AI VIETNAM
AI Insight Course

Linear Regression and Vectorization

Quang-Vinh Dinh
Ph.D. in Computer Science

Year 2020

Outline

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

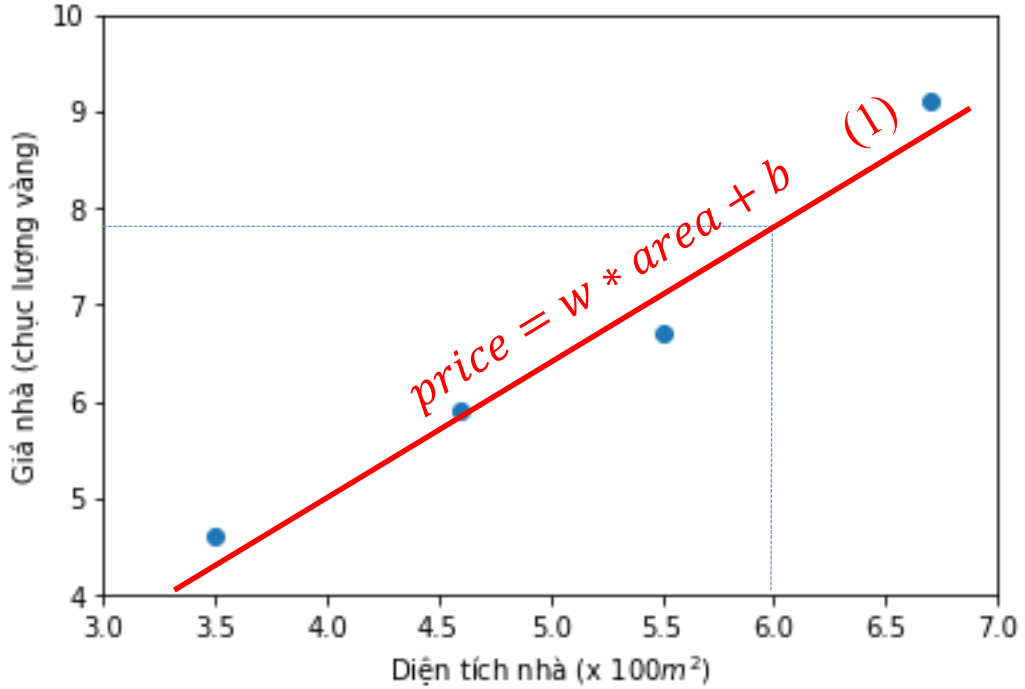
Computational graph

❖ 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

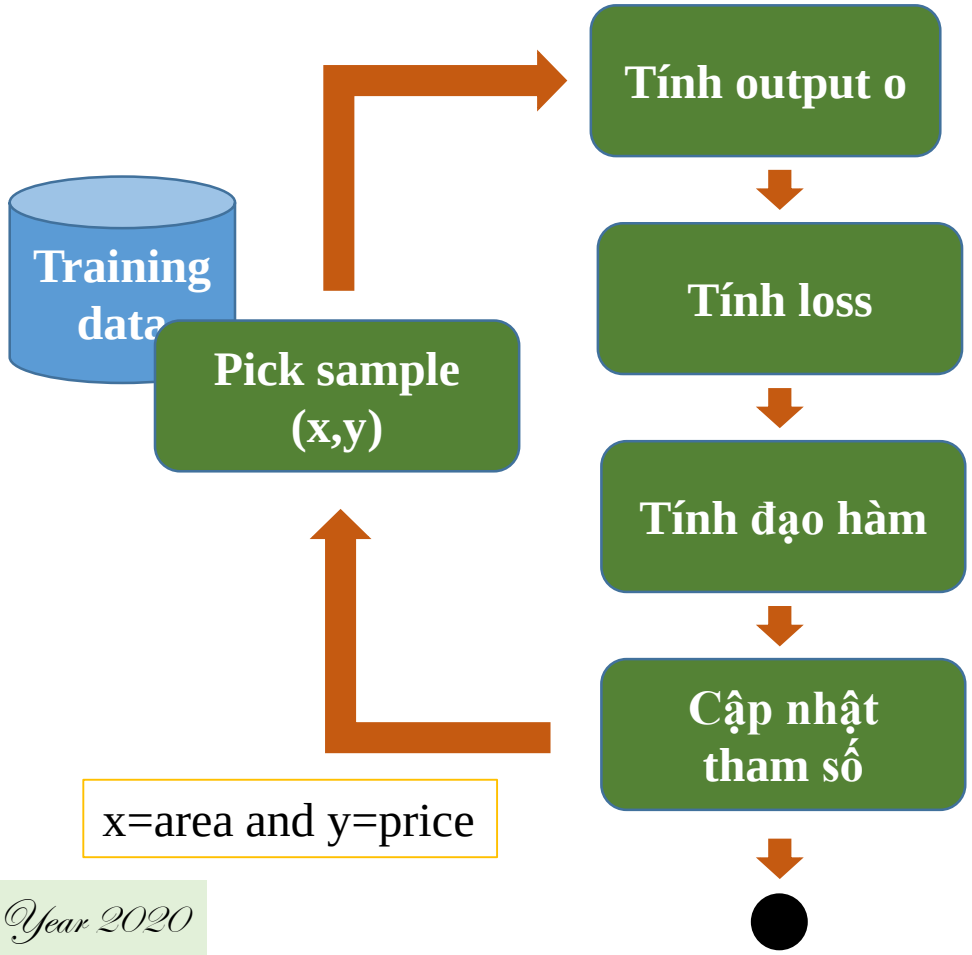
Given sample data



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

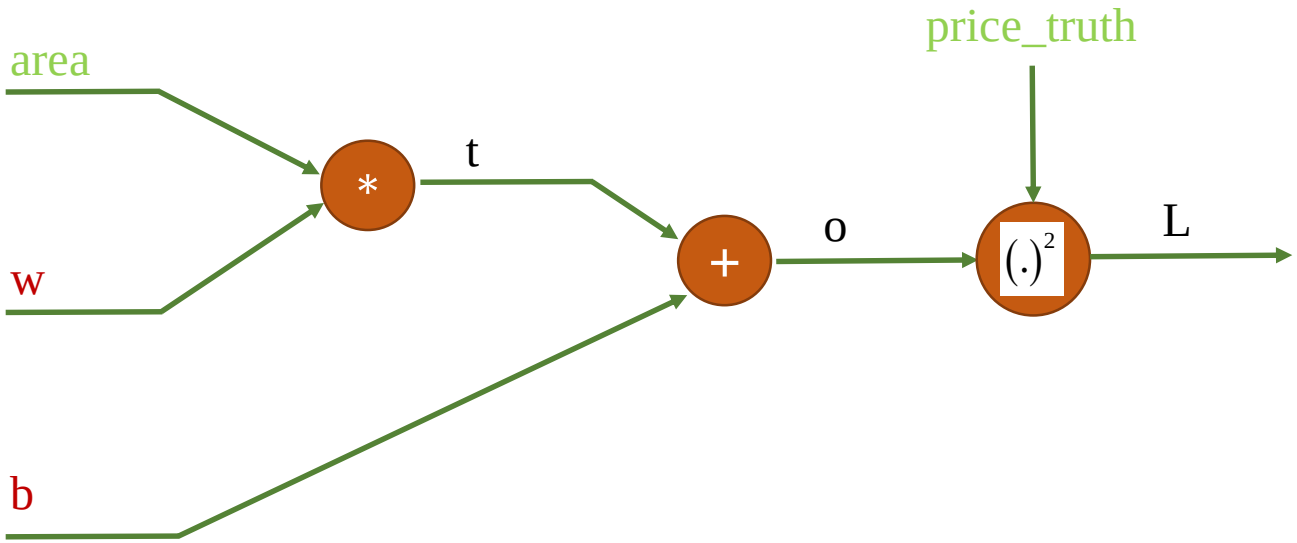
Computational graph

❖ House price prediction

❖ One-sample training

$$price = w * area + b$$

$$t = w * area$$

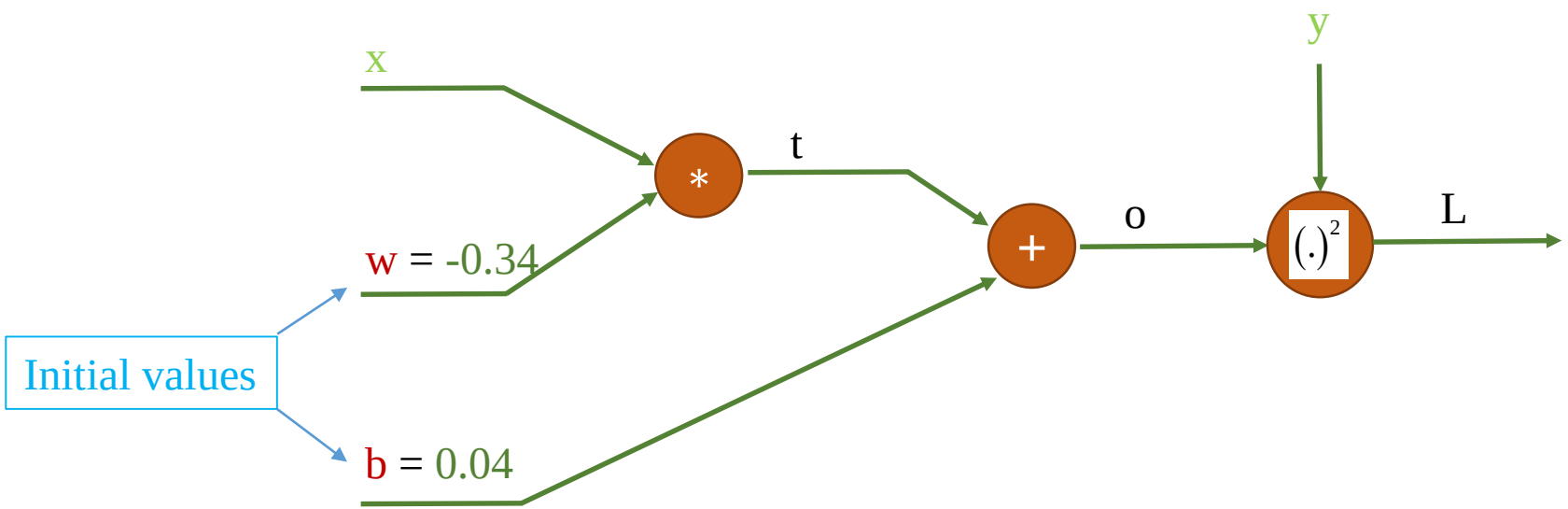


Computational graph

❖ House price prediction

❖ One-sample training

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



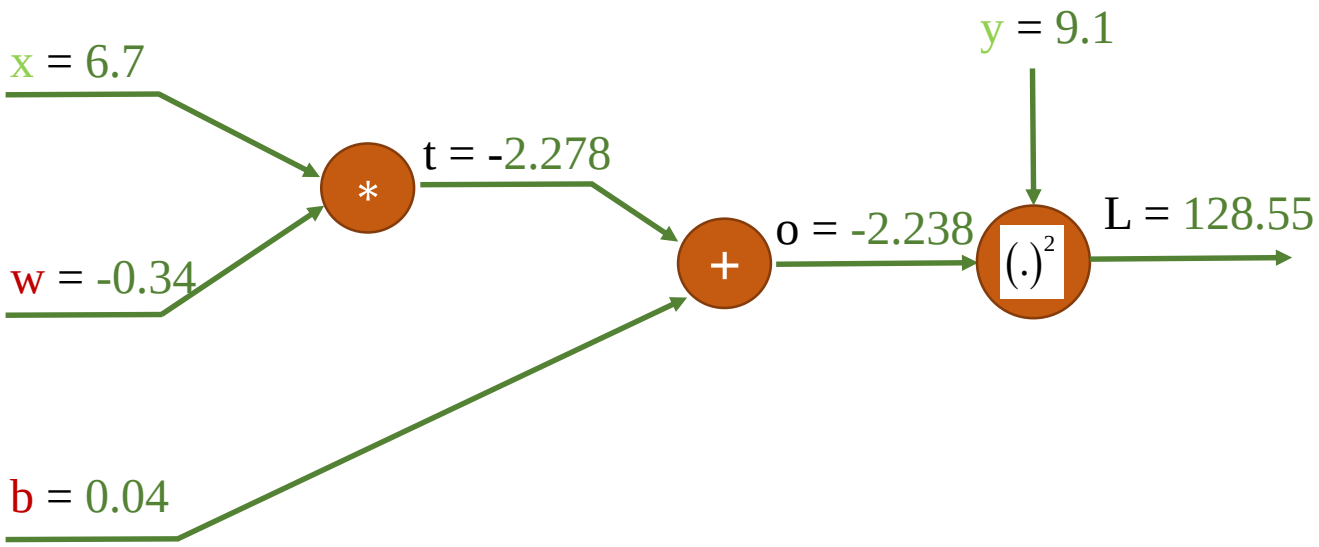
Computational graph

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

❖ House price prediction

❖ One-sample training

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

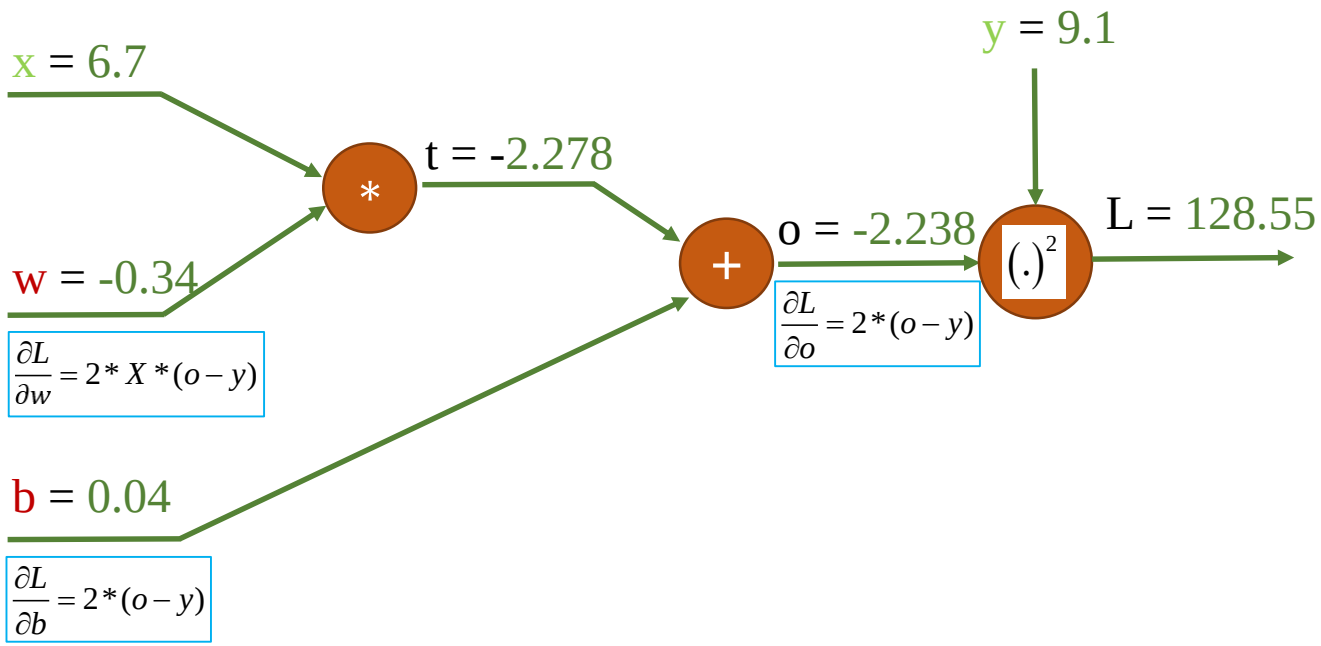


Computational graph

❖ House price prediction

❖ One-sample training

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

