AI VIETNAM
All-in-One Course

From Linear Regression to Logistic Regression

Quang-Vinh Dinh
PhD in Computer Science

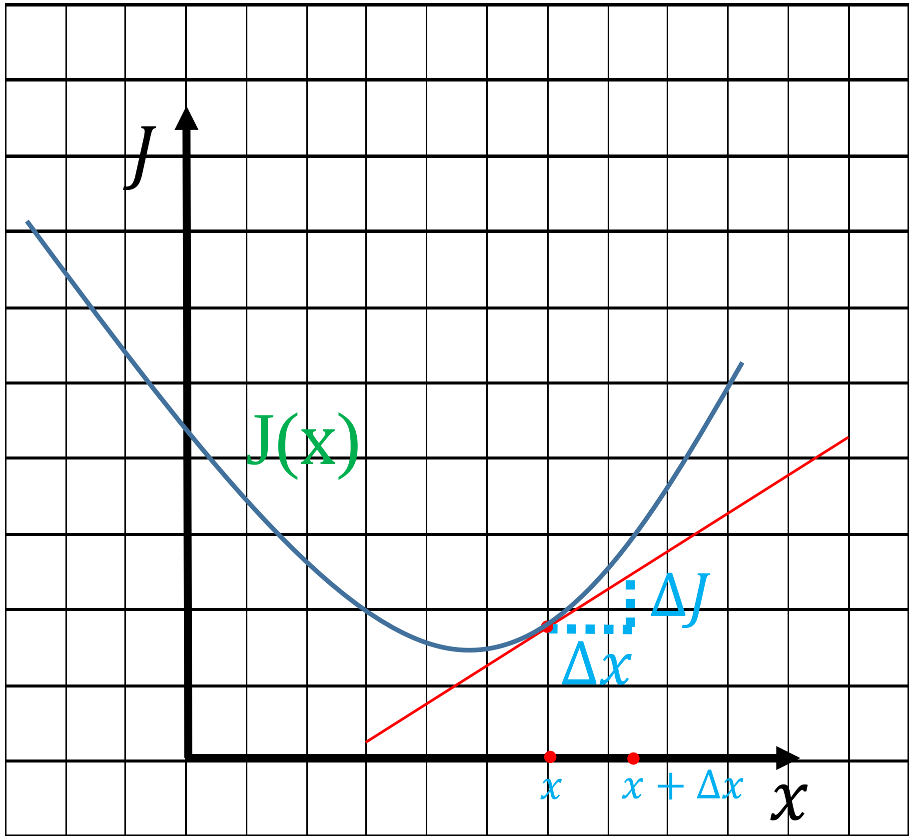
Year 2023

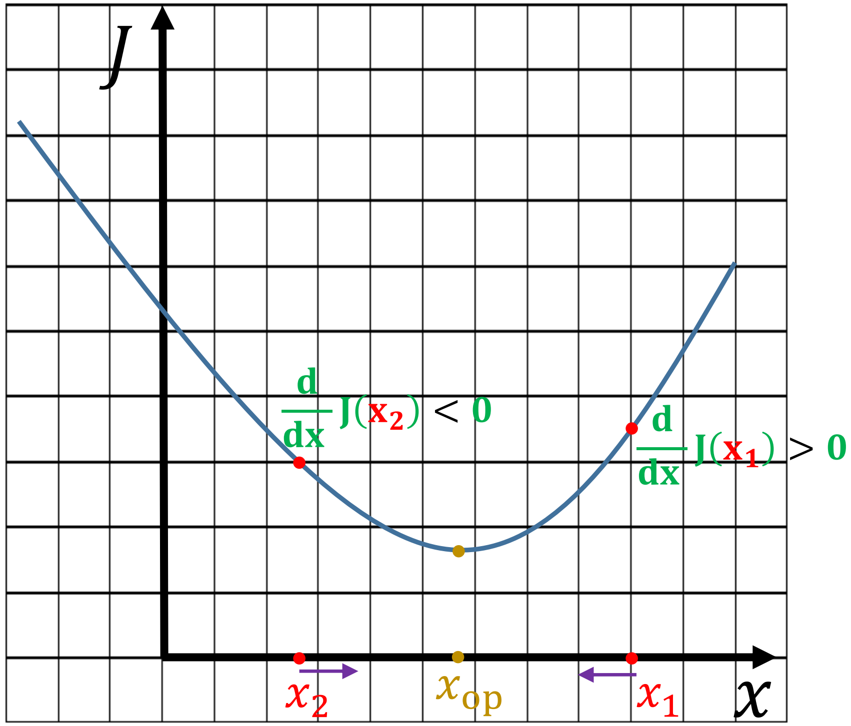
Outline

- Optimization Review
- Linear Regression Review
- Logistic Regression
- Examples
- Vectorization
- Implementation (optional)

Optimization

❖ Gradient descent





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

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

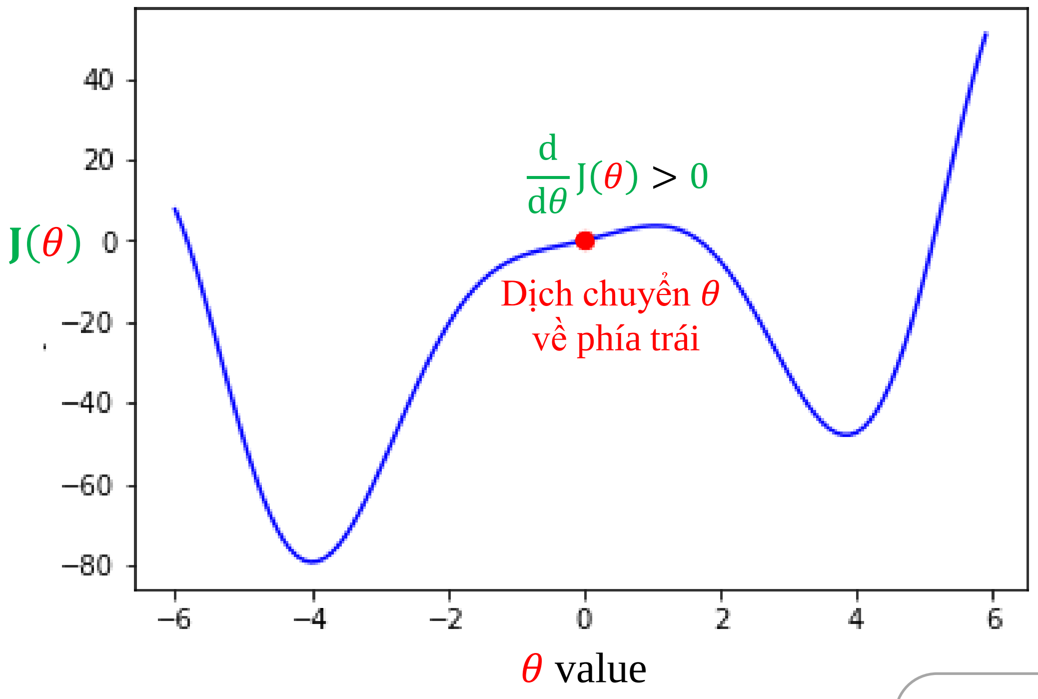
Derivate at x_{old}

learning rate

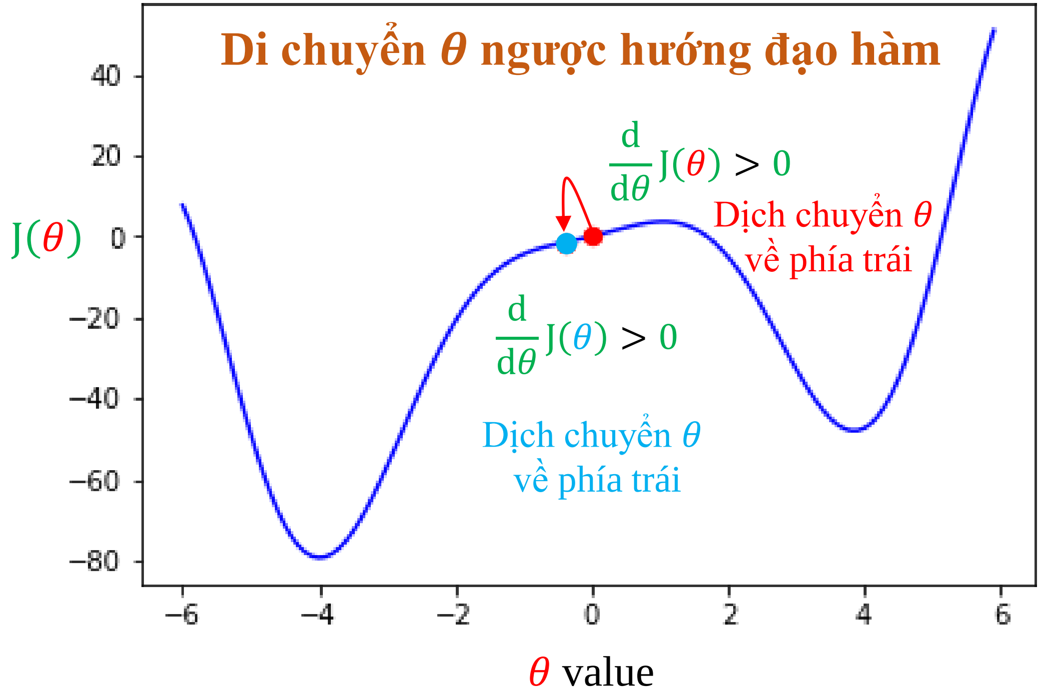
Optimization

❖ **Gradient descent**

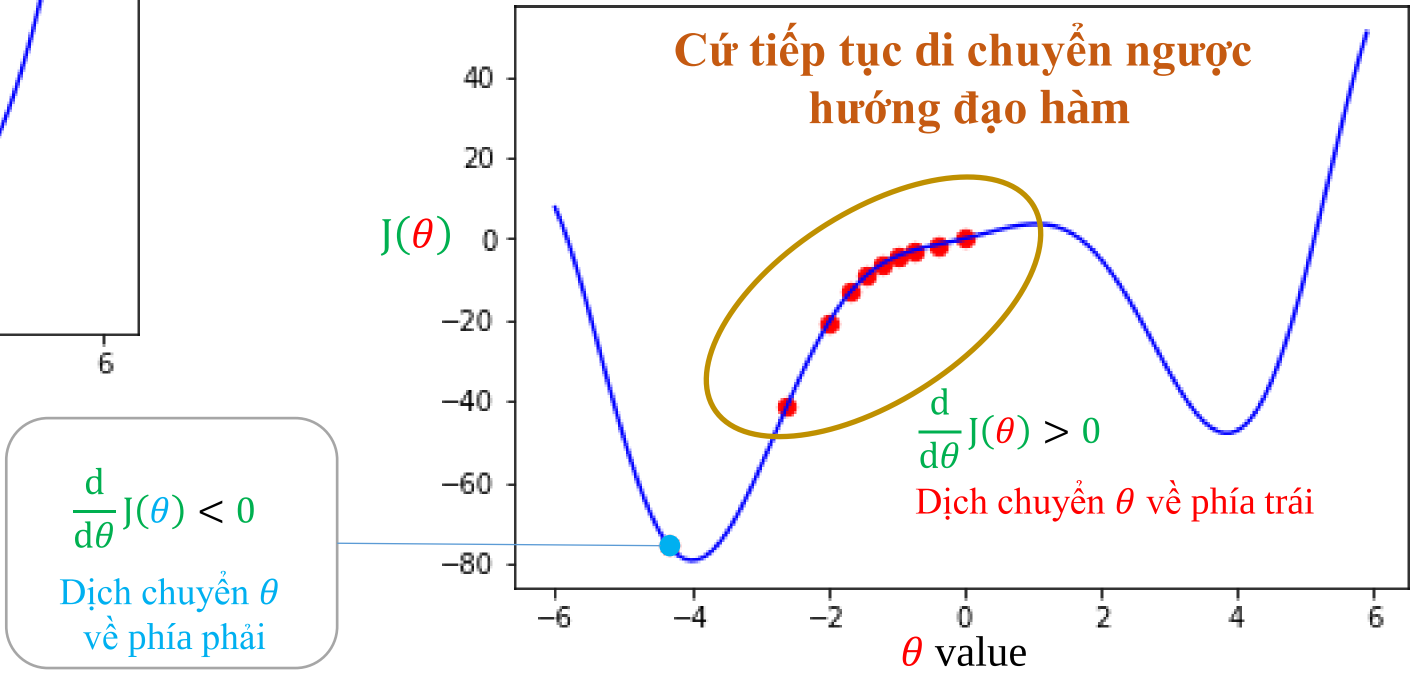
Khởi tạo giá trị θ



Di chuyển θ ngược hướng đạo hàm



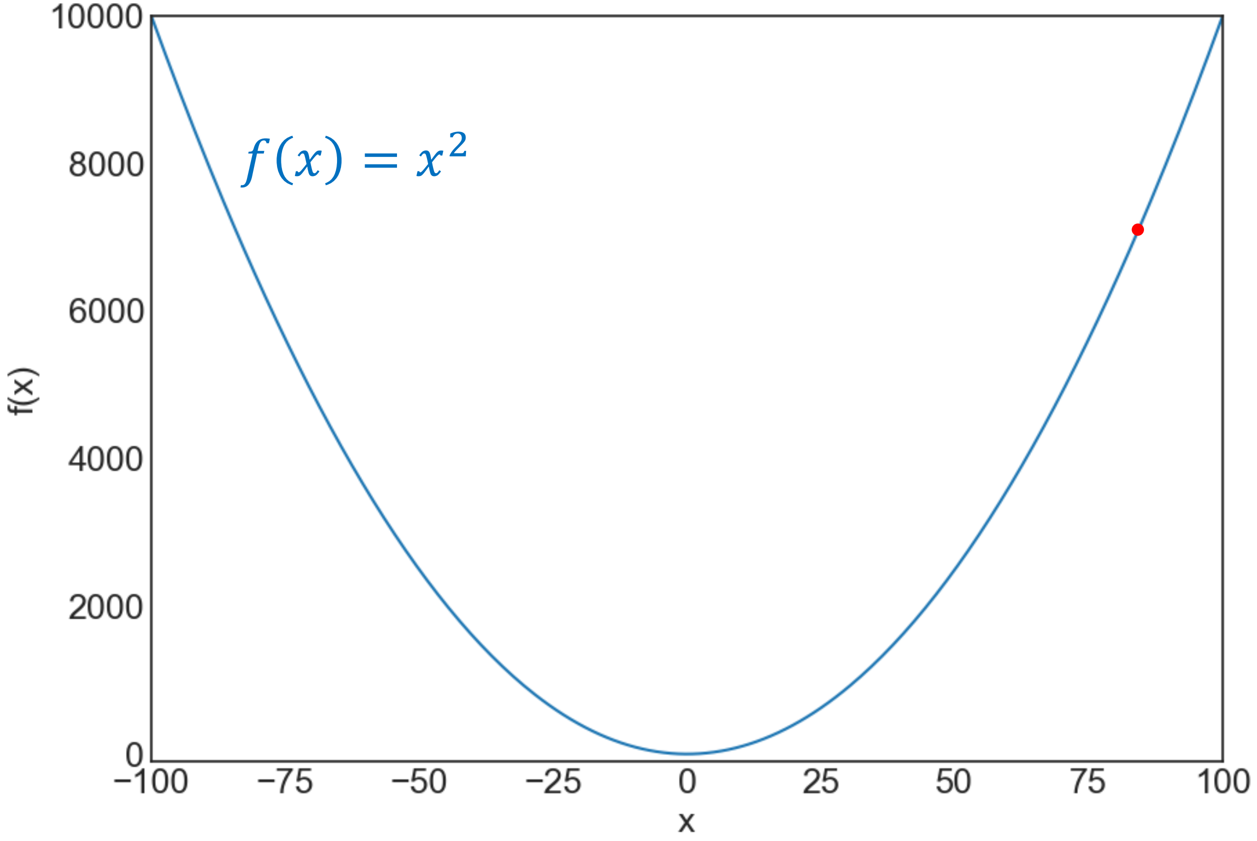
Cứ tiếp tục di chuyển ngược hướng đạo hàm

