AI VIETNAM
Free AI Course 2020

Computational Graph and Linear Regression

Quang-Vinh Dinh
Ph.D. in Computer Science

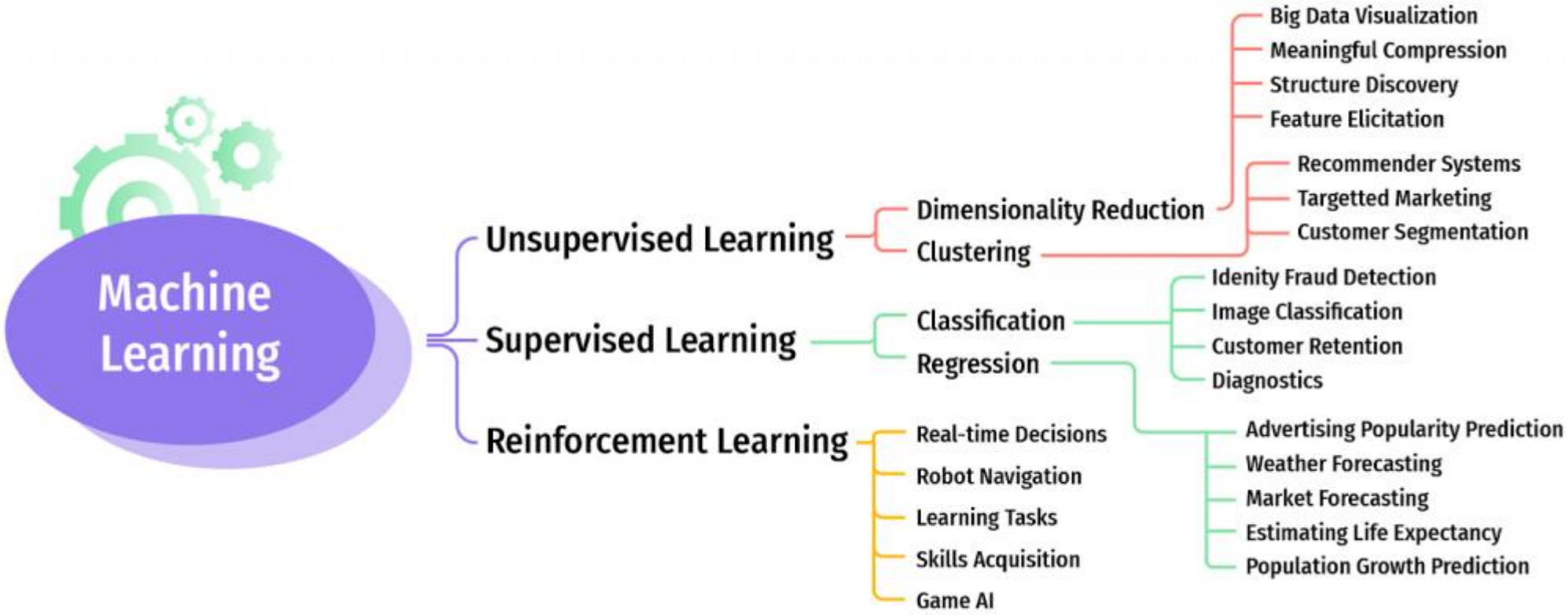
Year 2020

Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
- Generalized formulae

Machine Learning

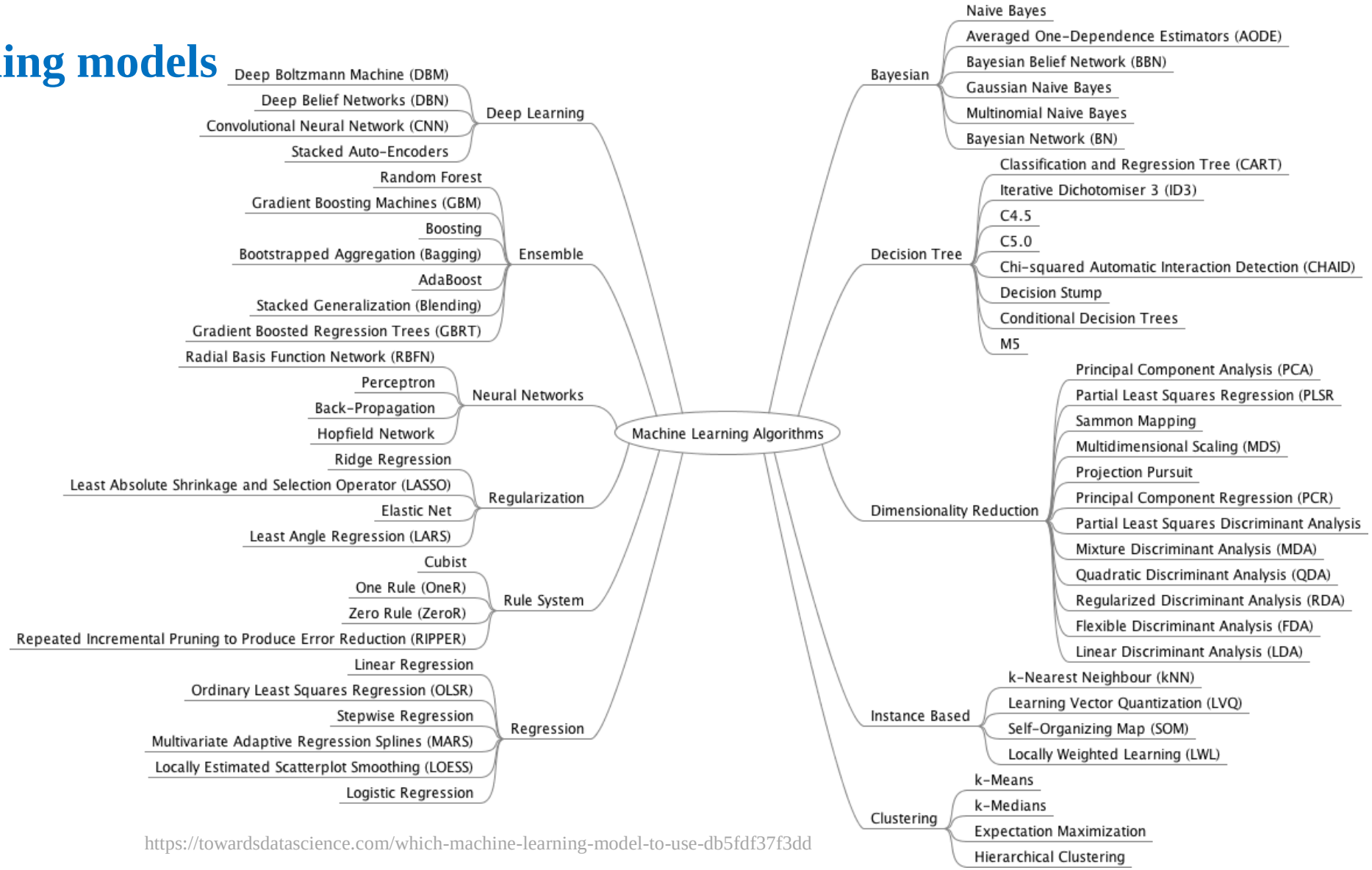
❖ Overview



https://idapgroup.com/blog/types-of-machine-learning-out-there/

Machine Learning

Machine learning models



https://towardsdatascience.com/which-machine-learning-model-to-use-db5fdf37f3dd

Machine Learning

❖ Supervised learning

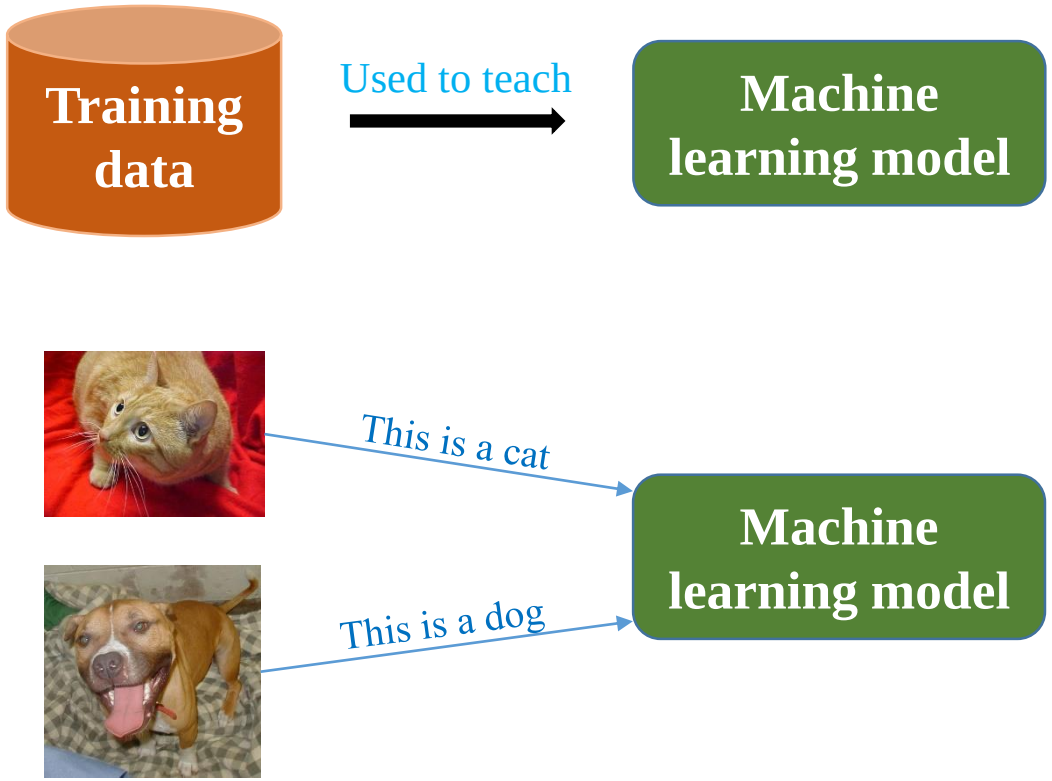
Input and output data is provided

- Training data
- Cats
- Dogs

From Cat-Dog dataset

Machine Learning

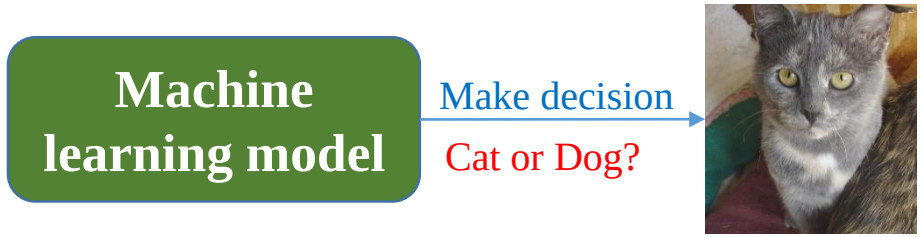
❖ Supervised learning



Training phase

From Cat-Dog dataset

Testing data (≠ training data)



Testing phase

Machine Learning

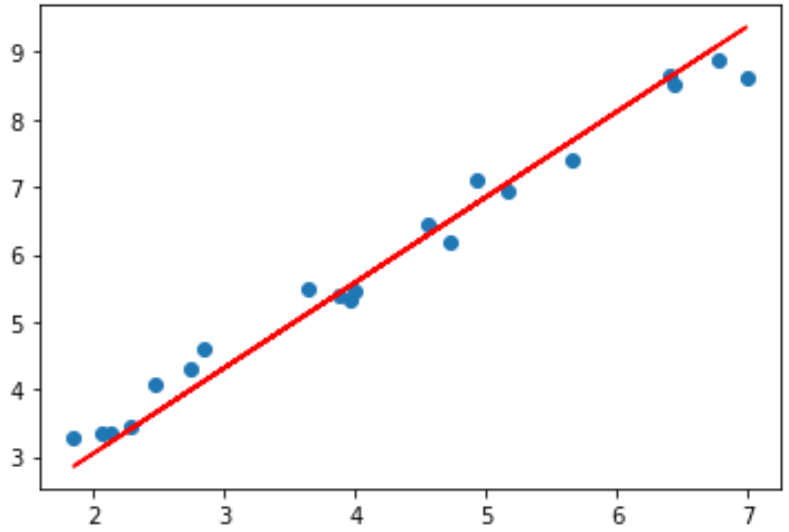
❖ Supervised learning

❖Regression (prediction)

Linear regression models ← Linear equations

Linear equation = $w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$

where $\boldsymbol{w}$ is a weight vector
and $\boldsymbol{x}$ is feature vector



Machine Learning

❖ Supervised learning

❖ Linear Regression: Data processing

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

House price data

Features			Label
TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9

Advertising data

Model: y = $w_1 x_1 + b$

price = a $* area + b$

Model: y = $w_1 x_1 + w_2 x_2 + w_3 x_3 + b$

Sale = $w_1 * TV + w_2 * Radio + w_3 * Newspaper + b$

Machine Learning

❖ Supervised learning

❖ Linear Regression: Data processing

Boston House Price Data

Features | Label

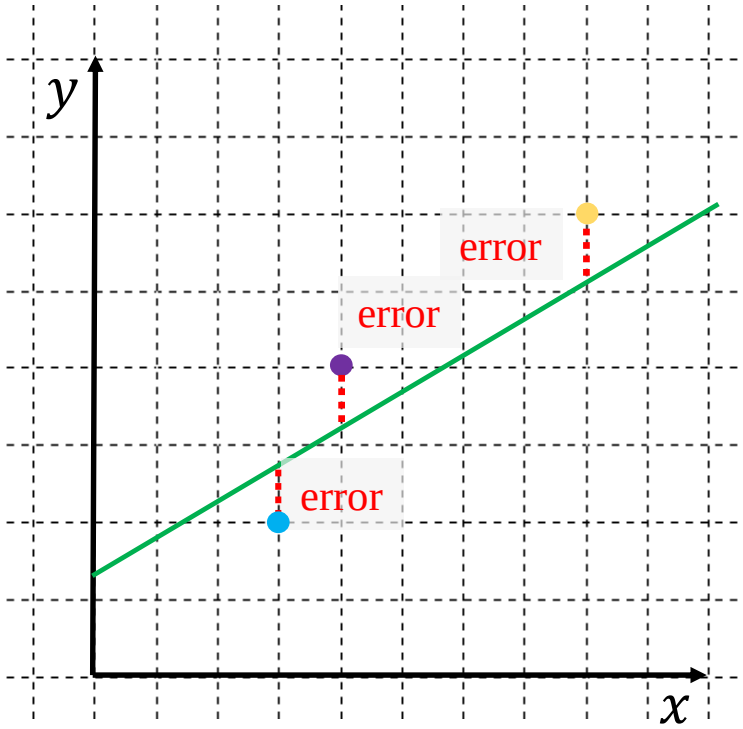
crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9

$$\text{medv} = w_1 * x_1 + \cdots + w_{13} * x_{13} + b$$

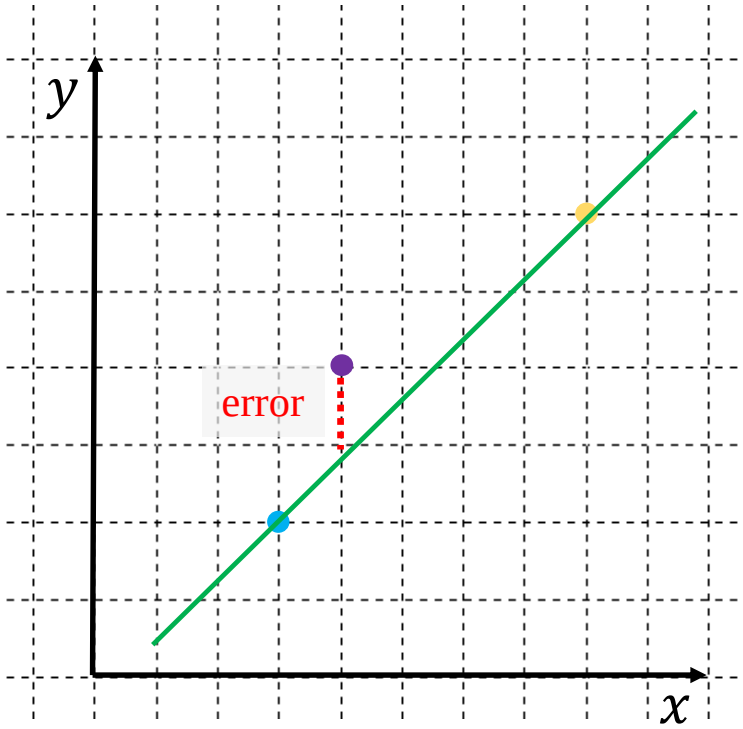
Machine Learning

❖ Linear regression

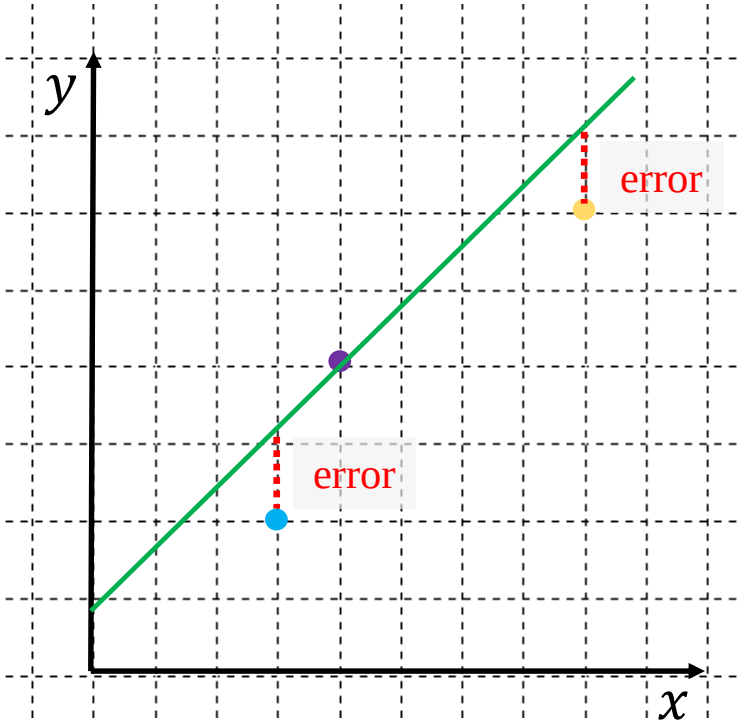
❖Aim to fit data



$y = a_1 x + b_1$



$y = a_2 x + b_2$



$y = a_3 x + b_3$

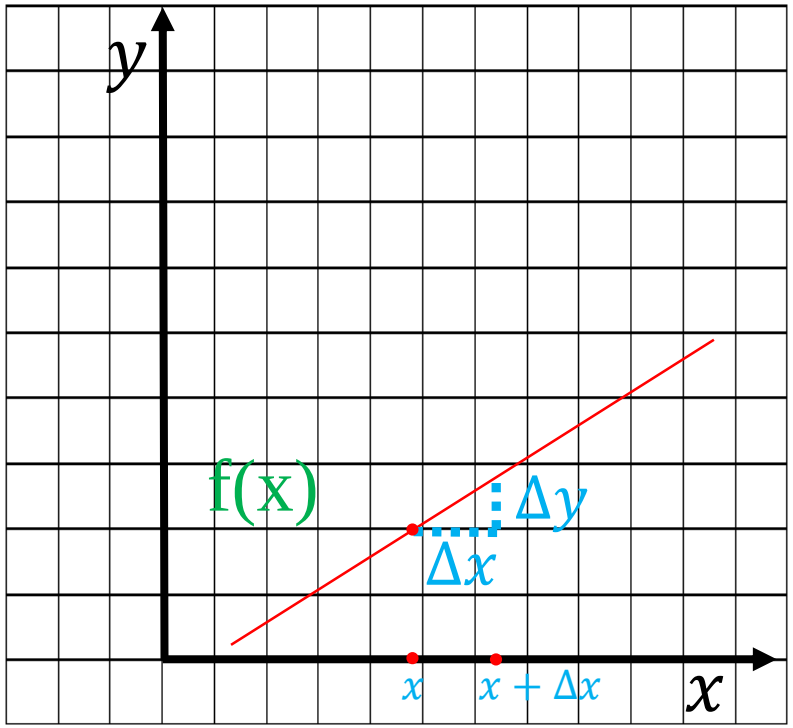
How to find w and b that have the smallest error?

Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
- Generalized formulae

Derivative/Gradient

❖ A cue to optimize a function



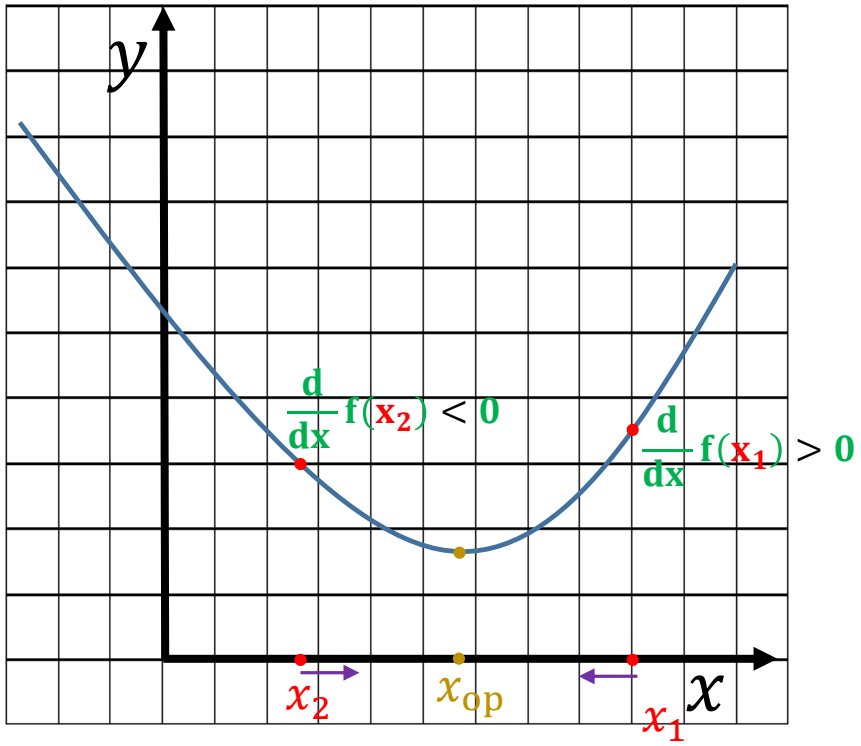
$$\text{Đạo hàm} = \frac{Thay\ đổi\ theo\ y}{Thay\ đổi\ theo\ x} = \frac{\Delta y}{\Delta x}$$

$$\frac{d}{dx} f(x) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Δx cần tiến về 0 để đường tiếp tuyến tiến về hàm f(x) trong vùng lân cận tại x

Derivative/Gradient

❖ A cue to optimize a function



Quan sát: x_{op} ở vị trí ngược hướng đạo hàm tại x_1 và x_2

Cách xử lý việc di chuyển ngược hướng đạo hàm cho x_1 và x_2 (để tìm x_{op}) khác nhau hình thành các thuật toán tối ưu hóa khác nhau

Cách cập nhật giá trị x đơn giản

$$x = x - \eta \frac{d}{dx} f(x)$$

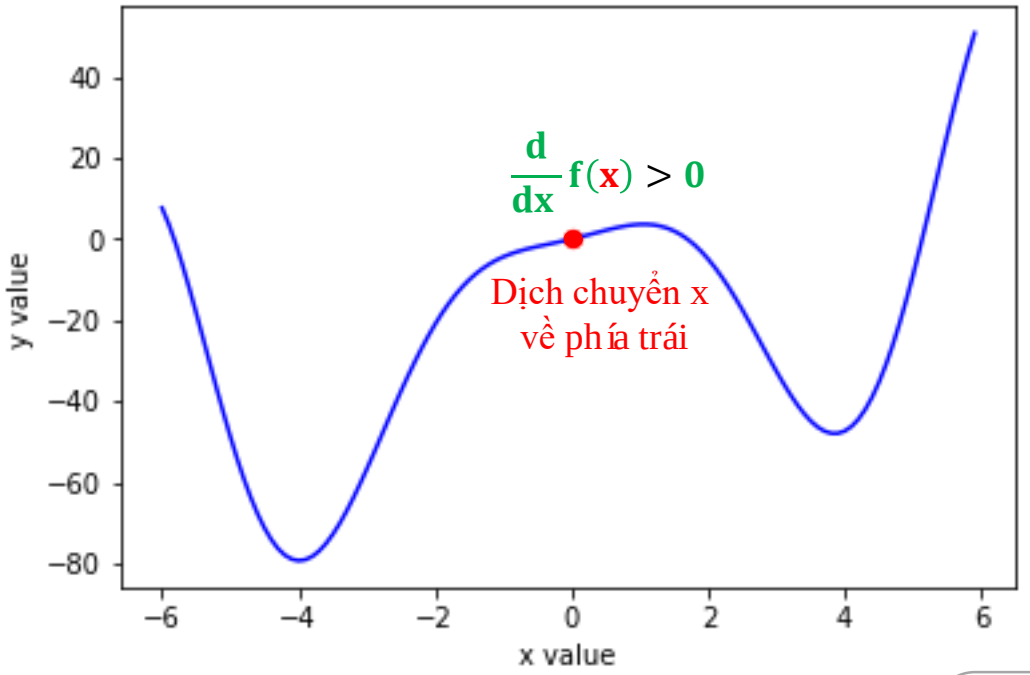
Đạo hàm tại x

Trọng số

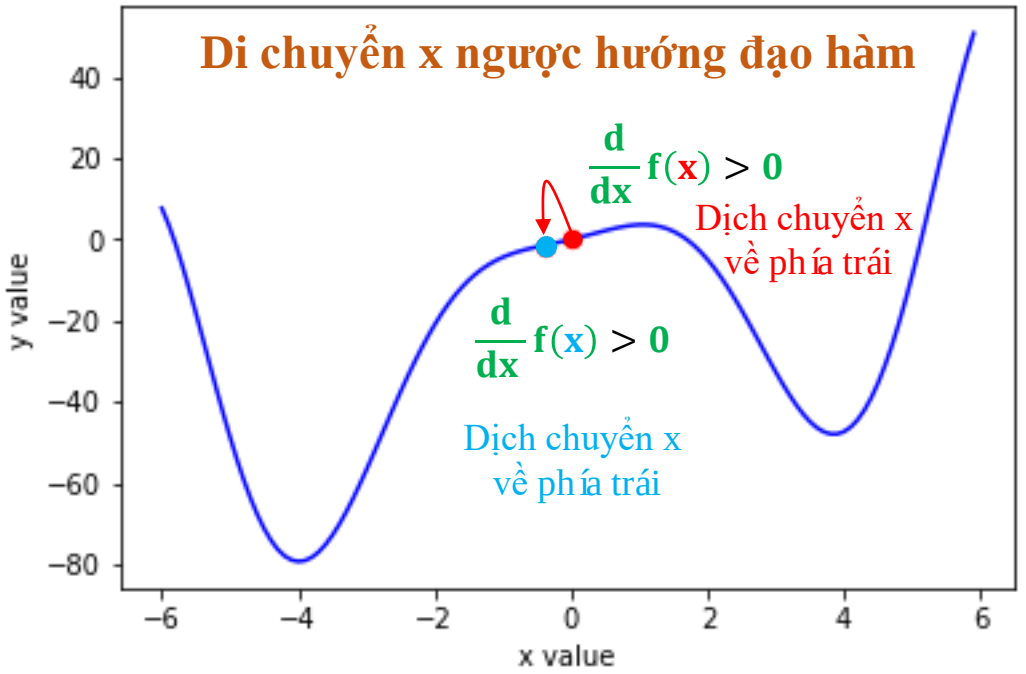
Derivative/Gradient

❖ A cue to optimize a function

Khởi tạo giá trị x



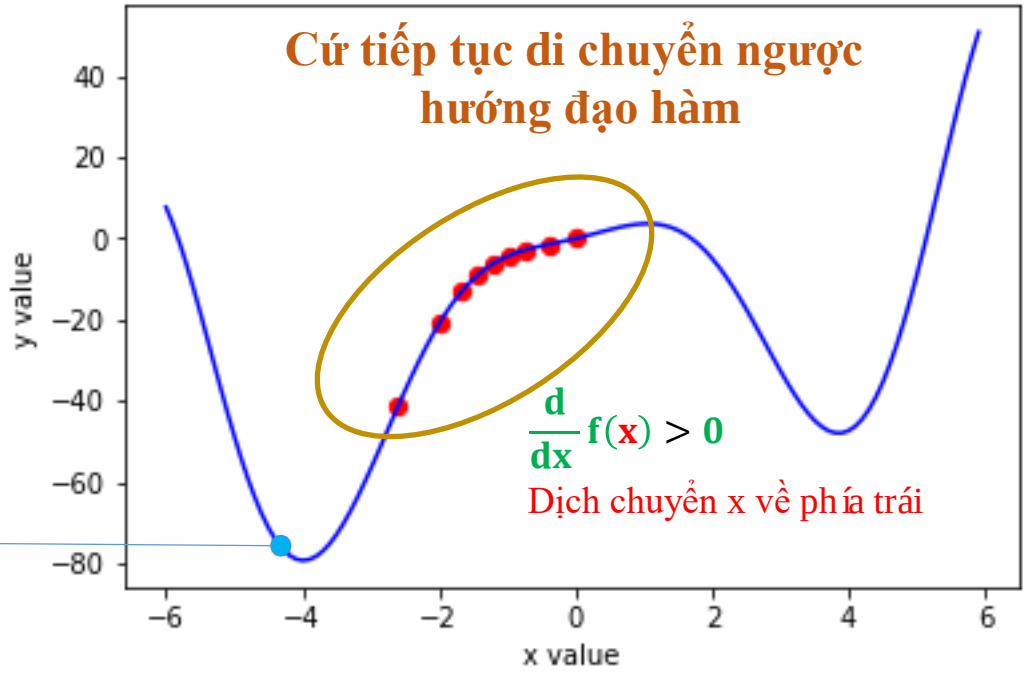
Di chuyển x ngược hướng đạo hàm



$\frac{d}{dx} f(x) < 0$

Dịch chuyển x về phía phải

Cứ tiếp tục di chuyển ngược hướng đạo hàm

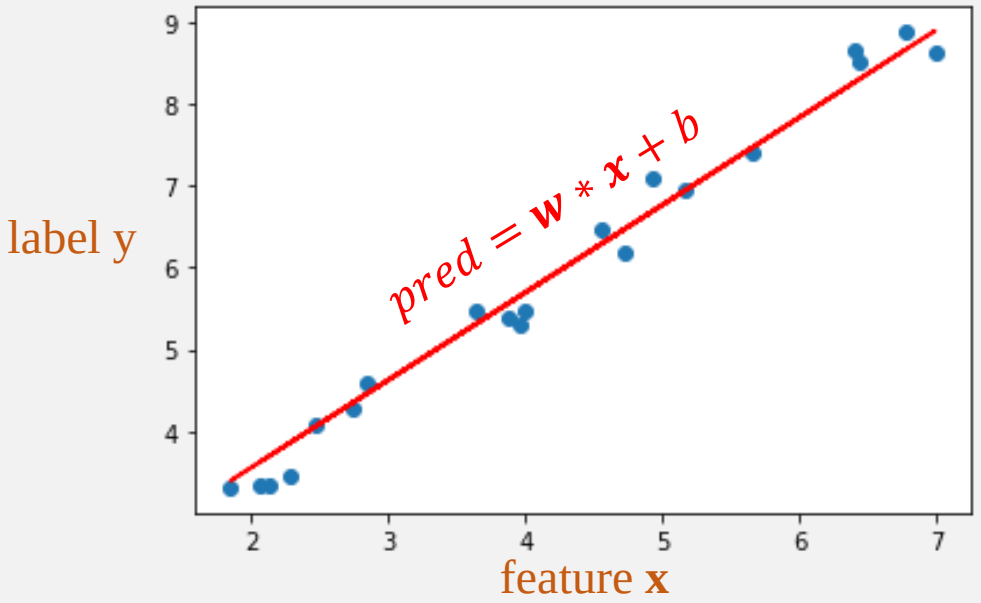


Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
- Generalized formulae

Linear Regression

Model the relationship between feature x and label y



Using a linear equation to fit data

Samples (x, y) are given in advance

Linear equation

$$o = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

where o is a predicted value,
$w_1, w_2, \ldots, w_n$ and b are parameters
and $\boldsymbol{x} = [x_1 \;\; x_2 \;\; \ldots \;\; x_n]^T$ is feature vector.

Error (loss) computation

Idea: compare predicted values o and label values y

Squared loss

$$L(\mathbf{w}, b) = (o - y)^2$$

How to find optimal w and b?

Linear Regression

Linear equation

$$o = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

where o is a predicted value,
$w_1, w_2, \ldots, w_n$ and b are parameters
and $\boldsymbol{x} = [x_1 \;\; x_2 \;\; \ldots \;\; x_n]^T$ is feature vector.

Error (loss) computation

Idea: compare predicted values o and label values y

Squared loss

$$L(\mathbf{w}, b) = (o - y)^2$$

How to find optimal **w** and b?

Use gradient descent to minimize the loss function

Tính đạo hàm

$$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial w_j} = 2x_j(o - y)$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial w_j} = 2(o - y)$$