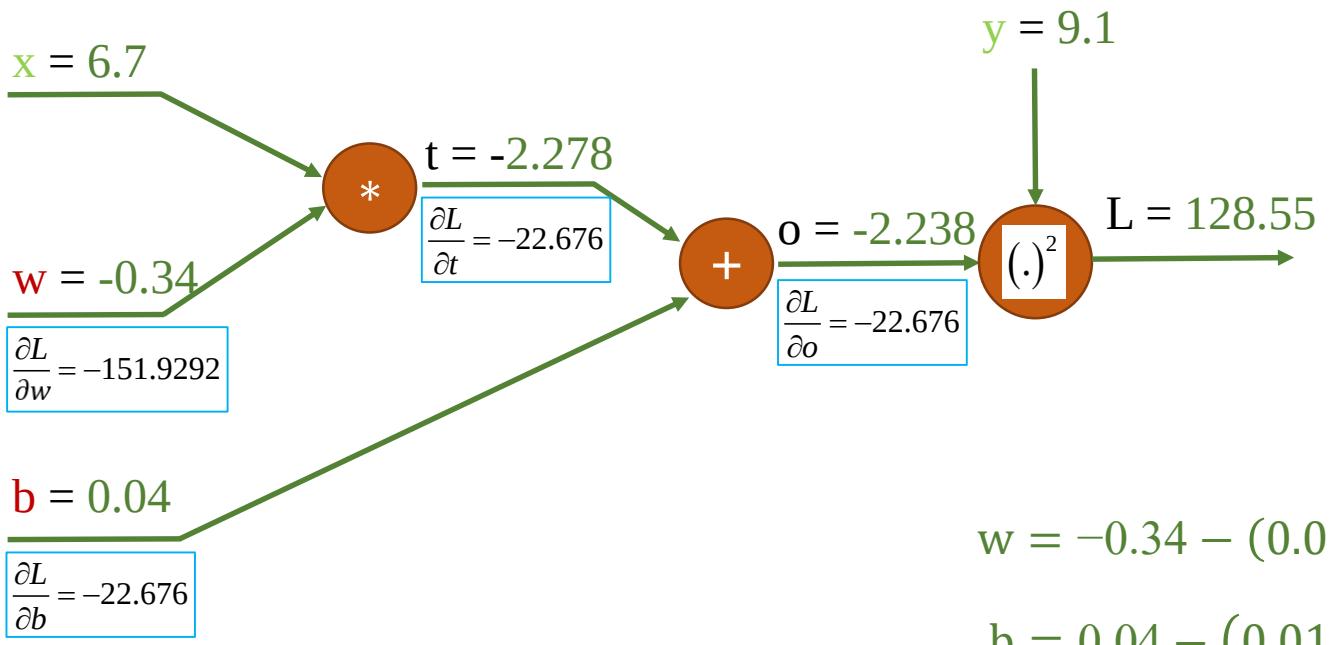


Computational graph

❖ 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 $\eta = 0.01$

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

$$b = 0.04 - (0.01 * (-22.67)) = 0.266$$

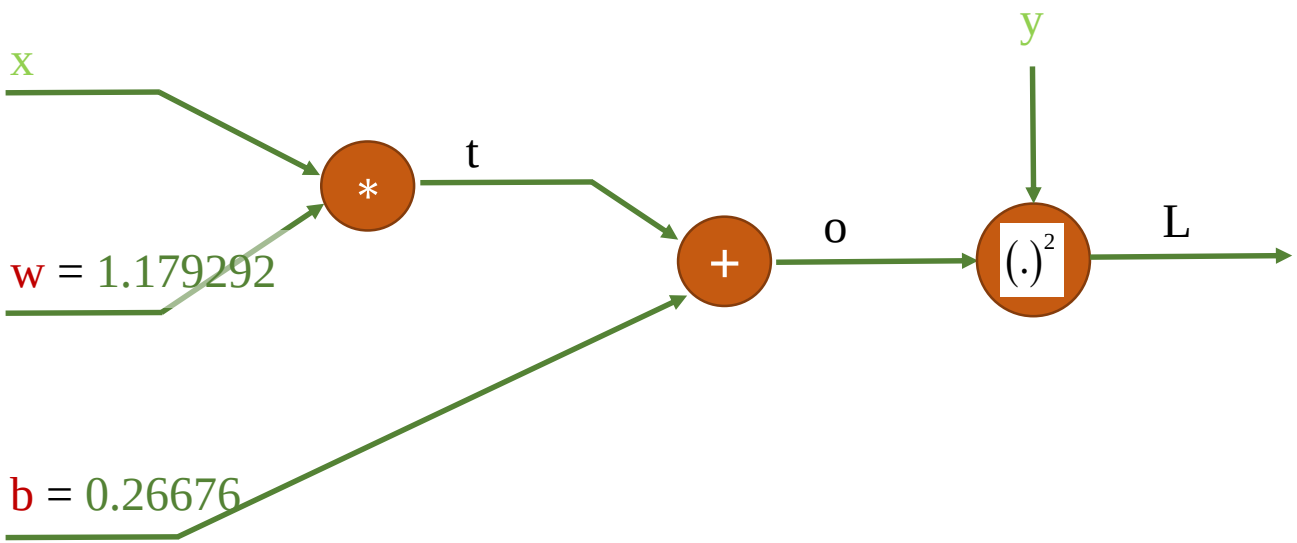
Computational graph

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

❖ House price prediction

❖ One-sample training

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



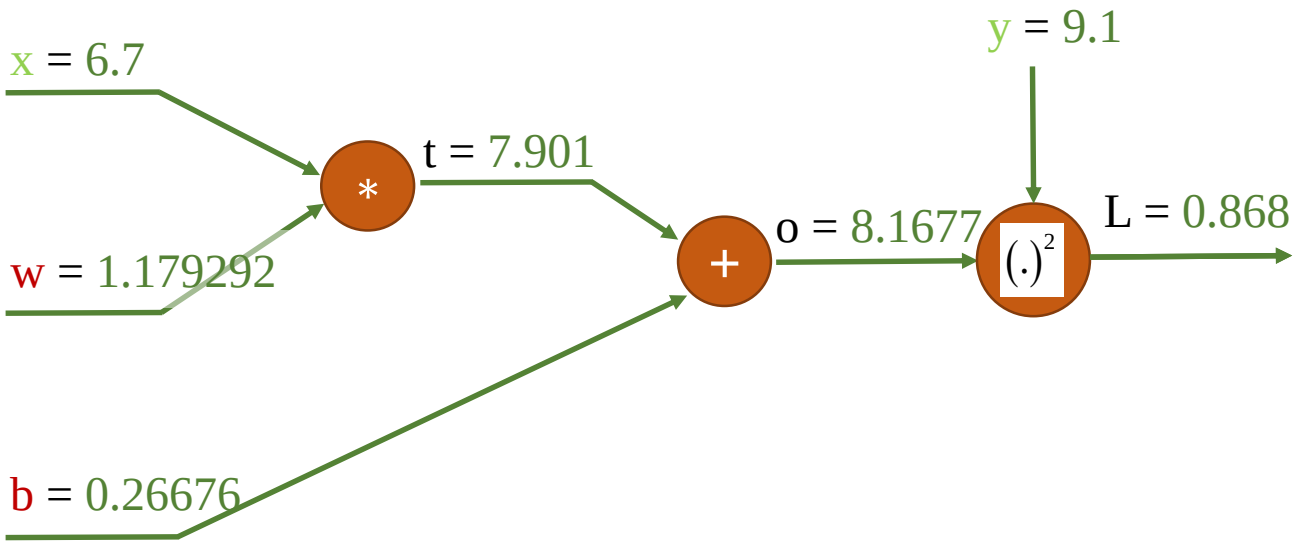
Computational graph

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

❖ House price prediction

❖ One-sample training

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



previous L = 128.55

Updated a and b values help to reduce the L value

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

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

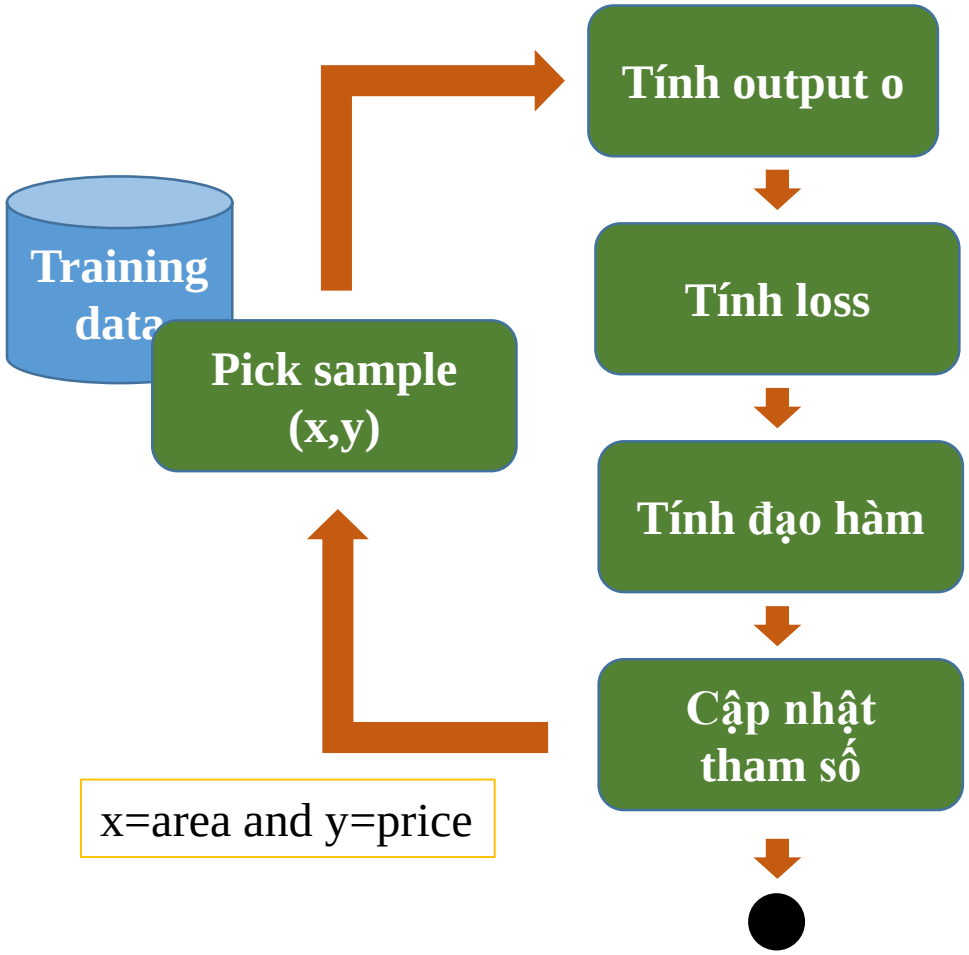
η is learning rate

Generalized formula

Computational graph

❖ House price prediction

❖ Implementation: Vectorization




```python
# Implementation - vectorization

# load data
import numpy as np
from numpy import genfromtxt
import matplotlib.pyplot as plt

data = genfromtxt('data.csv', delimiter=',')
areas  = data[:,0]
prices = data[:,1]
data_size = areas.size

print(type(areas))
print('areas: ', areas)
print('prices: ', prices)
print('data_size: ', data_size)

plt.scatter(areas, prices)
plt.xlabel('Diện tích nhà (x 100$m^2$)')
plt.ylabel('Giá nhà (chục lượng vàng)')
plt.xlim(3,7)
plt.ylim(4,10)
plt.show()
```

Computational graph

❖ House price prediction

❖ Implementation: Vectorization

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

```python
# forward
def predict(x,theta):
    return x.dot(theta)

# compute gradient
def gradient(z,y,x):
    dtheta = 2*x*(z-y)

    return dtheta

# update weights
def update_weight(theta,n,dtheta):
    dtheta_new = theta - n*dtheta

    return dtheta_new
```

Computational graph

❖ House price prediction

❖ Implementation: Vectorization

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

```
1  # vector [x, b]
2  data = np.c_[areas, np.ones((data_size, 1))]
3  print(data)
4
5  # init weight
6  n = 0.01
7  theta = np.array([-0.34, 0.04]) #[w, b]
8  print('theta', theta)
```

