AI VIETNAM
All-in-One Course
(TA Session)

Vectorized Linear Regression

Extra

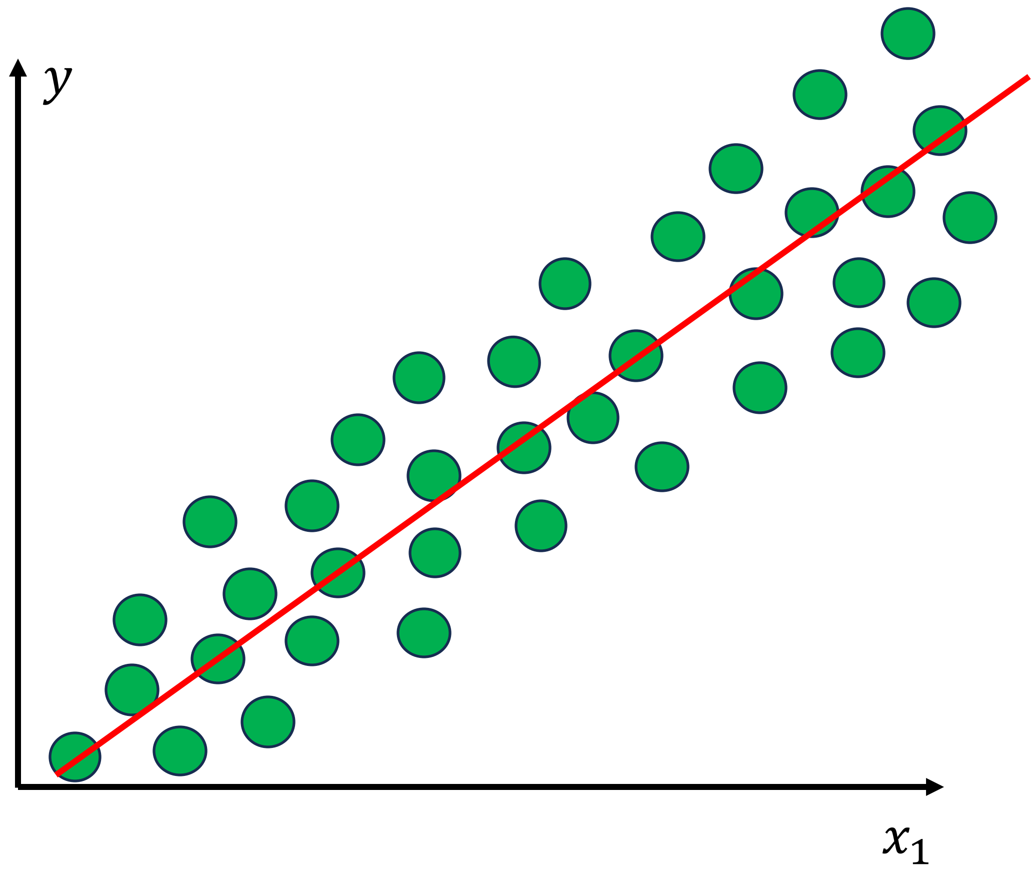
AI VIET NAM
@aivietnam.edu.vn

Dinh-Thang Duong – TA

Yen-Linh Vu – STA

Getting Started

❖ Objectives



Our objectives:

- Revise the simple Linear Regression.
- Discuss a limitation of Simple Linear Regression implementation.
- Introduction to Vectorized Linear Regression.
- Implement the vectorized version of Linear Regression using Gradient Descent.

AI VIETNAM
All-in-One Course
(TA Session)

Outline

- Introduction
- Vectorized Linear Regression
- Code Implementation
- Question

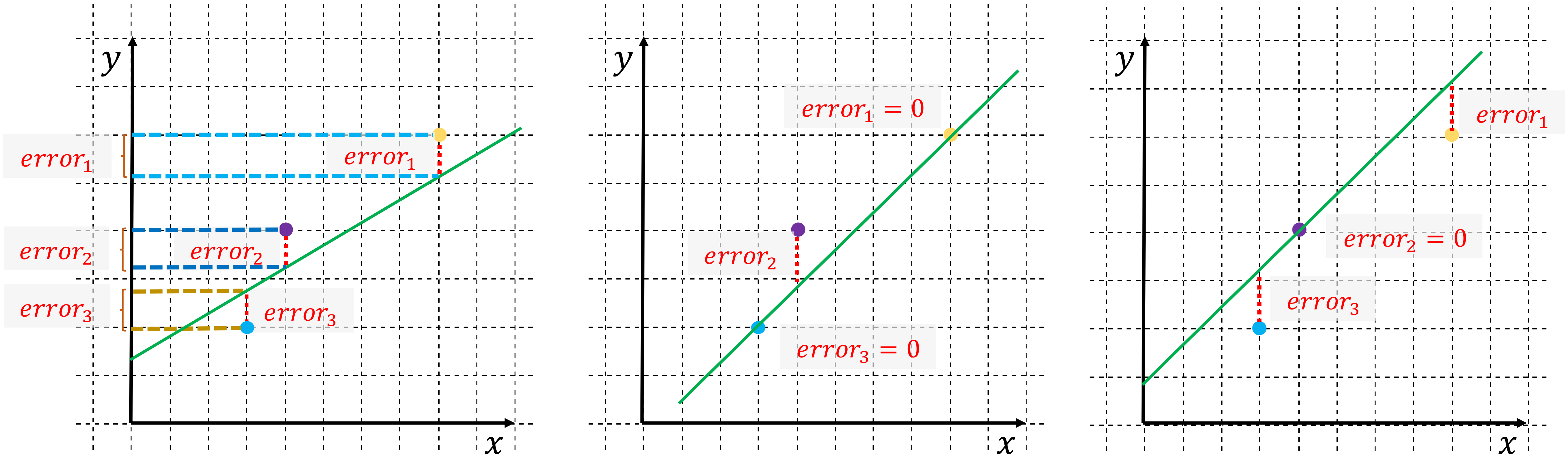
Introduction

Introduction

❖ Recap: Simple Linear Regression

Training data

$$error_i = distance(\hat{y}_i, y_i)$$



$\hat{y} = w_1 x + b_1$ $\hat{y} = w_2 x + b_2$ $\hat{y} = w_3 x + b_3$

Find w and b whose model has the smallest error, where $loss = \sum_i error_i$

How?

