



Machine Learning 05

Kihyun Shin
DMSE, HBNU

Feature scaling

Feature and parameter values

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

↑
size

↑
bedrooms

x_1 : size (feet²)

range: 300 - 2,000

large

x_2 : # bedrooms

range: 0 - 5

small

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

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

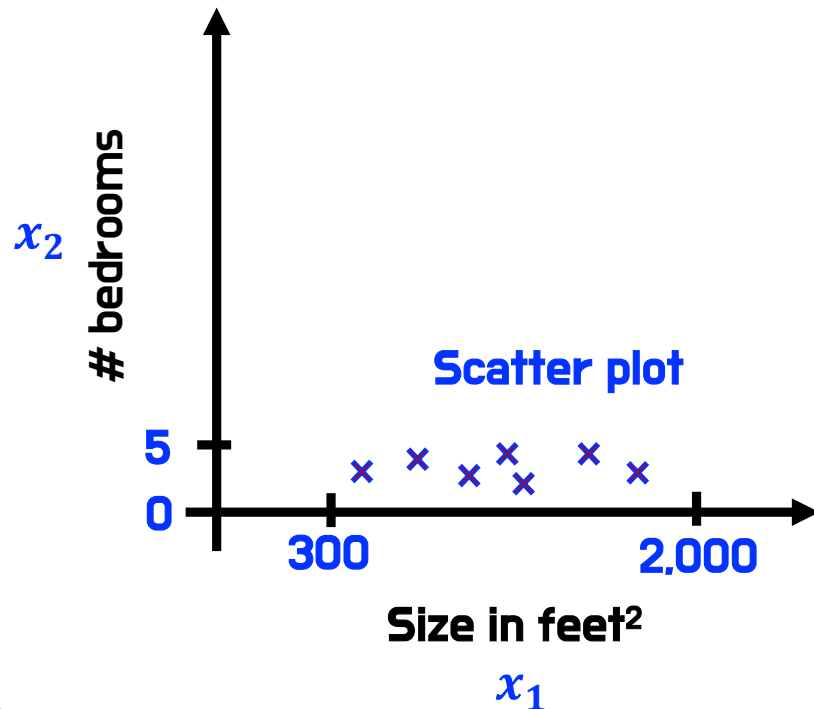
$$\widehat{price} = \$500k \quad \text{More reasonable}$$



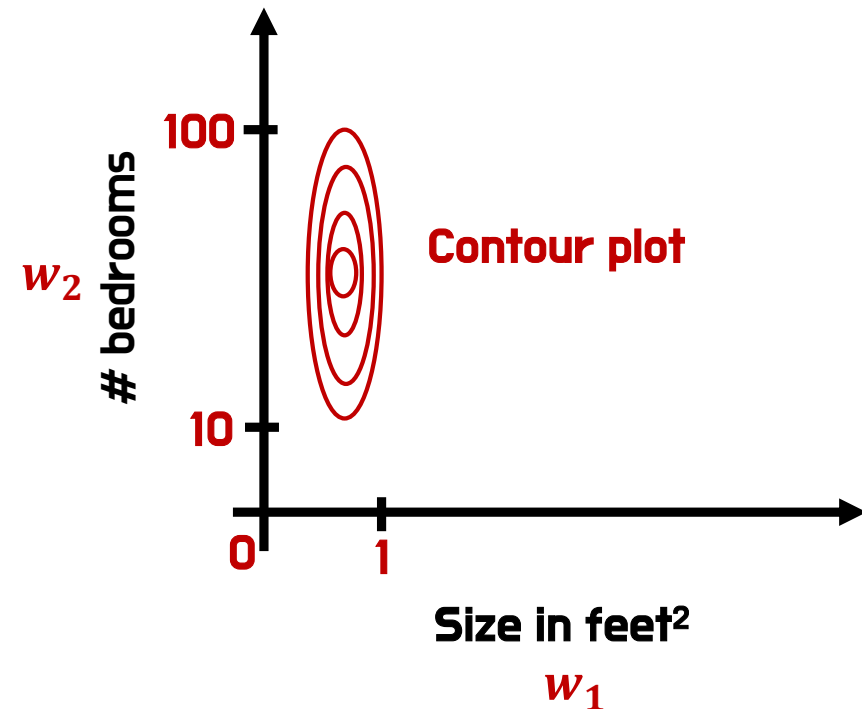
Feature size and parameter size

	Size of feature x_j	Size of parameter w_j
Size in feet ²		
# bedrooms		

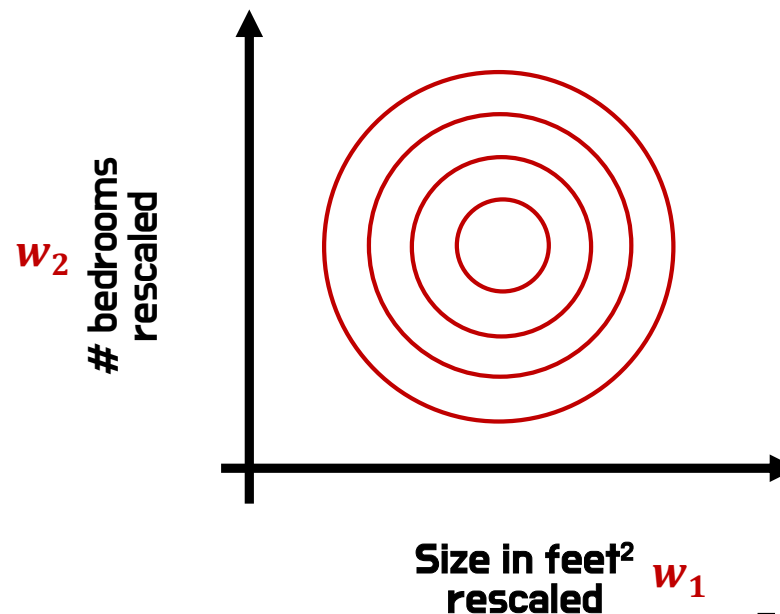
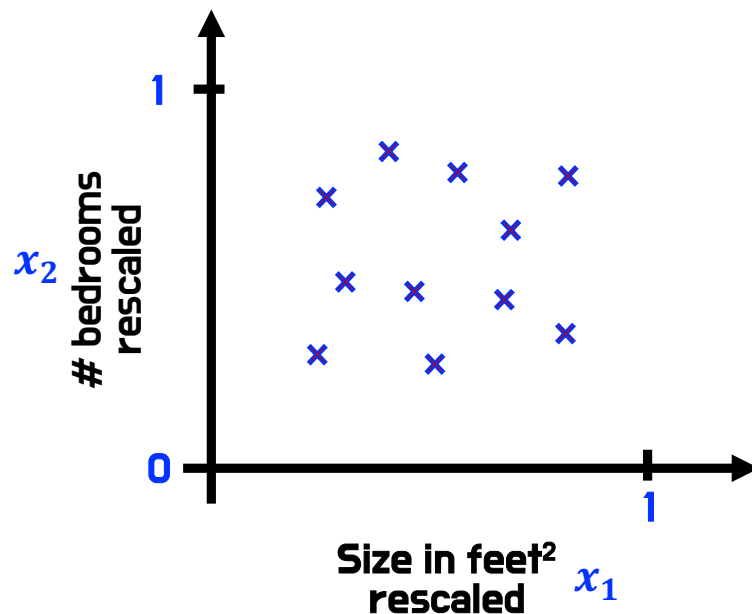
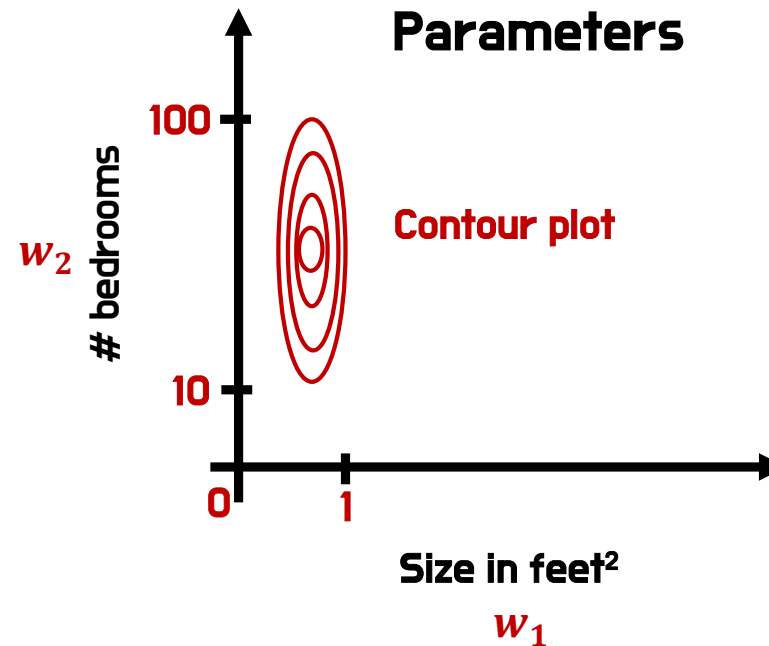
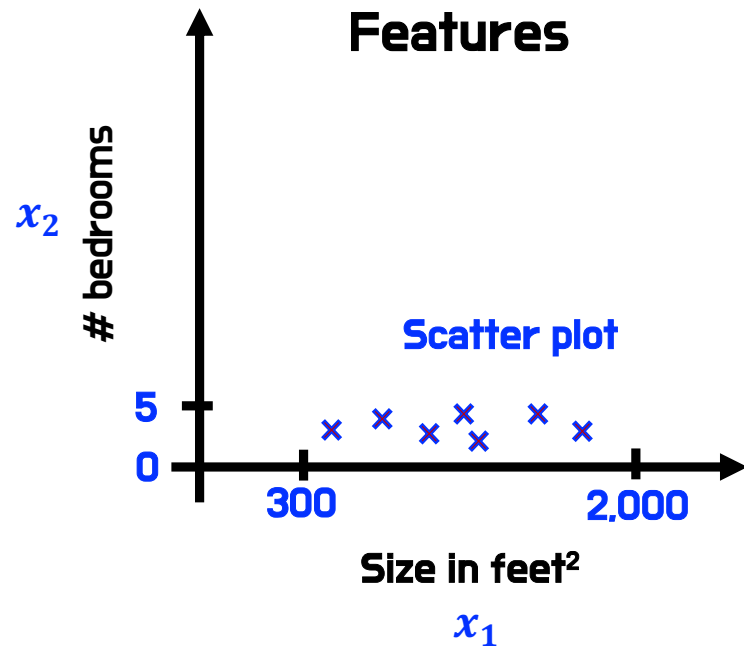
Features



