Vectorized Implementation for Linear Regression

(Naive and Curious Appoach)

Quang-Vinh Dinh
Ph.D. in Computer Science

Year 2024

Motivation

❖ Linear regression

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

House price data

TV (Features)	Radio (Features)	Newspaper (Features)	Sales (Label)
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: $\hat{y} = w_1x_1 + b$

price = a $* area + b$

Model: $\hat{y} = w_1x_1 + w_2x_2 + w_3x_3 + b$

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

Linear Regression

1) Pick a sample (x_1, x_2, x_3, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = w_1 * TV + w_2 * R + w_3 * N + b$$

$$\hat{y} = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + b$$

3) Compute loss

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

4) Compute derivative

$$\frac{\partial L}{\partial w_1} = 2x_1(\hat{y} - y) \qquad \frac{\partial L}{\partial w_3} = 2x_3(\hat{y} - y)$$

$$\frac{\partial L}{\partial w_2} = 2x_2(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w_1 = w_1 - \eta \frac{\partial L}{\partial w_1} \qquad w_3 = w_3 - \eta \frac{\partial L}{\partial w_3}$$

$$w_2 = w_2 - \eta \frac{\partial L}{\partial w_2} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

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

$$\text{Sale} = w_1 * TV + w_2 * Radio + w_3 * Newspaper + b$$

$$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

1) Pick a sample (x_1, x_2, x_3, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = w_1 * TV + w_2 * R + w_3 * N + b$$

$$\hat{y} = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + b$$

3) Compute loss

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

4) Compute derivative

$$\frac{\partial L}{\partial w_1} = 2x_1(\hat{y} - y) \qquad \frac{\partial L}{\partial w_3} = 2x_3(\hat{y} - y)$$

$$\frac{\partial L}{\partial w_2} = 2x_2(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w_1 = w_1 - \eta \frac{\partial L}{\partial w_1} \qquad w_3 = w_3 - \eta \frac{\partial L}{\partial w_3}$$

$$w_2 = w_2 - \eta \frac{\partial L}{\partial w_2} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

```python
# compute output and loss
def predict(x1, x2, x3, w1, w2, w3, b):
    return w1*x1 + w2*x2 + w3*x3 + b
def compute_loss(y_hat, y):
    return (y_hat - y)**2

# compute gradient
def compute_gradient_wi(xi, y, y_hat):
    dl_dwi = 2*xi*(y_hat-y)
    return dl_dwi
def compute_gradient_b(y, y_hat):
    dl_db = 2*(y_hat-y)
    return dl_db

# update weights
def update_weight_wi(wi, dl_dwi, lr):
    wi = wi - lr*dl_dwi
    return wi
def update_weight_b(b, dl_db, lr):
    b  = b - lr*dl_db
    return b
```

Motivation

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

if area=6.0, price=?

if TV=55.0, Radio=34.0, and Newspaper=62.0, price=?

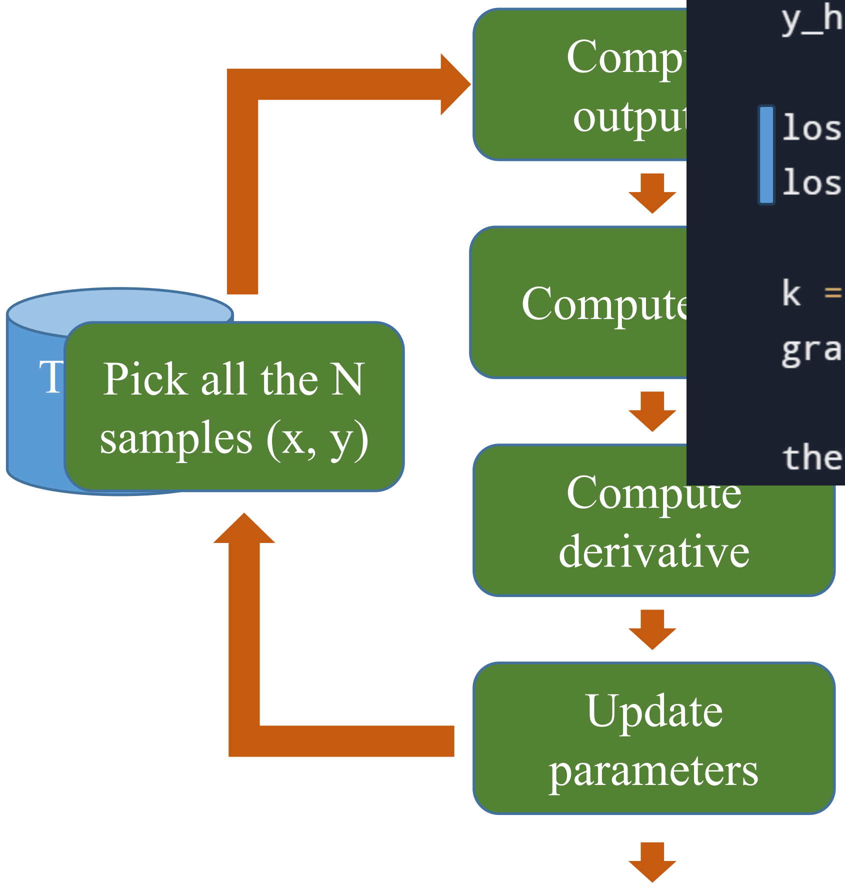
Features													Label
crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	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

Boston House Price Data

Objectives

❖ **Using vector and matrix for notation**

❖ **Using Numpy for implementation**




```python
for epoch in range(epoch_max):
    y_hat = x.dot(theta)

    loss = np.multiply((y_hat-y), (y_hat-y))
    losses.append(np.mean(loss))

    k = 2*(y_hat-y)
    gradients = x.T.dot(k) / N

    theta = theta - lr*gradients
```

1) Pick all the N samples from training data

2) Compute output $\hat{\mathbf{y}}$

$$\hat{\boldsymbol{y}} = X\boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)(\hat{y} - y)^{T} \frac{1}{N}$$

4) Compute derivative

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

$$L'_{\boldsymbol{\theta}} = X^{T}\boldsymbol{k}$$

5) Update parameters

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

Outline

SECTION 1

1-sample Vectorization

SECTION 2

m-sample Vectorization

SECTION 3

N-sample Vectorization

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

$\hat{y} = wx + b$ $\quad \cancel{x} = \begin{bmatrix} 1 \\ x \end{bmatrix}$ $\quad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$

$X \cdot \theta^T$

Linear Regression (1-samples)

Transpose

$$\vec{v} = \begin{bmatrix} v_1 \\ \dots \\ v_n \end{bmatrix} \qquad \vec{v}^T = [v_1 \quad \dots v_n]$$

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}^T \rightarrow \begin{bmatrix} 1 & 2 \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \qquad A^T = \begin{bmatrix} a_{11} & \dots & a_{m1} \\ \dots & \dots & \dots \\ a_{1n} & \dots & a_{mn} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T \rightarrow \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

```python
import numpy as np

# create data
data = np.array([1,2,3])
factor = 2

# broadcasting
result_multiplication = data*factor
```

```
[1 2 3]
[2 4 6]
```

Multiply with a number

$$\alpha\vec{u} = \alpha \begin{bmatrix} u_1 \\ \dots \\ u_n \end{bmatrix} = \begin{bmatrix} \alpha u_1 \\ \dots \\ \alpha u_n \end{bmatrix}$$

data $\quad\quad\quad\quad$ **result**

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} * \boxed{2} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

Linear Regression (1-samples)

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

Traditional

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

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

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

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

$$\hat{y} = wx + b \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} \rightarrow \boldsymbol{\theta}^T = [b \quad w]$$

$$\hat{y} = wx + b1 = [b \quad w] \begin{bmatrix} 1 \\ x \end{bmatrix} = \boldsymbol{\theta}^T \boldsymbol{x}$$

dot product

Linear Regression (1-samples)

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

Traditional

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

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

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

$$\hat{y} = wx + b \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x}$$

$$L(\hat{y}, y) = (\hat{y} - y)^2$$

numbers

What will we do?

