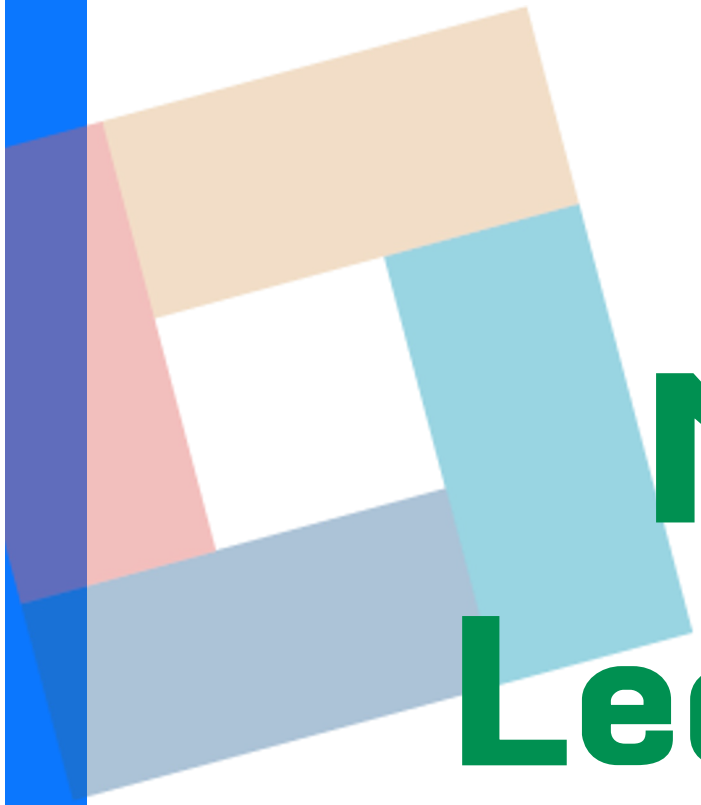


Machine Learning 05

Kihyun Shin
DMSE, HBNU

TheMAC

Feature scaling

Feature and parameter values

$\widehat{price} = w_1 x_1 + w_2 x_2 + b$

x_1: size (feet^2)

range: 300 - 2,000

large

x_2: # bedrooms

range: 0 - 5

small

size

bedrooms

House: x_1= 2,000, x_2=5, price = $500k

Size of the parameters w_1, w_2 ???

w_1=50, w_2=0.1, b=50,

$\widehat{price} = 50 * 2,000 + 0.1 * 5 + 50$

$\widehat{price}$ = $100,050.5k

w_1=0.1, w_2=50, b=50,

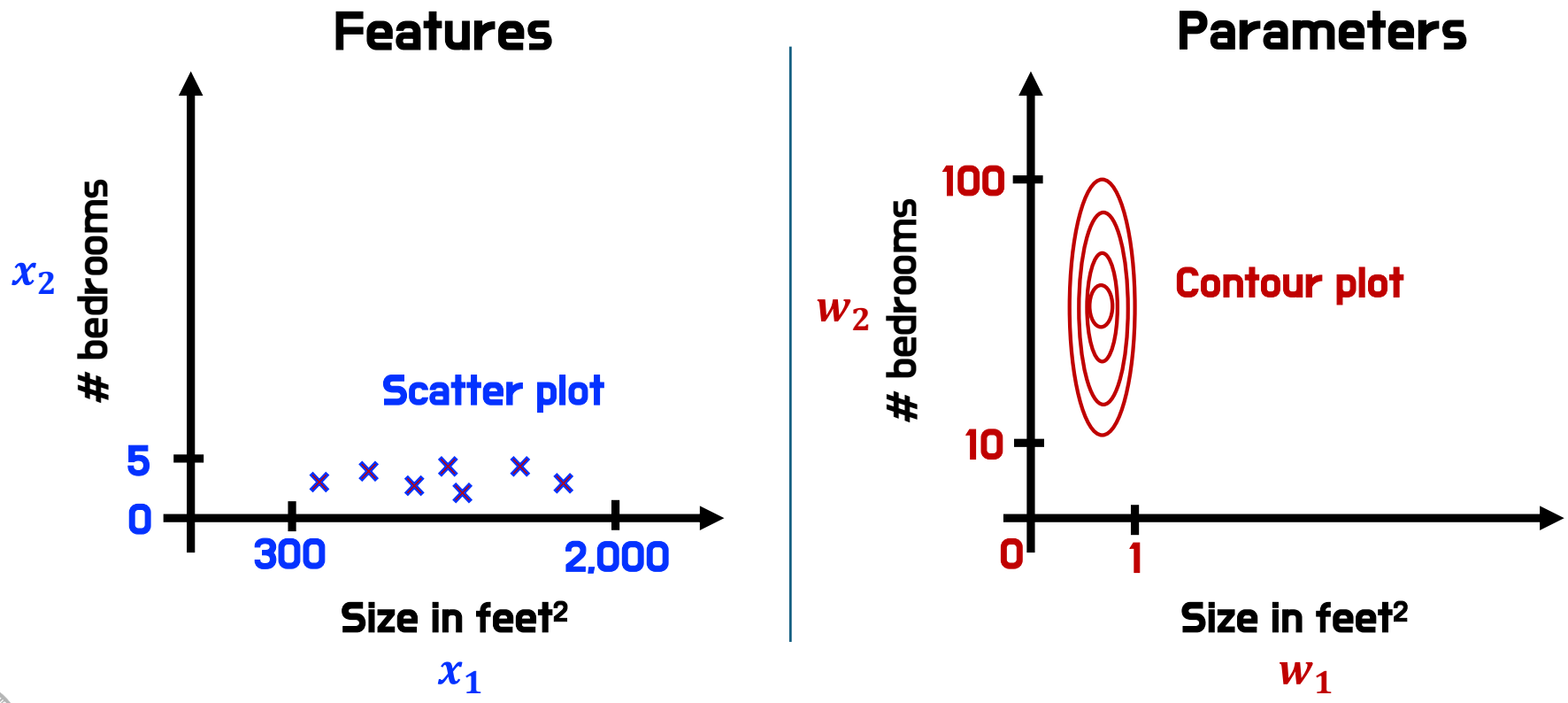
small large

$\widehat{price} = 0.1 * 2,000 + 50 * 5 + 50$

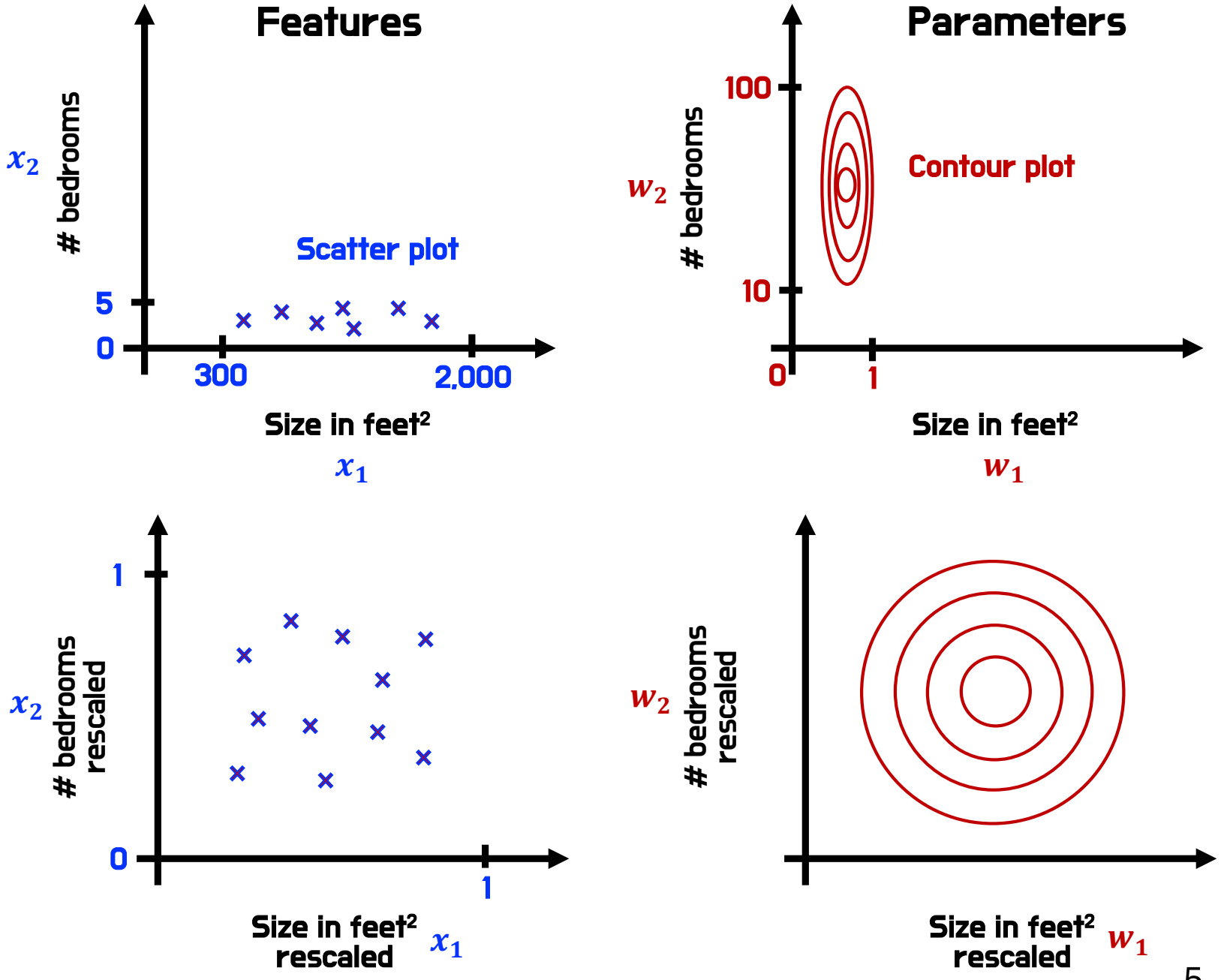
$\widehat{price}$ = $500k More reasonable