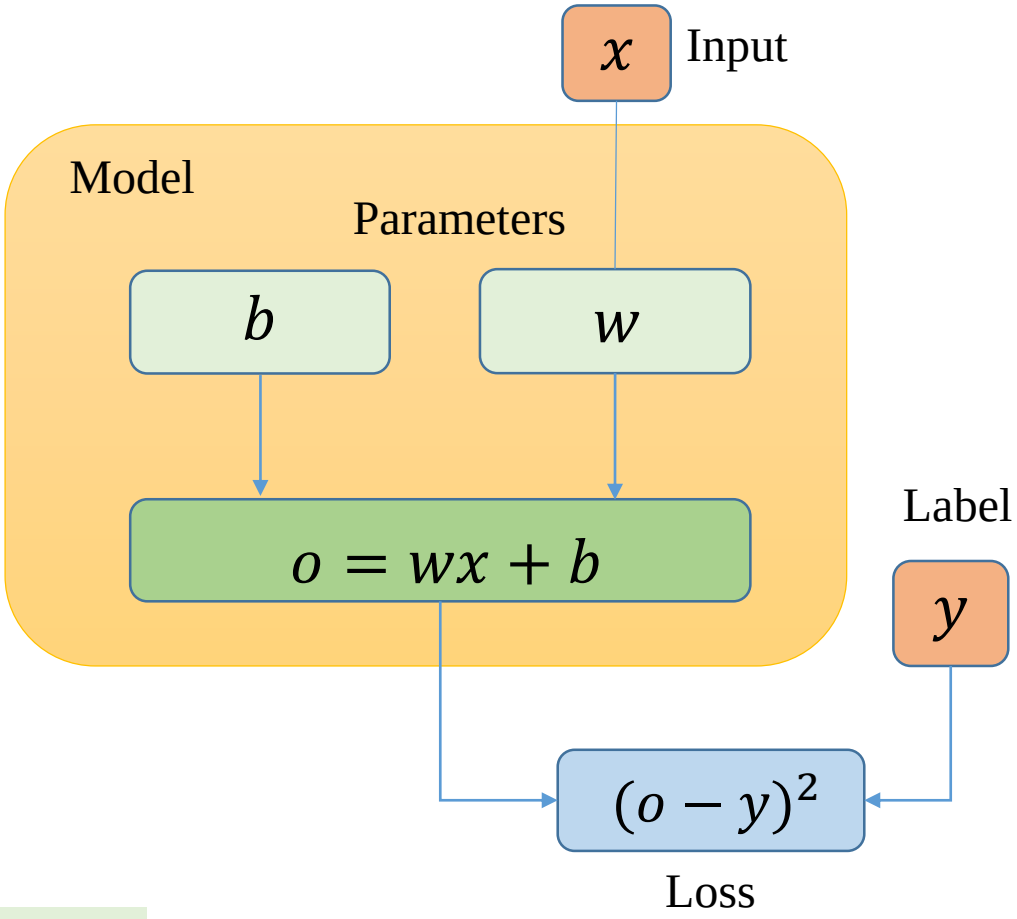
Cập nhật tham số

$$w_j = w_j - \eta L'_{w_j}$$
$$b = b - \eta L'_b$$

η is learning rate

Linear Regression

Diagram



Cheat sheet

Tính output o

$$o = \boldsymbol{wx} + b$$

Tính Loss

$$L = (o-y)^2$$

Tính đạo hàm

$$L'_{w_j} = 2x_j(o-y)$$

$$L'_b = 2(o-y)$$

Cập nhật tham số

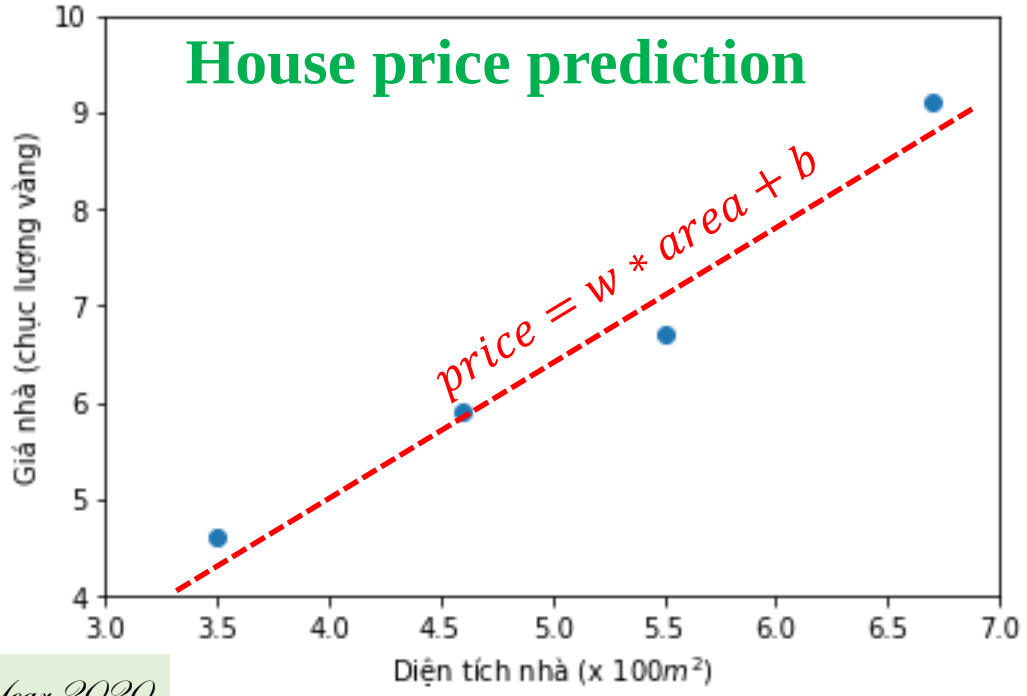
$$w_j = w_j - \eta L'_{w_j}$$

$$b = b - \eta L'_b$$

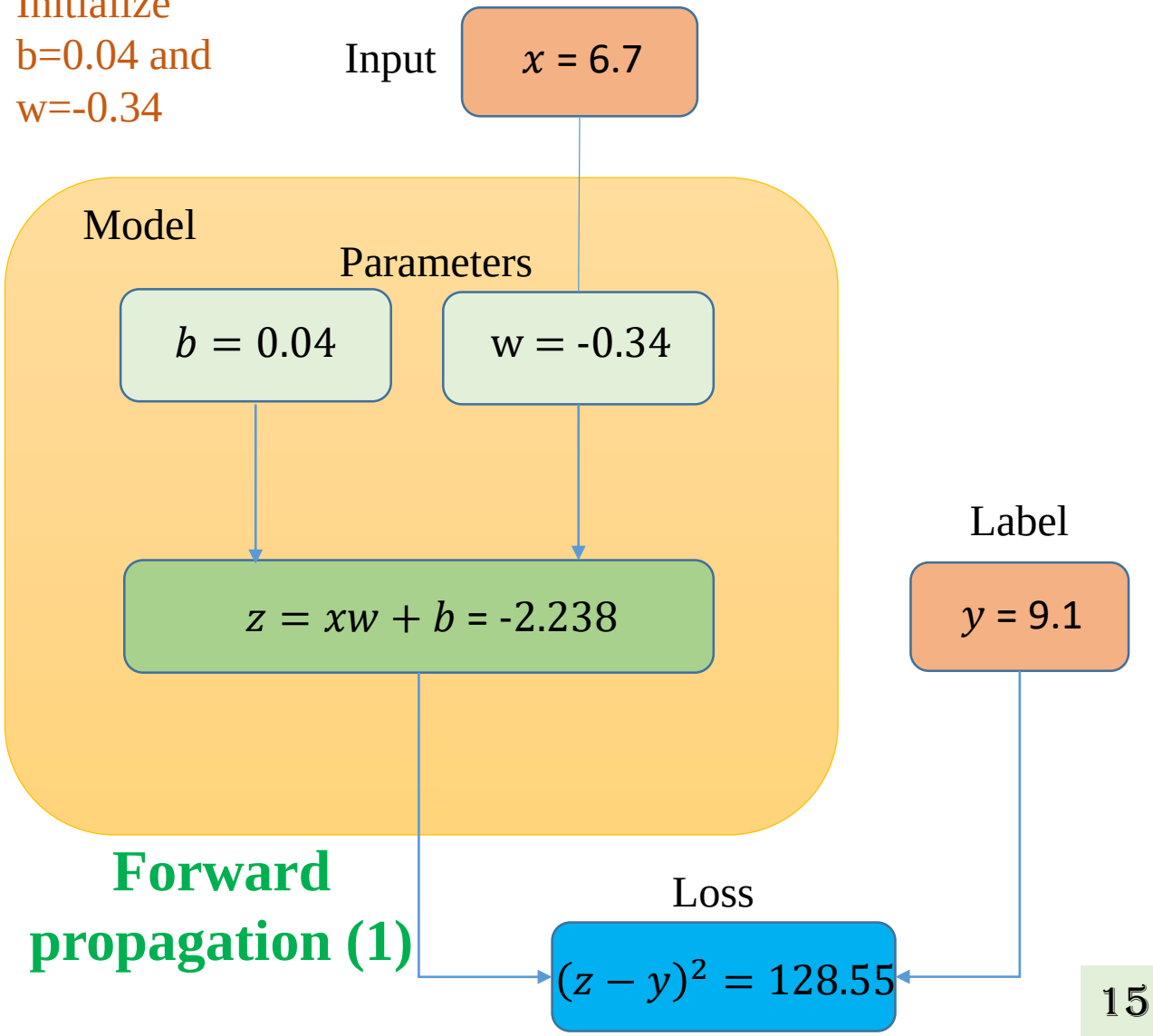
Linear Regression

Given sample data

area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

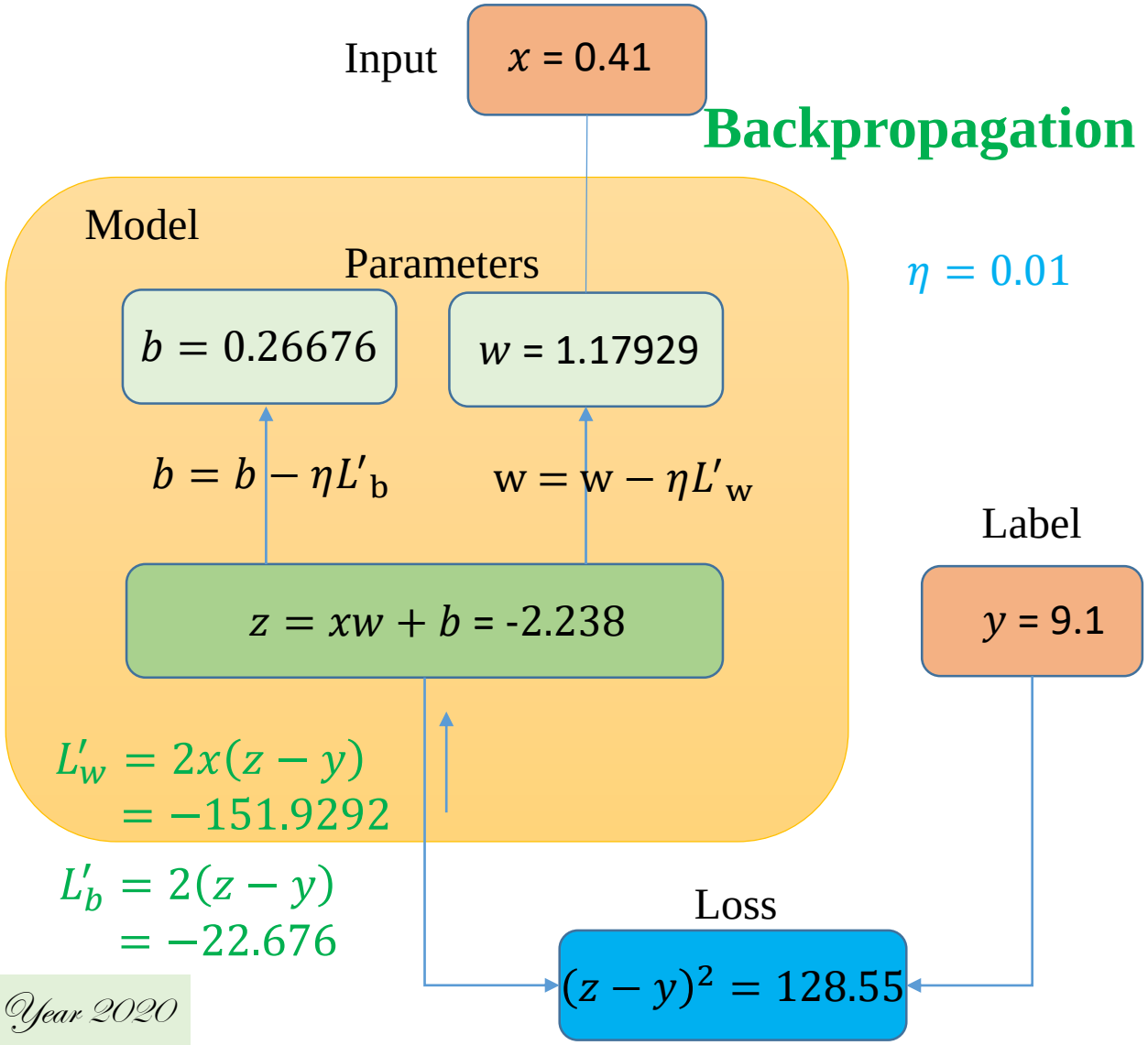


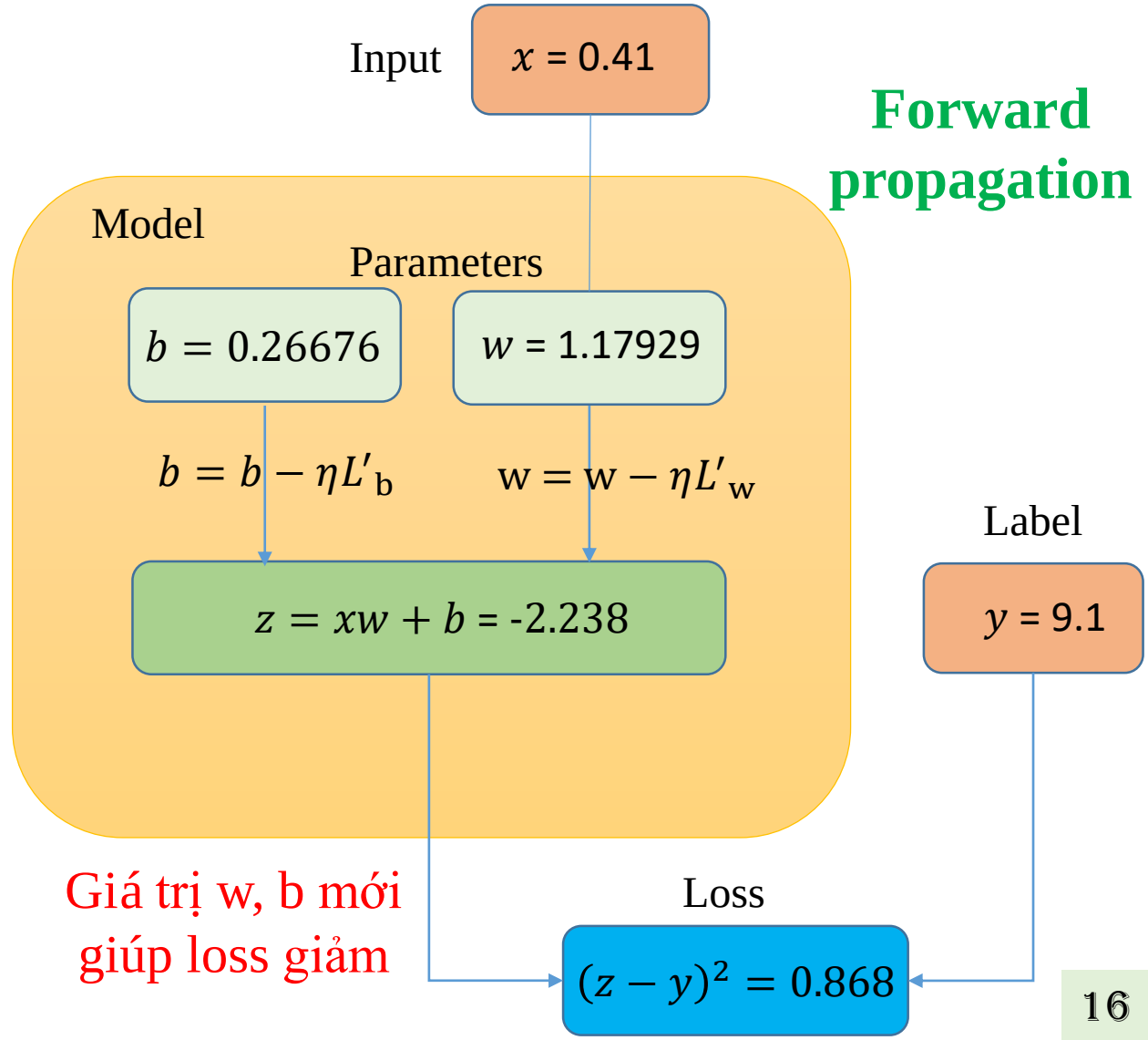
Initialize b=0.04 and w=-0.34



Forward propagation (1)

