

# Parrot Deep Learning

Session 01.

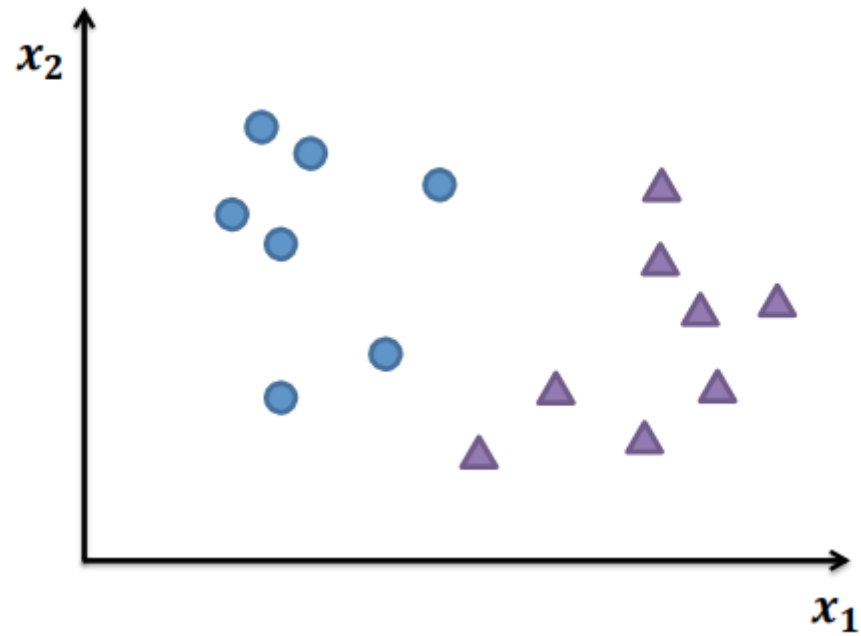
xor gate, linear classifier, mlp



# *Deeeeeeep learning*

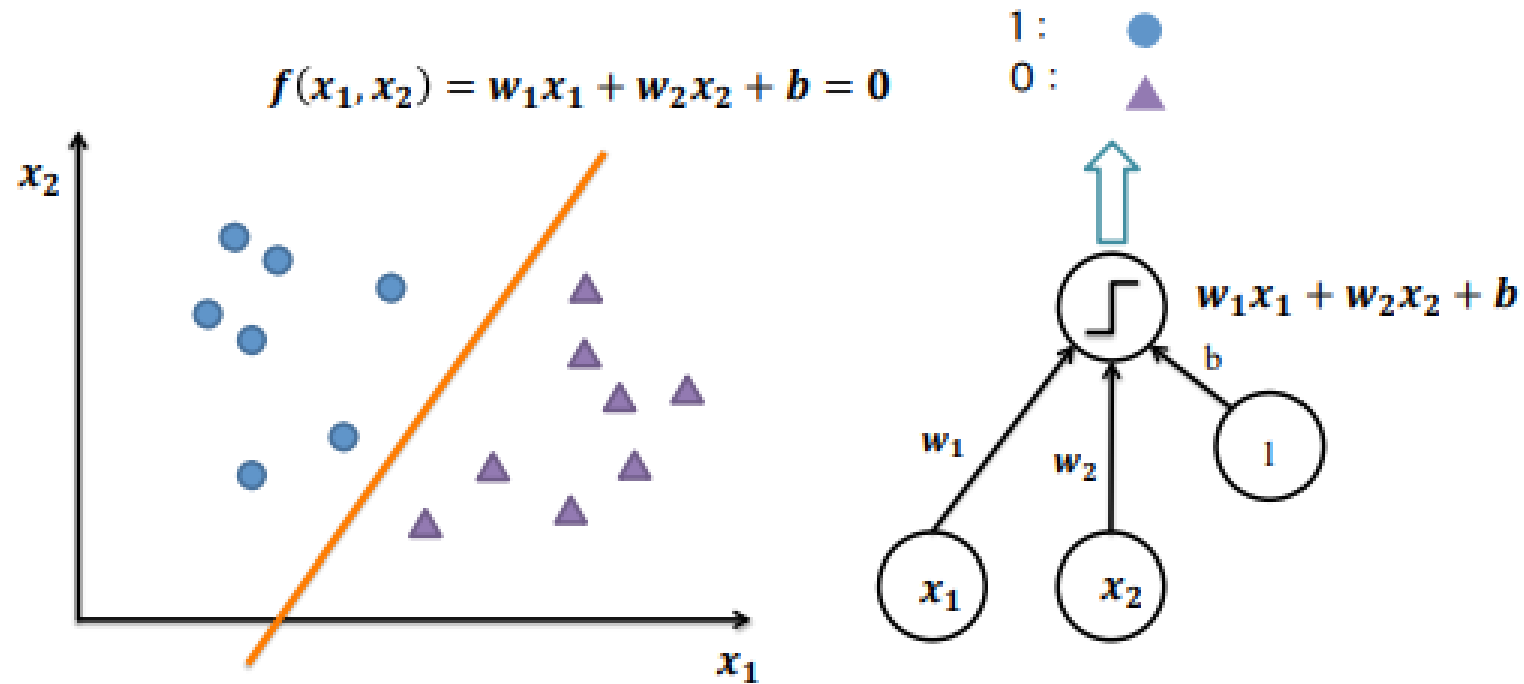
- 딥러닝이 그래서 도대체 뭘까요?
- 대체 뭐가 deep 하죠..?
- 왜 deep 해야 하나요? deep 하면 뭐가 좋죠?
- 우리 머신러닝도 잘해왔지 않나요..?
- 딥러닝은 어떻게 예측하나요?