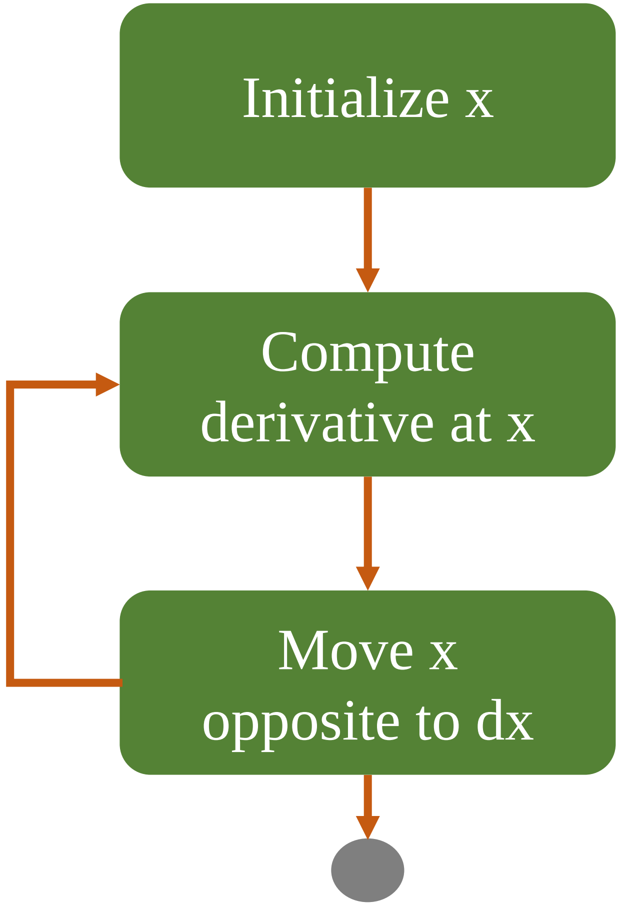


Optimization

❖ Square function

$$x_t = x_{t-1} - \eta f'(x_{t-1})$$





$$-100 \leq x \leq 100$$
$$x \in \mathbb{N}$$

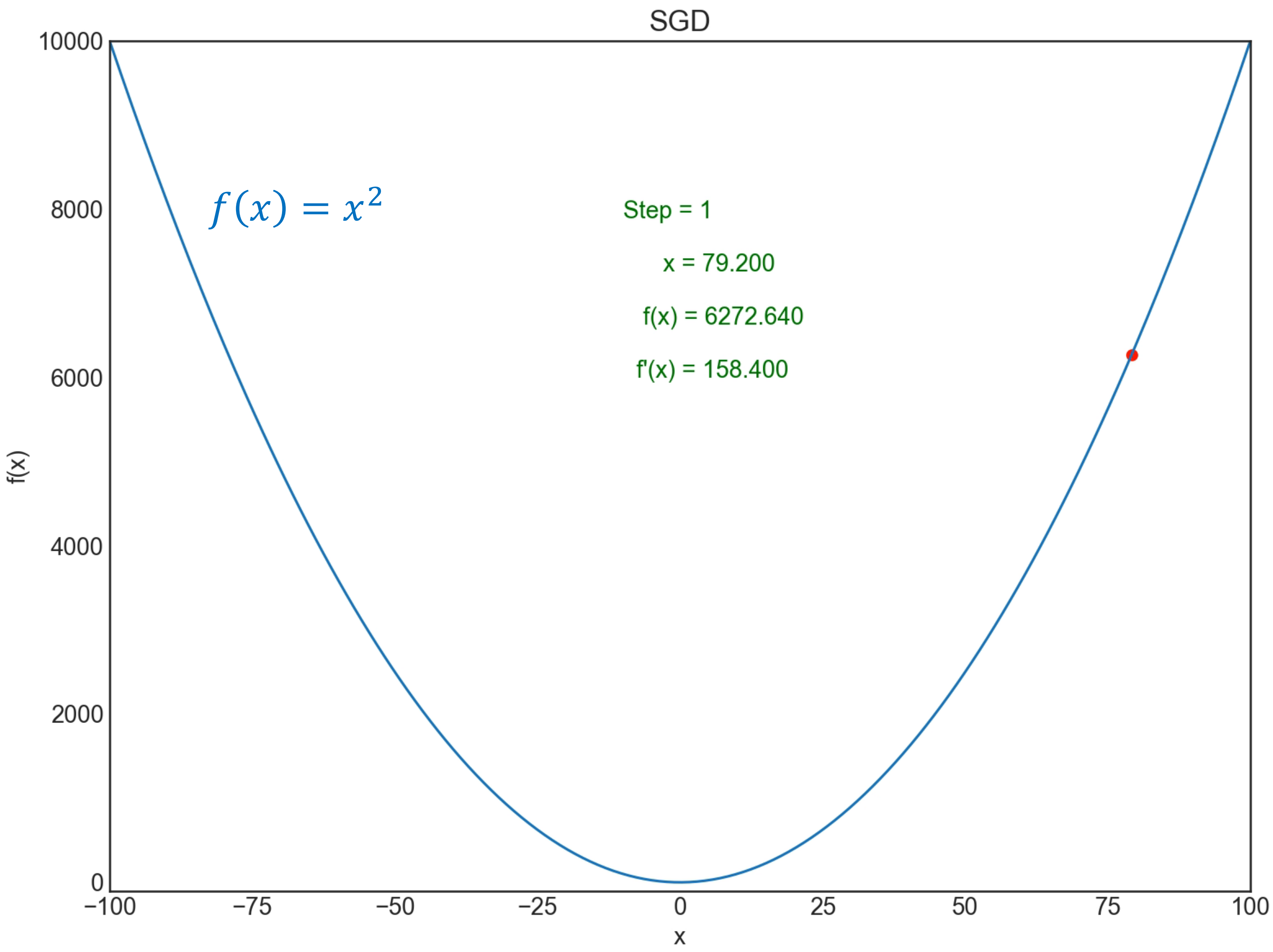
Optimization

❖ **Square function**

$$x_0 = 99.0$$

$$\eta = 0.1$$

$$x_t = x_{t-1} - \eta f'(x)$$



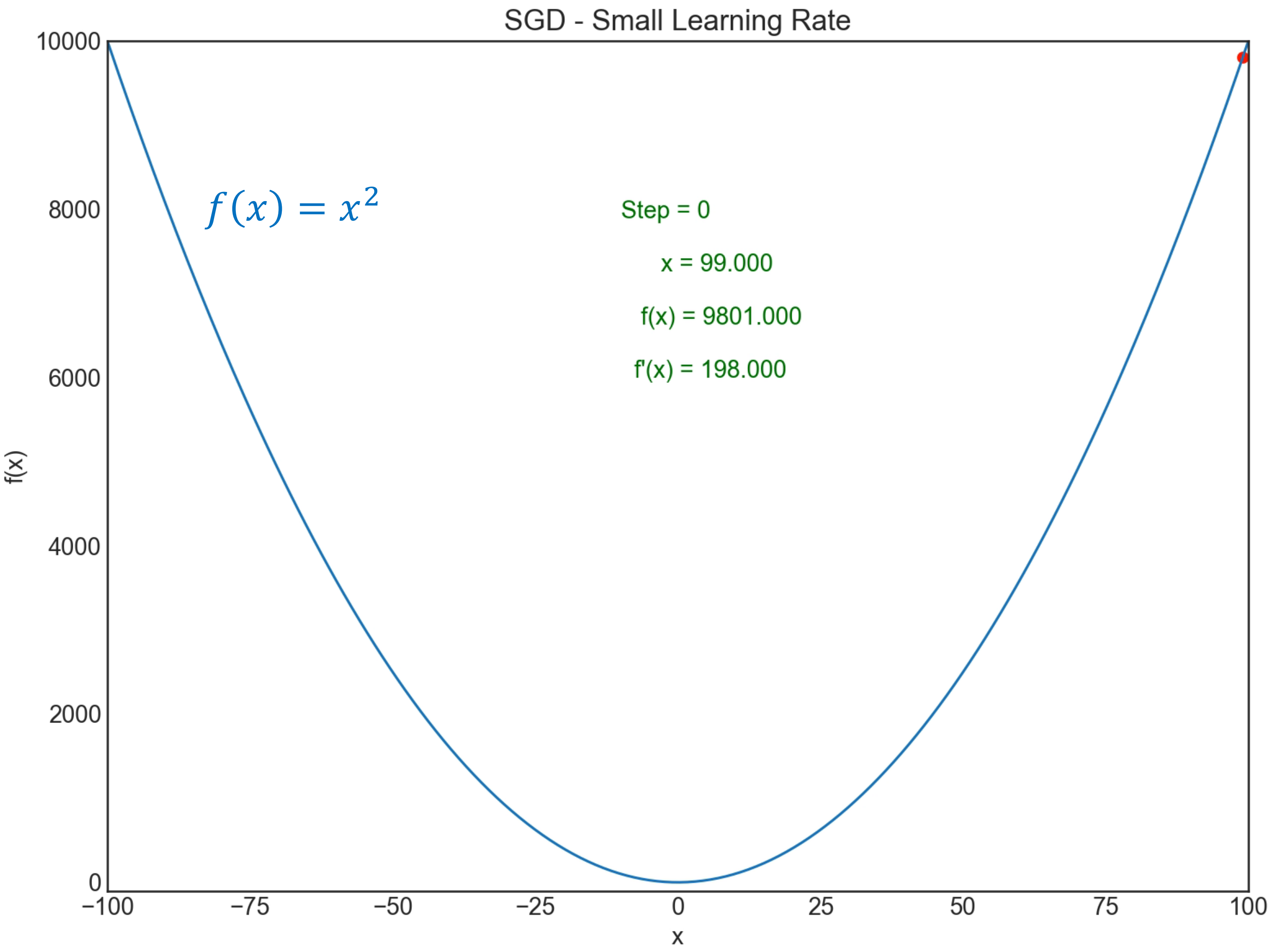
Optimization

❖ **Square function**

$$x_0 = 99.0$$

$$\eta = 0.001$$

$$x_t = x_{t-1} - \eta f'(x)$$



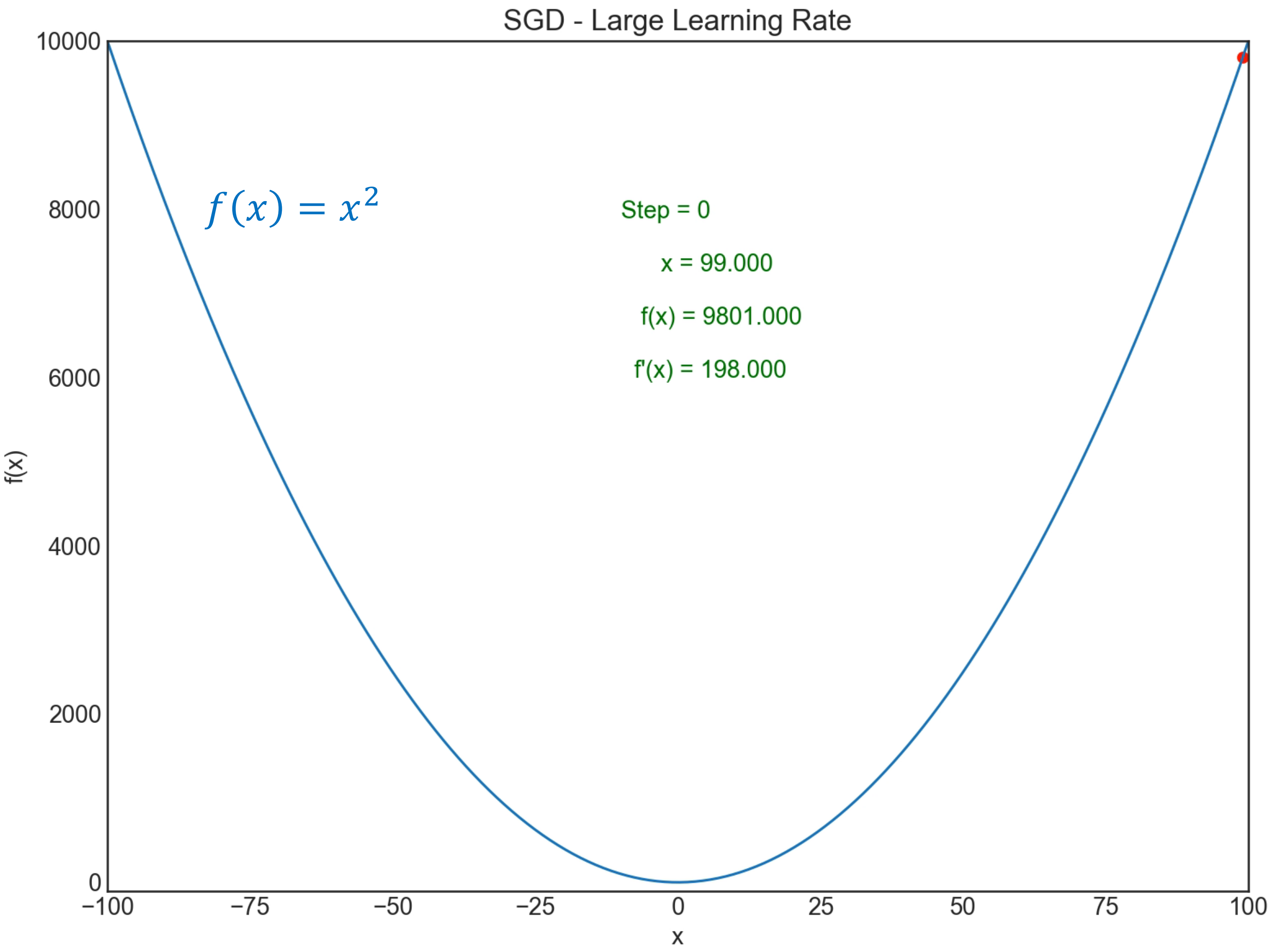
Optimization

❖ **Square function**

$$x_0 = 99.0$$

$$\eta = 0.8$$

$$x_t = x_{t-1} - \eta f'(x)$$



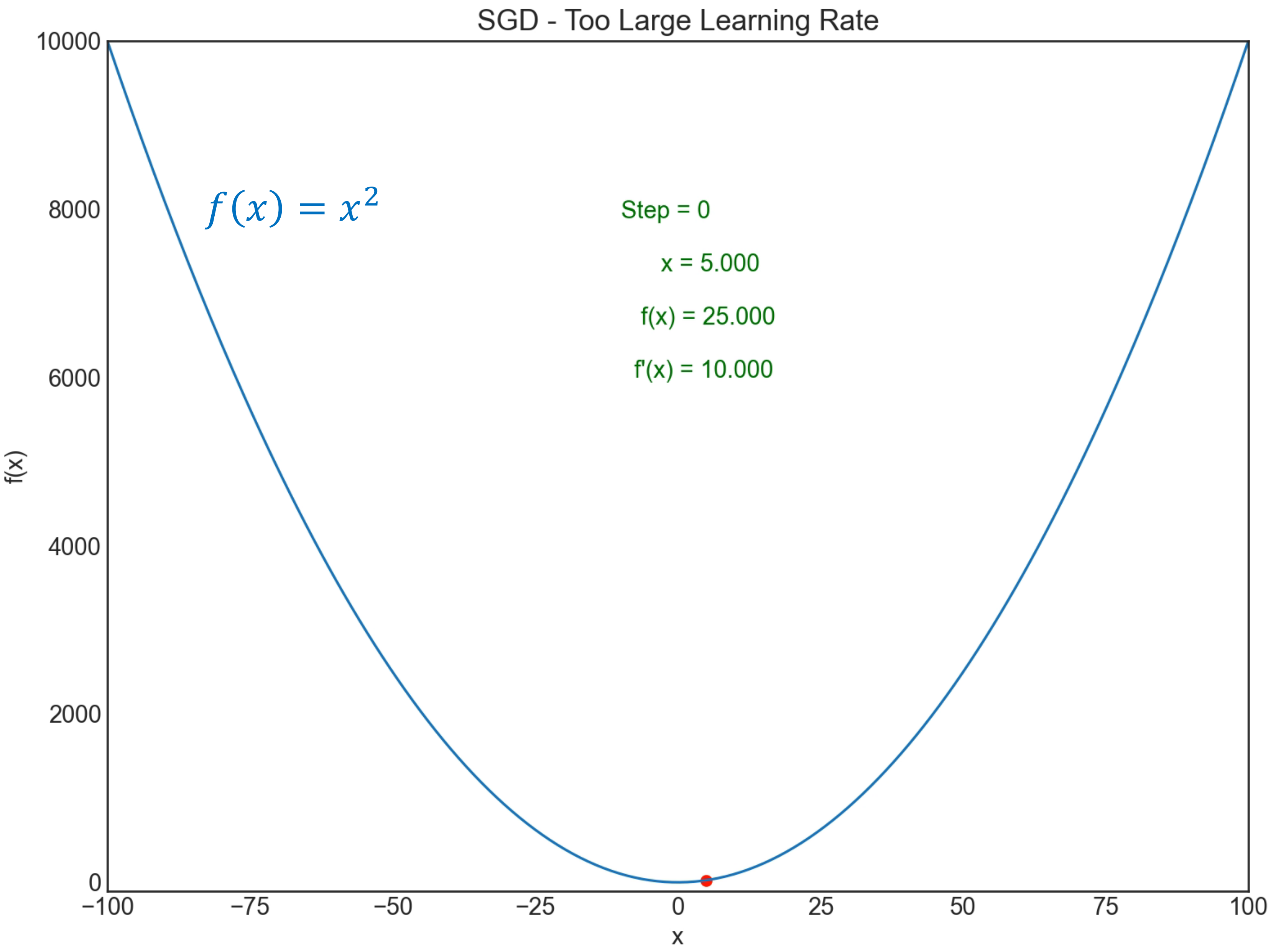
Optimization

❖ **Square function**

$x_0 = 99.0$

$\eta = 1.1$

$$x_t = x_{t-1} - \eta f'(x)$$



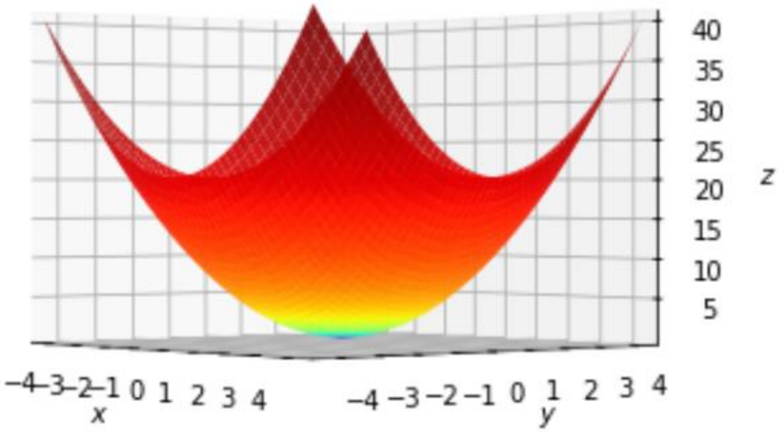
Optimization

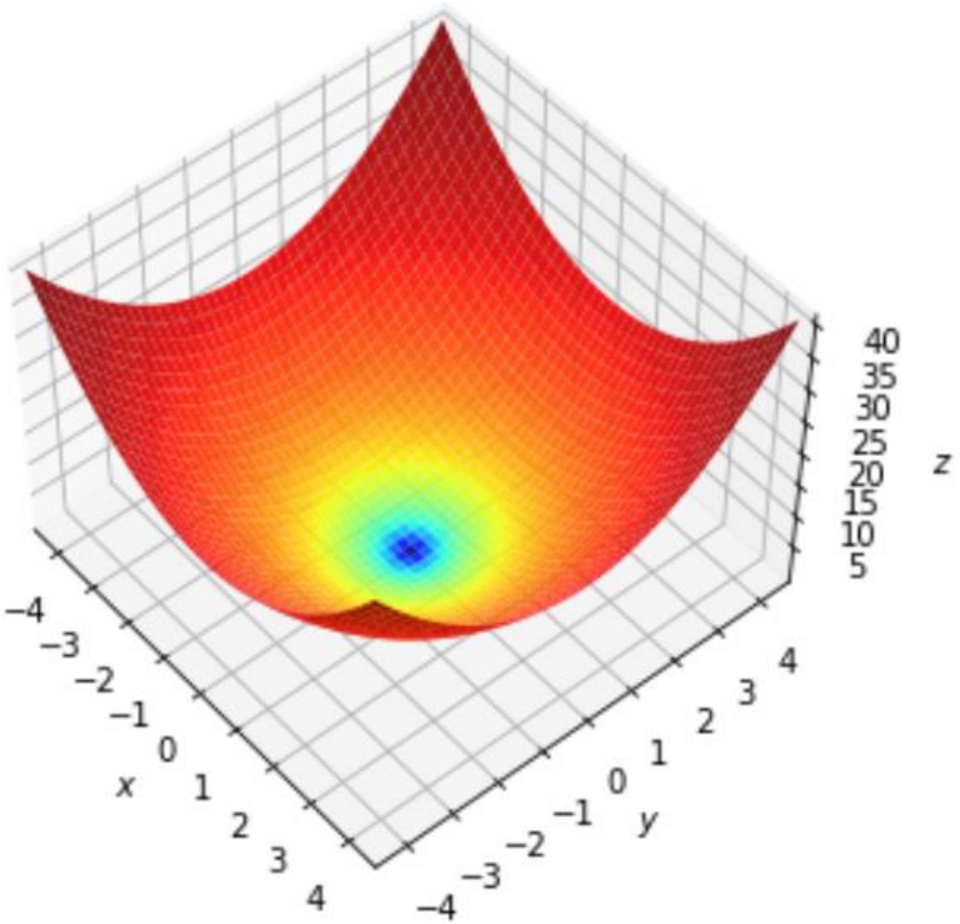
❖ Optimization: 2D function

$$f(x, y) = x^2 + y^2$$

$$-100 \leq x, y \leq 100$$

$$x, y \in \mathbb{N}$$





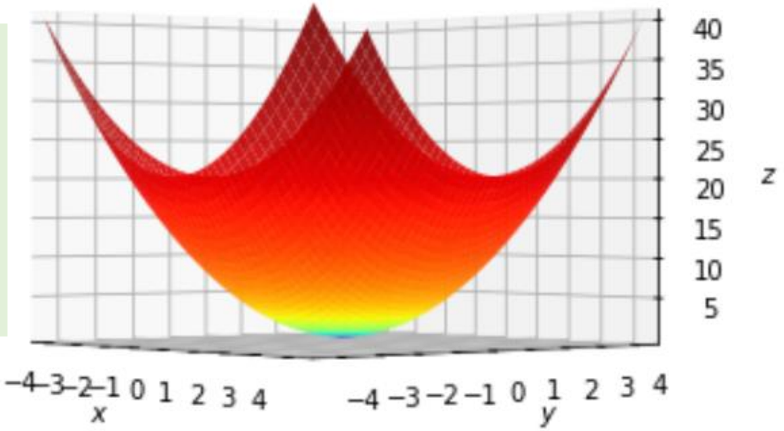
Derivative

❖ Optimization: 2D function

$$f(x,y) = x^2 + y^2$$

$$-100 \le x, y \le 100$$

$$x, y \in \mathbb{N}$$



$$x = x - \eta \frac{\partial f(x,y)}{\partial x}$$

$$y = y - \eta \frac{\partial f(x,y)}{\partial y}$$

$\eta = 1.0$

$$x_0 = 3.0 \qquad y_0 = 4.0$$

$$\frac{\partial f(x_0,y_0)}{\partial x} = 6.0 \qquad \frac{\partial f(x_0,y_0)}{\partial y} = 8.0$$

$$x_1 = 2.0 \qquad y_1 = 3.0$$

$$\frac{\partial f(x_1,y_1)}{\partial x} = 4.0 \qquad \frac{\partial f(x_1,y_1)}{\partial y} = 6.0$$

$$x_2 = 1.0 \qquad y_2 = 2.0$$

$$\frac{\partial f(x_2,y_2)}{\partial x} = 2.0 \qquad \frac{\partial f(x_2,y_2)}{\partial y} = 4.0$$

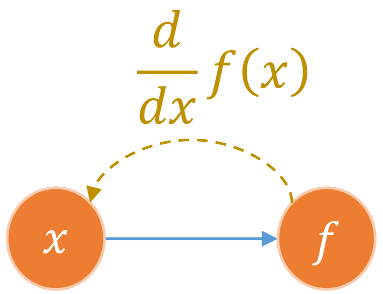
$$x_3 = 0.0 \qquad y_3 = 1.0$$

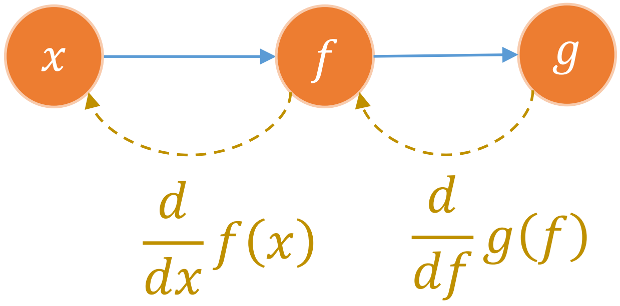
$$\frac{\partial f(x_3,y_3)}{\partial x} = 0.0 \qquad \frac{\partial f(x_3,y_3)}{\partial y} = 0.0$$

$$x_4 = 0.0 \qquad y_4 = 0.0$$

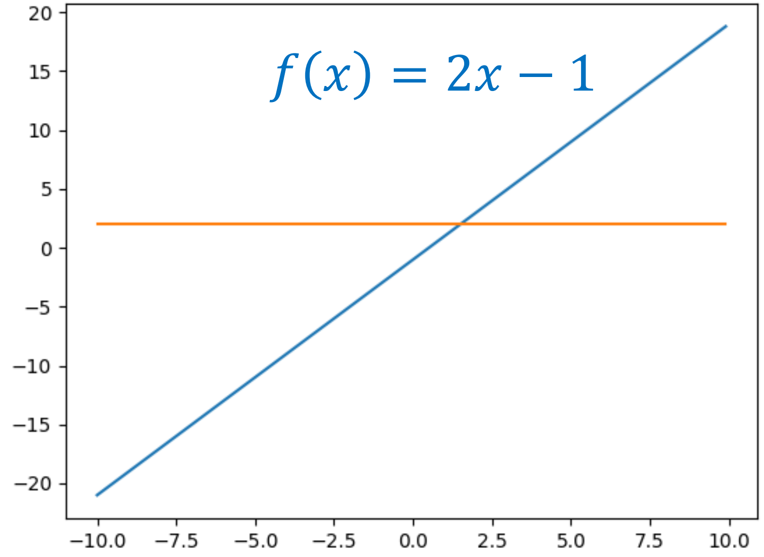
Optimization

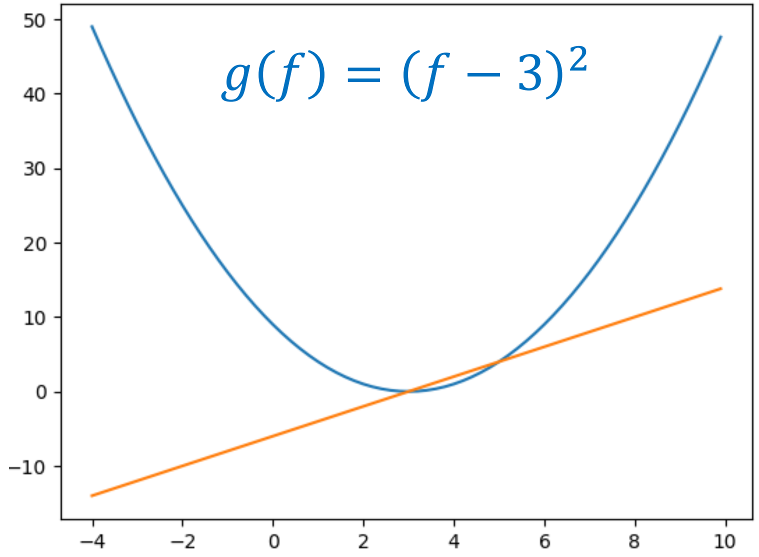
❖ For composite function

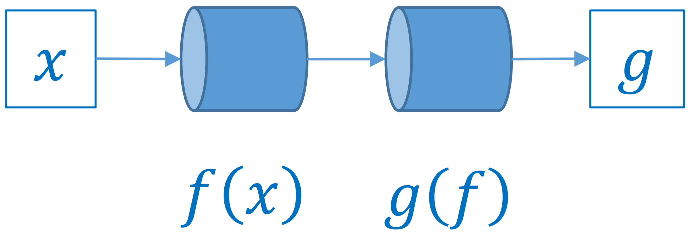


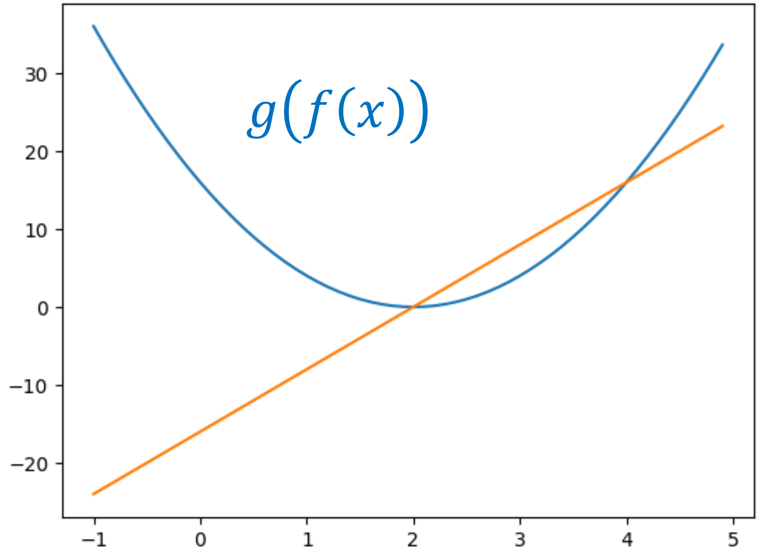


$$\frac{d}{dx}g\big(f(x)\big) = \left[\frac{d}{df}g(f)\right] * \left[\frac{d}{dx}f(x)\right]$$



