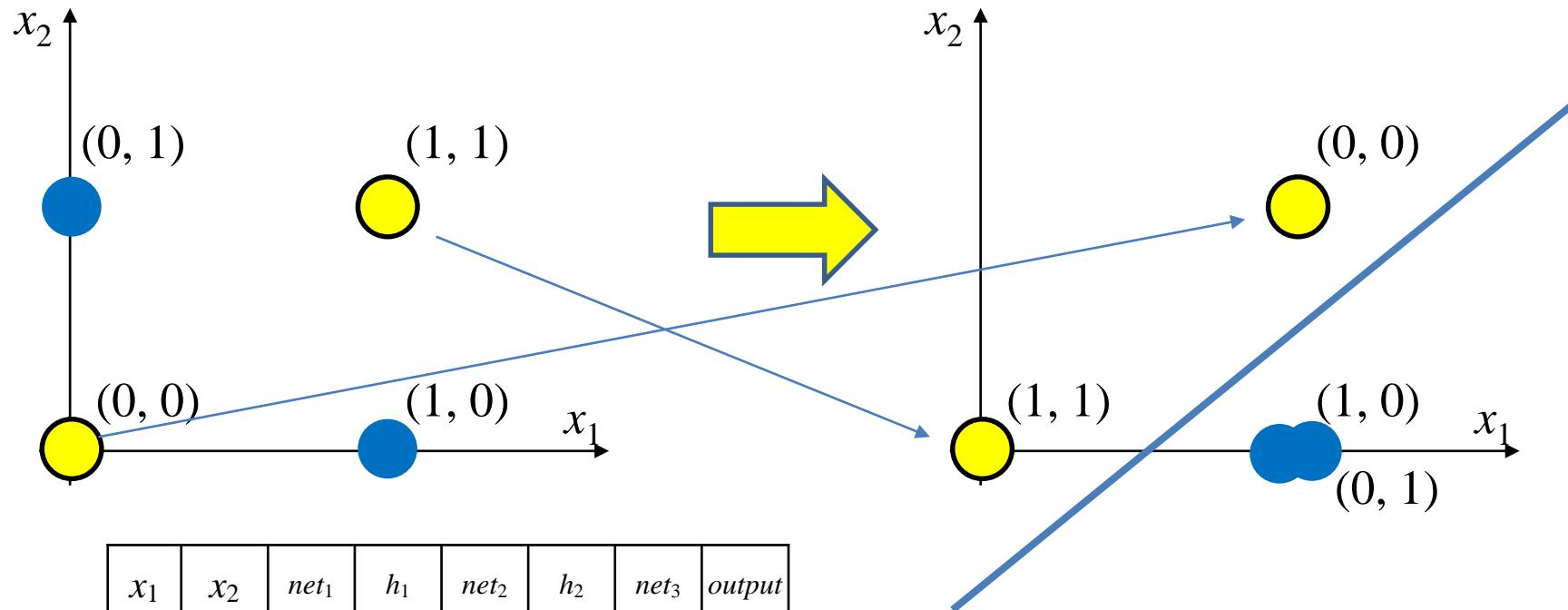
$$net_j = \sum_{i=0}^n x_i w_i \quad h_1 = f(net_1)$$



*Deep neural  
networks*

# 첫번째 layer 의 역할은?

- Input 을 선형 분리 가능(linearly separable)하도록 변경



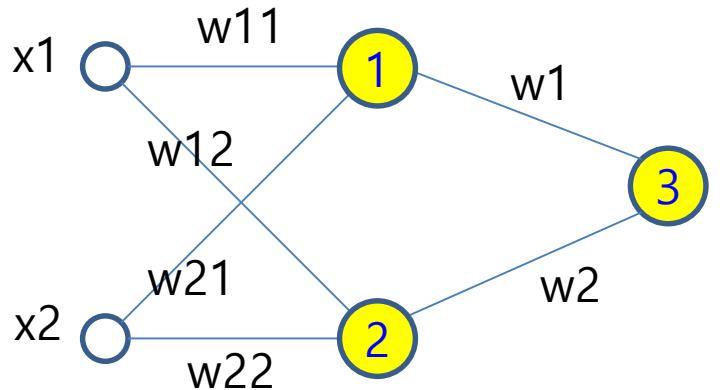
$x_1$	$x_2$	$net_1$	$h_1$	$net_2$	$h_2$	$net_3$	output
0	0	1.5	1	0.5	1	-0.5	0
1	0	0.5	1	-0.5	0	0.5	1
0	1	0.5	1	-0.5	0	0.5	1
1	1	-0.5	0	-1.5	0	-0.5	0