# Linear classifier



- $y = ax + b$ 의 형태로 나눌 수 있다
- $f(x_1, x_2) = w_1x_1 + w_2x_2 + b = 0$

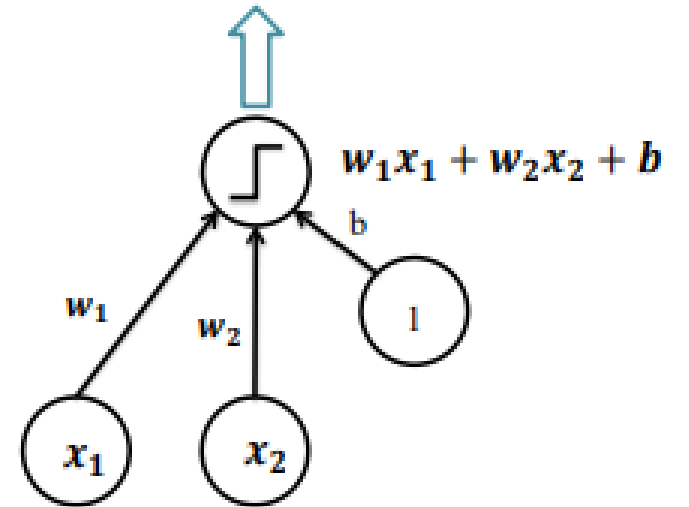
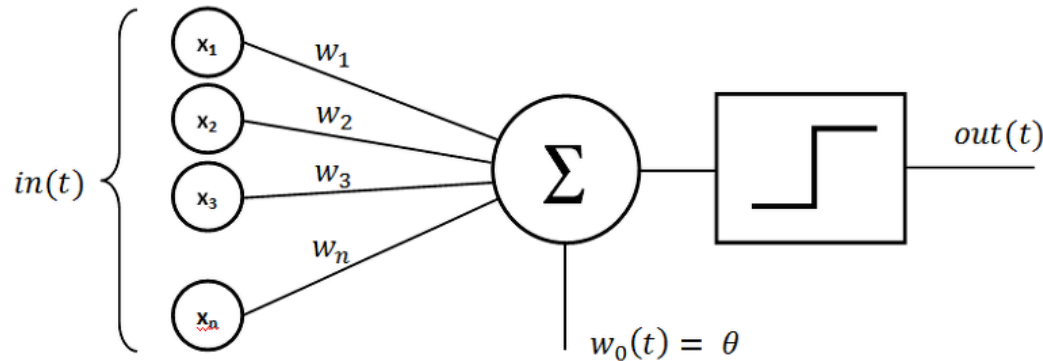
# Linear classifier



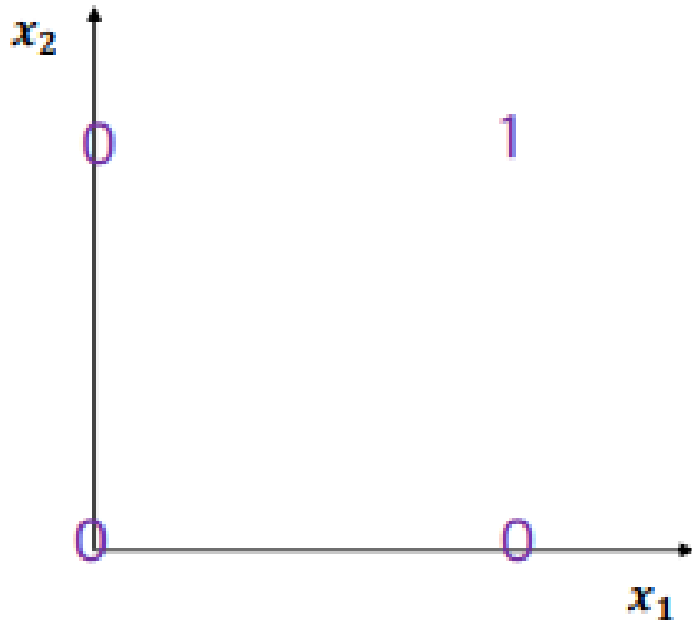
- 다수의 입력값을 하나의 출력으로 만들고 있네요

# 퍼셉트론 - *Perceptron*

- 사람의 뉴런과 비슷한 인공 신경망의 구성 요소입니다
- 입력에 가중치에 곱해 합친 후 활성화 함수를 적용시킵니다
- 가중치가 크다? 결과에 큰 영향을 준다. 중요한 입력값이다



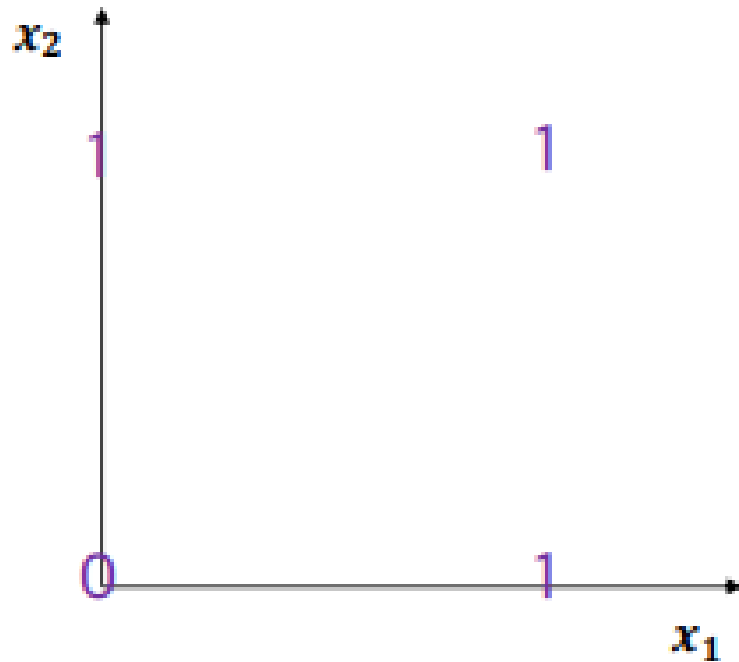
## Linear classifier – AND 분류



Input		Output
$x_1$	$x_2$	$y$
0	0	0
0	1	0
1	0	0
1	1	1

- $f(x_1, x_2) = w_1x_1 + w_2x_2 + b = 0$
- $w_1, w_2, b$ 를 추정해 봅시다 (1, 1, -1.5)