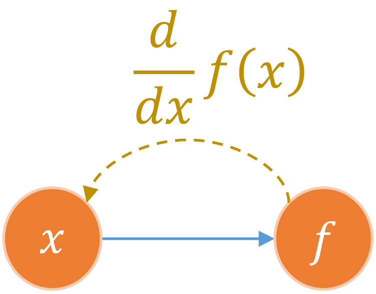


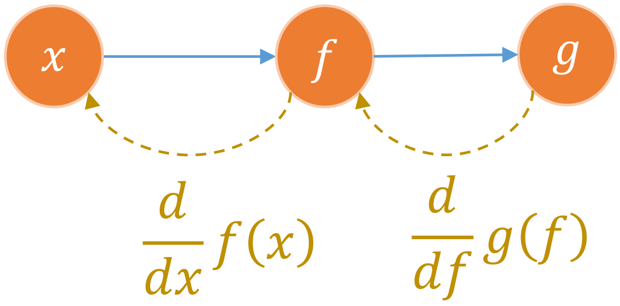




Optimization

❖ For composite function





$$\frac{d}{dx}g\big(f(x)\big) = \left[\frac{d}{df}g(f)\right] * \left[\frac{d}{dx}f(x)\right]$$

$$f(x) = 2x - 1$$

$$g(f) = (f - 3)^2$$

$$\Rightarrow \quad g(x) = (2x - 1 \ - 3)^2$$

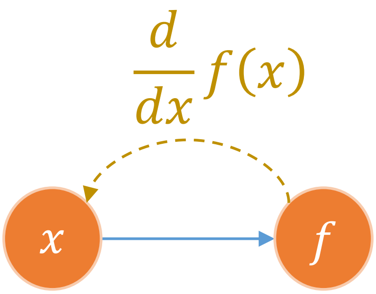
$$= (2x \ - 4)^2$$

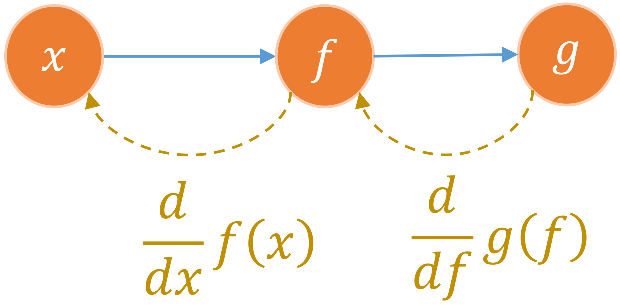
$$\Rightarrow \quad g'(x) = 4(2x - 4)$$

$$= 8x - 16$$

Optimization

❖ **For composite function and chain rule**





$$\frac{d}{dx}g\big(f(x)\big) = \left[\frac{d}{df}g(f)\right] * \left[\frac{d}{dx}f(x)\right]$$

$$f(x) = 2x - 1$$

$$g(f) = (f - 3)^2$$

$$f'(x) = 2$$

$$g'(f) = 2(f - 3)$$

$$\frac{dg}{dx} = \frac{dg}{df}\frac{df}{dx}$$

$$= 2(f - 3)2$$

$$= 4(2x - 1 - 3)$$

$$= 8x - 16$$

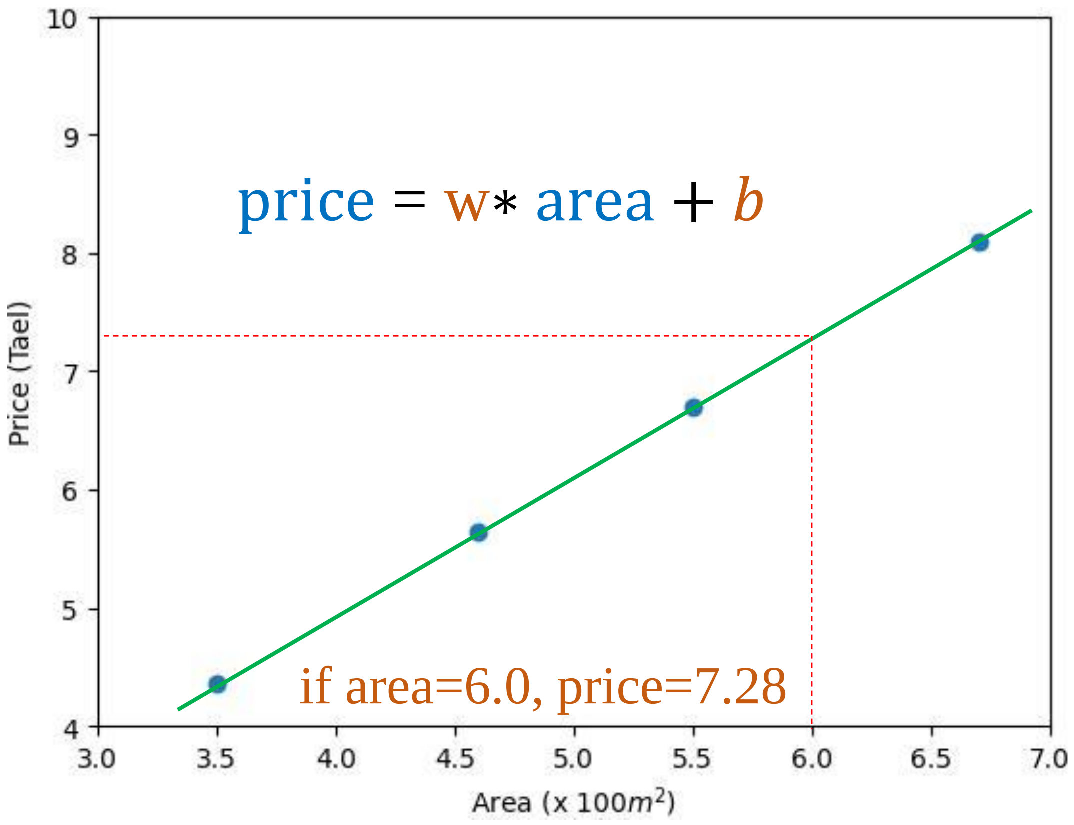
Outline

- Optimization Review
- Linear Regression Review
- Logistic Regression
- Examples
- Vectorization
- Implementation (optional)

House Price Prediction

Feature	Label
area	**price**
6.7	8.1
4.6	5.6
3.5	4.3
5.5	6.7

House price data



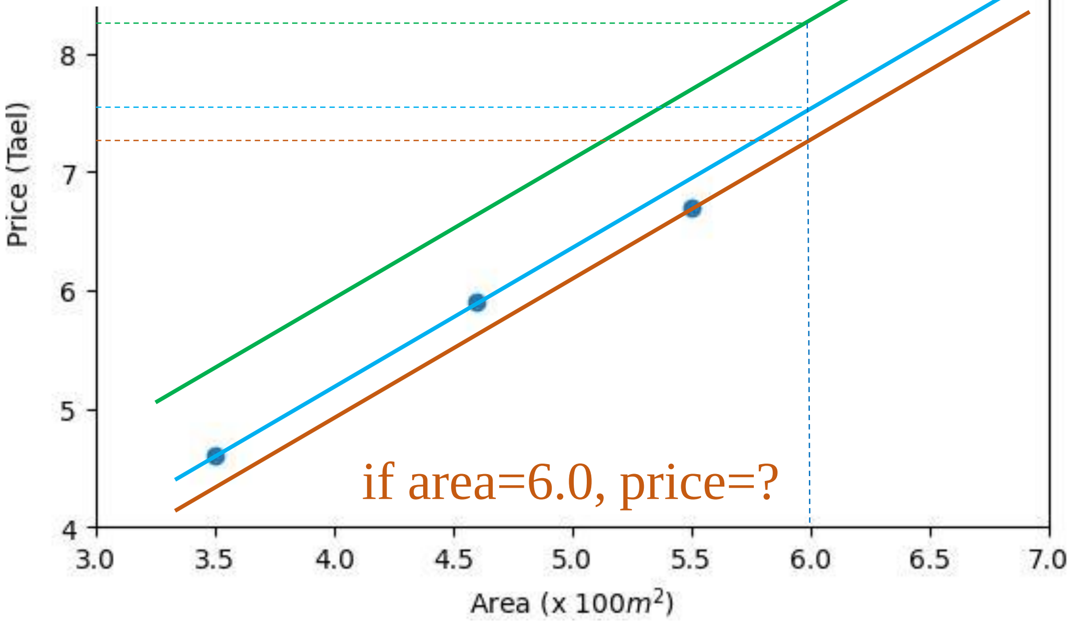
Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

House price data

$$\text{price} = w_1 * \text{area} + b_1$$

$$\text{price} = w_2 * \text{area} + b_2$$

$$\text{price} = w_3 * \text{area} + b_3$$



Linear Regression

❖ Area-based house price prediction

$$\text{predicted\_price} = w * \text{area} + b$$

$$\text{error} = (\text{predicted\_price} - \text{real\_price})^2$$

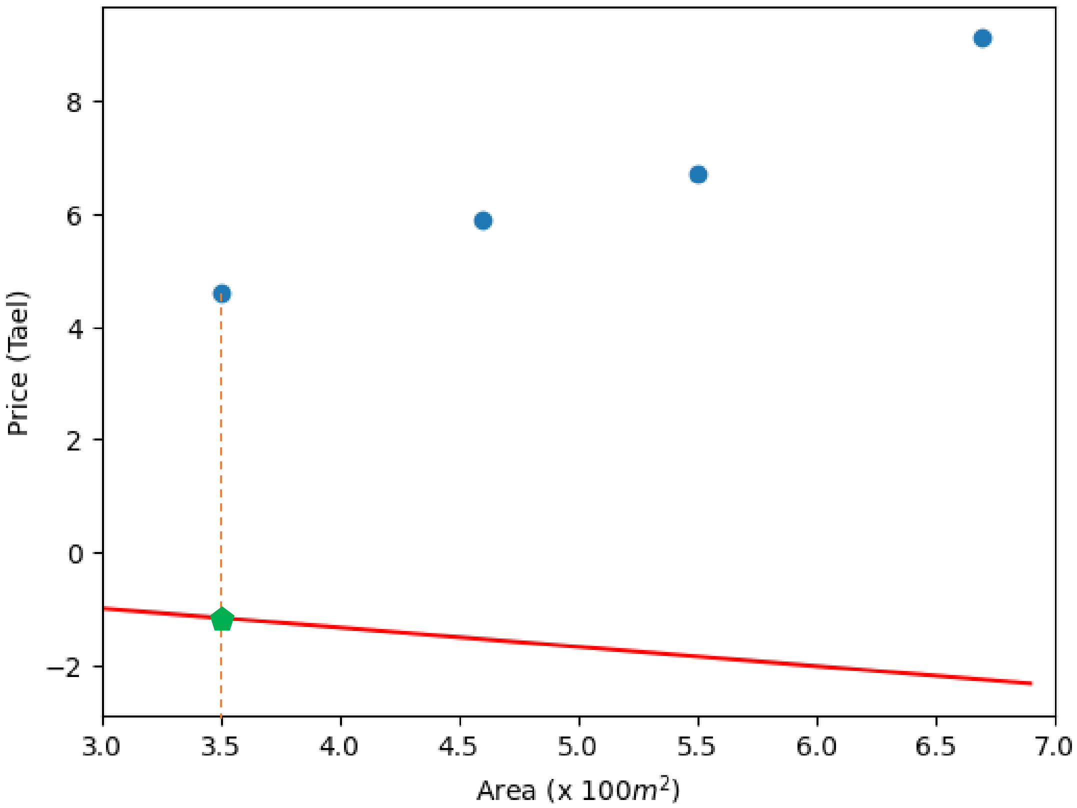
$$\hat{y} = wx + b$$

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

area	price	predicted	error
6.7	9.1	-2.238	128.55
4.6	5.9	-1.524	55.11
3.5	4.6	-1.15	33.06
5.5	6.7	-1.83	72.76

w = -0.34

b = 0.04



Linear Regression

❖ Area-based house price prediction

$$\text{predicted\_price} = w * \text{area} + b$$

$$\text{error} = (\text{predicted\_price} - \text{real\_price})^2$$

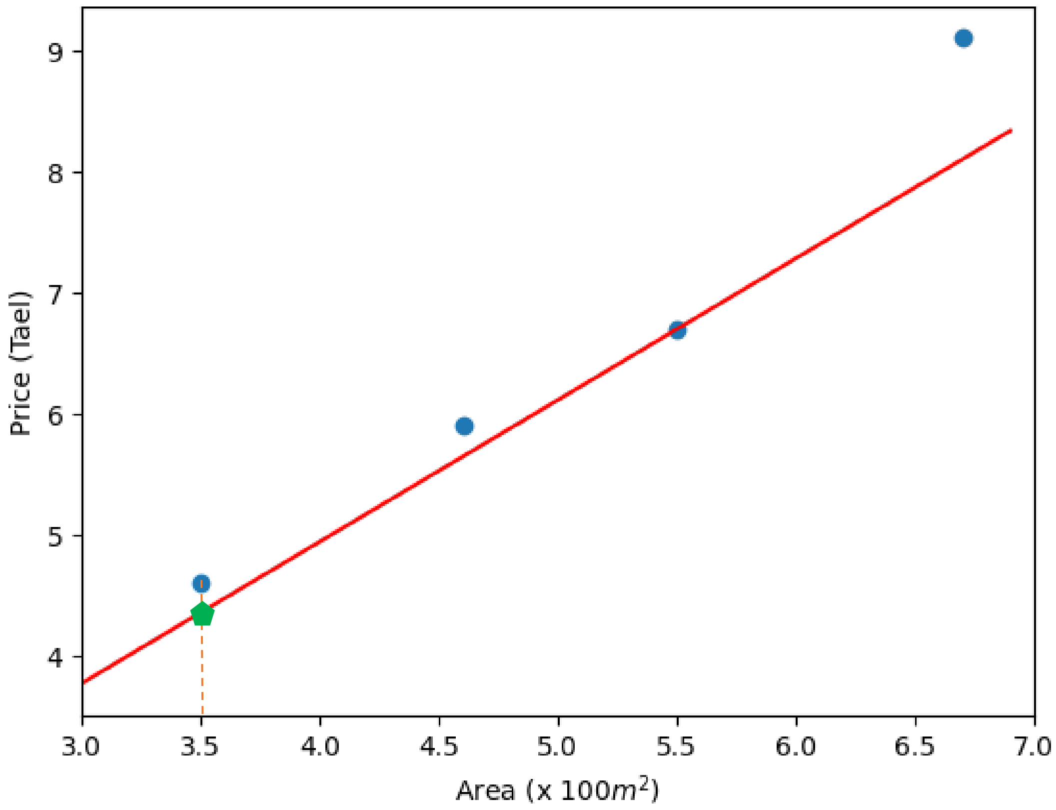
$$\hat{y} = wx + b$$

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

w = 1.17

b = 0.26

area	price	predicted	error
6.7	9.1	8.099	1.002
4.6	5.9	5.642	0.066
3.5	4.6	4.355	0.06
5.5	6.7	6.695	0.00002