Computational graph

❖ House price prediction

❖ Implementation: Vectorization

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

```python
# how long
epoch_max = 10

for epoch in range(epoch_max):
    for i in range(data_size):
        # get a sample
        x = data[i]
        y = prices[i:i+1]
        print('sample: ', x, y)

        # predict z
        z = predict(x, theta)
        print('z: ', z)

        # compute loss
        loss = (z-y)*(z-y)
        print('Loss: ', loss)

        # compute gradient
        dtheta = gradient(z,y,x)
        print('dtheta: ', dtheta)

        # update weights
        theta = update_weight(theta,n,dtheta)
        print('theta_new: ', theta)
        print('\n\n')
```

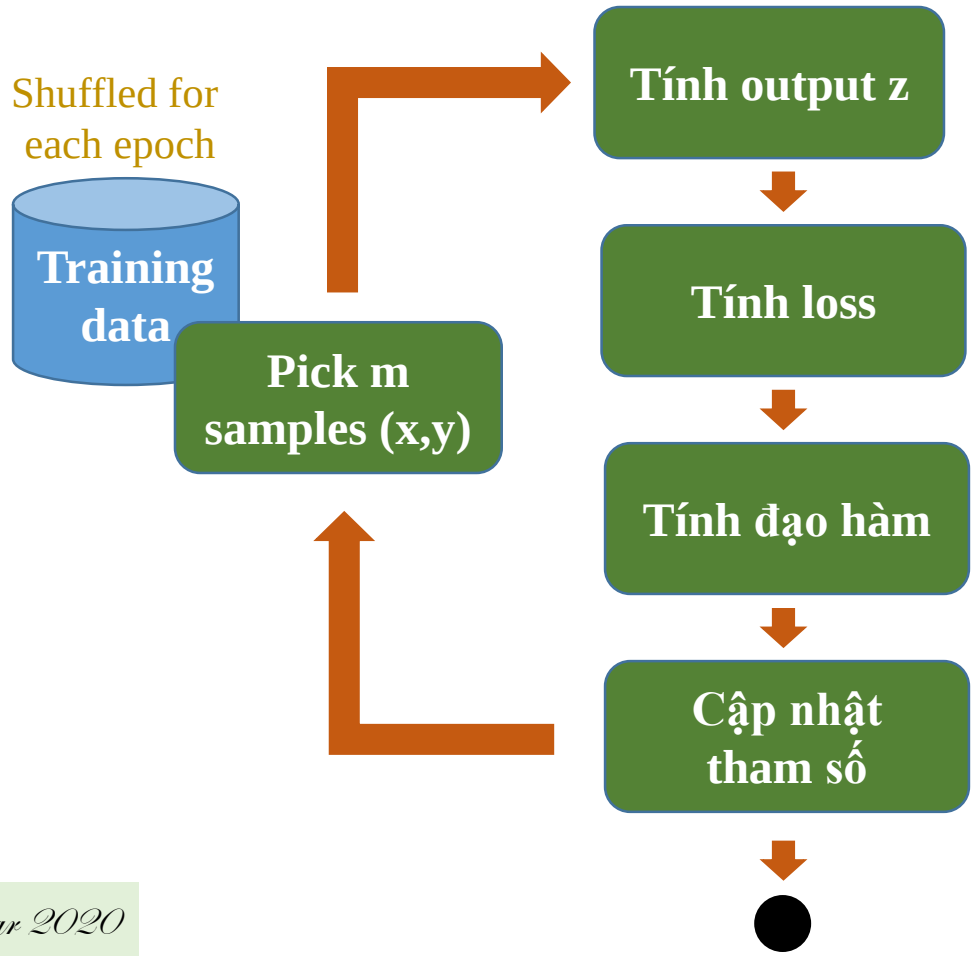
Outline

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

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

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

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

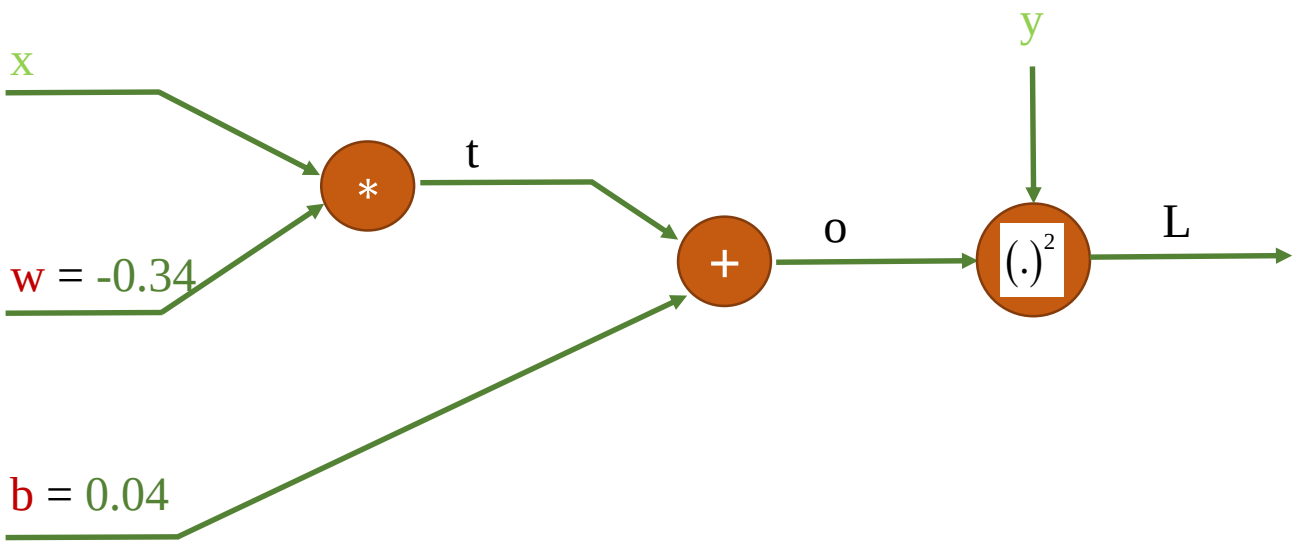
Learning rate η

Computational graph

❖ House price prediction

❖ m-sample training (1<m<N)

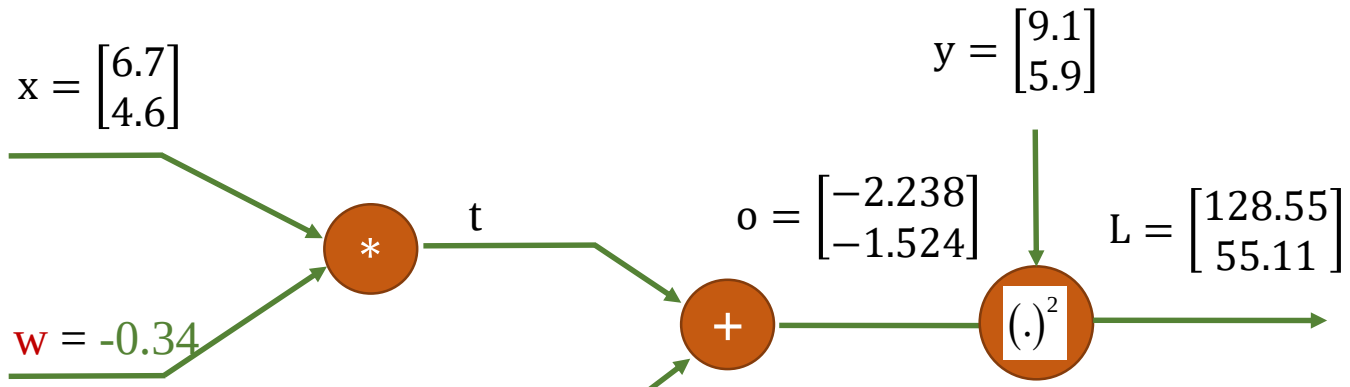
$$t = wx \qquad o = t + b \qquad L = (o - y)^2$$