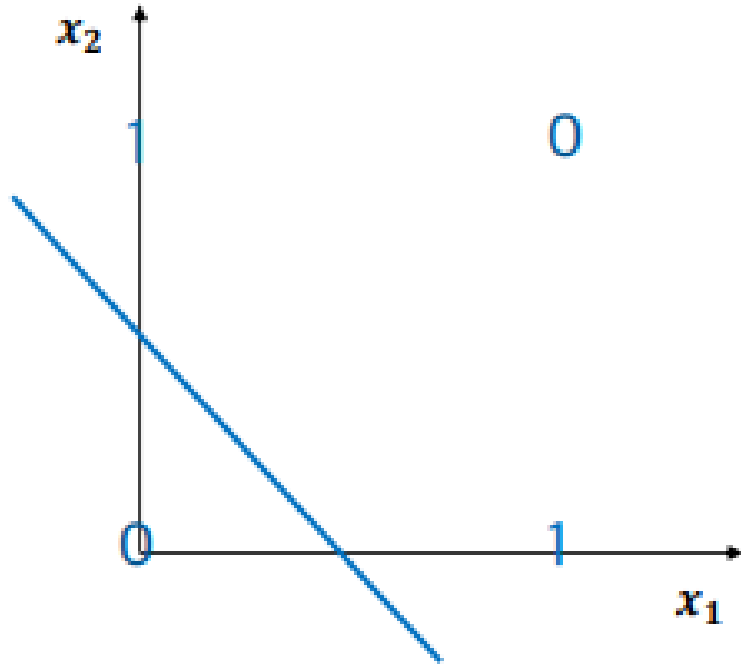
## Linear classifier – OR 분류



Input		Output
$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	1

- $f(x_1, x_2) = w_1x_1 + w_2x_2 + b = 0$
- $w_1, w_2, b$ 를 추정해 봅시다 (1, 1, -0.5)

## Linear classifier – XOR 분류

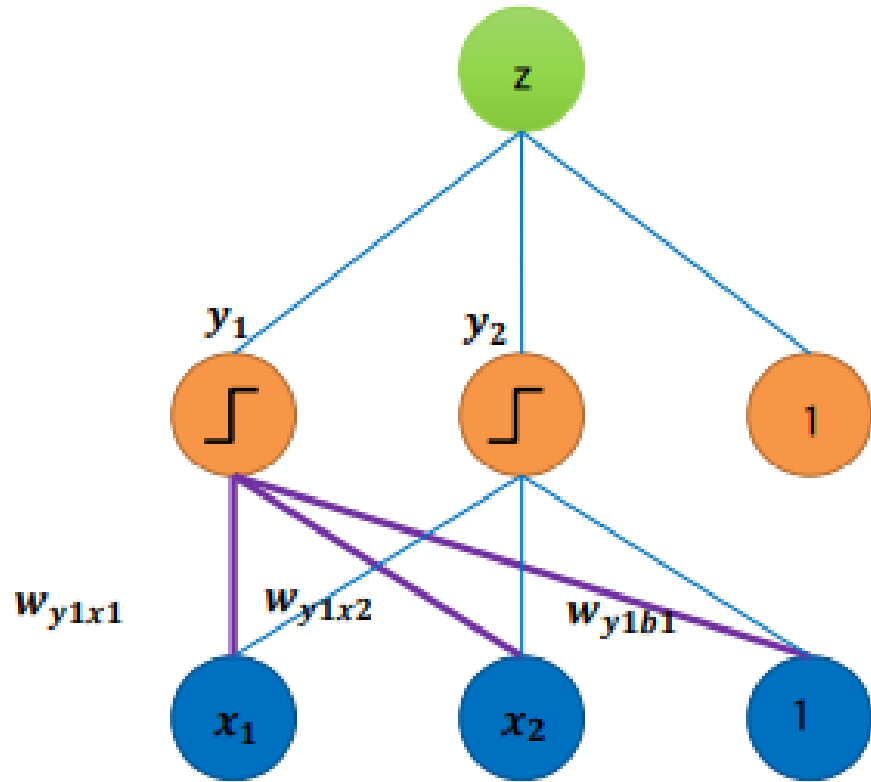


Input		Output
$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

- 이 녀석은 어찌죠..?
- 어떻게 하면 선으로 1과 0을 분리 할 수 있을까요?