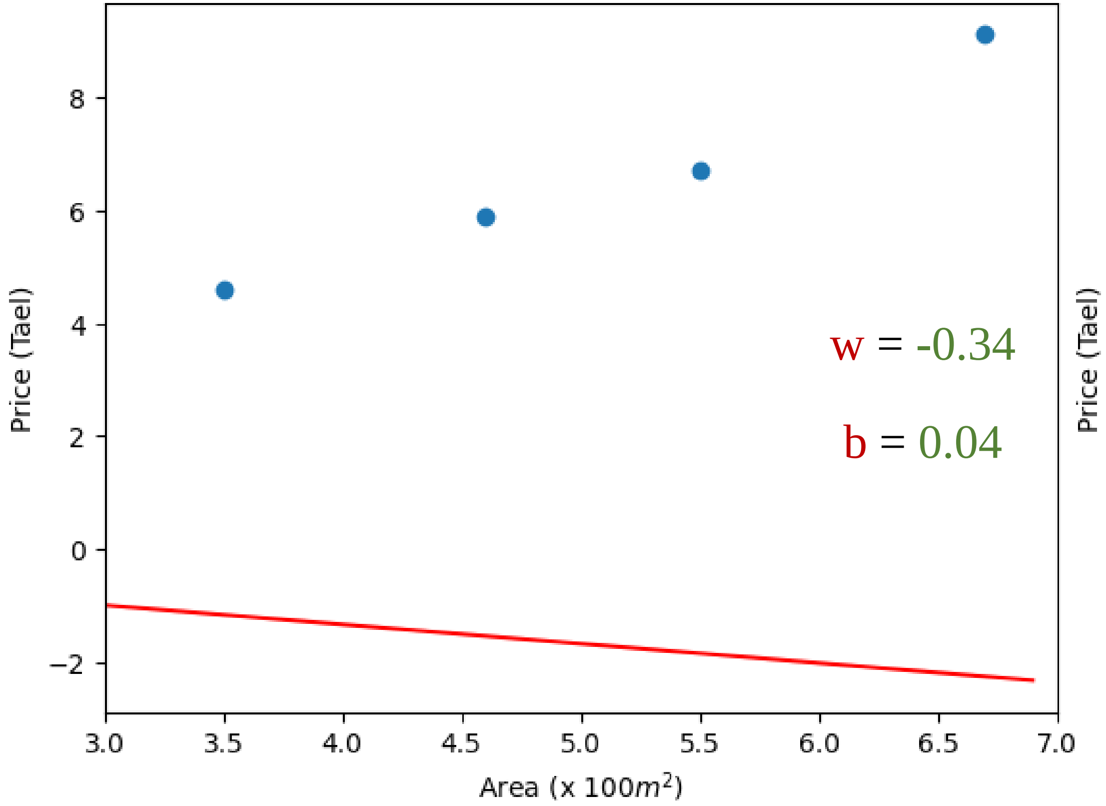
Linear Regression

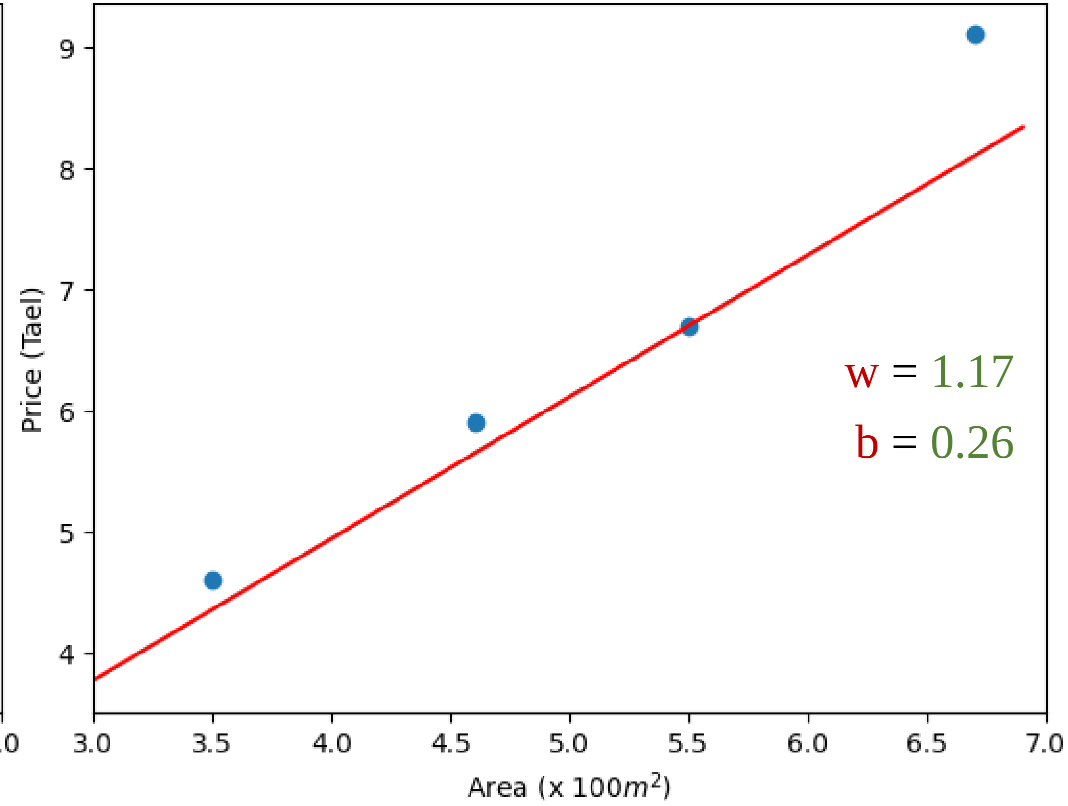
❖ **Area-based house price prediction**

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

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

How to change w and b so that $L(\hat{y}, y)$ reduces





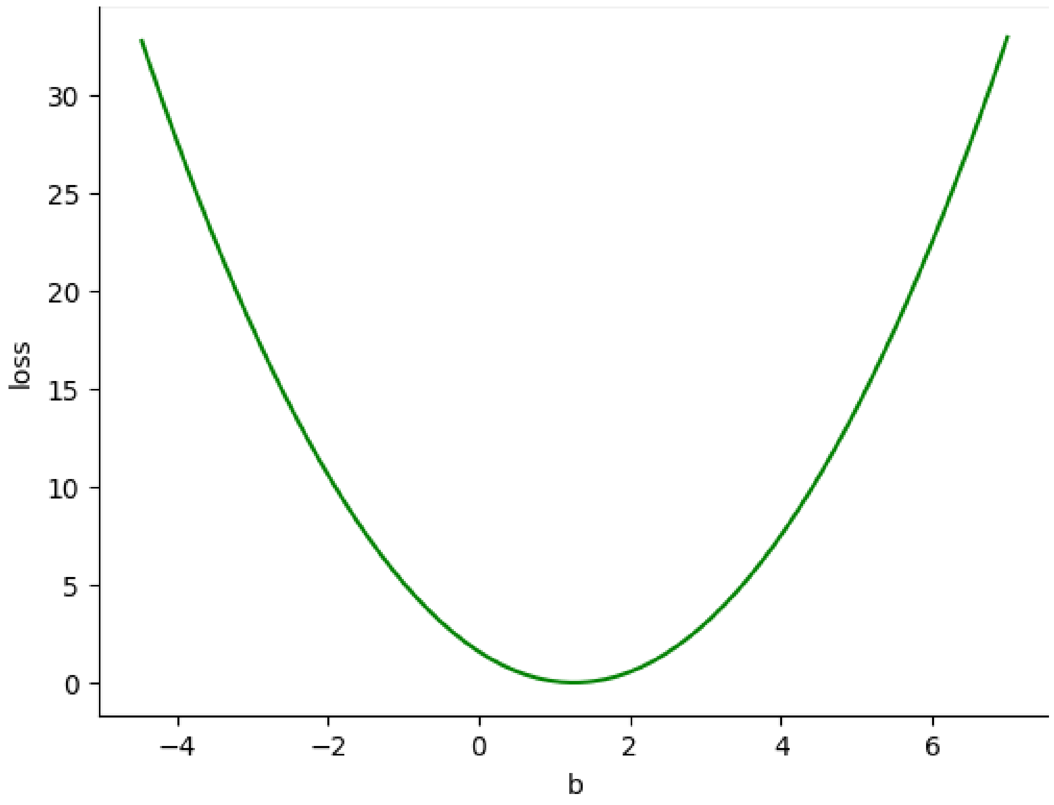
Linear Regression

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

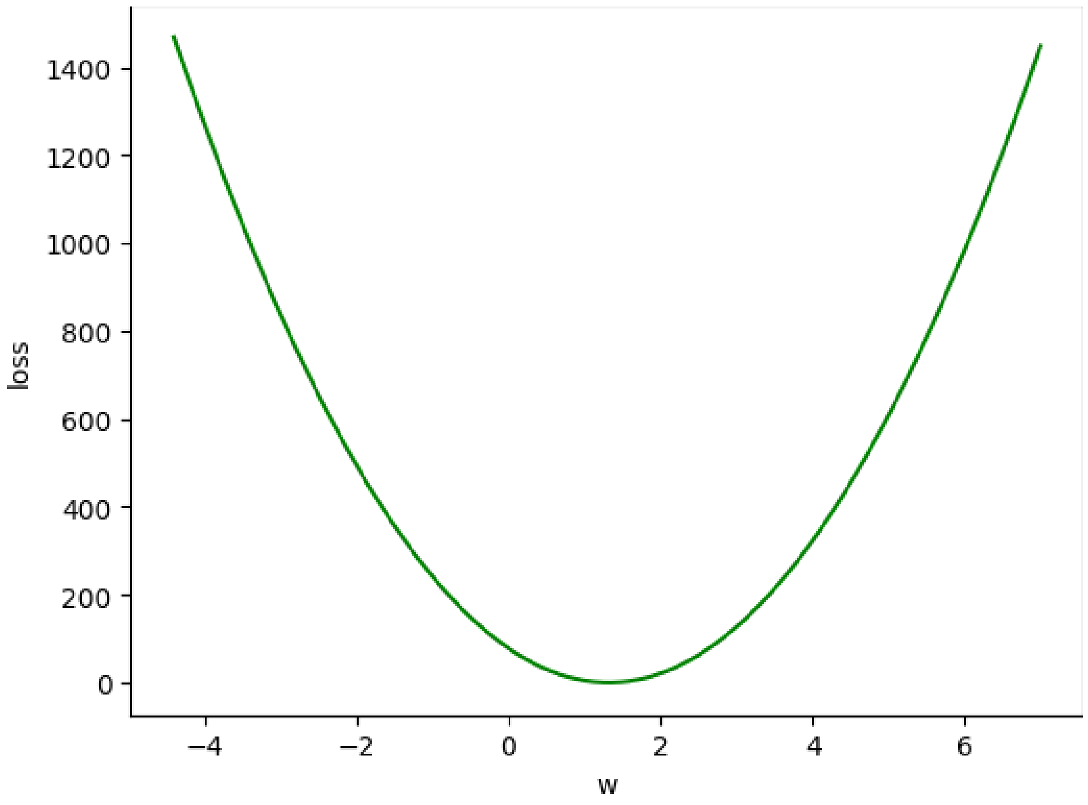
$$L(\hat{y}, y) = (\hat{y} - y)^2$$

❖ Understanding the loss function

How to change w and b so that $L(\hat{y}, y_i)$ reduces



Different b values with a fixed w value



Different w values with a fixed b value

Linear Regression

Linear equation

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

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

w and b are parameters

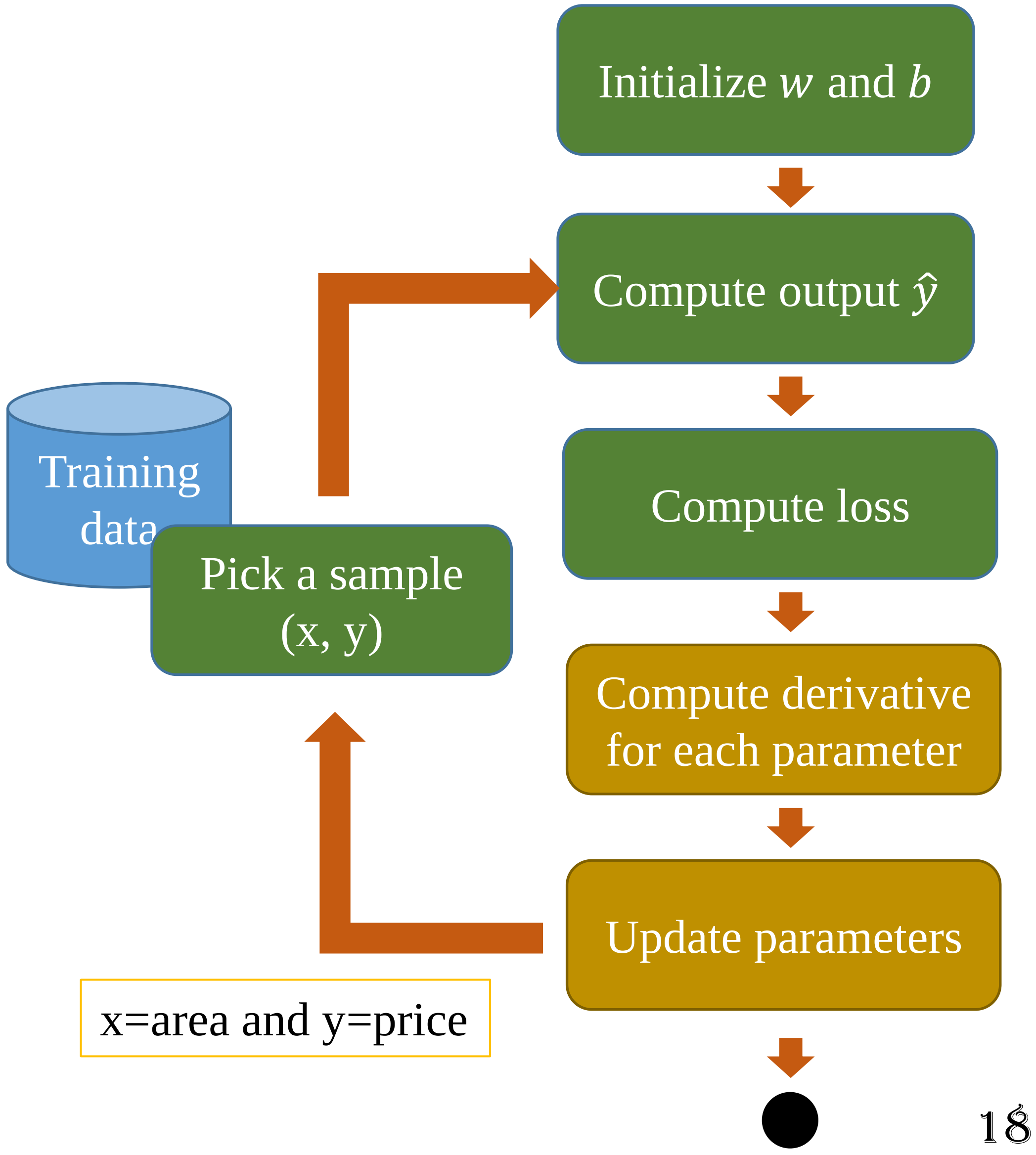
and x is input feature

Error (loss) computation

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

Squared loss

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



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

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

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

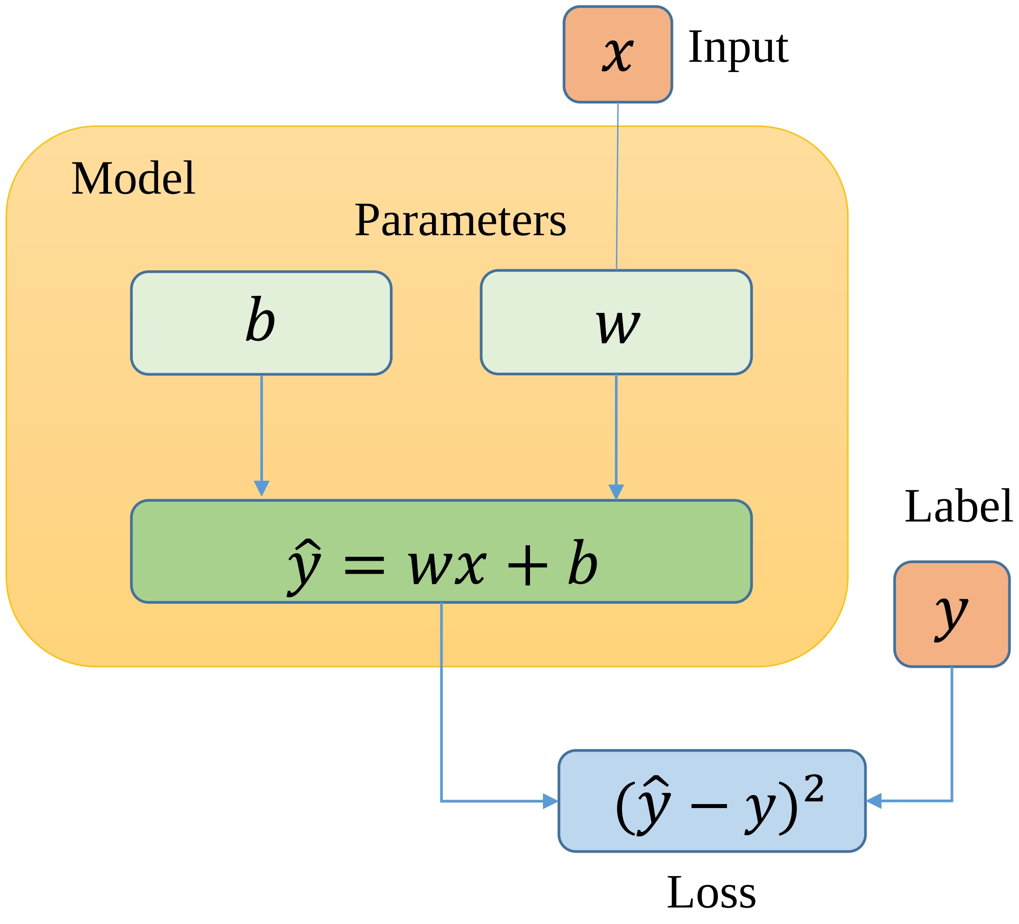
$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