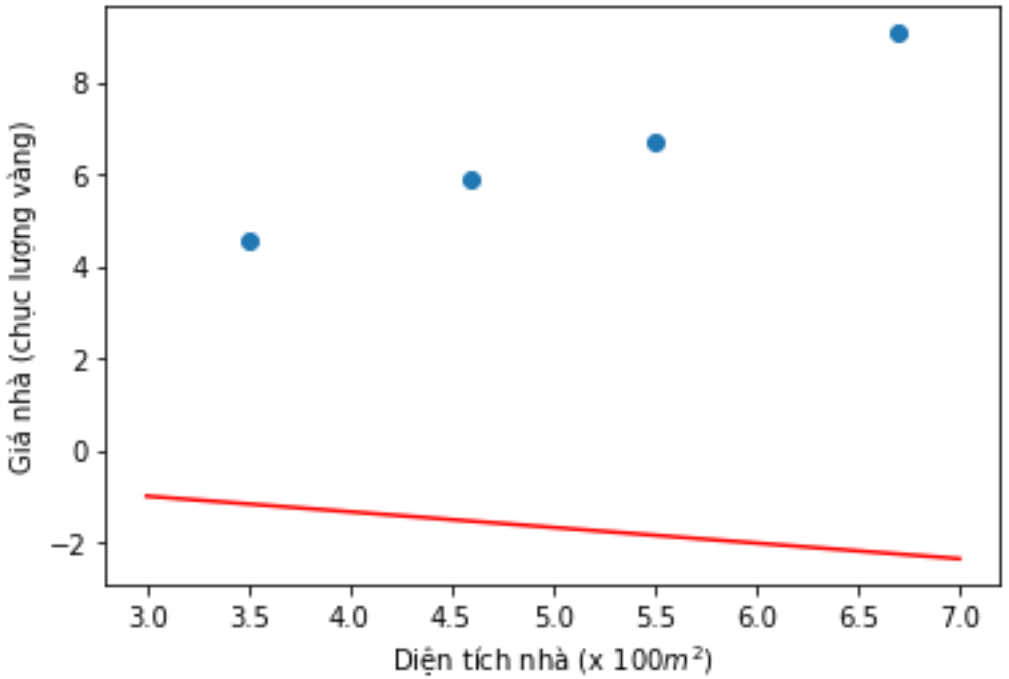
Linear Regression





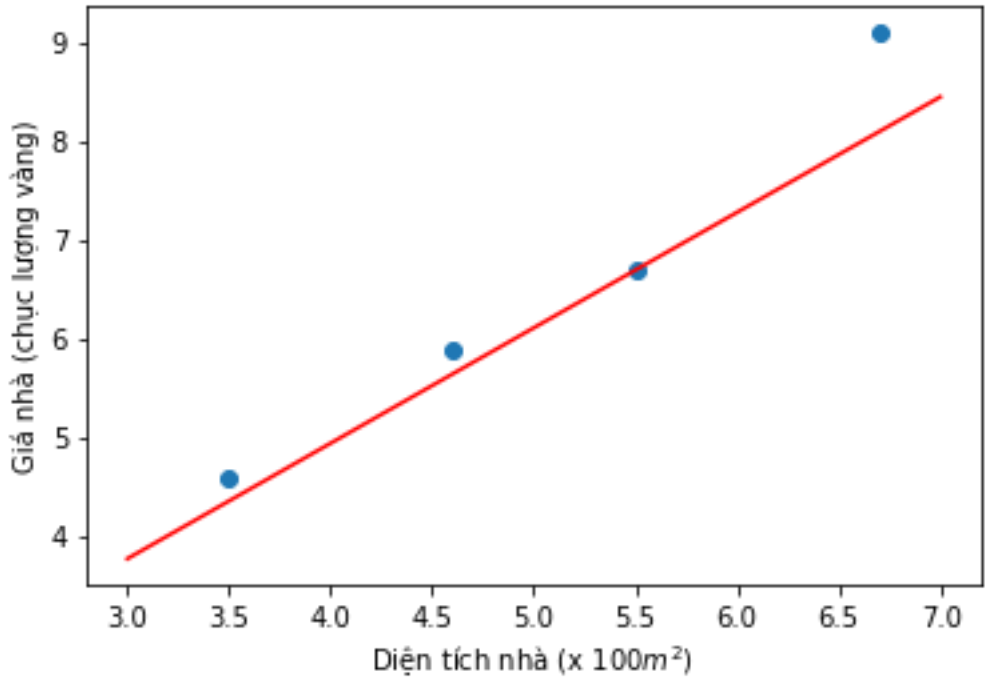
Linear Regression

Model prediction before and after the first update



w = -0.34 b = 0.04 L = 128.55

Before updating



w = 1.179292 b = 0.26676 L = 0.868

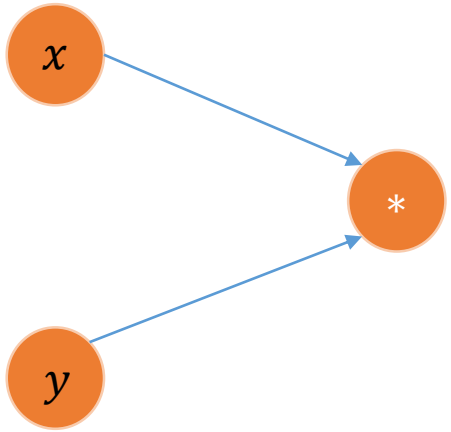
After updating

Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
- Generalized formulae

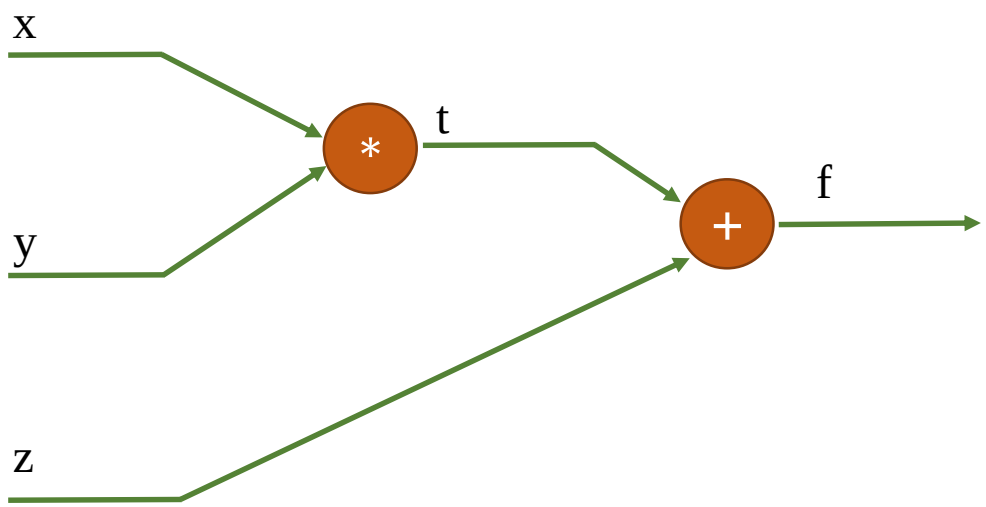
Computational graph

- A directed graph
- Nodes represent variables or operations



- Construct computational graph for $f(x, y, z) = x * y + z$
- Rewrite $f(x, y, z)$ as

$$f(t, z) = t + z \quad \text{where} \quad t = x * y$$

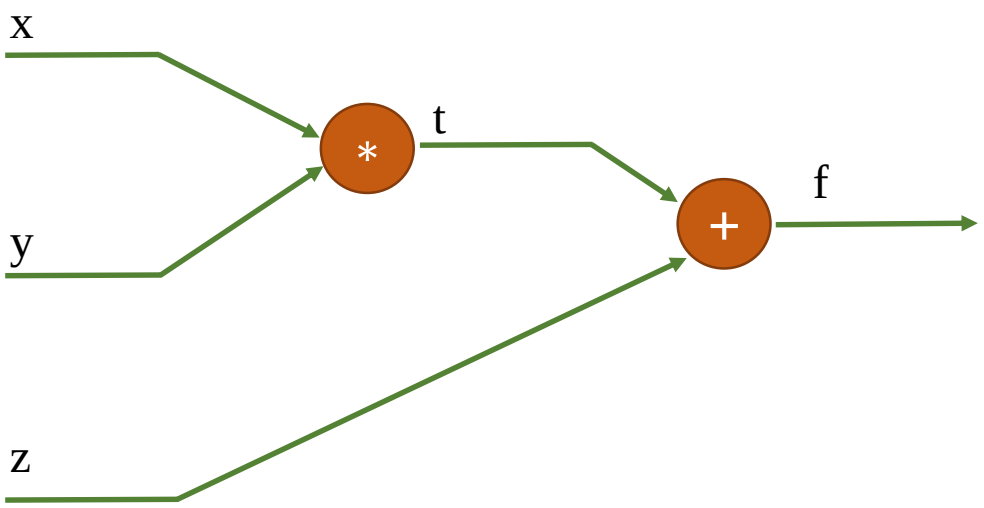


Computational graph

❖ **Construct computational graph for**
$f(x, y, z) = x * y + z$

❖ **Rewrite $f(x, y, z)$ as**

$f(t, z) = t + z$ where $t = x * y$



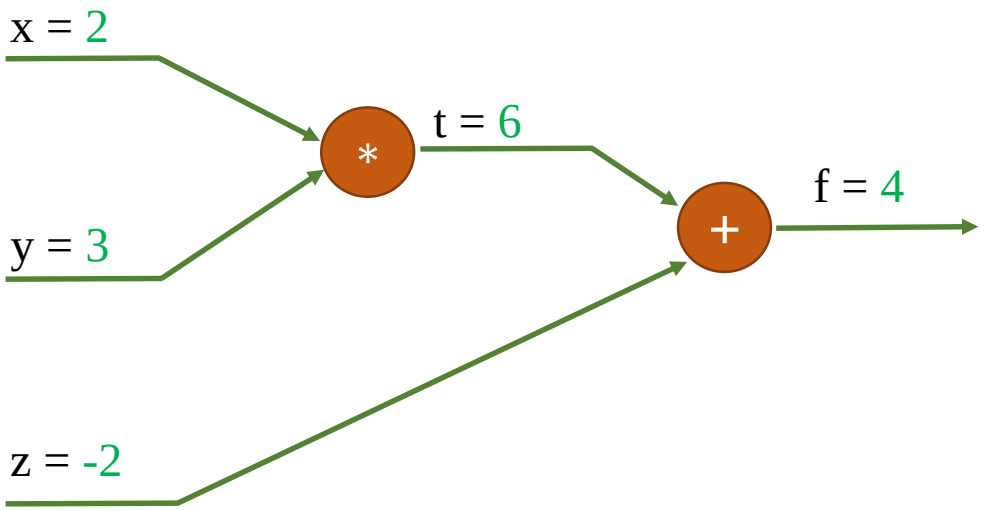
Partial derivative

$$\frac{\partial t}{\partial x} = y \qquad \frac{\partial f}{\partial z} = 1 \qquad \frac{\partial f}{\partial x} = y$$

$$\frac{\partial t}{\partial y} = x \qquad \frac{\partial f}{\partial t} = 1 \qquad \frac{\partial f}{\partial y} = x$$

Computational graph

❖ **Compute $f(x, y, z)$ với $x = 2$, $y = 3$ và $z = -2$.**



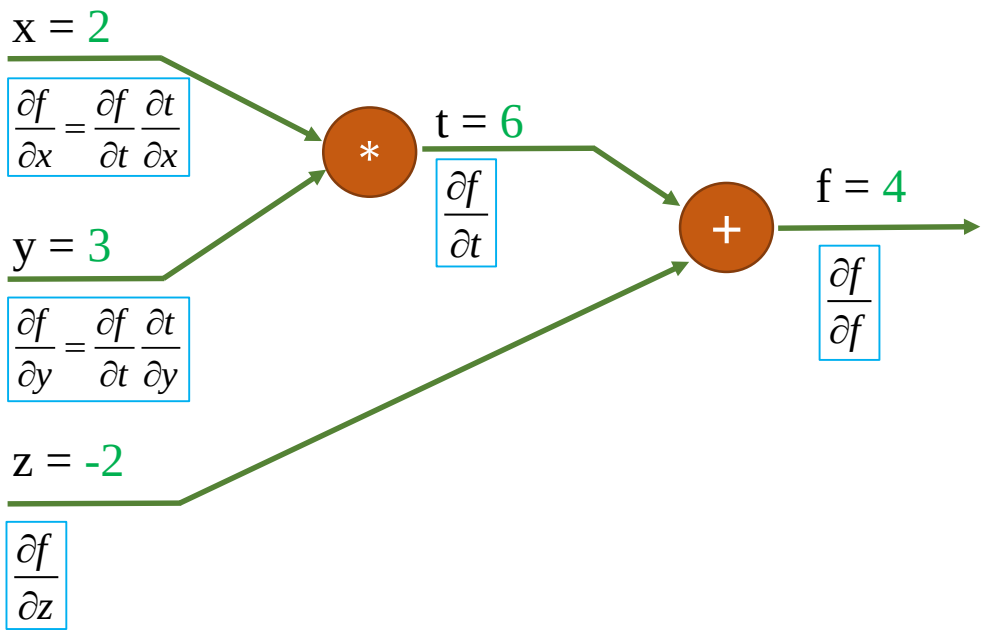
Partial derivative

$$\frac{\partial t}{\partial x} = y \qquad \frac{\partial f}{\partial z} = 1 \qquad \frac{\partial f}{\partial x} = y$$

$$\frac{\partial t}{\partial y} = x \qquad \frac{\partial f}{\partial t} = 1 \qquad \frac{\partial f}{\partial y} = x$$

Computational graph

❖ **Compute $f(x, y, z)$ với $x = 2$, $y = 3$ và $z = -2$.**



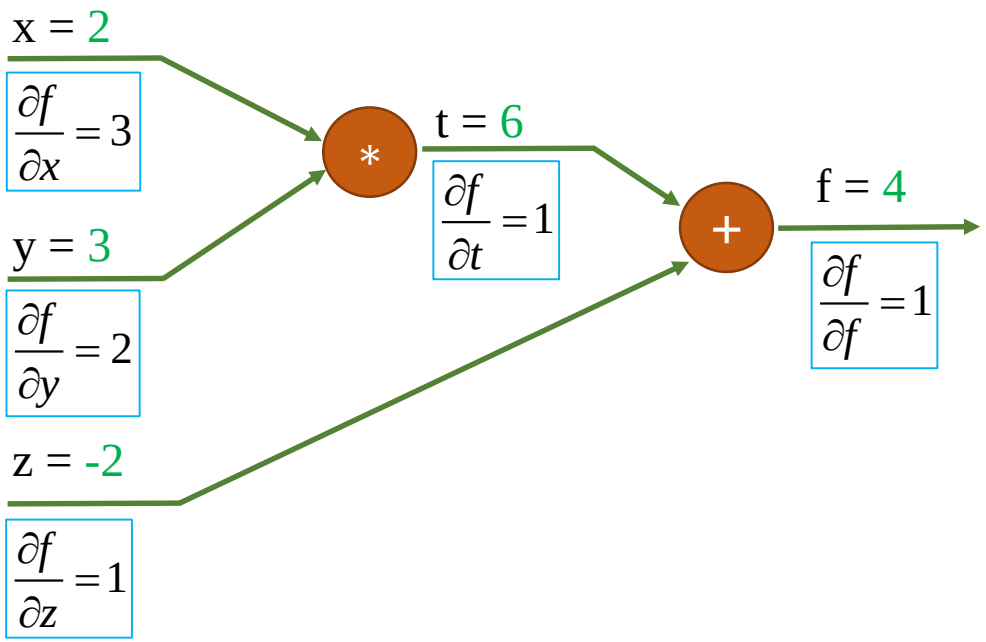
Partial derivative

$$\frac{\partial t}{\partial x} = y \qquad \frac{\partial f}{\partial z} = 1 \qquad \frac{\partial f}{\partial x} = y$$

$$\frac{\partial t}{\partial y} = x \qquad \frac{\partial f}{\partial t} = 1 \qquad \frac{\partial f}{\partial y} = x$$

Computational graph

❖ **Compute $f(x, y, z)$ với $x = 2$, $y = 3$ và $z = -2$.**



Partial derivative

$$\frac{\partial t}{\partial x} = y \qquad \frac{\partial f}{\partial z} = 1 \qquad \frac{\partial f}{\partial x} = y$$

$$\frac{\partial t}{\partial y} = x \qquad \frac{\partial f}{\partial t} = 1 \qquad \frac{\partial f}{\partial y} = x$$

Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
  - 1-sample training
- Generalized formulae

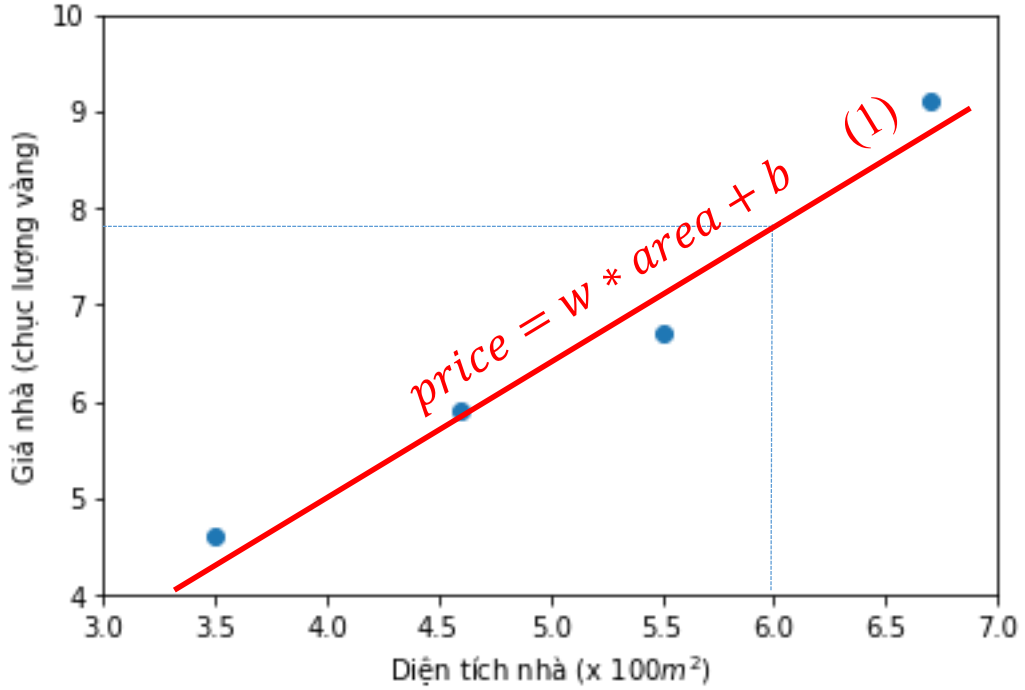
Computational graph

❖ House price predictions

❖ How much for a 600-m^2 house?