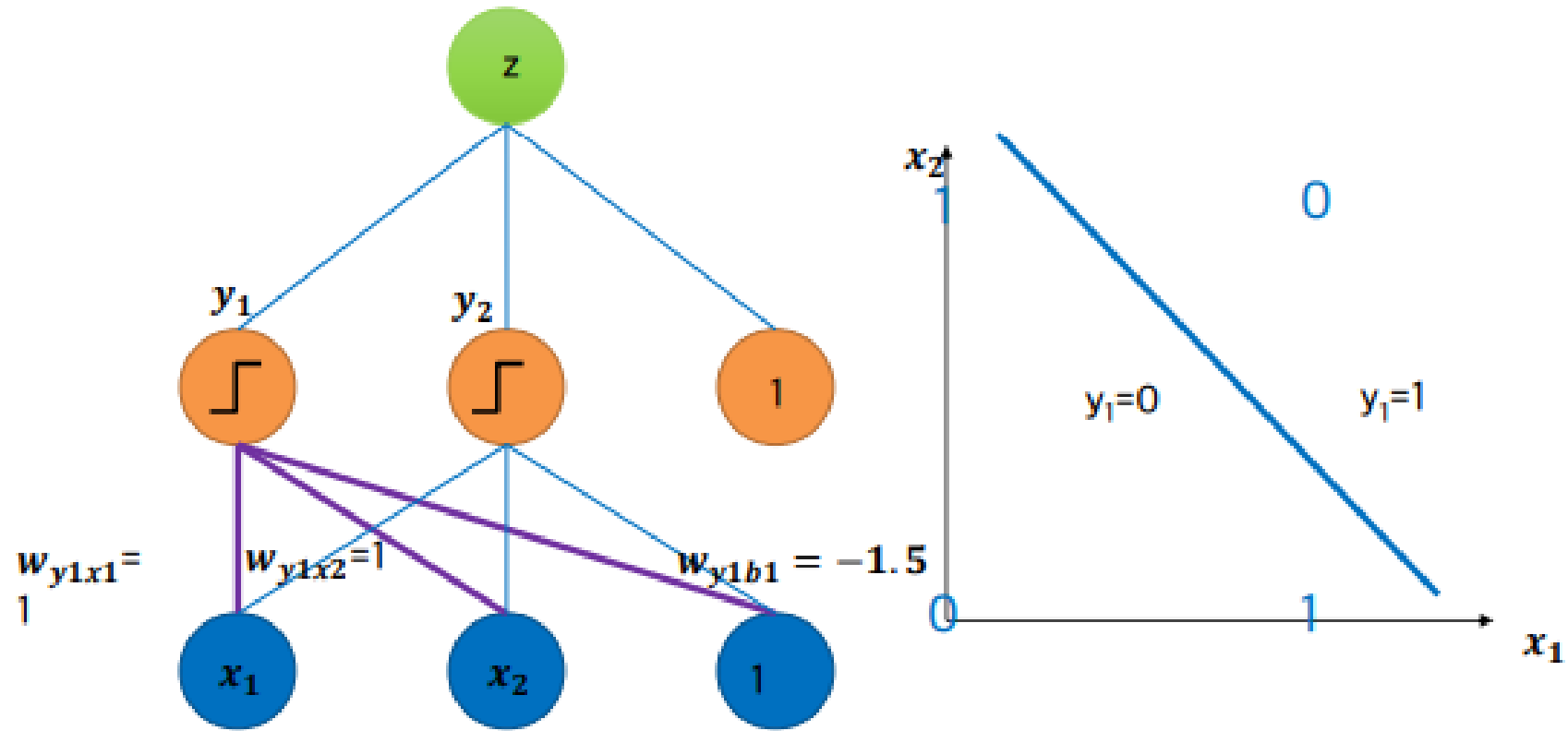
# Multi-layer Perceptron



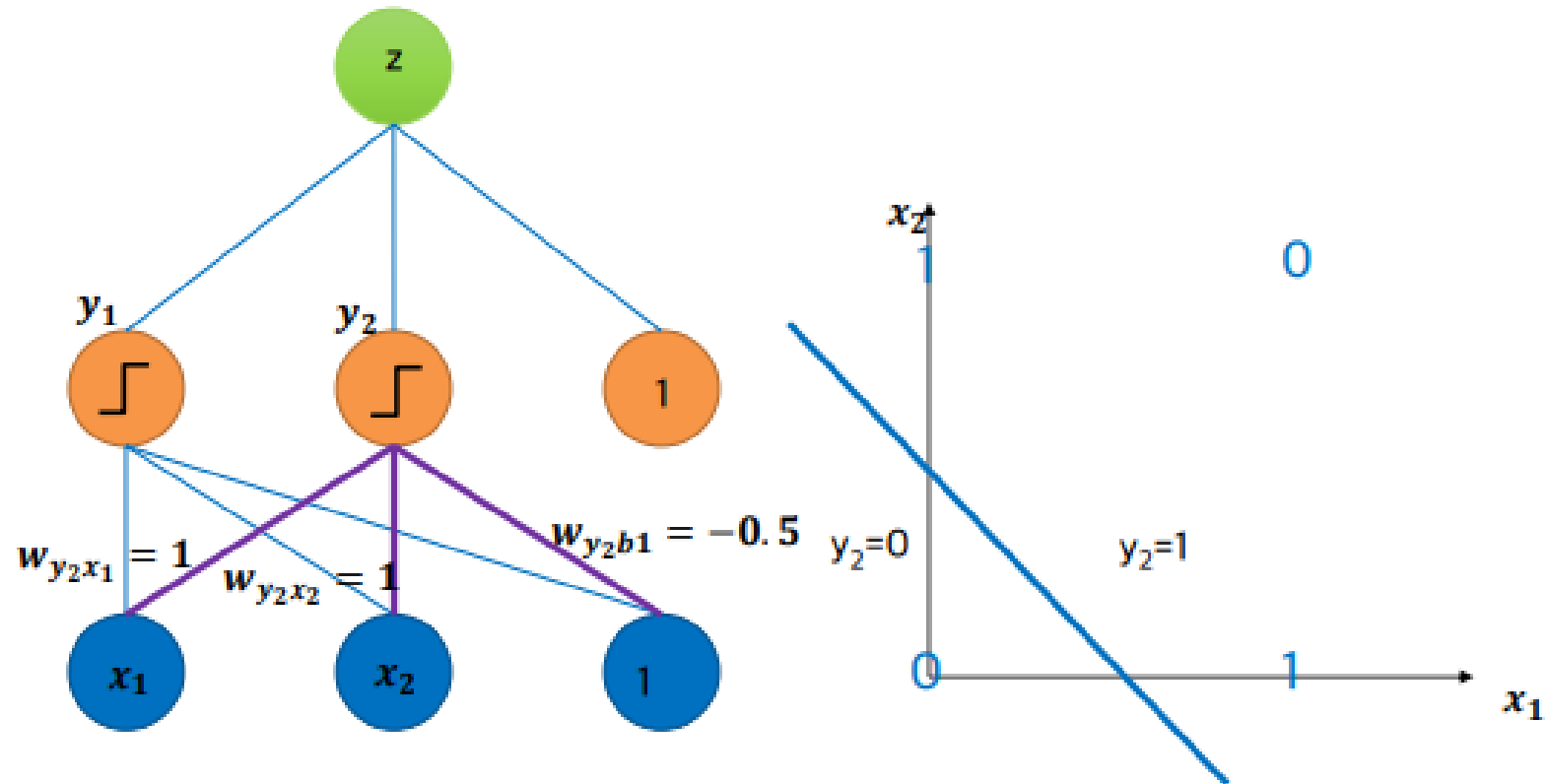
여기엔 그림을 그리겠죠

- 하나의 레이어가 아닌, 다중 레이어를 사용하여 이를 구현 가능합니다
- 선대를 잘 배웠다면 하나의 퍼셉트론은 일종의 affine linear transformation입니다