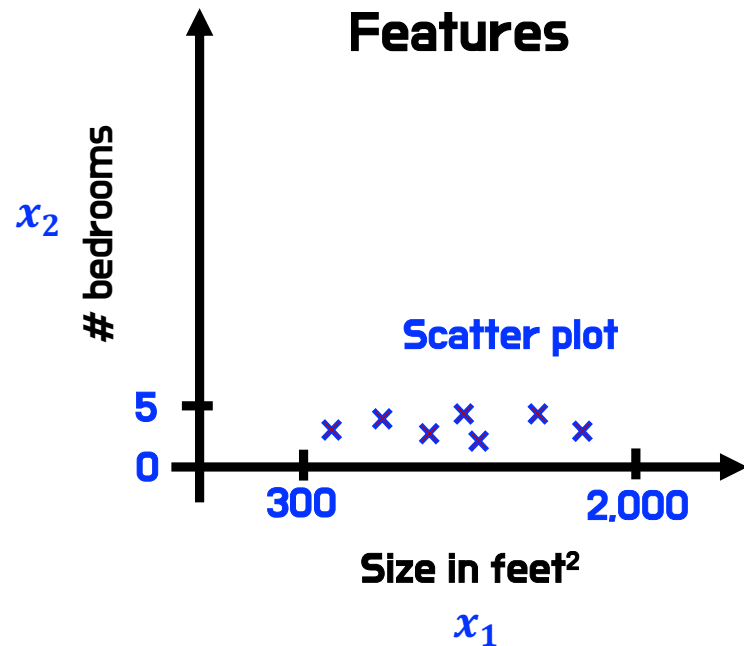
Parameters



Feature size and gradient descent



Feature scaling



$$300 \leq x_1 \leq 2,000$$

$$0 \leq x_2 \leq 5$$



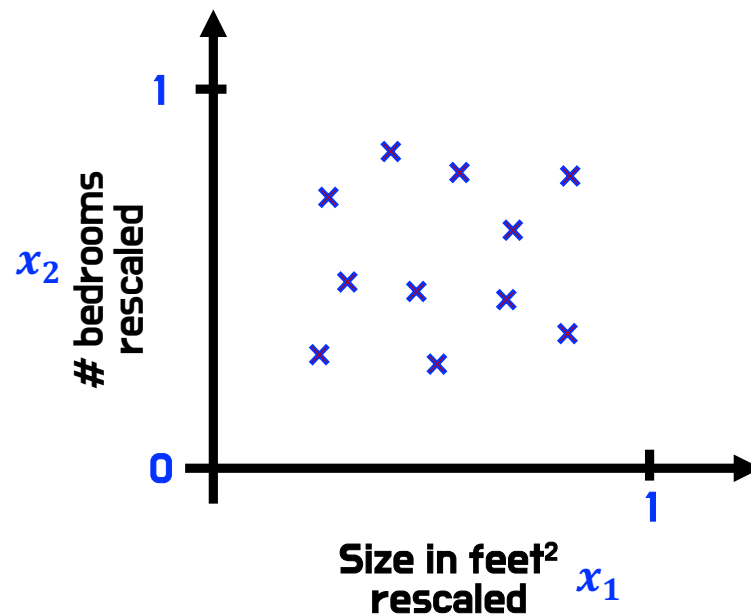
$$x_{1,scaled} = \frac{x_1}{2,000 \text{ Max}}$$

