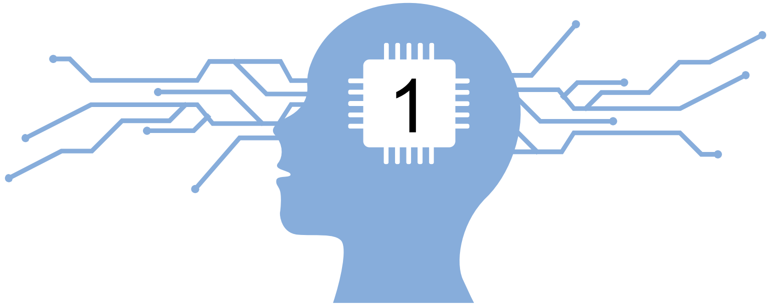
Kecerdasan Buatan (Artificial Intelligence)

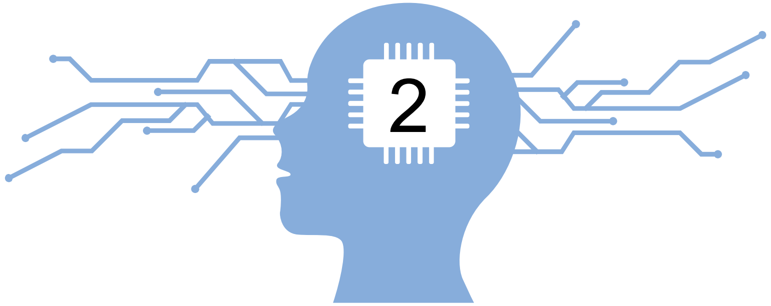
Regression

Chandra Prasetyo Utomo

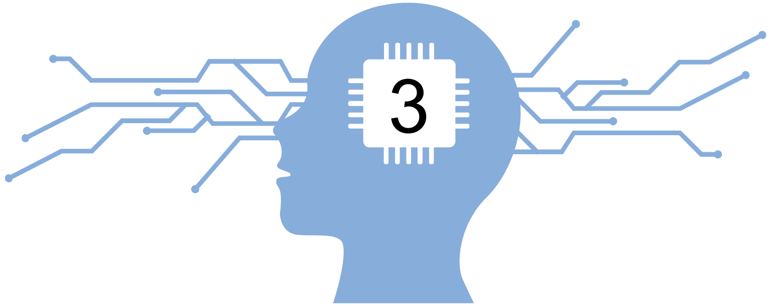
Learning Objectives



Understand supervised learning training and testing procedure.



Understand model representation in linear regression algorithm.



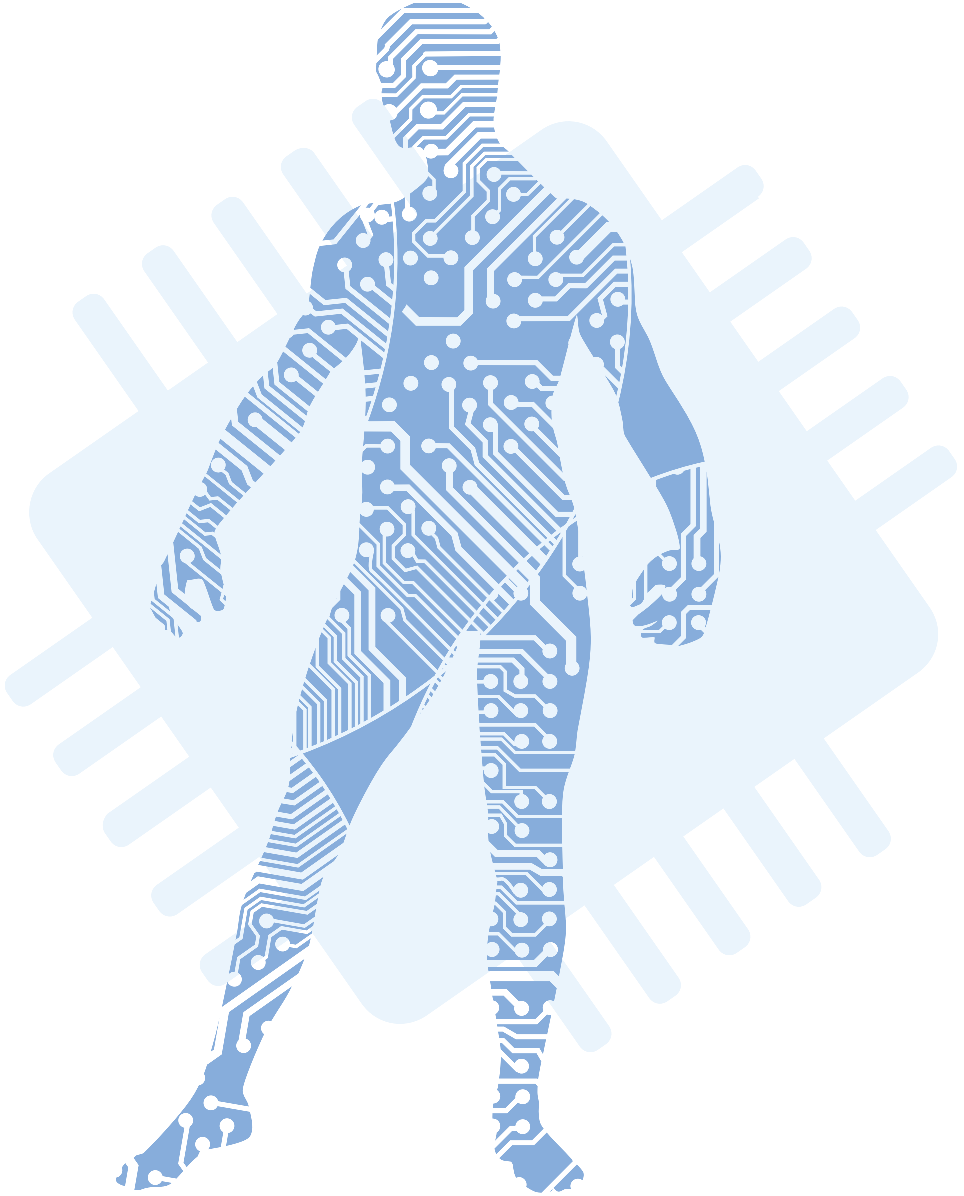
Understand model learning using gradient descent for linear regression.