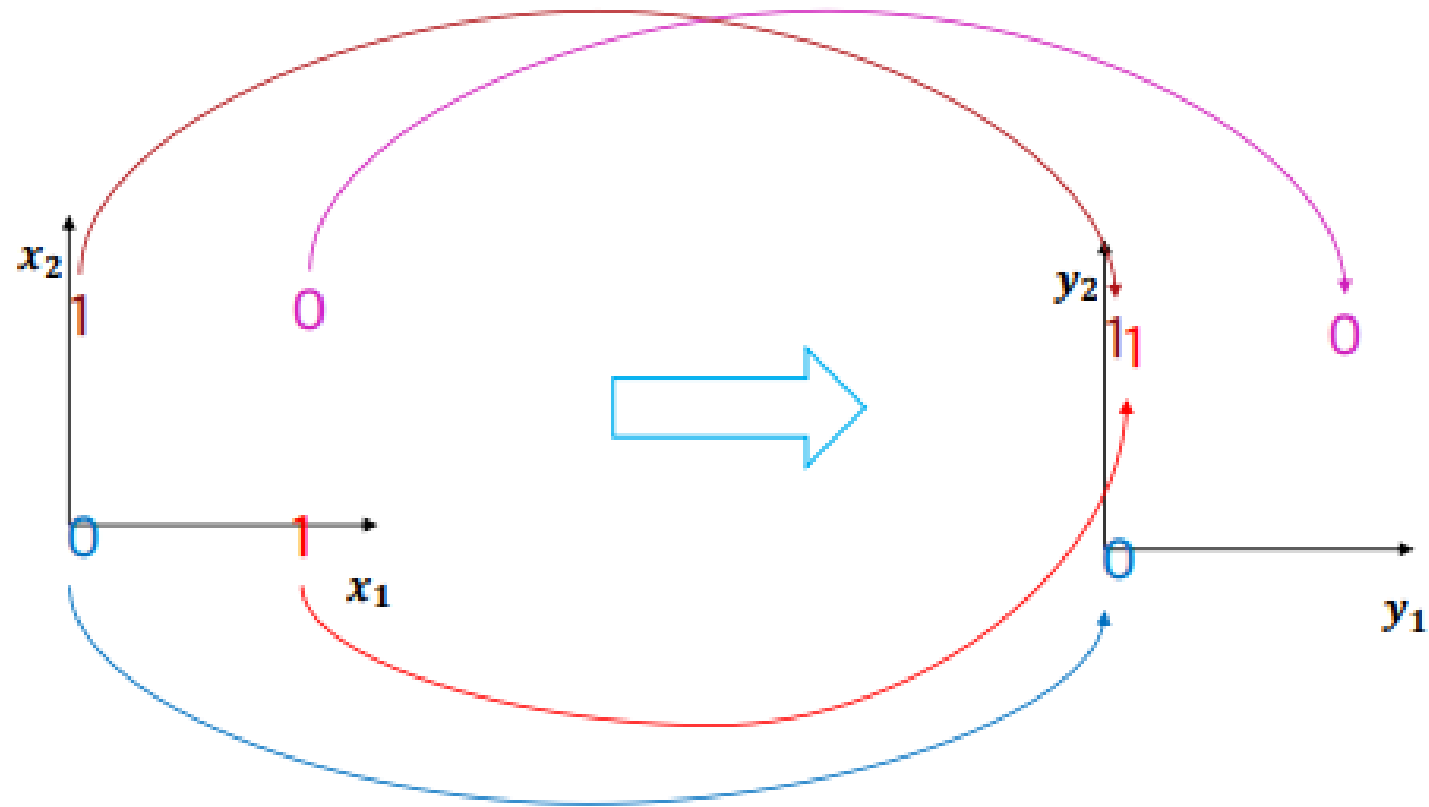
# Multi-layer Perceptron



# Multi-layer Perceptron

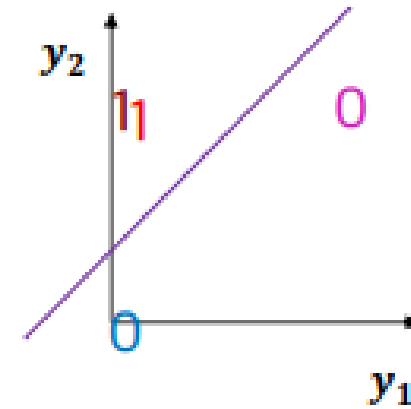
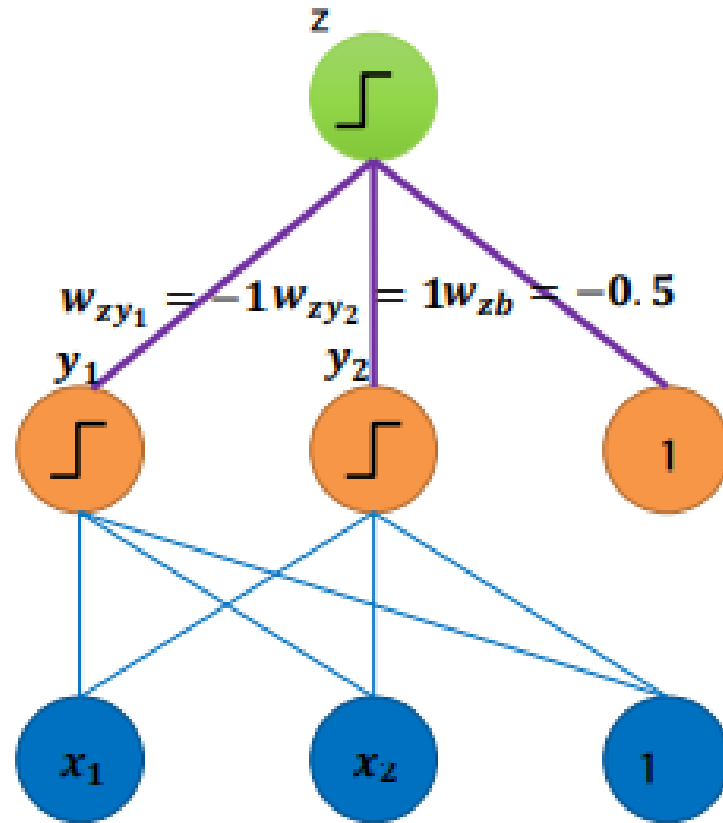