Introduction

❖ Recap: Simple Linear Regression

Linear equation

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

where $\hat{y}$ is a predicted value,

w and b are parameters

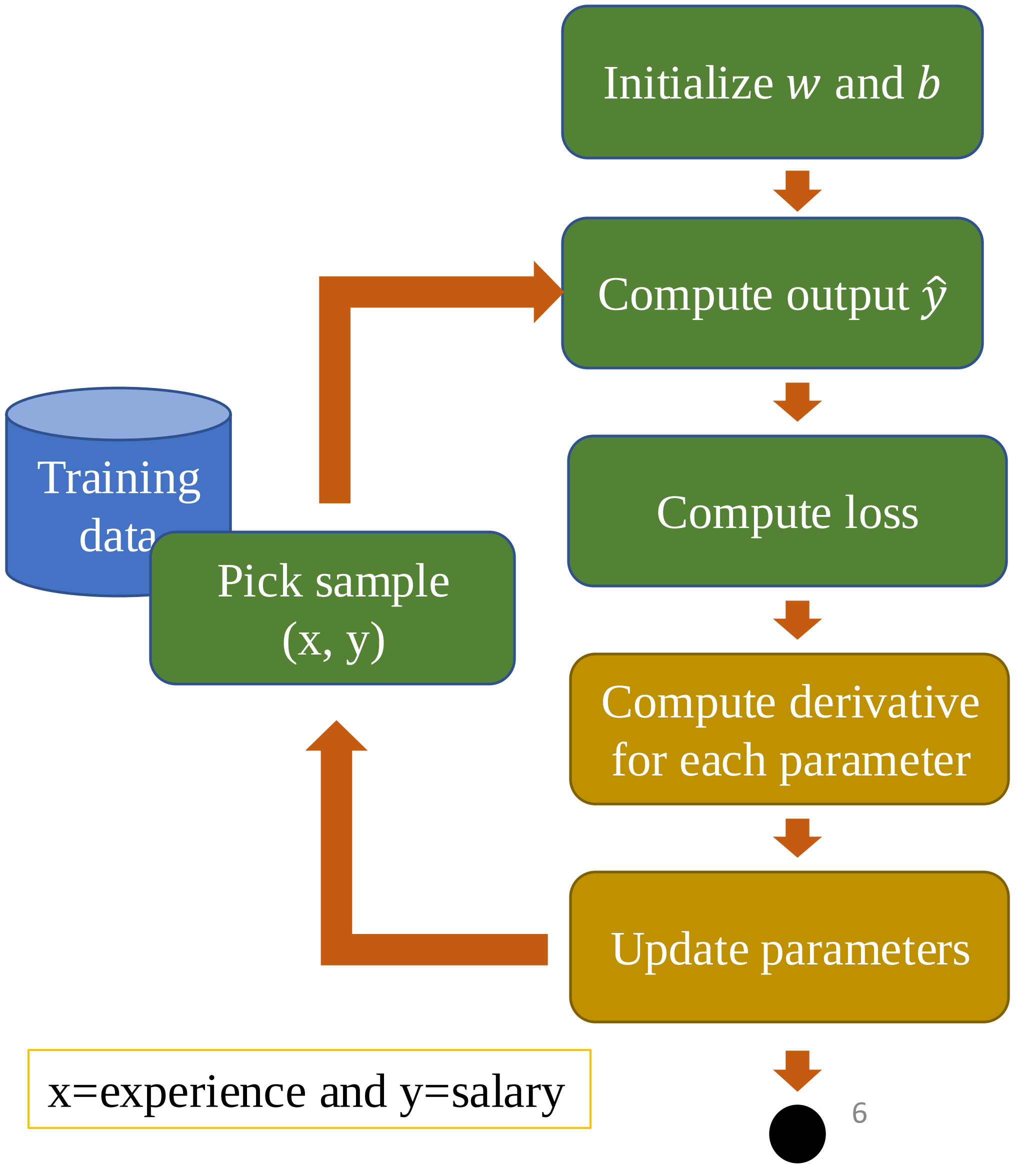
and x is input feature

Error (loss) computation

Idea: compare predicted values $\hat{y}$ and label values y

Squared loss

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



x=experience and y=salary

Introduction

❖ Recap: Simple Linear Regression

Linear equation

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

where $\hat{y}$ is a predicted value,

w and b are parameters

and x is input feature

Error (loss) computation

Idea: compare predicted values $\hat{y}$ and label values y

Squared loss

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

Find better w and b

Use gradient descent to minimize the loss function

Compute derivate for each parameter

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

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

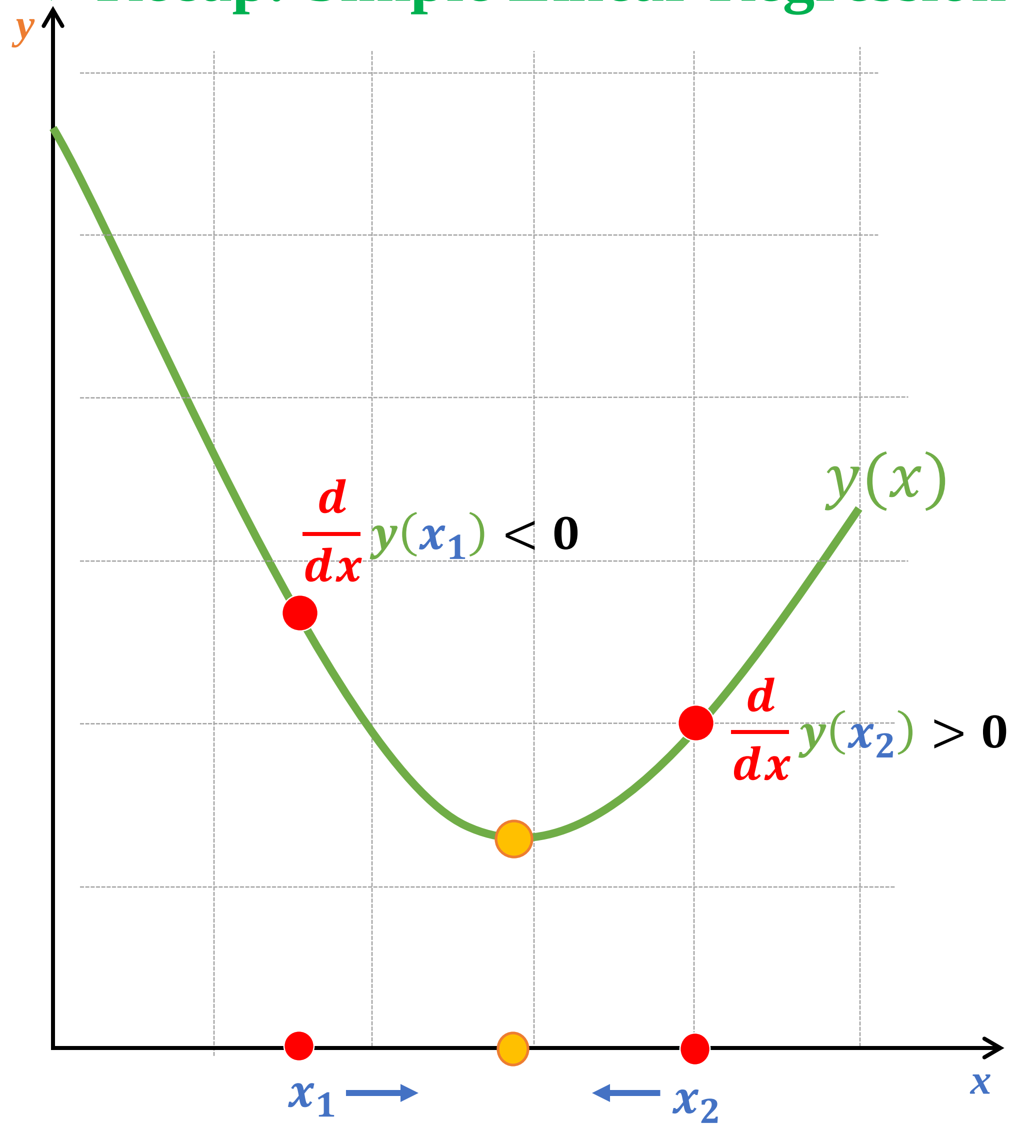
Update parameters

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

η is learning rate

Introduction

❖ Recap: Simple Linear Regression



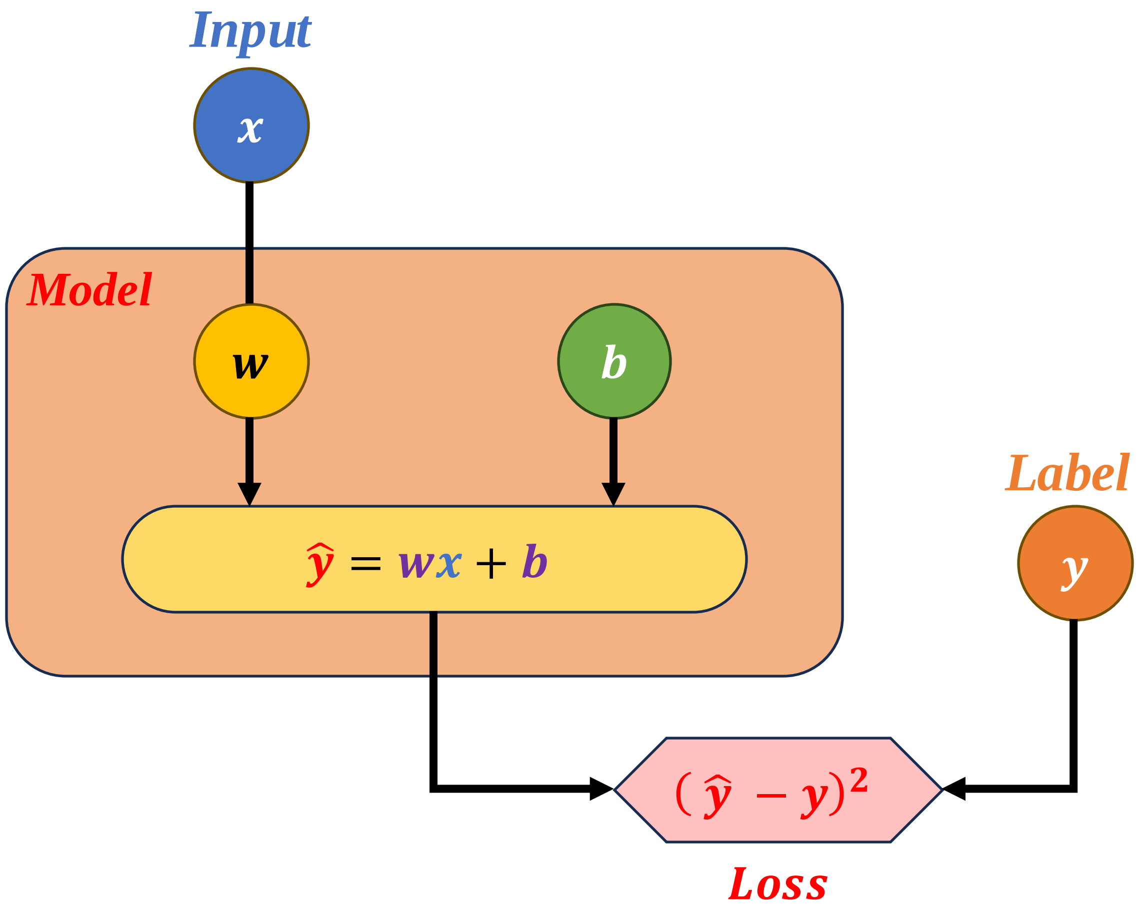
$$\frac{d}{dx}y(x) = \lim_{\Delta x \to 0} \frac{y(x+\Delta x) - y(x)}{\Delta x}$$

$$x_{new} = x_{old} - \eta \frac{d}{dx}y(x_{old})$$

learning rate

Introduction

❖ Recap: Simple Linear Regression



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

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

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w}$$

$$b_{new} = b_{old} - \eta \frac{\partial L}{\partial b}$$

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

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

Introduction

❖ Recap: Simple Linear Regression

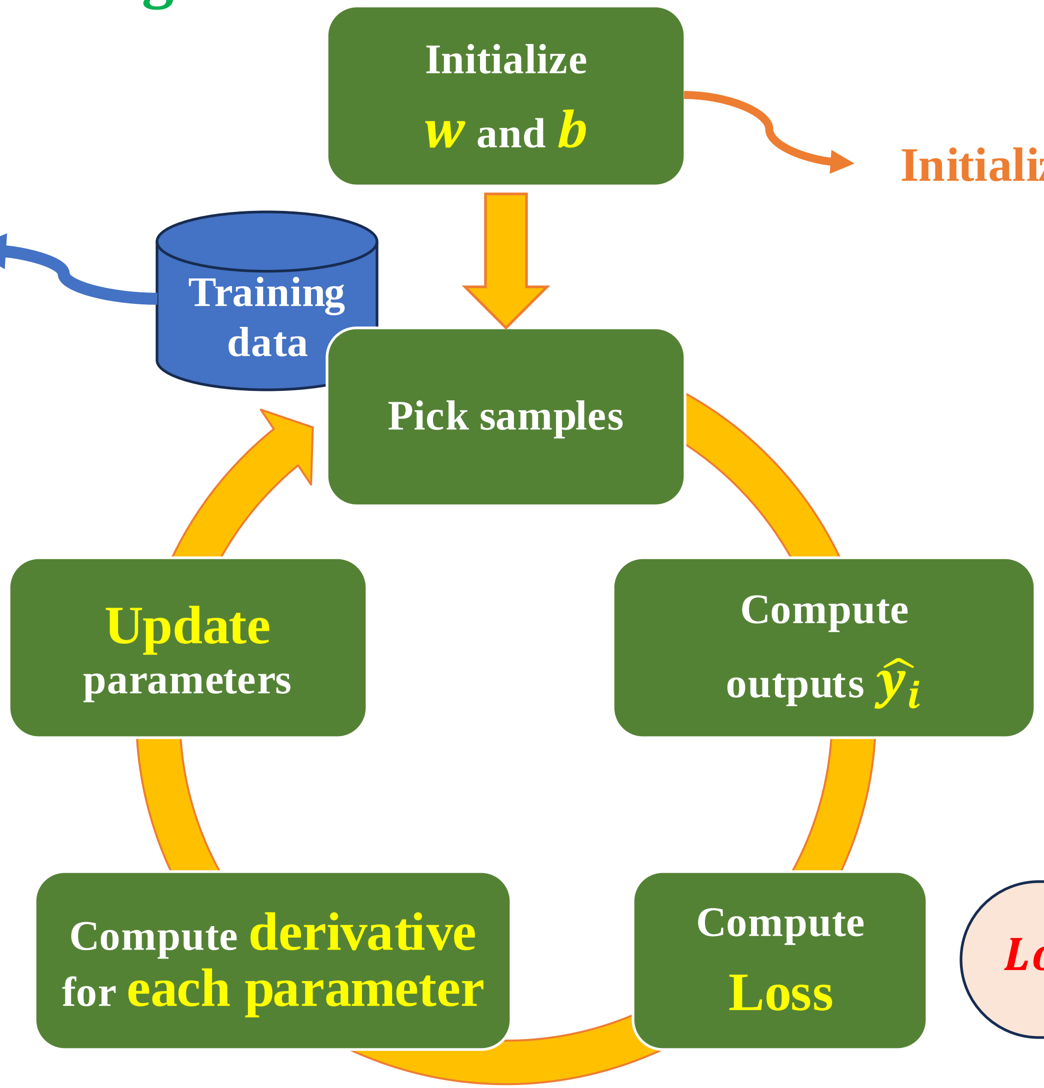
	Experience	Salary
1	3	60
2	4	55
3	5	66
4	6	93