Feature size and parameter size

	Size of feature x_j	Size of parameter w_j
Size in feet^2	⟷	↔
# bedrooms	↔	⟷

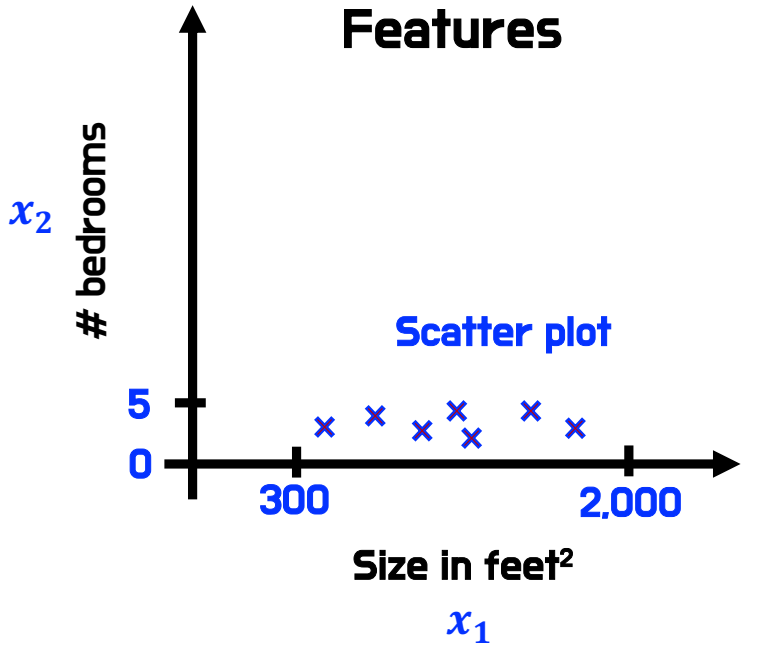


Feature size and gradient descent



TheMAC

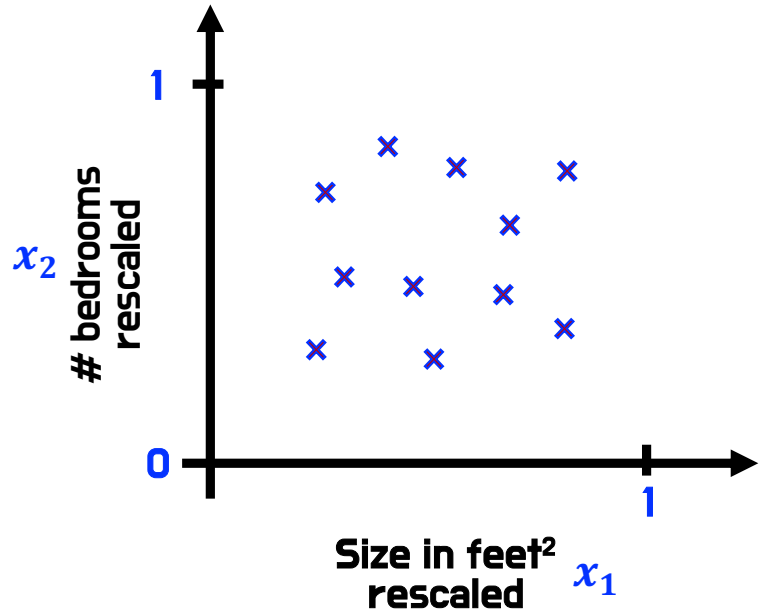
Feature scaling



$300 \leq x_1 \leq 2{,}000$ $\quad$ $0 \leq x_2 \leq 5$

$$x_{1,scaled} = \frac{x_1}{2,000} \quad x_{2,scaled} = \frac{x_2}{5}$$

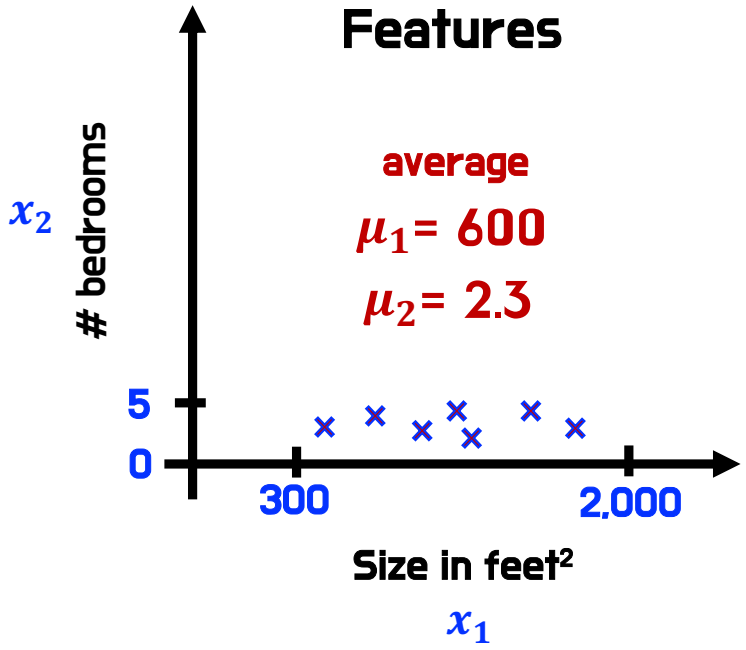
Max Max



$0.15 \leq x_{1,rescaled} \leq 1$ $\quad$ $0 \leq x_{2,rescaled} \leq 1$

TheMAC

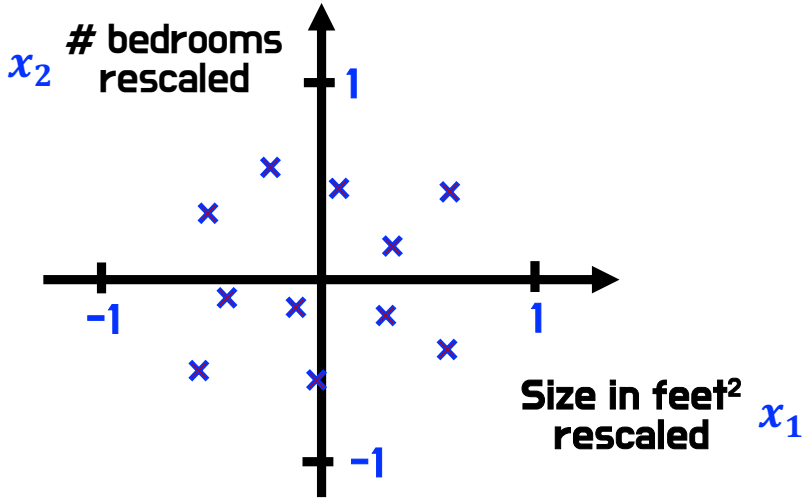
Mean normalization



$300 \leq x_1 \leq 2{,}000$ $\quad$ $0 \leq x_2 \leq 5$

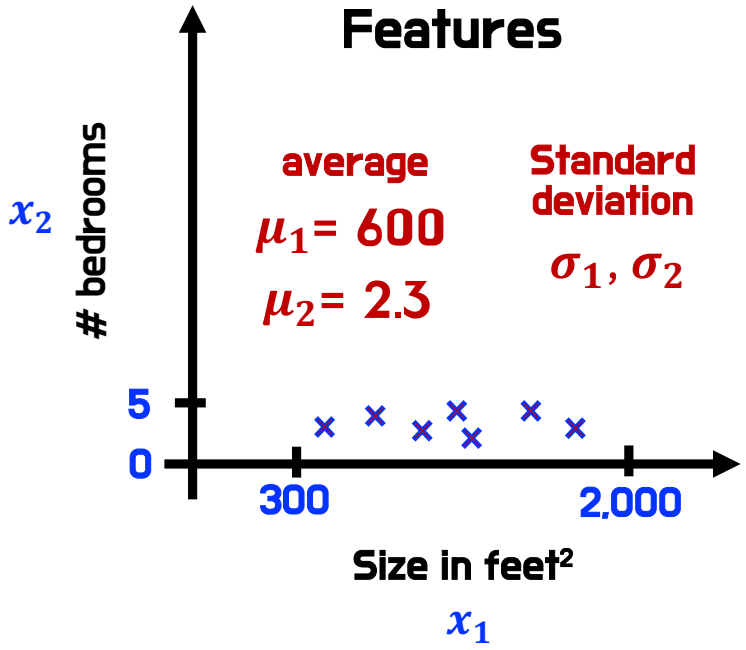
$$x_1 = \frac{x_1 - \mu_1}{2{,}000 - 300} \qquad x_2 = \frac{x_2 - \mu_2}{5 - 0}$$

Max Min $\quad$ Max Min



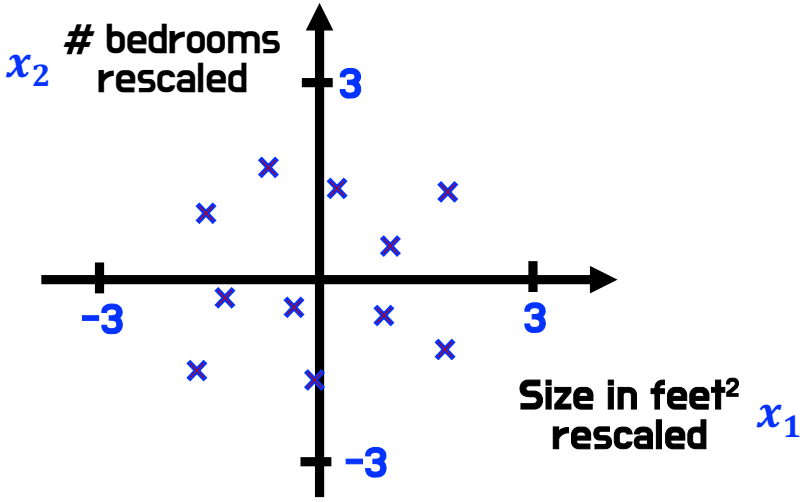
$-0.18 \leq x_1 \leq 0.82$ $\quad$ $-0.46 \leq x_2 \leq 0.54$

Z-score normalization



$300 \leq x_1 \leq 2{,}000$ $\quad$ $0 \leq x_2 \leq 5$

$$x_1 = \frac{x_1 - \mu_1}{\sigma_1} \qquad x_2 = \frac{x_2 - \mu_2}{\sigma_2}$$



$-0.67 \leq x_1 \leq 3.1$ $\quad$ $-1.6 \leq x_2 \leq 1.9$

TheMAC

Feature scaling

Aim for about $-1 \leq x_j \leq 1$ for each feature x_j

$-3 \leq x_j \leq 3$

$-0.3 \leq x_j \leq 0.3$

Acceptable ranges

$0 \leq x_1 \leq 3$	**Okay, no rescaling**
$-2 \leq x_2 \leq 0.5$	**Okay, no rescaling**
$-100 \leq x_3 \leq 100$	**Too large → rescale**
$-0.001 \leq x_4 \leq 0.001$	**Too large → rescale**
$98.6 \leq x_5 \leq 105$	**Too large → rescale**

Checking gradient descent
For convergence

Gradient descent

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b}[\frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(x^{(i)}\right) - y^{(i)}\right)^2]$$

$$w_j = w_j - \alpha\frac{\partial}{\partial w_j}J(\vec{w},b) \longrightarrow \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$

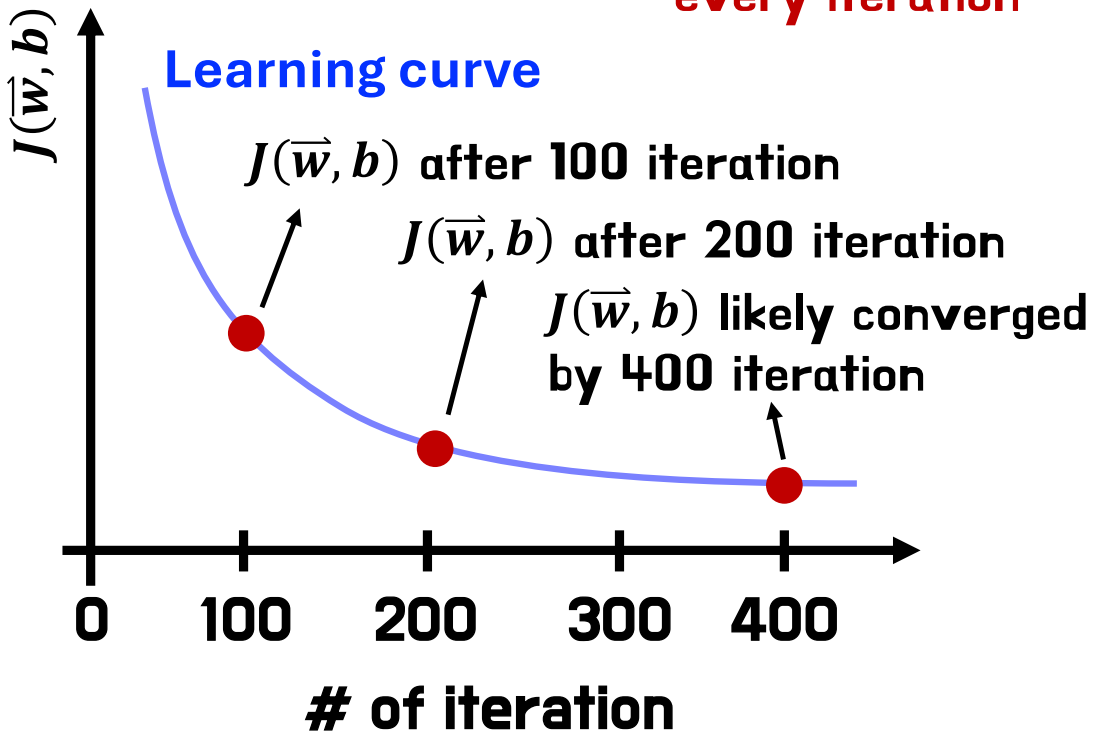
$$b = b - \alpha\frac{\partial}{\partial b}J(\vec{w},b) \longrightarrow \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)$$