Linear Regression (1-samples)

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

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

Traditional

3) Compute loss

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

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

$$\hat{y} = wx + b \qquad \cancel{\boldsymbol{x}} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$$

$$\begin{cases} \dfrac{\partial L}{\partial b} = 2(\hat{y} - y) = 2\times(\hat{y} - y)\times 1 \\ \dfrac{\partial L}{\partial w} = 2x(\hat{y} - y) = 2\times(\hat{y} - y)\times x \end{cases}$$

$$2 \times (\hat{y} - y).X$$

$$\underbrace{\begin{bmatrix} 2\times(\hat{y} - y)\times 1 \\ 2\times(\hat{y} - y)\times x \end{bmatrix}}_{} = 2(\hat{y} - y)\begin{bmatrix} 1 \\ x \end{bmatrix} = 2(\hat{y} - y)\boldsymbol{x} = \begin{bmatrix} \dfrac{\partial L}{\partial b} \\ \dfrac{\partial L}{\partial w} \end{bmatrix} = L'_{\boldsymbol{\theta}} \rightarrow L'_{\boldsymbol{\theta}} = 2\cancel{\boldsymbol{x}}(\hat{y} - y)$$

common factor

Linear Regression (1-samples)

$$\hat{y} = wx + b \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} \qquad L'_{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \end{bmatrix}$$

Traditional

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

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

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

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

$$\begin{cases} b = b - \eta \dfrac{\partial L}{\partial b} \\ w = w - \eta \dfrac{\partial L}{\partial w} \end{cases}$$

$$\boldsymbol{\theta} \qquad \boldsymbol{\theta} \qquad L'_{\boldsymbol{\theta}}$$

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

AI VIET NAM
@aivietnam.edu.vn

Linear Regression (1-samples)

Traditional

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

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

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

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

Vectorized

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

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

$\hat{y} = wx + b$

$\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$

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

1. $\cancel{\boldsymbol{x}} = \begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ 6.7 \end{bmatrix}$

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

$\eta = 0.01$

1) Pick a sample $(\mathbf{x}, y)$ from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

Input $\boldsymbol{x}$

Model

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

2. $\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x}$ = -2.238

Label

$y = 9.1$

Loss

3. $(\hat{y} - y)^2 = 128.5$

update

4. $L'_{\boldsymbol{\theta}} = 2\cancel{x}(\hat{y} - y) = \begin{bmatrix} -22.658 \\ -151.81 \end{bmatrix}$

5. $\boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix} - \eta \begin{bmatrix} -22.658 \\ -151.81 \end{bmatrix} = \begin{bmatrix} 0.2755 \\ 1.1781 \end{bmatrix}$

Linear Regression (1-samples)

❖ Implementation (vectorization using numpy)

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

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

```python
import numpy as np

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

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

    return dtheta

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

    return dtheta_new
```

$\hat{y} = wx + b$

$\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$

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

1 $\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ 3.5 \end{bmatrix}$

Given $\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix}$

$\eta = 0.01$

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

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

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

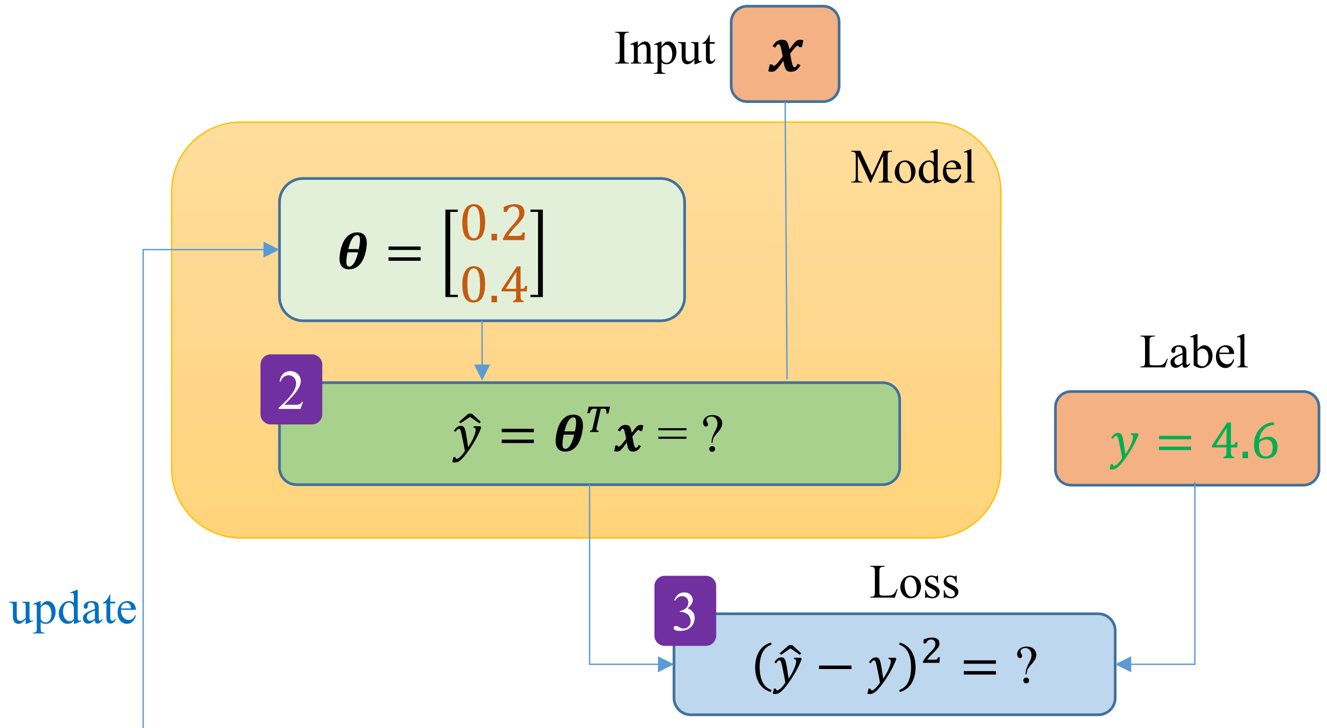
4) Compute derivative

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

5) Update parameters

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

η is learning rate



4 $L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(\hat{y} - y) = \begin{bmatrix} ? \\ ? \end{bmatrix}$

5 $\boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} - \eta \begin{bmatrix} ? \\ ? \end{bmatrix} = \begin{bmatrix} ? \\ ? \end{bmatrix}$

```python
data = np.genfromtxt('data.csv', delimiter=',')
N = 4

areas  = data[:, 0].reshape(N, 1)
prices = data[:, 1].reshape(N,)

# vector [1, area]
features = np.hstack([np.ones((N,1)), areas])
```

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

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

# update weights
def update_weight(theta, lr, dtheta):
    dtheta_new = theta - lr*dtheta
    return dtheta_new
```

```python
lr = 0.01
epoch_max = 10

# [b, w]
theta = np.array([0.049, -0.34])

for epoch in range(epoch_max):
    for i in range(N):
        # get a sample
        x = features[i,:]
        y = prices[i]