area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

Given sample data

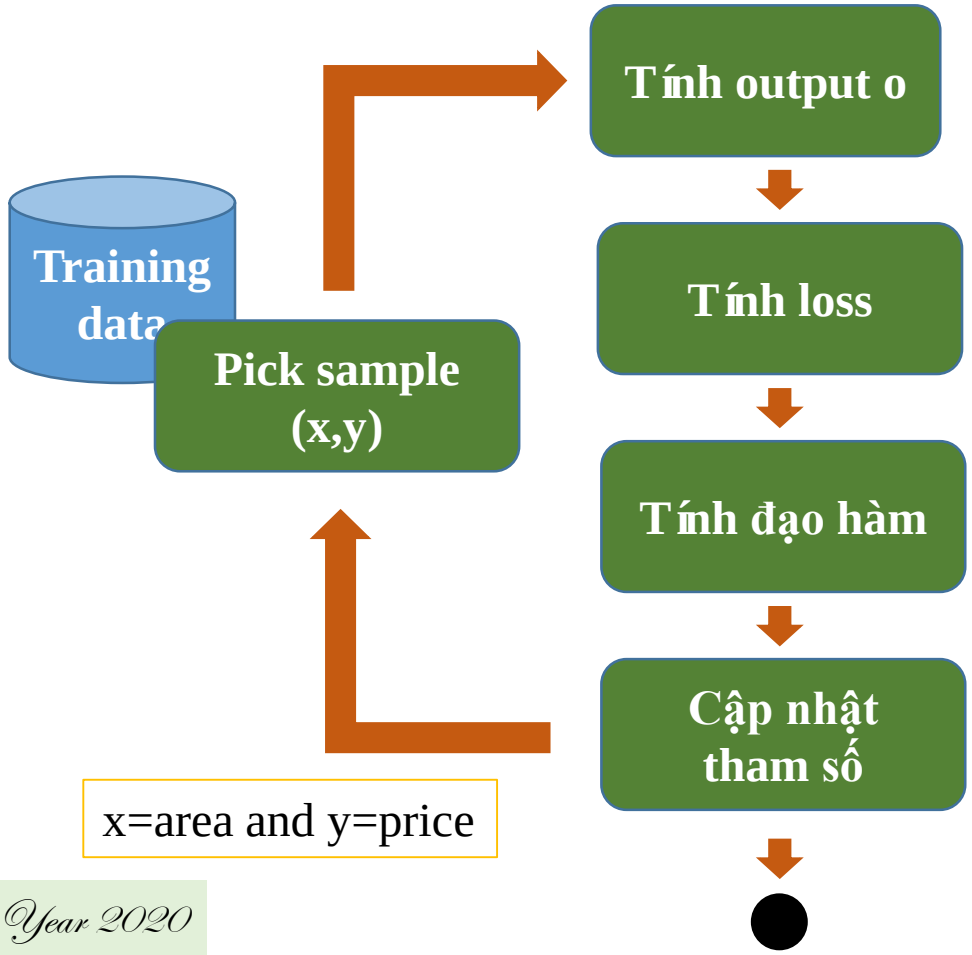


How to compute (1) using computational graph

Computational graph

❖ House price prediction

❖ One-sample training



1) Pick a sample (x, y) from training data

2) Tính output o

$$o = wx + b$$

3) Tính loss

$$L = (o - y)^2$$

4) Tính đạo hàm

$$L'_w = 2x(o - y)$$

$$L'_b = 2(o - y)$$

5) Cập nhật tham số

$$w = w - \eta L'_w$$

$$b = b - \eta L'_b$$

Learning rate η

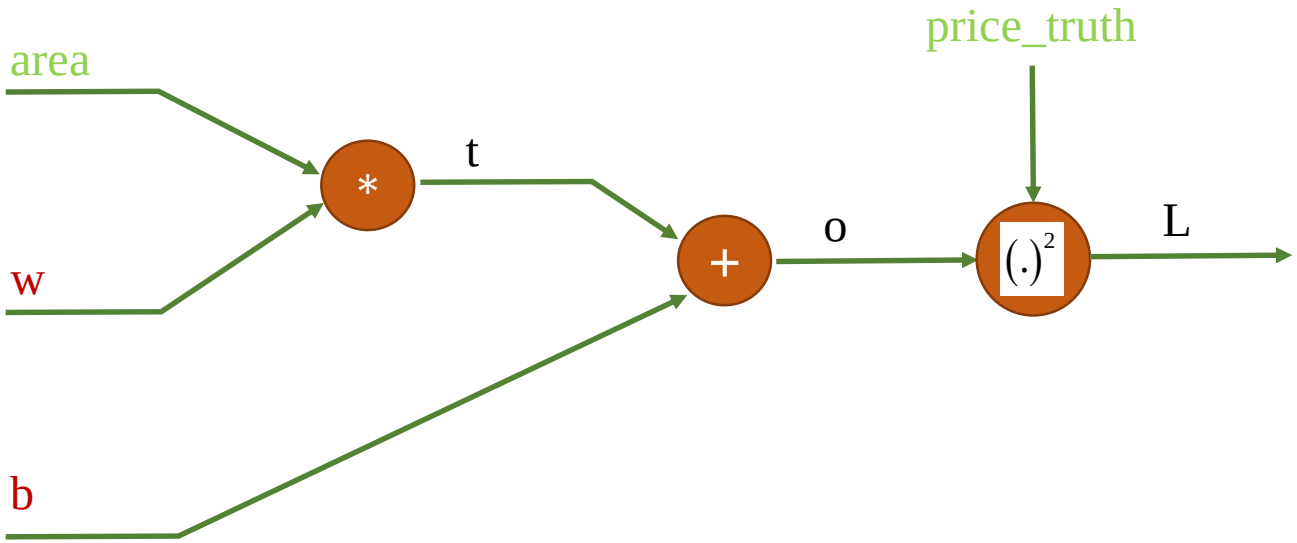
Computational graph

❖ House price prediction

❖ One-sample training

$$price = o = w * area + b$$

$$t = w * area$$

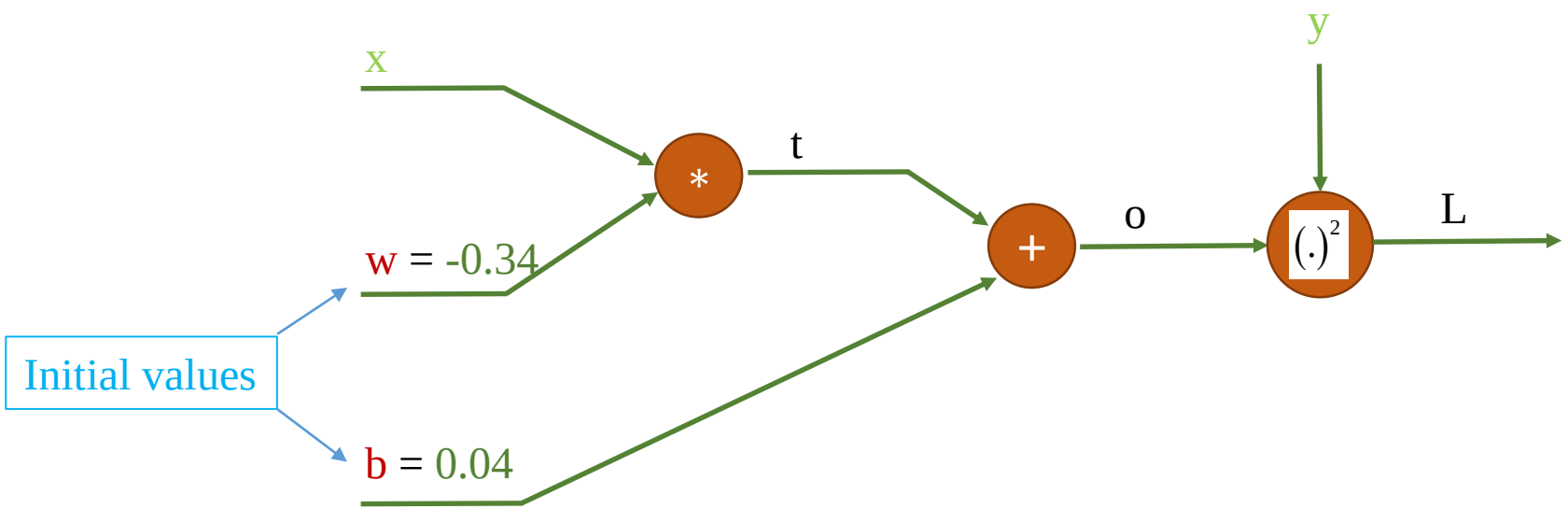


Computational graph

❖ House price prediction

❖ One-sample training

$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$

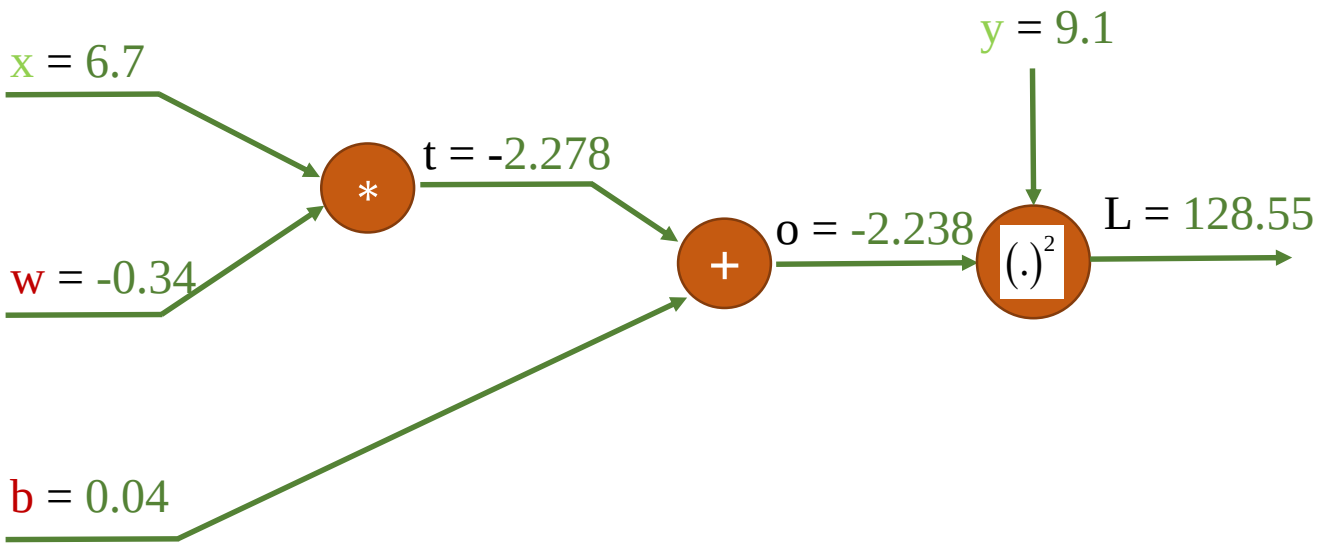


Computational graph

❖ House price prediction

❖ One-sample training

$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$

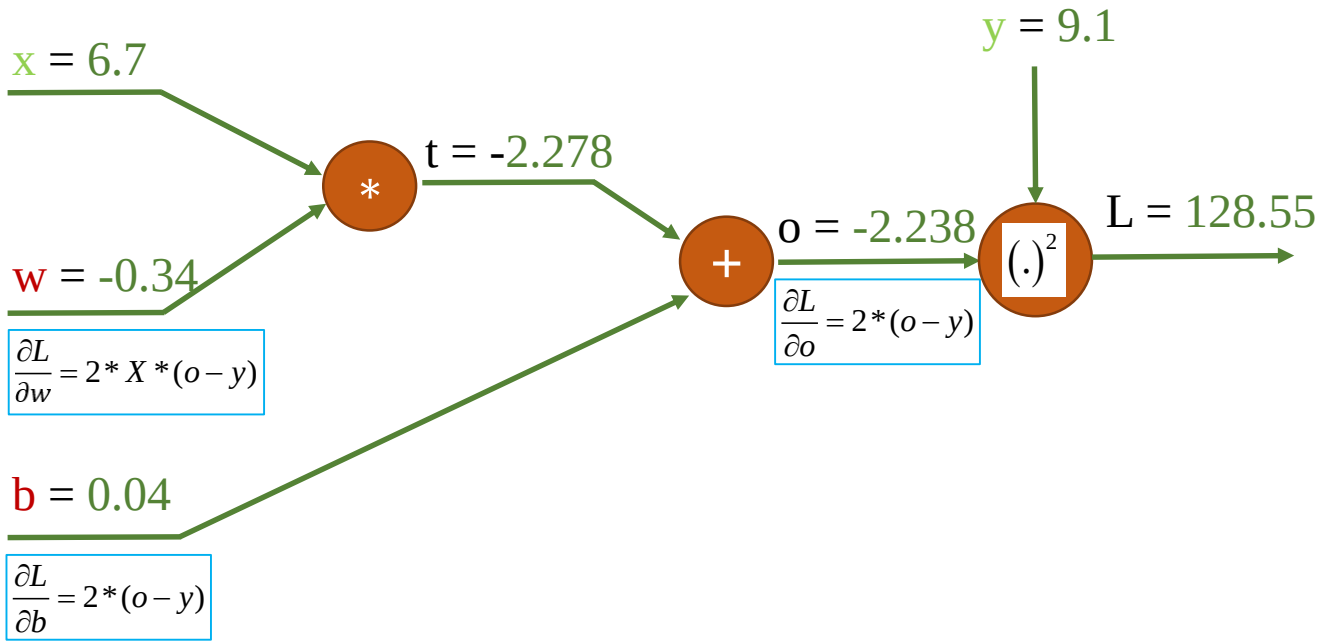


Computational graph

❖ House price prediction

❖ One-sample training

$t = wx$ $o = t + b$ $L = (o - y)^2$

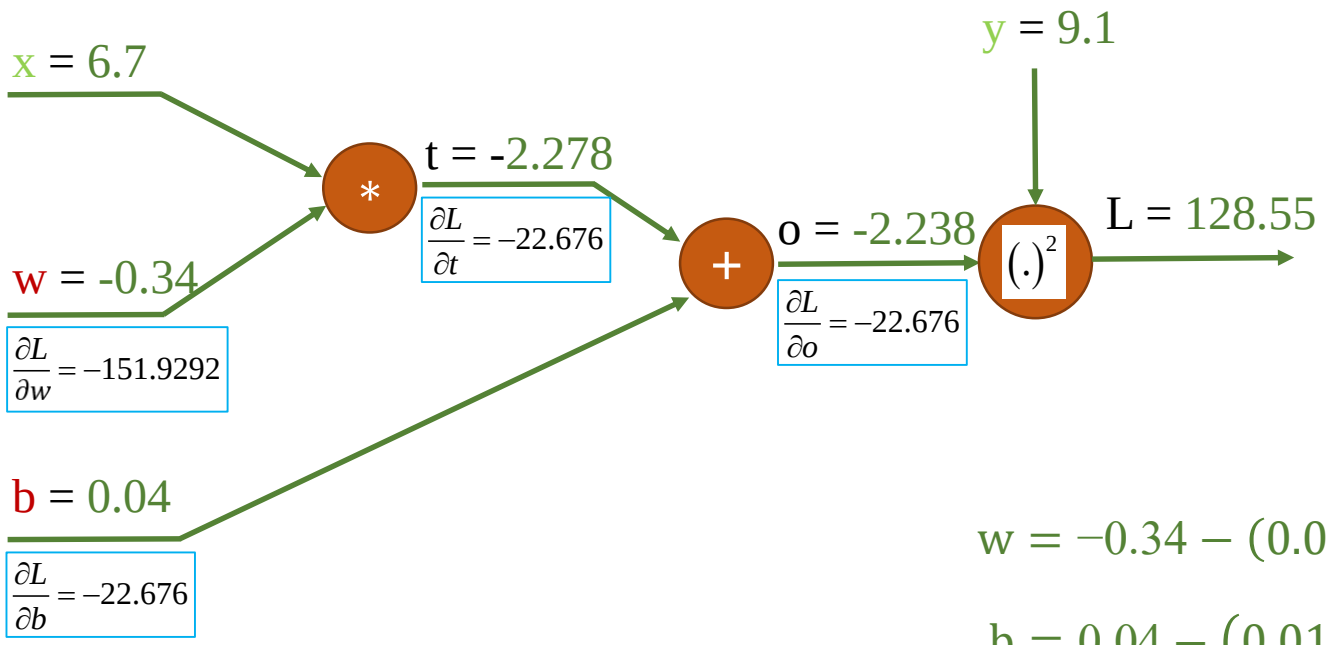


Computational graph

❖ House price prediction

❖ One-sample training

$t = wx$ $\quad o = t + b$ $\quad L = (o - y)^2$



Cách cập nhật a và b

$$w = w - \eta * \frac{\partial L}{\partial w}$$

$$b = b - \eta * \frac{\partial L}{\partial b}$$

Learning rate $\eta = 0.01$

$$w = -0.34 - (0.01 * (-151.9)) = 1.179$$

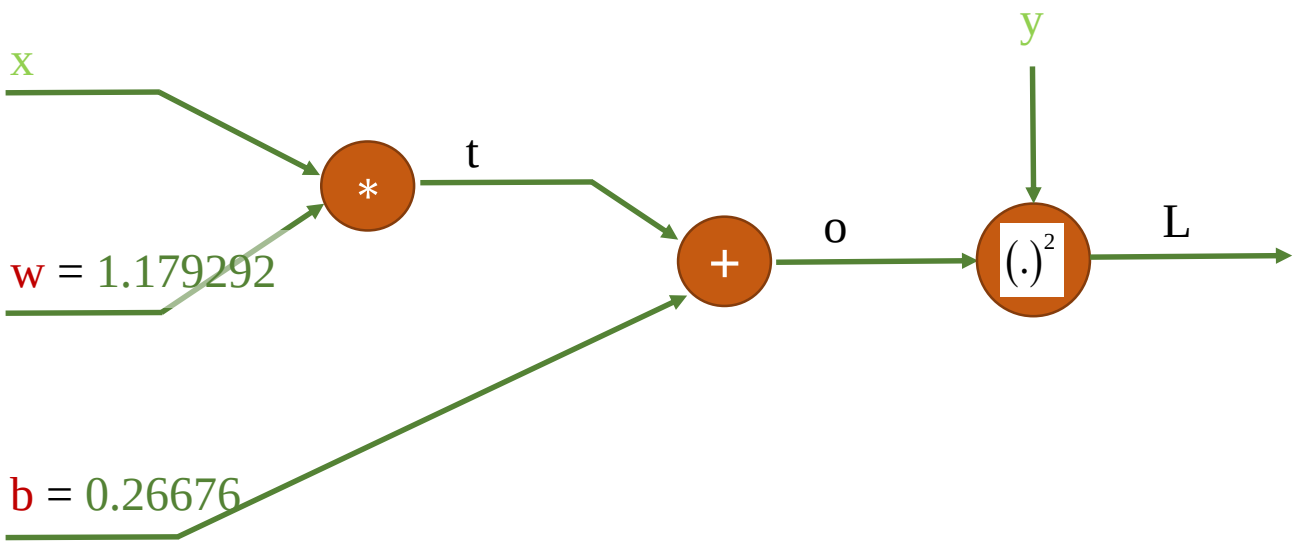
$$b = 0.04 - \left(0.01 * (-22.67)\right) = 0.266$$

