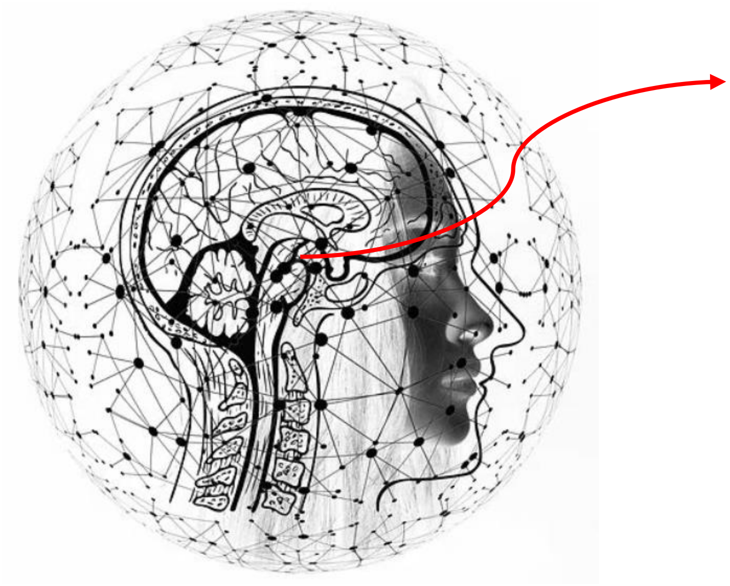
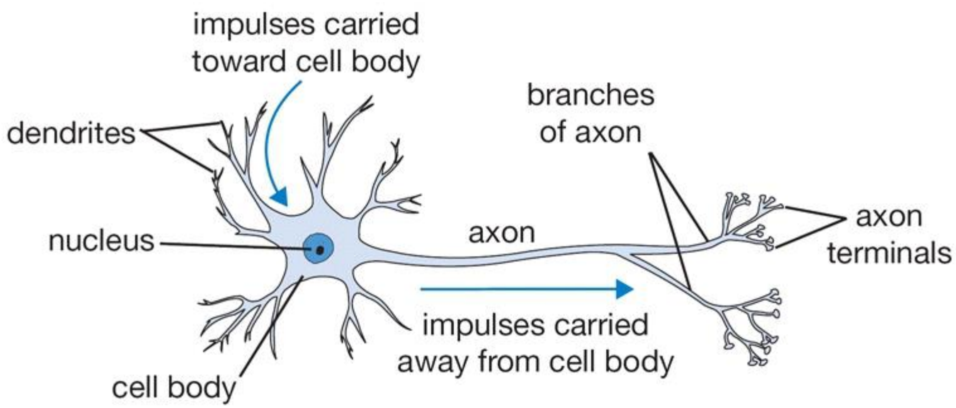
인공지능개론

뉴럴네트워크

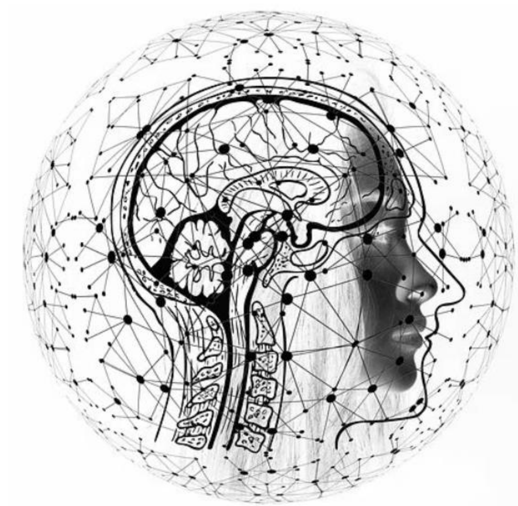
Neural Networks

- A neural network is a mathematical model inspired by **biological neural networks**
  - Intelligence comes from their connection weights
  - Connection weights are decided by learning or adaptation