        # predict y_hat
        y_hat = predict(x, theta)

        # compute loss
        loss = (y_hat-y)*(y_hat-y)

        # compute gradient
        dtheta = gradient(y_hat, y, x)

        # update weights
        theta = update_weight(theta, lr, dtheta)
```

Advertising Problem

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

if TV=55.0, Radio=34.0, and Newspaper=62.0, price=?

$$\hat{y} = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + b$$

$$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \theta = \begin{bmatrix} b \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

1) Pick a sample $(\mathbf{x}, y)$ from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

Outline

SECTION 1

1-sample Vectorization

SECTION 2

m-sample Vectorization

SECTION 3

N-sample Vectorization

Parameter Initialization

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

Way 1 for constructing matrix **x**

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

Way 2 for constructing matrix **x**

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

Linear Regression

❖ Vectorization

House price data

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

m = 4

Model

$$\text{price} = \text{w} * area + b$$

$$\hat{y} = wx + b$$

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$

Parameter Initialization

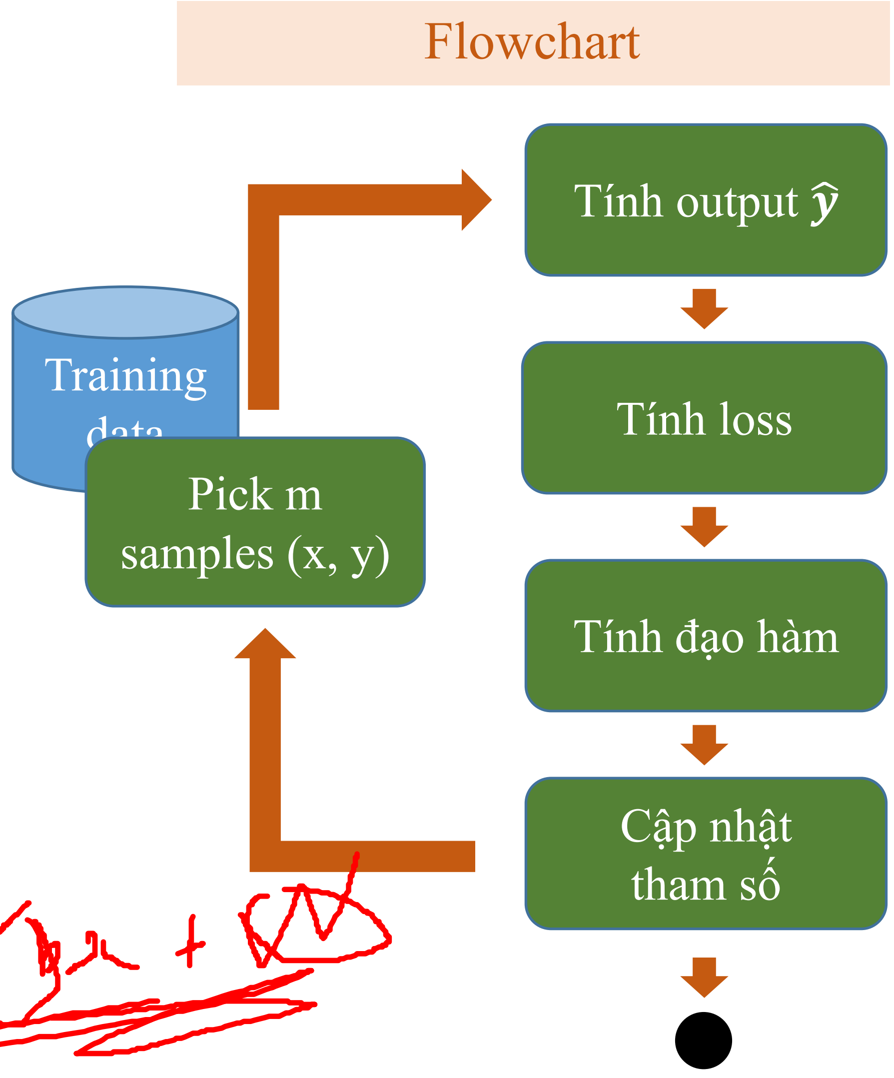
$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} \begin{matrix} b \\ w \end{matrix}$$

Way 1 for constructing matrix x

$$\text{X} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

Way 2 for constructing matrix x

$$\text{X} = \begin{bmatrix} 6.7 & 1 \\ 4.6 & 1 \end{bmatrix}$$

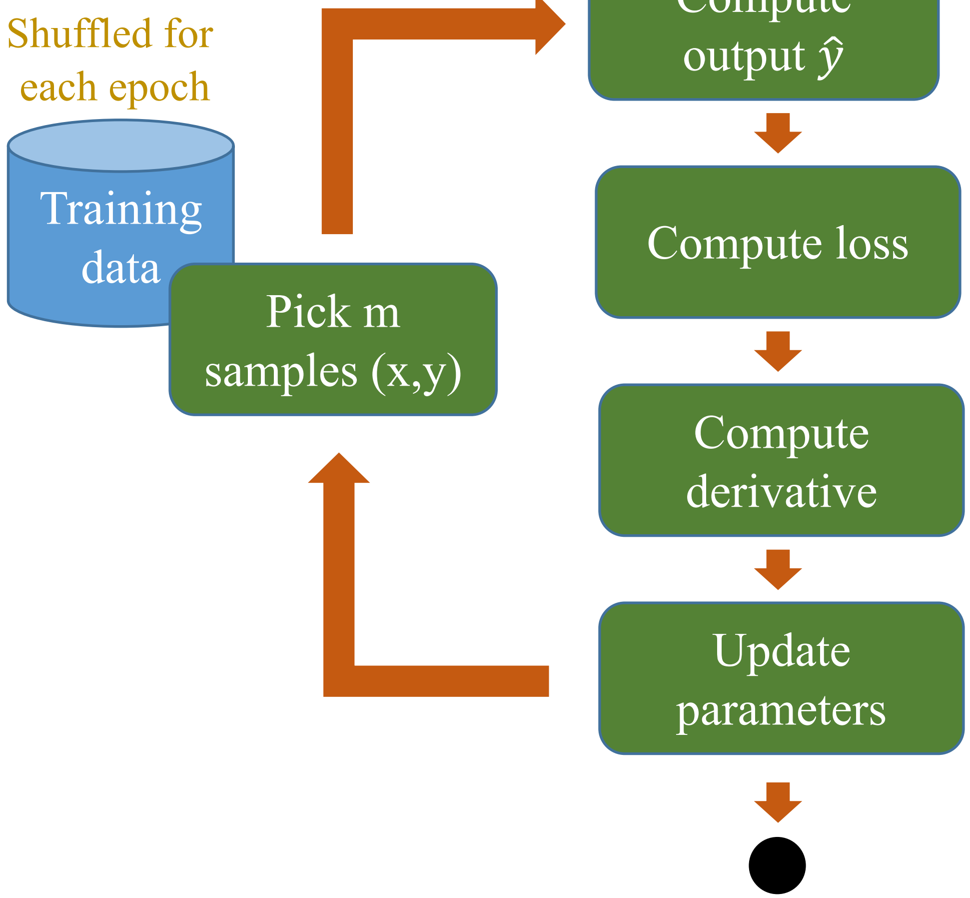


Linear Regression (m-samples)

	Feature	Label
	area	price
	6.7	9.1
	4.6	5.9

❖ House price prediction

❖ m-sample training (1<m<N)



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

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = wx^{(i)} + b \qquad \text{for } 0 \leq i < m$$

3) Compute loss

$$L = \frac{1}{m}\sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

$$L'^{(i)}_w = 2x^{(i)}(\hat{y}^{(i)} - y^{(i)})$$

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

5) Update parameters

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

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

Learning rate η

Linear Regression (m-samples)

Feature	Label
area	price
6.7	9.1
4.6	5.9

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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

$$L = \frac{1}{m}\sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

Learning rate η

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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

$$L = \frac{1}{m}\sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

$$L_{\boldsymbol{\theta}}'^{(i)} = 2\boldsymbol{x}^{(i)}(\hat{y}^{(i)} - y^{(i)}) \text{ for } 0 \le i < m$$

5) Update parameters

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

η is learning rate

Implementation - Differences

Feature	Label
area	price
6.7	9.1
4.6	5.9

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

2) Compute output $\hat{y}$

$$\hat{y} = \underline{\boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}}$$

3) Compute loss

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

$\rightarrow$

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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

Linear Regression (m-samples)

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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

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

# number of epochs
epoch_max = 10

# mini-batch size
m = 2
```

Linear Regression

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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

        # some variables
        sum_of_losses = 0
        gradients = np.zeros((2,))
        for index in range(j, j+m):
            # get mini-batch
            x_i = data[index]
            y_i = prices[index]