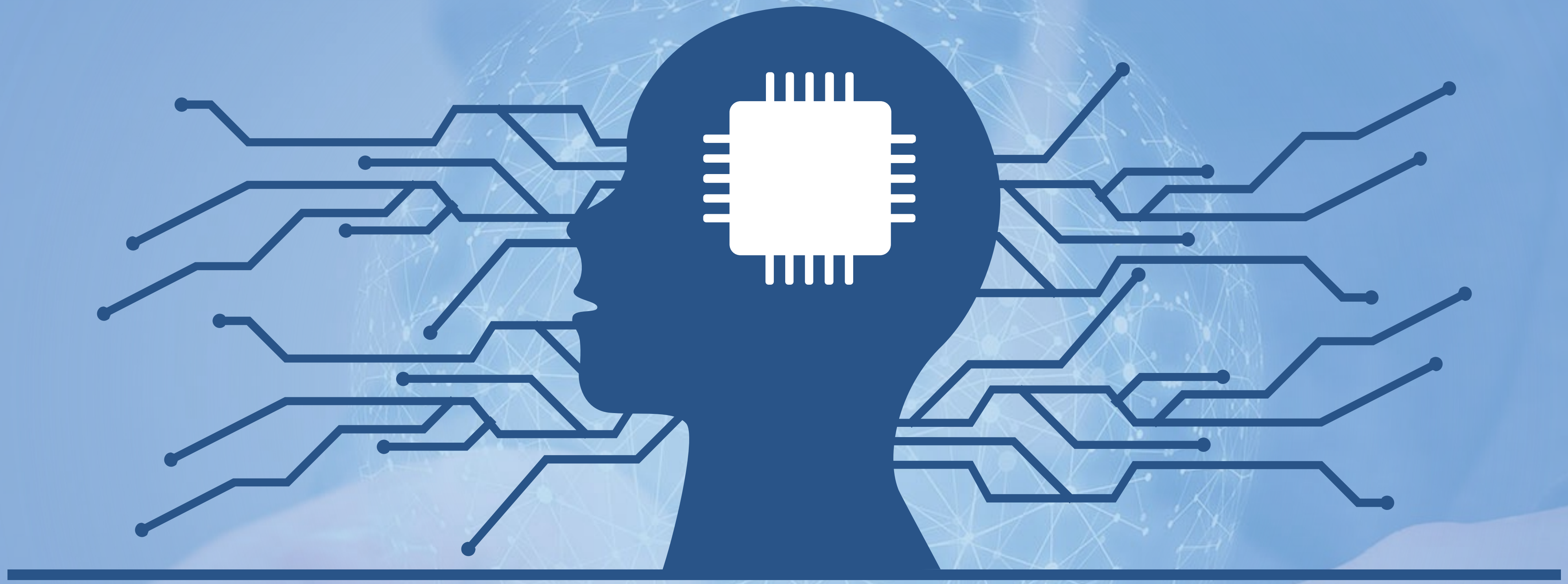
Lecture Outline

1 **Supervised Learning**

2 **Model Representation**

3 **Model Learning**

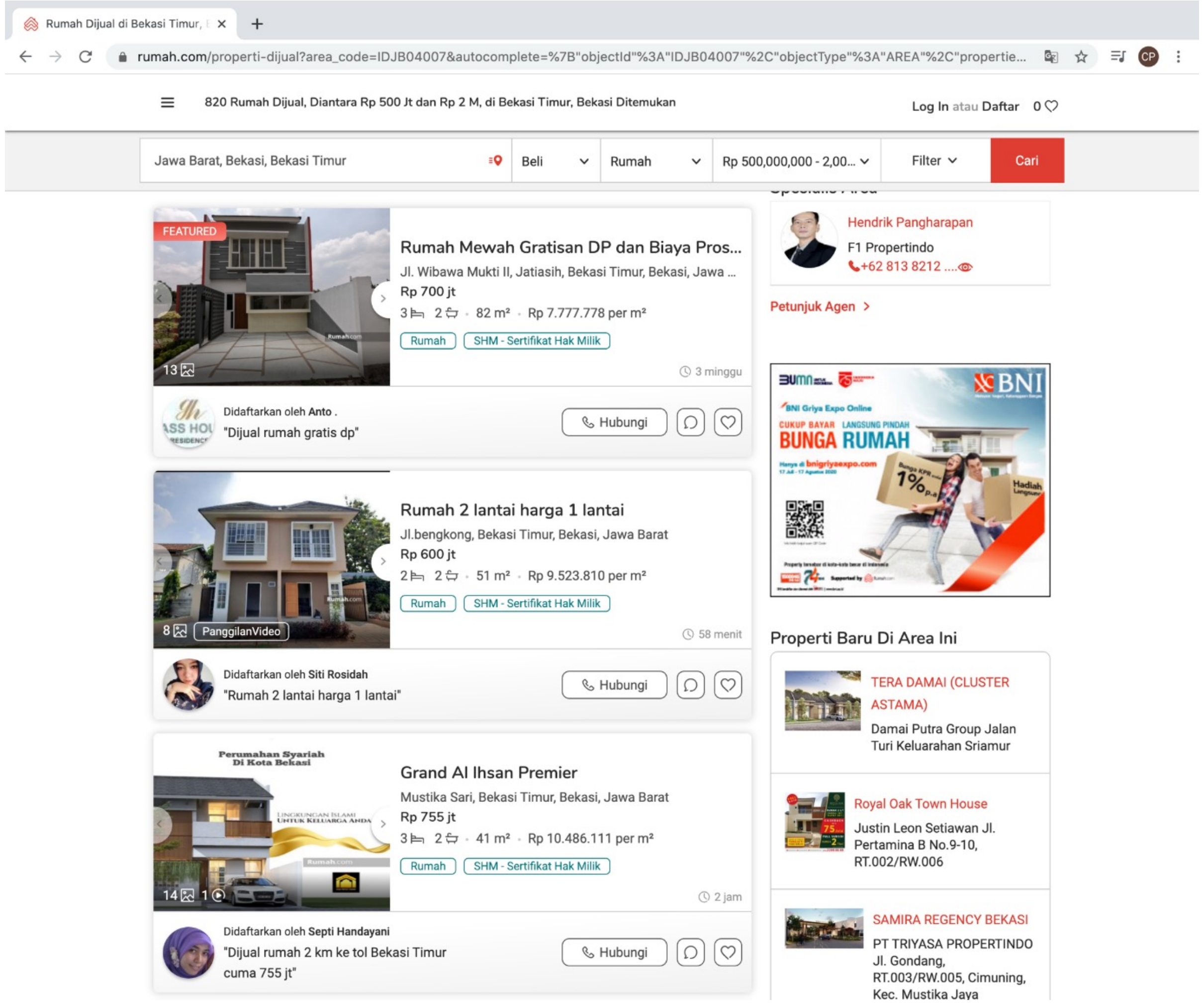




Section 1

Supervised Learning

House Price Prediction in East Bekasi



Size (m^2)	Bed	Bath	…	Price (IDR million)
82	3	2	…	700
51	2	2	…	600
41	3	2	…	755
…	…	…	…	…

Size (m^2)	Bed	Bath	…	Price (IDR million)
98	4	3	…	???
45	3	1	…	???
36	2	1	…	???
…	…	…	…	…