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w}$$

$$b_{new} = b_{old} - \eta \frac{\partial L}{\partial b}$$

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

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



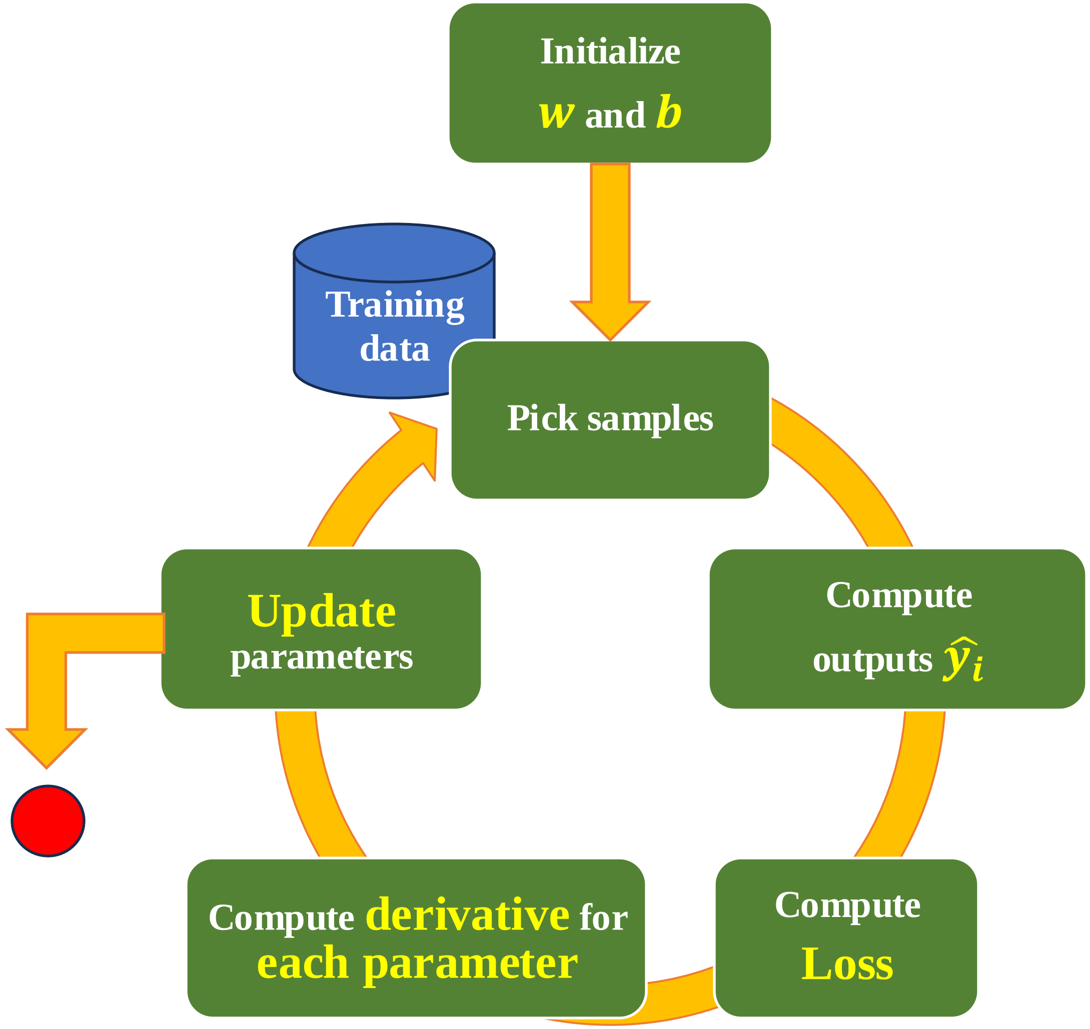
Initialize random parameters

$$\hat{y}_i = wx_i + b$$

$$Loss = \sum_i (\hat{y}_i - y_i)^2 = \sum_i error_i$$

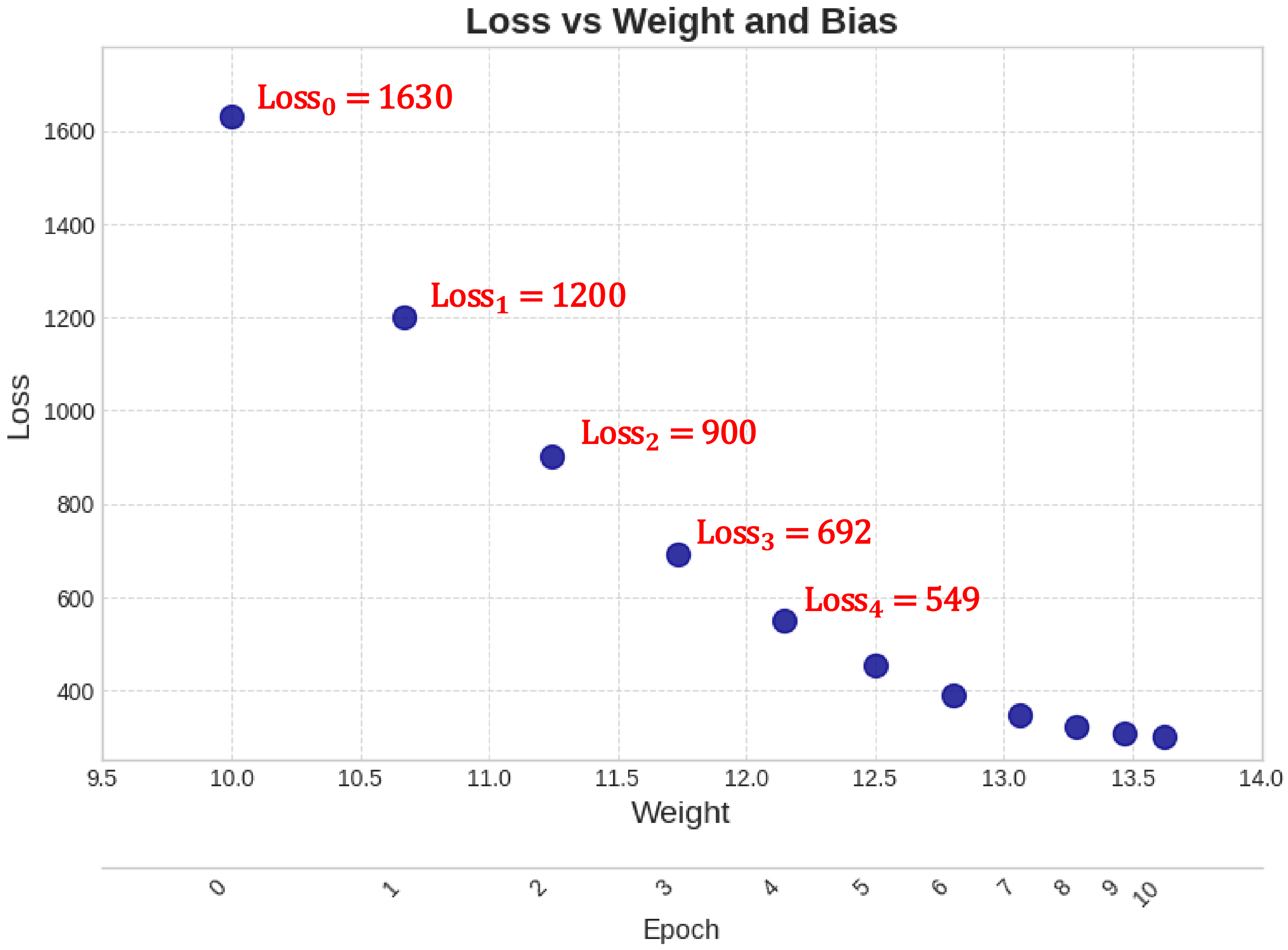
Introduction

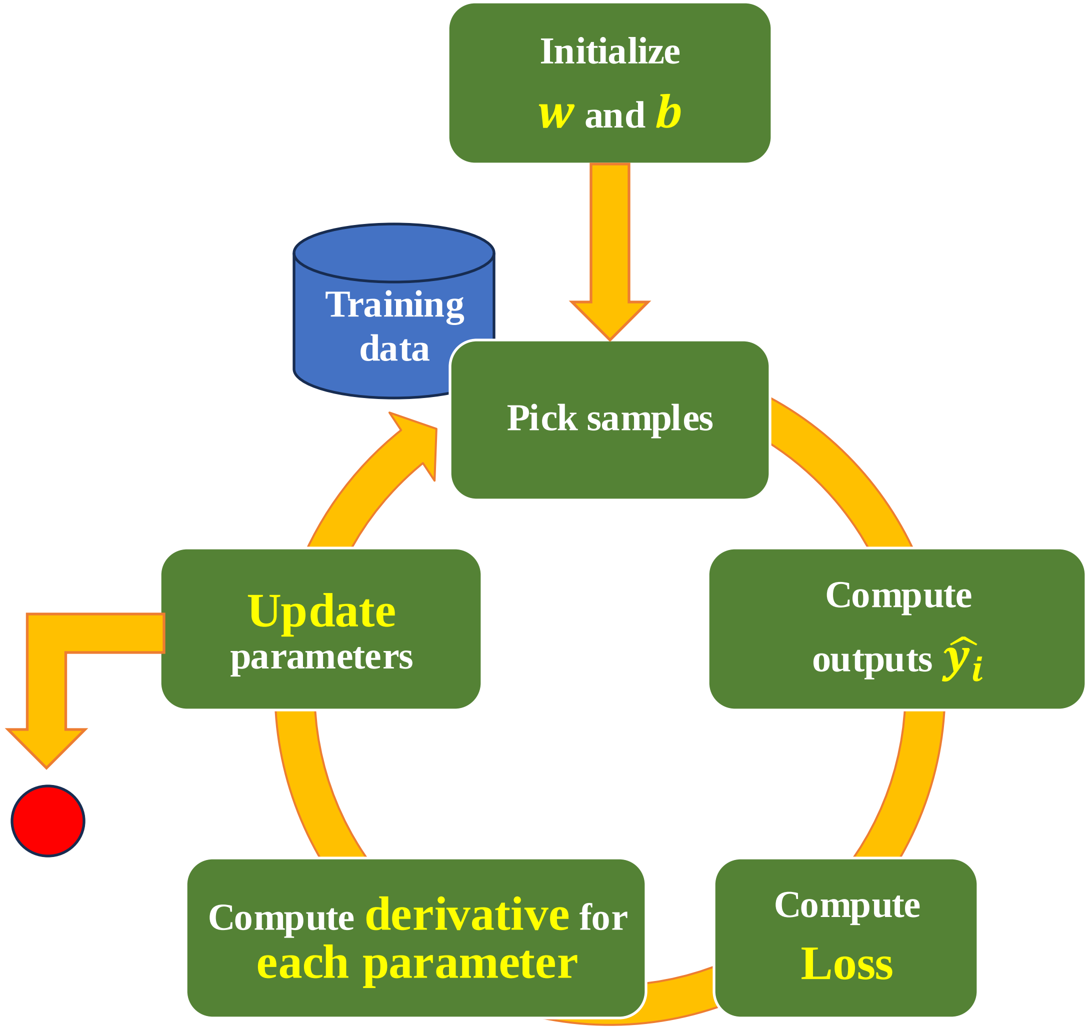
❖ Recap: Simple Linear Regression



Introduction

❖ Recap: Simple Linear Regression

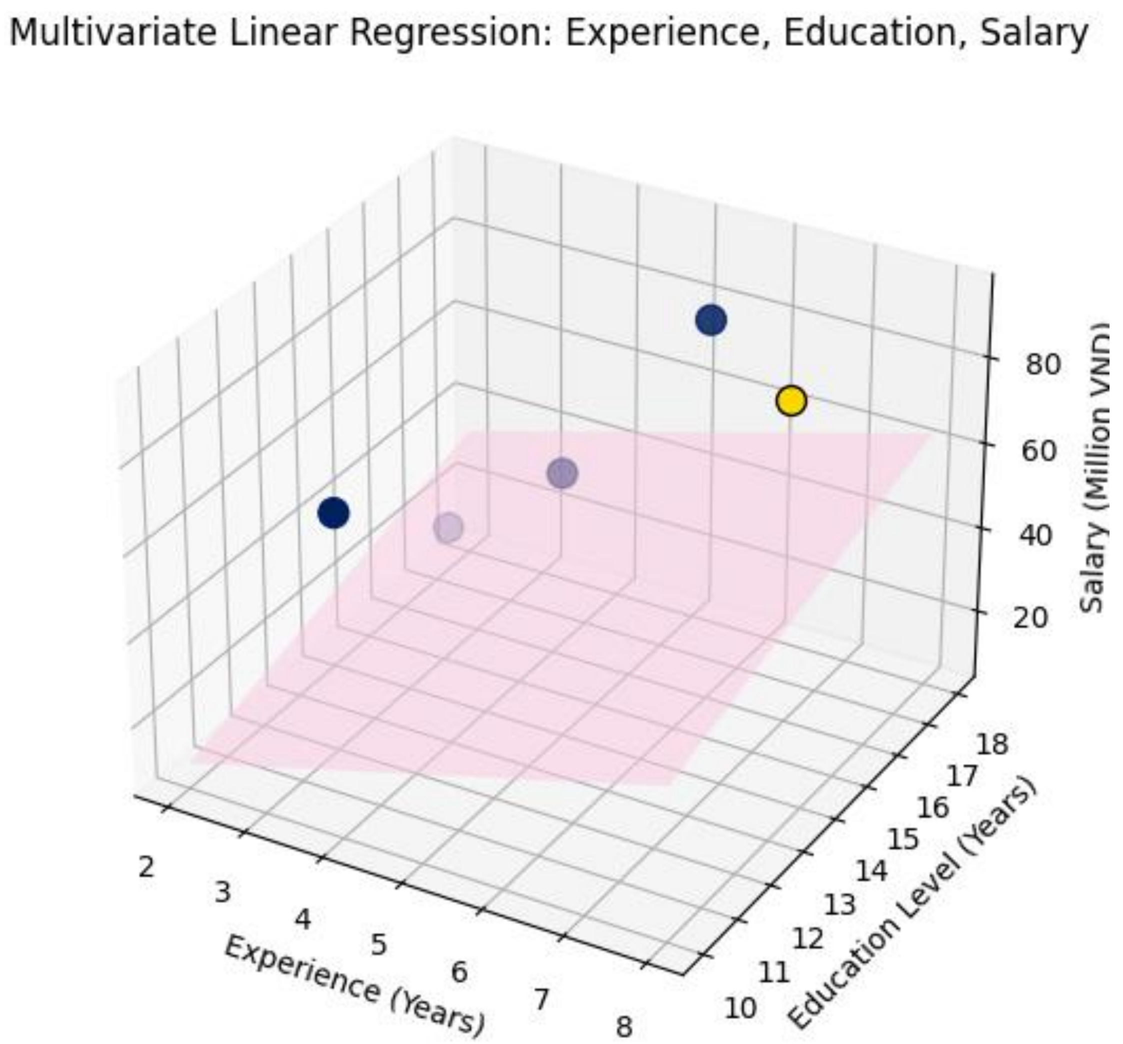




Vectorized Linear Regression

Vectorized Linear Regression

❖ Multivariate Dataset



Experience	Education	Salary
3	12	60
4	13	55
5	14	66
6	15	93
7	16	?