# Multi-layer Perceptron



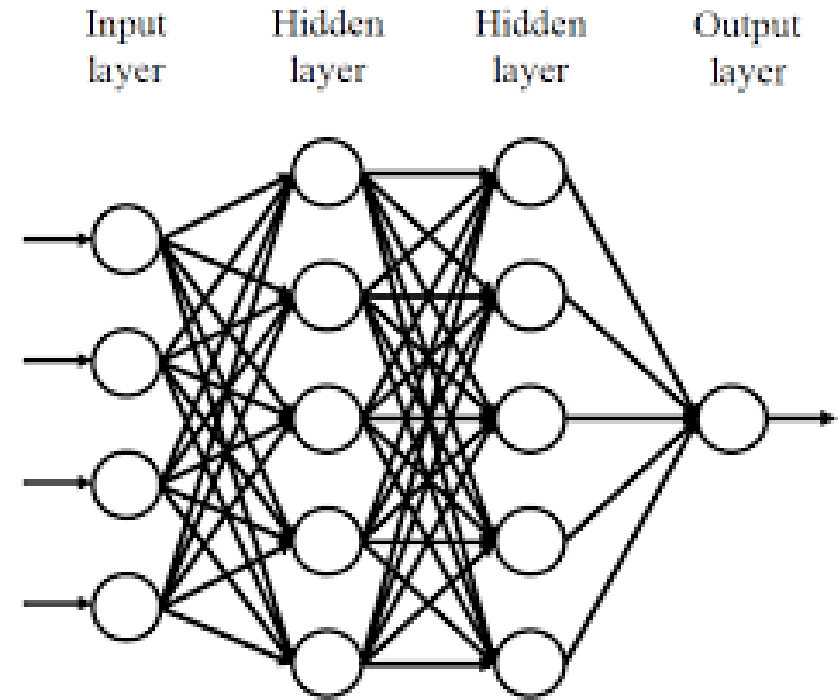
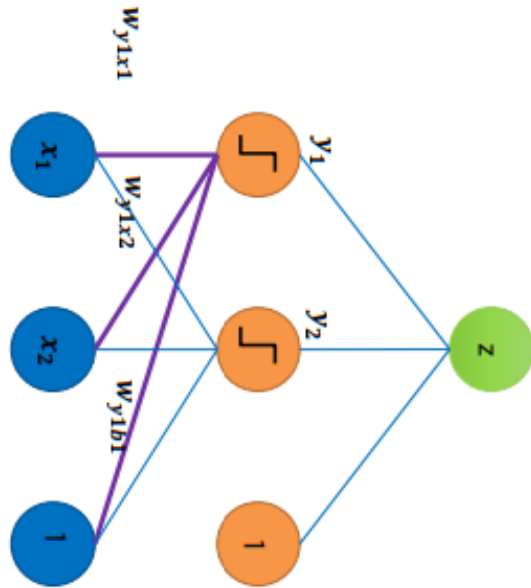
- 어 이거라면?

# Multi-layer Perceptron



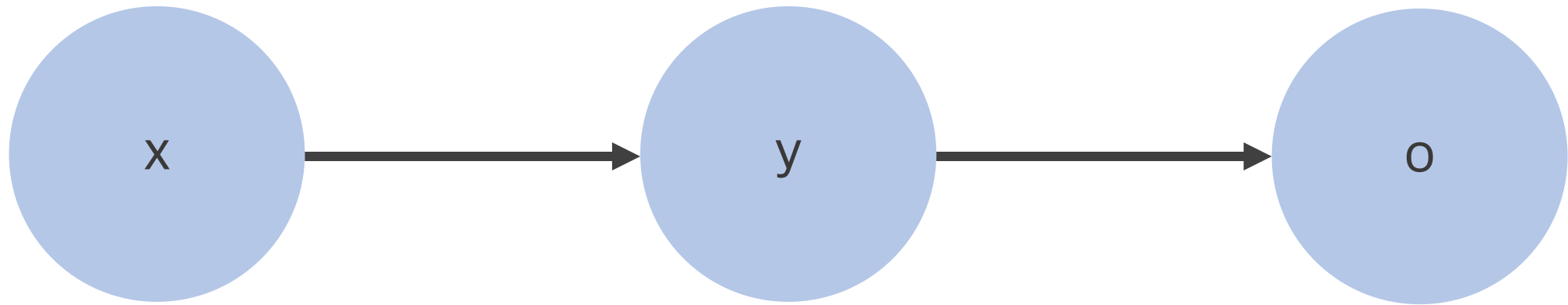
- 이게 왜 멀티 레이어지...?