            # predict y_hat_i
            y_hat_i = x_i.dot(theta)

            # compute loss
            l_i = (y_hat_i - y_i)*(y_hat_i - y_i)

            # compute gradient
            gradient_i = x_i*2*(y_hat_i - y_i)

            # accumulate gradients
            gradients = gradients + gradient_i
            sum_of_losses = sum_of_losses + l_i

        # normalize
        sum_of_losses = sum_of_losses/2
        gradients     = gradients/2
```

Linear Regression (m-samples)

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

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \le i < m$$

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

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

way 1

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

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

Linear Regression (m-samples)

Way 1.1

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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

$$L = \frac{1}{m}\sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

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

$$\hat{\mathbf{y}} = \boldsymbol{\theta}^T X$$

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

Linear Regression (m-samples)

Way 1.1

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

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \le i < m$$

3) Compute loss

$$L = \frac{1}{m}\sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

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

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T X = [-2.238 \quad -1.524]$$

$$L = \frac{1}{\mathrm{m}}\left[(\hat{y}^{(0)} - y^{(0)})^2 + (\hat{y}^{(1)} - y^{(1)})^2\right]$$

$$= \frac{1}{\mathrm{m}}\left[(\hat{y}^{(0)} - y^{(0)}) \quad (\hat{y}^{(1)} - y^{(1)})\right]\begin{bmatrix} (\hat{y}^{(0)} - y^{(0)}) \\ (\hat{y}^{(1)} - y^{(1)}) \end{bmatrix}$$

$$= \frac{1}{\mathrm{m}}(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

$$= \frac{1}{2}(128.5 + 55.11) = 91.8$$

Linear Regression (m-samples)

Way 1.1

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

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \le i < m$$

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

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

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} \qquad L = \frac{1}{\mathrm{m}}(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X} = [-2.238 \quad -1.524]$$

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

$$\mathrm{X} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

Linear Regression (m-samples)

Way 1.1

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

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \le i < m$$

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} \qquad L = \frac{1}{\mathrm{m}} (\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X} = [-2.238 \quad -1.524]$$

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

$$\mathrm{X} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

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

Take a look and improve

Linear Regression (m-samples)

Way 1.1

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

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \le i < m$$

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

$$\mathrm{X} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

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

$$L = \frac{1}{\mathrm{m}} (\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

$$\hat{y} = \boldsymbol{\theta}^T \mathrm{X} = [-2.238 \quad -1.524]$$

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

$$\mathrm{X} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

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

Take a look and improve

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

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

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

Linear Regression (m-samples)

Way 1.1

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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} \qquad L = \frac{1}{\mathrm{m}}(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

$$\hat{y} = \boldsymbol{\theta}^T \mathrm{X} = [-2.238 \quad -1.524]$$

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

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

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

Take a look and improve

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

$$L_{\boldsymbol{\theta}}' = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \mathrm{X}\right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -151.92 & -68.301 \\ -22.676 & -14.848 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -220.23 \\ -37.524 \end{bmatrix}$$

Linear Regression (m-samples)

Way 1.1

	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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

$$L_{\boldsymbol{\theta}}^{\prime(i)} = 2\boldsymbol{x}^{(i)}(\hat{y}^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < m$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L_{\boldsymbol{\theta}}^{\prime(i)}}{m}$$

η is learning rate

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

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} \qquad L = \frac{1}{\mathrm{m}}(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

$$\hat{y} = \boldsymbol{\theta}^T \mathrm{X} = [-2.238 \quad -1.524]$$

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

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \mathrm{X}\right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -220.23 \\ -37.524 \end{bmatrix}$$

Linear Regression (m-samples)

Way 1.1

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

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \le i < m$$

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} \qquad L = \frac{1}{\mathrm{m}} (\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

$$\hat{y} = \boldsymbol{\theta}^T \mathrm{X} = [-2.238 \quad -1.524]$$

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

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \mathrm{X} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -220.23 \\ -37.524 \end{bmatrix}$$

$\eta = 0.01$

$m = 2$

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

$$= \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} - 0.005 \begin{bmatrix} -220.23 \\ -37.524 \end{bmatrix} = \begin{bmatrix} 0.761 \\ 0.227 \end{bmatrix}$$

Linear Regression (m-samples)

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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

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

4) Compute derivative

$$L_{\boldsymbol{\theta}}^{\prime(i)} = 2\boldsymbol{x}^{(i)}(\hat{y}^{(i)} - y^{(i)}) \text{ for } 0 \leq i < m$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L_{\boldsymbol{\theta}}^{\prime(i)}}{m}$$

η is learning rate

Generalized formula

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

2) Compute output $\hat{y}$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X}$$

3) Compute loss

$$L = \frac{1}{\mathrm{m}}(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

4) Tính đạo hàm

$\odot$ is element-wise multiplication

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

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

5) Cập nhật tham số

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

η is learning rate

More generalized formula

Linear Regression

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

2) Compute output $\hat{y}$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X}$$

3) Compute loss

$$L = \frac{1}{\mathrm{m}}\left(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}}\right)\left(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}}\right)^{\mathrm{T}}$$

4) Compute derivative

$$\boldsymbol{k} = 2\left(\hat{\boldsymbol{y}} - \boldsymbol{y}^T\right)$$

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

$\odot$ is element-wise multiplication

5) Update parameters

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

η is learning rate

```python
import numpy as np
import matplotlib.pyplot as plt

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

# vector [x, b]^T
data = np.vstack([areas, np.ones((N,))])

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

# params
lr = 0.01
epoch_max = 1
m = 2

# logging
losses = []
```

Linear Regression (m-samples)

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

2) Compute output $\hat{y}$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X}$$

3) Compute loss

$$L = \frac{1}{\mathrm{m}}(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

4) Compute derivative

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

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

$\odot$ is element-wise multiplication

5) Update parameters

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

η is learning rate

More generalized formula

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

        # predict y_hat
        y_hat = theta.T.dot(x)

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

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

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

Linear Regression (m-samples)

Way 1.2

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

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \le i < m$$

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

$$L_{\boldsymbol{\theta}}^{\prime(i)} = 2\boldsymbol{x}^{(i)}(\hat{y}^{(i)} - y^{(i)}) \quad \text{for } 0 \le i < m$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L_{\boldsymbol{\theta}}^{\prime(i)}}{m}$$

η is learning rate

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

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} \qquad L = \frac{1}{\mathrm{m}}(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X} = [-2.238 \quad -1.524]$$

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

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

$$\mathrm{X} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

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

Take a look and improve

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

$$L'_{\boldsymbol{\theta}} = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \mathrm{X}\right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -151.92 & -68.301 \\ -22.676 & -14.848 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -220.23 \\ -37.524 \end{bmatrix}$$

Linear Regression (m-samples)

Way 1.2

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

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \le i < m$$

3) Compute loss

$$L = \frac{1}{m} \sum_i (\hat{y}^{(i)} - y^{(i)})^2$$

4) Compute derivative

$$L_{\boldsymbol{\theta}}'^{(i)} = 2\boldsymbol{x}^{(i)}(\hat{y}^{(i)} - y^{(i)}) \quad \text{for } 0 \le i < m$$

5) Update parameters

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

η is learning rate

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

$$\boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} \qquad L = \frac{1}{\mathrm{m}} (\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X} = [-2.238 \quad -1.524]$$

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

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

$$\mathrm{X} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix}$$

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

$$L_{\boldsymbol{\theta}}' = \mathrm{X}\boldsymbol{k}^T$$

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

$$L_{\boldsymbol{\theta}}' = \left(\begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{k} \end{bmatrix} \odot \mathrm{X}\right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -151.92 & -68.301 \\ -22.676 & -14.848 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -220.23 \\ -37.524 \end{bmatrix}$$

Linear Regression

Way 1.2

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

2) Compute output $\hat{y}$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X}$$

3) Compute loss

$$L = \frac{1}{\mathrm{m}} (\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

4) Compute derivative

$$\boxed{\begin{aligned} \boldsymbol{k} &= 2(\hat{\boldsymbol{y}} - \boldsymbol{y}^T) \\ L'_{\boldsymbol{\theta}} &= \mathrm{X}\boldsymbol{k}^T \end{aligned}}$$

5) Update parameters

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

η is learning rate

More generalized formula