$$x_{2,scaled} = \frac{x_2}{5 \text{ Max}}$$

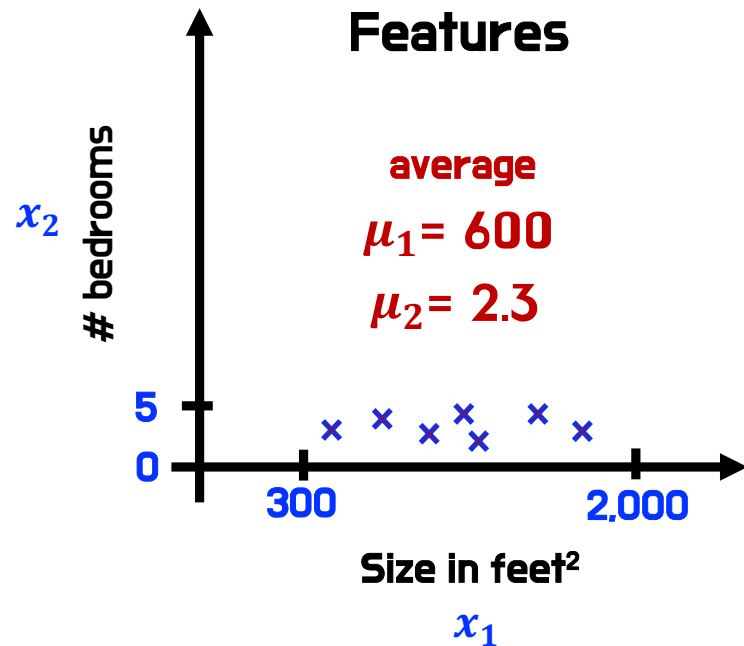


$$0.15 \leq x_{1,rescaled} \leq 1$$

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



Mean normalization



$$300 \leq x_1 \leq 2,000$$

$$0 \leq x_2 \leq 5$$

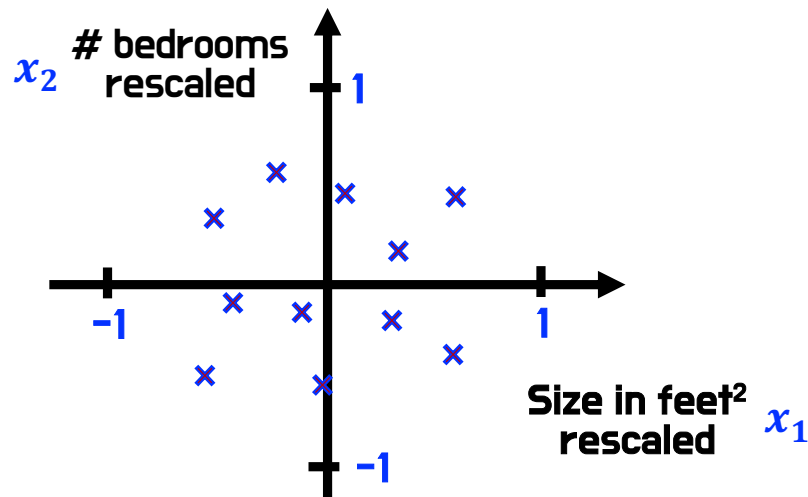


$$x_1 = \frac{x_1 - \mu_1}{2,000 - 300}$$

Max Min

$$x_2 = \frac{x_2 - \mu_2}{5 - 0}$$

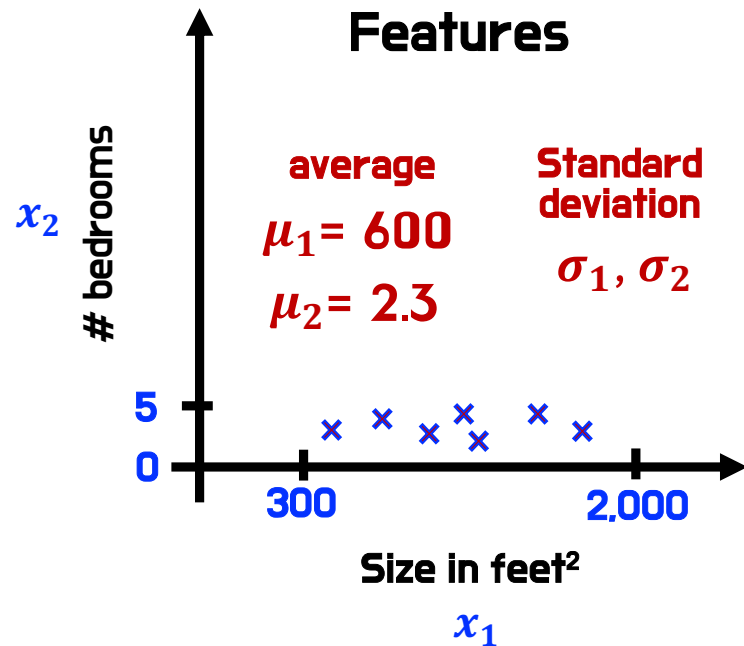
Max Min



$$-0.18 \leq x_1 \leq 0.82$$

$$-0.46 \leq x_2 \leq 0.54$$

Z-score normalization



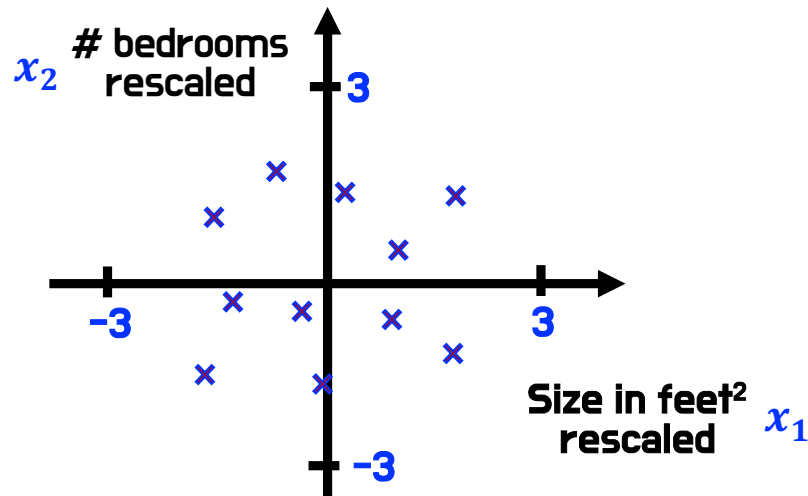
$$300 \leq x_1 \leq 2,000$$

$$0 \leq x_2 \leq 5$$



$$x_1 = \frac{x_1 - \mu_1}{\sigma_1}$$

$$x_2 = \frac{x_2 - \mu_2}{\sigma_2}$$



$$-0.67 \leq x_1 \leq 3.1$$

$$-1.6 \leq x_2 \leq 1.9$$



Feature scaling

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

$$-3 \leq x_j \leq 3$$

Acceptable ranges

$$-0.3 \leq x_j \leq 0.3$$

$$0 \leq x_1 \leq 3$$

Okay. no rescaling

$$-2 \leq x_2 \leq 0.5$$

Okay. no rescaling

$$-100 \leq x_3 \leq 100$$

Too large $\rightarrow$ rescale

$$-0.001 \leq x_4 \leq 0.001$$

Too large $\rightarrow$ rescale

$$98.6 \leq x_5 \leq 105$$

Too large $\rightarrow$ rescale

Checking gradient descent For convergence

Gradient descent

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

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

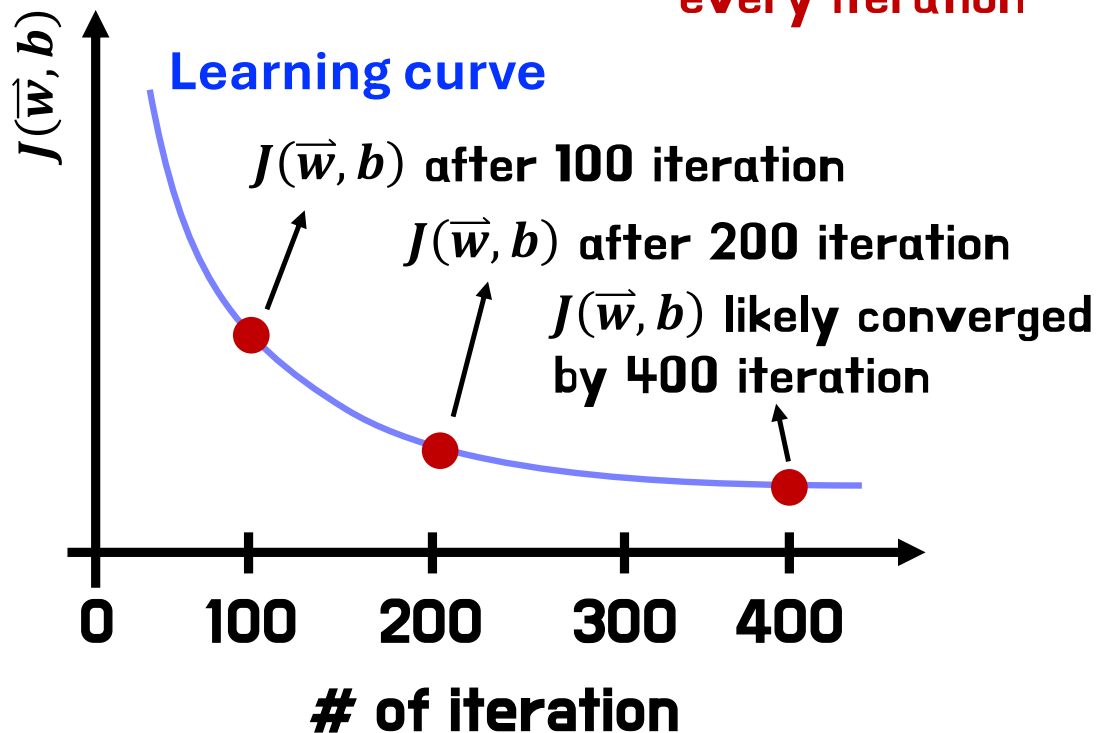
$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b) \longrightarrow \frac{1}{m} \sum_{i=1}^m (f_{\vec{w}, b}(\vec{x}^{(i)}) - \mathbf{y}^{(i)})$$

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}

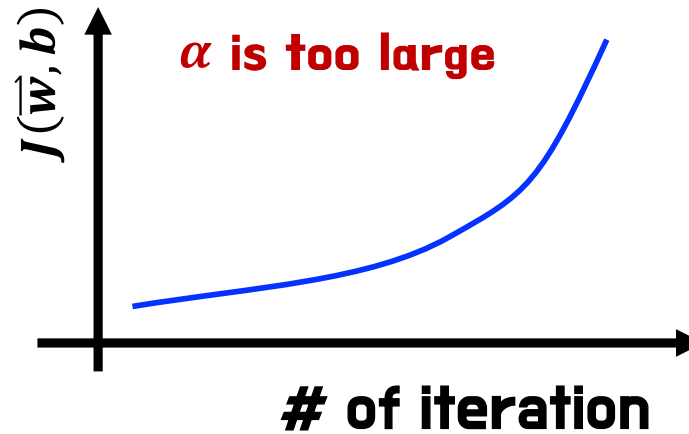
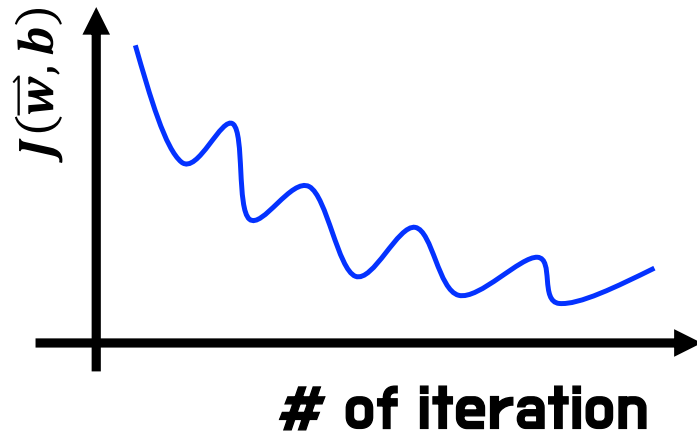
**If $J(\vec{w}, b)$ decrease by $\leq \epsilon$ in
one iteration,
Declare convergence,
(found parameters $\vec{w}, b$ to get
close to global minimum)**

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



Gradient descent

Identify problem with gradient descent



Debugging needed

Or

α is too large

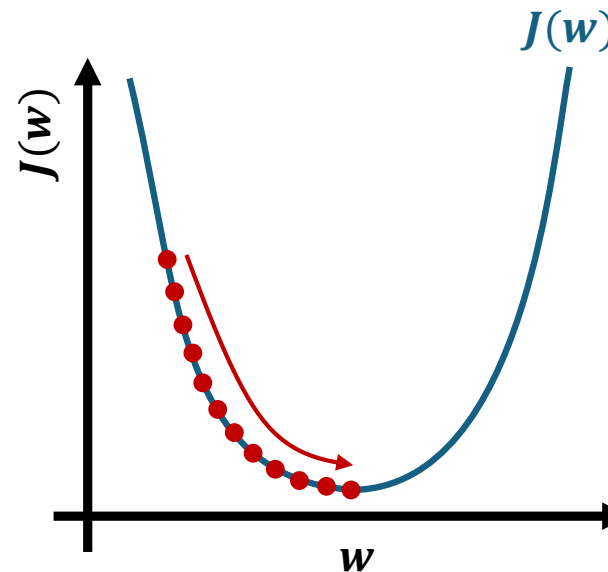
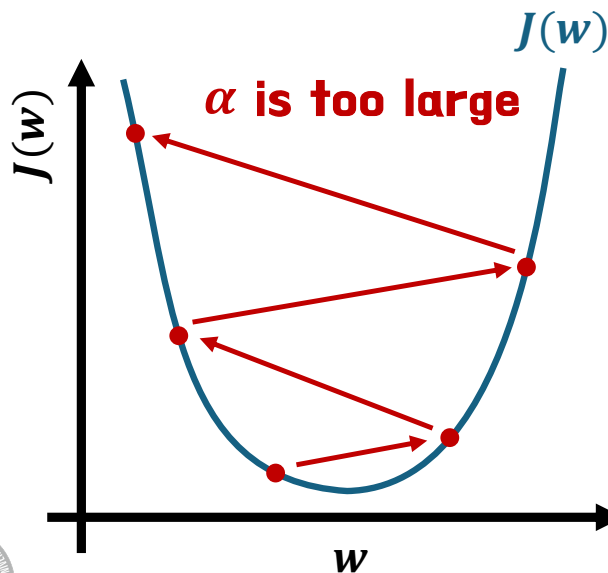
wrong

$$w_1 = w_1 + \alpha d_1$$

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

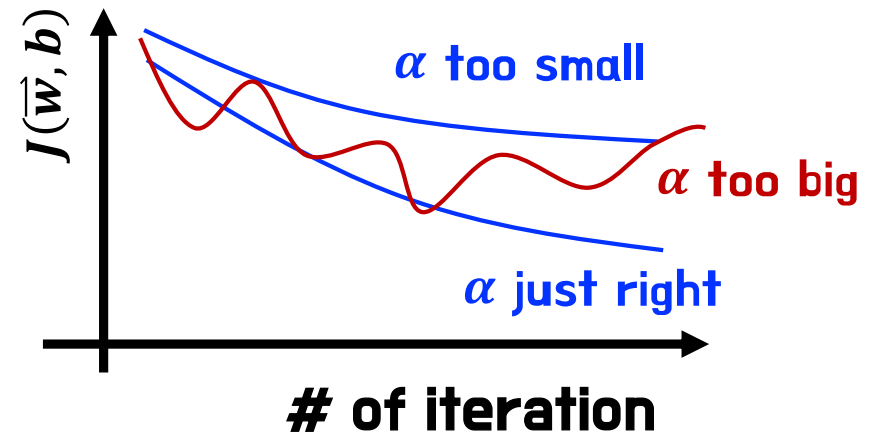
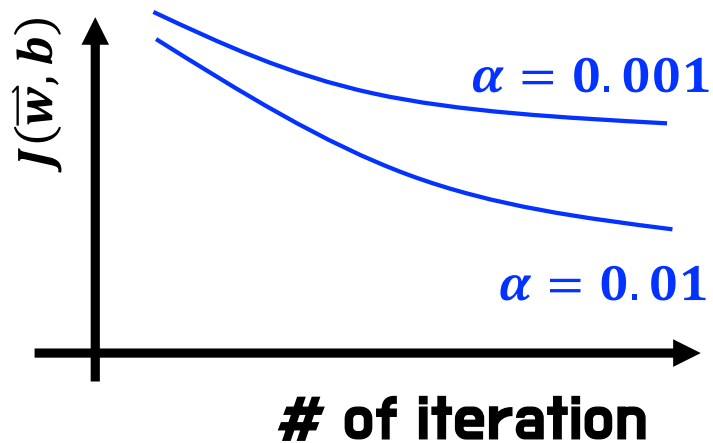
If α is too small,
Gradient descent
takes a lot of iteration
to converge



Gradient descent

Values of α to try:

... 0.001 0.01 0.1 1 ...