Computational graph

❖ House price prediction

❖ One-sample training

$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$

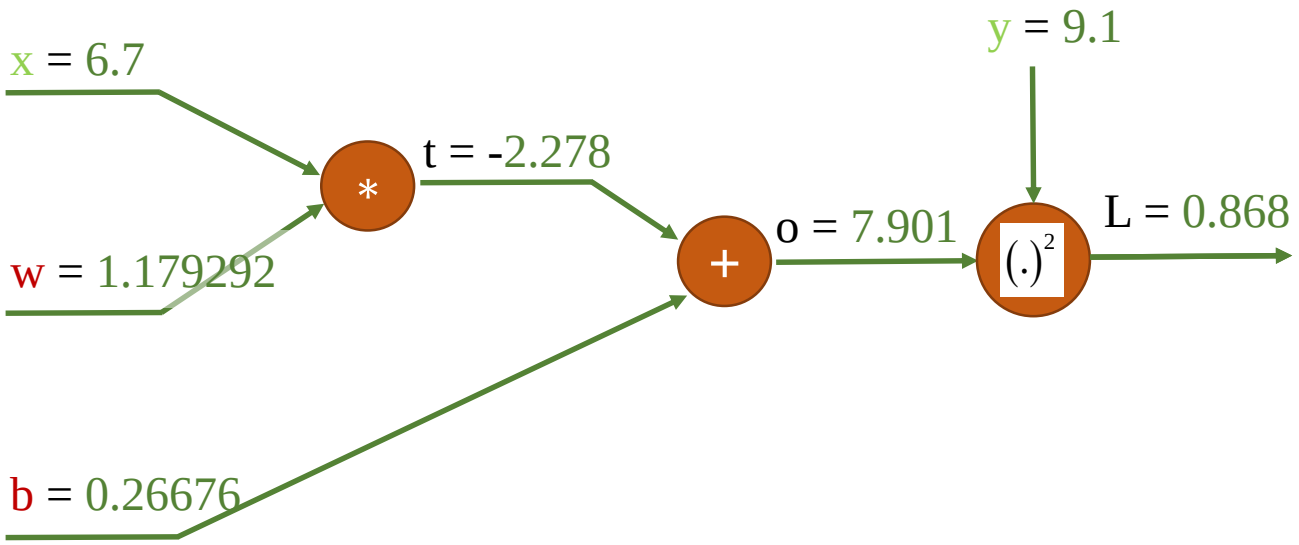


Computational graph

❖ House price prediction

❖ One-sample training

$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$



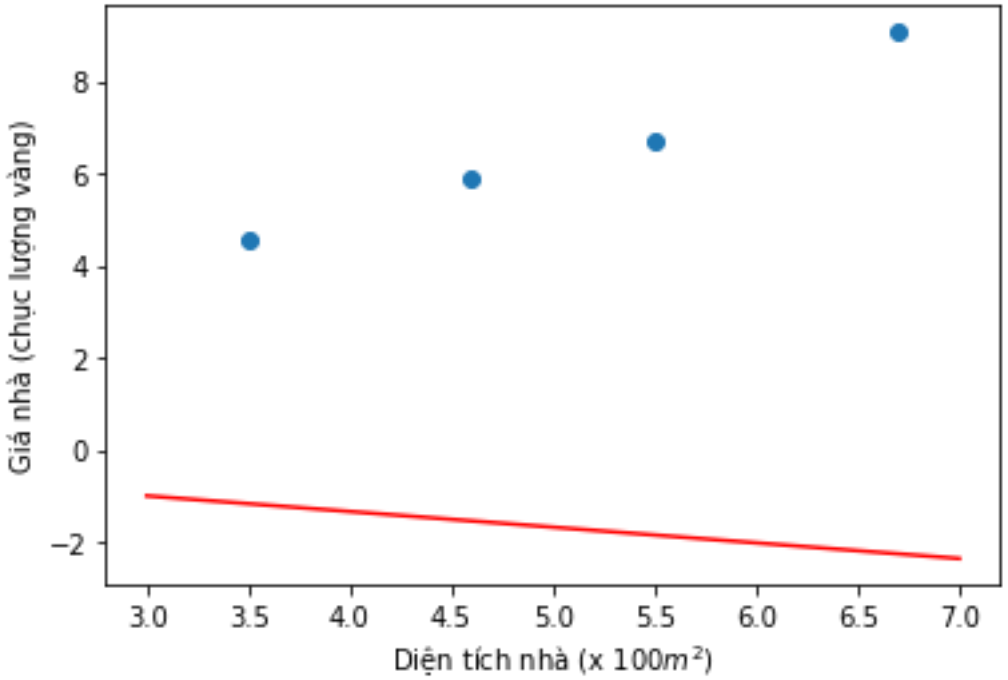
previous L = 128.55

Updated a and b values help to reduce the L value

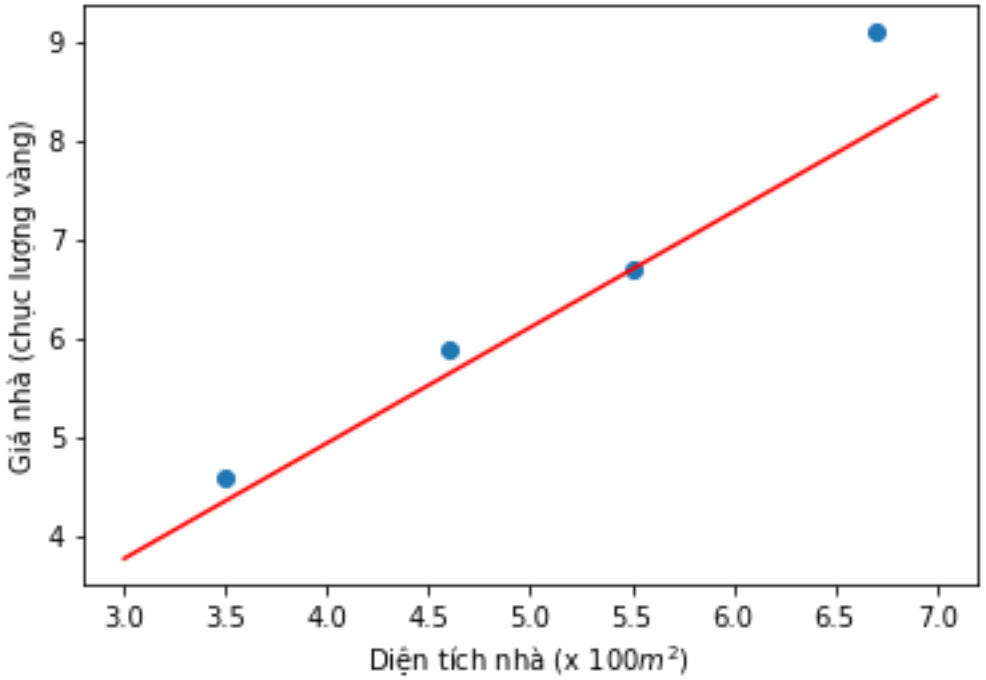
Computational graph

❖ House price prediction

❖ One-sample training



w = -0.34 b = 0.04 L = 128.55

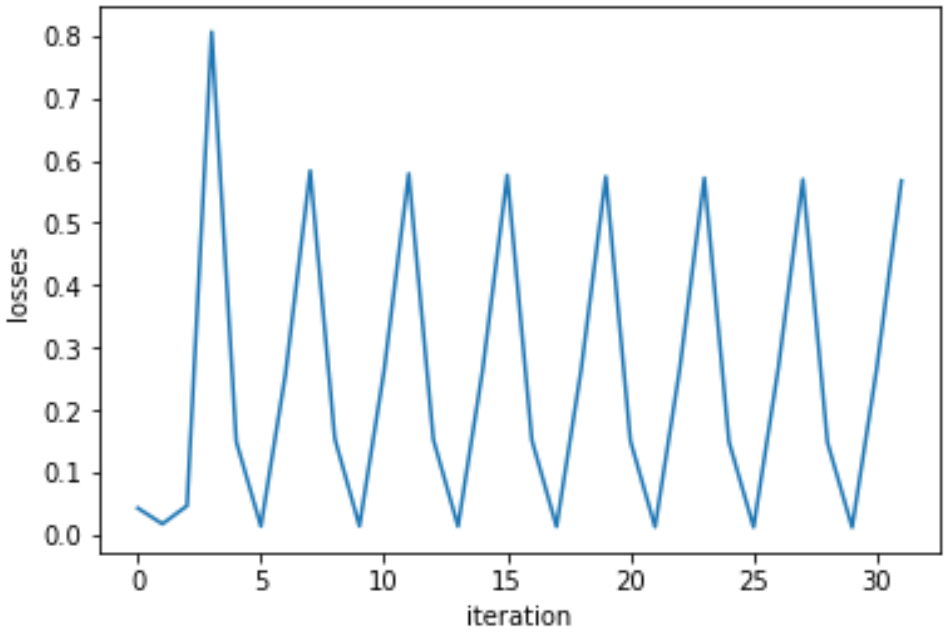


w = 1.179292 b = 0.26676 L = 0.868

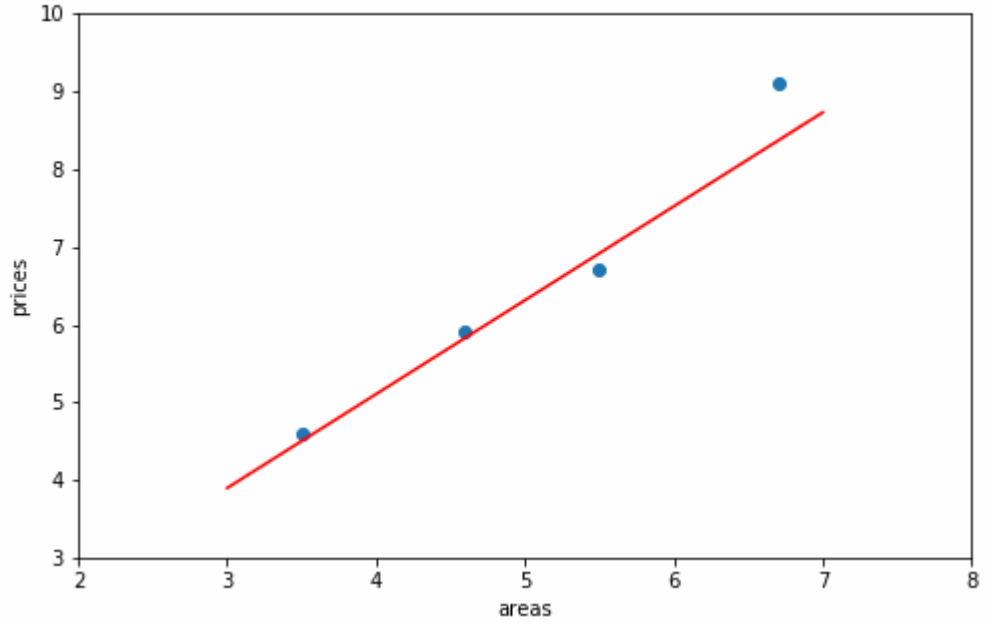
Computational graph

❖ House price prediction

❖ One-sample training



Losses for 30 iterations



Model updating for different iterations

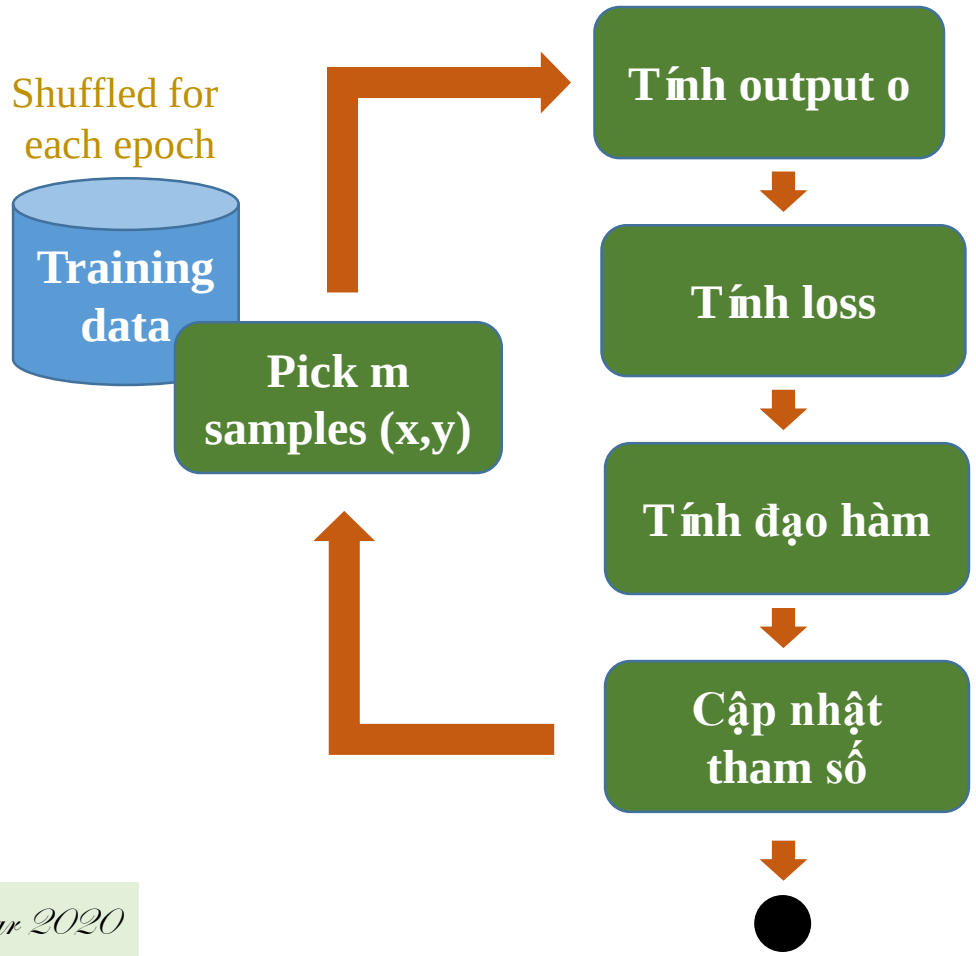
Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
  - 1-sample training
  - m-sample training
- Generalized formulae

Computational graph

❖ House price prediction

❖ m-sample training (1<m<N)



1) Pick m samples $(x^{(i)}, y^{(i)})$ from training data

2) Tính output o_i

$$o^{(i)} = wx^{(i)} + b \qquad \text{for } 0 \le i < m$$

3) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \le i < m$$

4) Tính đạo hàm

$$L_w'^{(i)} = 2x(o^{(i)} - y^{(i)})$$
$$L_b'^{(i)} = 2(o^{(i)} - y^{(i)}) \quad \text{for } 0 \le i < m$$

5) Cập nhật tham số

$$w = w - \eta \frac{\sum_i L_w'^{(i)}}{m}$$

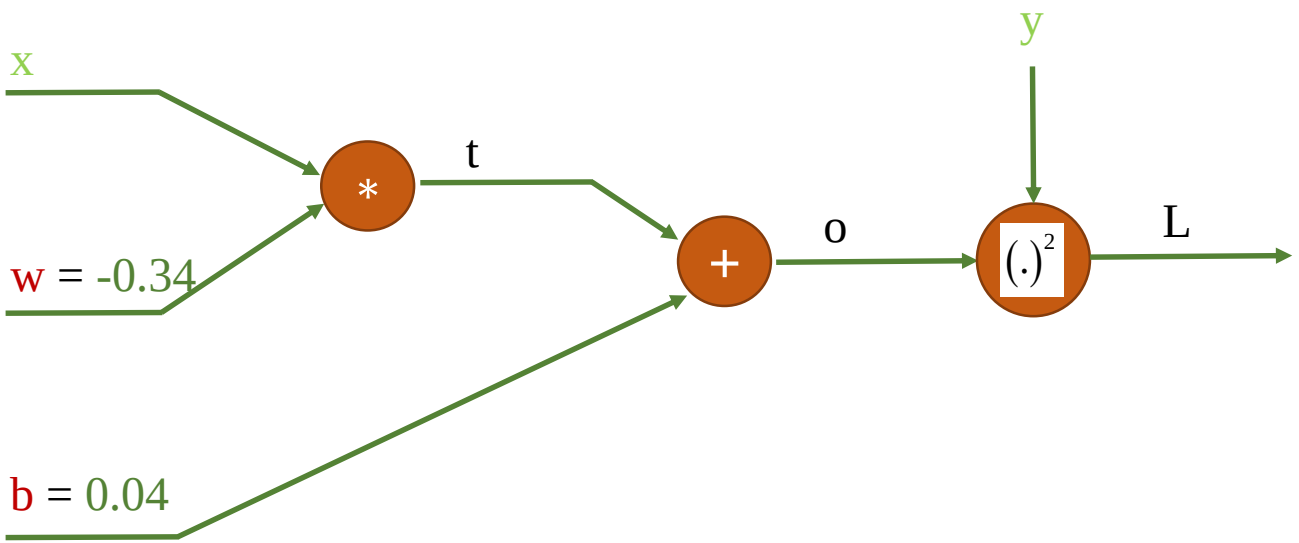
$$b = b - \eta \frac{\sum_i L_b'^{(i)}}{m}$$

Learning rate η

Computational graph

- House price prediction
  - m-sample training (1<m<N)

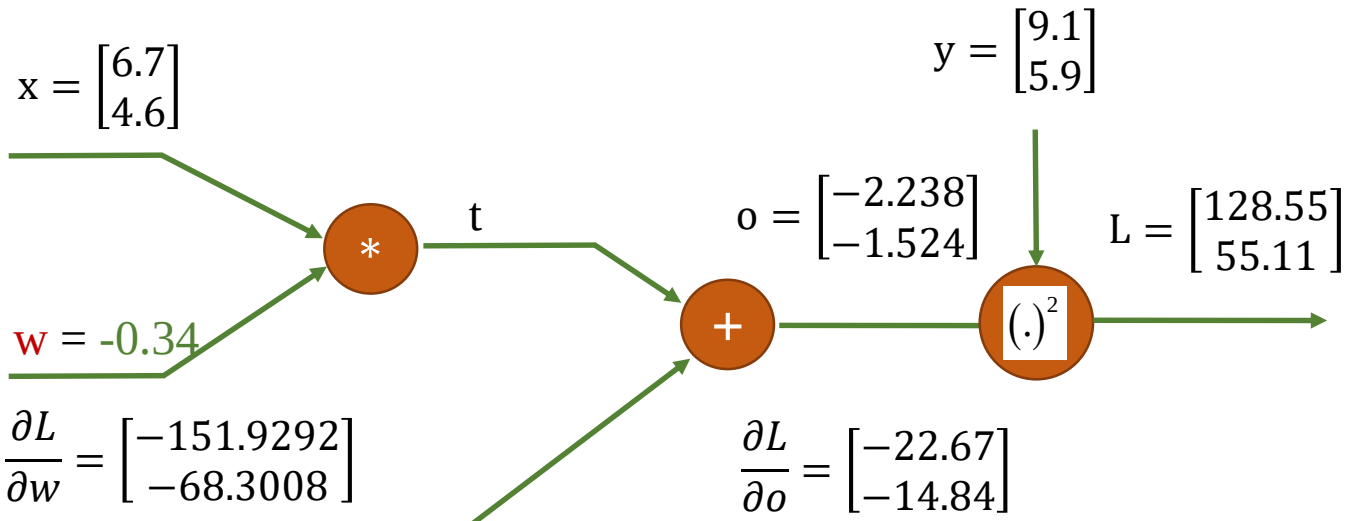
$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$