Source: rumah.com

House Price Prediction: Simplified

Training Data

x Size (m^2)	y Price (IDR million)
82	700
51	600
41	755
…	…

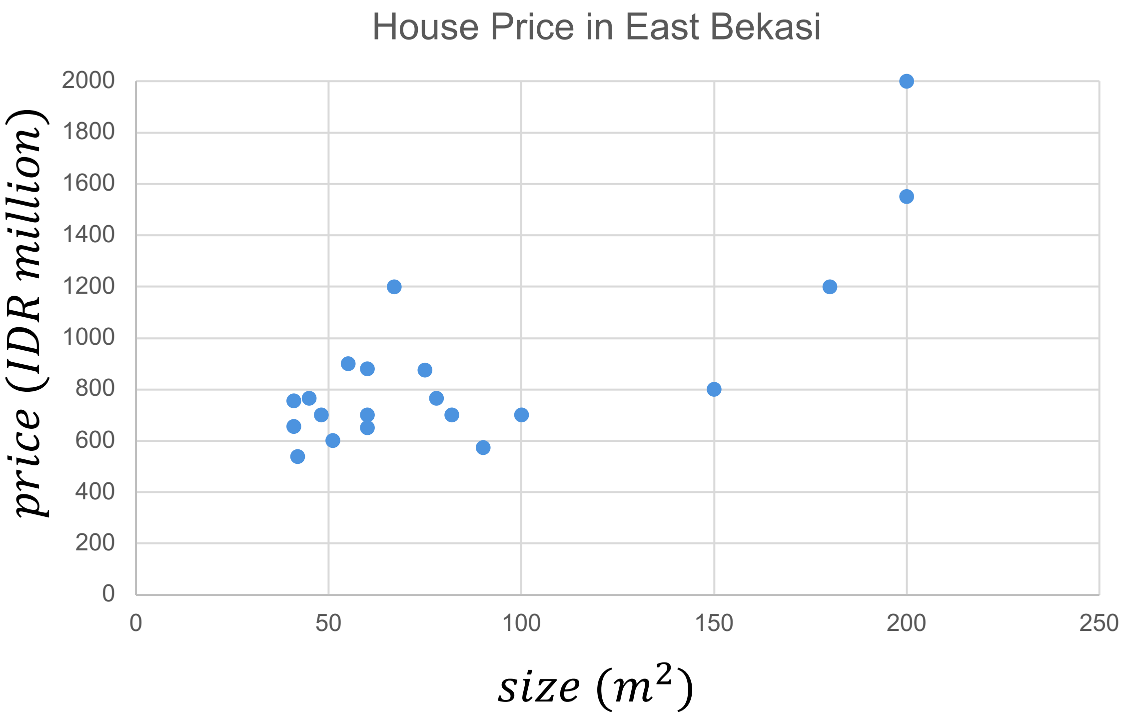
Testing Data

x Size (m^2)	y Price (IDR million)
98	???
45	???
36	???
…	…

$$x = size\ ;\ y = price$$

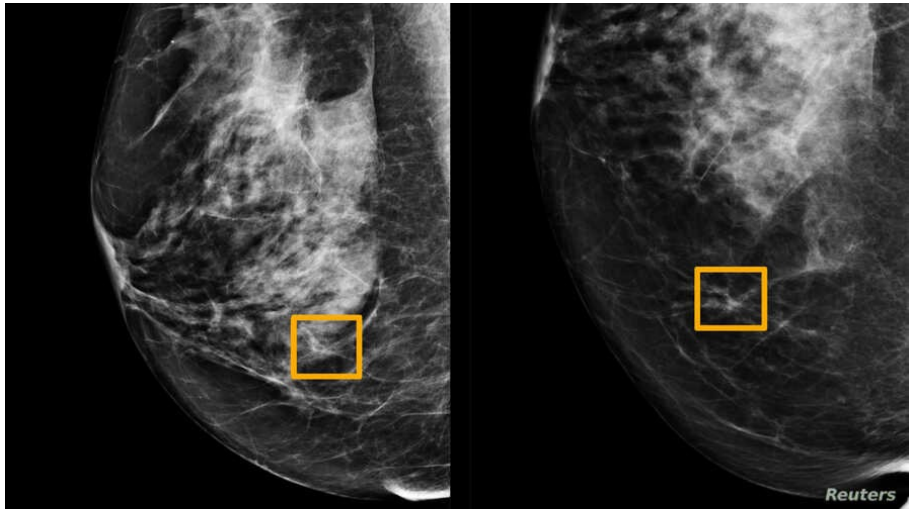
$$h(x) = y$$

Because y is a ***continuos variable****,*
then this will be a ***REGRESSION*** *problem.*



if y was a ***discrete variable*** *(****class****),*
then this would be a ***CLASSIFICATION*** *problem.*

Breast Cancer Diagnostic



Benign (B): not harmful
Malignant (M): deadly

Clump Thickness	Cell Size	Cell Shape	…	Diagnostic
5	4	4	…	B
8	10	10	…	M
4	1	1	…	B
…	…	…	…	…

Clump Thickness	Cell Size	Cell Shape	…	Diagnostic
6	8	8	…	?
3	2	1	…	?
7	4	6	…	?
…	…	…	…	…

Breast Cancer Diagnostic: Simplified

$Training\ Data$

Cell Size (x)	Diagnostic (y)
4	B
10	M
1	B
…	…

$Testing\ Data$

Cell Size (x)	Diagnostic (y)
8	?
2	?
4	?
…	…

$x = cell\ size$
$y = diagnostic$

$h(x) = y$

Because y is a ***discrete variable (class)****,*
then this is a ***CLASSIFICATION*** *problem.*

Supervised Learning Procedure

Training Data

Size (m^2)	Price (IDR million)
82	700
51	600
41	755
…	…

Training Data

Machine Learning Algorithm

Trained Model

$h(x) = y$

Testing Data

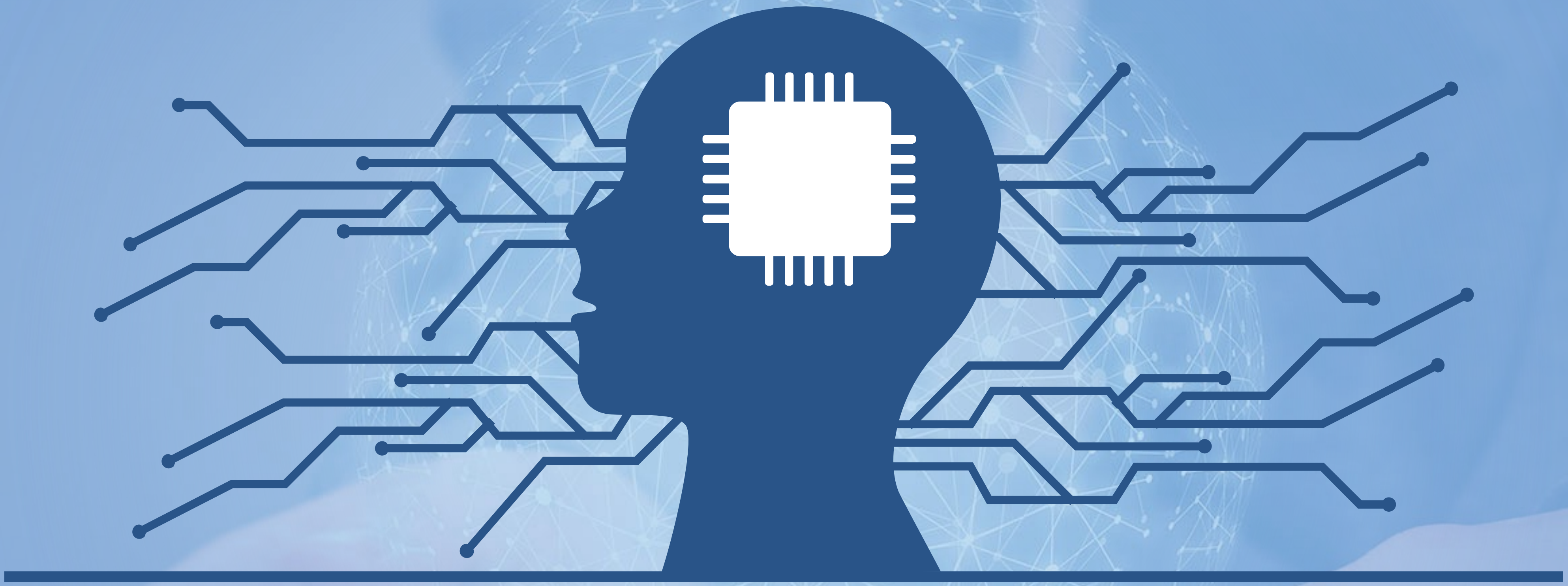
x

Prediction

y

Testing Data

Size (m^2)	Price (IDR million)
98	???
45	???
36	???
…	…



Section 2

Model Representation

Notation

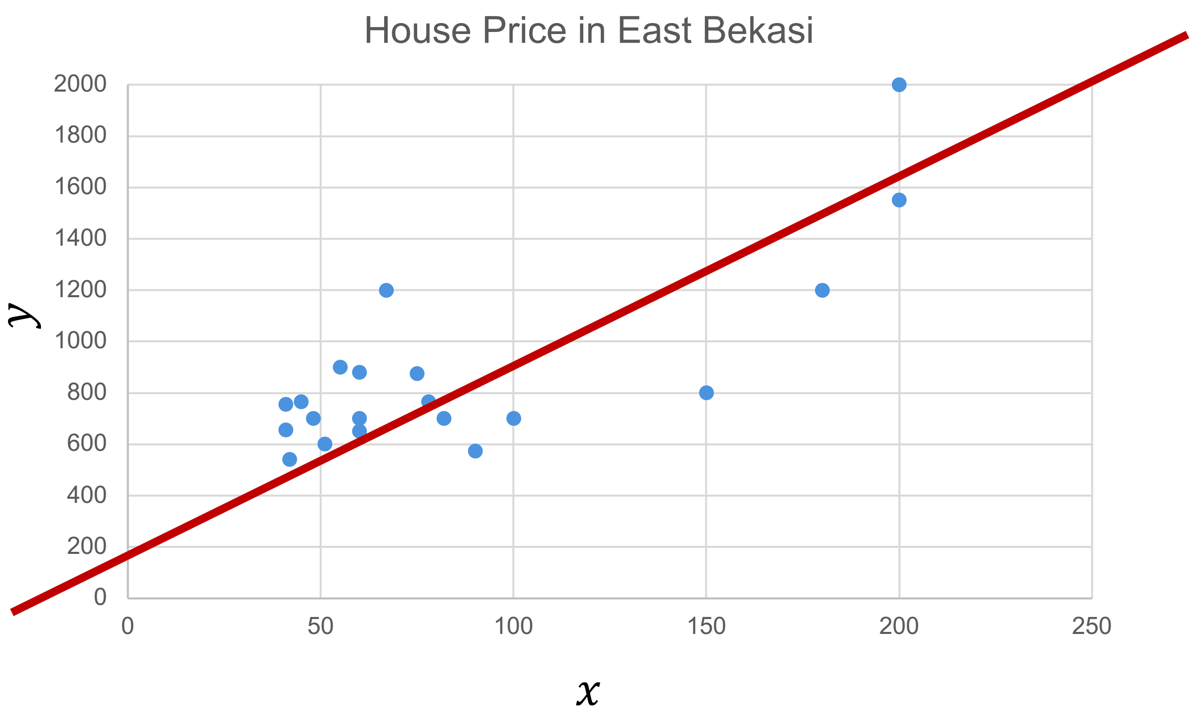
Training Data of Housing Prices (East Bekasi)