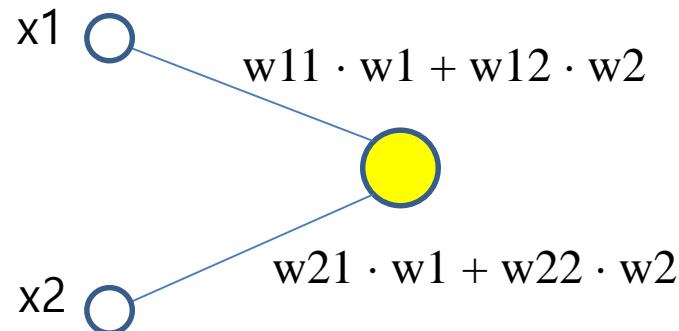
# Non-linear activation function 의 필요성

- Network 에 non-linear activation function 이 없으면?
  - Linear activation functions → a single layer of weights
- ∴ Multilayered networks 에는 non-linear activation function 필요.

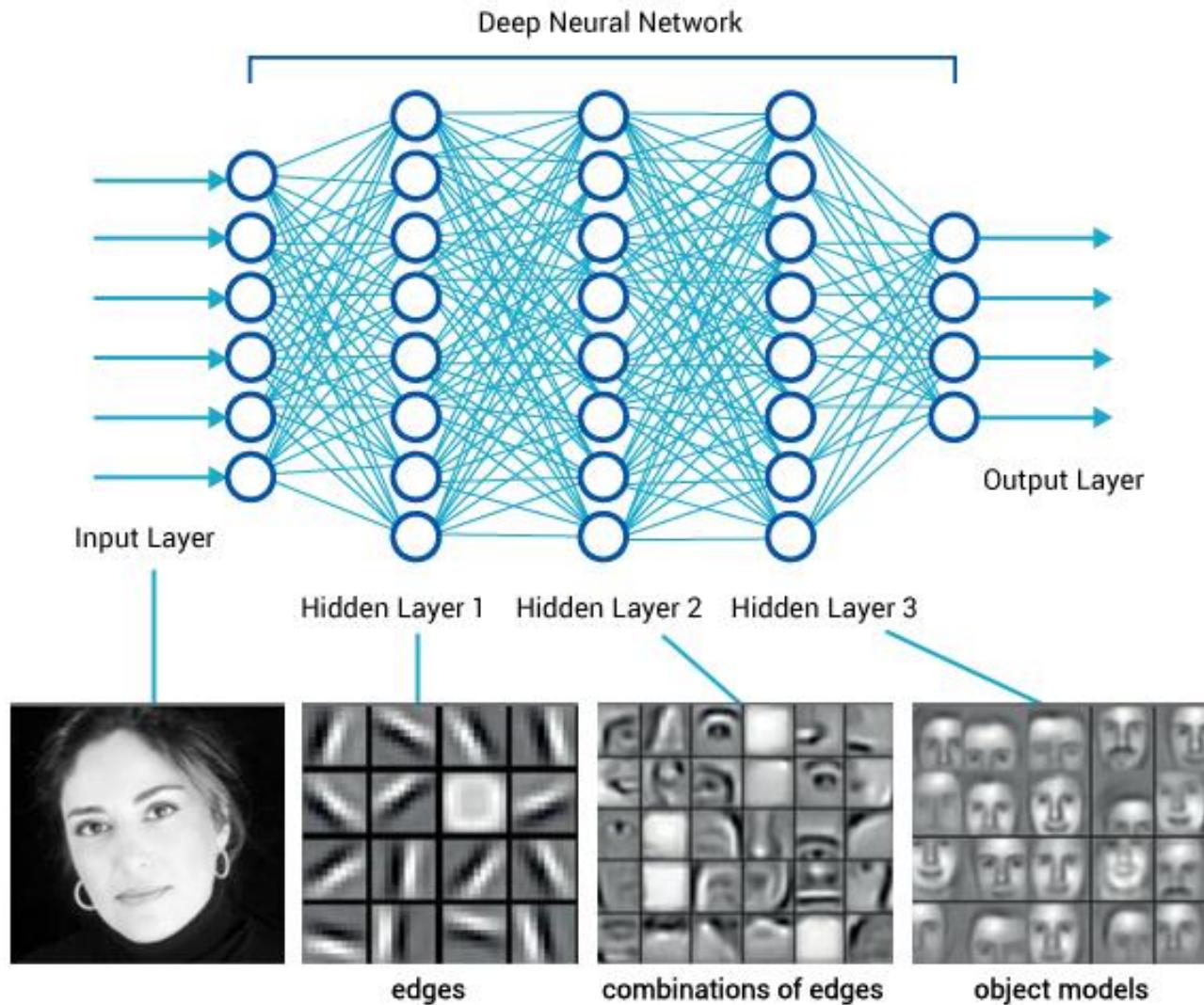
$$\begin{aligned} \text{net3} &= \text{net1 } w1 + \text{net2 } w2 \\ &= (\text{x1 } w11 + \text{x2 } w21) w1 + (\text{x1 } w12 + \text{x2 } w22) w2 \\ &= \text{x1 } (w11 w1 + w12 w2) + \text{x2 } (w21 w1 + w22 w2) \quad \rightarrow \text{single layer !!} \end{aligned}$$