Computational graph

- **House price prediction**
  - **m-sample training (1<m<N)**

m = 2



$$\frac{sum(\frac{\partial L}{\partial w})}{m} = -110.115$$

$$\frac{sum(\frac{\partial L}{\partial b})}{m} = -18.762$$

$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$

Computational graph

- **House price prediction**
  - **m-sample training (1<m<N)**

Cách cập nhật a và b

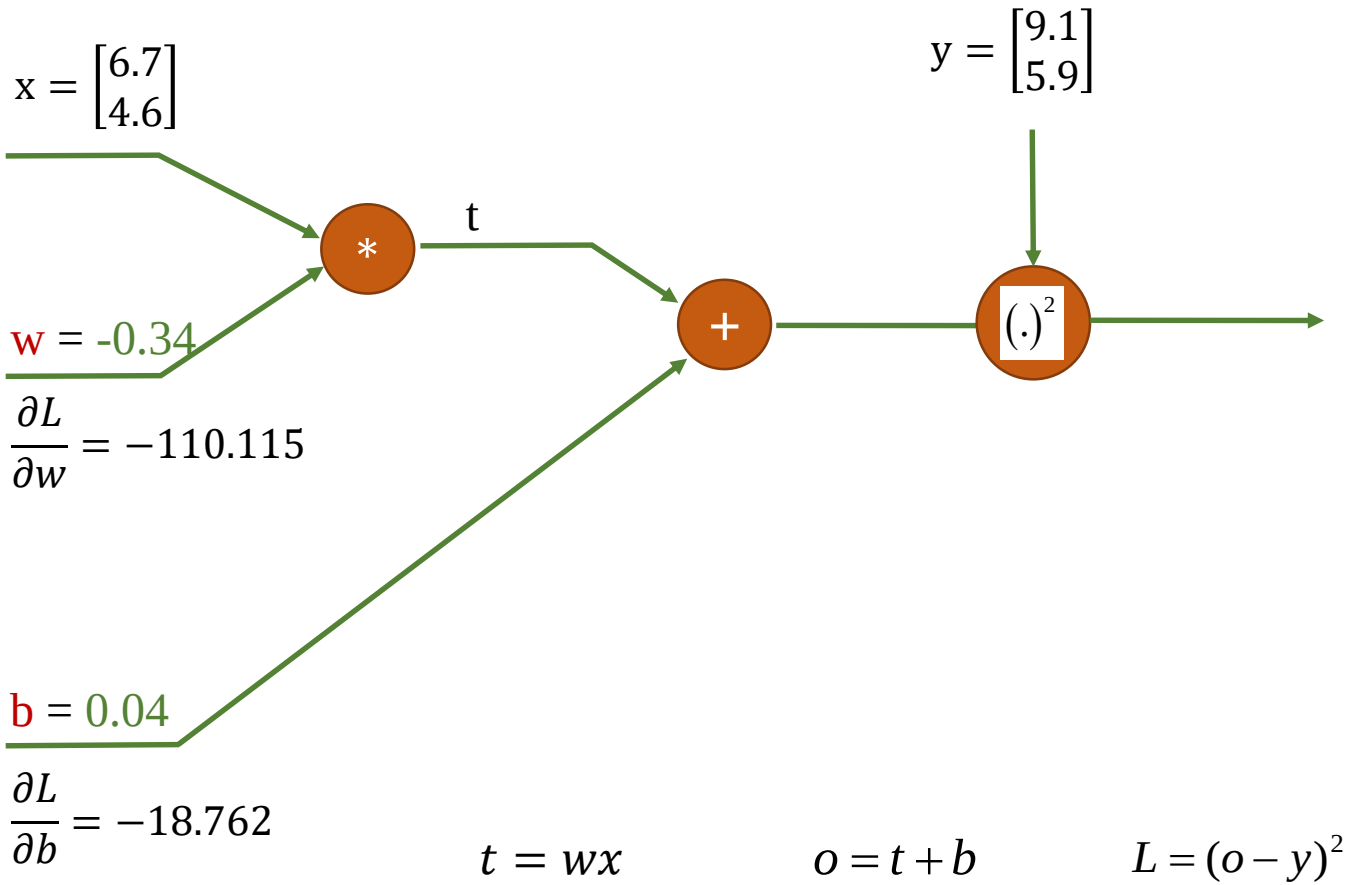
$$w = w - \eta * \frac{\partial L}{\partial w}$$

$$b = b - \eta * \frac{\partial L}{\partial b}$$

Learning rate $\eta = 0.01$

$$w = -0.34 - (0.01 * (-110.115)) = 0.761$$

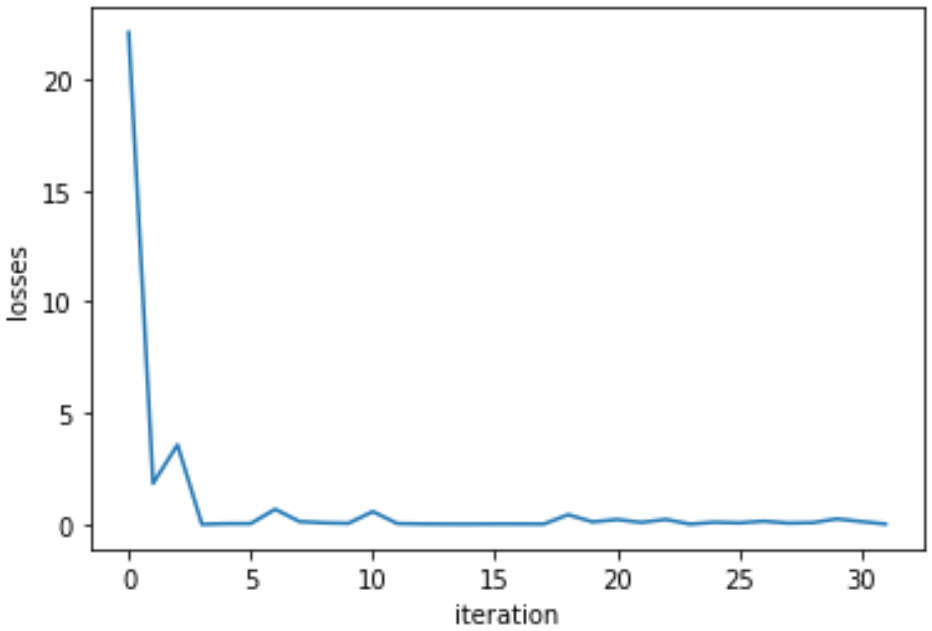
$$b = 0.04 - (0.01 * (-18.762)) = 0.227$$



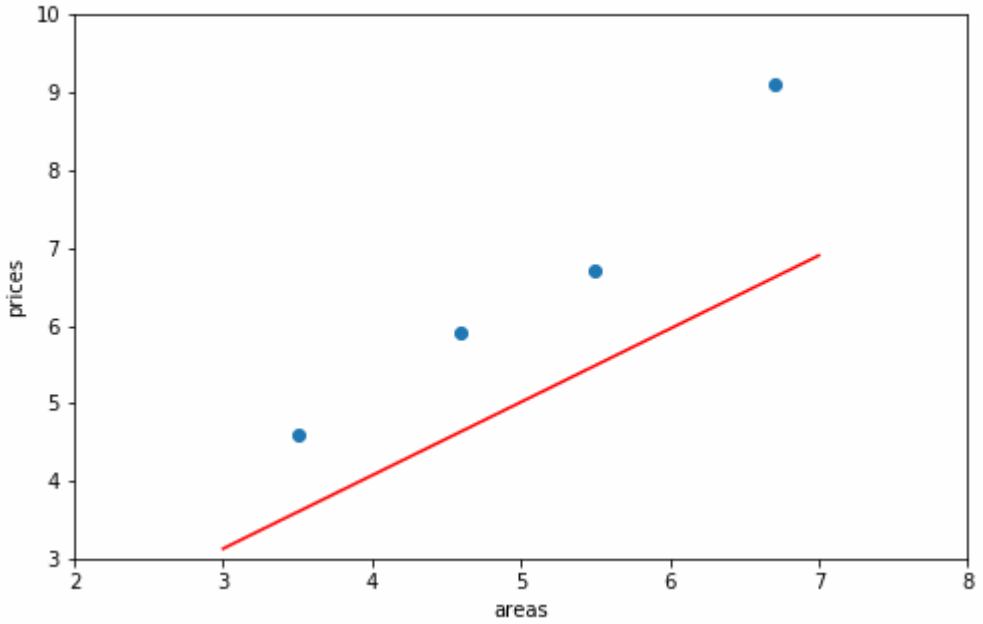
$t = wx$ $o = t + b$ $L = (o - y)^2$

Computational graph

- House price prediction
  - m-sample training (1<m<N)



Losses for 30 iterations



Model updating for different iterations

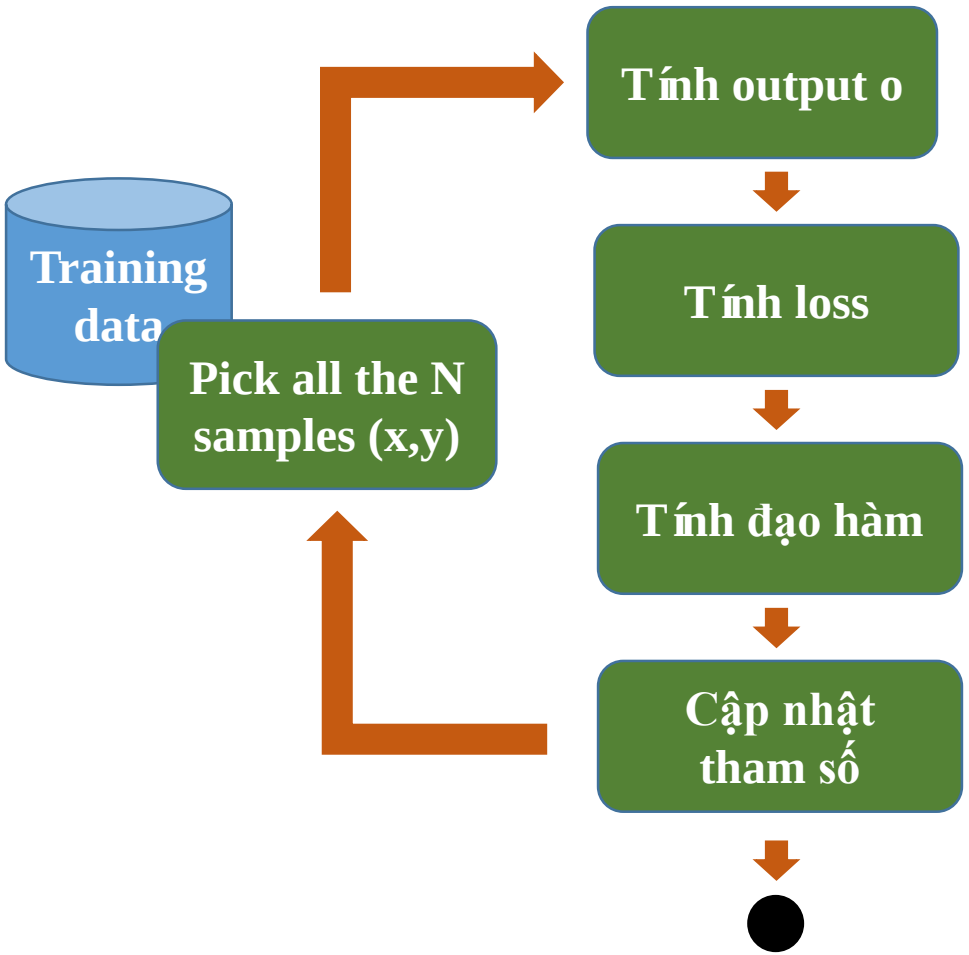
Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
  - 1-sample training
  - m-sample training
  - N-sample training
- Generalized formulae

Computational graph

❖ House price prediction

❖ N-sample training



1) Pick all the N samples $(x^{(i)}, y^{(i)})$ from training data

2) Tính output o_i

$$o^{(i)} = wx^{(i)} + b \qquad \text{for } 0 \le i < N$$

3) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \le i < N$$

4) Tính đạo hàm

$$L_w'^{(i)} = 2x(o^{(i)} - y^{(i)})$$
$$L_b'^{(i)} = 2(o^{(i)} - y^{(i)}) \quad \text{for } 0 \le i < N$$