	x	y
	size in m²	Price (IDR) in million
1	82	700
2	51	600
3	41	755
	…	…
m	67	1200

x = features / input variable
y = target / output variable
m = number of training examples
(x, y) = a single training example
$\left(x^{(i)}, y^{(i)}\right)$ = i$^{\text{th}}$ training example

$x^{(1)} = 82$
$x^{(2)} = 51$
$y^{(2)} = 600$
$(x^{(3)}, y^{(3)}) = (41, 755)$

Linear Model



$$h_\theta(x) = \theta_0 + \theta_1\, x$$

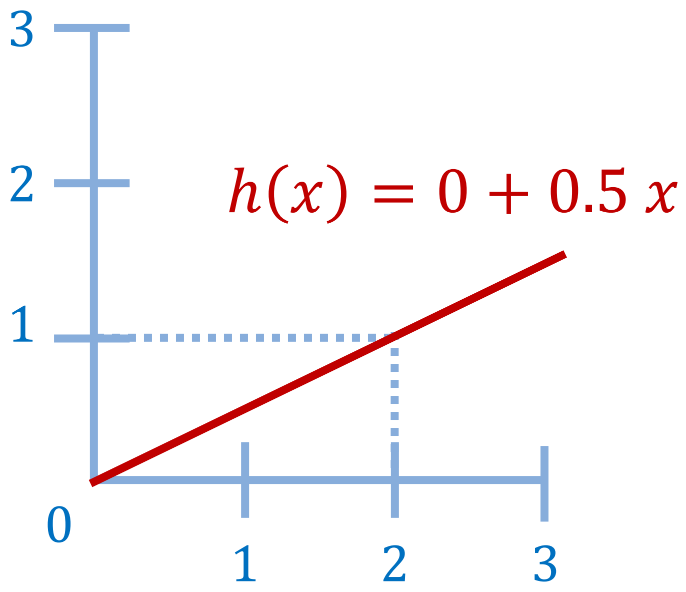
h : a linear model, maps from x to y

θ_0 , θ_1 : parameters

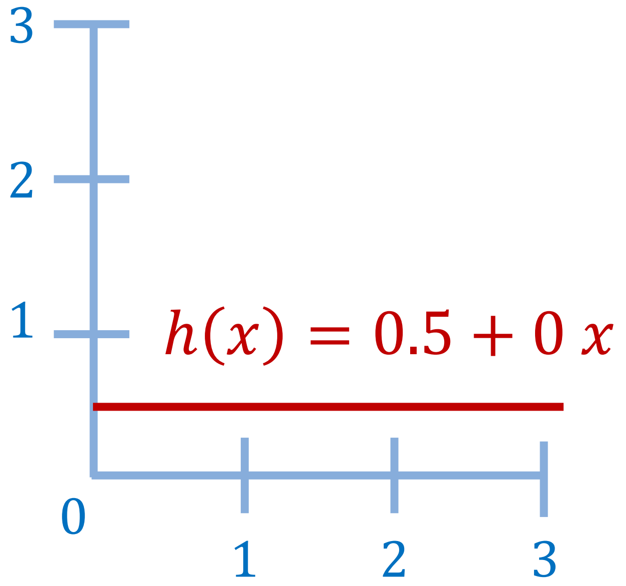
x : input

Hypothesis

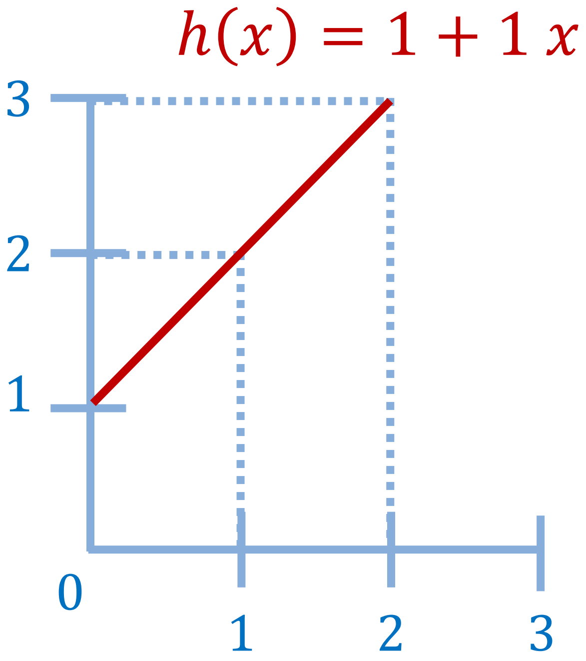
$$h_\theta(x) = \theta_0 + \theta_1\, x$$



$\theta_0 = 0$
$\theta_1 = 0.5$

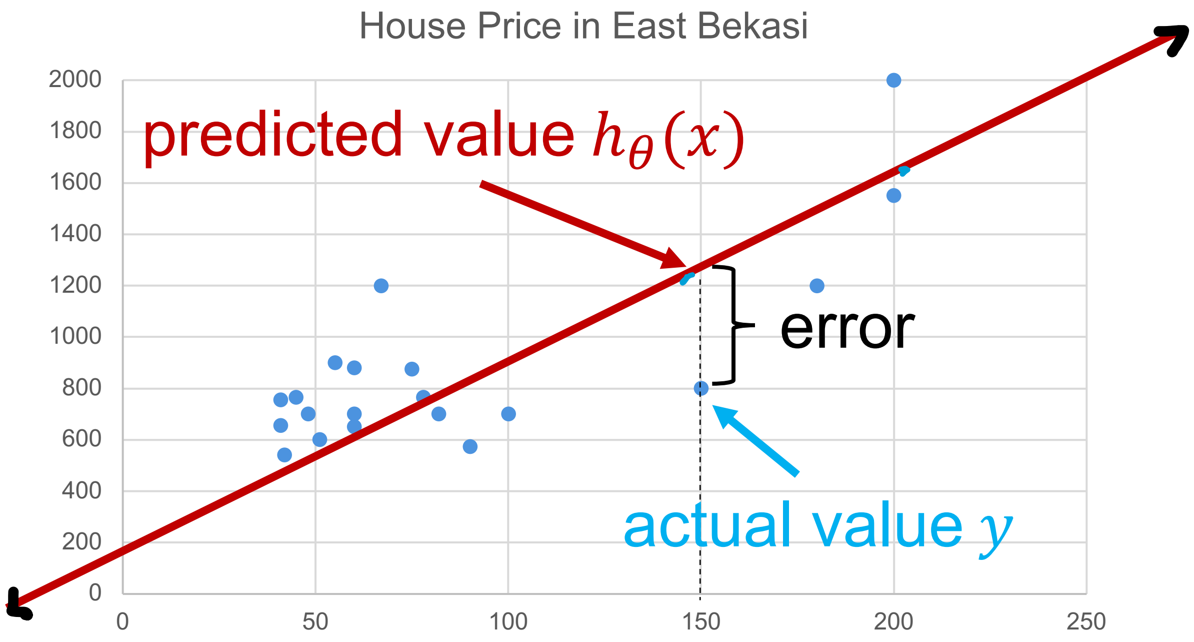


$\theta_0 = 0.5$
$\theta_1 = 0$



$\theta_0 = 1$
$\theta_1 = 1$

Mean Squared Error (MSE)