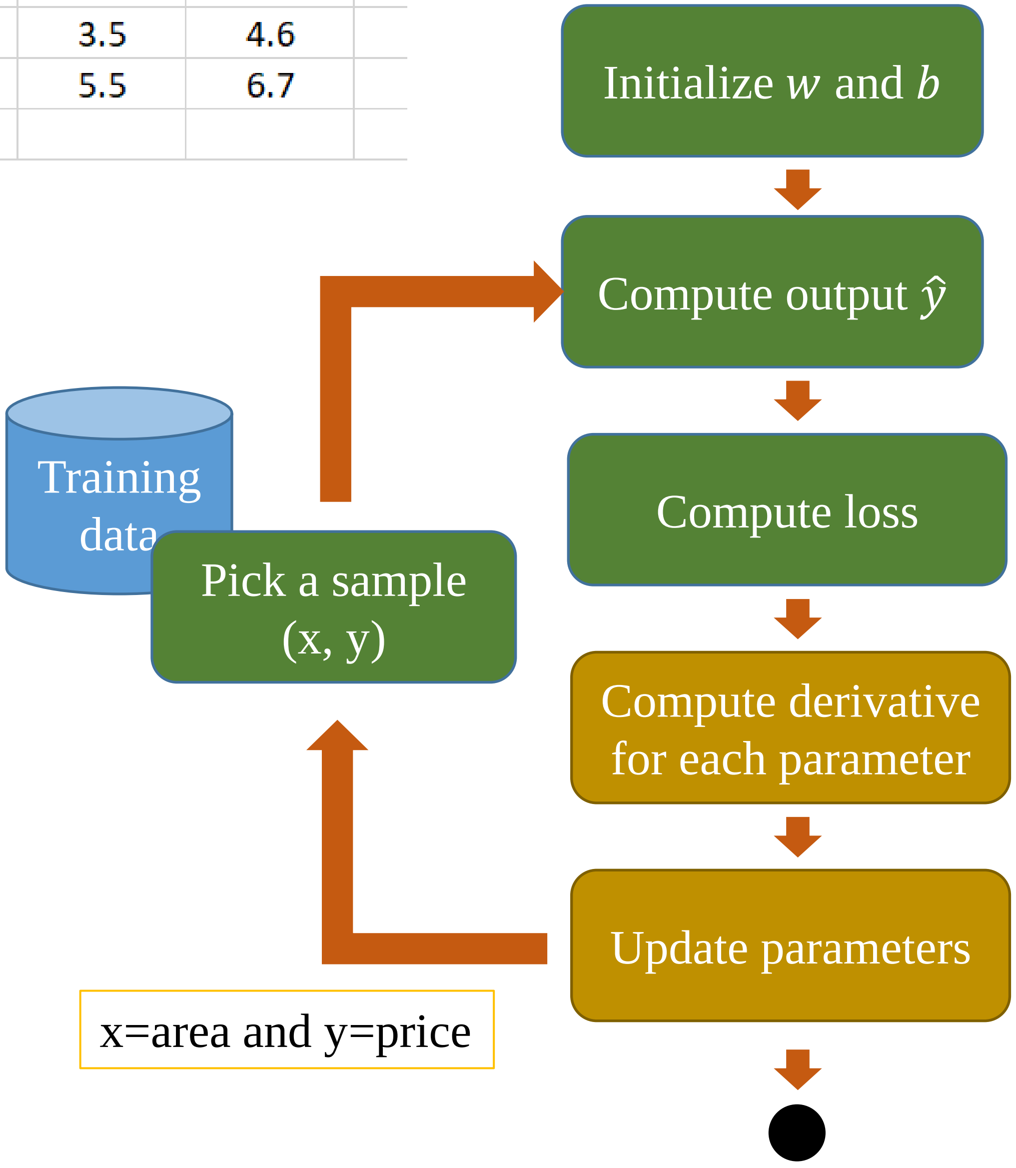
η is learning rate

Linear Regression

❖ Example

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



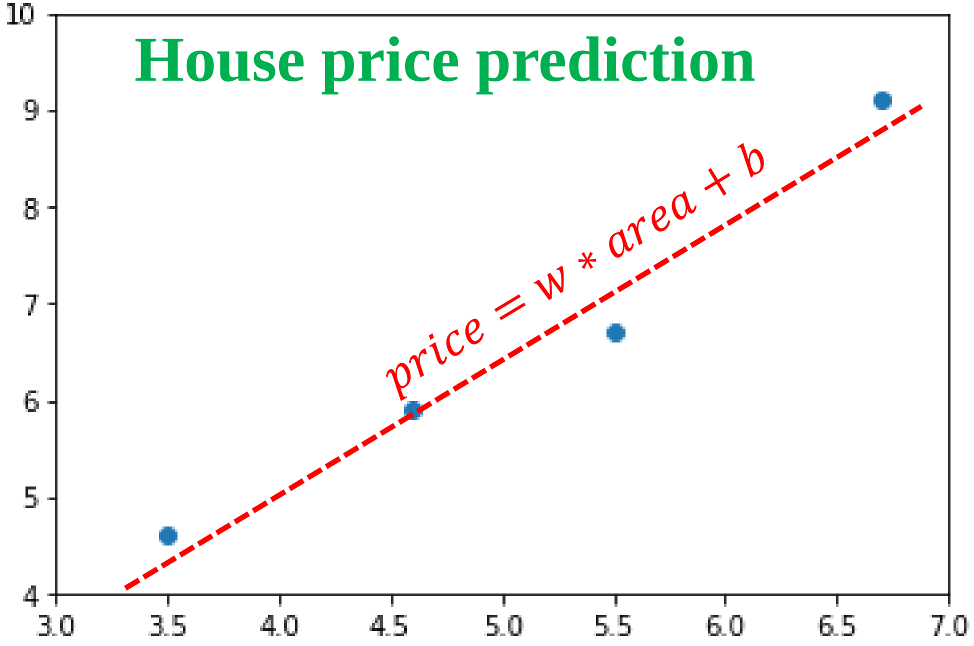


Linear Regression

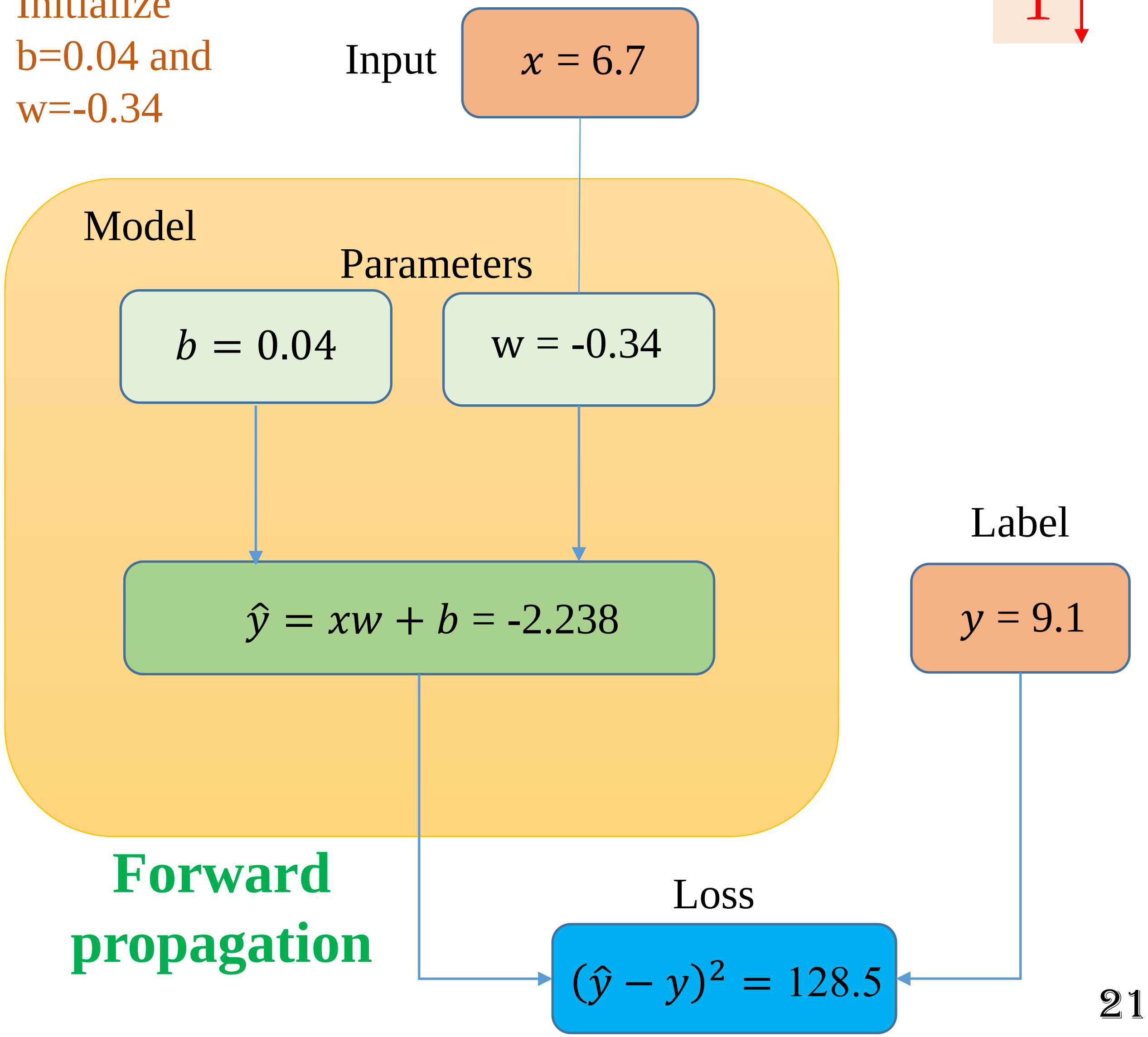
1

Given sample data

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

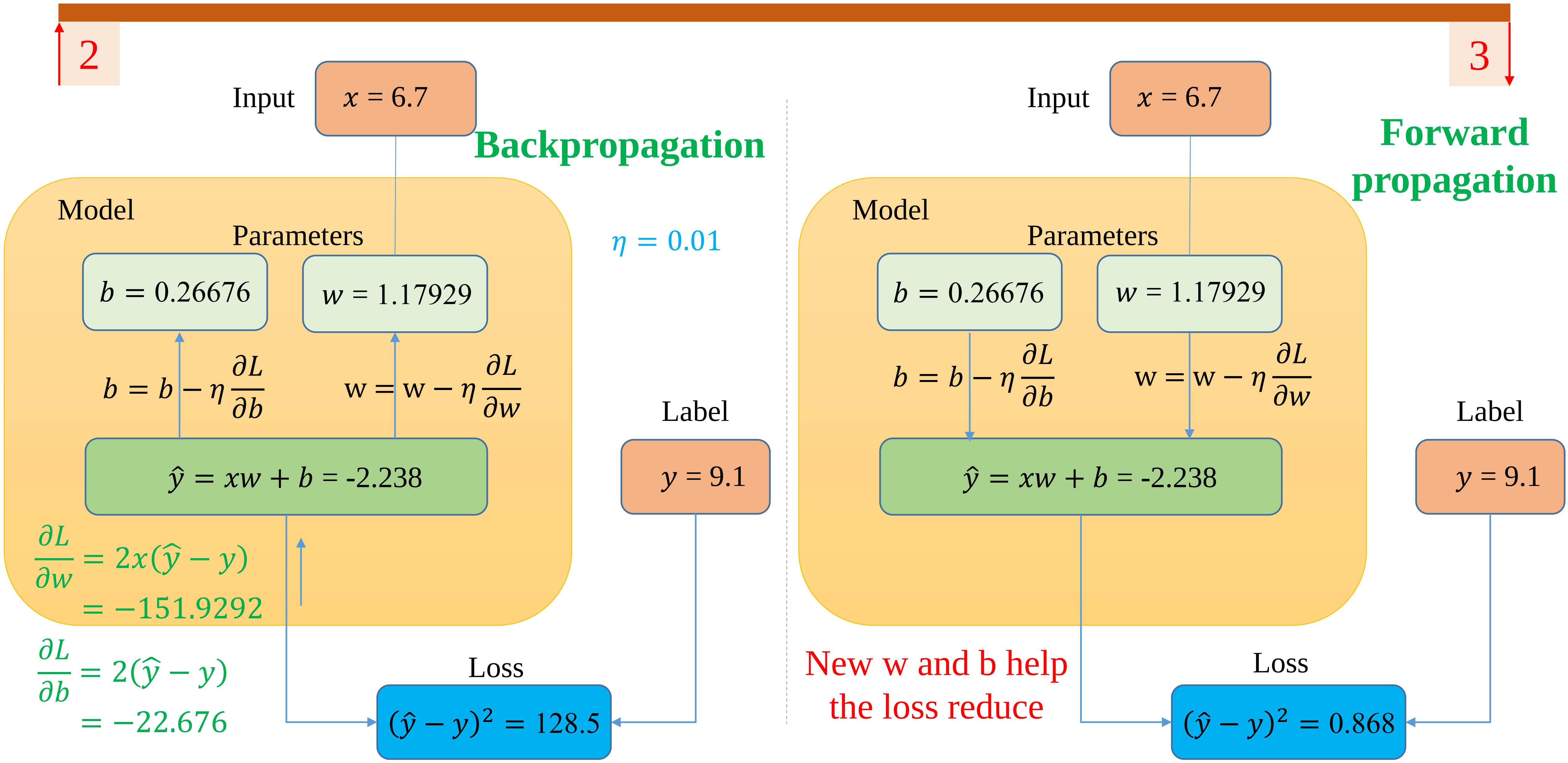


Initialize
b=0.04 and
w=-0.34



Forward propagation

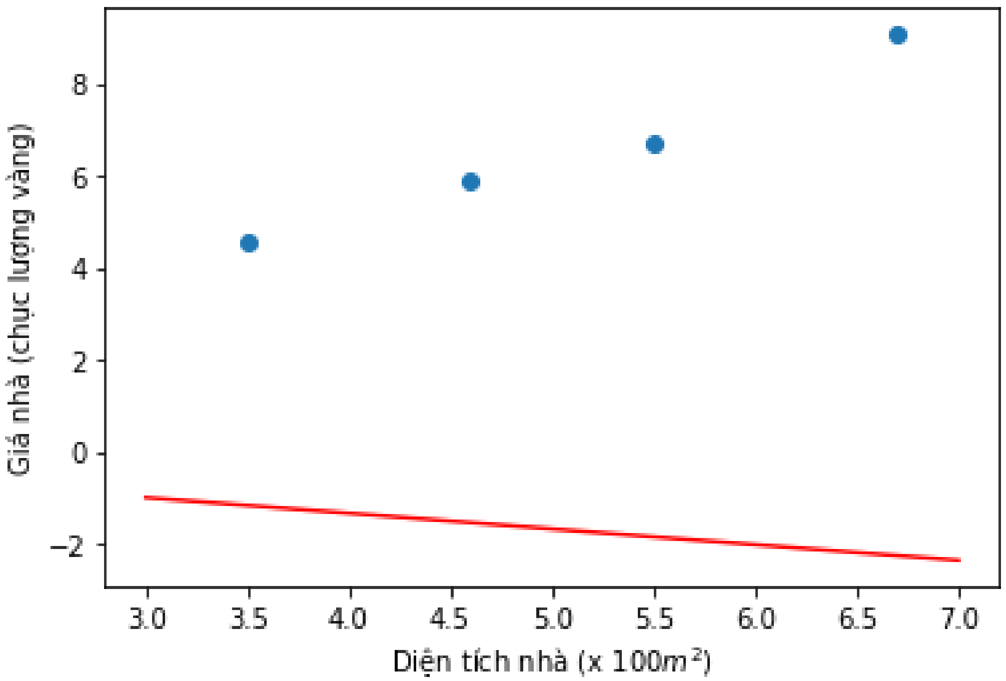
Linear Regression

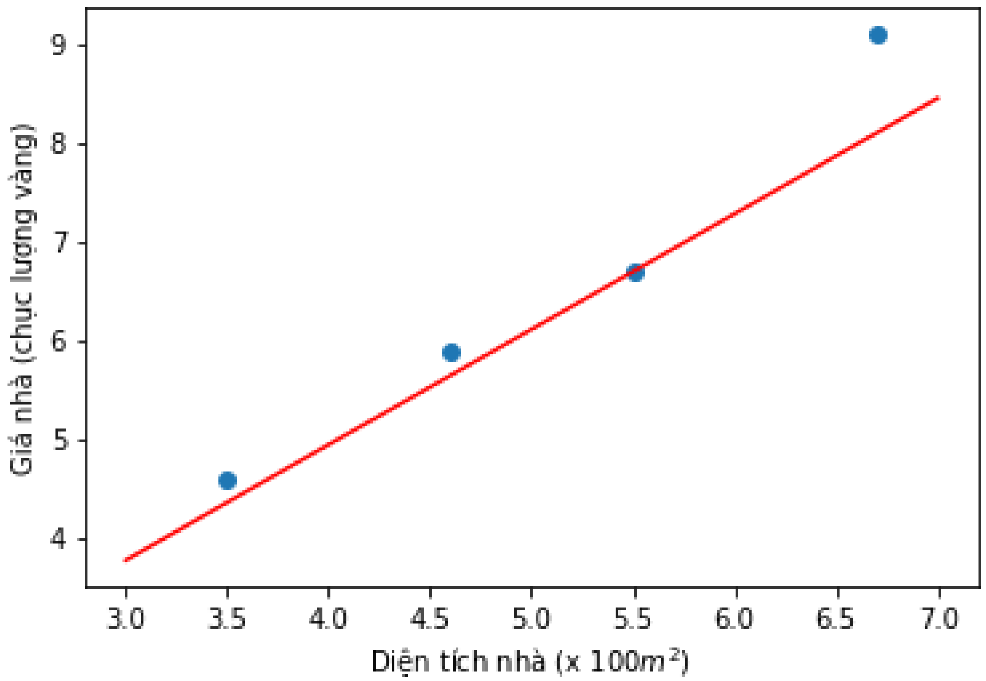


Linear Regression

❖ Toy example

Model prediction before and after the first update





w = -0.34 b = 0.04 L = 128.55

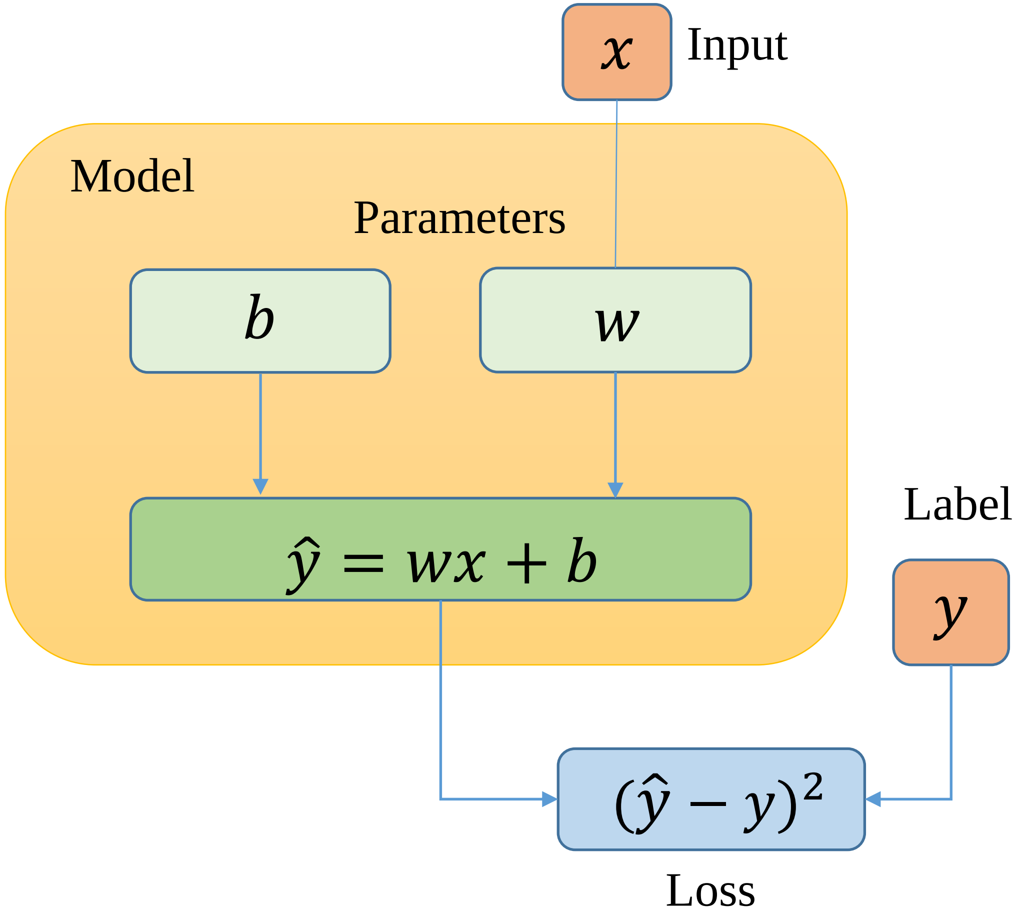
w = 1.179292 b = 0.26676 L = 0.868

Before updating

After updating

Linear Regression

❖ Summary (one feature and one sample)



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

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

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

3) Compute loss

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

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

Outline

- Optimization Review
- Linear Regression Review
- Logistic Regression
- Examples
- Vectorization
- Implementation (optional)

AI VIETNAM
All-in-One Course

Idea of Logistic Regression

❖ Linear regression

Area-based House Price Data

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

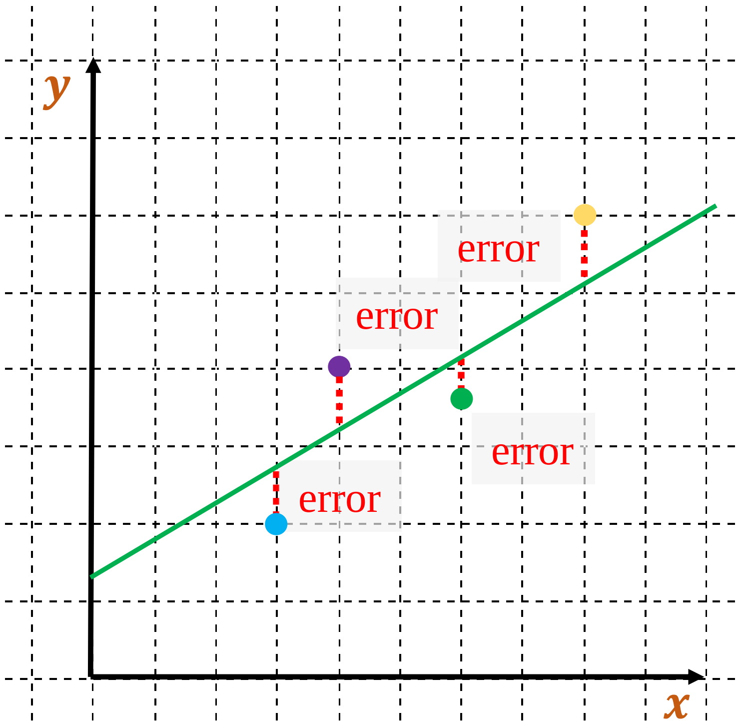
Training data

construct →

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

$$\hat{y} \in (-\infty \quad +\infty)$$

Model



Find the line $\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x}$ that is best fitting to given data, then use $\hat{y}$ to predict for new data

Idea of Logistic Regression

❖ Given a new kind of data

Petal_Length	Category
1.4	Flower A
1	Flower A
1.5	Flower A
3	Flower B
3.8	Flower B
4.1	Flower B

Feature: Petal_Length — Label: Category

Flower A: Category 0

Flower B: Category 1

Assign numbers to categories

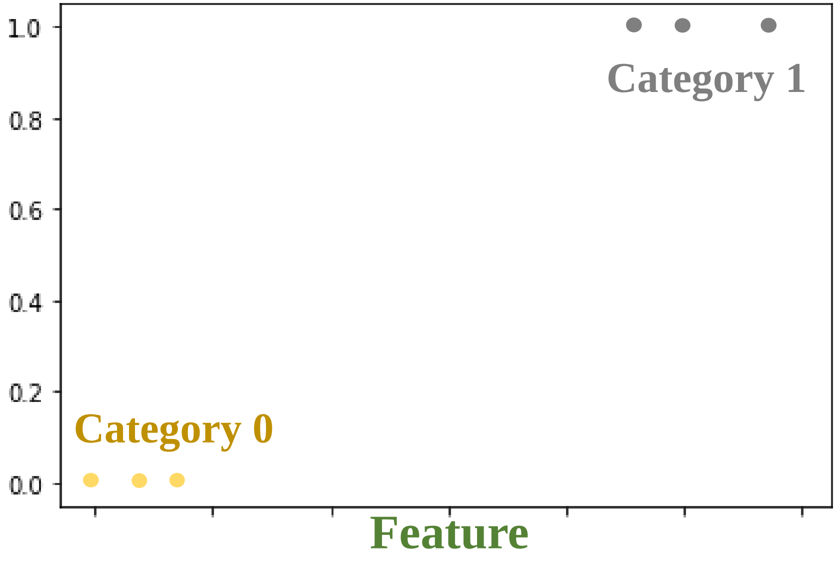
Petal_Length	Category
1.4	0
1	0
1.5	0
3	1
3.8	1
4.1	1

Feature: Petal_Length — Label: Category

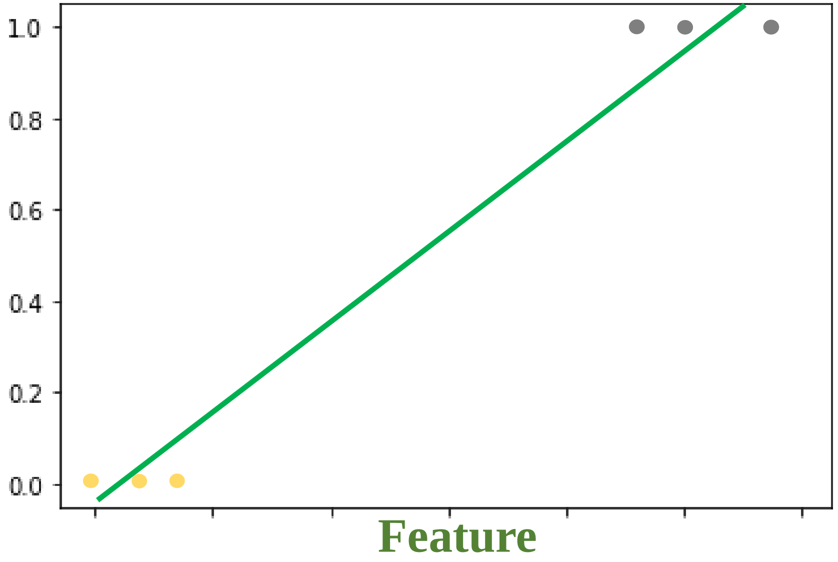
0: Category 0

1: Category 1

Plot data



A line is not suitable for this data



Idea of Logistic Regression

Sigmoid function

$$\sigma(u) = \frac{1}{1 + e^{-z}}$$

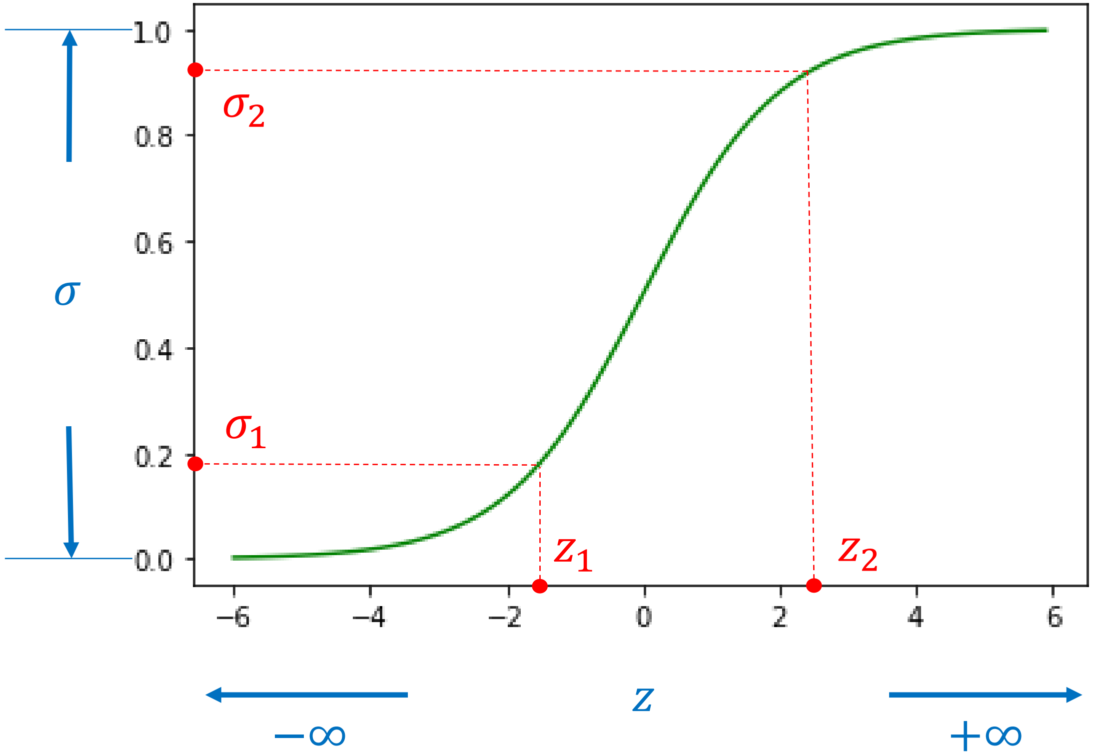
$$z \in (-\infty \quad +\infty)$$

$$\sigma(u) \in (0 \quad 1)$$

Property

$$\forall z_1 z_2 \in [a \;\; b] \text{ and } z_1 \leq z_2$$

$$\rightarrow \sigma(z_1) \leq z(u_1)$$



Idea of Logistic Regression

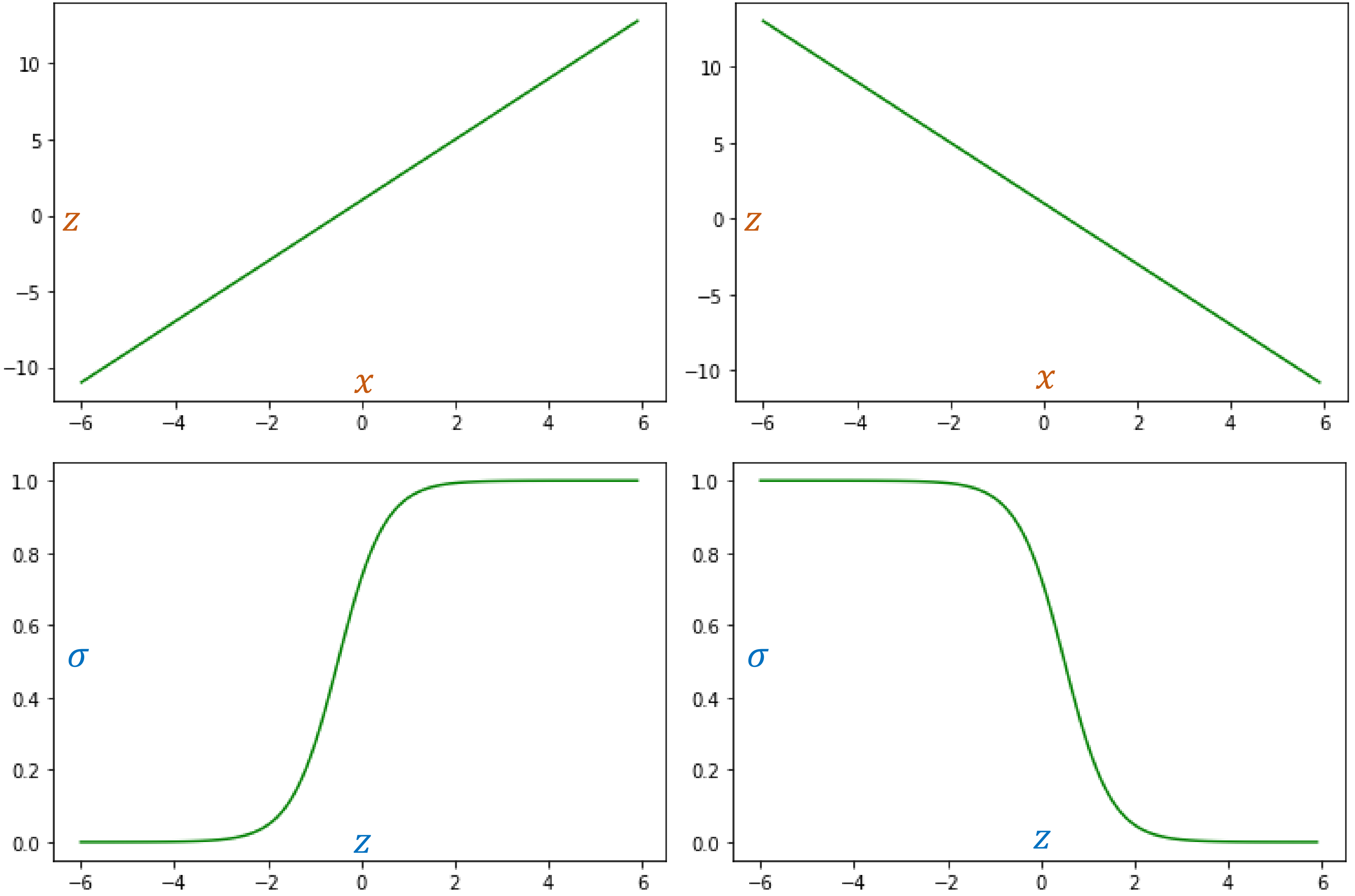
$$z = wx + b$$

$$z \in (-\infty \quad +\infty\,)$$

$$z = wx + b$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\sigma(z) \in (0 \quad 1\,)$$