Gradient descent

Make sure gradient descent is working correctly

Objective: $\min_{\vec{w},b} J(\vec{w},b)$

$J(\vec{w},b)$ should decrease after every iteration



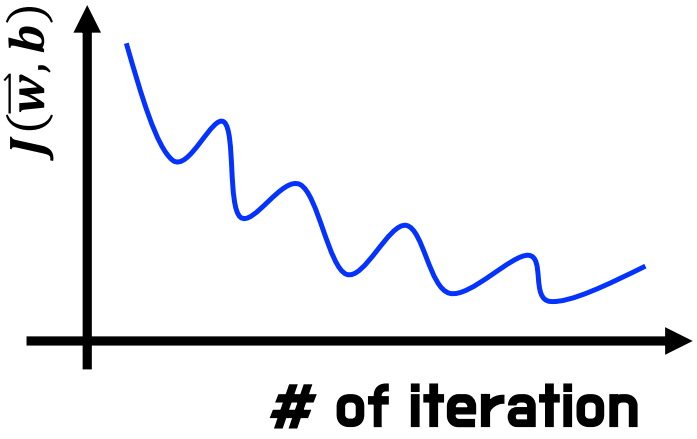
Automatic convergence test
Let ε"epsilon" be 10^{-3}

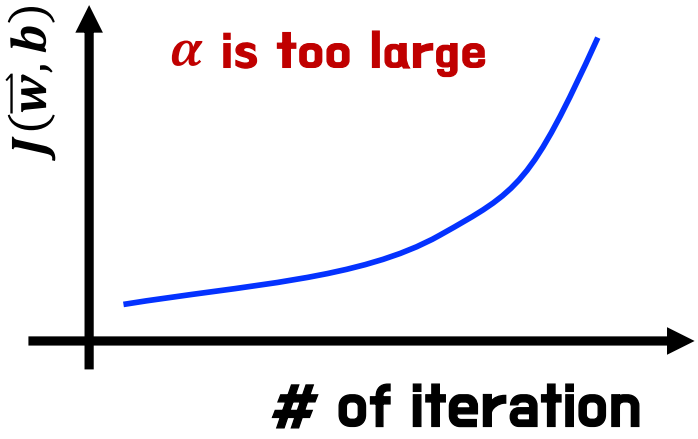
If $J(\vec{w},b)$ decrease by $\leq \varepsilon$ in one iteration,
Declare convergence,
(found parameters $\vec{w},b$ to get close to global minimum)

iteration needed varies
→ 30, 1,000, 100,000

Gradient descent

Identify problem with gradient descent





Debugging needed
Or
α is too large

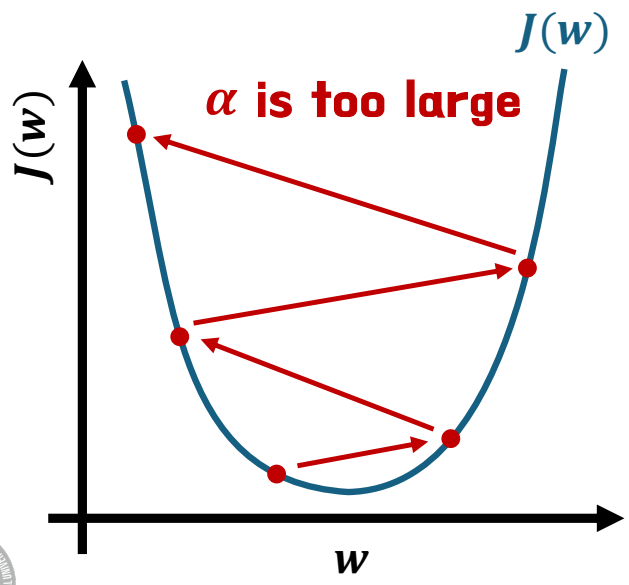
wrong

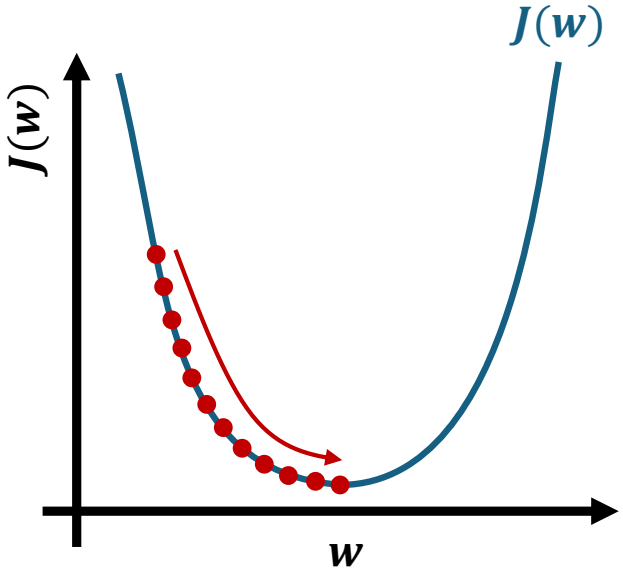
$$w_1 = w_1 + \alpha d_1$$

correct

$$w_1 = w_1 - \alpha d_1$$

Adjust learning rate





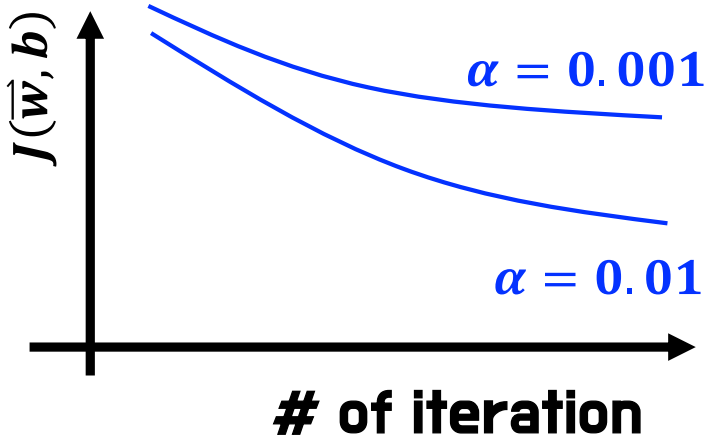
With a small enough α, $J(\vec{w}, b)$ should decrease on every iteration

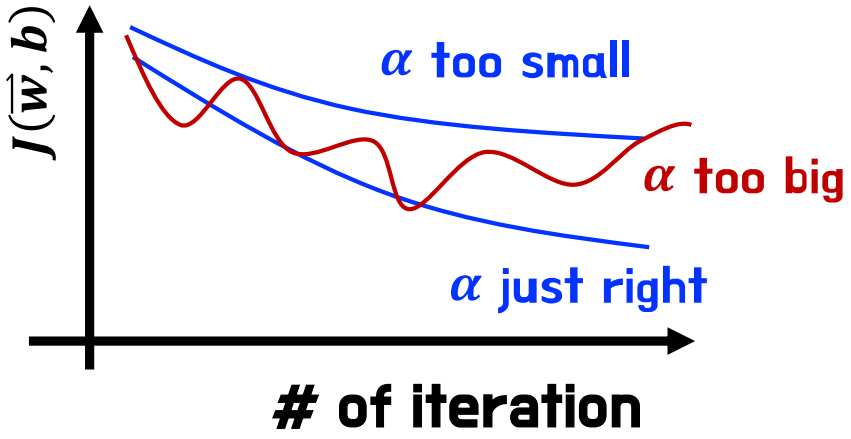
If α is too small, Gradient descent takes a lot of iteration to converge

Gradient descent

Values of α to try:

... 0.001 0.01 0.1 1 ...