```python
import numpy as np
import matplotlib.pyplot as plt

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

# vector [x, b]^T
data = np.vstack([areas, np.ones((N,))])

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

# params
lr = 0.01
epoch_max = 1
m = 2

# logging
losses = []
```

Linear Regression (m-samples)

Way 1.2

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

2) Compute output $\hat{y}$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X}$$

3) Compute loss

$$L = \frac{1}{\mathrm{m}} (\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})(\hat{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$$

4) Compute derivative

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

$$L'_{\boldsymbol{\theta}} = \mathrm{X}\boldsymbol{k}^T$$

5) Update parameters

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

η is learning rate

More generalized formula

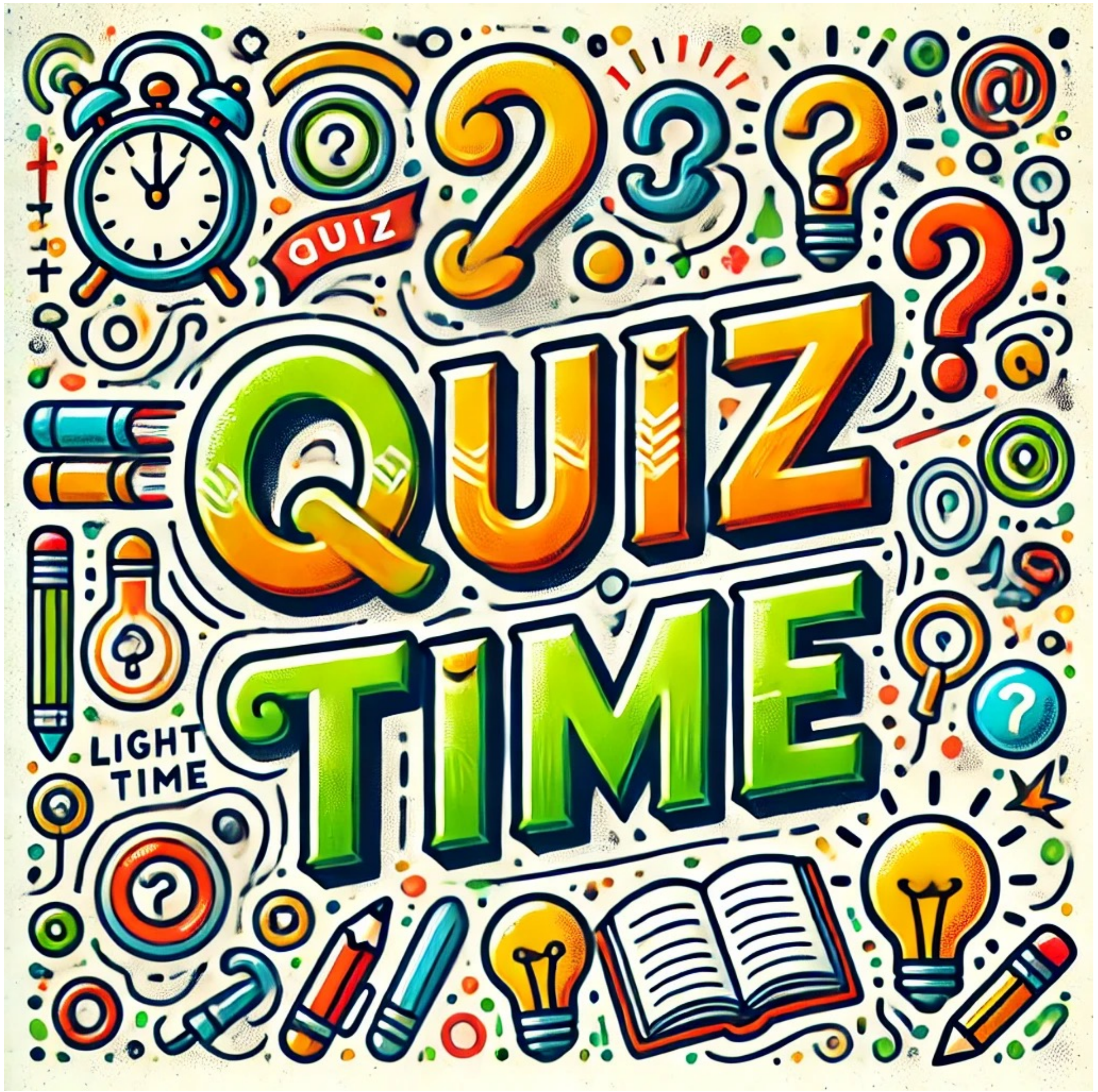
```python
for epoch in range(epoch_max):
    for i in range(0, N, m):
        # get m samples
        x = data[:, i:i+m]
        y = prices[i:i+m, :]

        # predict y_hat
        y_hat = theta.T.dot(x)

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

        # compute gradient
        k = 2*(y_hat-y.T)
        gradients = x.dot(k.T) / m

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


QUIZ
QUIZ TIME
LIGHT TIME

Linear Regression (m-samples)

❖ What about this arrangement?

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

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

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

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

$$\hat{y} = X\boldsymbol{\theta} = \begin{bmatrix} -2.238 \\ -1.524 \end{bmatrix}$$

$$L = \frac{1}{\text{m}}(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\text{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y}) = \frac{1}{2}(128.5 + 55.11) = 91.8$$

$$\boldsymbol{k} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y}) = \begin{bmatrix} -22.676 \\ -14.848 \end{bmatrix}$$

$$L'_{\boldsymbol{\theta}} = \boldsymbol{k}^T X = \begin{bmatrix} -22.676 & -14.848 \end{bmatrix} \begin{bmatrix} 6.7 & 1 \\ 4.6 & 1 \end{bmatrix} = \begin{bmatrix} -220.23 & -37.524 \end{bmatrix}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \left(\frac{L'_{\boldsymbol{\theta}}}{m} \right)^T$$

$$= \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} - 0.005 \begin{bmatrix} -220.23 \\ -37.524 \end{bmatrix} = \begin{bmatrix} 0.761 \\ 0.227 \end{bmatrix}$$

Linear Regression (m-samples)

❖ What about this arrangement?

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

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

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

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

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

2) Compute output $\hat{y}$

$$\hat{\boldsymbol{y}} = X\boldsymbol{\theta}$$

3) Compute loss

$$L = \frac{1}{m}(\hat{\boldsymbol{y}} - \boldsymbol{y})^T(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

4) Compute derivative

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

$$L'_{\boldsymbol{\theta}} = \boldsymbol{k}^T X$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \left(\frac{L'_{\boldsymbol{\theta}}}{m}\right)^T$$

η is learning rate

Linear Regression (m-samples)

❖ What about this arrangement?

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

$$\mathrm{X} = \begin{bmatrix} 6.7 & 1 \\ 4.6 & 1 \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

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

$$\hat{y} = \mathrm{X}\boldsymbol{\theta} = \begin{bmatrix} -2.238 \\ -1.524 \end{bmatrix}$$

$$L = \frac{1}{\mathrm{m}}(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

$$\boldsymbol{k} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y}) = \begin{bmatrix} -22.676 \\ -14.848 \end{bmatrix}$$

$$L'_{\boldsymbol{\theta}} = \mathrm{X}^T\boldsymbol{k} = \begin{bmatrix} 6.7 & 4.6 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -22.676 \\ -14.848 \end{bmatrix} = \begin{bmatrix} -220.23 \\ -37.524 \end{bmatrix}$$

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

$$= \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} - 0.005 \begin{bmatrix} -220.23 \\ -37.524 \end{bmatrix} = \begin{bmatrix} 0.761 \\ 0.227 \end{bmatrix}$$

Linear Regression (m-samples)

❖ What about this arrangement?