Idea of Logistic Regression

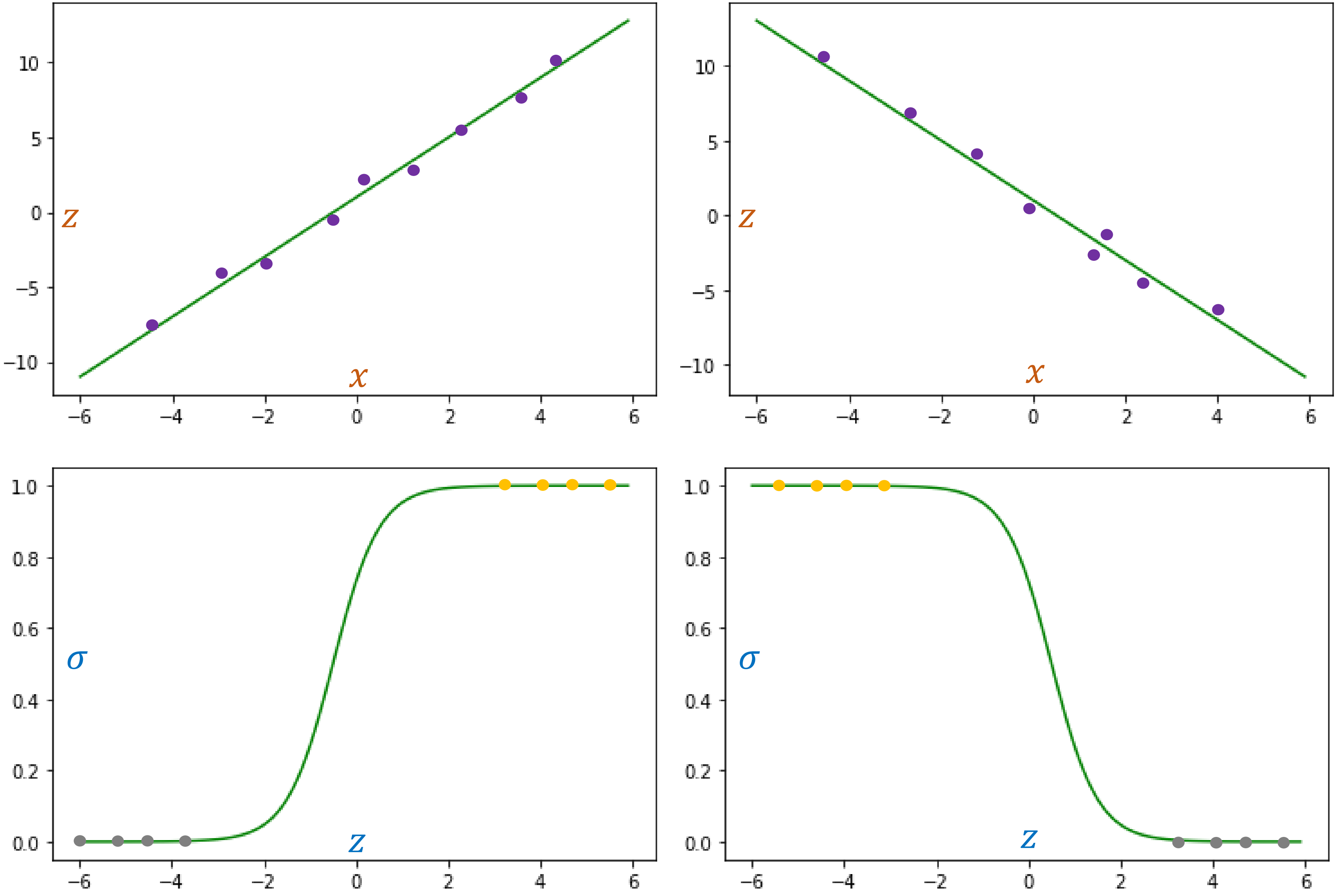
$z = wx + b$

$z \in (-\infty \quad +\infty)$

$z = wx + b$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$\sigma(z) \in (0 \quad 1)$

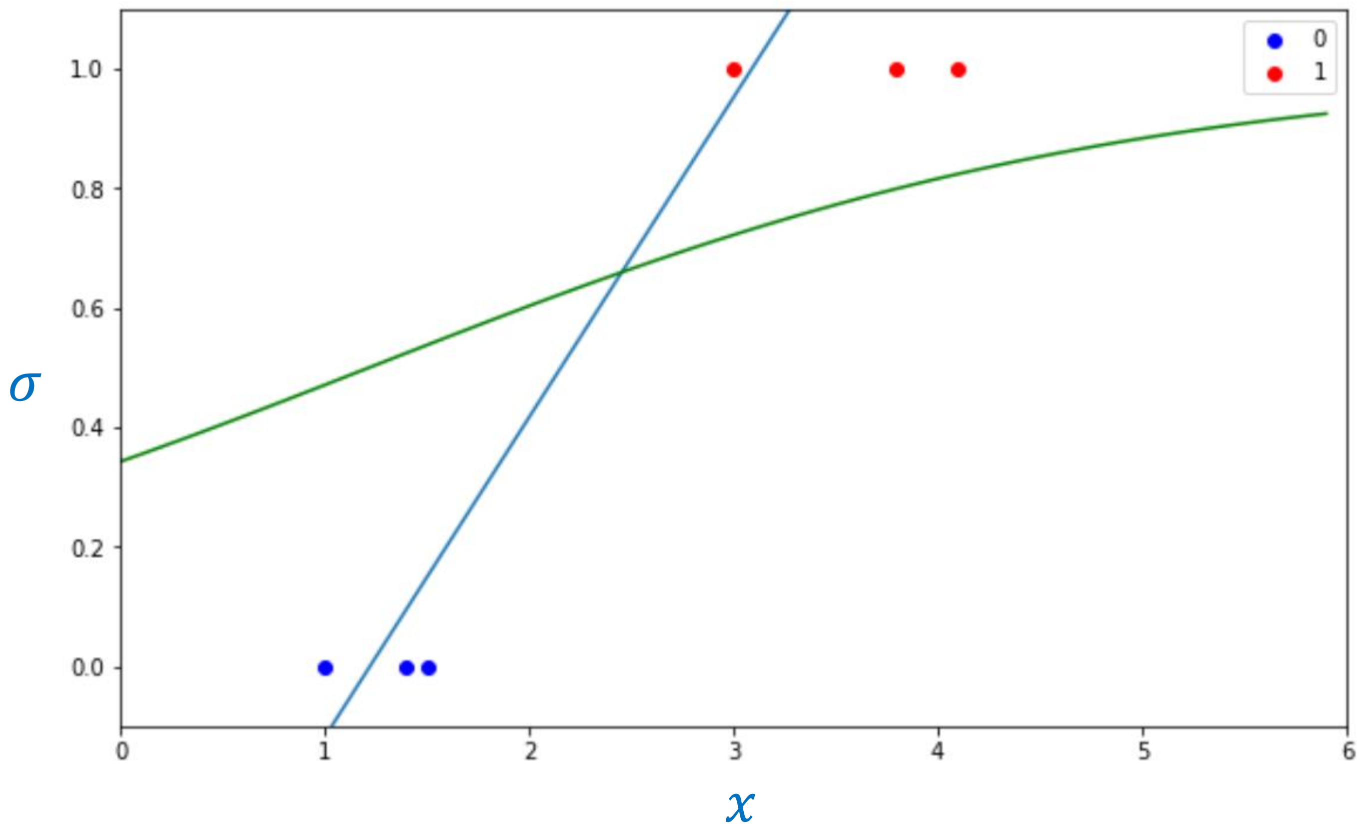


Idea of Logistic Regression

Petal_Length (Feature)	Category (Label)
1.4	0
1	0
1.5	0
3	1
3.8	1
4.1	1

Category 0

Category 1



z	$\sigma(z)$
0.095	0.52
-0.119	0.47
0.1485	0.53
0.951	0.72
1.379	0.79
1.5395	0.82

$$z = wx + b$$

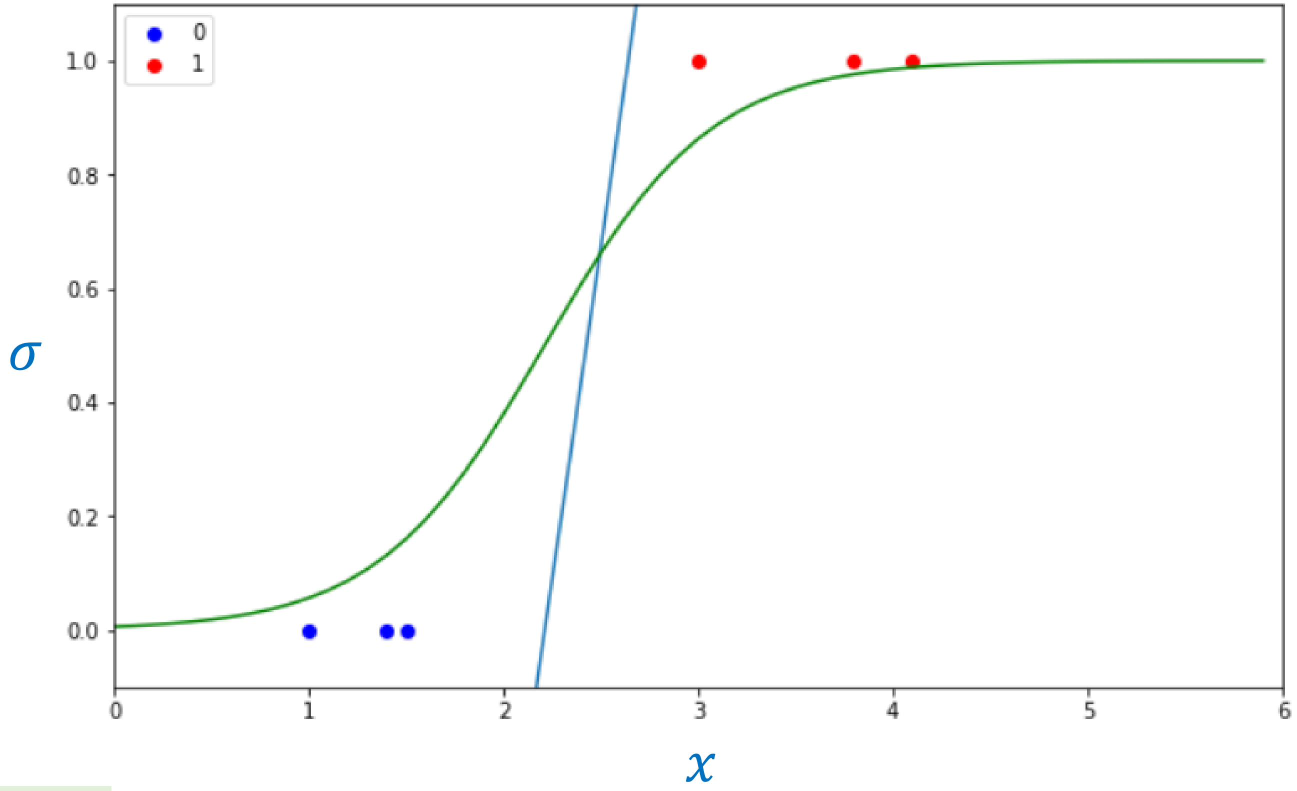
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\sigma(z) \in (0 \quad 1)$$

$$z = 0.535 * x - 0.654$$

Idea of Logistic Regression

Feature	Label	
Petal_Length	Category	
1.4	0	Category 0
1	0	
1.5	0	
3	1	Category 1
3.8	1	
4.1	1	



z	$\sigma(z)$
-1.89	0.1309
-2.82	0.0559
-1.65	0.1598
1.837	0.8625
3.701	0.9759
4.401	0.9878

$$z = wx + b$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\sigma(z) \in (0 \quad 1)$$

$$z = 2.331 * x - 5.156$$

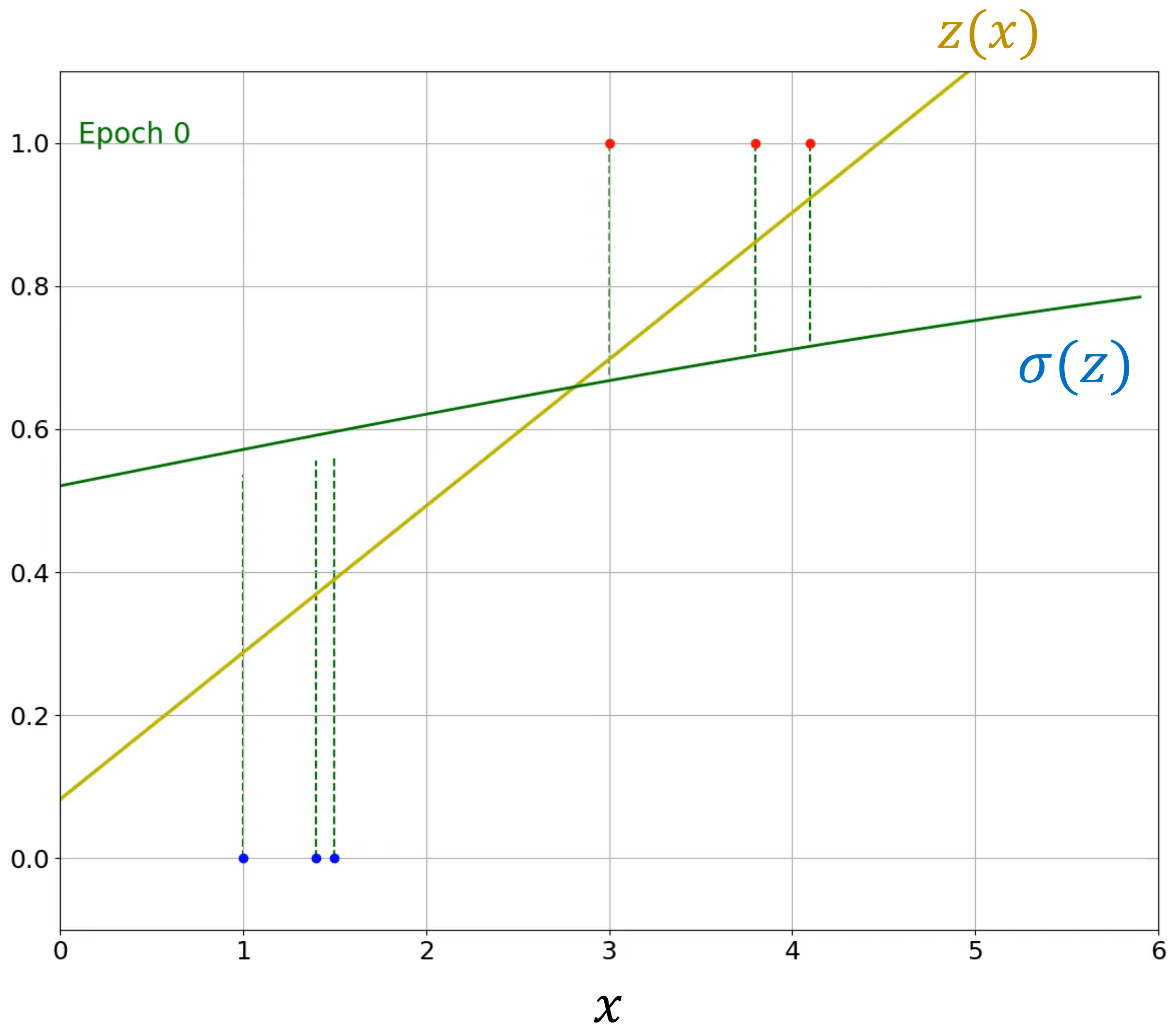
Idea of Logistic Regression

Feature	Label	
Petal_Length	Category	
1.4	0	Category 0
1	0	
1.5	0	
3	1	Category 1
3.8	1	
4.1	1	

$$z = wx + b$$

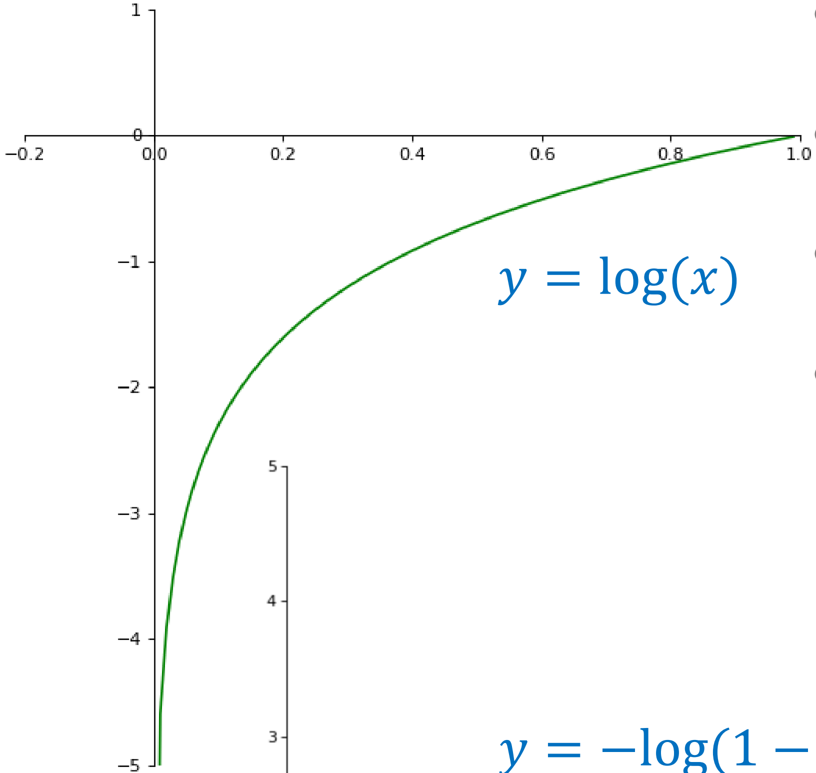
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