$$h_\theta(x) = \theta_0 + \theta_1\, x$$

Error = predicted – actual

$= h_\theta(x) - y$

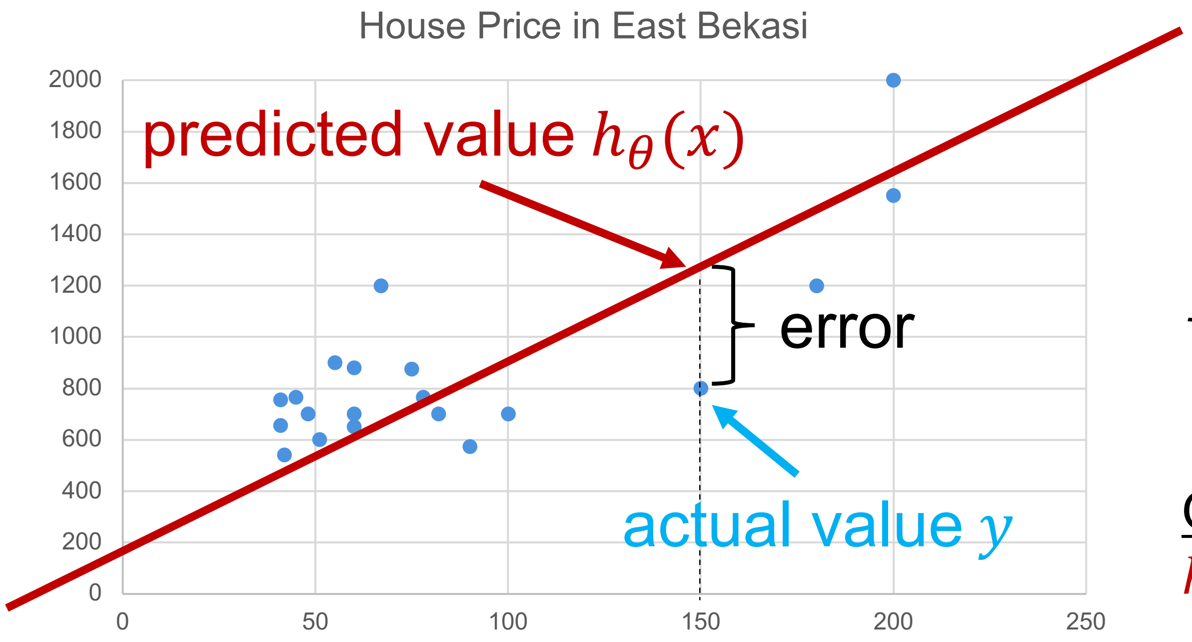
Squared Error = Error^2

$= (h_\theta(x) - y)^2$

MSE = $\frac{1}{m}\{\, (h_\theta(x^{(1)}) - y^{(1)})^2 + (h_\theta(x^{(2)}) - y^{(2)})^2 + \ldots + \; + \; (h_\theta(x^{(m)}) - y^{(m)})^2 \,\}$

$= \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

Cost Function $J(\theta_0, \theta_1)$



$$h_\theta(x) = \theta_0 + \theta_1\, x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

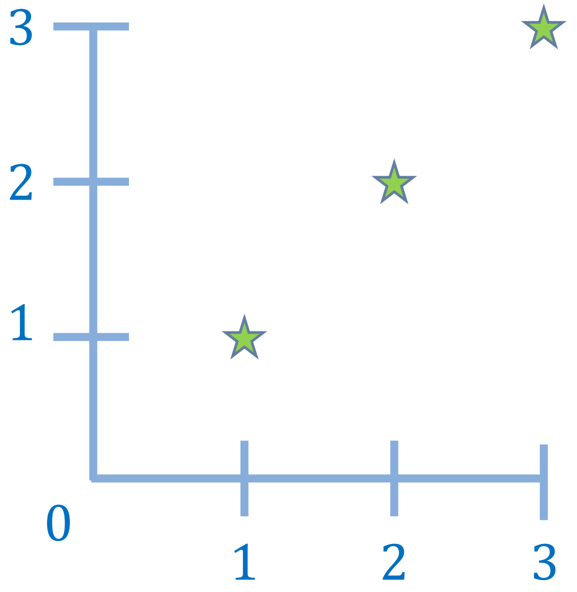
Objective: Find θ_0 , θ_1 such that
$h_\theta(x)$ is close to y for training example (x, y)