# Feed-forward Neural Network



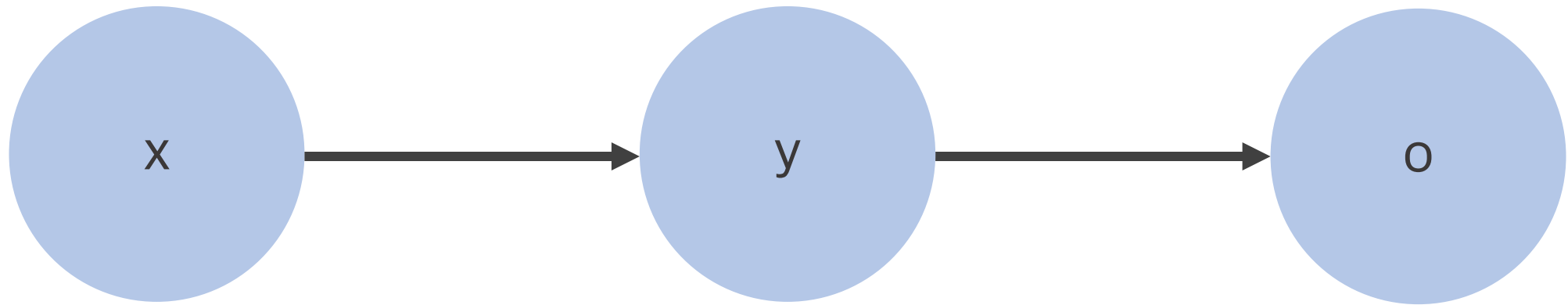
- 여기서 히든 레이어가 2개 이상이라면? 딥~러닝
- 실제로는 히든 레이어와 출력 레이어에는 활성화 함수를 적용시킵니다

# *Feed-forward Neural Network*



- 하나의 층을 가진 일렬의 구성이라 생각해 봅시다
- $y = ax + b$ ,  $o = cy + d$
- $o = c(ax + b) + d$ ,  $o = (ac)x + (bc + d)$
- $o = a'x + b'$

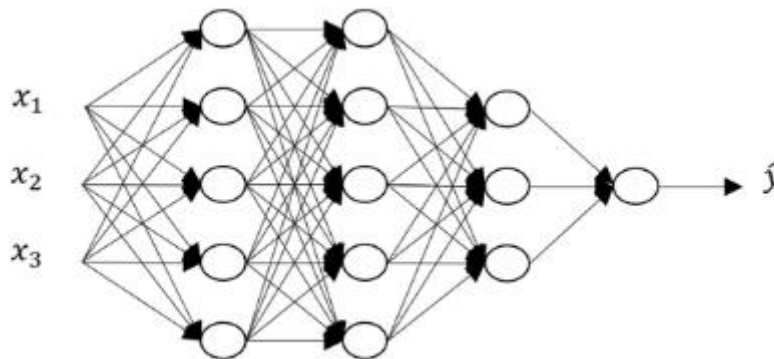
# *Feed-forward Neural Network*



- $o = a'x + b'$
- 마치 레이어가 하나 인 것과 동일한 효과만 나타냅니다
- 그렇기에 비선형 함수, 활성화 함수를 각 레이어마다 적용시킵니다
- ReLU, 시그모이드, tanh...

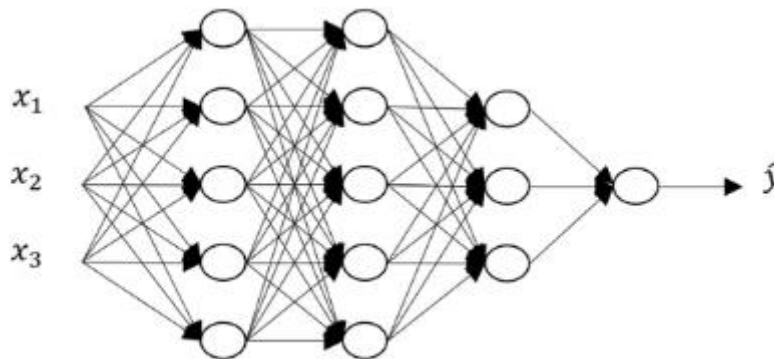


# 딥러닝으로 추론하기 - *FF*



- 행렬로 생각해 봅시다
- $\mathbf{z}^{[l]} = \mathbf{w}^{[l]} \mathbf{a}^{[l-1]} + \mathbf{b}^{[l]}$
- $\mathbf{a}^{[l]} = \mathbf{g}^{[l]}(\mathbf{z}^{[l]})$
- $\mathbf{g}^{[l]}$ 은 레이어  $l$ 의 활성화 함수 (ReLU, tanh 등),  $\mathbf{a}^{[0]} = \mathbf{X}$

# DNN 학습 – 파라미터와 하이퍼파라미터

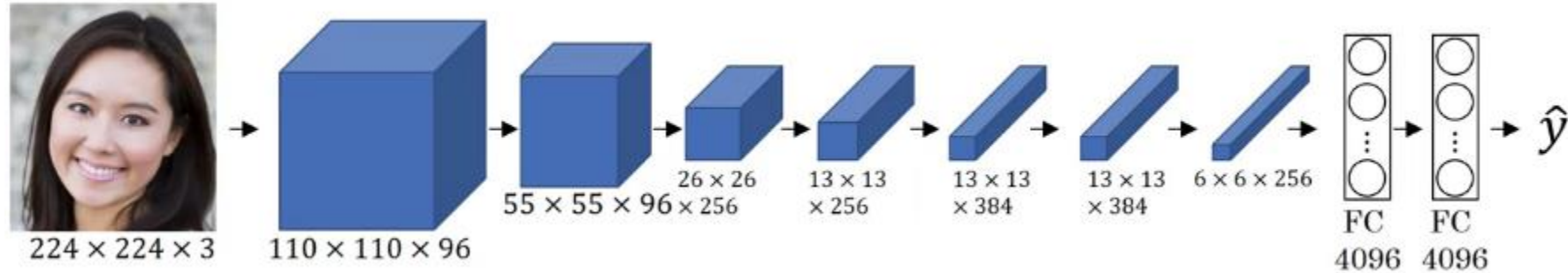


- 딥러닝을 학습시킨다는 말은 파라미터 ( $w$ ,  $a$ )를 변화시켜 예측을 잘 하도록 만든다는 의미입니다
- 데이터, 컴퓨팅 파워(GPU), 믿음
- 우리가 할 일은 모델의 하이퍼 파라미터를 잘 정해주기!
- Learning rate, 데이터 학습 수, 히든 레이어 수, 히든 레이어의 유닛 수, 활성화 함수 등등...