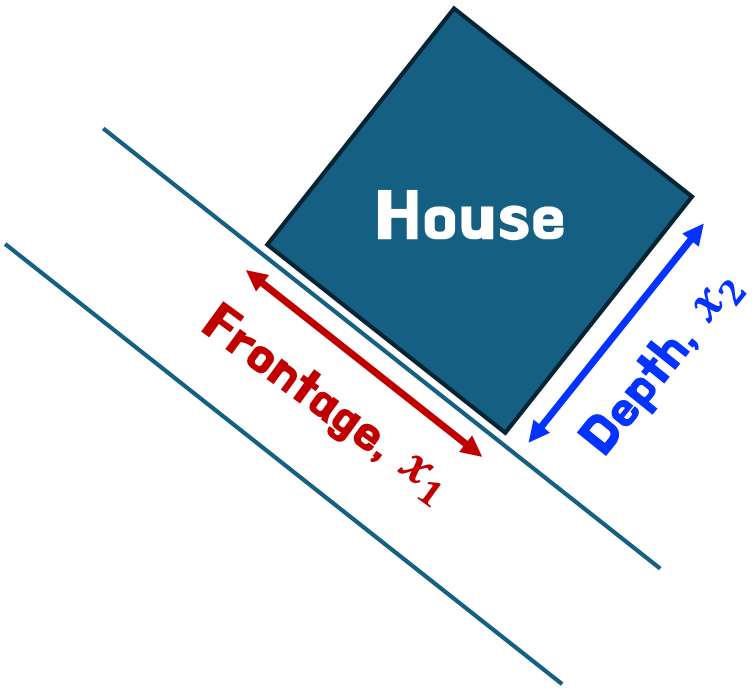


Feature engineering

Feature engineering

$$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + b$$

Frontage Depth



Area = frontage * depth

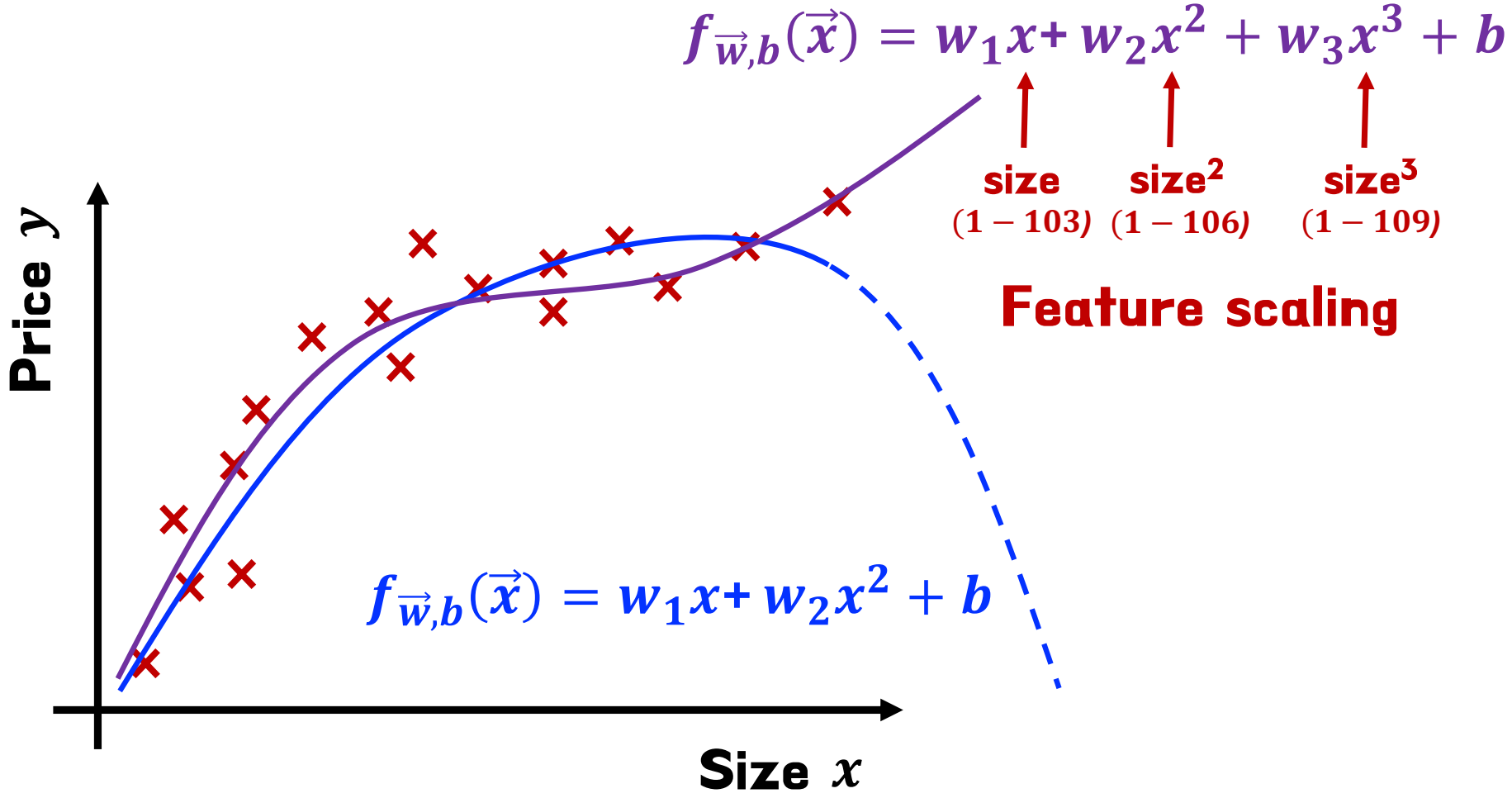
$x_3 = x_1 x_2$ **New feature**

$$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + \boldsymbol{w_3 x_3} + b$$

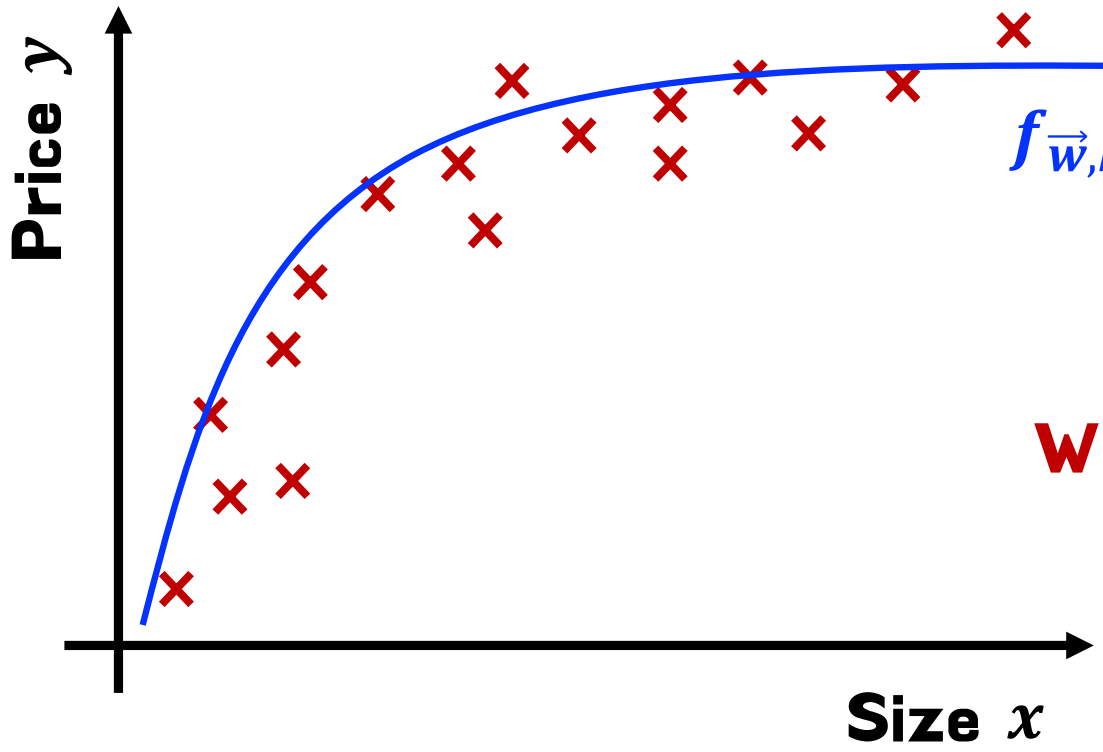
Feature engineering: using intuition to design new features, by transforming or combining original features

Polynomial regression

Polynomial regression



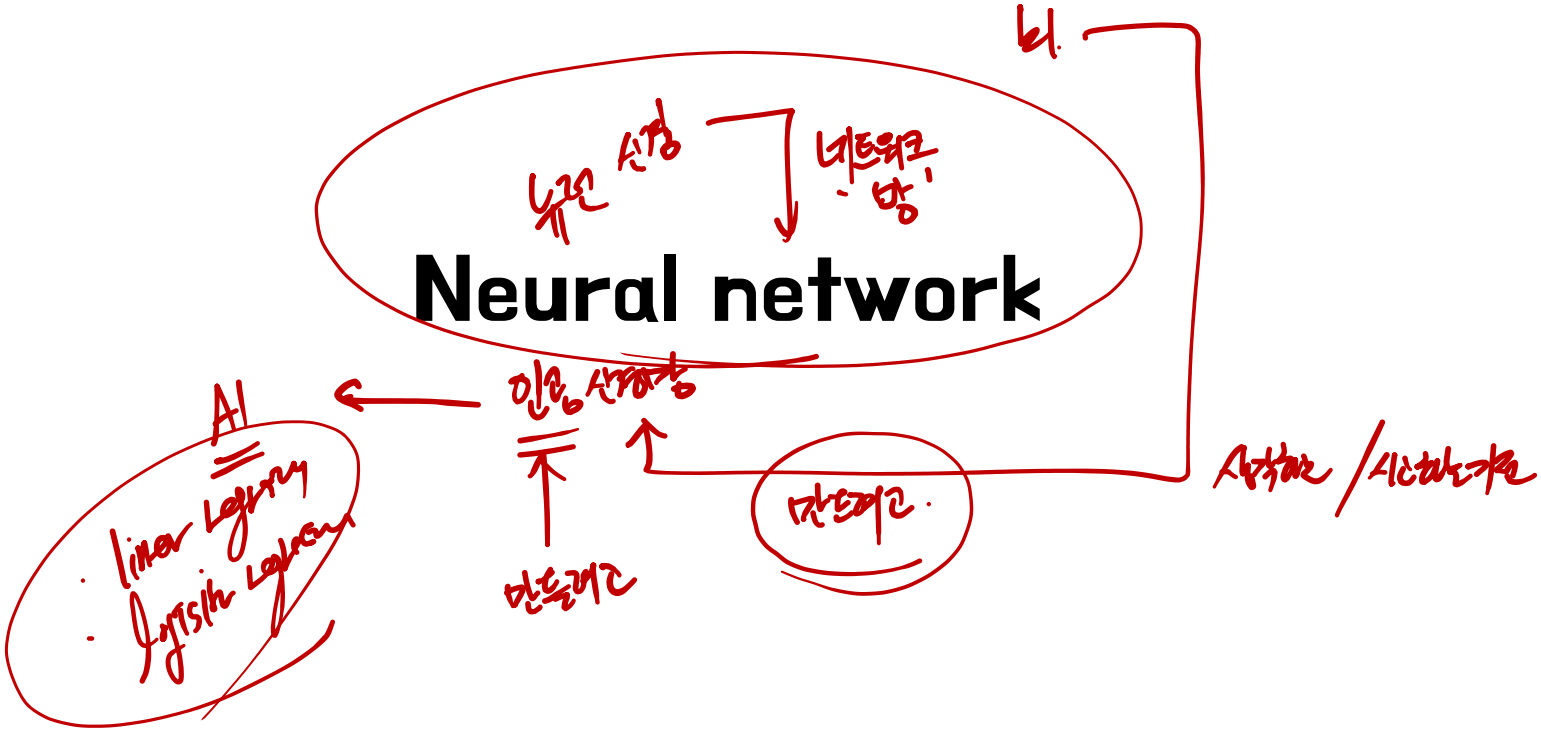
Choice of feature



$$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 \sqrt{x} + b$$

size $\sqrt{size}$

What features to use ?


Neural network

Neural networks

Origin: Algorithms that try to mimic the brain

Used in 1980's and early 1990's

→ Handwriting recognition (postal code, check)

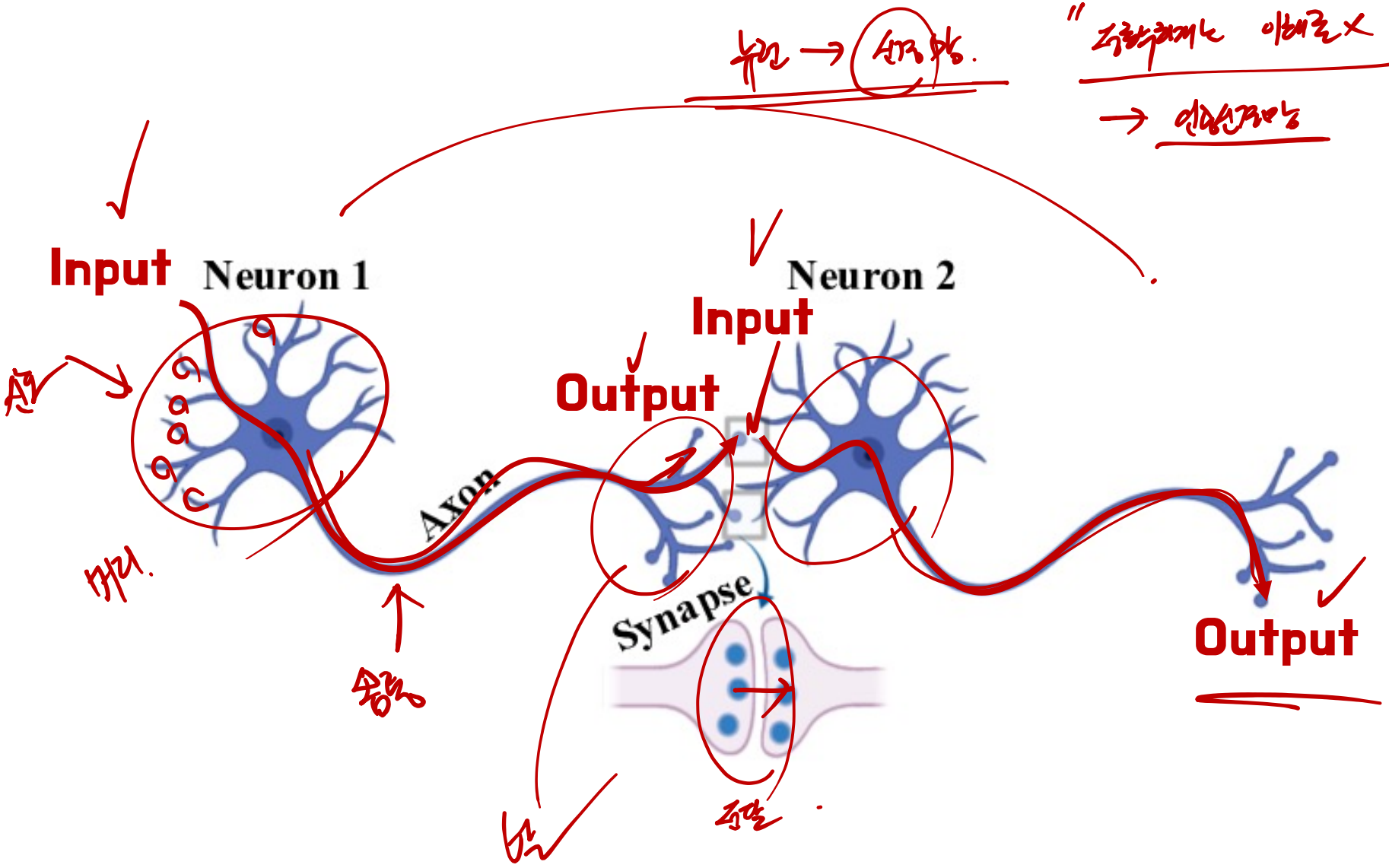
Resurgence from around 2005.