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

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

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

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

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

2) Compute output $\hat{y}$

$$\hat{\boldsymbol{y}} = X\boldsymbol{\theta}$$

3) Compute loss

$$L = \frac{1}{m}(\hat{\boldsymbol{y}} - \boldsymbol{y})^T(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

4) Compute derivative

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

$$L'_{\boldsymbol{\theta}} = X^T\boldsymbol{k}$$

5) Update parameters

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

η is learning rate

Outline

SECTION 1

1-sample Vectorization

SECTION 2

m-sample Vectorization

SECTION 3

N-sample Vectorization

Parameter Initialization

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

Way 1 for constructing matrix **x**

$$X = \begin{bmatrix} 6.7 & 4.6 & 3.5 & 5.5 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

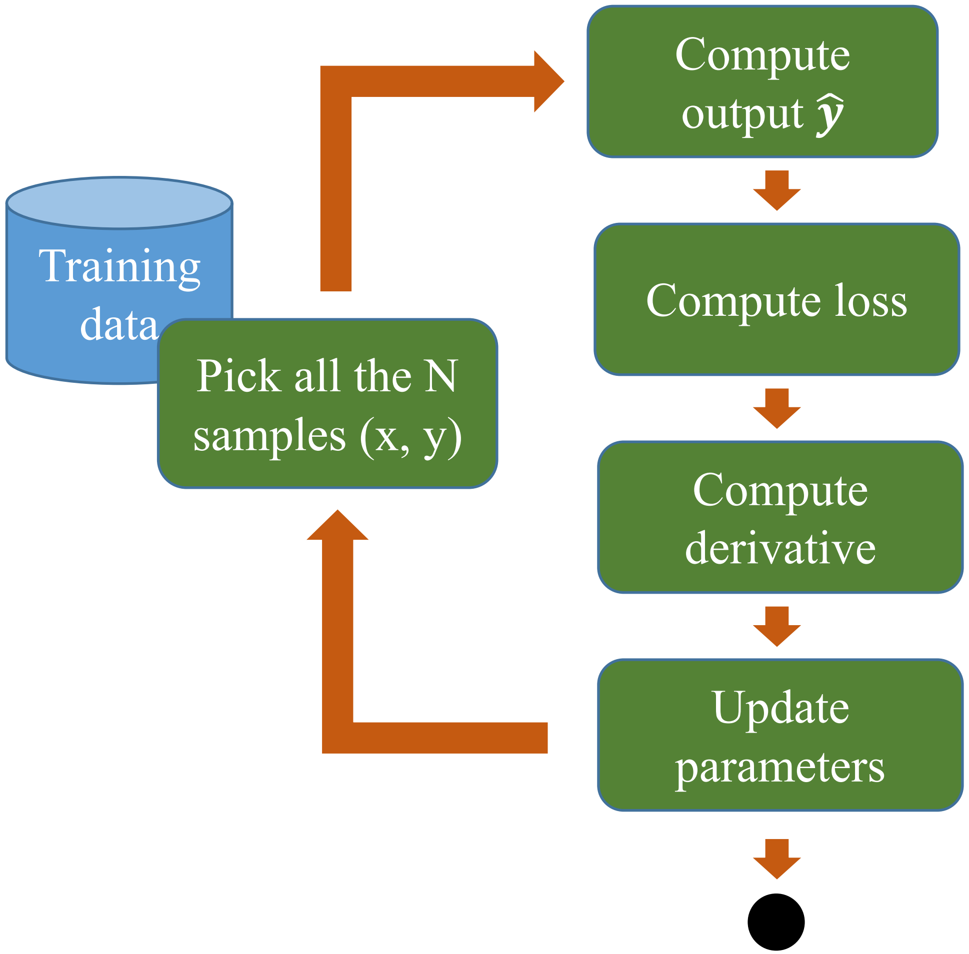
Way 2 for constructing matrix **x**

$$X = \begin{bmatrix} 6.7 & 1 \\ 4.6 & 1 \\ 3.5 & 1 \\ 5.5 & 1 \end{bmatrix}$$

Linear Regression (m-samples)

❖ House price prediction

❖ N-sample training



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

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

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

4) Compute derivative

$$L_w'^{(i)} = 2x^{(i)}(\hat{y}^{(i)} - y^{(i)})$$
$$L_b'^{(i)} = 2(\hat{y}^{(i)} - y^{(i)}) \qquad \text{for } 0 \leq i < N$$

5) Update parameters

$$w = w - \eta \frac{\sum_i L_w'^{(i)}}{N}$$
$$b = b - \eta \frac{\sum_i L_b'^{(i)}}{N} \qquad \text{Learning rate } \eta$$

Linear Regression (N-samples)

1) Pick all the N samples from training data

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

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

4) Compute derivative

$$L_w'^{(i)} = 2x^{(i)}(\hat{y}^{(i)} - y^{(i)})$$
$$L_b'^{(i)} = 2(\hat{y}^{(i)} - y^{(i)}) \qquad \text{for } 0 \leq i < N$$

5) Update parameters

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

$$b = b - \eta \frac{\sum_i L_b'^{(i)}}{N} \qquad \eta \text{ is learning rate}$$

Friendly version

1) Pick all the N samples from training data

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

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

4) Compute derivative

$$L_{\boldsymbol{\theta}}'^{(i)} = 2\boldsymbol{x}^{(i)}(\hat{y}^{(i)} - y^{(i)}) \text{ for } 0 \leq i < N$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L_{\boldsymbol{\theta}}'^{(i)}}{N} \qquad \eta \text{ is learning rate}$$

Generalized formula

Linear Regression (N-samples)

1) Pick all the N samples from training data

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

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

4) Compute derivative

$$L_{\boldsymbol{\theta}}^{\prime(i)} = 2\boldsymbol{x}^{(i)}(\hat{y}^{(i)} - y^{(i)}) \text{ for } 0 \leq i < N$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L_{\boldsymbol{\theta}}^{\prime(i)}}{N} \qquad \eta \text{ is learning rate}$$

Generalized formula

```
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))]

n_epochs = 10
lr = 0.01

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

Linear Regression

1) Pick all the N samples from training data

2) Compute output $\hat{y}^{(i)}$

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

3) Compute loss

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

4) Compute derivative

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

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L'^{(i)}_{\boldsymbol{\theta}}}{N} \qquad \eta \text{ is learning rate}$$

Generalized formula

```
losses = [] # for debug
for epoch in range(n_epochs):
    sum_of_losses = 0
    gradients = np.zeros((2,1))

    for index in range(data_size):
        # get data
        x_i = data[index:index+1]
        y_i = prices[index:index+1]