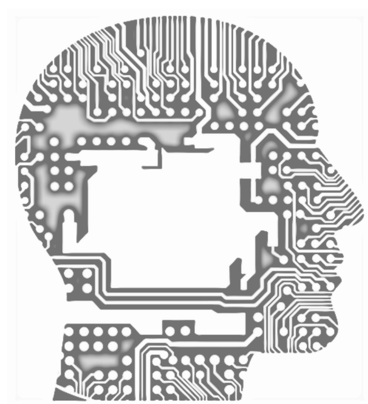


Neural Networks



Category
Dog
Cat
Pig
Chicken
Eagle
Bug
Dragon

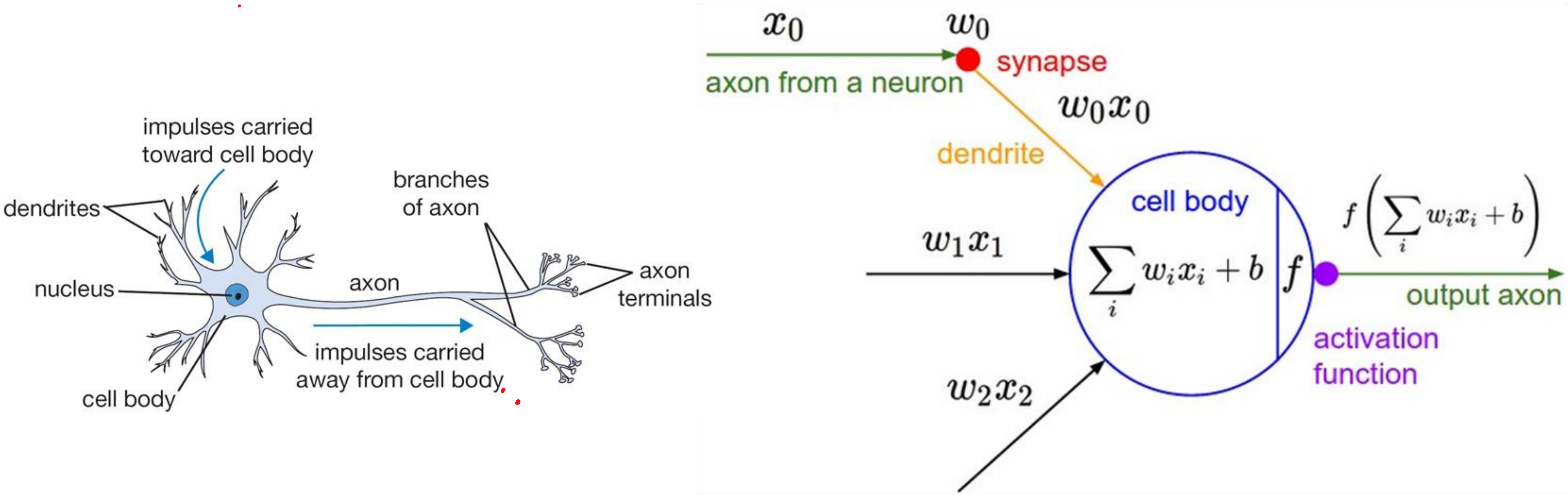
96	**255**	**7**	**45**
4	0	8	27
186	102	85	86
211	45	37	189



Neural Networks

A neural network is a mathematical model inspired by **biological neural networks**

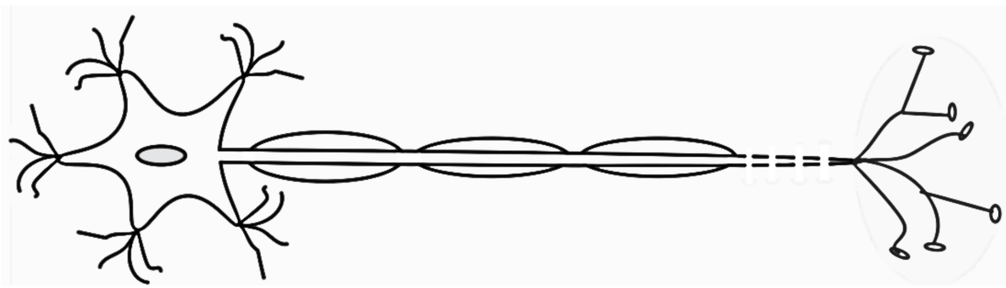
- Intelligence comes from their connection weights
- Connection weights are decided by learning or adaptation