→ Deep learning

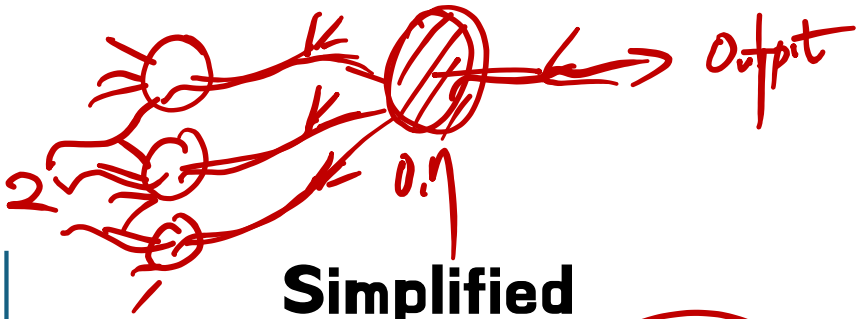
→ Speech → Images → text (NLP) → ...

Weather estimation, Chaos theory, Black hole,
Materials development, Drug discovery, Auto pilot

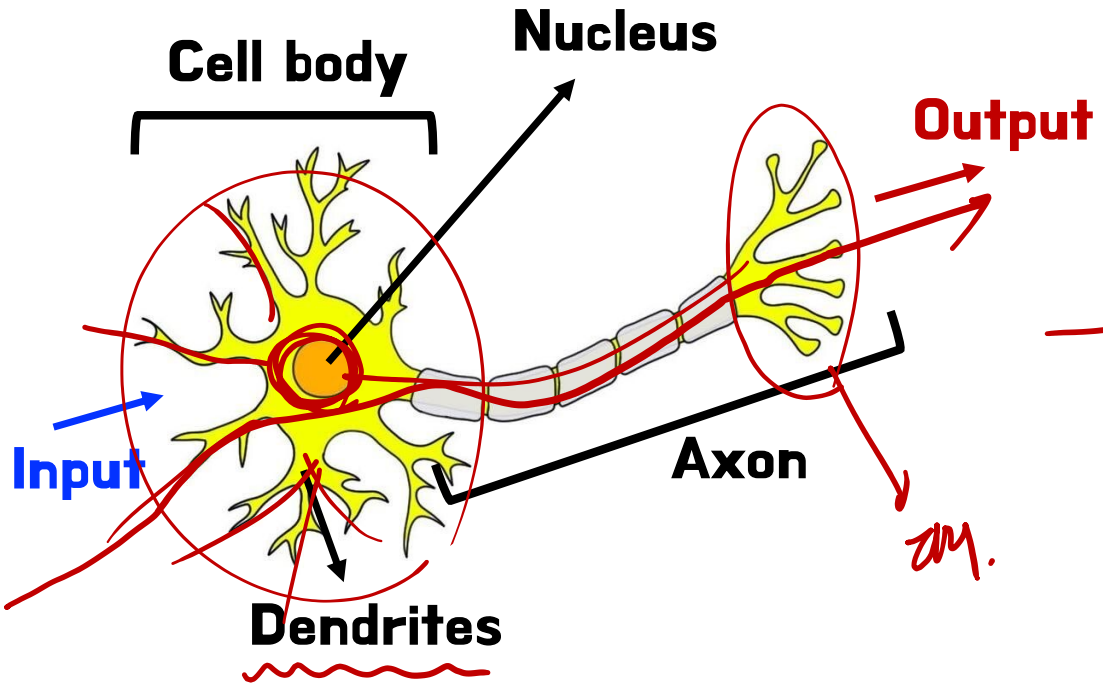
Neurons in brain



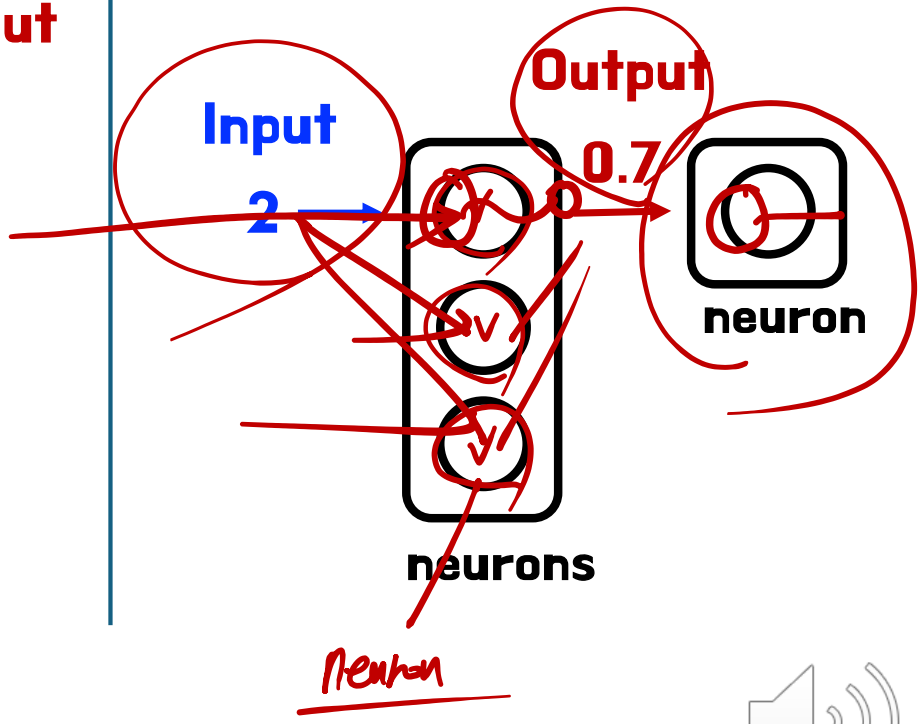
Simplified model



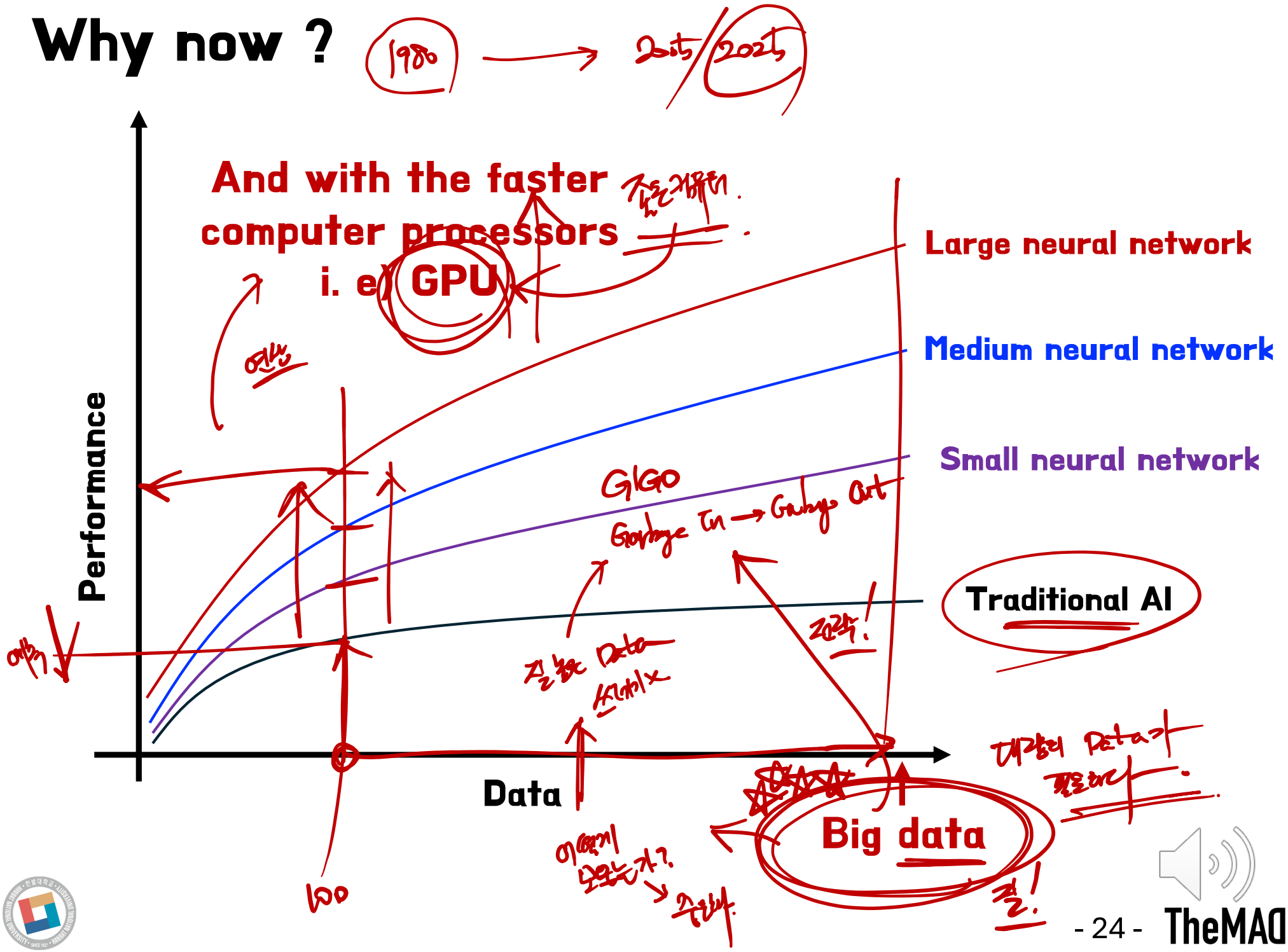
Biological neuron



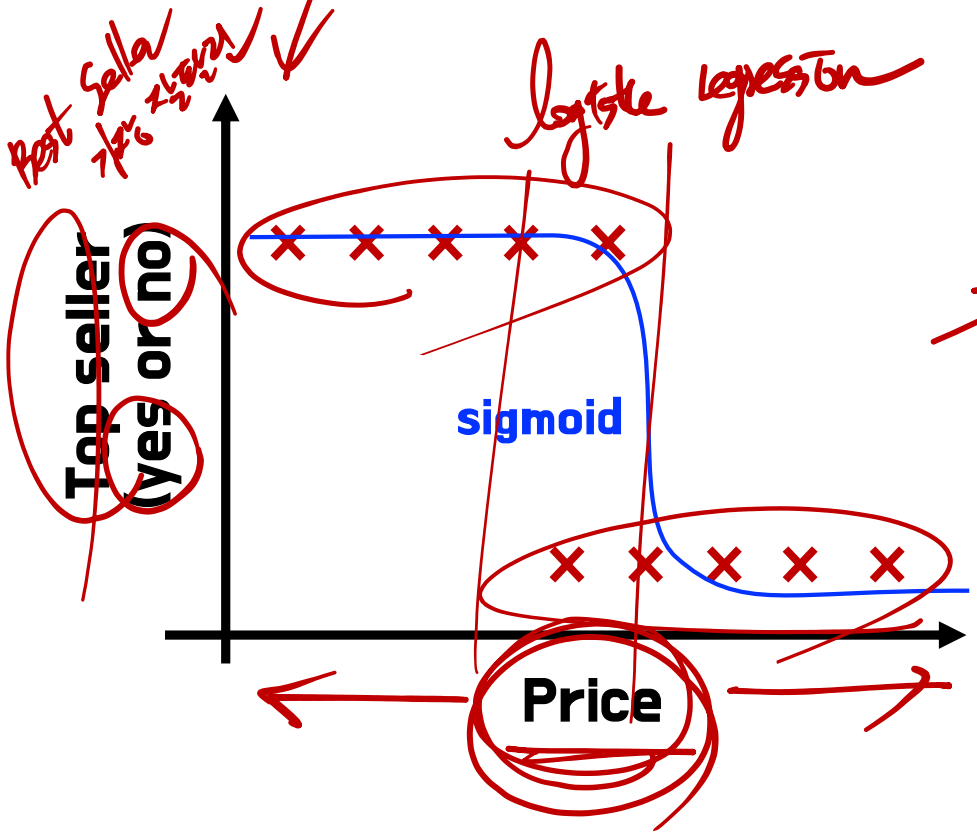
Simplified mathematical model of a neuron



Why now ?


And with the faster computer processors i. e. GPU
Large neural network
Medium neural network
Small neural network
Traditional AI
Performance
Data
GIGO
Big data