Computational graph

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

m = 2



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

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

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

Computational graph

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

Cách cập nhật a và b

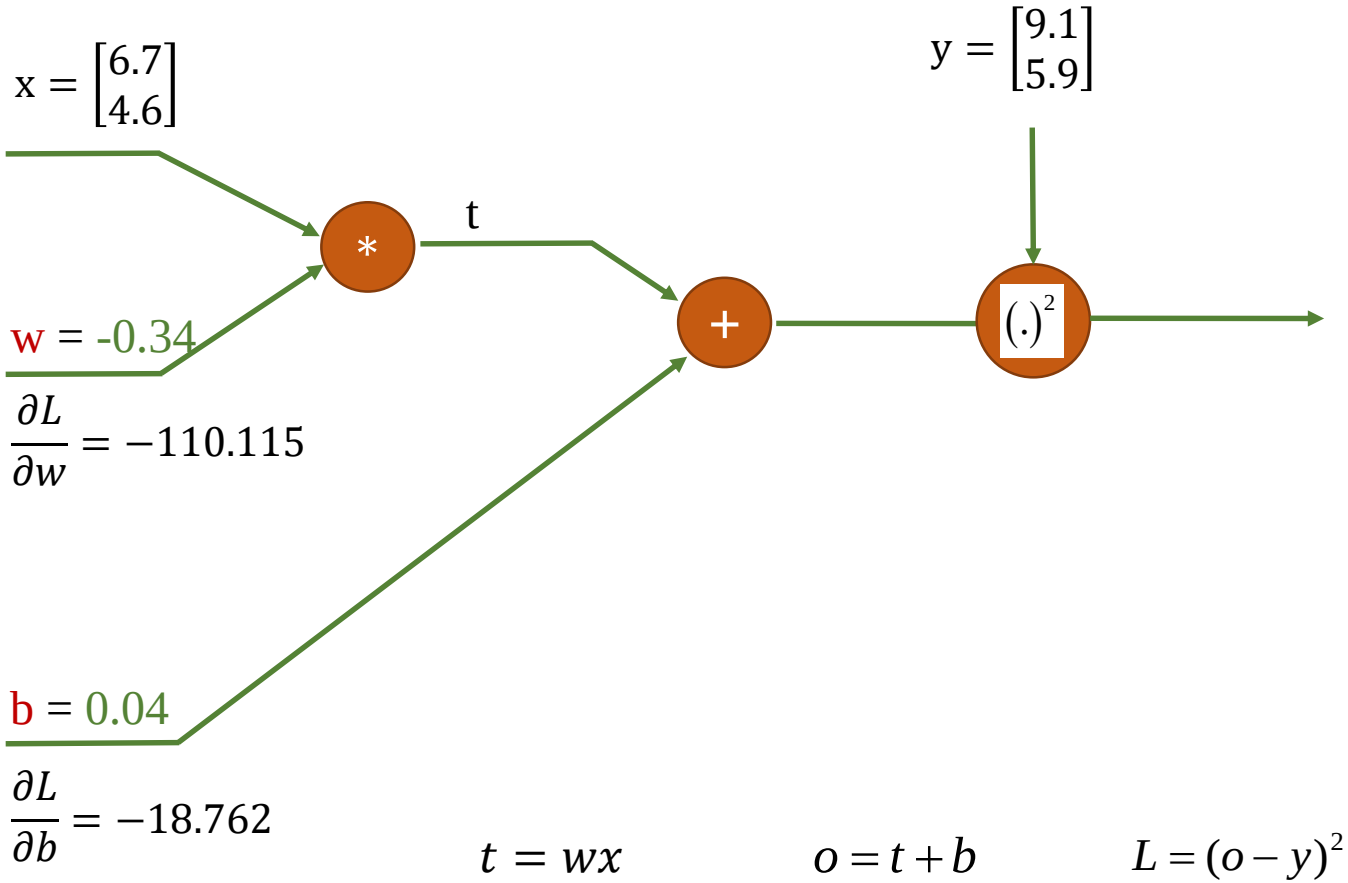
$$w = w - \eta * \frac{\partial L}{\partial w}$$

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

Learning rate $\eta = 0.01$

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

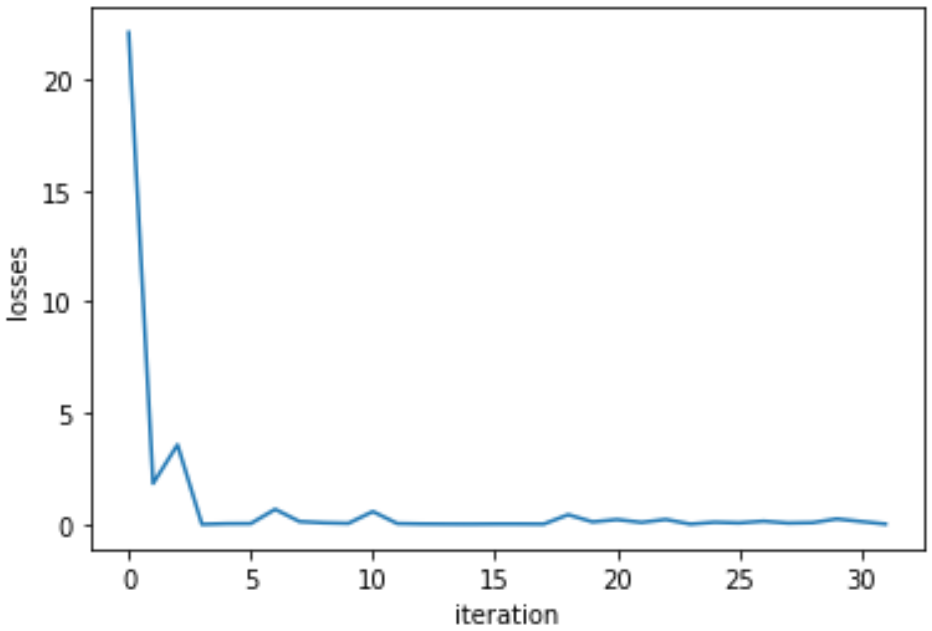
$$b = 0.04 - (0.01 * (-18.762)) = 0.227$$



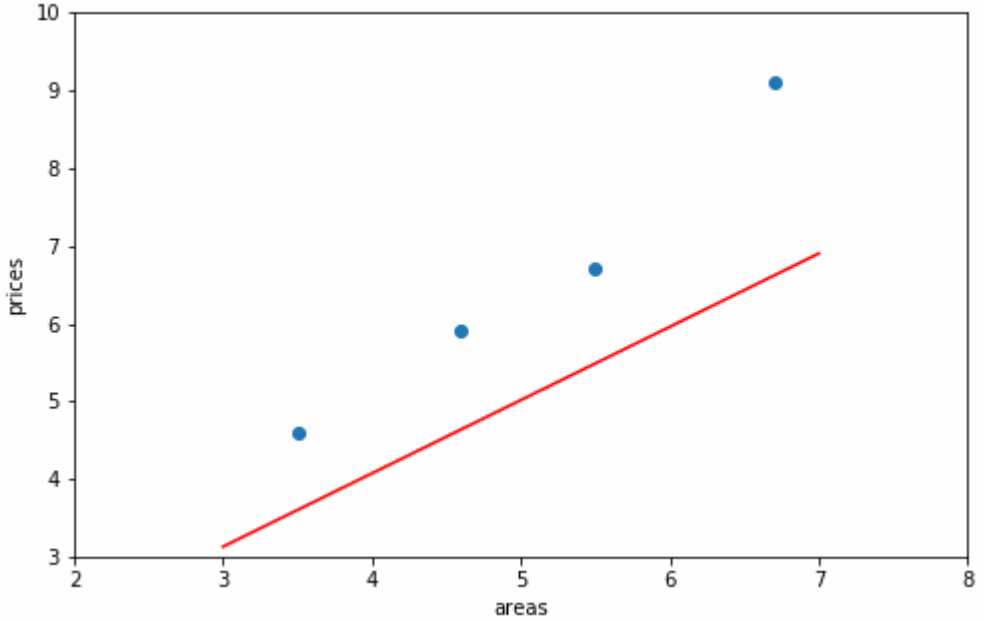
$t = wx$ $o = t + b$ $L = (o - y)^2$

Computational graph

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



Losses for 30 iterations



Model updating for different iterations

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 (m-samples)

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

```python
# load data

import numpy as np
from numpy import genfromtxt
import matplotlib.pyplot as plt

data = genfromtxt('data.csv', delimiter=',')
areas  = data[:,0]
prices = data[:,1]
data_size = areas.size

print(type(areas))
print('areas: ', areas)
print('prices: ', prices)
print('data_size: ', data_size)

plt.scatter(areas, prices)
plt.xlabel('areas')
plt.ylabel('prices')
plt.xlim(3,7)
plt.ylim(4,10)
plt.show()
```

Linear Regression (m-samples)

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

```
1  # vector [x, b]
2  data = np.c_[areas, np.ones((data_size, 1))]
3  print(data)
4
5  # init weight
6  eta = 0.01
7  theta = np.array([-0.34, 0.04]) #[w, b]
8  print('theta', theta)
9
10 # how long
11 epoch_max = 1
12
13 # mini-batch size
14 m = 2
```

Linear Regression (m-samples)

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

```
for epoch in range(epoch_max):
    for j in range(0, data_size, m):

        gradients = np.zeros((2,))
        for index in range(j, j+m):
            xi = data[index]
            yi = prices[index]

            # predict z/o
            oi = xi.dot(theta)

            # compute loss
            li = (oi - yi)*(oi - yi)

            # compute gradient
            g_li = 2*(oi - yi)
            gradient_i = xi*g_li

            gradients = gradients + gradient_i
            sum_of_losses = sum_of_losses + li

        gradients     = gradients/2
        theta = theta - eta*gradients
```

Generalized formula

Linear Regression (m-samples)