$$\underset{\theta_0, \theta_1}{\text{minimize}}\; J(\theta_0, \theta_1)$$

Cost Function – Example in a Simplified Model

Training Data

x	*y*
1	1
2	2
3	3



Simplified Model: $h_\theta(x) = \theta_1 \, x$

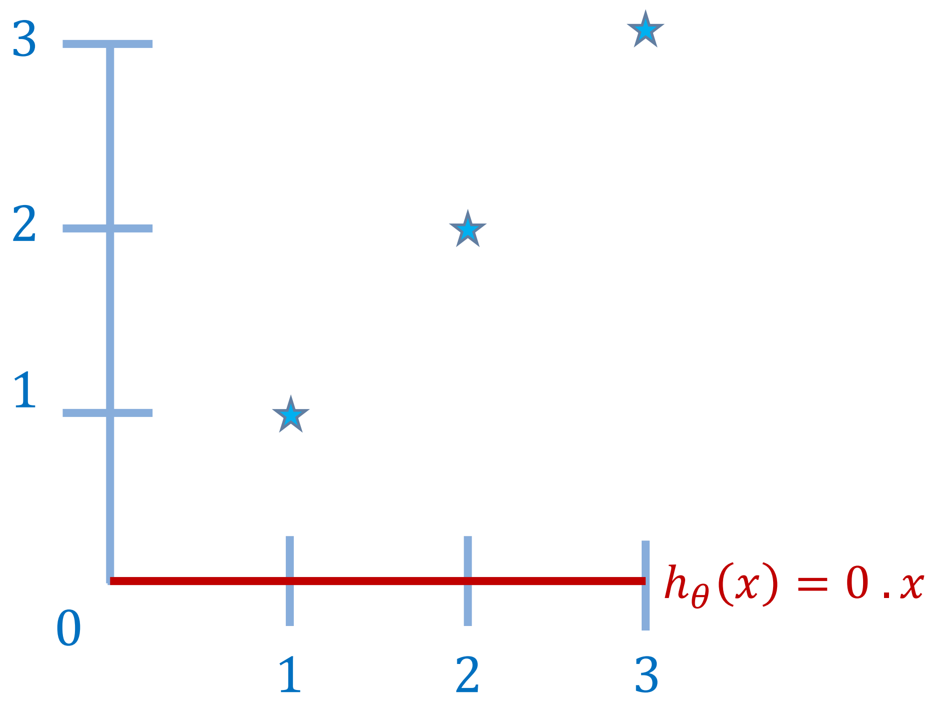
Parameter: θ_1

Cost Function: $J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$

Objective: $\underset{\theta_1}{\text{minimize}} \; J(\theta_1)$

Cost Function – Example in a Simplified Model

$\theta_1 = 0.0$



$h_\theta(1) = 0\cdot$
$h_\theta(2) = 0\,.$
$h_\theta(3) = 0\,.$

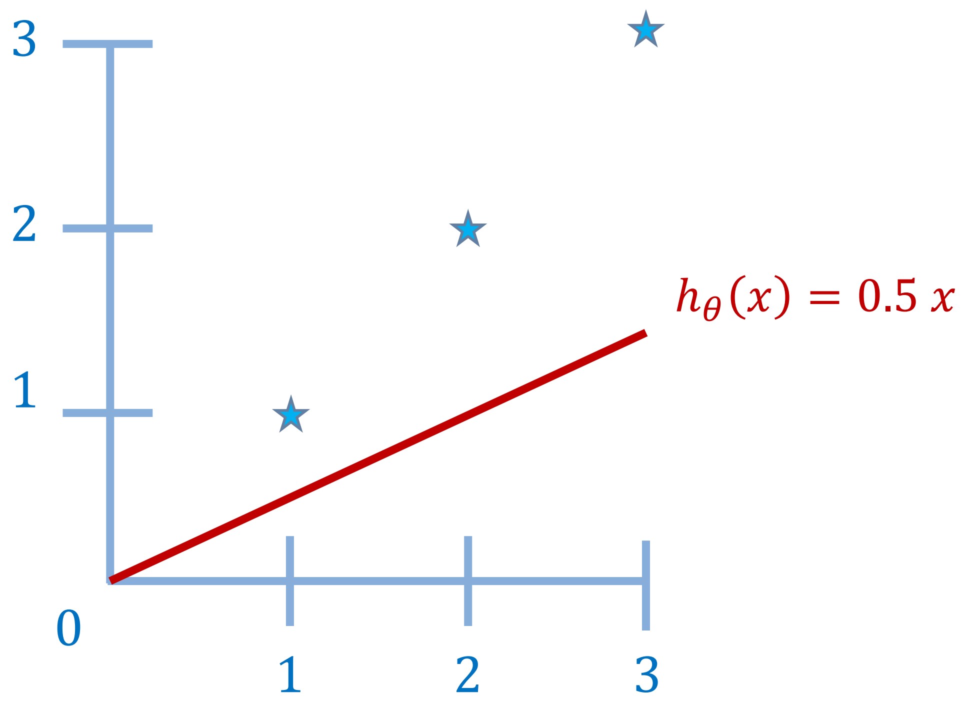
$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$x^{(i)}$	$y^{(i)}$	$h_\theta(x^{(i)})$	$h_\theta(x^{(i)}) - y^{(i)}$	$(h_\theta(x^{(i)}) - y^{(i)})^2$
1	1	0	0 – 1 = -1	$(-1)^2 = 1$
2	2	0	0 – 2 = -2	$(-2)^2 = 4$
3	3	0	0 – 3 = -3	$(-3)^2 = 9$
$\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$				1 + 4 + 9 = 14
$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$				14 / 6 = 2.33

$J(0.0) = 2.33$

Cost Function – Example in a Simplified Model

$\theta_1 = 0.5$



$h_\theta(1) = 0.5$
$h_\theta(2) = 1.0$
$h_\theta(3) = 1.5$

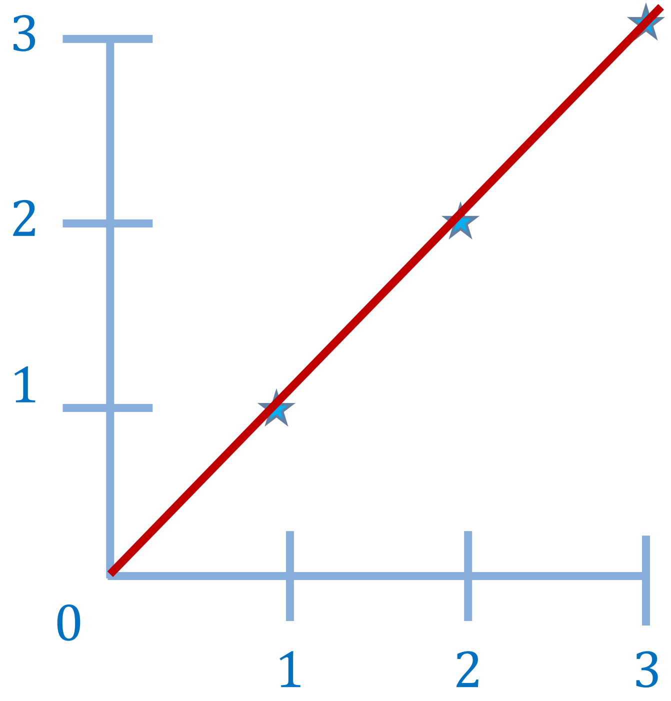
$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$x^{(i)}$	$y^{(i)}$	$h_\theta\left(x^{(i)}\right)$	$h_\theta\left(x^{(i)}\right) - y^{(i)}$	$\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$
1	1	0.5	0.5 – 1 = -0.5	$(-0.5)^2 = 0.25$
2	2	1.0	1.0 – 2 = -1.0	$(-1.0)^2 = 1.00$
3	3	1.5	1.5 – 3 = -1.5	$(-1.5)^2 = 2.25$
$\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$				0.25 + 1.00 + 2.25 = 3.50
$\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$				3.50 / 6 = 0.58

$J(0.5) = 0.58$

Cost Function – Example in a Simplified Model

$\theta_1 = 1.0$



$h_\theta(x) = 1 . x$

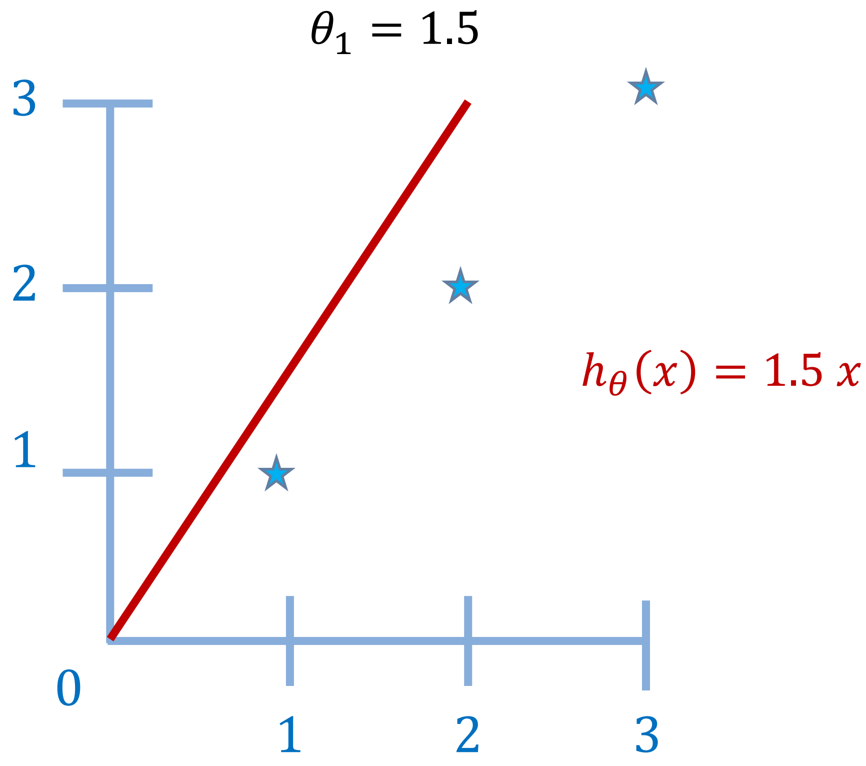
$h_\theta(1) = 1$
$h_\theta(2) = 2$
$h_\theta(3) = 3$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$x^{(i)}$	$y^{(i)}$	$h_\theta(x^{(i)})$	$h_\theta(x^{(i)}) - y^{(i)}$	$(h_\theta(x^{(i)}) - y^{(i)})^2$
1	1	1	0	0
2	2	2	0	0
3	3	3	0	0
$\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$				0
$\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$				0