Vectorized Linear Regression

❖ Recap: Vector

x ~ (1)

5

Scalar

x ~ (4)

5
9
2
1

Vector

x ~ (4, 4)

5	**6**	**7**	**9**
9	**8**	**7**	**6**
2	**3**	**4**	**5**
1	**2**	**3**	**4**

Matrix

Vectorized Linear Regression

❖ Recap: Vector

x ~ (4, 5)

5	6	7	8	9
9	8	7	6	5
2	3	4	5	6
1	2	3	4	5

Matrix

x ~ (5, 4)

5	9	2	1
6	8	3	2
7	7	4	3
8	6	5	4
9	5	6	5

Transpose of a matrix

x ~ (4, 4)

5	6	7	8
9	8	7	6
2	3	4	5
1	2	3	4

Square Matrix

Exercise 01

❖ Recap: Vector

Some matrix properties:

- $AB \neq BA$ $\quad (A \in \mathbb{R}^{m\times m}, B \in \mathbb{R}^{m\times m})$
- $(rA)^T = rA^T$ $\quad (A \in \mathbb{R}^{m\times N})$
- $(A+B)^T = A^T + B^T$ $\quad (A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{m\times n})$
- $(AB)^T = B^T A^T$ $\quad (A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times k})$

$$A + B = B + A$$

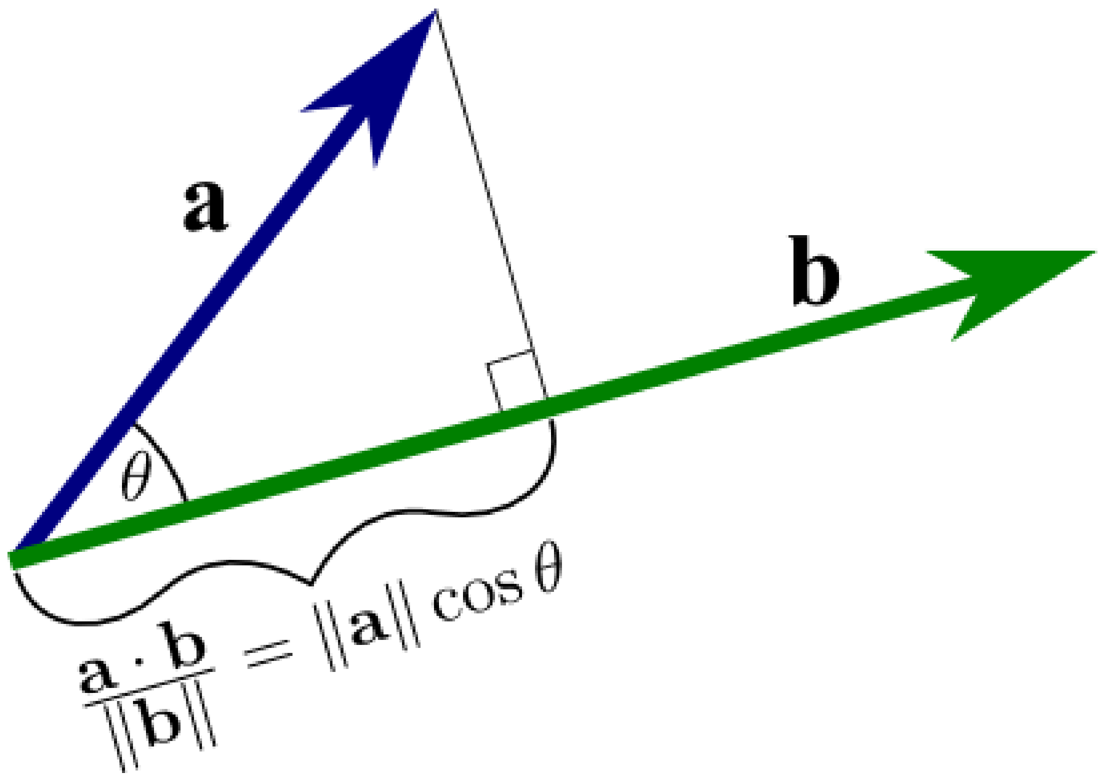
$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix}$$

$$A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{m\times n}$$

$$A + B = \begin{bmatrix} (a_{11}+b_{11}) & \cdots & (a_{1n}+b_{1n}) \\ \vdots & \ddots & \vdots \\ (a_{m1}+b_{m1}) & \cdots & (a_{mn}+b_{mn}) \end{bmatrix} = \begin{bmatrix} (b_{11}+a_{11}) & \cdots & (b_{1n}+a_{1n}) \\ \vdots & \ddots & \vdots \\ (b_{m1}+a_{m1}) & \cdots & (b_{mn}+a_{mn}) \end{bmatrix} = B + A$$

Vectorized Linear Regression

❖ Recap: Dot Product



- $a \cdot b = \|a\|\|b\| \cos\theta$
- $a \cdot b = \sum_{i=1} a_i b_i$

- $a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$
- $a \cdot b = a_1 b_1 + a_2 b_2 + a_3 b_3$

https://mathinsight.org/dot_product

Vectorized Linear Regression

❖ Recap: Dot Product Example

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & c_5 & c_6 \\ c_7 & c_8 & c_9 \end{bmatrix}$$

https://charchithowitzer.medium.com/matrix-multiplication-why-is-it-a-big-deal-cc8ef7490008

Vectorized Linear Regression

❖ Vectorized Linear Regression: Conception

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

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

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

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

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

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} \rightarrow \boldsymbol{\theta}^T = \begin{bmatrix} b & w \end{bmatrix}$$

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

dot product

Vectorized Linear Regression

❖ Vectorized Linear Regression: Conception

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

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

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

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

numbers

What will we do?

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 Linear Regression

❖ Vectorization idea

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

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

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

Vectorized Linear Regression

❖ Vectorized Linear Regression: Conception

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

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

$$\left\{ \begin{array}{l} b = b - \eta \dfrac{\partial L}{\partial b} \\ w = w - \eta \dfrac{\partial L}{\partial w} \end{array} \right.$$

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

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

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad L'_{\boldsymbol{\theta}} = \begin{bmatrix} \dfrac{\partial L}{\partial b} \\ \dfrac{\partial L}{\partial w} \end{bmatrix}$$

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

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

Vectorized Linear Regression

❖ Comparison

Simple

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

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

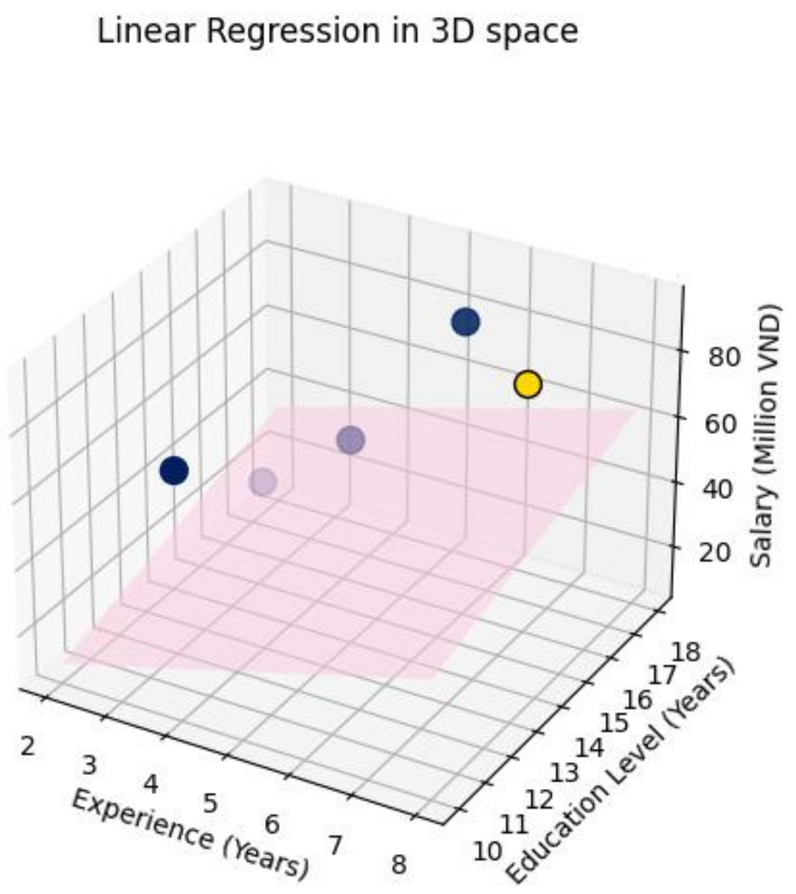
5) Update parameters

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

η is learning rate

Vectorized Linear Regression

❖ Example



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

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \rightarrow \boldsymbol{x}^T = [1 \quad x_1 \quad x_2]$$

$$\hat{y} = \underline{b + w_1 x_1 + w_2 x_2} = \underline{[1 \quad x_1 \quad x_2] \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}} = \boldsymbol{x}^T \boldsymbol{\theta}$$

dot product

Vectorized Linear Regression

❖ N-samples vectorization

Experience	Education	Salary
3	12	60
4	13	55
5	14	66
6	15	93
7	16	?

$$\boldsymbol{x}^{(1)} = \begin{bmatrix} 1 \\ x_1^{(1)} \\ x_2^{(1)} \end{bmatrix}$$

$$\boldsymbol{x}^{(2)} = \begin{bmatrix} 1 \\ x_1^{(2)} \\ x_2^{(2)} \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \end{bmatrix}$$

Vectorized Linear Regression

❖ N-samples vectorization

Experience	Education	Salary
3	12	60
4	13	55
5	14	66
6	15	93
7	16	?

$$\hat{y}^{(1)} = b + w_1 x_1^{(1)} + w_2 x_2^{(1)}$$

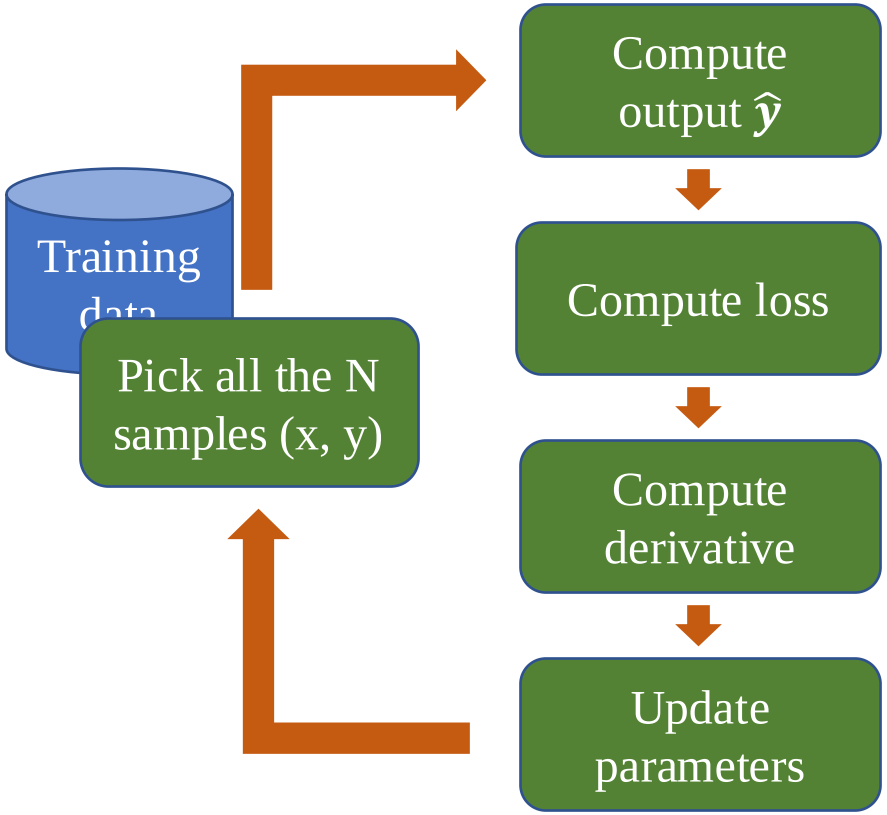
$$\hat{y}^{(2)} = b + w_1 x_1^{(2)} + w_2 x_2^{(2)}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{x\theta} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \end{bmatrix} \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} b + w_1 x_1^{(1)} + w_2 x_2^{(1)} \\ b + w_1 x_1^{(2)} + w_2 x_2^{(2)} \end{bmatrix} = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \end{bmatrix}$$

Vectorized Linear Regression

❖ N-samples vectorization



1) Pick all the N samples from training data

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

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

3) Compute loss

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

4) Compute derivative

$$\boldsymbol{k} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y}) \qquad L'_{\boldsymbol{\theta}} = \boldsymbol{x}^T\boldsymbol{k}$$