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

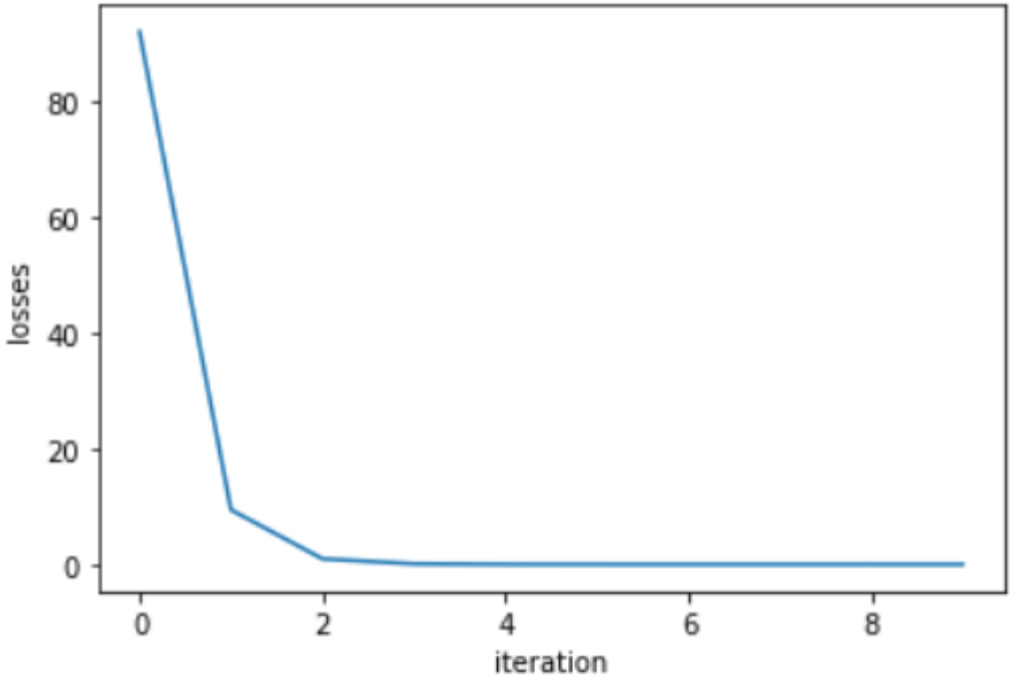
2) Cập nhật tham số

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

η is learning rate

```python
import matplotlib.pyplot as plt

plt.plot(losses)
plt.xlabel('iteration')
plt.ylabel('losses')
plt.show()
```



Generalized formula

Linear Regression (m-samples)

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

More vectorization

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

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix}$$

Generalized formula

Linear Regression (m-samples)

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

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix}$$

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

$$= [-0.34 \quad 0.049] \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} = [-2.238 \quad -1.524]$$

Generalized formula

Linear Regression (m-samples)

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

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix}$$

$$\boldsymbol{o} = \boldsymbol{\theta}^T \boldsymbol{x} = [-2.238 \quad -1.524]$$

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y}^T)^2$$

$$= ([-2.238 \quad -1.524] - [9.1 \quad 5.9])^2$$

$$= [128.5 \quad 55.11]$$

Generalized formula

Linear Regression (m-samples)

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

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix}$$

$$\boldsymbol{o} = \boldsymbol{\theta}^T \boldsymbol{x} = [-2.238 \quad -1.524]$$

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y}^T)^2 = [128.5 \quad 55.11]$$

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y}^T) = [-22.676 \quad -14.848]$$

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

Generalized formula

Linear Regression (m-samples)

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

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix}$$

$$\boldsymbol{o} = \boldsymbol{\theta}^T \boldsymbol{x} = [-2.238 \quad -1.524]$$

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y}^T)$$

$$= [-22.676 \quad -14.848]$$

$$\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} = \begin{bmatrix} -22.676 & -14.848 \\ -22.676 & -14.848 \end{bmatrix}$$

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y}^T)^2$$

$$= [128.5 \quad 55.11]$$

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

Linear Regression (m-samples)

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

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix}$$

$$\boldsymbol{o} = \boldsymbol{\theta}^T \boldsymbol{x} = [-2.238 \quad -1.524]$$

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y}^T) = 2(\boldsymbol{o} - \boldsymbol{y}^T)$$

$$= [-22.676 \quad -14.848]$$

$$\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} = \begin{bmatrix} -22.676 & -14.848 \\ -22.676 & -14.848 \end{bmatrix}$$

Gradient for w from $x^{(i)}$

$$\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} = \begin{bmatrix} -151.92 & -68.301 \\ -22.676 & -14.848 \end{bmatrix}$$

Gradient for b from $x^{(i)}$

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y}^T)^2$$

$$= [128.5 \quad 55.11]$$

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

Generalized formula

AI VIETNAM
AI Insight Course

Linear Regression (m-samples)

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

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

Generalized formula

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix}$$

$$\boldsymbol{o} = \boldsymbol{\theta}^T \boldsymbol{x} = [-2.238 \quad -1.524]$$

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y}^T) = 2(\boldsymbol{o} - \boldsymbol{y}^T)$$

$$= [-22.676 \quad -14.848]$$

$$\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} = \begin{bmatrix} -22.676 & -14.848 \\ -22.676 & -14.848 \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} = \begin{bmatrix} -151.92 & -68.301 \\ -22.676 & -14.848 \end{bmatrix}$$

$$L'_{\theta} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -151.92 & -68.301 \\ -22.676 & -14.848 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -220.21 \\ -37.524 \end{bmatrix}$$

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y}^T)^2$$

$$= [128.5 \quad 55.11]$$

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

Linear Regression (m-samples)

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

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix}$$

$$\boldsymbol{o} = \boldsymbol{\theta}^T \boldsymbol{x} = [-2.238 \quad -1.524]$$

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y}^T)^2 = [128.5 \quad 55.11]$$

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y}^T)$$

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -220.21 \\ -37.524 \end{bmatrix}$$

Generalized formula

Linear Regression (m-samples)

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

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

1.1) Tính output $o^{(i)}$

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

1.2) Tính loss

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

1.3) Tính đạo hàm

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

2) Cập nhật tham số

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

η is learning rate

$$\boldsymbol{x} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix}$$

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y}^T)^2 = [128.5 \quad 55.11]$$

$$\boldsymbol{o} = \boldsymbol{\theta}^T \boldsymbol{x} = [-2.238 \quad -1.524]$$

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y}^T)$$

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -220.21 \\ -37.524 \end{bmatrix}$$

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

$$= \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} - 0.005 \begin{bmatrix} -220.21 \\ -37.524 \end{bmatrix} = \begin{bmatrix} 0.761 \\ 0.227 \end{bmatrix}$$

Generalized formula

Linear Regression (m-samples)

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

1.1) Tính output $o^{(i)}$

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

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

1) Pick m samples $(\boldsymbol{x}, \boldsymbol{y})$ from training data

1.1) Tính output $\boldsymbol{o}$

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

1.2) Tính loss

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y})^2$$

1.3) Tính đạo hàm

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y})$$

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$\odot$ is element-wise multiplication

1.4) Cập nhật tham số

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

η is learning rate

More generalized formula

Linear Regression (m-samples)

1) Pick m samples $(\boldsymbol{x}, \boldsymbol{y})$ from training data

1.1) Tính output $\boldsymbol{o}$

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

1.2) Tính loss

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y})^2$$

1.3) Tính đạo hàm

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y})$$

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$\odot$ is element-wise multiplication

1.4) Cập nhật tham số

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

η is learning rate

```python
import numpy as np
from numpy import genfromtxt

data = genfromtxt('data.csv', delimiter=',')
areas  = data[:,0]
prices = data[:,1:]
data_size = areas.size

# vector [x, b]
data = np.c_[areas, np.ones((data_size, 1))]
data = data.T

# init weight
eta = 0.01
theta = np.array([[-0.34], [0.04]]) #[w, b]

# how long
epoch_max = 10

# mini-batch size
m = 2
```

More generalized formula

Linear Regression (m-samples)

1) Pick m samples $(\boldsymbol{x}, \boldsymbol{y})$ from training data

1.1) Tính output $\boldsymbol{o}$

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

1.2) Tính loss

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y})^2$$

1.3) Tính đạo hàm

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y})$$

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$\odot$ is element-wise multiplication

1.4) Cập nhật tham số

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

η is learning rate

```
for epoch in range(epoch_max):
    for i in range(0, data_size, m):
        # get m samples
        x = data[:, i:i+m]
        y = prices[i:i+m, :]

        # predict z/o
        z = theta.T.dot(x)

        # compute loss
        loss = np.multiply((z-y.T), (z-y.T))
        losses.append(np.mean(loss))

        # compute gradient
        b = 2*(z-y.T)
        gradients = np.multiply(x, np.vstack((b, b)))
        gradients = gradients.dot(np.ones((m, 1)))/m

        # update weights
        theta = theta - eta*gradients
```