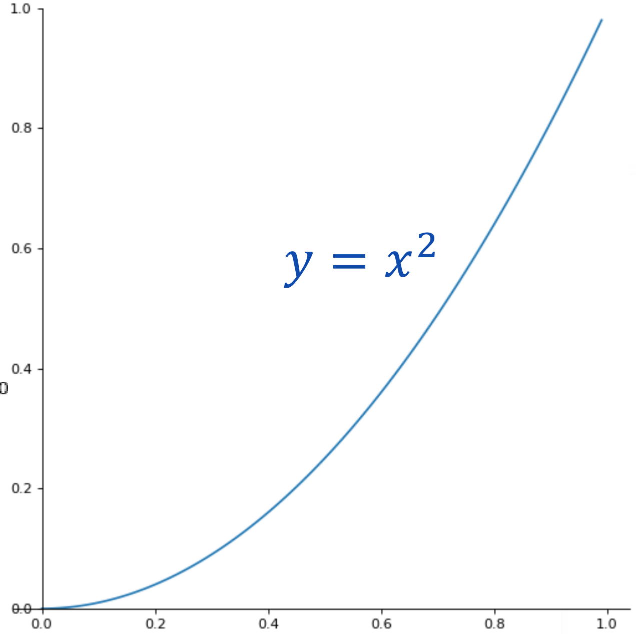
$$\sigma(z) \in (0 \quad 1)$$

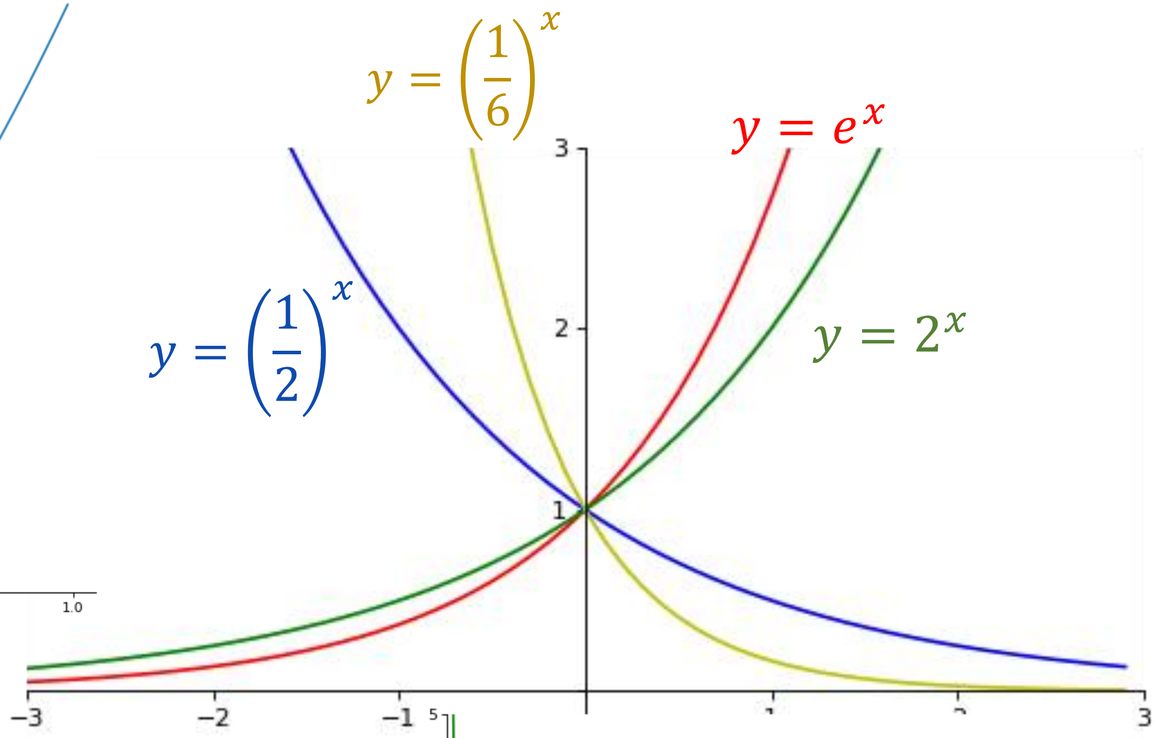


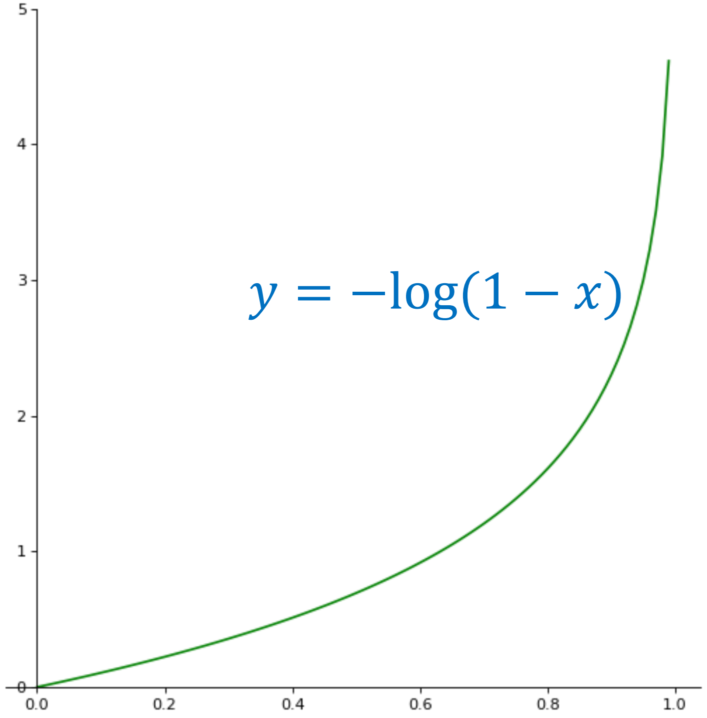
How to evaluate the performance of a model?

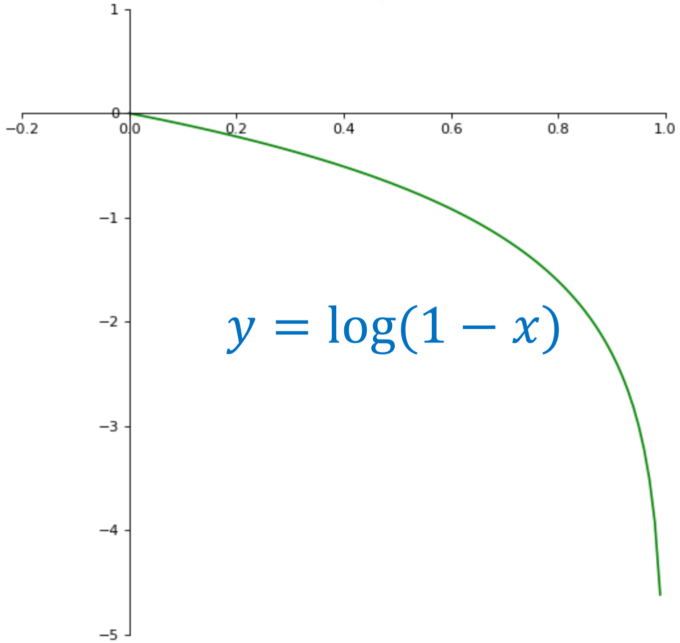
❖ Suggested Functions

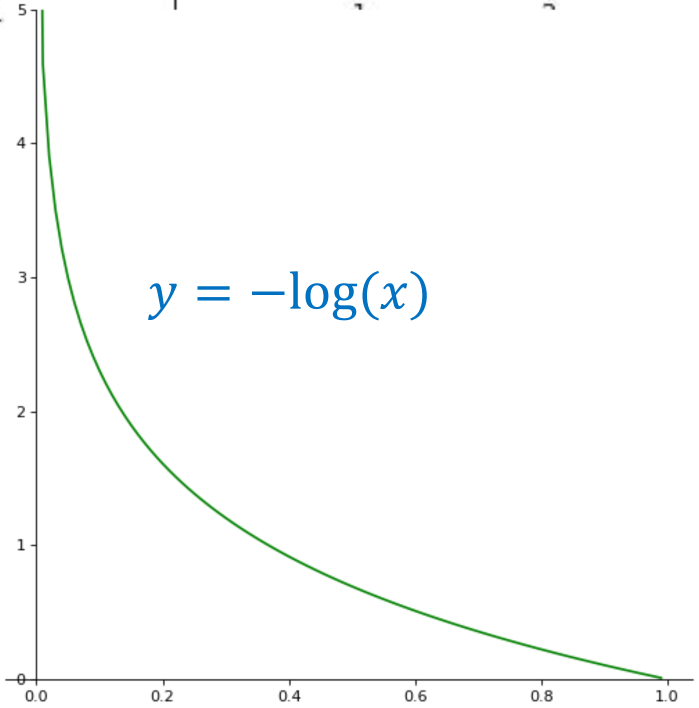












Idea of Logistic Regression

❖ Loss function

Feature	Label
Petal_Length	Category
1.4	0
1	0
1.5	0
3	1
3.8	1
4.1	1

Category 0

Category 1

$$z = wx + b$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\sigma(z) \in (0 \quad 1)$$

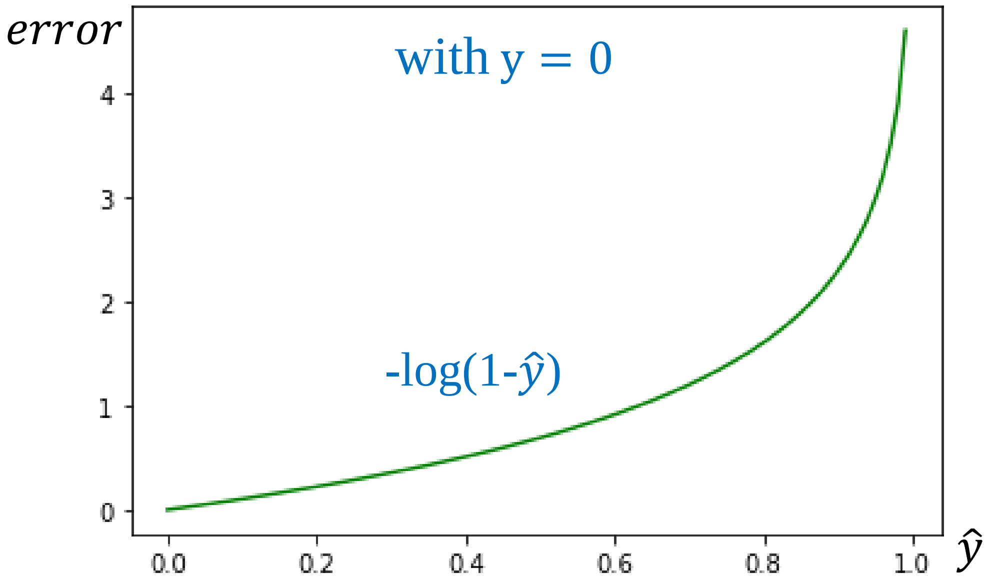
if y = 1

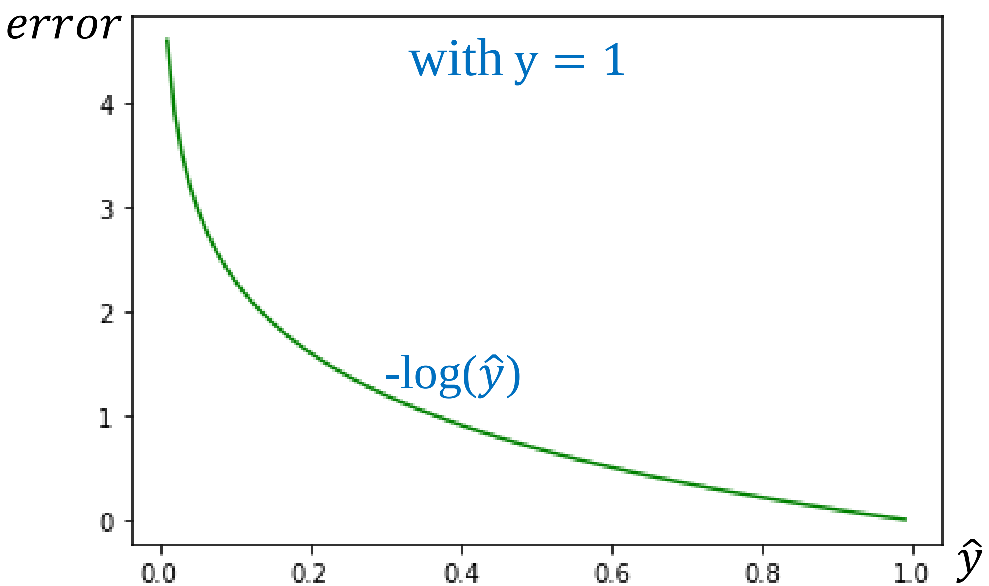
$$L(\hat{y}) = -\log(\hat{y})$$

if y = 0

$$L(\hat{y}) = -\log(1 - \hat{y})$$

How to remove if?





Idea of Logistic Regression

❖ Loss function

Feature | Output | Label

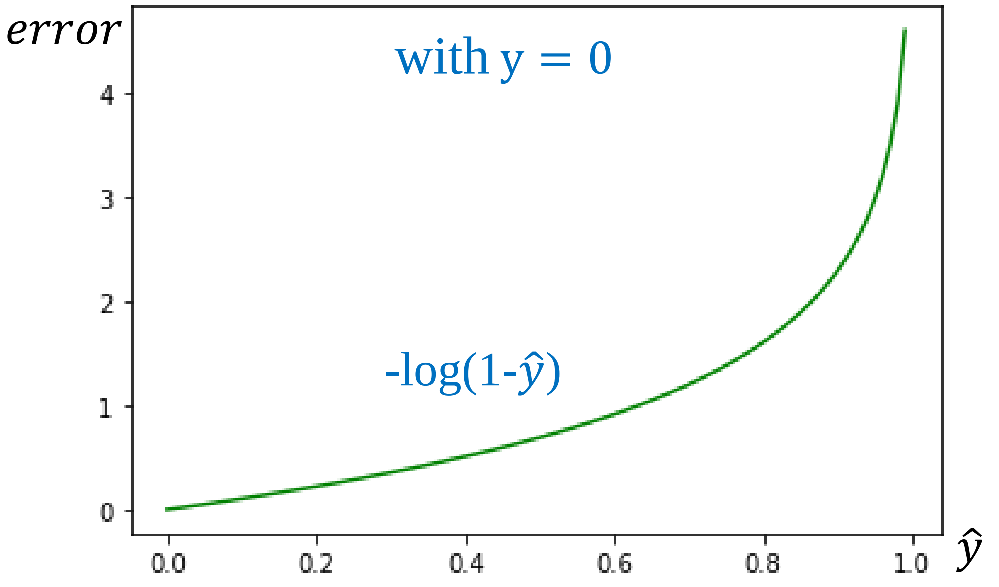
Input	Output	Label
…	0.3	0
…	0.8	0
…	0.7	0
…	0.4	0
…	0.6	1
…	0.8	1
…	0.9	1
…	0.2	1

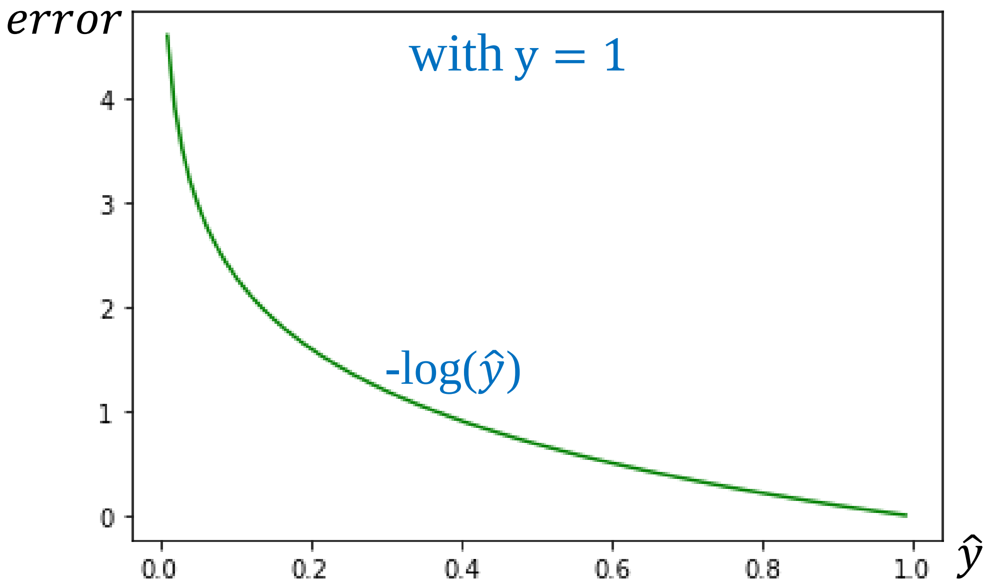
if $y = 0$

$$L(\hat{y}) = -\log(1-\hat{y})$$

if $y = 1$

$$L(\hat{y}) = -\log(\hat{y})$$





Binary cross-entropy

$$L(y,\hat{y}) = -y\log\hat{y} - (1-y)\log(1-\hat{y})$$

Introduce the loss function in another way

Idea of Logistic Regression

❖ Given a new kind of data

Feature | **Label**

Petal_Length	Category	
1.4	Flower A	Category 0
1	Flower A	Category 0
1.5	Flower A	Category 0
3	Flower B	Category 1
3.8	Flower B	Category 1
4.1	Flower B	Category 1

Assign numbers to categories

Feature | **Label**

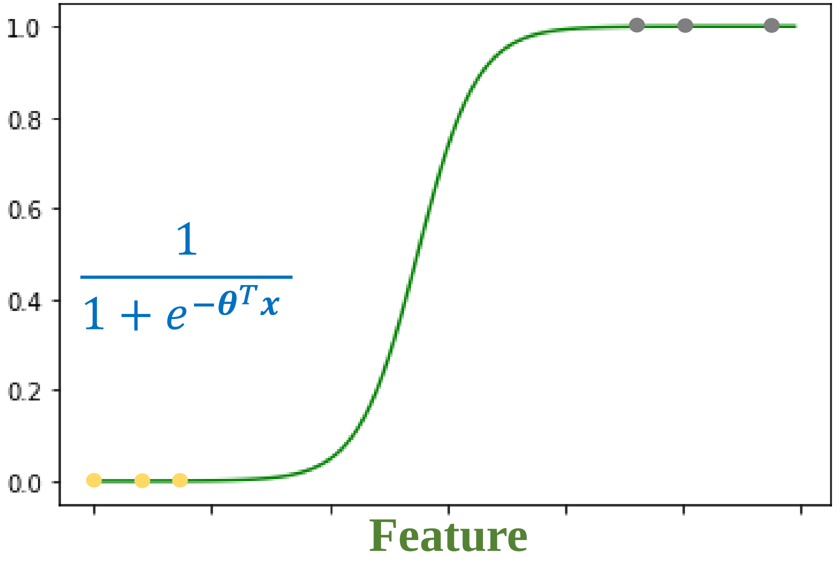
Petal_Length	Category	
1.4	0	Category 0
1	0	Category 0
1.5	0	Category 0
3	1	Category 1
3.8	1	Category 1
4.1	1	Category 1

Sigmoid function could fit the data

$$z = wx + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\hat{y} \in (0 \quad 1)$$



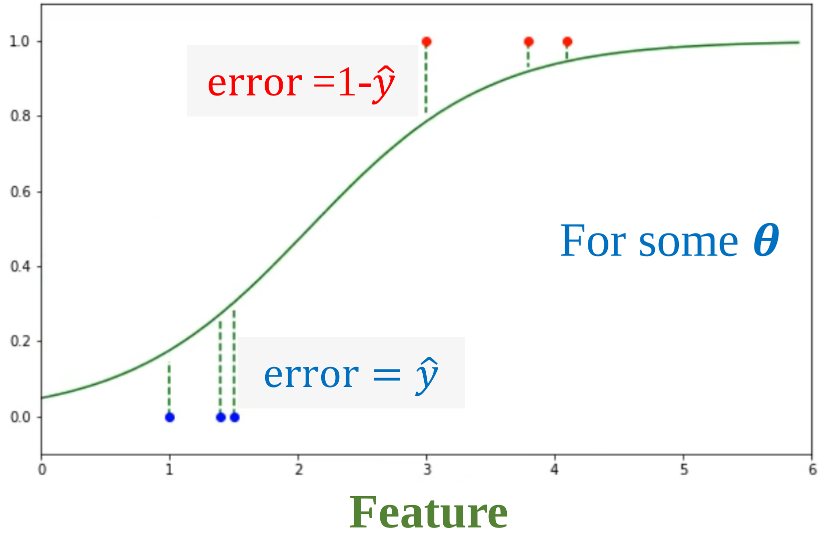
Error

if $y = 1$

$\quad$ error $= 1 - \hat{y}$

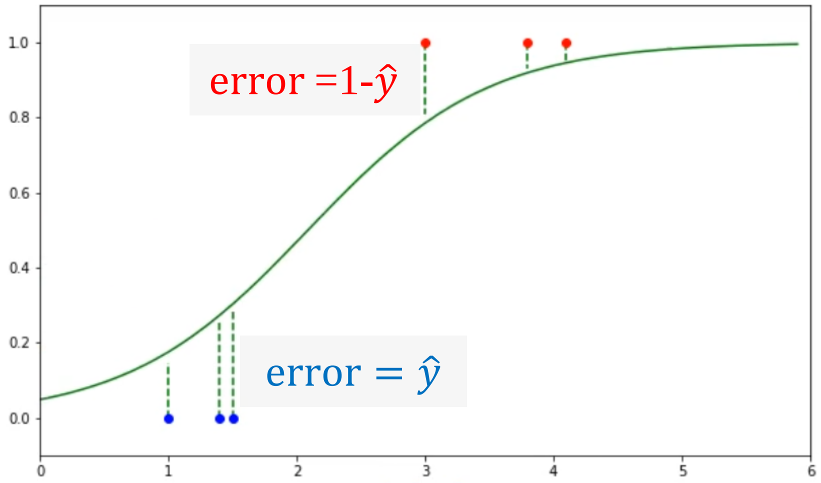
if $y = 0$

$\quad$ error $= \hat{y}$



Idea of Logistic Regression

❖ Construct loss



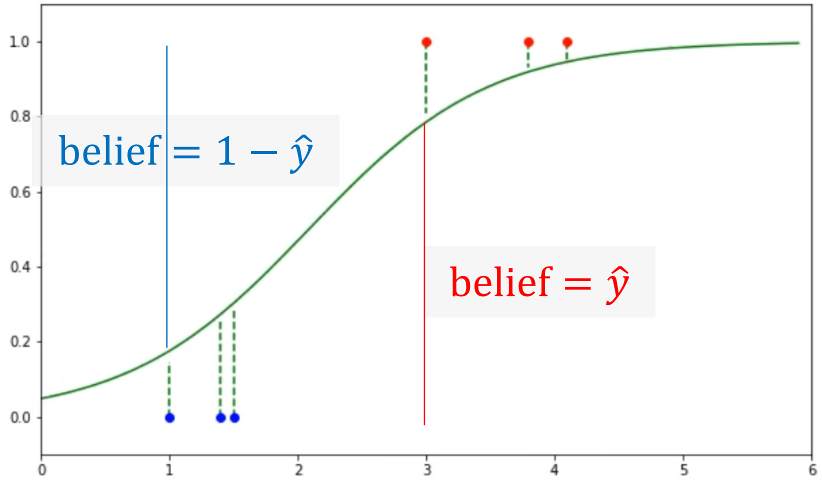
Error

if $y = 1$

$\quad$ error $= 1 - \hat{y}$

if $y = 0$

$\quad$ error $= \hat{y}$



Belief

if $y = 1$

$\quad$ belief $= \hat{y}$

if $y = 0$

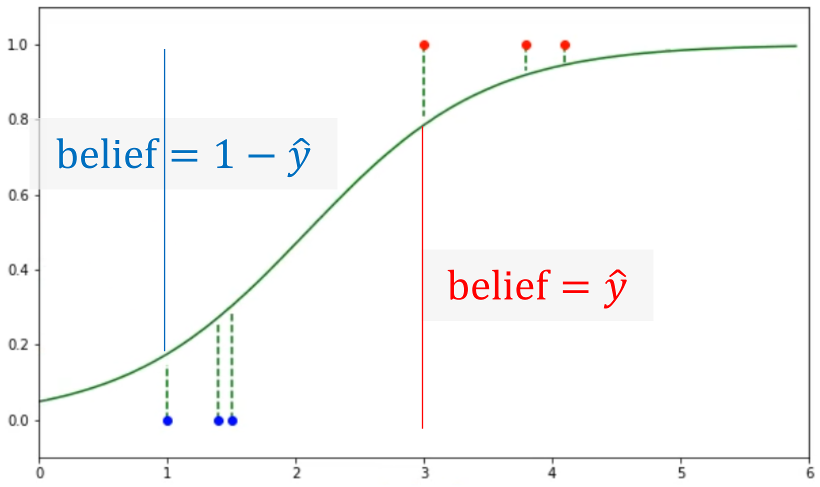
$\quad$ belief $= 1 - \hat{y}$

$$P = \hat{y}^{\,y}(1 - \hat{y})^{1-y}$$

Minimize error ~ maximize belief ~ Minimize (-belief)