Feature engineering

Feature engineering

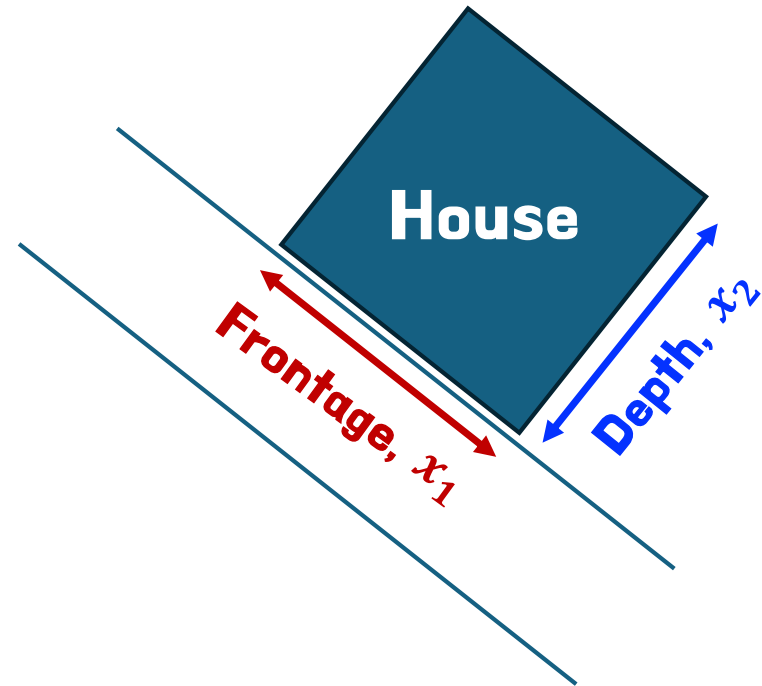
$$f_{\vec{w},b}(\vec{x}) = w_1 \underset{\substack{\uparrow \\ \text{Frontage}}}{x_1} + w_2 \underset{\substack{\uparrow \\ \text{Depth}}}{x_2} + b$$

Area = frontage * depth

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

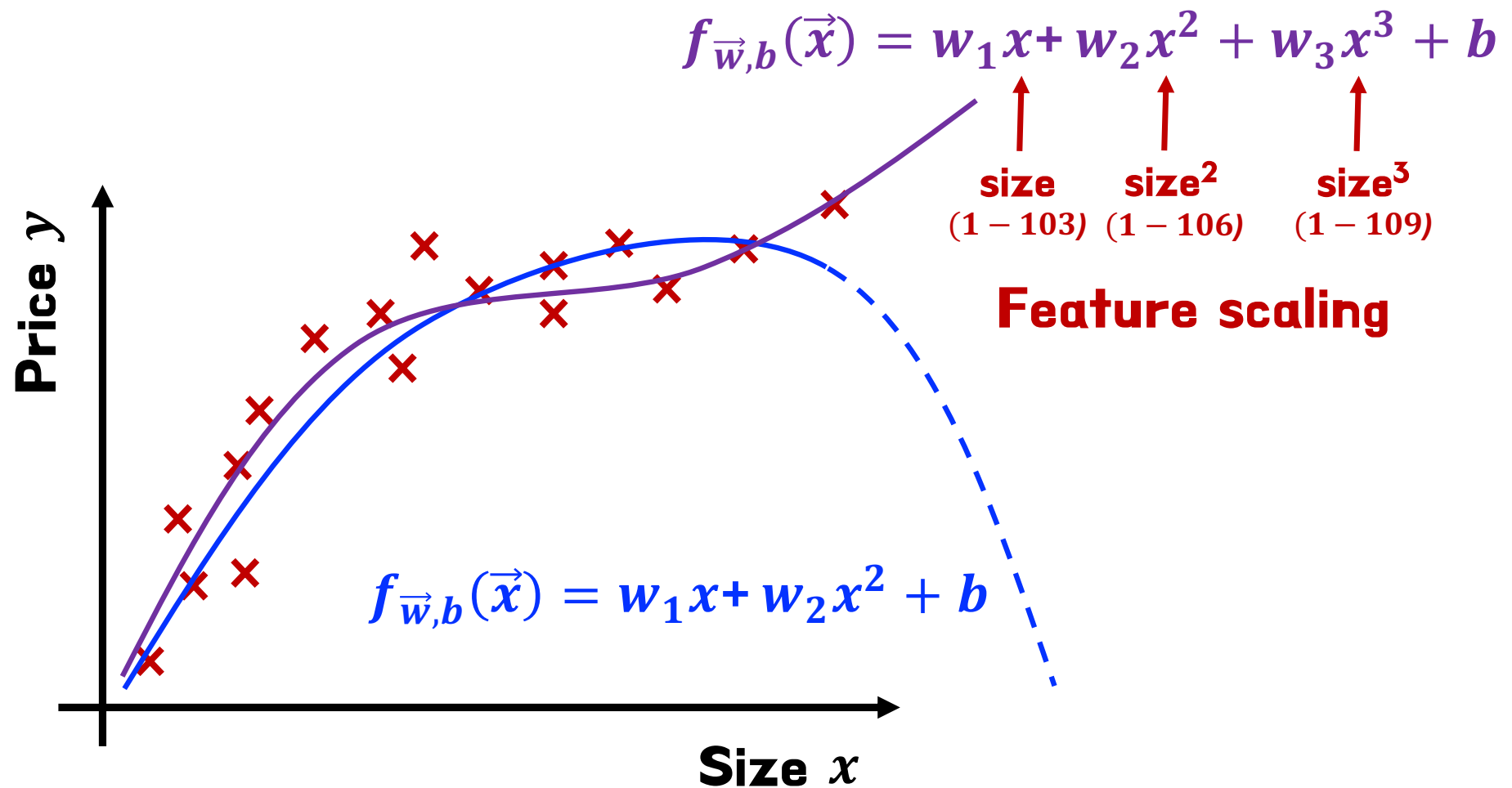
$$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + \underset{\text{New feature}}{w_3 x_3} + b$$

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

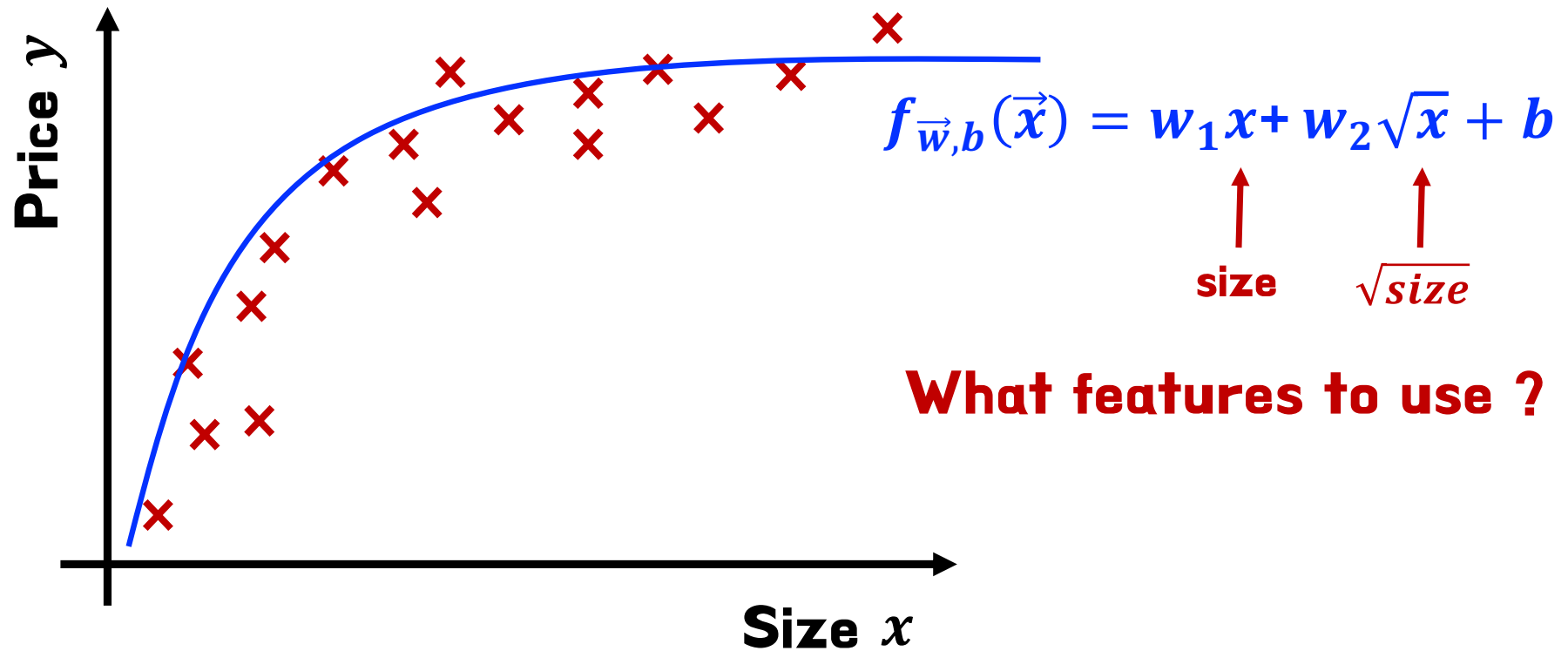


Polynomial regression

Polynomial regression



Choice of feature



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.