More generalized formula

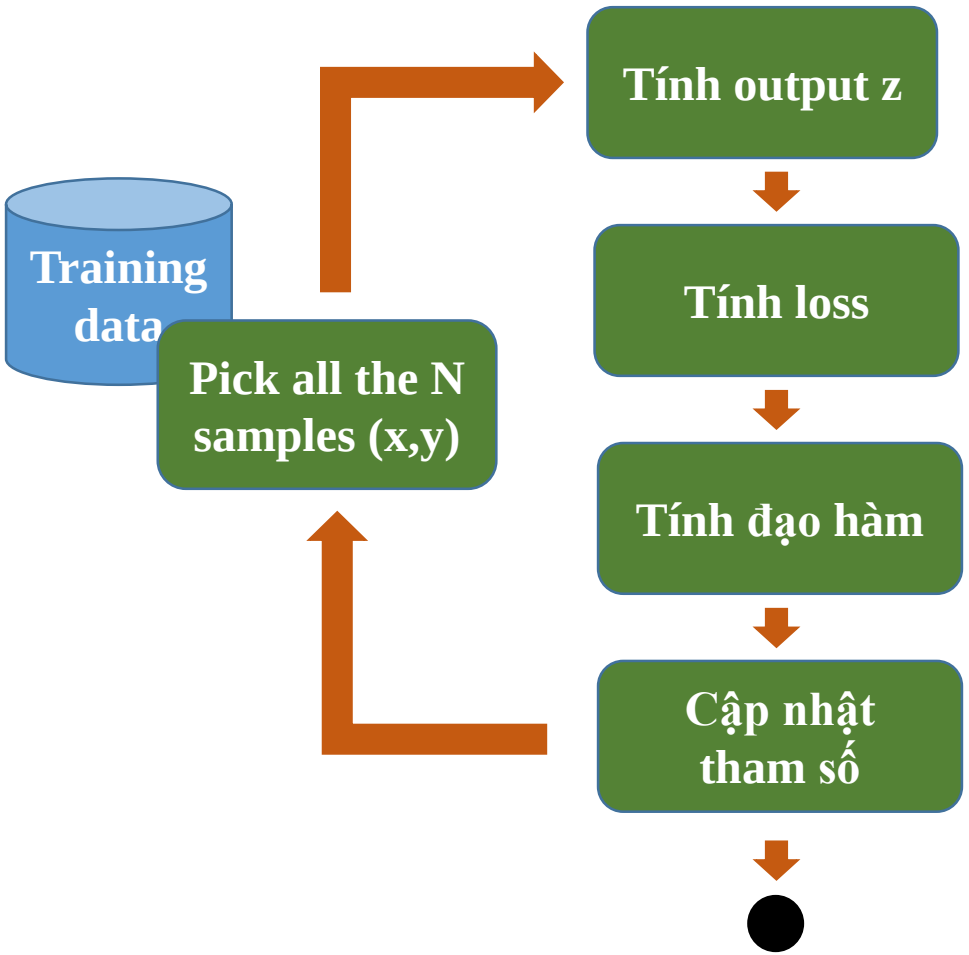
Outline

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

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

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

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

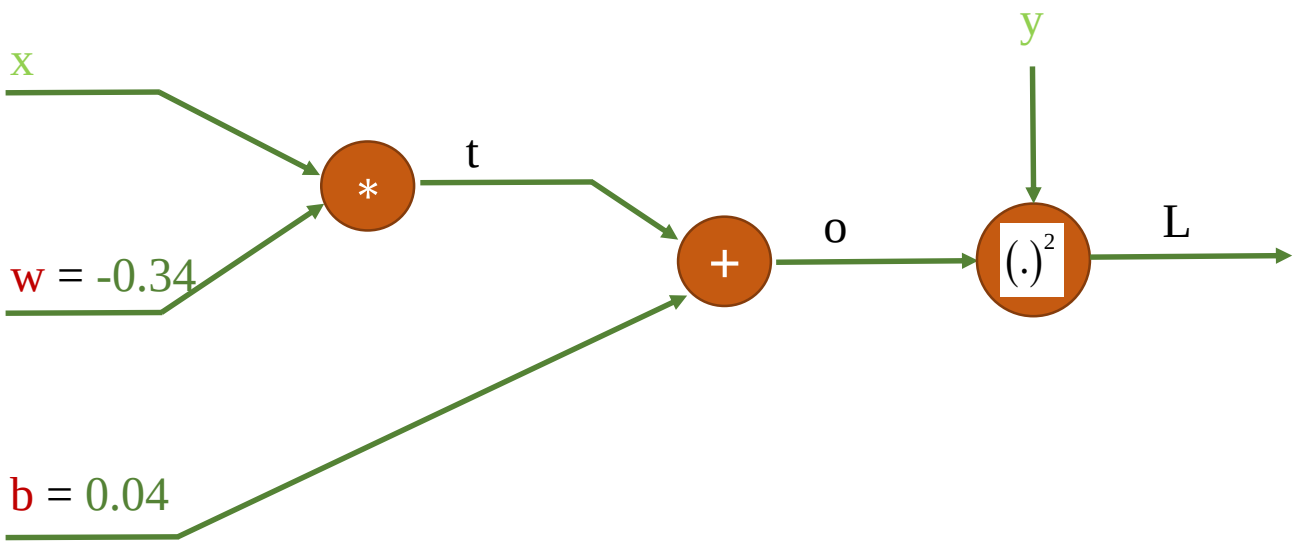
Learning rate η

Computational graph

❖ House price prediction

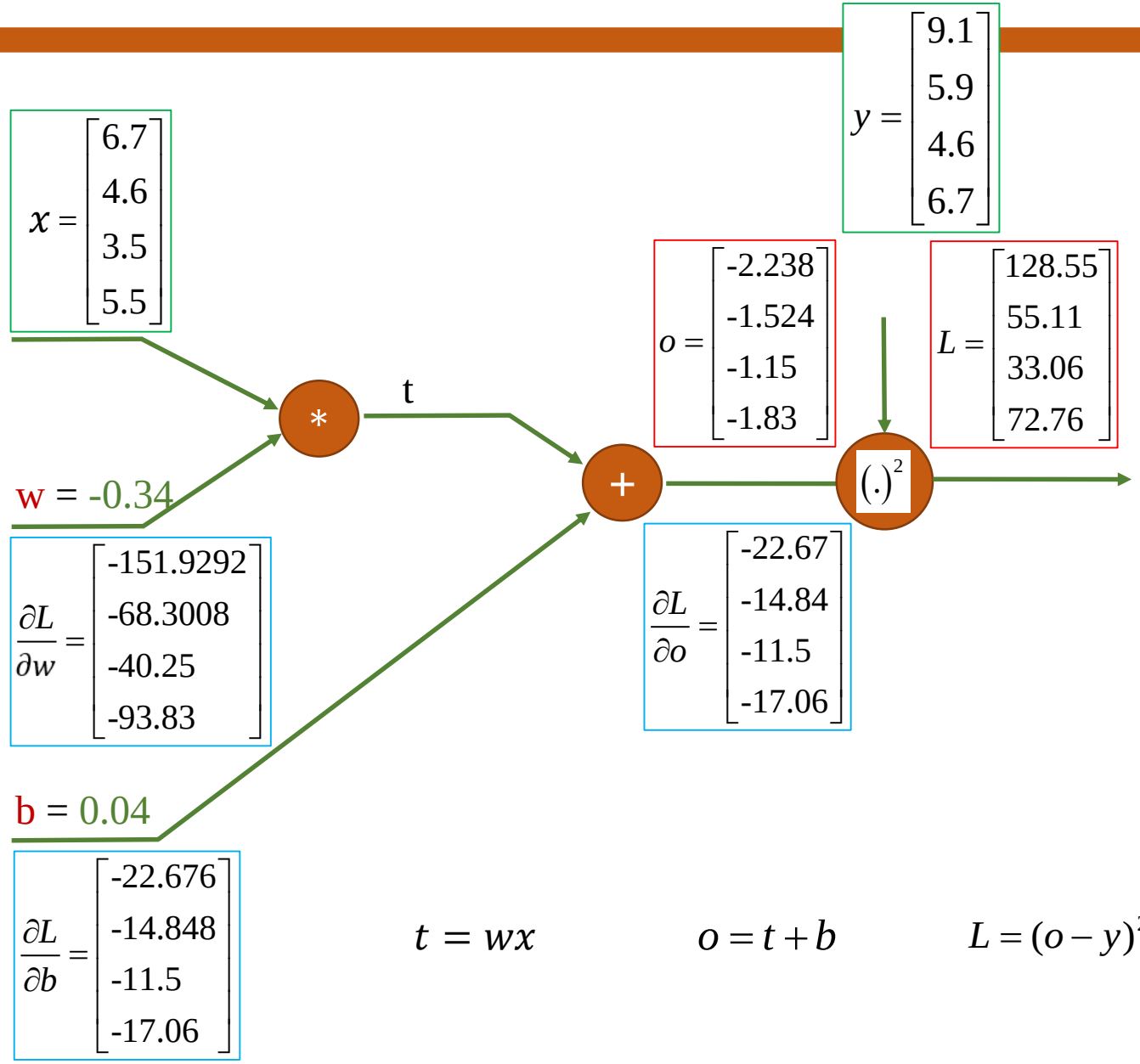
❖ N-sample training

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



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

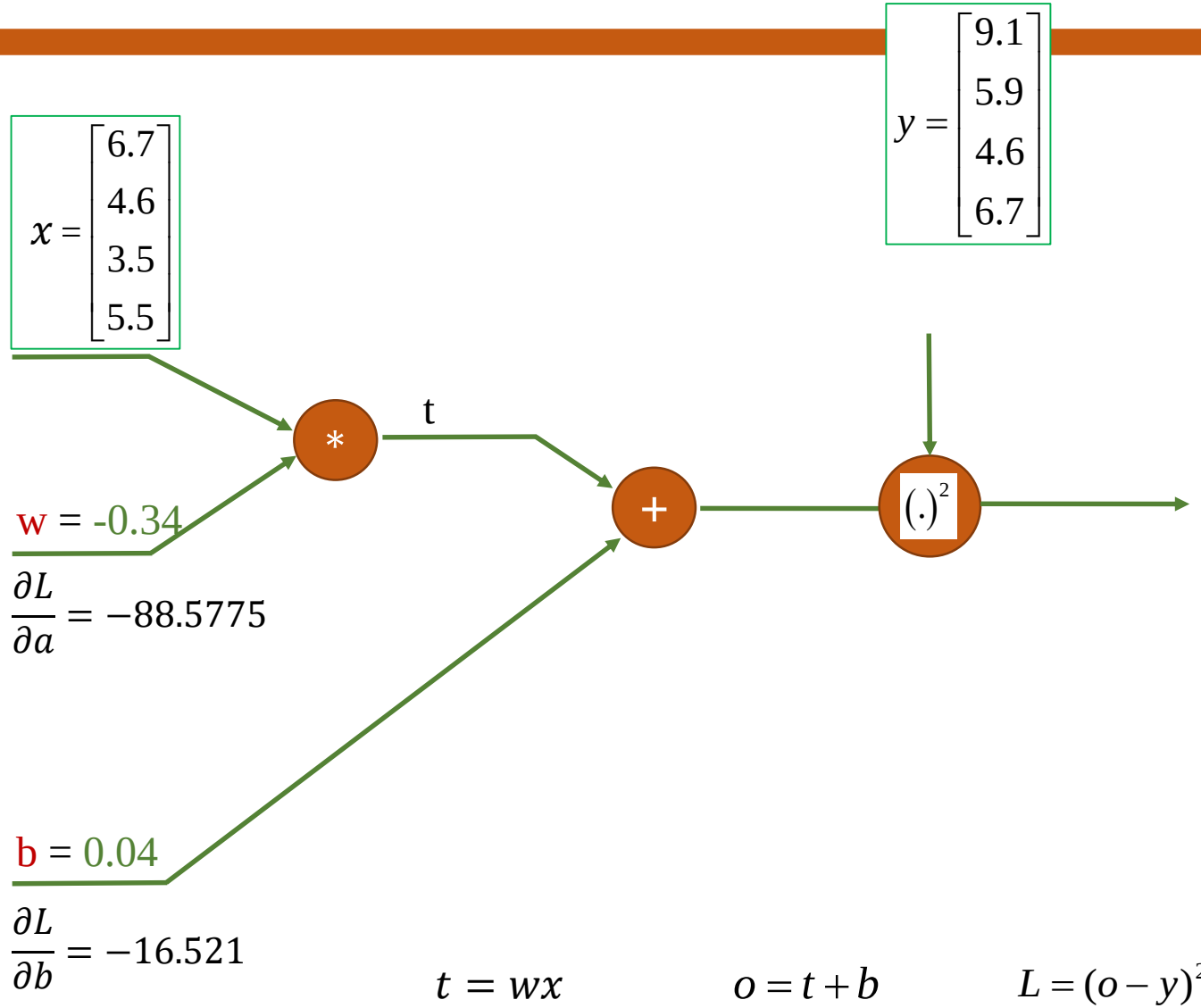
$$w = w - \eta * \frac{\partial L}{\partial w}$$

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

Learning rate $\eta = 0.01$

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

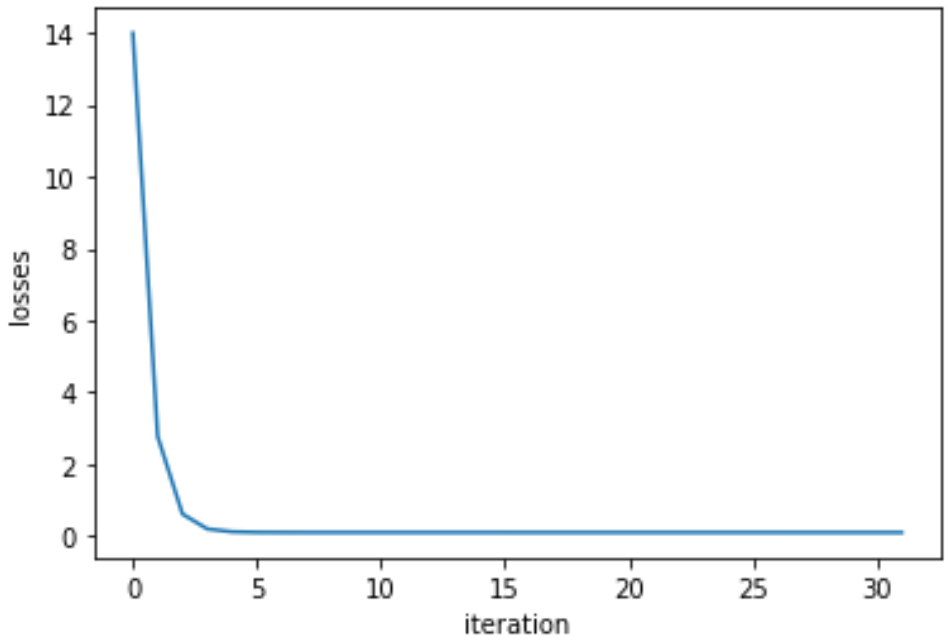
$$b = 0.04 - \left(0.01 * (-16.521)\right) = 0.205$$



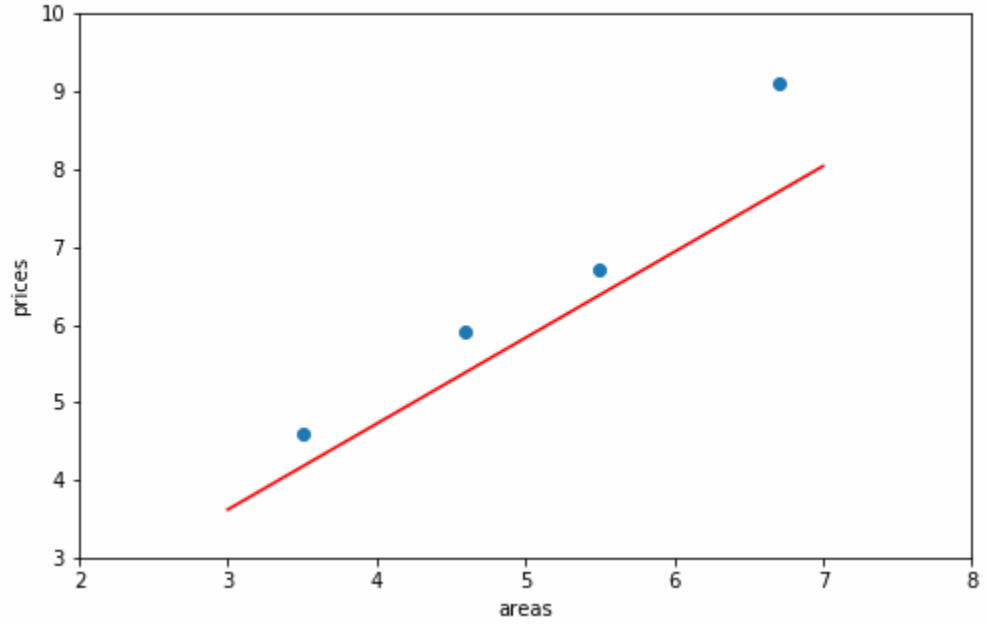
Computational graph

❖ House price prediction

❖ N-sample training



Losses for 30 iterations



Model updating for different iterations

Linear Regression (N-samples)

1) Pick all the N samples from training data

2) Tính output $o^{(i)}$

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

3) Tính loss

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

4) Tính đạo hàm

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

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 $o^{(i)}$

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

3) Tính loss

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

4) Tính đạo hàm

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

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 (N-samples)

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

1) Pick all the N samples from training data

2) Tính output $\boldsymbol{o}$

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

3) Tính loss

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y})^2$$

4) Tính đạo hàm

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y})$$

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$\odot$ is element-wise multiplication

5) Cập nhật tham số

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

η is learning rate

More generalized formula

Linear Regression (N-samples)

1) Pick all the N samples from training data