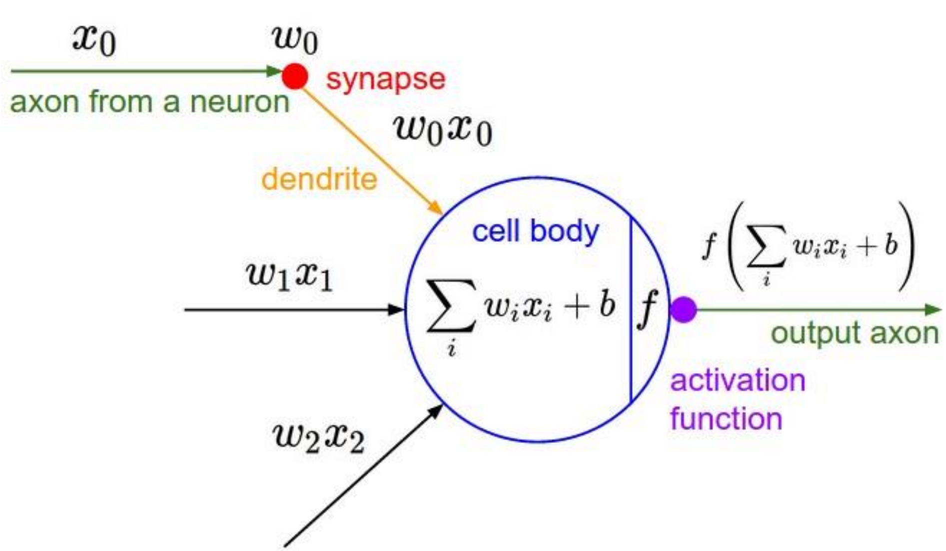


출처: http://cs231n.github.io/neural-networks-1/

Neural Networks

Single neuron model



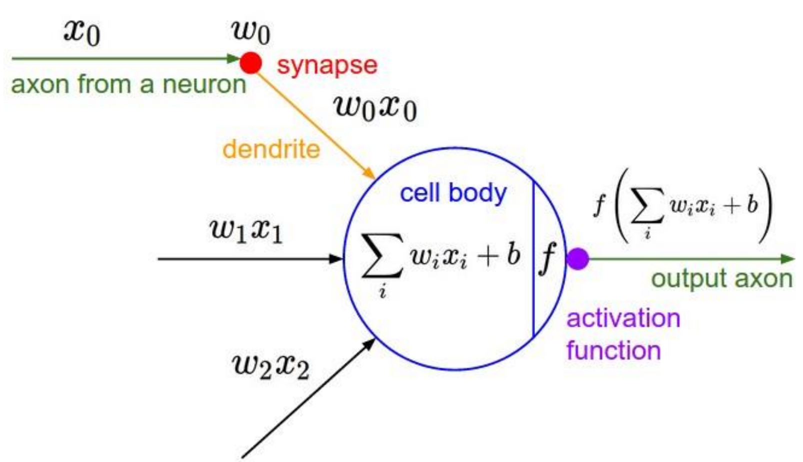


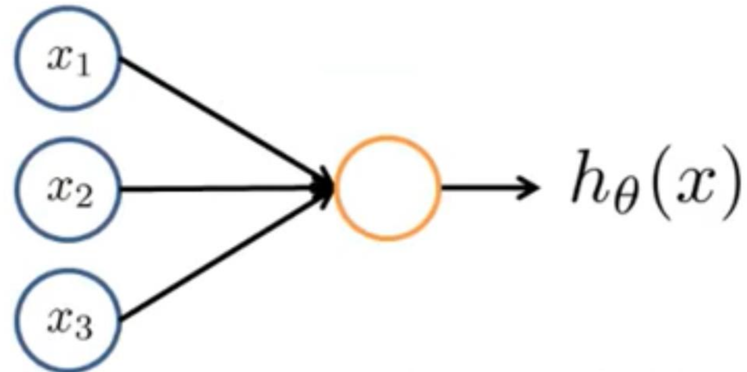
- Input
- Weight
- Threshold (Activation)