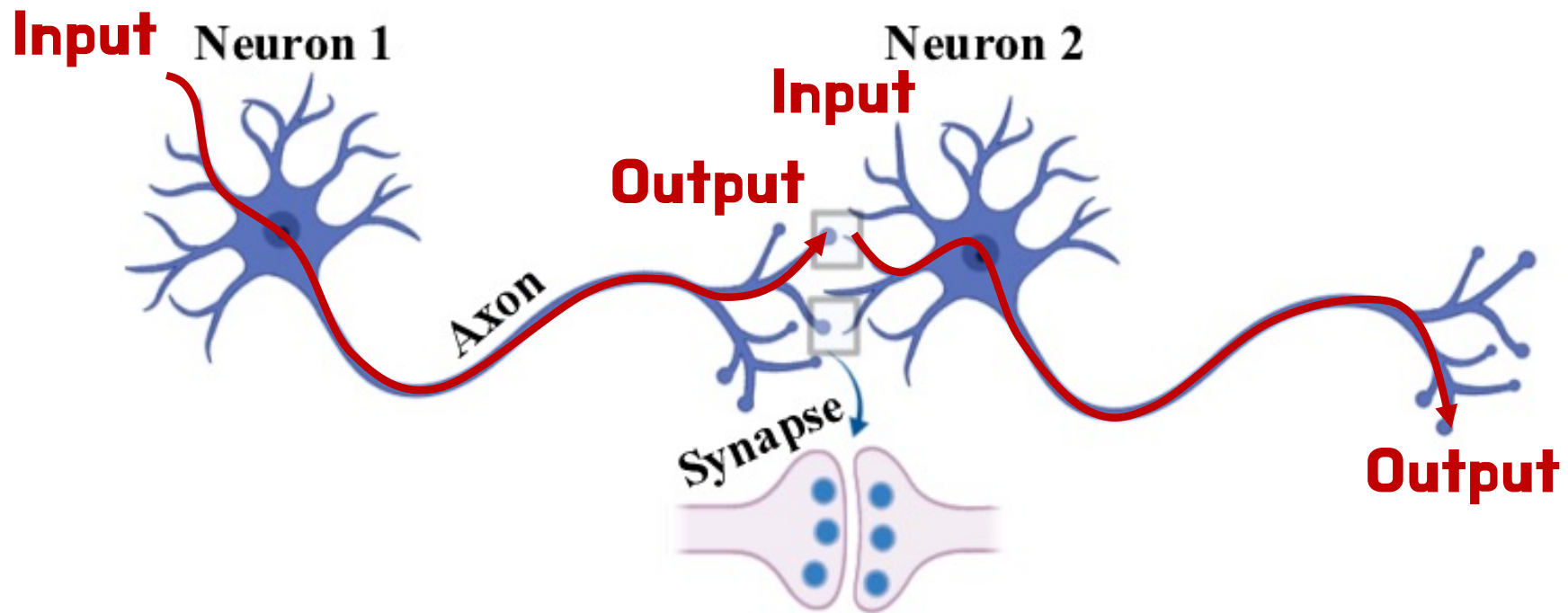
→ Deep learning

→ Speech → Images → text (NLP) → ...



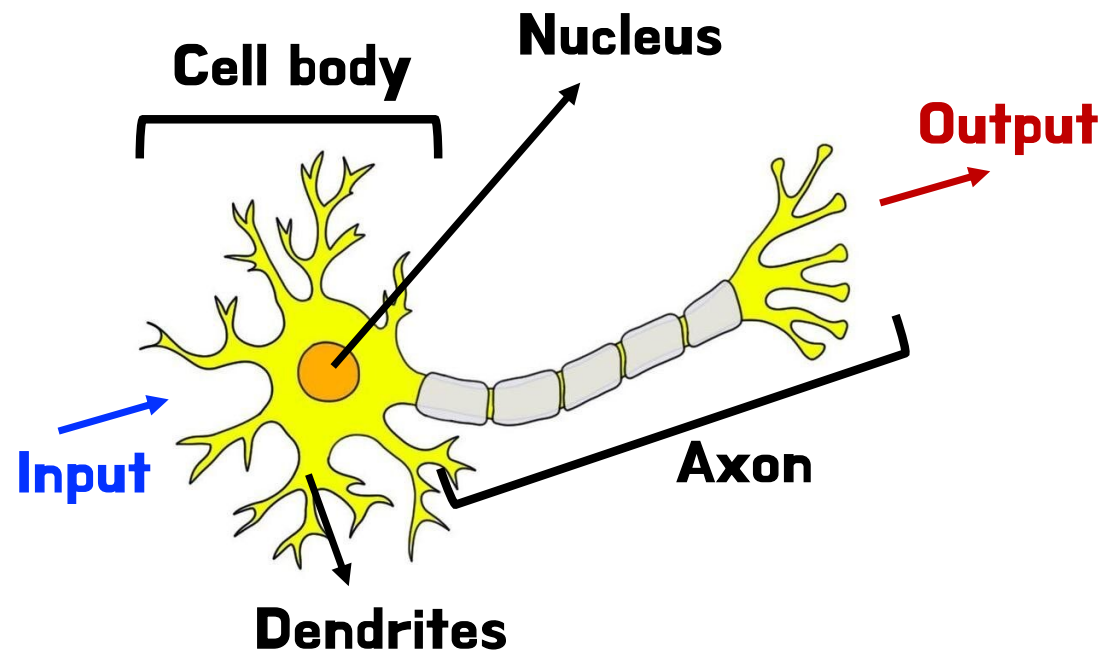
**Weather estimation, Chaos theory, Black hole,
Materials development, Drug discovery, Auto pilot**

Neurons in brain

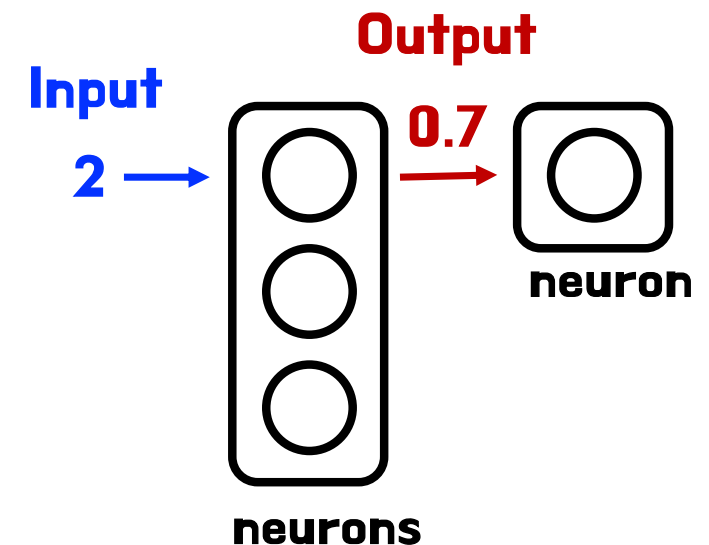


Simplified model

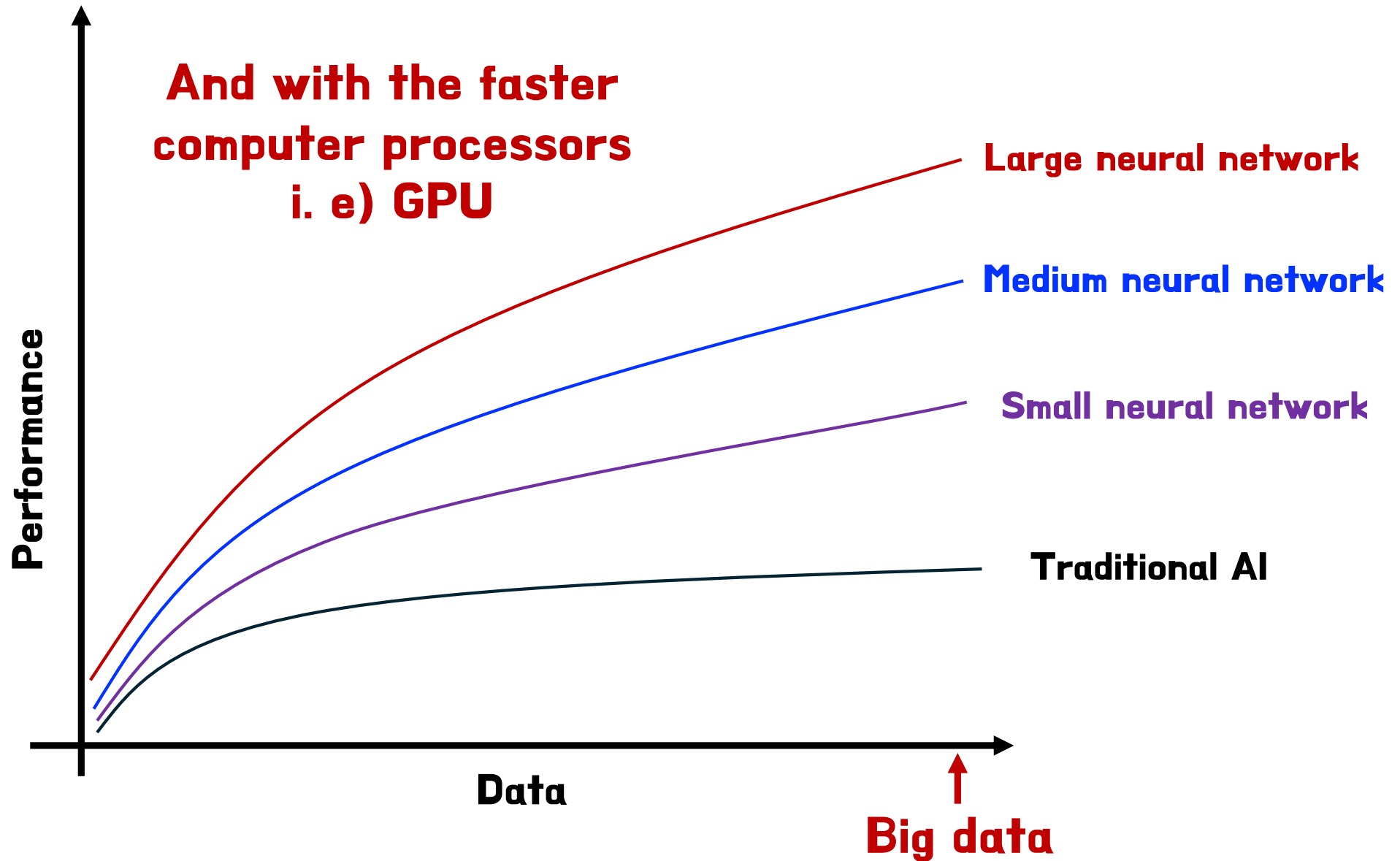
Biological neuron



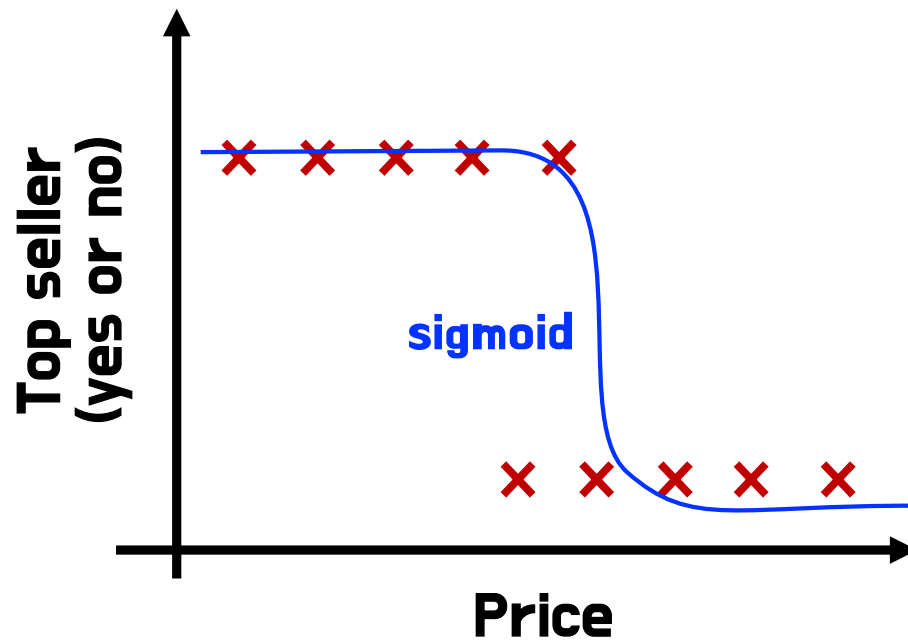
Simplified mathematical model of a neuron



Why now ?