$J(1.0) = 0$

Cost Function – Example in a Simplified Model



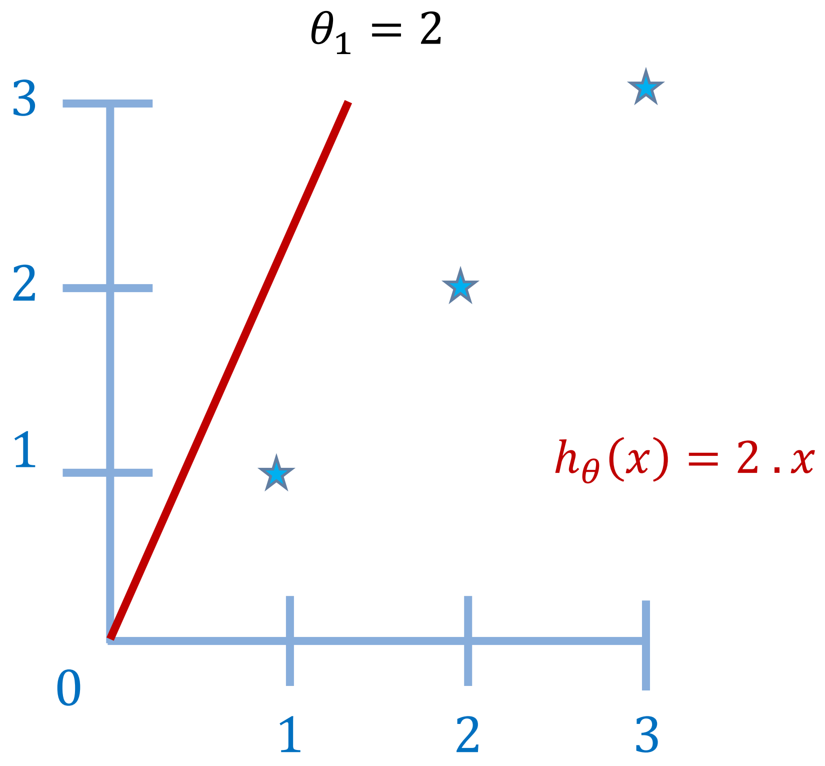
$h_\theta(1) = 1.5$
$h_\theta(2) = 3.0$
$h_\theta(3) = 4.5$

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$x^{(i)}$	$y^{(i)}$	$h_\theta\left(x^{(i)}\right)$	$h_\theta\left(x^{(i)}\right) - y^{(i)}$	$\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$
1	1	1.5	0.5	0.25
2	2	3.0	1.0	1.00
3	3	4.5	1.5	2.25
$\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$				3.50
$\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$				3.50 / 6 = 0.58

$J(1.5) = 0.58$

Cost Function – Example in a Simplified Model



$h_\theta(1) = 2$
$h_\theta(2) = 4$
$h_\theta(3) = 6$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

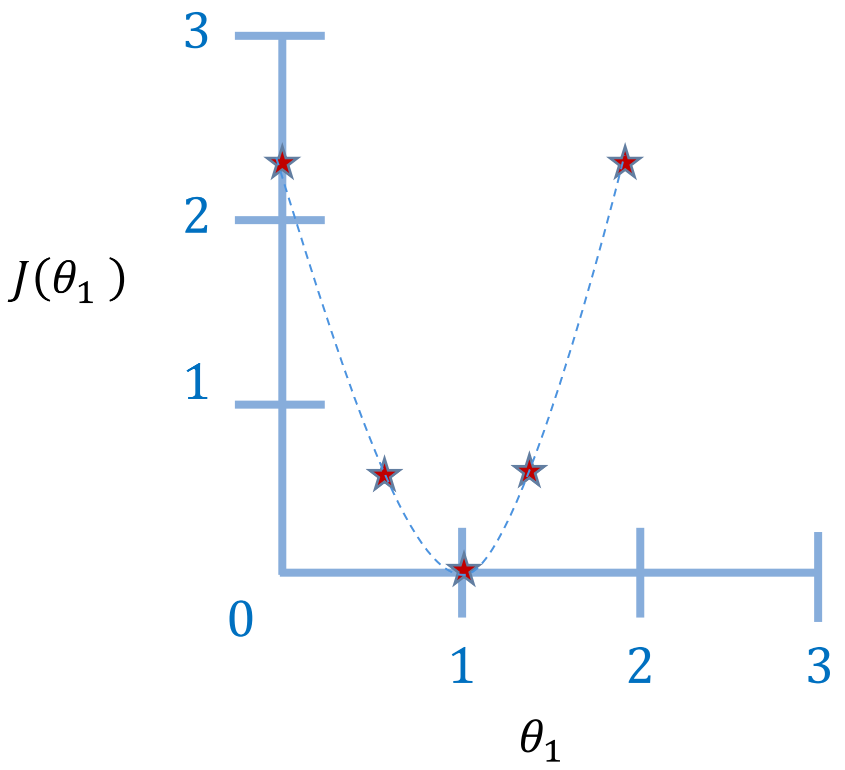
$x^{(i)}$	$y^{(i)}$	$h_\theta\left(x^{(i)}\right)$	$h_\theta\left(x^{(i)}\right) - y^{(i)}$	$\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$
1	1	2	1	1
2	2	4	2	4
3	3	6	3	9
$\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$				14
$\frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$				14 / 6 = 2.33

$J(2) = 2.33$

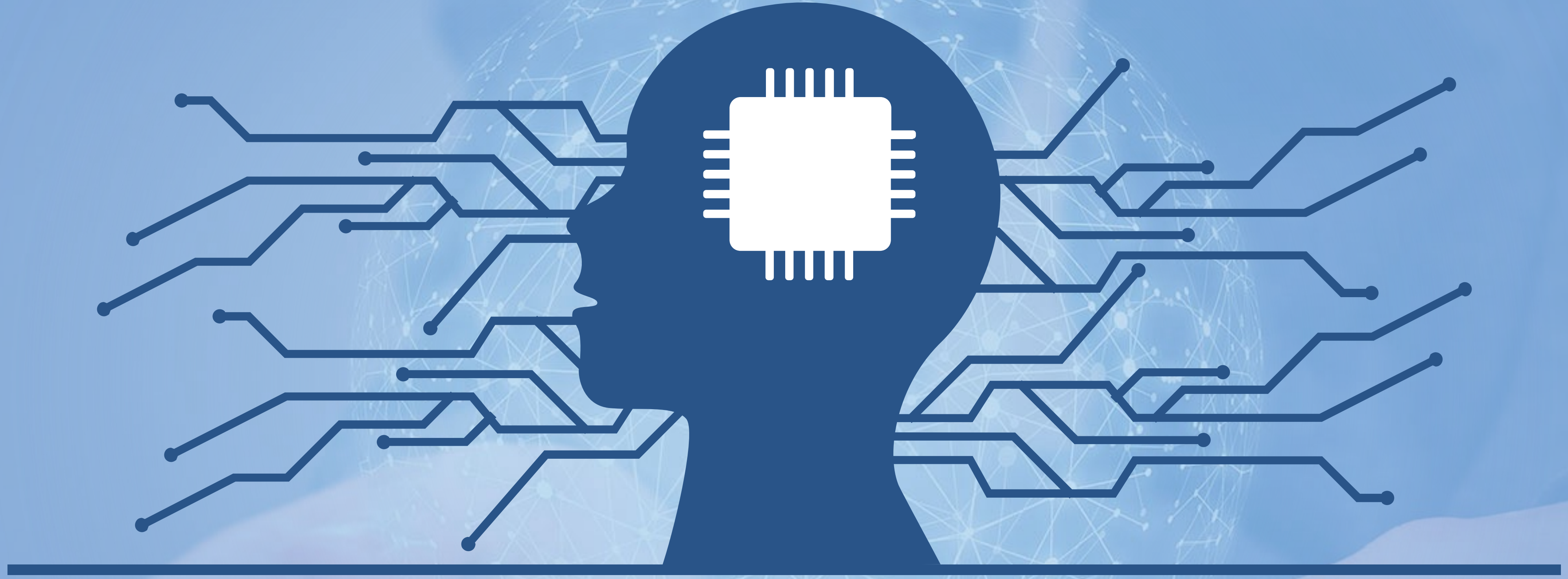
Cost Function – Example in a Simplified Model