5) Cập nhật tham số

$$w = w - \eta \frac{\sum_i L_w'^{(i)}}{N}$$

$$b = b - \eta \frac{\sum_i L_b'^{(i)}}{N}$$

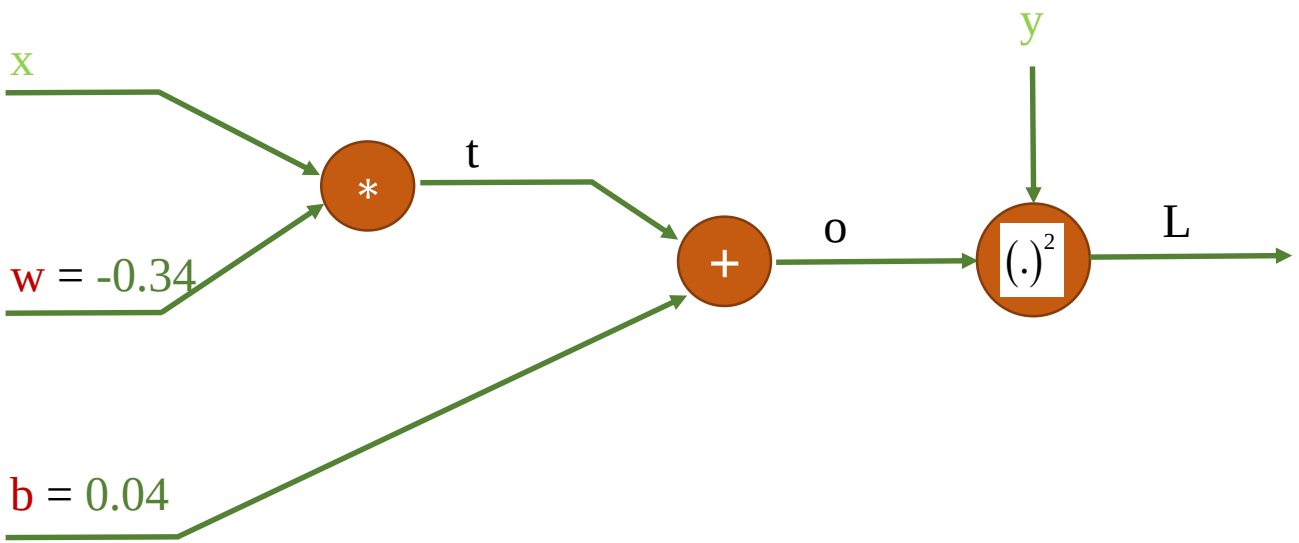
Learning rate η

Computational graph

❖ House price prediction

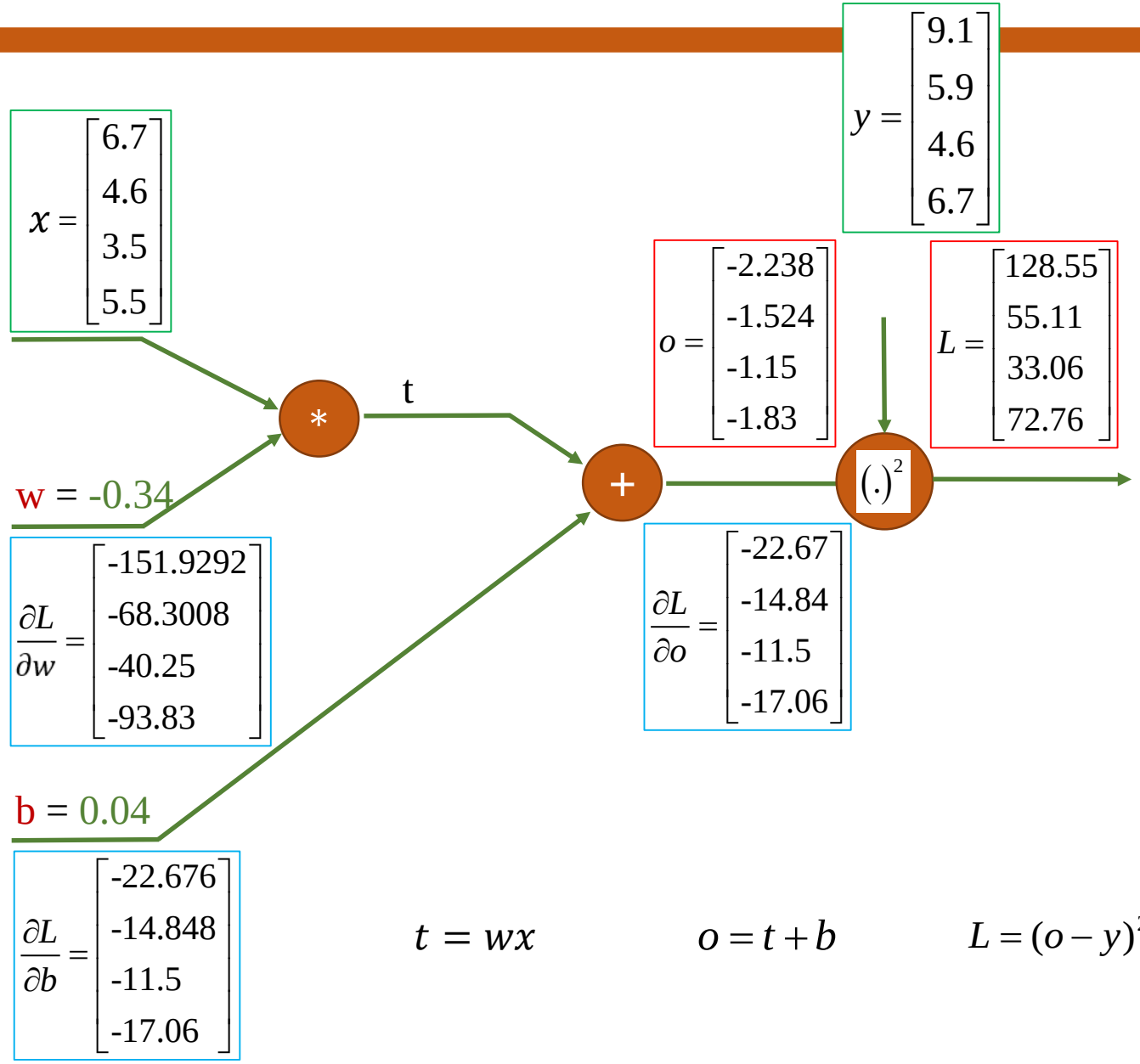
❖ N-sample training

$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$



Computational graph

- **House price prediction**
  - **N-sample training**



$$\frac{sum(\frac{\partial L}{\partial w})}{4} = -88.5775$$

$$\frac{sum(\frac{\partial L}{\partial b})}{4} = -16.521$$

$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$

Computational graph

❖ House price prediction

❖ N-sample training

Cách cập nhật a và b

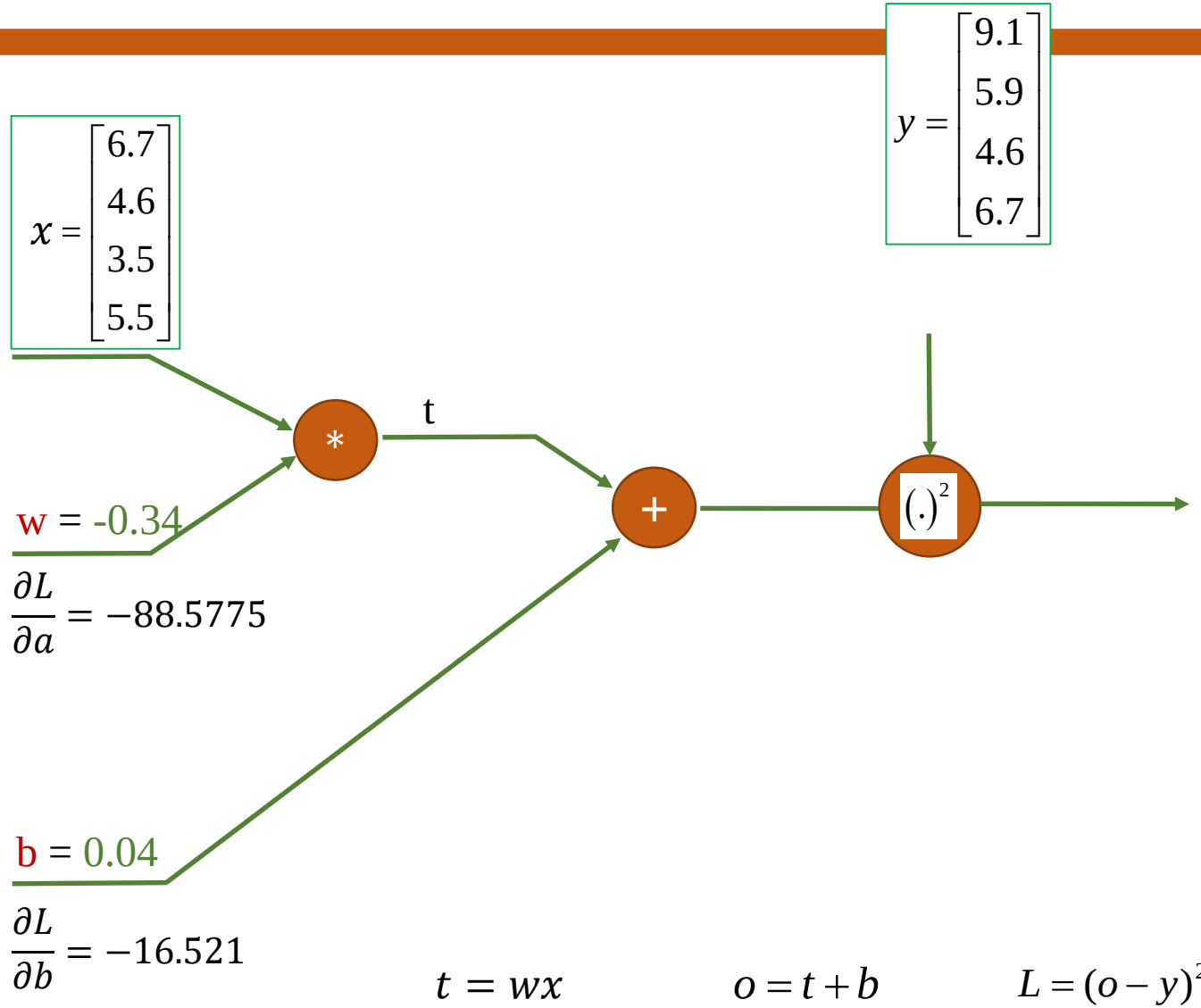
$$w = w - \eta * \frac{\partial L}{\partial w}$$

$$b = b - \eta * \frac{\partial L}{\partial b}$$

Learning rate $\eta = 0.01$

$$w = -0.34 - (0.01 * (-88.5775)) = 0.54$$

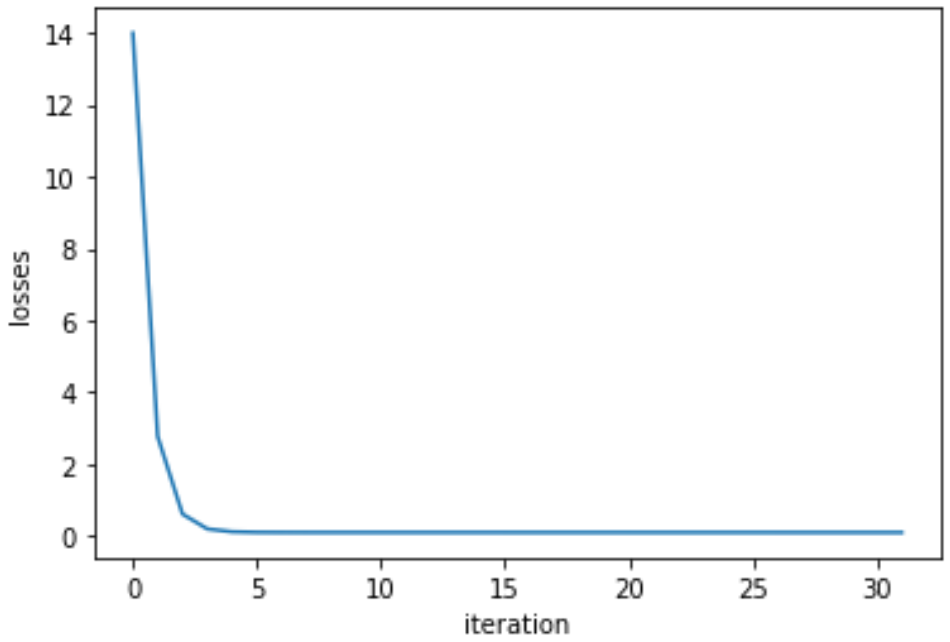
$$b = 0.04 - \left(0.01 * (-16.521)\right) = 0.205$$



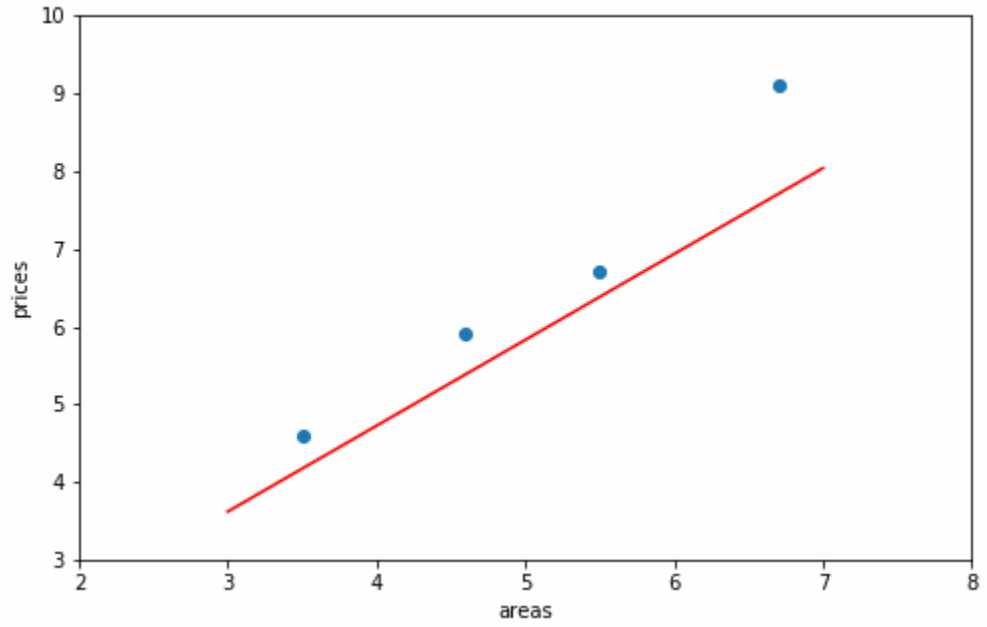
Computational graph

❖ House price prediction

❖ N-sample training



Losses for 30 iterations



Model updating for different iterations

Outline

- Machine Learning
- Derivative/Gradient
- Linear Regression
- Computational Graph
- Generalized formulae

Linear Regression

❖ Generalized formula

House price data

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

Model

price = w $*$ $area + b$

y = $wx + b$

Model (vectorization)

y = $\boldsymbol{\theta}^T \boldsymbol{x}$ where $\boldsymbol{\theta}^T = [b \quad w]^T$

$\boldsymbol{x} = [x_0 \quad area]^T$

$x_0 = 1$

Advertising data

Features			Label
TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9

Model

Sale = $w_1 * TV + w_2 * Radio + w_3 * Newspaper + b$

y = $w_1 x_1 + w_2 x_2 + w_3 x_3 + b$

Model (vectorization)

y = $\boldsymbol{\theta}^T \boldsymbol{x}$ where $\boldsymbol{\theta}^T = [b \quad w_1 \quad w_2 \quad w_3]^T$

$\boldsymbol{x} = [x_0 \quad TV \quad Radio \quad Newspaper]^T$

$x_0 = 1$

Linear Regression

❖ Generalized formula

Boston House Price Data

Features													Label
crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9

Model

$$\text{medv} = w_1 * x_1 + \cdots + w_{13} * x_{13} + b$$

Model (vectorization)

$$y = \boldsymbol{\theta}^T \boldsymbol{x} \quad \text{where} \quad \boldsymbol{\theta}^T = [b \quad w_1 \quad \ldots \quad w_{13}]^T$$
$$\boldsymbol{x} = [x_0 \quad x_1 \quad \ldots \quad x_{13}]^T$$
$$x_0 = 1$$

Linear Regression (1-sample)

1) Pick a sample (x, y) from training data

2) Tính output o

$$o = wx + b$$

3) Tính loss

$$L = (o - y)^2$$

4) Tính đạo hàm

$$L'_w = 2x(o - y)$$
$$L'_b = 2(o - y)$$

5) Cập nhật tham số

$$w = w - \eta L'_w$$
$$b = b - \eta L'_b$$

η is learning rate

Friendly version

1) Pick a sample (x, y) from training data

2) Tính output o

$$o = \boldsymbol{\theta}^T \boldsymbol{x}$$

3) Tính loss

$$L = (o - y)^2$$

4) Tính đạo hàm

$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(o - y)$$

5) Cập nhật tham số

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate

Generalized formula

Linear Regression (m-samples)

1) Pick m samples $(x^{(i)}, y^{(i)})$ from training data

1.1) Tính output $o^{(i)}$

$$o^{(i)} = wx^{(i)} + b \quad \text{for } 0 \leq i < m$$

1.2) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < m$$

1.3) Tính đạo hàm

$$L'^{(i)}_w = 2x(o^{(i)} - y^{(i)})$$
$$L'^{(i)}_b = 2(o^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < m$$

2) Cập nhật tham số

$$w = w - \eta \frac{\sum_i L'^{(i)}_w}{m}$$

$$b = b - \eta \frac{\sum_i L'^{(i)}_b}{m}$$

η is learning rate

Friendly version

1) Pick m samples $(x^{(i)}, y^{(i)})$ from training data

1.1) Tính output $o^{(i)}$

$$o^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} \quad \text{for } 0 \leq i < m$$

1.2) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < m$$

1.3) Tính đạo hàm

$$L'^{(i)}_{\boldsymbol{\theta}} = 2\boldsymbol{x}(o^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < m$$

2) Cập nhật tham số

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L'^{(i)}_{\boldsymbol{\theta}}}{m}$$

η is learning rate

Generalized formula

Linear Regression (N-samples)

1) Pick all the N samples from training data

2) Tính output $o^{(i)}$

$$o^{(i)} = wx^{(i)} + b \quad \text{for } 0 \leq i < N$$

3) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < N$$

4) Tính đạo hàm

$$L'^{(i)}_w = 2x(o^{(i)} - y^{(i)})$$
$$L'^{(i)}_b = 2(o^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < N$$

5) Cập nhật tham số

$$w = w - \eta \frac{\sum_i L'^{(i)}_w}{N}$$

$$b = b - \eta \frac{\sum_i L'^{(i)}_b}{N}$$

η is learning rate

Friendly version

1) Pick all the N samples from training data

2) Tính output $o^{(i)}$

$$o^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} \quad \text{for } 0 \leq i < N$$

3) Tính loss

$$L^{(i)} = (o^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < N$$

4) Tính đạo hàm

$$L'^{(i)}_{\boldsymbol{\theta}} = 2\boldsymbol{x}(o^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < N$$

5) Cập nhật tham số

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L'^{(i)}_{\boldsymbol{\theta}}}{N}$$

η is learning rate

Generalized formula

Linear Regression

❖ House price prediction

Demo

```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] ::
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> for epoch in range(n_epochs):
...     sum_of_losses = 0
...     gradients = np.zeros((2,1))
...
...     for index in range(4):
...         xi = X_b[index:index+1]
...         yi = y[index:index+1]
```

Linear Regression

❖ Advertising-based sale prediction

Demo

```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] ::
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> for epoch in range(n_epochs):
...     sum_of_losses = 0
...     gradients = np.zeros((2,1))
...
...     for index in range(4):
...         xi = X_b[index:index+1]
...         yi = y[index:index+1]
```

Linear Regression

❖ House price prediction

❖Boston data

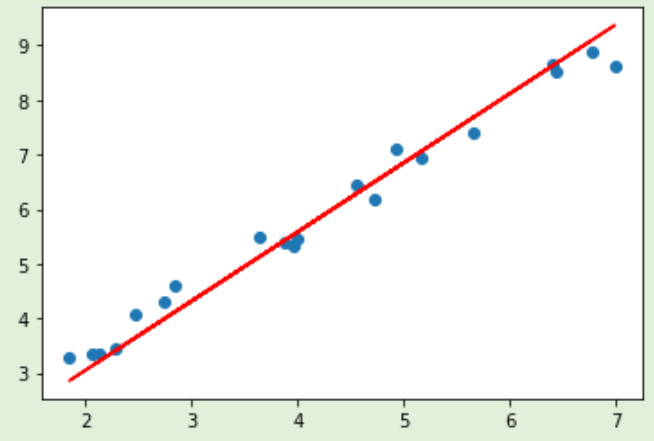
Demo

```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] ::
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> for epoch in range(n_epochs):
...     sum_of_losses = 0
...     gradients = np.zeros((2,1))
...
...     for index in range(4):
...         xi = X_b[index:index+1]
...         yi = y[index:index+1]
```

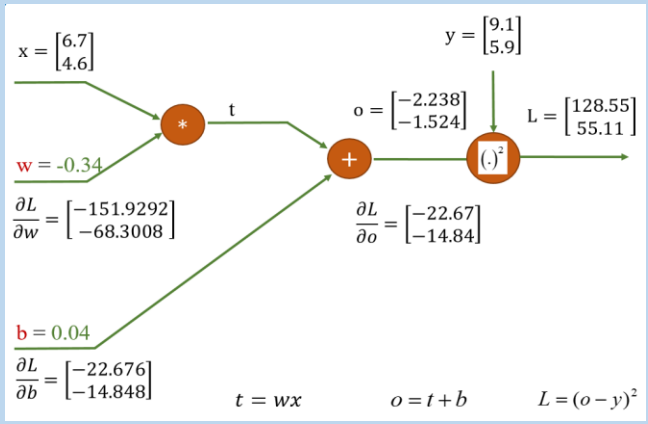
AI VIETNAM
Free AI Course 2020

Summary

Linear regression



Computational graph



Formulae cheat sheet

1) Pick a sample (x, y) from training data

2) Tính output o

$$o = \boldsymbol{\theta}^T \boldsymbol{x}$$

3) Tính loss

$$L = (o - y)^2$$

4) Tính đạo hàm

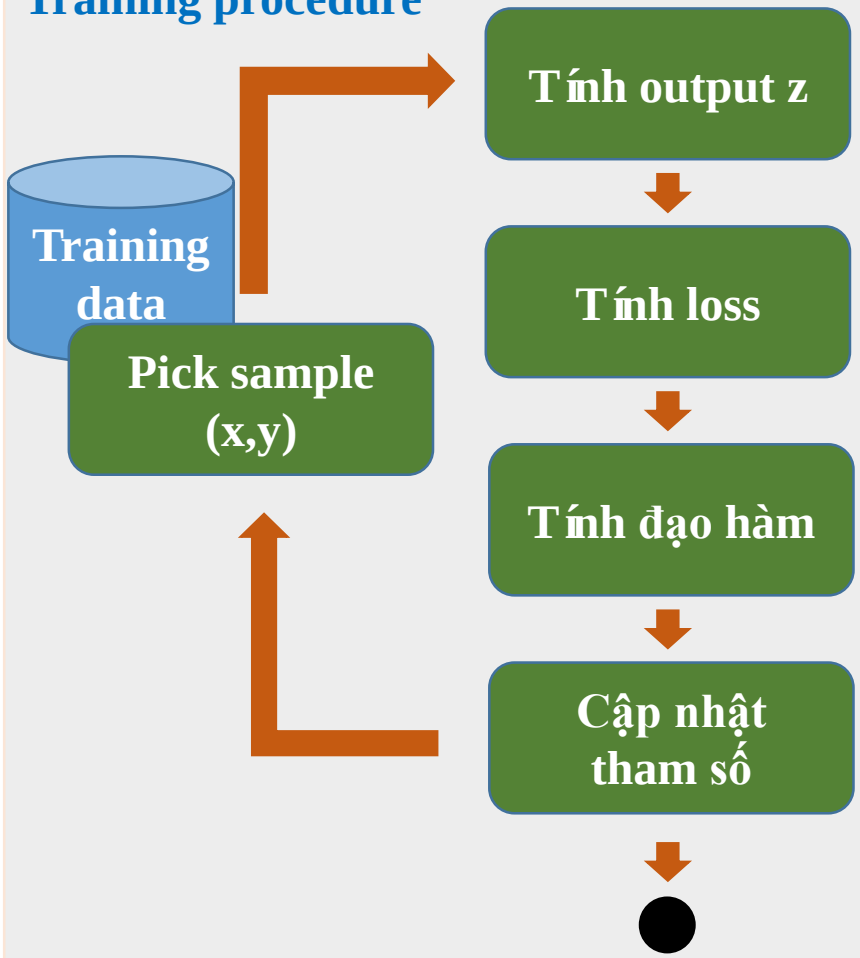
$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(o - y)$$

5) Cập nhật tham số

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate

Training procedure




Thank you!