5) Update parameters

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

η is learning rate

Vectorized Linear Regression

❖ Forward # 1

Initialize

$w_1 = 3$

$w_2 = 2$

$b = 10$

$\eta = 0.001$

N = 4

N_EPOCH = 2

Experience	Education	Salary
3	12	60
4	13	55
5	14	66
6	15	93

Input Vector

$$\boldsymbol{x} = \begin{bmatrix} 1 & 3 & 12 \\ 1 & 4 & 13 \\ 1 & 5 & 14 \\ 1 & 6 & 15 \end{bmatrix}$$

Model

$w_1 = 3$
$w_2 = 2$

$b = 10$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{x}\boldsymbol{\theta} = \begin{bmatrix} 43 \\ 48 \\ 53 \\ 58 \end{bmatrix}$$

Label Vector

$$\boldsymbol{y} = \begin{bmatrix} 60 \\ 55 \\ 66 \\ 93 \end{bmatrix}$$

Epoch: 1

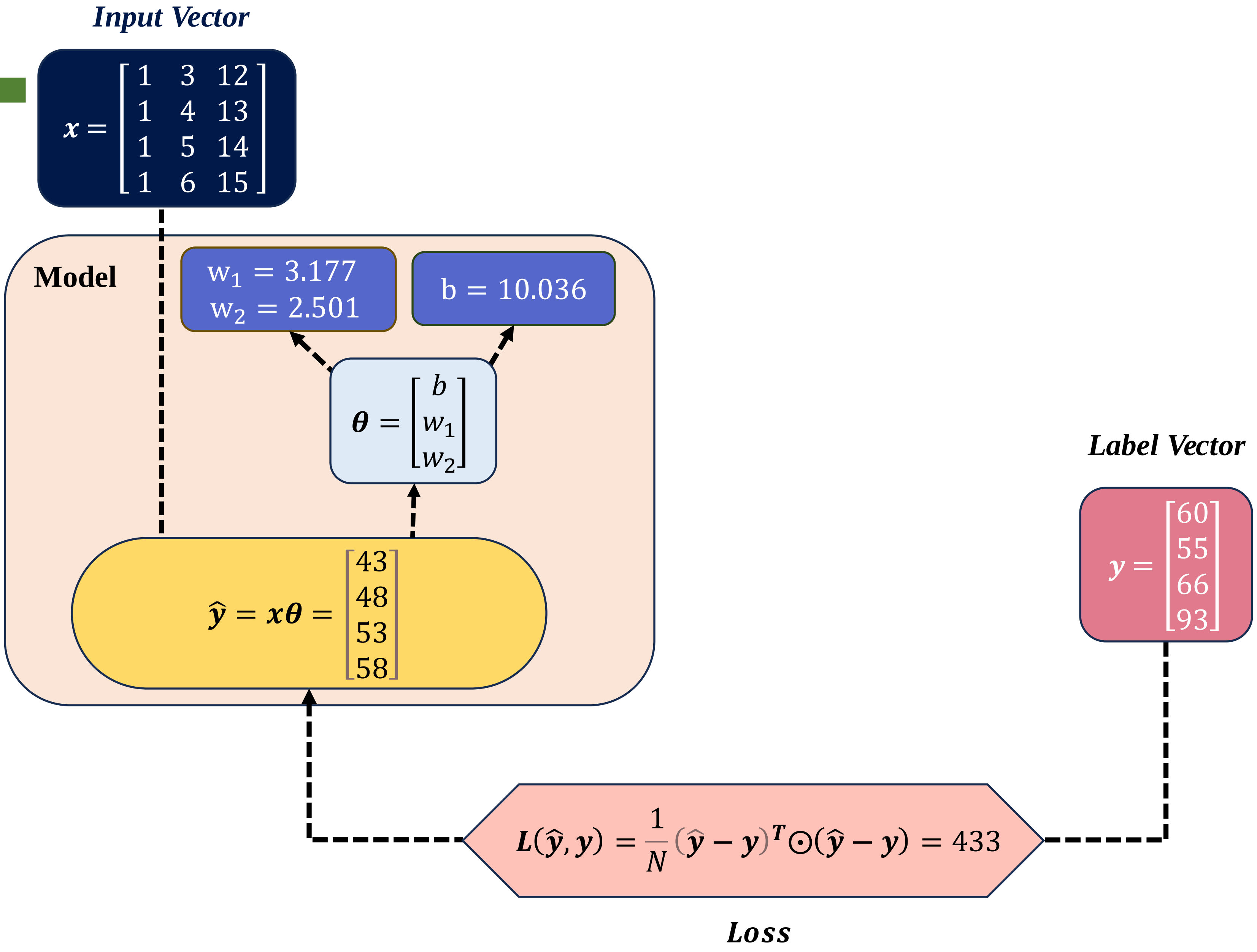
$$\boldsymbol{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{N}(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\boldsymbol{T}} \odot (\hat{\boldsymbol{y}} - \boldsymbol{y}) = 433$$

Loss

AI VIETNAM
All-in-One Course
(TA Session)

Vectorized Linear Regression

❖ Backpropagation #1



<u>Initialize</u>

$w_1 = 3$

$w_2 = 2$

$b = 10$

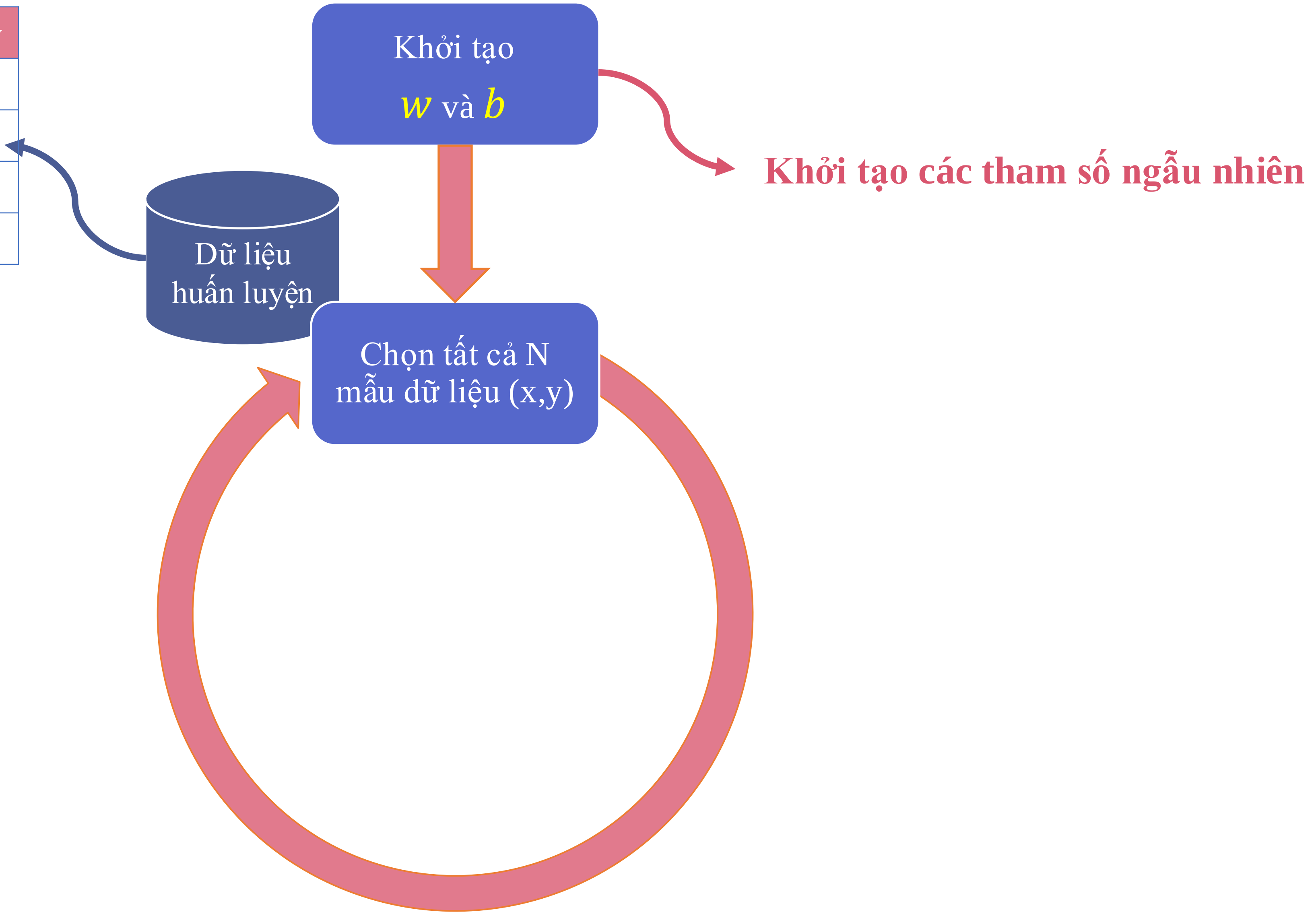
$\eta = 0.001$

N = 4

N_EPOCH = 2

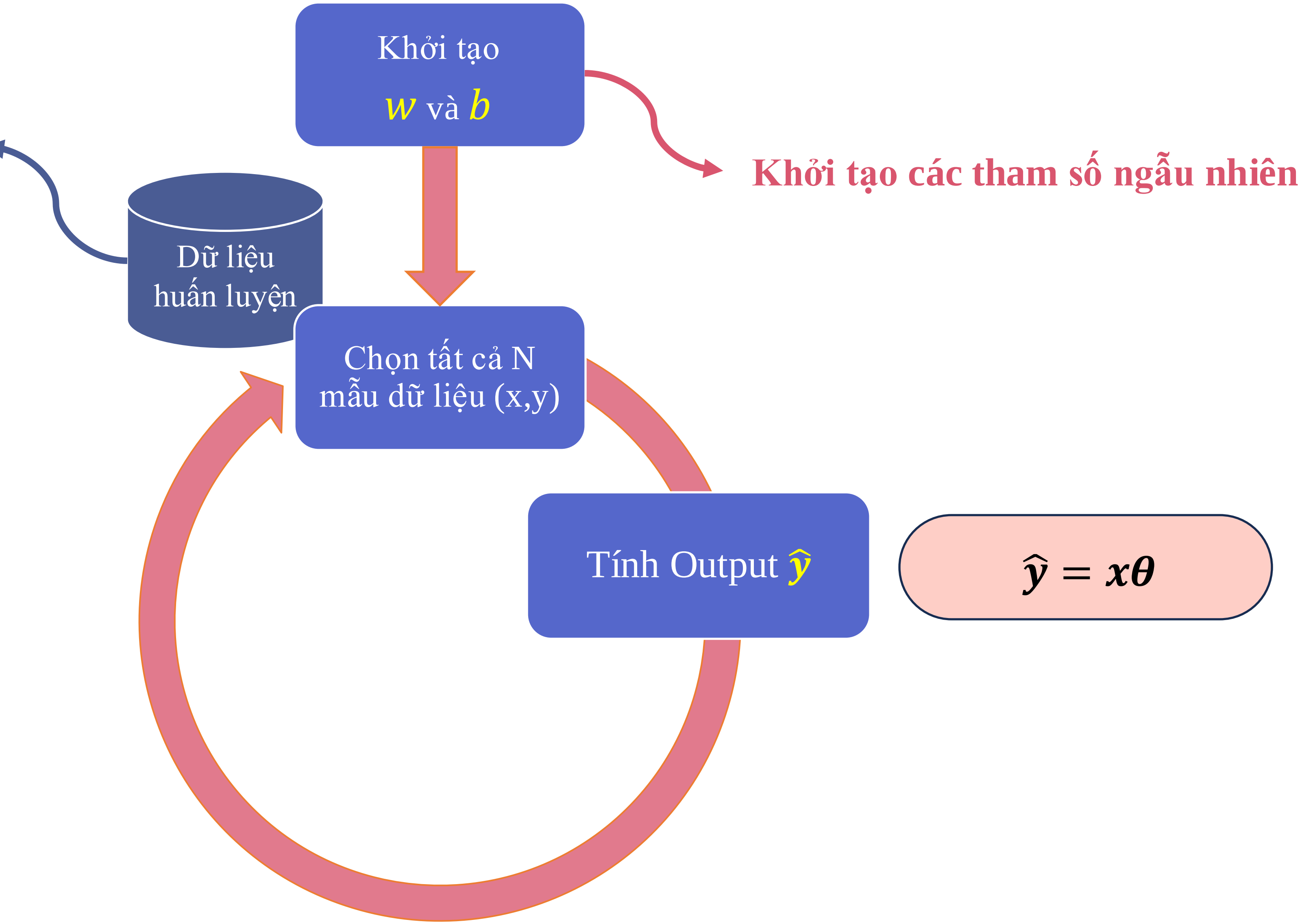
Vectorized Linear Regression

	Experience	Education	Salary
0	3	12	60
1	4	13	55
2	5	14	66
3	6	15	93



Vectorized Linear Regression

	Experience	Education	Salary
0	3	12	60
1	4	13	55
2	5	14	66
3	6	15	93



Vectorized Linear Regression

	Experience	Education	Salary
0	3	12	60
1	4	13	55
2	5	14	66
3	6	15	93

Dữ liệu huấn luyện

Khởi tạo w và b

Khởi tạo các tham số ngẫu nhiên

Chọn tất cả N mẫu dữ liệu (x,y)