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

θ_1	$J(\theta_1)$
0.0	2.33
0.5	0.58
1.0	0.00
1.5	0.58
2.0	2.33



The closer θ_1 to the optimal value, the smaller its cost function $J(\theta_1)$



Section 3

Model Learning

Idea

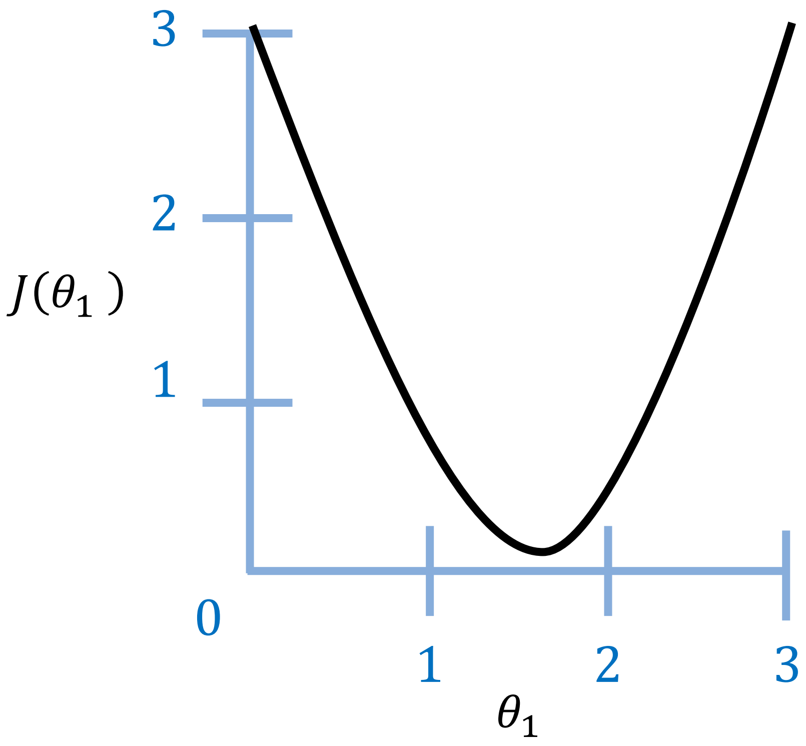
Define a cost function $J(\theta_0, \theta_1)$

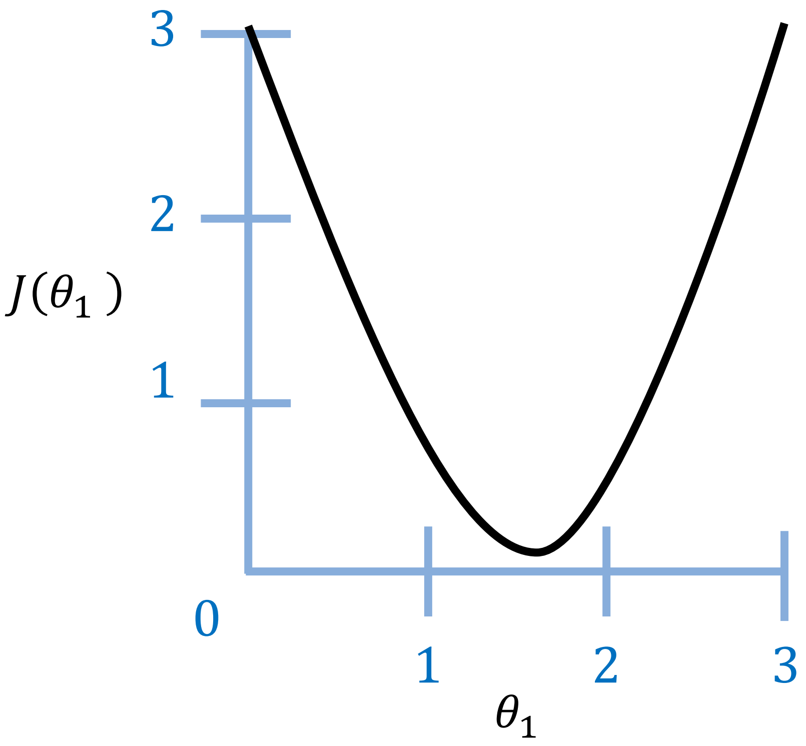
Try $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Plan:

- Initialize θ_0, θ_1.
- Iteratively update θ_0, θ_1 to minimize $J(\theta_0, \theta_1)$ until reach minimum point.

Intuition: what we need to "go down"?





Need two things:

1. Direction
2. Movement

Gradient Descent Algorithm

Repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{\delta}{\delta\theta_0} J(\theta_0, \theta_1)$$

$$\theta_1 := \theta_1 - \alpha \frac{\delta}{\delta\theta_1} J(\theta_0, \theta_1)$$

}

α = "learning rate" (set manually)

$\frac{\delta}{\delta\theta_j} J(\theta_0, \theta_1)$ = "direction" (partial derivative)

Choose Learning Rate α (Manually)

$$\theta_1 := \theta_1 - \alpha \frac{\delta}{\delta\theta_1} J(\theta_0, \theta_1)$$

If α is too small, gradient descent can be slow.

If α is too big, gradient descent can overshoot the minimum (fail to converge).

Compute Partial Derivative

$$\frac{\delta}{\delta\theta_j}J(\theta_0,\theta_1) = \frac{\delta}{\delta\theta_j}\frac{1}{2m}\ \sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$= \frac{\delta}{\delta\theta_j}\frac{1}{2m}\ \sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

$$\frac{\delta}{\delta\theta_0}J(\theta_0,\theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\frac{\delta}{\delta\theta_1}J(\theta_0,\theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

AI

Gradient Descent for Linear Regression

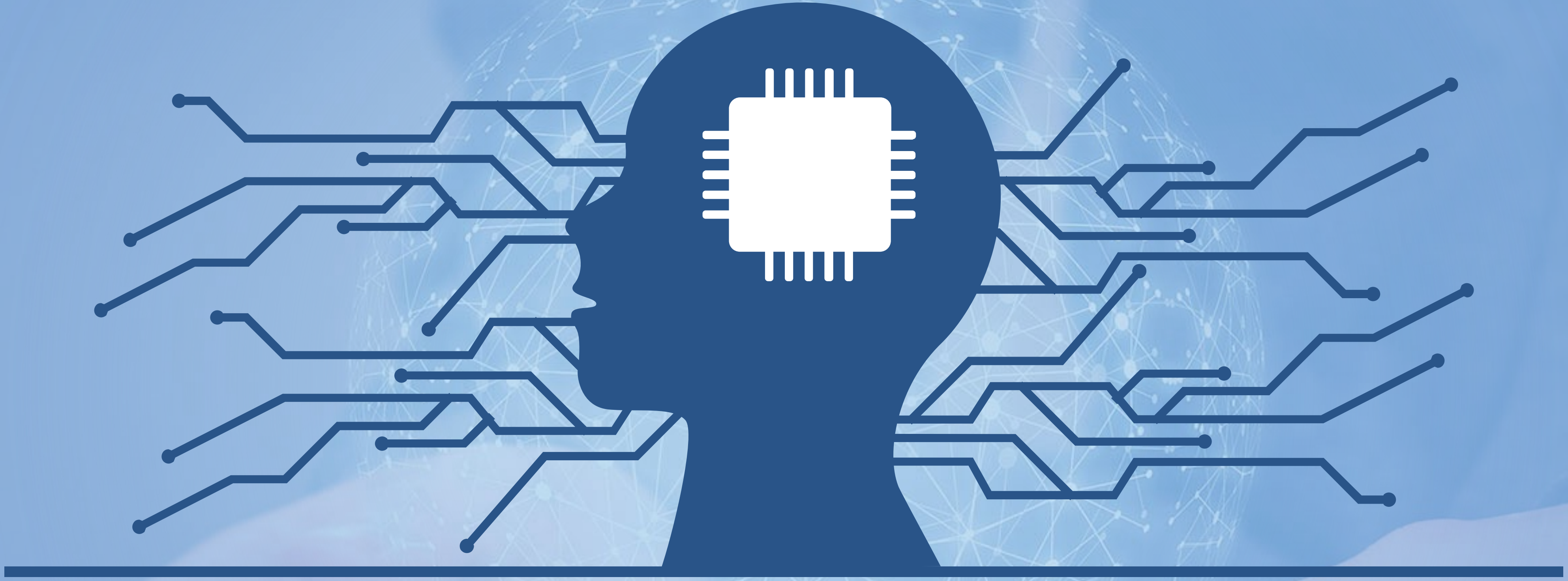
Repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x^{(i)}$$

}

update
θ_0 and θ_1
simultaneously



Thank you!

Acknowledgment

Slide Template:

https://www.free-powerpoint-templates-design.com/

Slide Creator:

Chandra Prasetyo Utomo

Reference:

Machine Learning course (Coursera) from Andrew Ng