# *Deep Learning*을 그래서 왜 하는걸까

- 딥러닝은 머신 러닝에 비해 아키텍처가 복잡하고, 더 크고, 무겁습니다
- 더 많은 파라미터, 더 많은 학습 시킬 것
- 따라서? 더 좋은 컴퓨터, 더 많은 데이터를 요구합니다
- 그러나 더 좋은 성능을 보여줍니다
- 이미지 인식, 무작위 이미지 생성, 음성 번역, chatGPT..?
- 성능이 왜 좋을까요?
- 데이터가 충분할 경우 무엇을 학습 했는지 알아보시다

# Deep Learning

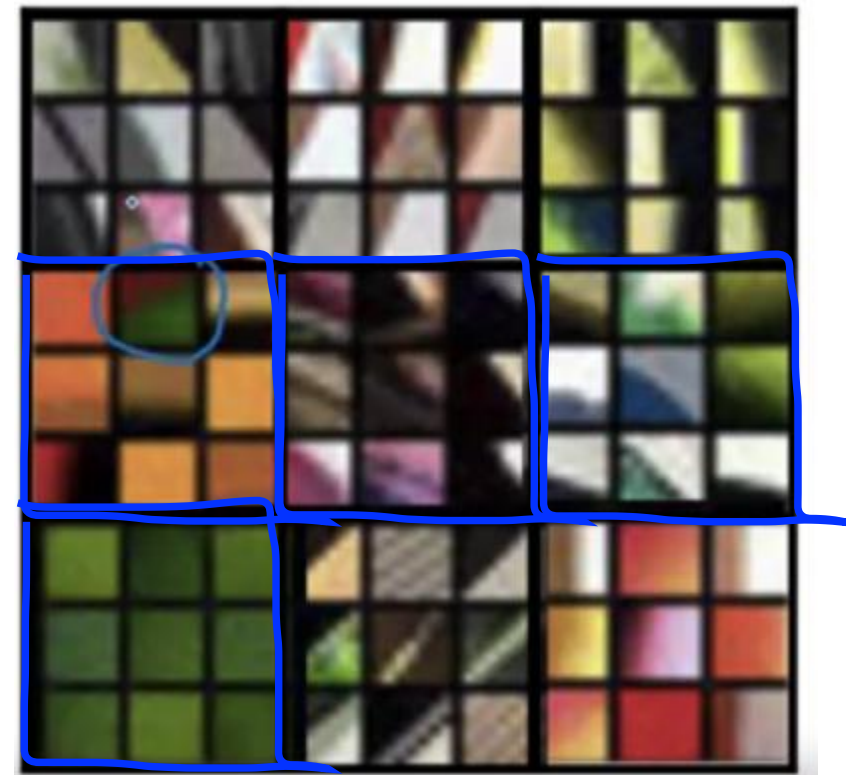


- 이미지 인식에 많이 사용하는 CNN을 예시로 알아보시다
- 여러 레이어를 통과시켜 이미지를 학습하는구나~ 정도로만 생각하셔도 충분합니다
- 각 레이어에선 어떤 정보가 학습될까요?
- 이런 저런 입력을 넣어보며, 각 유닛의 활성이 최대화되는 이미지를 통해 알아보시다

# Deep Learning



- 일부 유닛의 최대화 이미지 사진이다
- 대각선 요소를 인식하는구나~를 알 수 있다



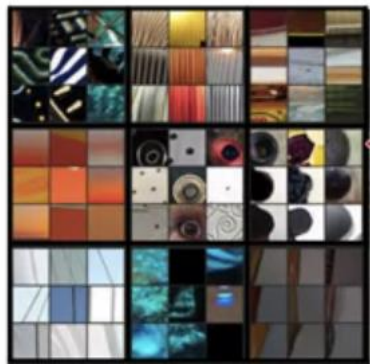
각 퍼셉트론들의 특징이 나타나는 것을 볼 수 있다

- 오른쪽 더 많은 유닛 정보에서 수직선 검출, 주황색 검출, 오른쪽 대각선 검출 등을 하는 유닛을 볼 수 있다

# Deep Learning



Layer 1



Layer 2



Layer 3



Layer 4




Layer 5

- 각 레이어를 통과함에 따라 정보를 더 압축하고, 더 많은 사실을 뽑아낸다
- 단순한 선과 색을 넘어 복잡한 형태와 객체를 인식, 구분 할 수 있다
- 레이어가 늘어난다면? 더 복잡한 정보를 학습 가능하겠죠
- 물론 그만큼 더 많은 데이터가 필요합니다

# 여러분들이 문득 떠올리실 의문점

- Machine Learning은 그럼 어디다 쓰지..?
- 성능도 안 나오는거 같고...
- DL은 많은 데이터와 컴퓨팅 파워를 요구합니다
- 준서님 80% 성능 나오는 모델 제작해 주세요
  - > 아 최신 기술 써야해서 5000만원 정도 필요 할 것 같습니다!
- 데이터가 적거나, 성능이 중요하지 않거나, 컴퓨팅 파워가 낮거나 등등
- 필요한 수준과 성능에 맞춰 사용합니다

## *Next lecture*

- 확률 vs 가능성
- 좋은 모델이란 뭘까?
- 어떻게 모델을 학습시키지?
- 수식이 부족해서 아쉽네..
- 다음시간부터 알아보도록 합시다~
- 



# QnA

- 질문 있으신가요?

gpu는 행렬연산을 잘해주는 분이다