Demand prediction



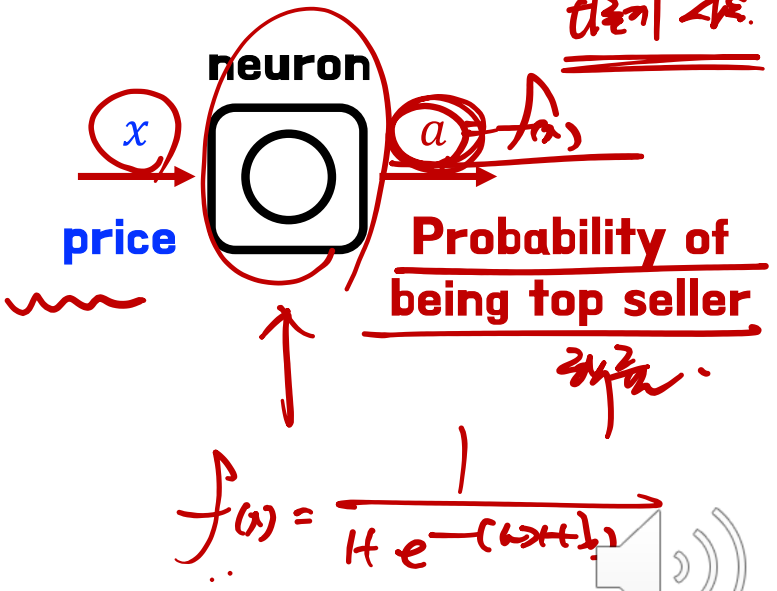
x = price

input

$$a = f(x) = \frac{1}{1 + e^{-(wx+b)}}$$

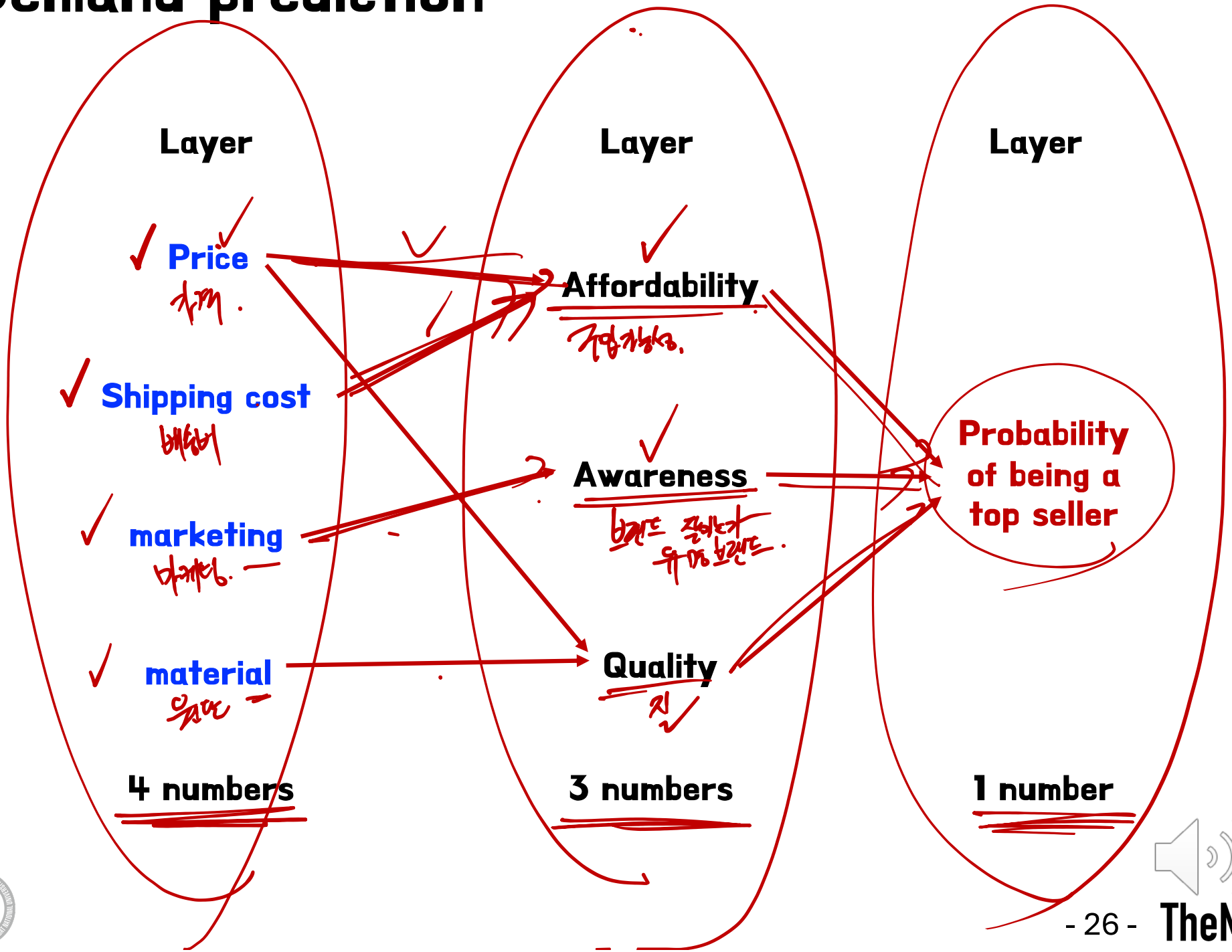
output

activation

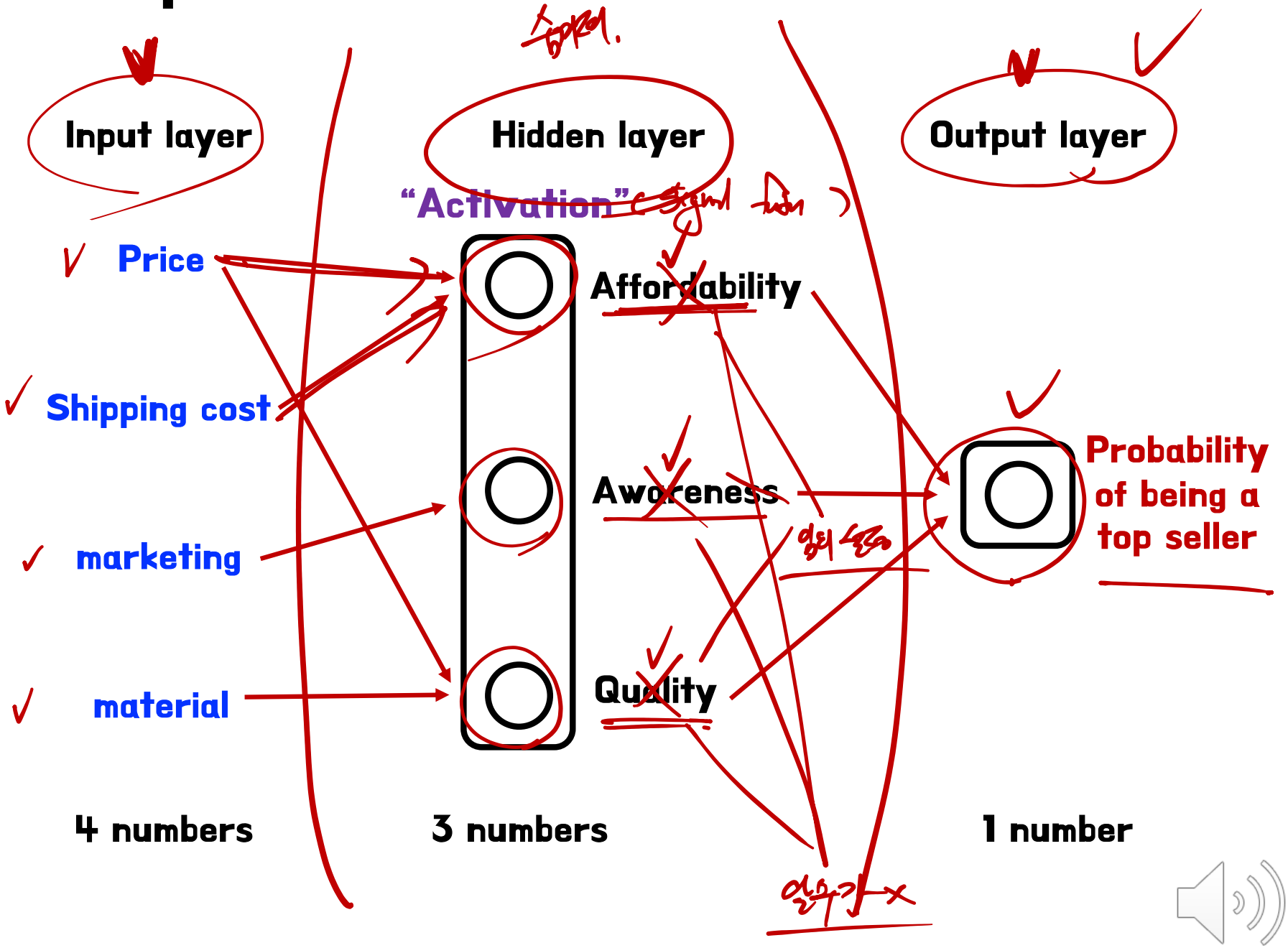


$$f(x) = \frac{1}{1 + e^{-(wx+b)}}$$

Demand prediction