Tính Output $\hat{y}$

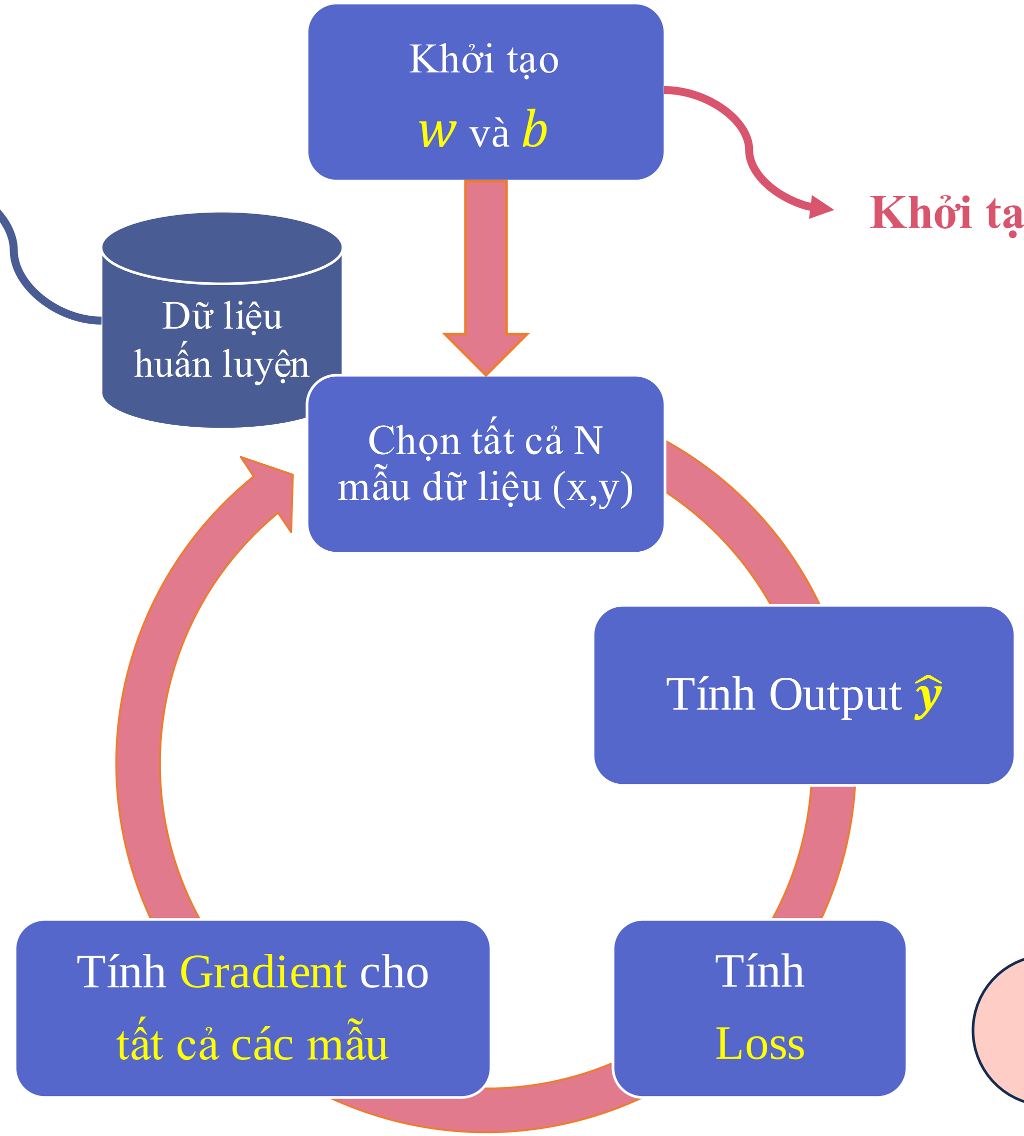
$$\hat{\boldsymbol{y}} = \boldsymbol{x\theta}$$

Tính Loss

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

Vectorized Linear Regression

	Experience	Education	Salary
0	3	12	60
1	4	13	55
2	5	14	66
3	6	15	93



Khởi tạo các tham số ngẫu nhiên

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

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

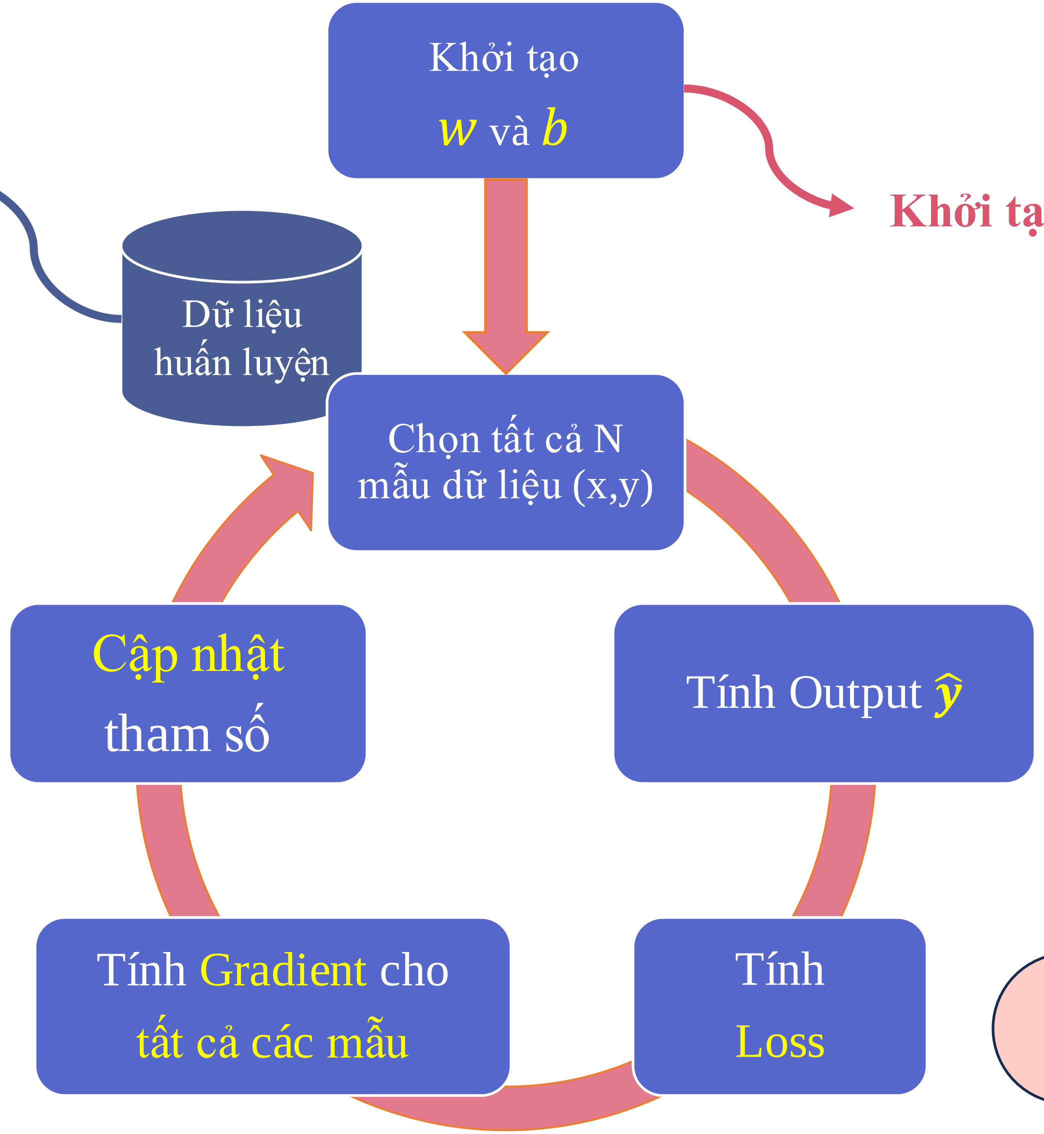
$$\boldsymbol{k} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

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

AI VIETNAM
All-in-One Course
(TA Session)

Vectorized Linear Regression

	Experience	Education	Salary
0	3	12	60
1	4	13	55
2	5	14	66
3	6	15	93



Khởi tạo các tham số ngẫu nhiên

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

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

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

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

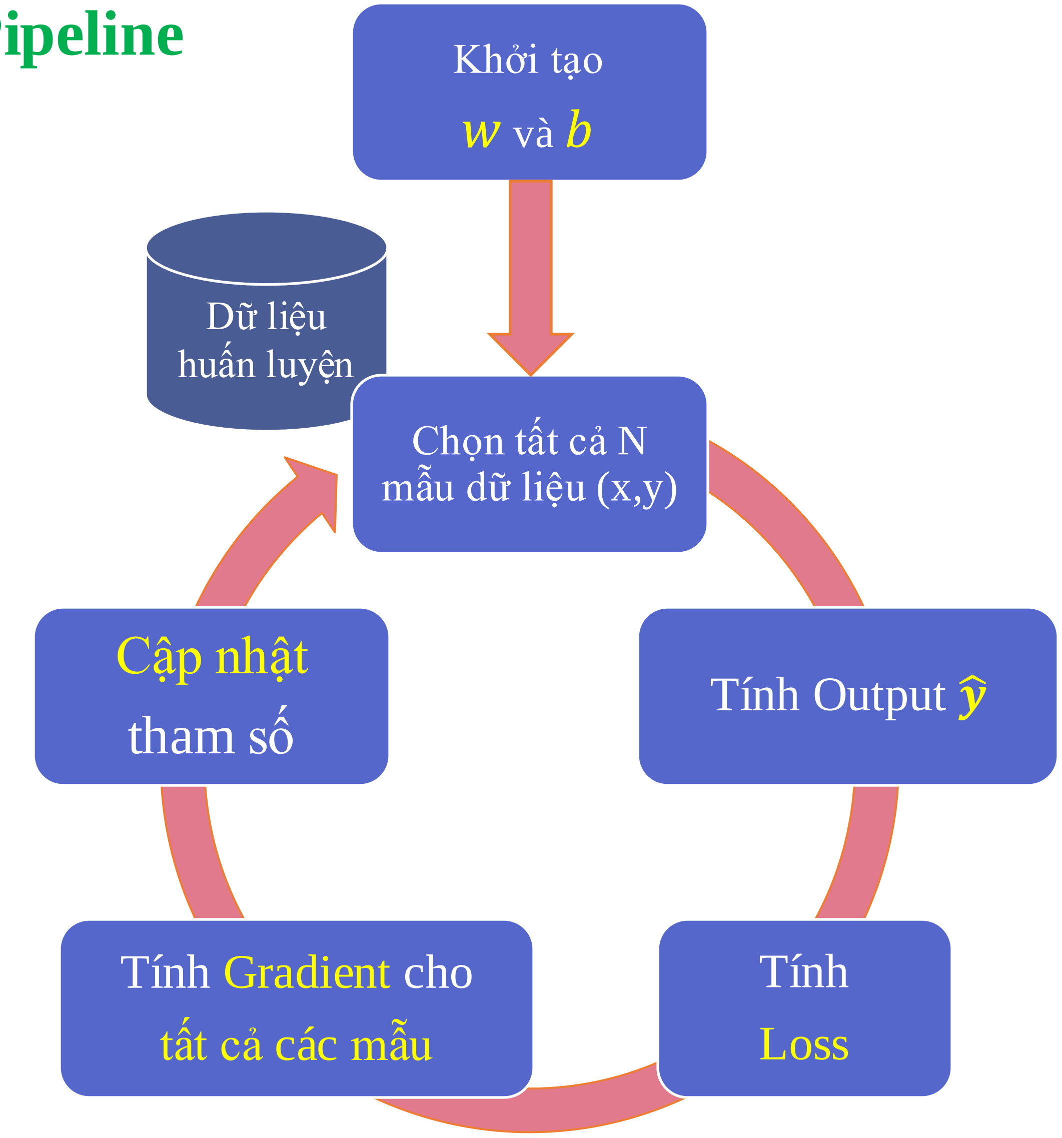
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\frac{L'_{\boldsymbol{\theta}}}{N}$$