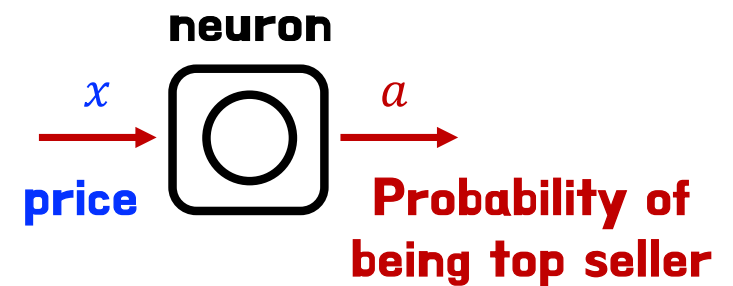
Demand prediction



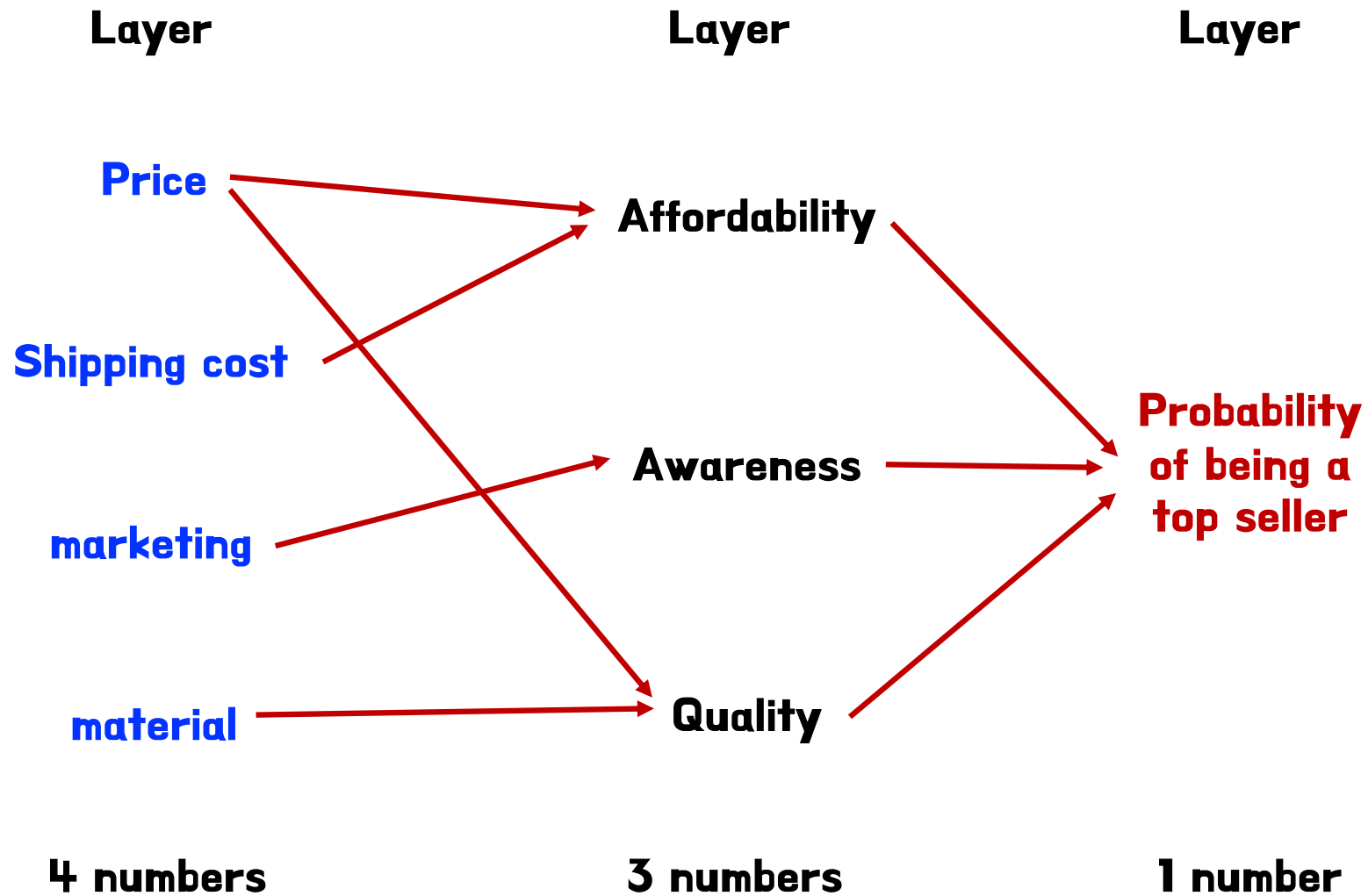
x = price **input**

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

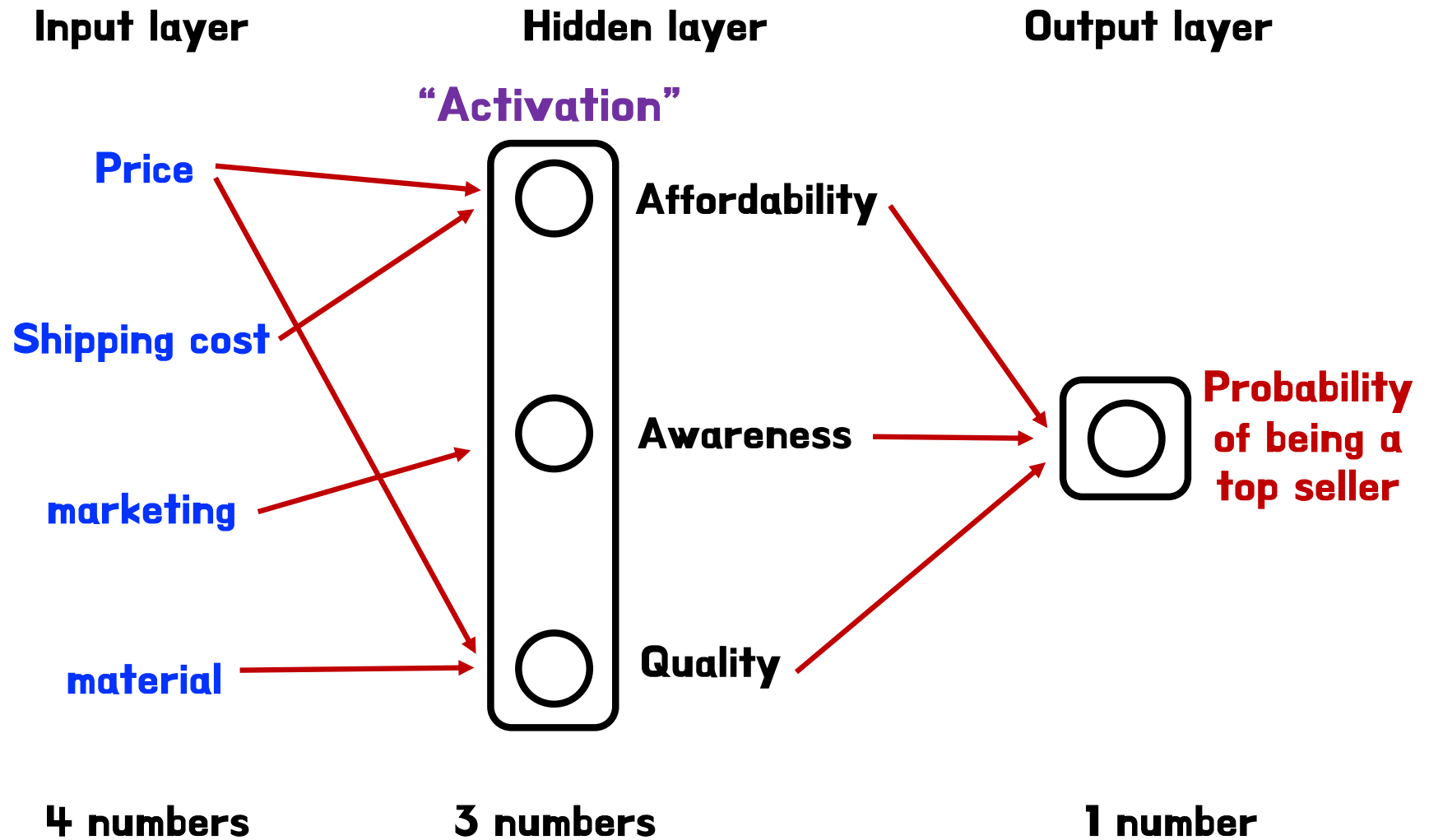
a ↑ **activation** **output**



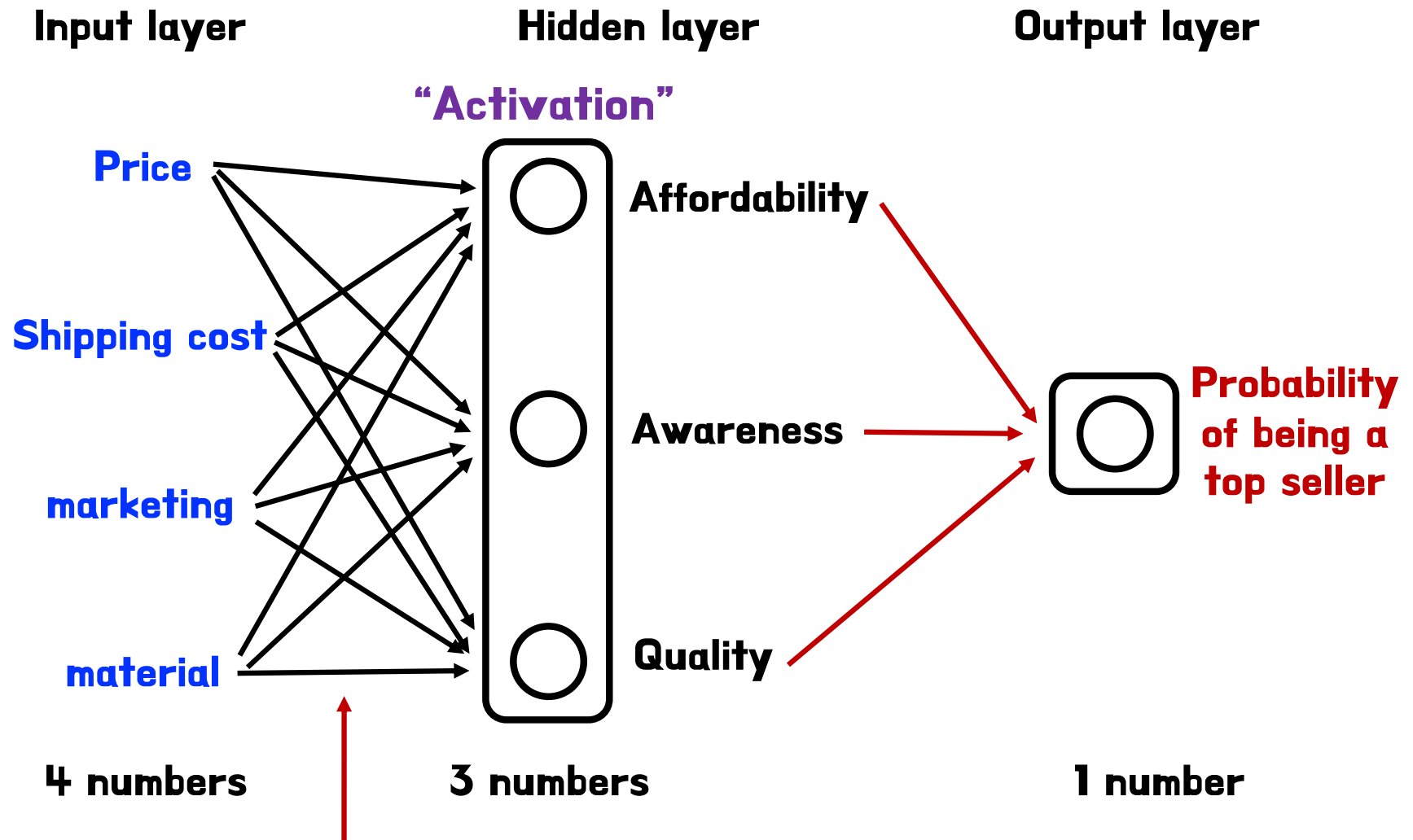