=

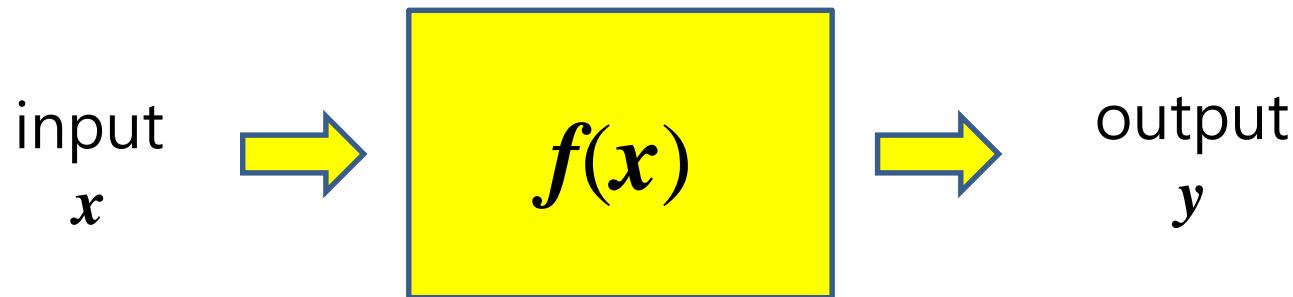


# Layers and their abstraction power in a deep net



# 학습 (learning)

- 같은 구조이지만 weights 값에 따라서 output 이 달라진다.



- Goal: Input 에 맞는 output을 구하는 함수  $f(x)$ 를 찾는 것.
- 함수  $f(x)$ 를 찾는 것은?  
== Input 에 맞는 output을 구하는 weights를 찾는다는 것.  
→ 학습 (learning)
- Weights의 개수는?



# 과제 #1 perceptron 구현

- Input 차원이 n인 1-layer perceptron 을 구현한다.
- Weight 값들을 random 값으로 초기화한다.
- 다음을 무한 반복한다.
  - 앞의 AND gate의 입력과 같은 값을 차례로 입력하여 output을 구한다.
  - 네 가지의 input에 대하여 output이 틀린 개수를 구한다.
  - 각 input 별 output 이 모두 맞으면 종료.
  - 임의의 값을 입력 받아 각 weight에 대입한다.

- C, C++언어로 구현
- 결과물:
  - source (주석 포함)
  - 결과 보고서 (어떤 값을 입력해야 무한 loop에서 빨리 나올 수 있는지?)
- Educlass 에 제출
- 마감: 9월9일 14:00