        # compute output y_hat_i
        y_hat_i = x_i.dot(theta)

        # compute loss
        l_i = (y_hat_i - y_i)*(y_hat_i - y_i)

        # compute gradient
        g_l_i = 2*(y_hat_i - y_i)
        gradient = x_i.T.dot(g_l_i)

        # accumulate gradient
        gradients = gradients + gradient
        sum_of_losses = sum_of_losses + l_i

    # normalize
    sum_of_losses = sum_of_losses/data_size
    gradients     = gradients/data_size

    # for debug
    losses.append(sum_of_losses[0][0])

    # update
    theta = theta - lr*gradients
```

Linear Regression

❖ Vectorization

House price data

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

Model

$$\text{price} = \text{w} * area + b$$

$$\hat{y} = wx + b$$

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$

Parameter Initialization

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

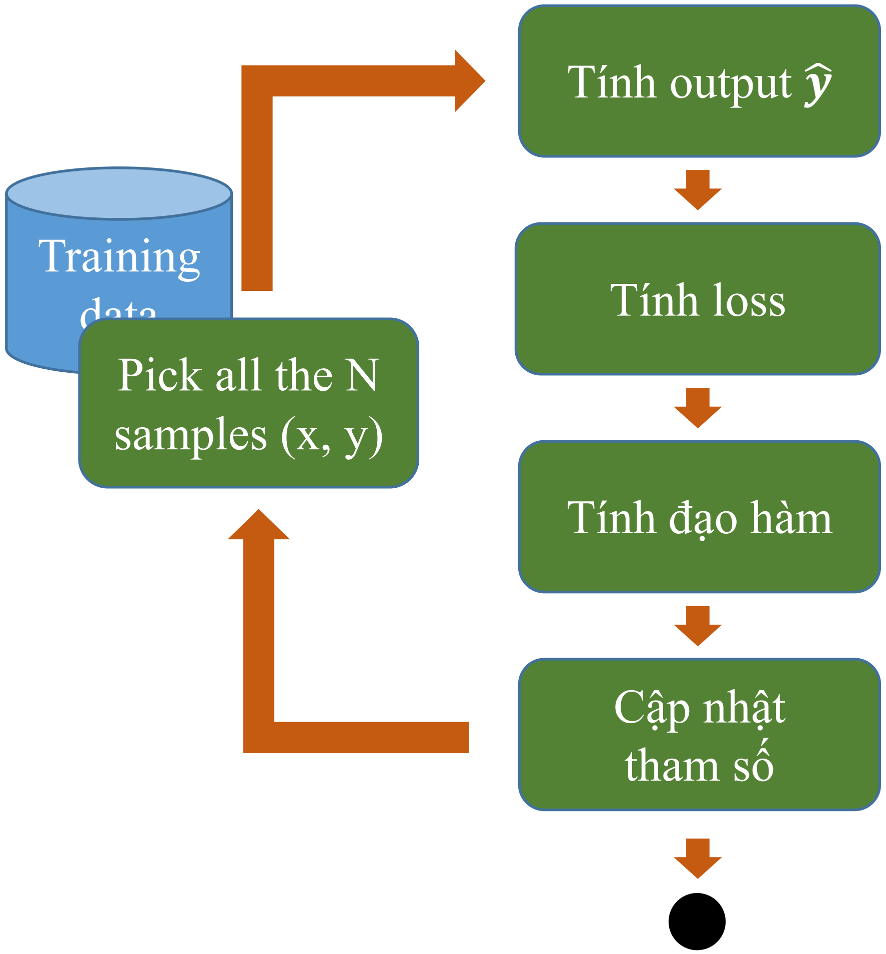
Way 1 for constructing matrix **x**

$$\text{X} = \begin{bmatrix} 6.7 & 4.6 & 3.5 & 5.5 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Way 2 for constructing matrix **x**

$$\text{X} = \begin{bmatrix} 6.7 & 1 \\ 4.6 & 1 \\ 3.5 & 1 \\ 5.5 & 1 \end{bmatrix}$$

Flowchart



Linear Regression [1]

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

House Price Data

Model

$$\text{price} = \mathrm{w} * area + b$$

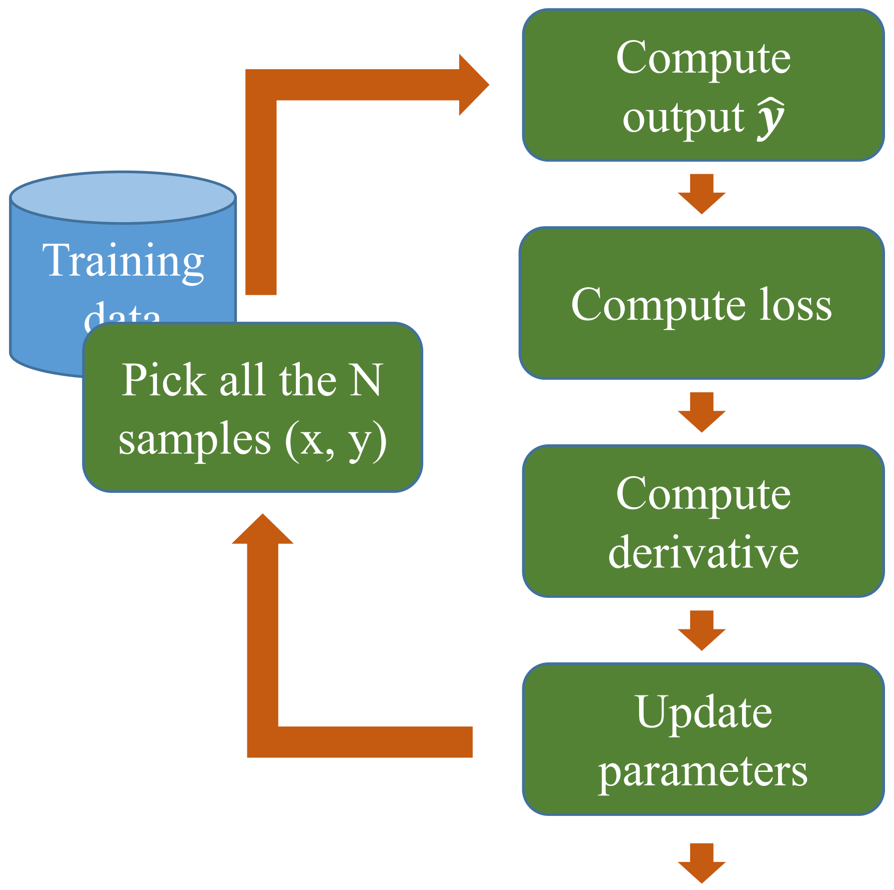
$$\hat{y} = wx + b$$

Parameter Initialization

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

Way 1

$$\mathrm{X} = \begin{bmatrix} 6.7 & 4.6 & 3.5 & 5.5 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$



1) Pick all the N samples from training data

2) Compute output $\hat{\boldsymbol{y}}$

$$\hat{\boldsymbol{y}} = \boldsymbol{\theta}^T \mathrm{X}$$

3) Compute loss

$$L = \frac{1}{N}(\hat{\boldsymbol{y}} - \boldsymbol{y})(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}$$

4) Compute derivative

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

$$L'_{\boldsymbol{\theta}} = \mathrm{X}\boldsymbol{k}^T$$

5) Update parameters

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

η is learning rate

Vectorization Approach

Linear Regression [2]

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

House Price Data

Way 2

$$X = \begin{bmatrix} 6.7 & 1 \\ 4.6 & 1 \\ 3.5 & 1 \\ 5.5 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$

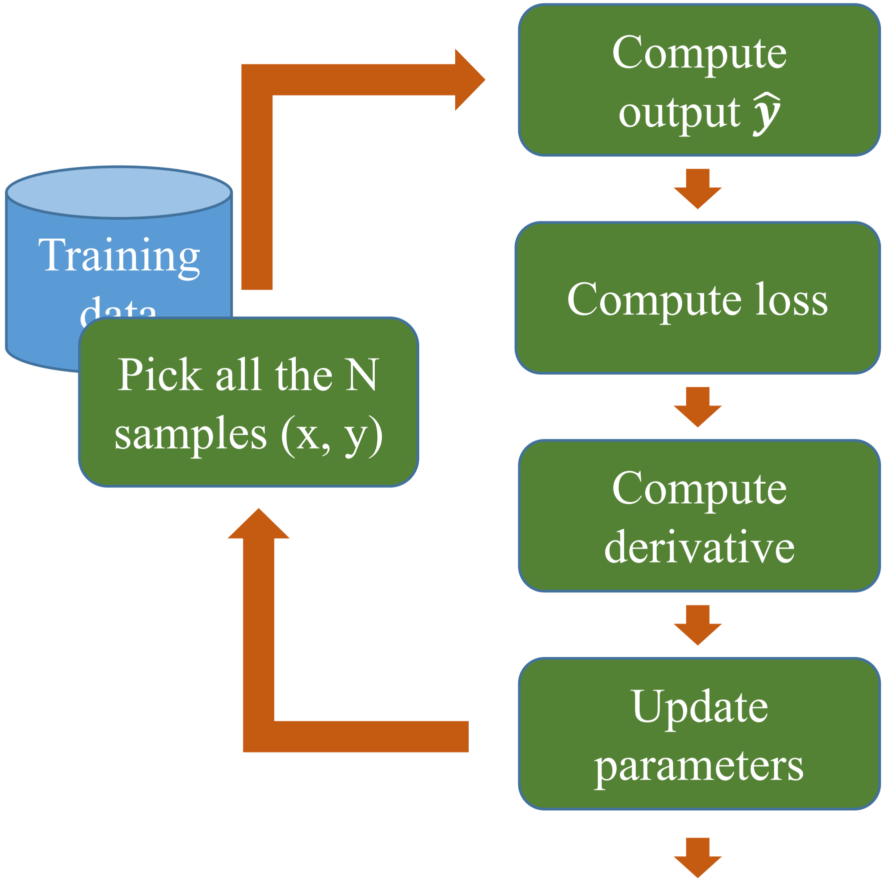
Model

$$\text{price} = \text{w} * area + b$$

$$\hat{y} = wx + b$$

Parameter Initialization