Demand prediction



Demand prediction

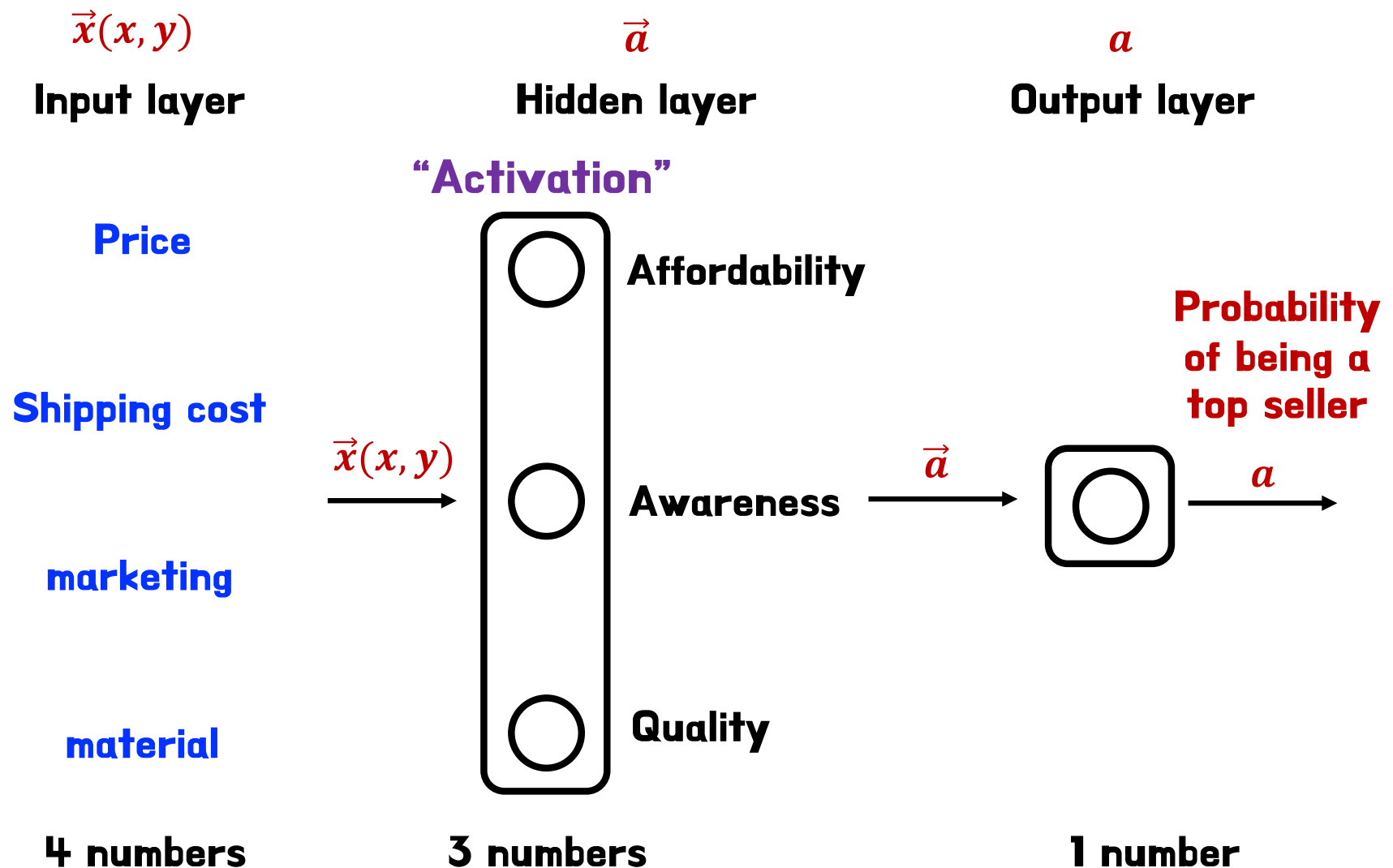


Demand prediction



Automatically figures out which one is more important

Demand prediction

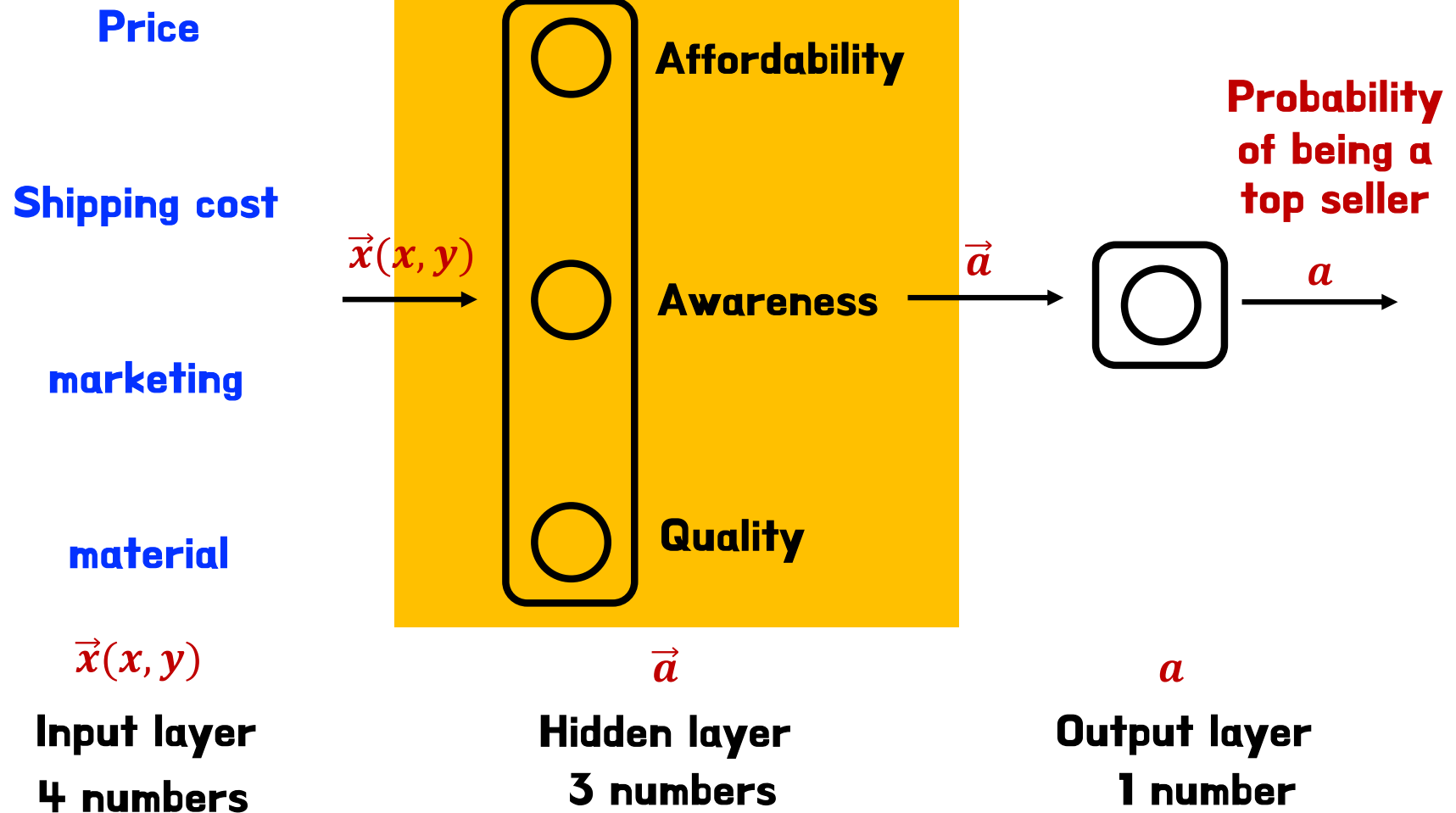
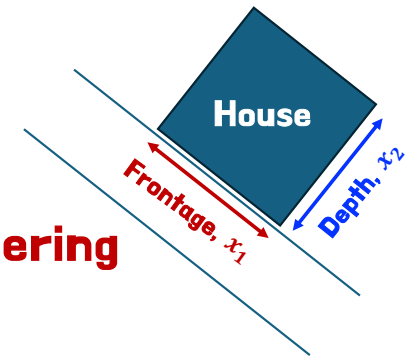