Our brain do

Neural Networks

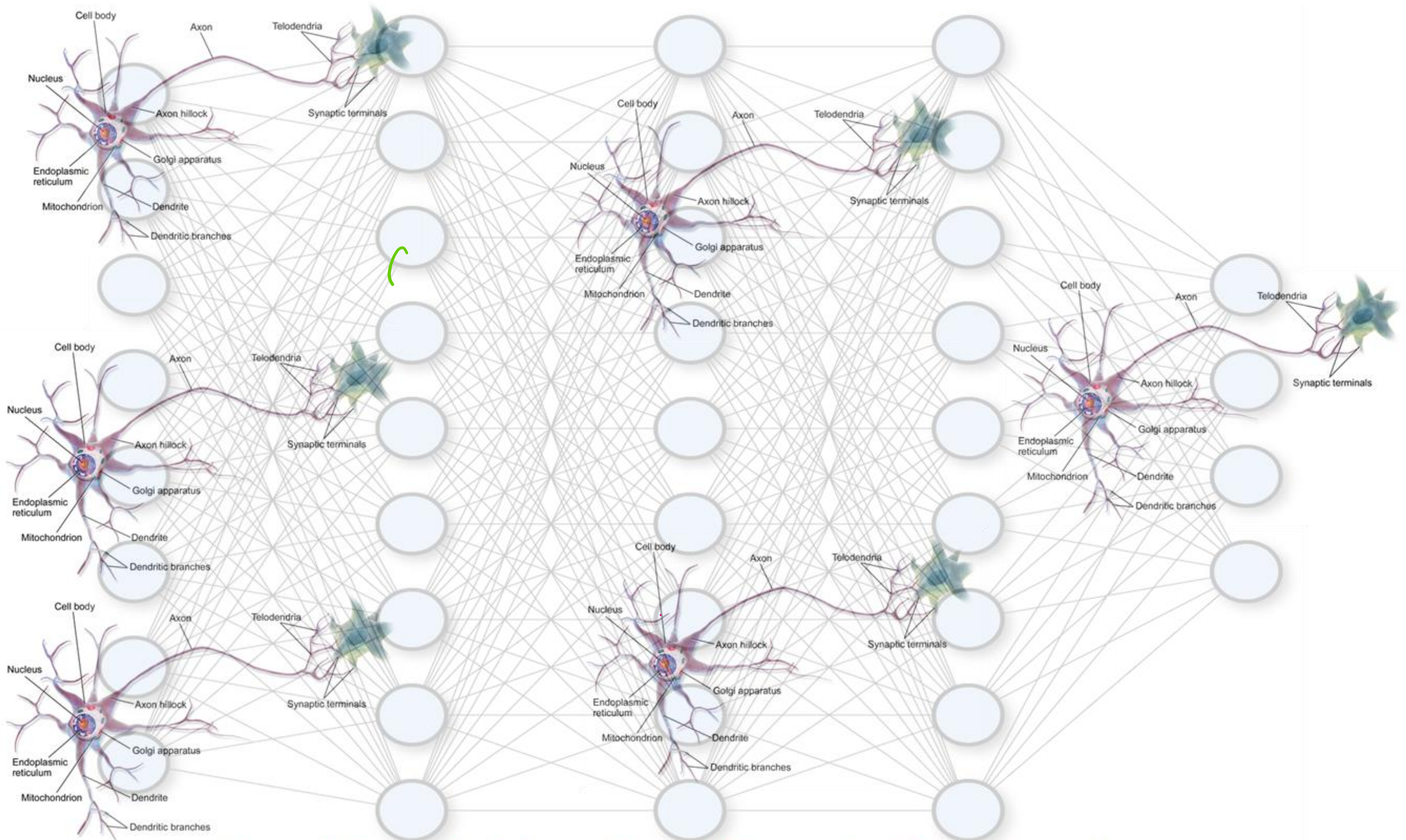
Single neuron model: Input / Weight / Activation / Bias