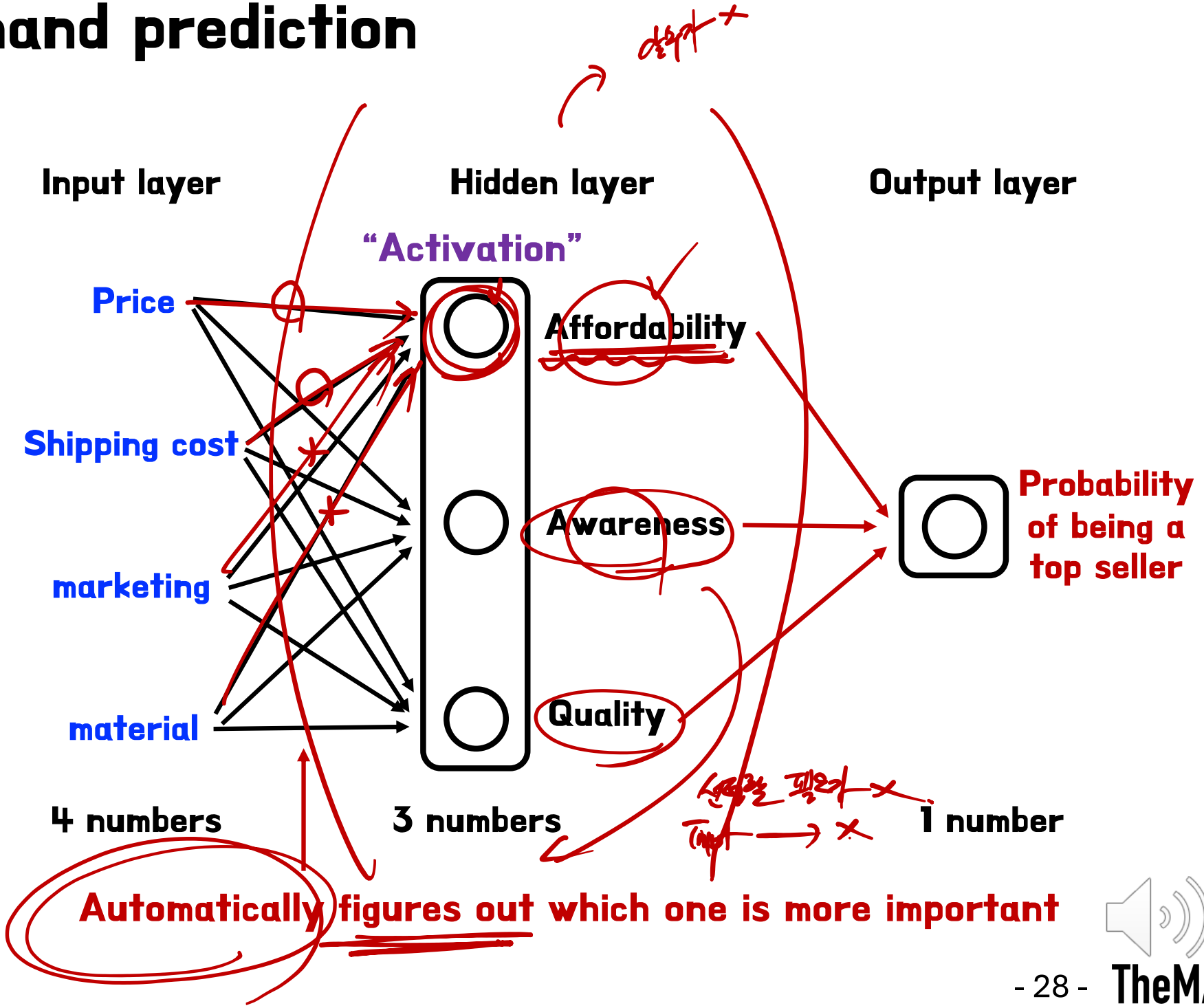


Demand prediction



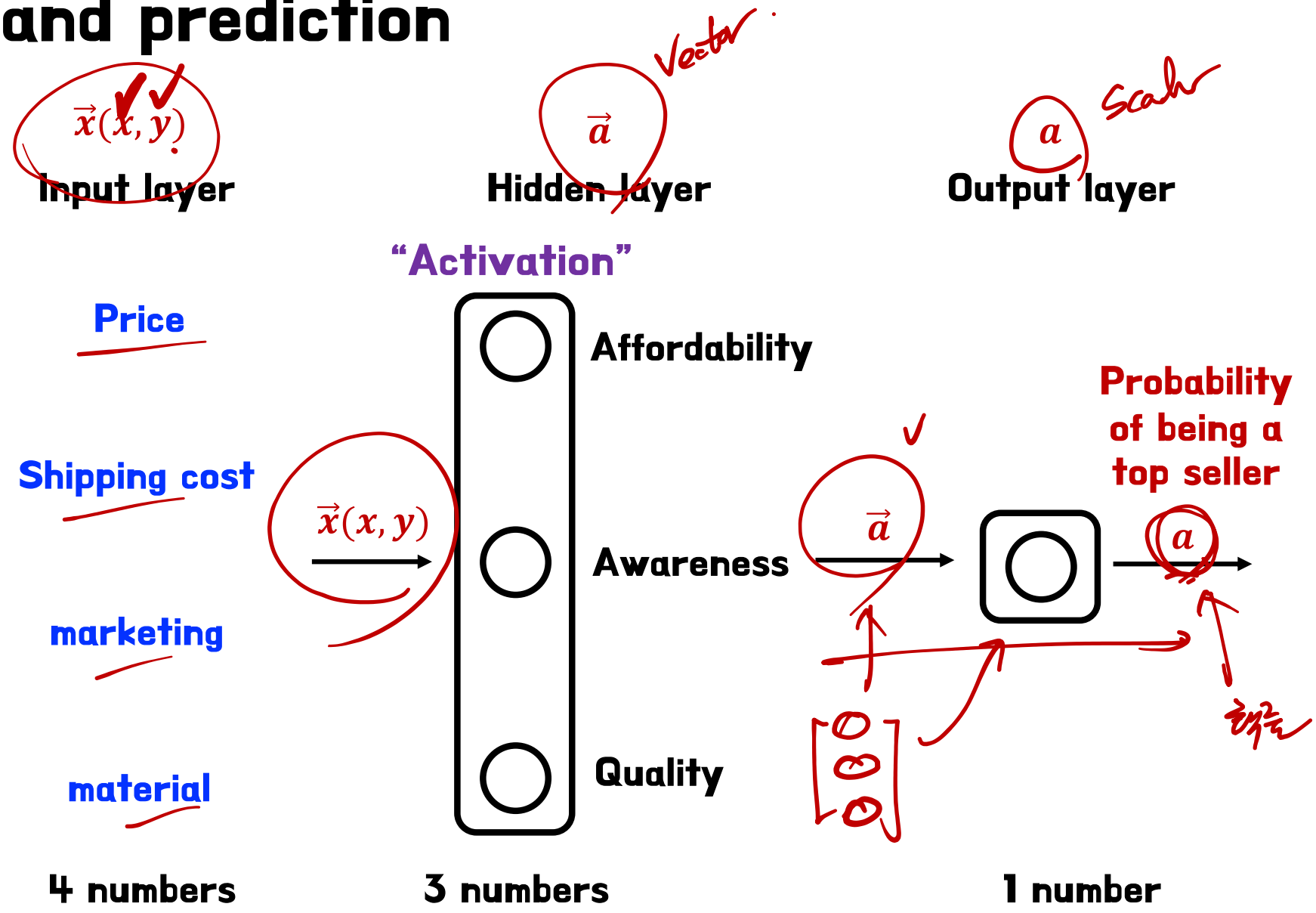
TheMAD

Demand prediction

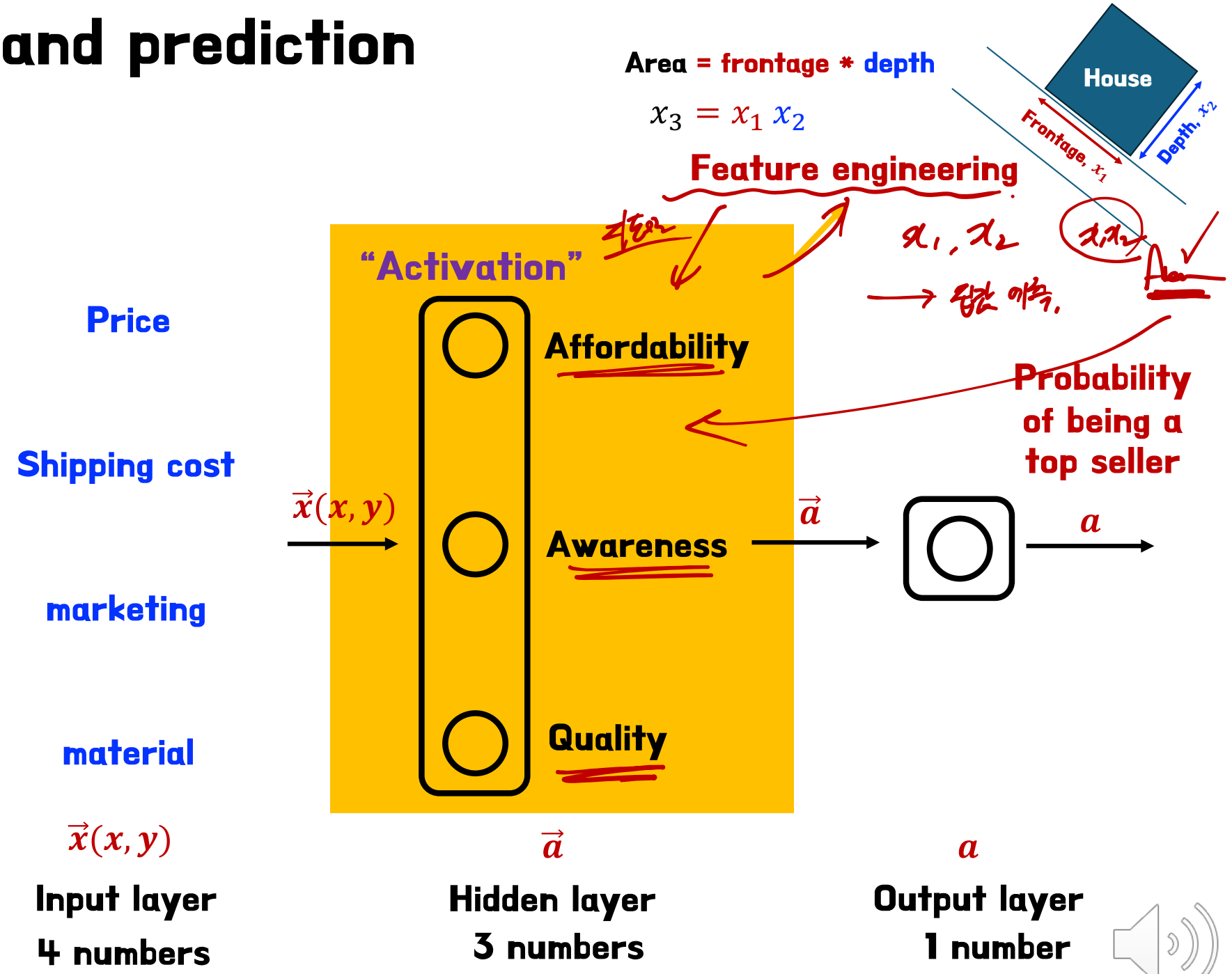


TheMAD

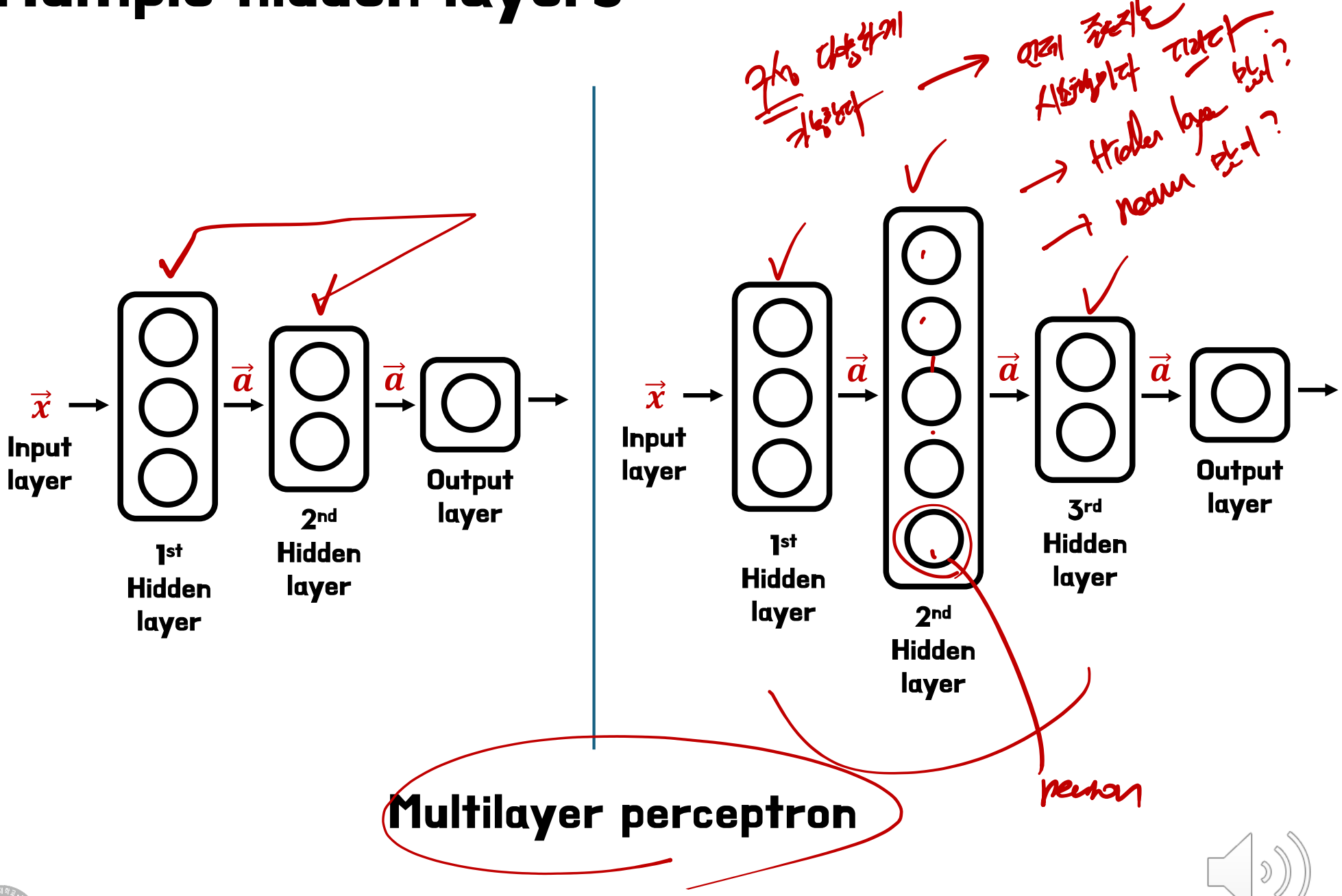
Demand prediction



Demand prediction



Multiple hidden layers