QUIZ

Code Implementation

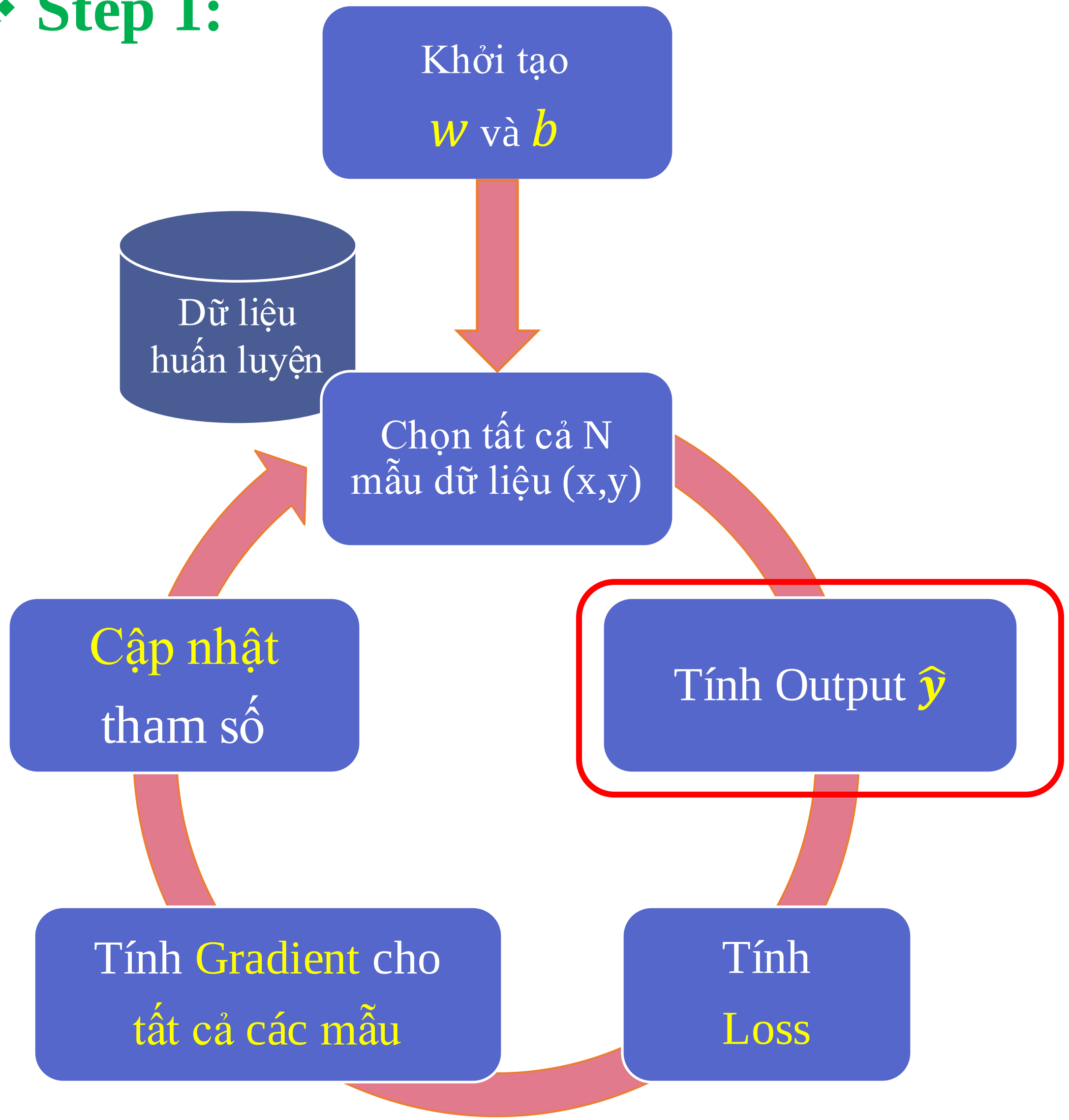
Code Implementation

❖ Gradient Descent Pipeline



Code Implementation

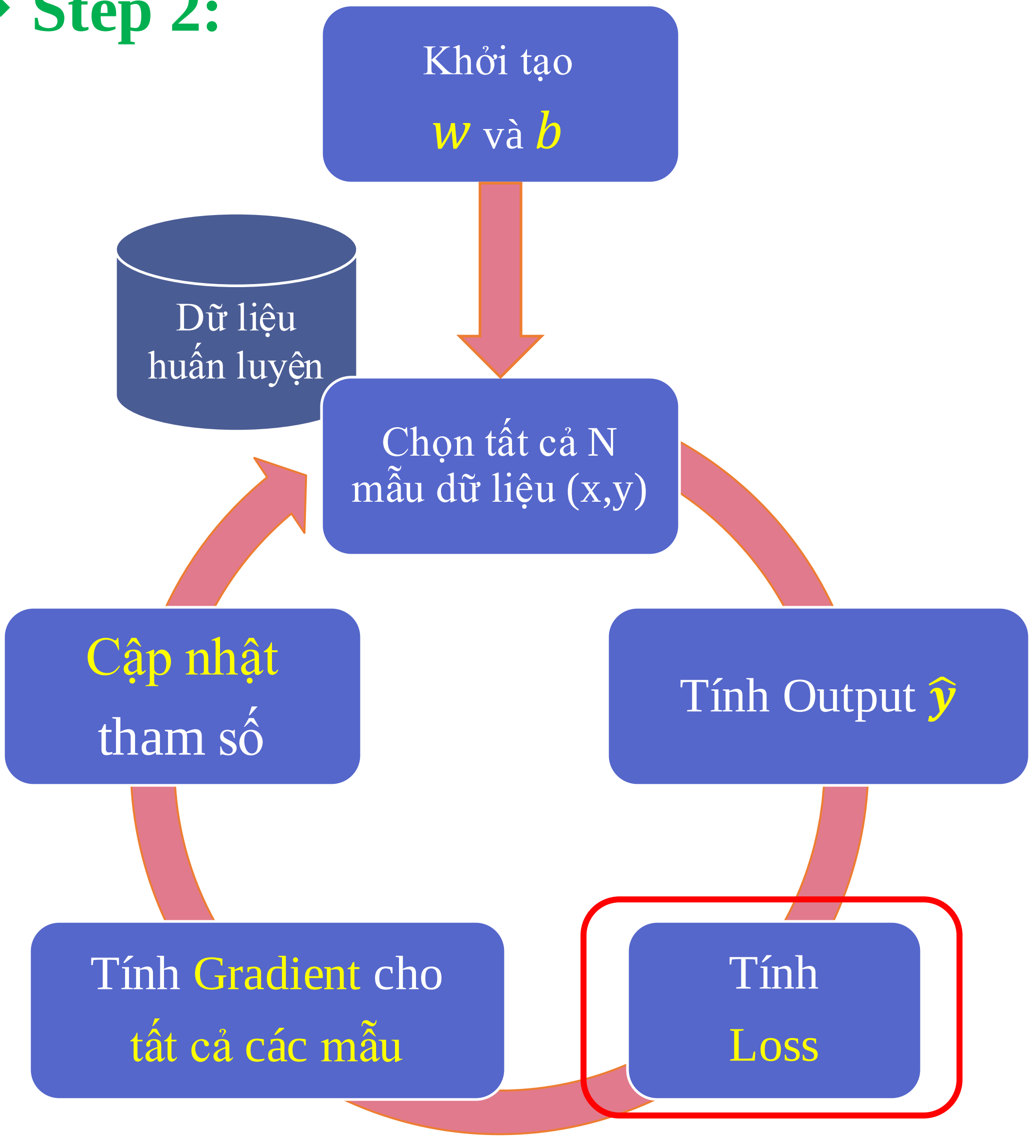
❖ Step 1:




```python
import numpy as np

def predict(X, theta):
    return X.dot(theta)

def compute_loss(y_hat, y):
    N = y.shape[0]

    return np.sum((y_hat - y) ** 2) / N

def compute_gradient(y_hat, y, X):
    N = y.shape[0]
    k = 2 * (y_hat - y)

    return X.T.dot(k) / N

def update_gradient(theta, gradient, lr):
    theta = theta - lr * gradient

    return theta
```

Code Implementation

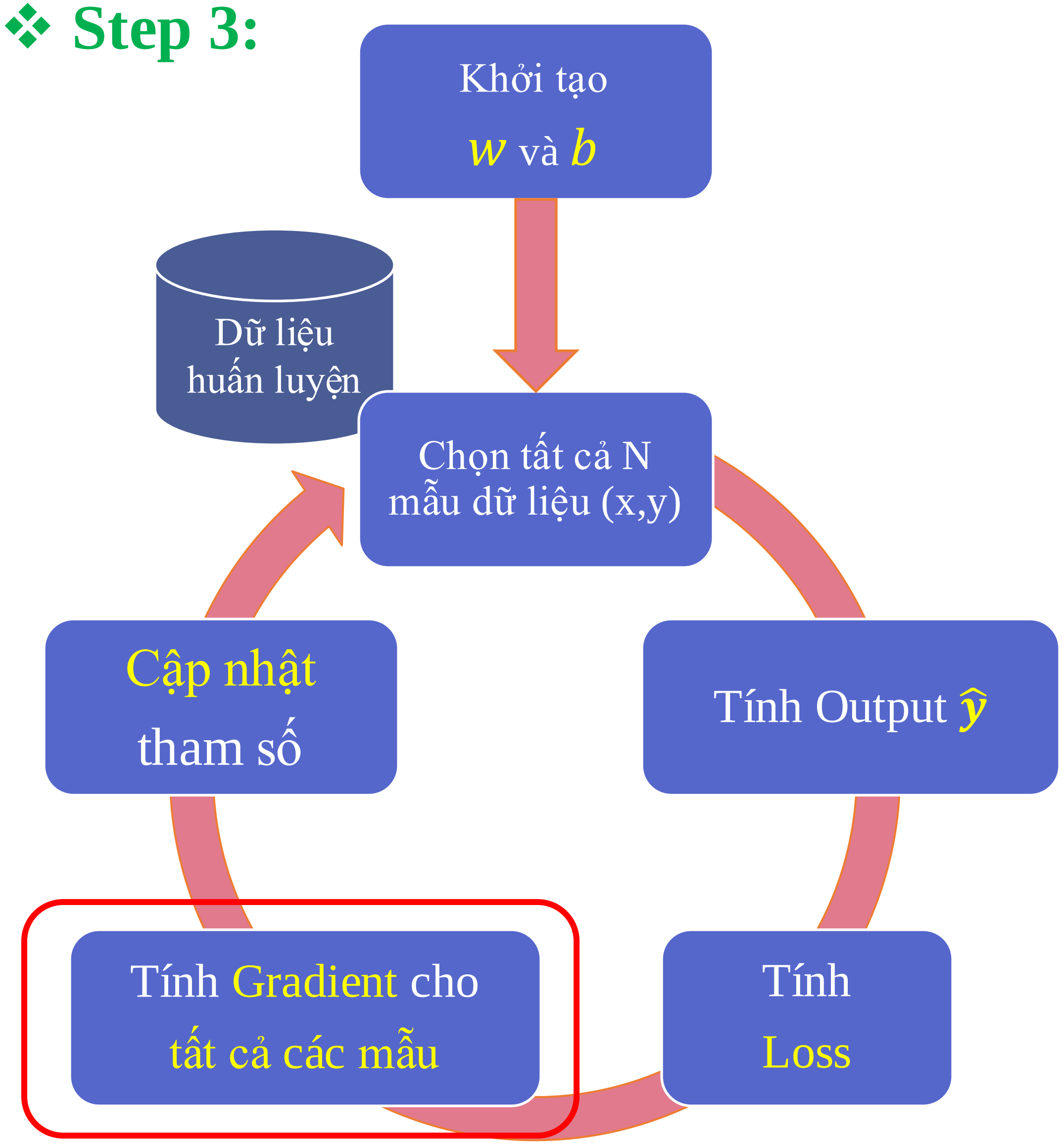
❖ Step 2:




```python
import numpy as np

def predict(X, theta):
    return X.dot(theta)

def compute_loss(y_hat, y):
    N = y.shape[0]

    return np.sum((y_hat - y) ** 2) / N

def compute_gradient(y_hat, y, X):
    N = y.shape[0]
    k = 2 * (y_hat - y)

    return X.T.dot(k) / N

def update_gradient(theta, gradient, lr):
    theta = theta - lr * gradient

    return theta
```