$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

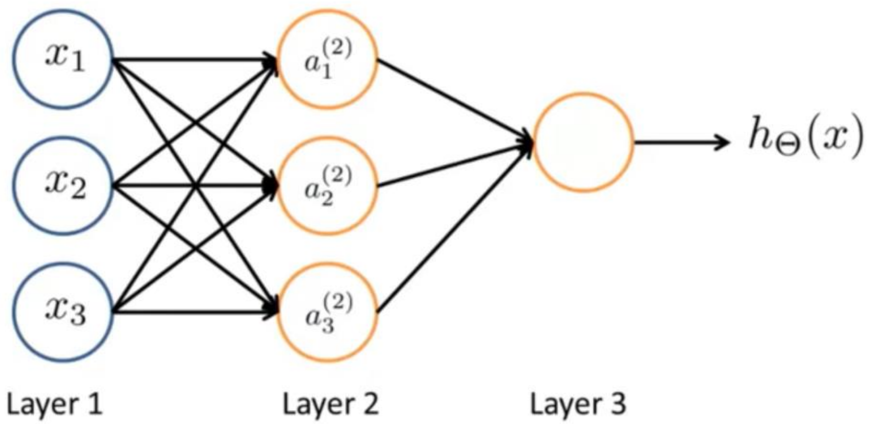
Neural Networks



Neural Network →Deep Neural Network →Deep Learning

Neural Networks

Neural networks: Input layer / hidden layer / output layer / Bias



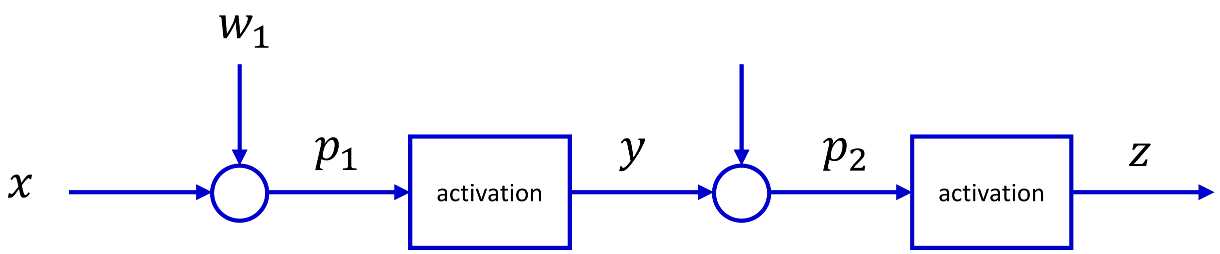
$a_i^{(j)}$ = "activation" of unit i in layer j

$\Theta^{(j)}$ = matrix of weights controlling function mapping from layer j to layer $j+1$

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

Neural Networks

Performance function P



$$\frac{\partial P}{\partial w_2} = \frac{\partial P}{\partial z}\frac{\partial z}{\partial w_2} = \frac{\partial P}{\partial z}\frac{\partial z}{\partial p_2}\frac{\partial p_2}{\partial w_2}$$