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



1) Pick all the N samples from training data

2) Compute output $\hat{\boldsymbol{y}}$

$$\hat{\boldsymbol{y}} = X\boldsymbol{\theta}$$

3) Compute loss

$$L = \frac{1}{N}(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

4) Compute derivative

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

$$L'_{\boldsymbol{\theta}} = \boldsymbol{k}^T X$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \left(\frac{L'_{\boldsymbol{\theta}}}{N}\right)^T$$

η is learning rate

Vectorization Approach

Linear Regression [3]

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

House Price Data

Way 2

$$X = \begin{bmatrix} 6.7 & 1 \\ 4.6 & 1 \\ 3.5 & 1 \\ 5.5 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$

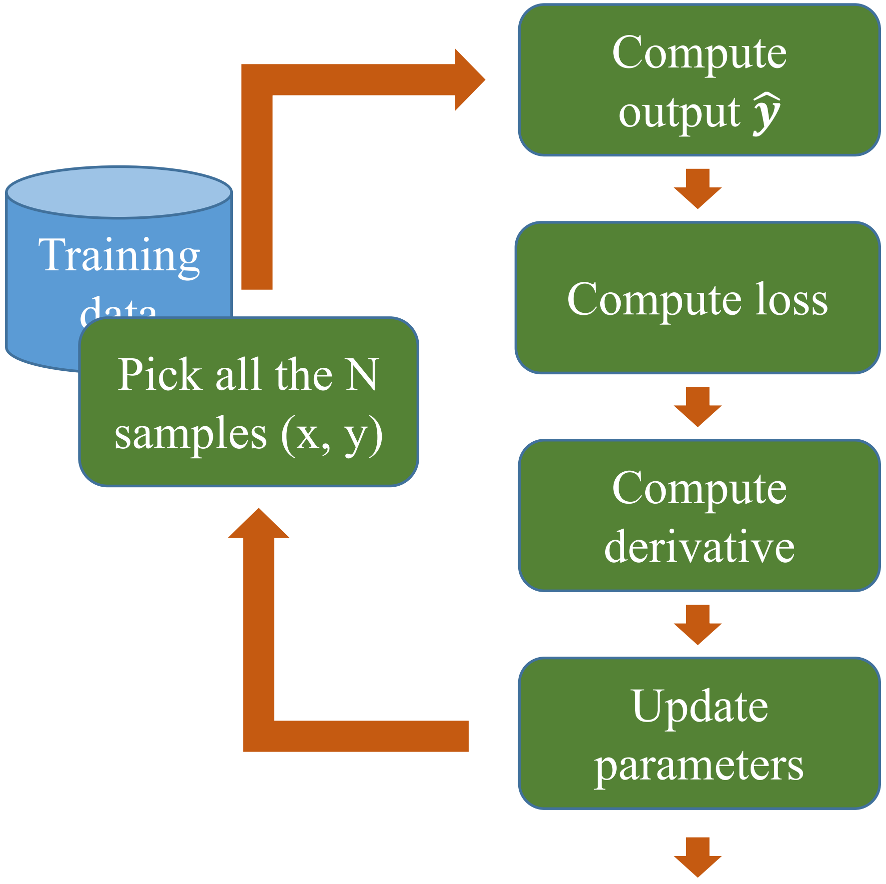
Model

$$\text{price} = \text{w} * area + b$$

$$\hat{y} = wx + b$$

Parameter Initialization

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



1) Pick all the N samples from training data

2) Compute output $\hat{\boldsymbol{y}}$

$$\hat{\boldsymbol{y}} = X\boldsymbol{\theta}$$

3) Compute loss

$$L = \frac{1}{N}(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

4) Compute derivative

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

$$L'_{\boldsymbol{\theta}} = X^T\boldsymbol{k}$$

5) Update parameters

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

η is learning rate

Vectorization Approach

Summary

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

House Price Data

Way 3

$$X = \begin{bmatrix} 6.7 & 1 \\ 4.6 & 1 \\ 3.5 & 1 \\ 5.5 & 1 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$

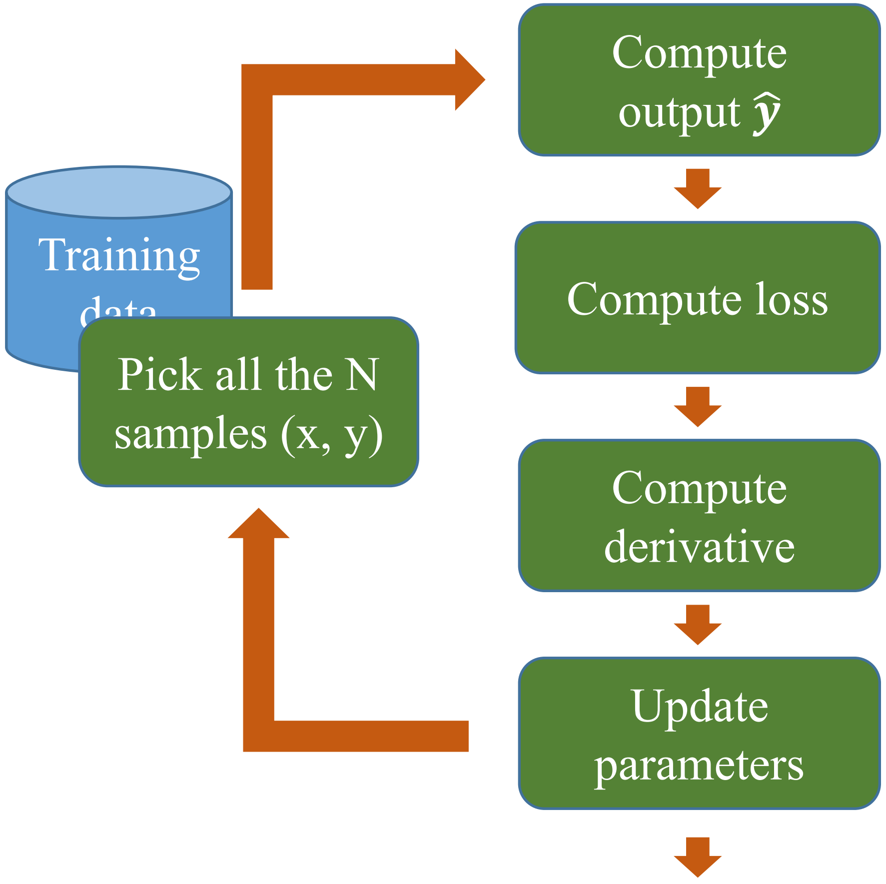
Model

$$\text{price} = \text{w} * area + b$$

$$\hat{y} = wx + b$$

Parameter Initialization

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



1) Pick all the N samples from training data

2) Compute output $\hat{\boldsymbol{y}}$

$$\hat{\boldsymbol{y}} = X\boldsymbol{\theta}$$

3) Compute loss

$$L = \frac{1}{N}(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

4) Compute derivative

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

$$L'_{\boldsymbol{\theta}} = X^T\boldsymbol{k}$$

5) Update parameters

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

η is learning rate

```
for epoch in range(epoch_max):
    y_hat = x.dot(theta)

    loss = np.multiply((y_hat-y), (y_hat-y))
    losses.append(np.mean(loss))

    k = 2*(y_hat-y)
    gradients = x.T.dot(k) / N

    theta = theta - lr*gradients
```