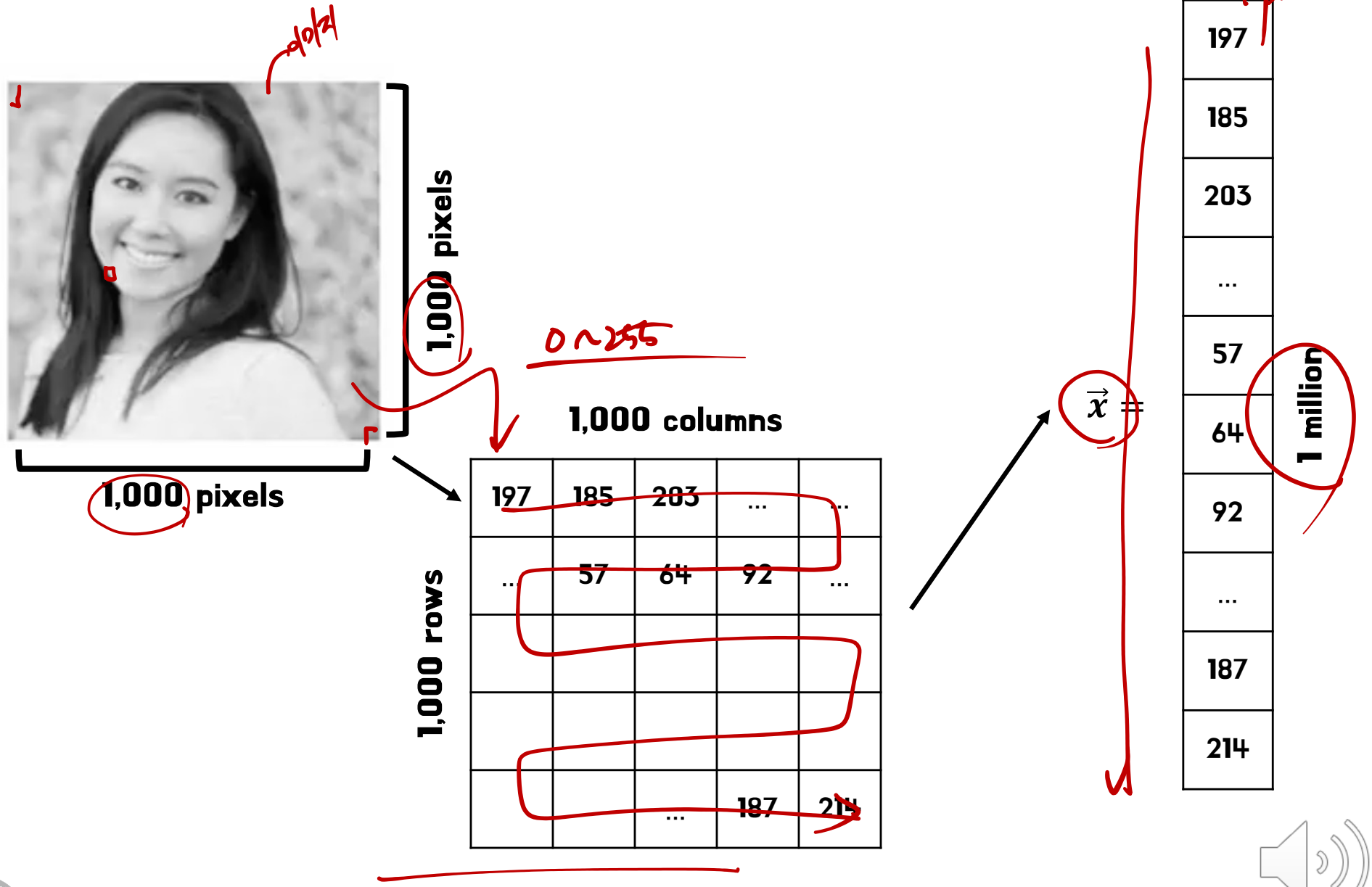


Multilayer perceptron

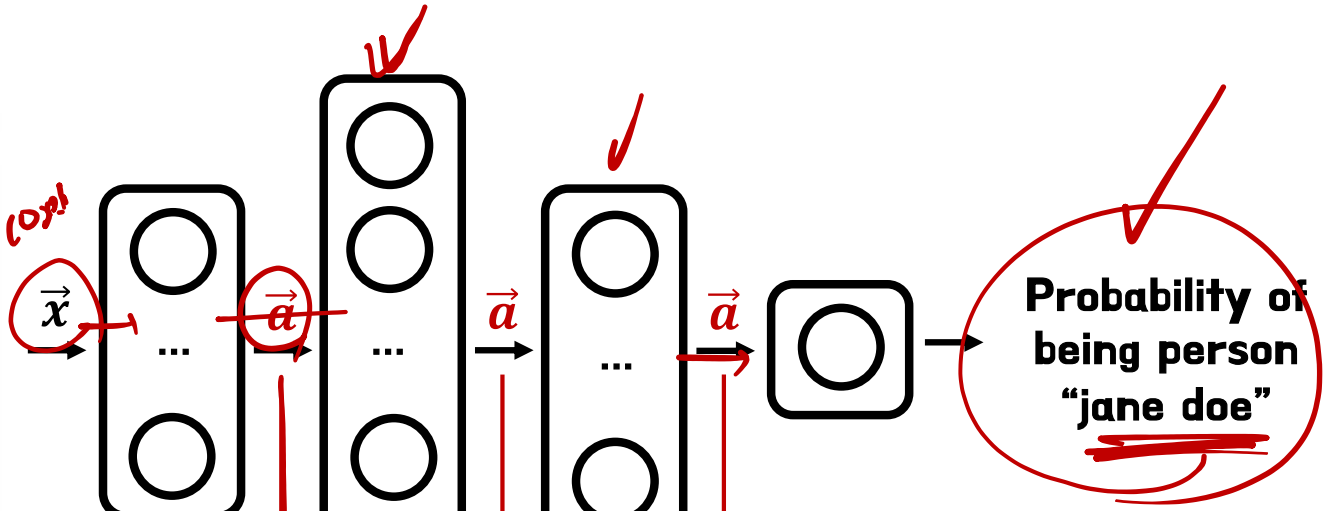
Recognizing images

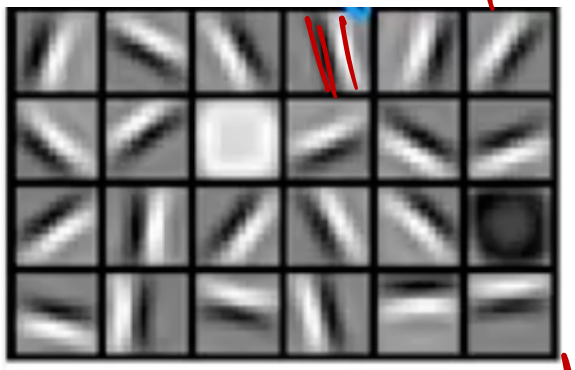
TheMAD

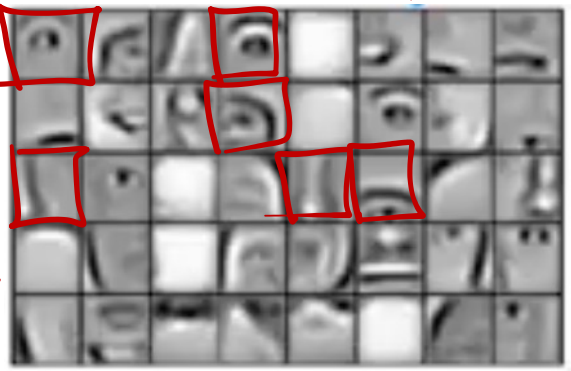
Face recognition



Face recognition







Activations are higher level of features