Demand prediction

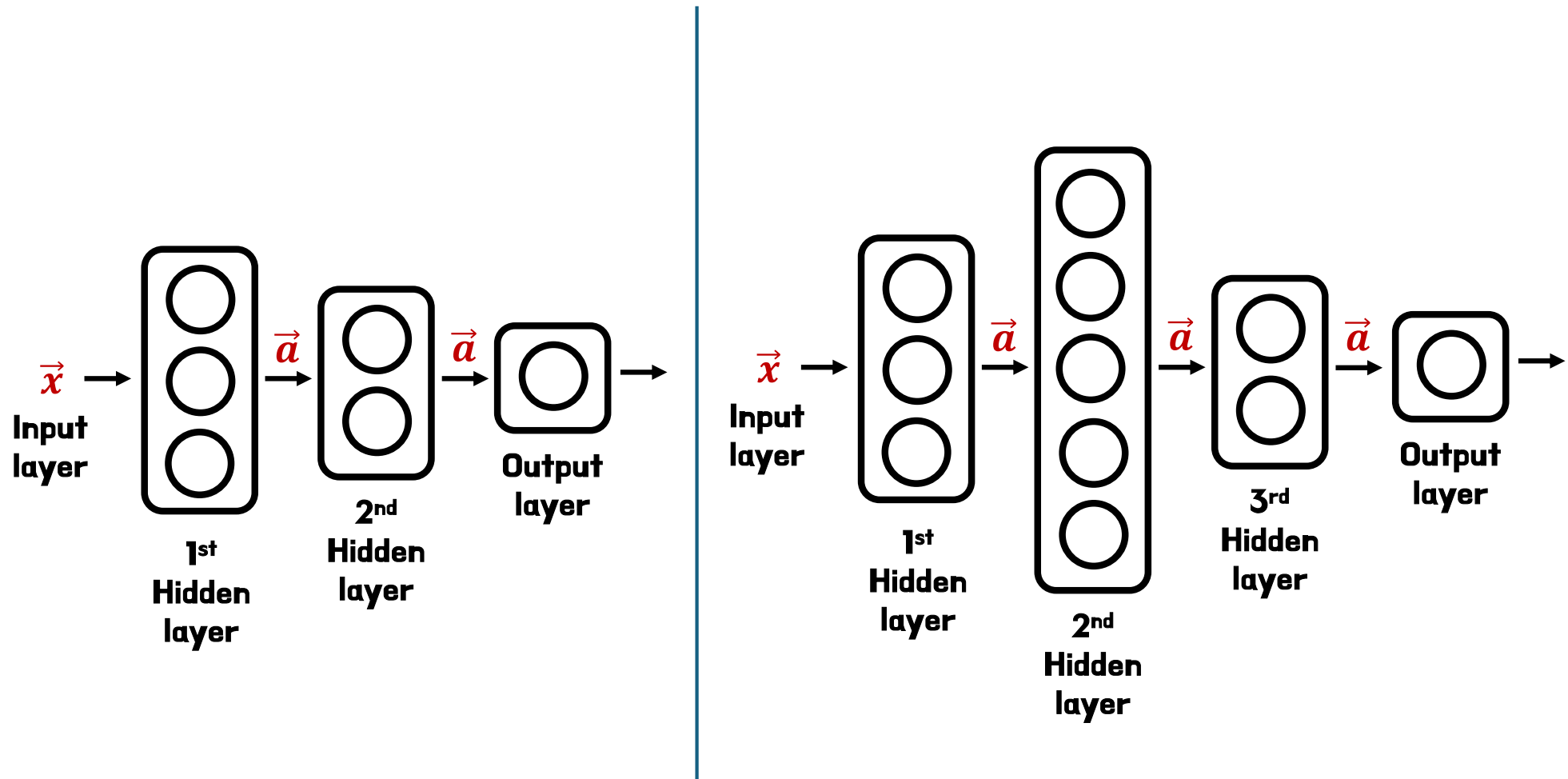
Area = frontage * depth

$$x_3 = x_1 x_2$$

Feature engineering



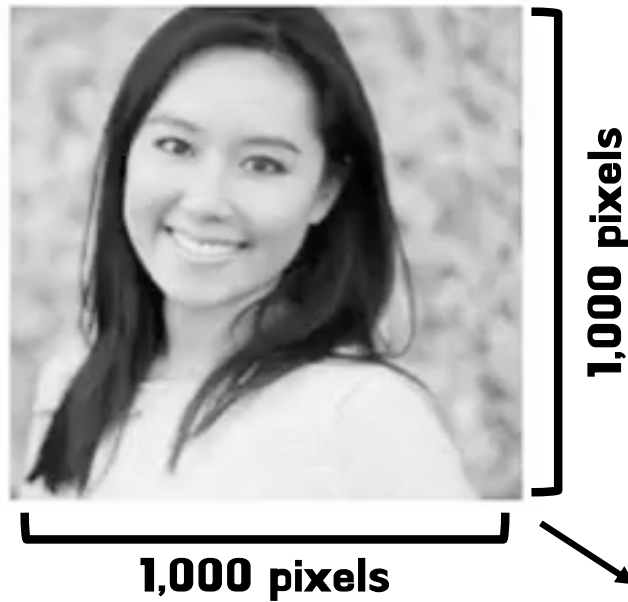
Multiple hidden layers



Multilayer perceptron

Recognizing images

Face recognition



1,000 columns

1,000 rows

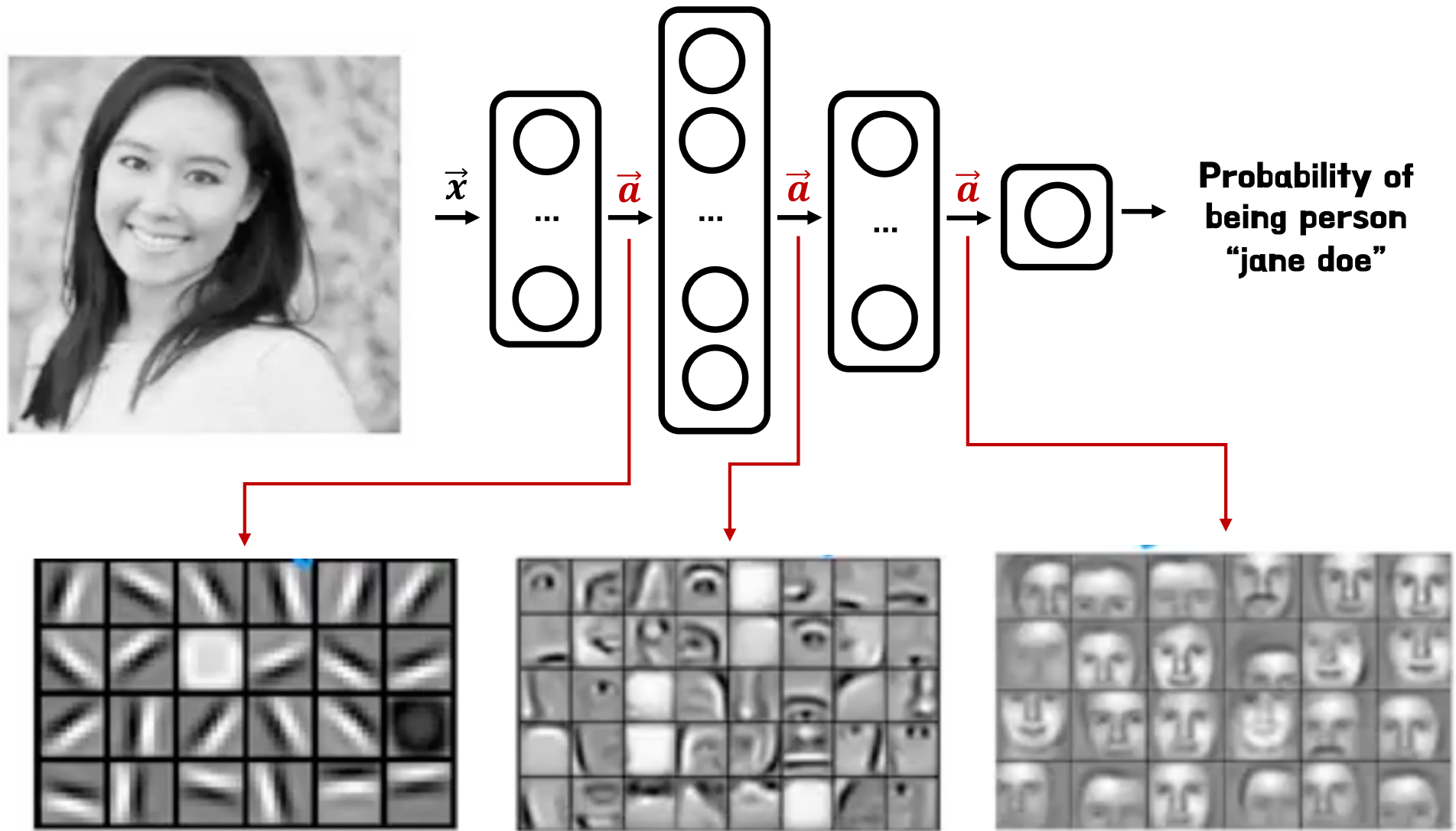
197	185	203	...	...
...	57	64	92	...
		...	187	214

$$\vec{x} =$$

1 million

197
185
203
...
57
64
92
...
187
214

Face recognition



Activations are higher level of features