Code Implementation

❖ Step 3:

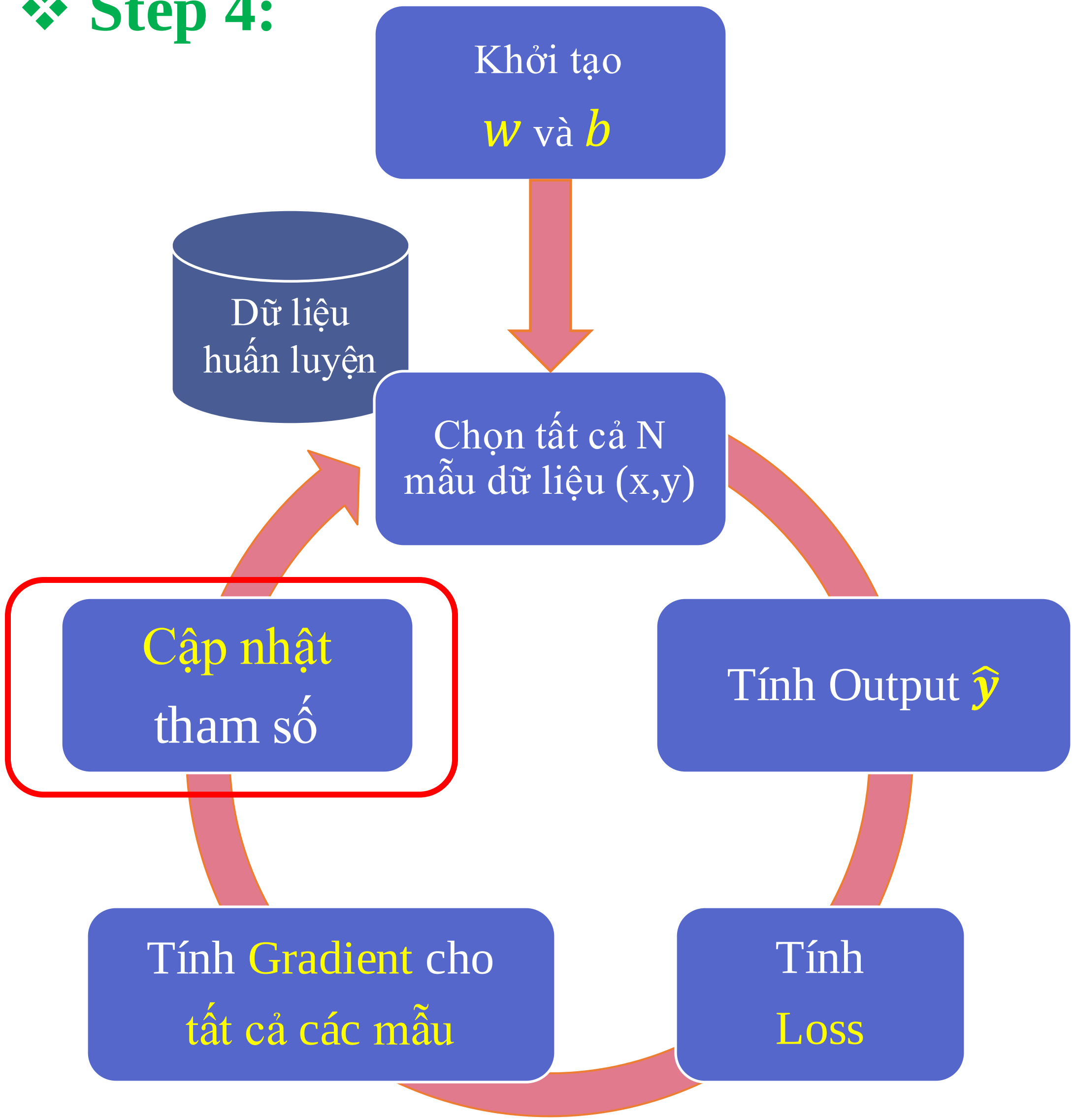



```python
import numpy as np

def predict(X, theta):
    return X.dot(theta)

def compute_loss(y_hat, y):
    N = y.shape[0]

    return np.sum((y_hat - y) ** 2) / N

def compute_gradient(y_hat, y, X):
    N = y.shape[0]
    k = 2 * (y_hat - y)

    return X.T.dot(k) / N

def update_gradient(theta, gradient, lr):
    theta = theta - lr * gradient

    return theta
```

AI VIETNAM
All-in-One Course
(TA Session)

Code Implementation

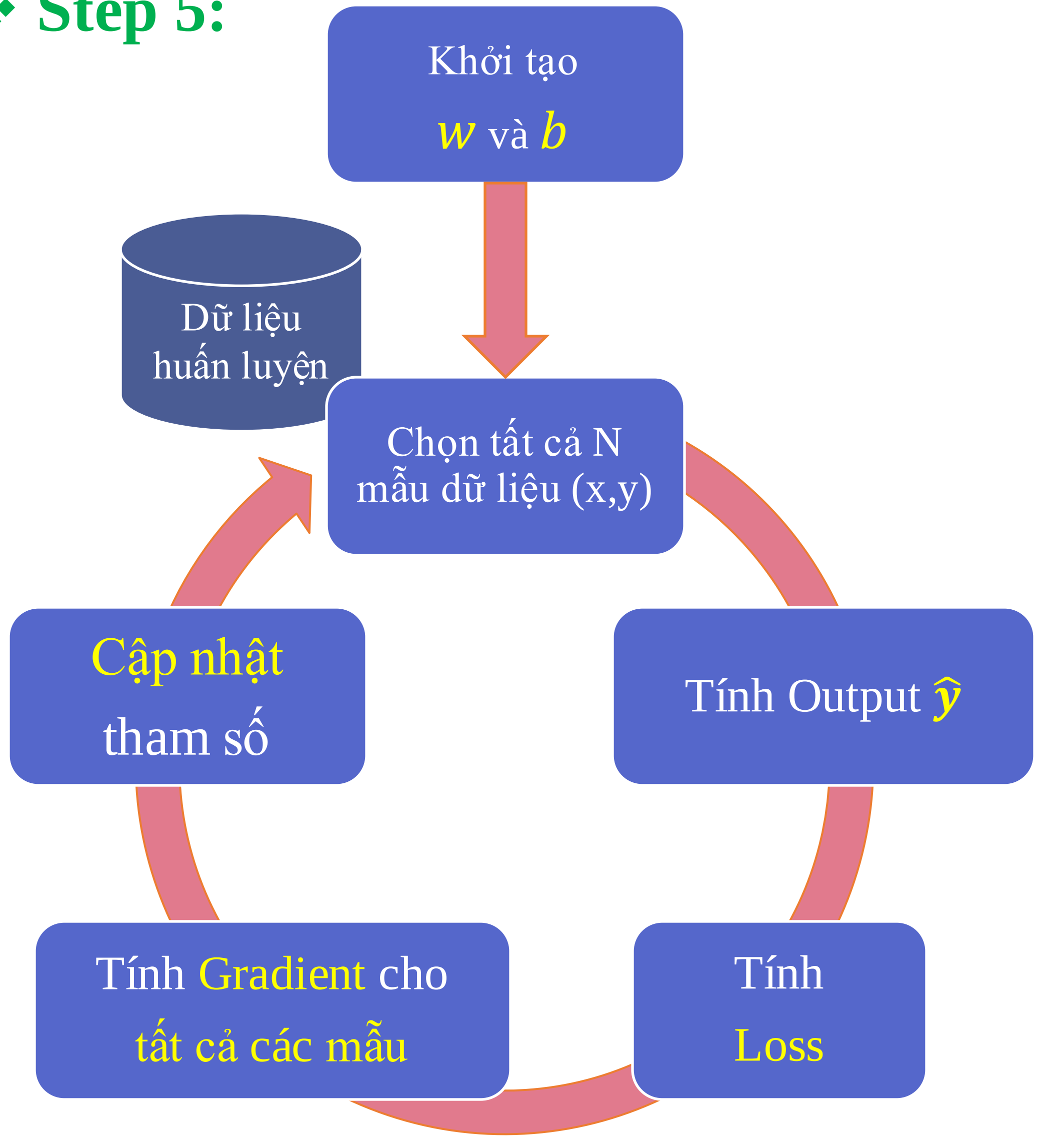
❖ Step 4:




```python
import numpy as np

def predict(X, theta):
    return X.dot(theta)

def compute_loss(y_hat, y):
    N = y.shape[0]

    return np.sum((y_hat - y) ** 2) / N

def compute_gradient(y_hat, y, X):
    N = y.shape[0]
    k = 2 * (y_hat - y)

    return X.T.dot(k) / N

def update_gradient(theta, gradient, lr):
    theta = theta - lr * gradient

    return theta
```

Code Implementation

❖ Step 5:

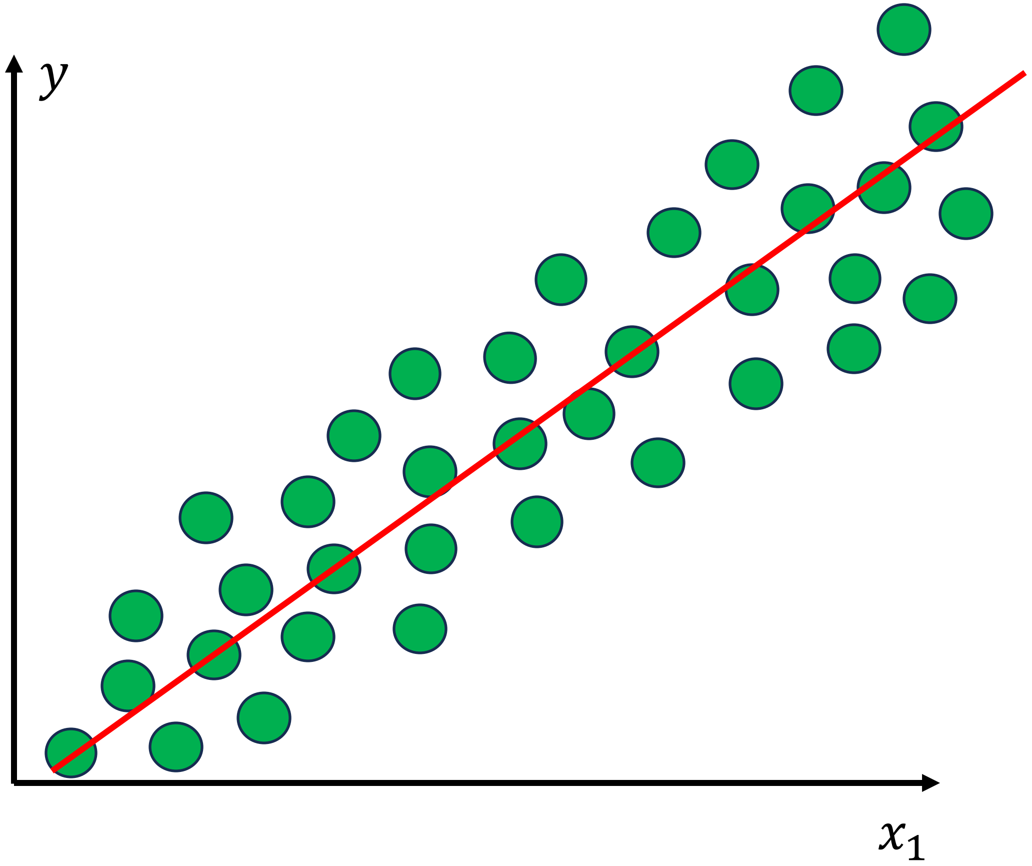



```python
# Các thông số huấn luyện
epoch_max = 4
lr = 0.001  # Giảm learning rate
N = len(y)
losses = []

# Huấn luyện qua các epoch
for epoch in range(epoch_max):
    # Dự đoán giá trị
    y_hat = X_bias.dot(theta)

    # Tính Loss (MSE) - Chia cho N
    loss = np.sum((y_hat - y) ** 2) / N
    losses.append(loss)

    # Tính gradient
    k = 2 * (y_hat - y)
    gradients = X_bias.T.dot(k) / N

    # Cập nhật tham số
    theta = theta - lr * gradients

    print(f'Epoch {epoch+1}: Loss = {loss}, theta = {theta}')
```

Summarization and QA

Summarization

❖ Summarization



In this lecture, we have discussed:

1. Introduction to Vectorized Linear Regression.
2. Discuss about how the Vectorized Linear Regression works.
3. Implement Vectorized Linear Regression.

Question

?


Thank you!