2) Tính output $\boldsymbol{o}$

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

3) Tính loss

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y})^2$$

4) Tính đạo hàm

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y})$$

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$\odot$ is element-wise multiplication

5) Cập nhật tham số

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

η is learning rate

```python
# full code
import numpy as np
from numpy import genfromtxt

data = genfromtxt('data.csv', delimiter=',')
areas  = data[:,0]
prices = data[:,1:]
data_size = areas.size

# vector [x, b]
data = np.c_[areas, np.ones((data_size, 1))]
data = data.T

n_epochs = 10
eta = 0.01

# init weight
theta = np.array([[-0.34],[0.04]])
```

Linear Regression (N-samples)

1) Pick all the N samples from training data

2) Tính output $\boldsymbol{o}$

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

3) Tính loss

$$\boldsymbol{L} = (\boldsymbol{o} - \boldsymbol{y})^2$$

4) Tính đạo hàm

$$\boldsymbol{k} = 2(\boldsymbol{o} - \boldsymbol{y})$$

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \boldsymbol{x} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$\odot$ is element-wise multiplication

5) Cập nhật tham số

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

η is learning rate

```
20 losses = [] # for debug
21 for epoch in range(n_epochs):
22     # compute output
23     z = theta.T.dot(data)
24
25     # compute loss
26     loss = np.multiply((z-prices.T), (z-prices.T))
27     losses.append(np.mean(loss))
28
29     # compute gradient
30     b = 2*(z-prices.T)
31     gradient = np.multiply(data, np.vstack((b, b)))
32     gradient = gradient.dot(np.ones((data_size, 1)))/data_size
33
34     # update weights
35     theta = theta - eta*gradient
```

More generalized formula

Linear Regression

	Advantages	Disadvantages
1 sample	Simple to understand and implement Faster learning on some problems Noisy update is beneficial sometime	Computationally expensive Noisy gradient signal Convergence problem
m sample	A balance between the robustness of 1-sample and the efficiency of N-sample	
N sample	Computationally efficient More stable error gradient parallel processing	Premature convergence Memory problem Training speed is slower


Thank you!