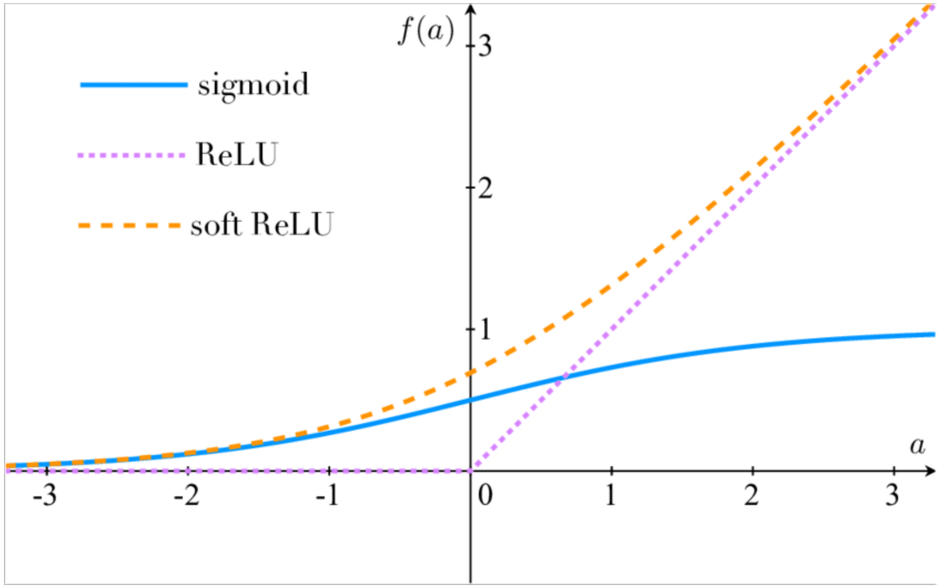
$$\frac{\partial P}{\partial w_1} = \frac{\partial P}{\partial z}\frac{\partial z}{\partial p_2}\frac{\partial p_2}{\partial y}\frac{\partial y}{\partial p_1}\frac{\partial w_1}{\partial w_1}$$

$\beta = \beta(1-\beta)$ where $\beta = \frac{1}{1+e^{-\alpha}}$

Neural Networks

Activation function

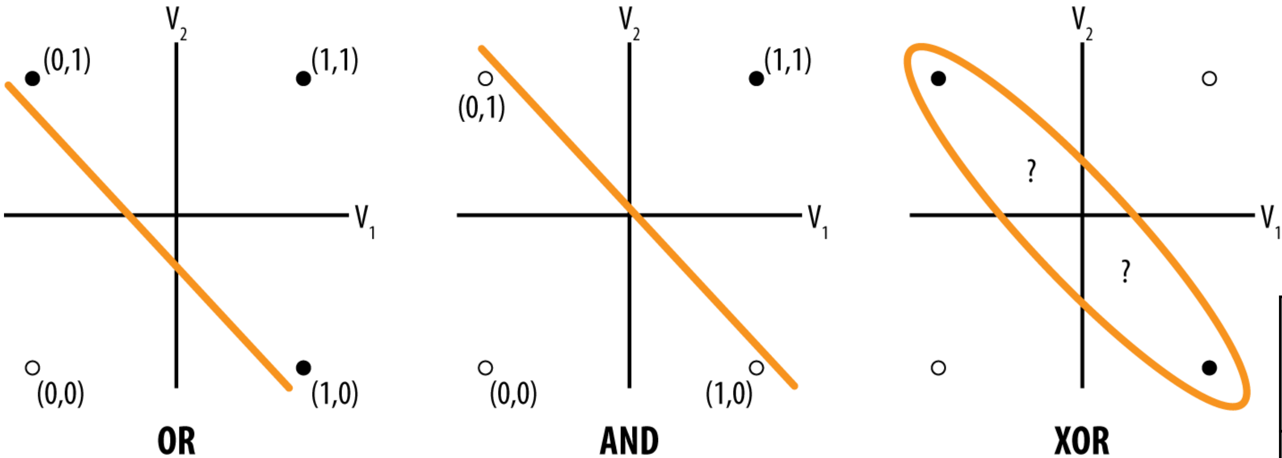
- Sigmoid
- ReLu
- Soft ReLu



[Chao Zhang, 2017]

Neural Networks

- Neural Network
  - Learn nonlinear model



Linear can solve

Linear can solve

Linear cannot

XOR logic

X1 (input1)	X2 (input2)	Y (Output)
0	0	0
0	1	1
1	0	1
1	1	0