Idea of Logistic Regression

❖ Construct loss



Belief

One sample

$$\text{belief} = P$$

$$\text{log\_belief} = \log P$$

$$\text{log\_belief} = y\log\hat{y} + (1-y)\log(1-\hat{y})$$

$$\text{loss} = -\text{log\_belief}$$
$$= -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$$

if $y = 1$

$\quad \text{belief} = \hat{y}$

if $y = 0$

$\quad \text{belief} = 1 - \hat{y}$

$$P = \hat{y}^{\,y}(1-\hat{y})^{1-y}$$

$$\mathrm{L}(\hat{y}, y) = -y\log\hat{y} - (1-y)\log(1-\hat{y})$$

Binary cross-entropy

Logarithm

Công thức phổ biến

$\log_a a = 1$

$\log_a xy = \log_a x + \log_a y$

Hàm log là hàm đơn điệu (~thứ tự không thay đổi)

$\forall x_1 x_2 \in [a \;\; b]$ và $x_1 \leq x_2$

$\rightarrow \log(x_1) \leq \log(x_1)$

Ứng dụng trong Machine Learning

Tìm bộ tham số **θ** cho một model sao cho model mô tả được dữ liệu training

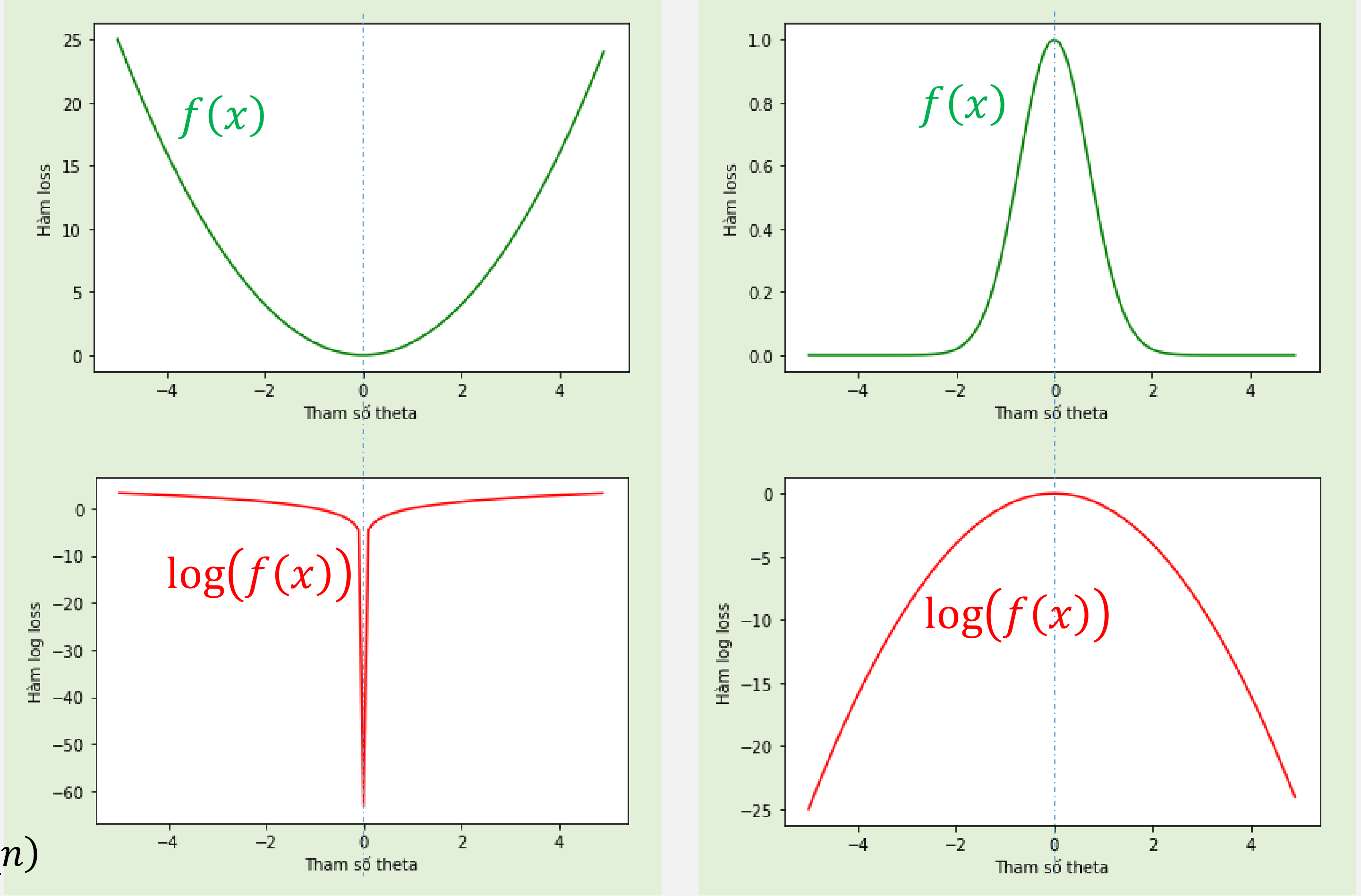
$$\underset{\theta}{\operatorname{argmax}} f(\theta) = \underset{\theta}{\operatorname{argmax}} P_\theta(\text{training data})$$

Với data sample được thu nhập độc lập với nhau

$$\underset{\theta}{\operatorname{argmax}} f(\theta) = \underset{\theta}{\operatorname{argmax}} P_\theta(\text{sample\_1}) * \cdots * P_\theta(\text{sample\_}n)$$

Dùng hàm log

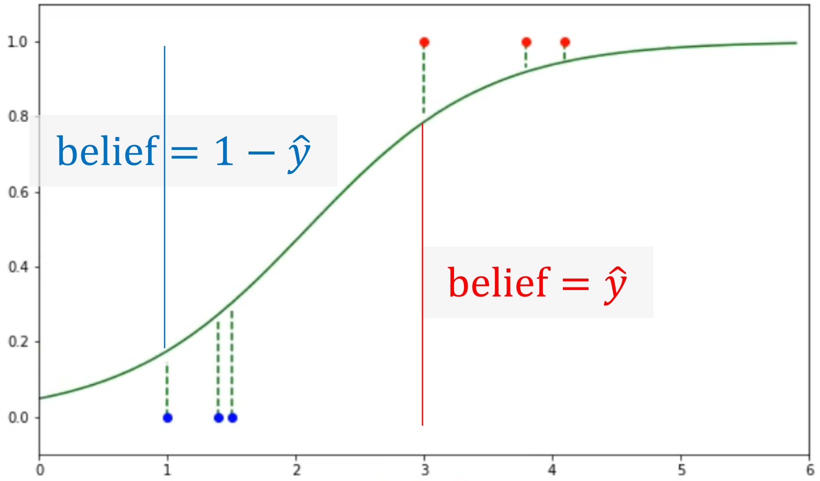
$$\underset{\theta}{\operatorname{argmax}} \log f(\theta) = \underset{\theta}{\operatorname{argmax}} [\log P_\theta(\text{sample\_1}) + \cdots + \log P_\theta(\text{sample\_}n)]$$



Ví trí cực đại của hàm $f(\theta)$ và $\log f(\theta)$ không thay đổi

Idea of Logistic Regression

❖ Construct loss



Belief

if $y_i = 1$
$\quad$ belief $= \hat{y}_i$
if $y_i = 0$
$\quad$ belief $= 1 - \hat{y}_i$

$$P_i = \hat{y}_i^{y_i}(1-\hat{y}_i)^{1-y_i}$$

$$\text{belief} = \prod_{i=1}^{n} P_i \qquad \text{since iid}$$

N samples

$$\text{log\_belief} = \sum_{i=1}^{n} \log P_i$$

$$\text{log\_belief} = \sum_{i=1}^{n} [y_i \log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)]$$

$$\text{loss} = -\text{log\_belief}$$
$$= -\sum_{i=1}^{n} [y_i \log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)]$$

$$\text{L} = \frac{1}{N}\left(-\boldsymbol{y}^T log(\hat{\boldsymbol{y}}) - (\boldsymbol{1}-\boldsymbol{y}^T) log(\boldsymbol{1}-\hat{\boldsymbol{y}})\right)$$

Binary cross-entropy

Idea of Logistic Regression

❖ Construct loss

Model and Loss

$$z = wx + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\mathrm{L}(\hat{y} - y) = -y\log(\hat{y}) - (1 - y)\log(1 - \hat{y})$$

Derivative

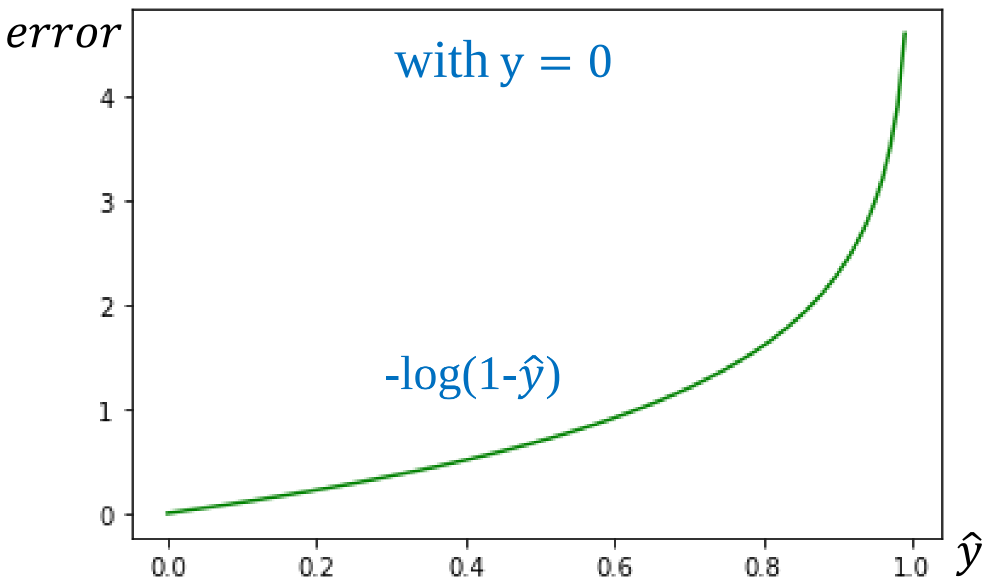
$$\frac{\partial L}{\partial \theta_i} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial \theta_i}$$

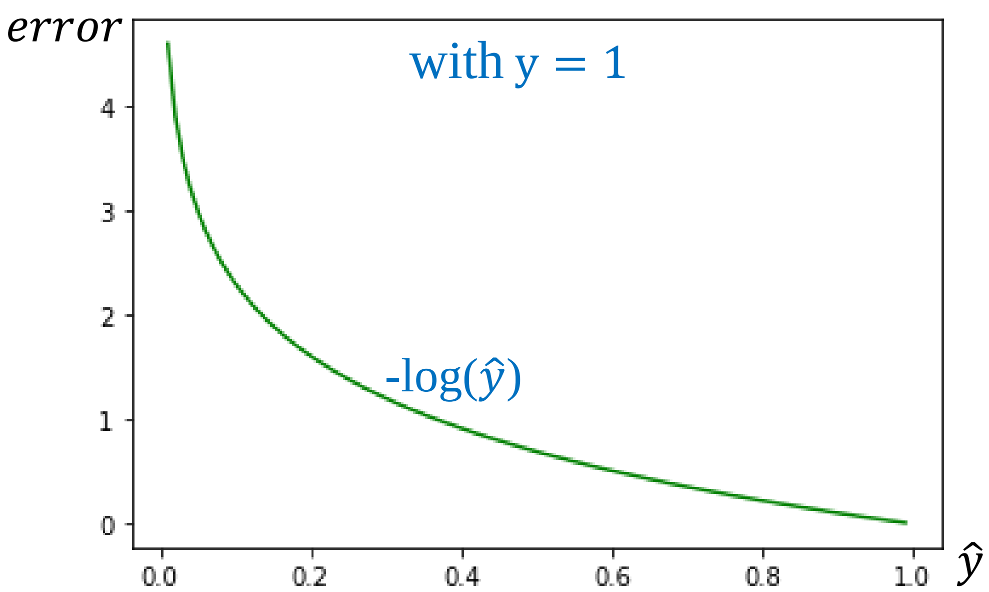
$$\frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1 - y}{1 - \hat{y}} = \frac{\hat{y} - y}{\hat{y}(1 - \hat{y})}$$

$$\frac{\partial \hat{y}}{\partial z} = \hat{y}(1 - \hat{y})$$

$$\frac{\partial z}{\partial \theta_i} = x_i$$

$$\frac{\partial L}{\partial \theta_i} = x_i(\hat{y} - y)$$





Idea of Logistic Regression

Feature	Label
Petal_Length	Category
1.4	0
1	0
1.5	0
3	1
3.8	1
4.1	1

Category 0

Category 1

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

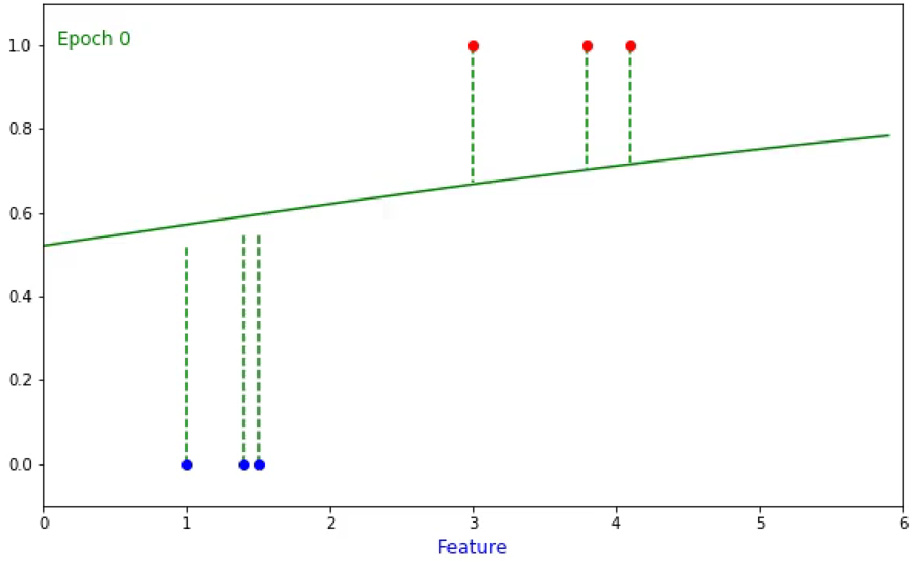
$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Feature	Label
Petal_Length	Category
1.4	1
1	1
1.5	1
3	0
3.8	0
4.1	0

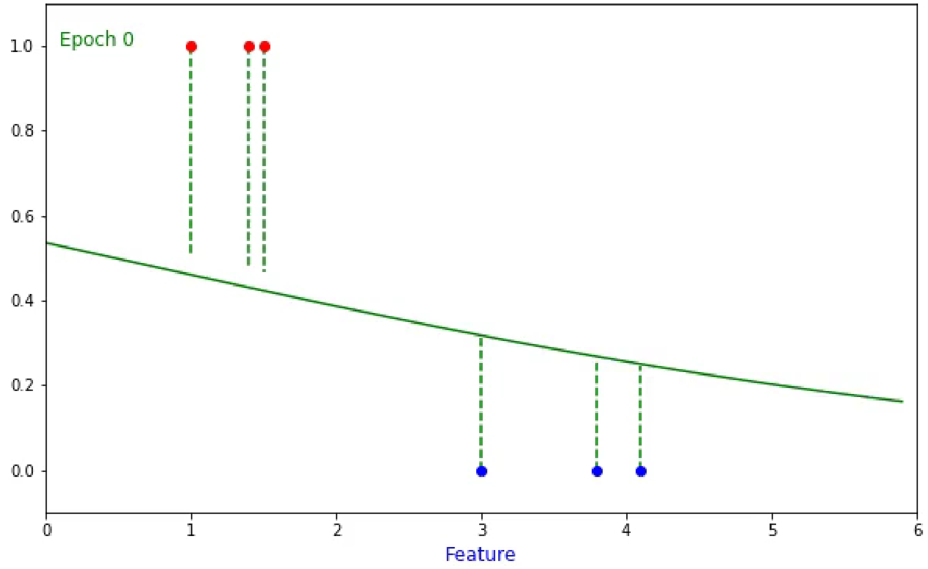
Category 0

Category 1

$$\frac{1}{1 + e^{-z}}$$



$$\frac{1}{1 + e^{-z}}$$



Outline

- Optimization Review
- Linear Regression Review
- Logistic Regression
- Examples
- Vectorization
- Implementation (optional)

Logistic Regression-Stochastic

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

2) Compute output $\hat{y}$

$$z = wx + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

4) Compute derivative

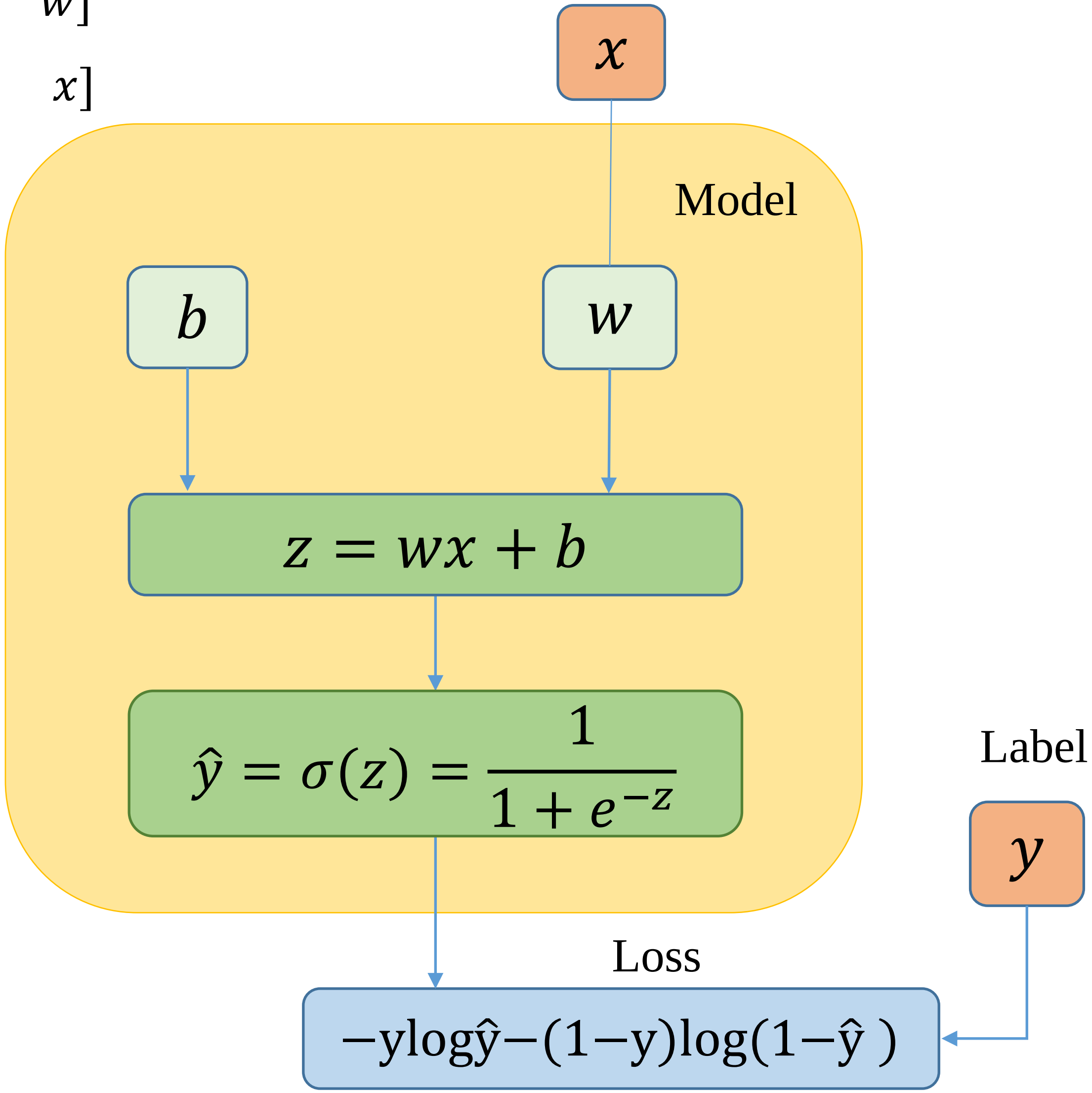
$$\frac{\partial L}{\partial w} = x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = (\hat{y} - y)$$

5) Update parameters

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

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

$$\boldsymbol{x}^T = [1 \quad x]$$

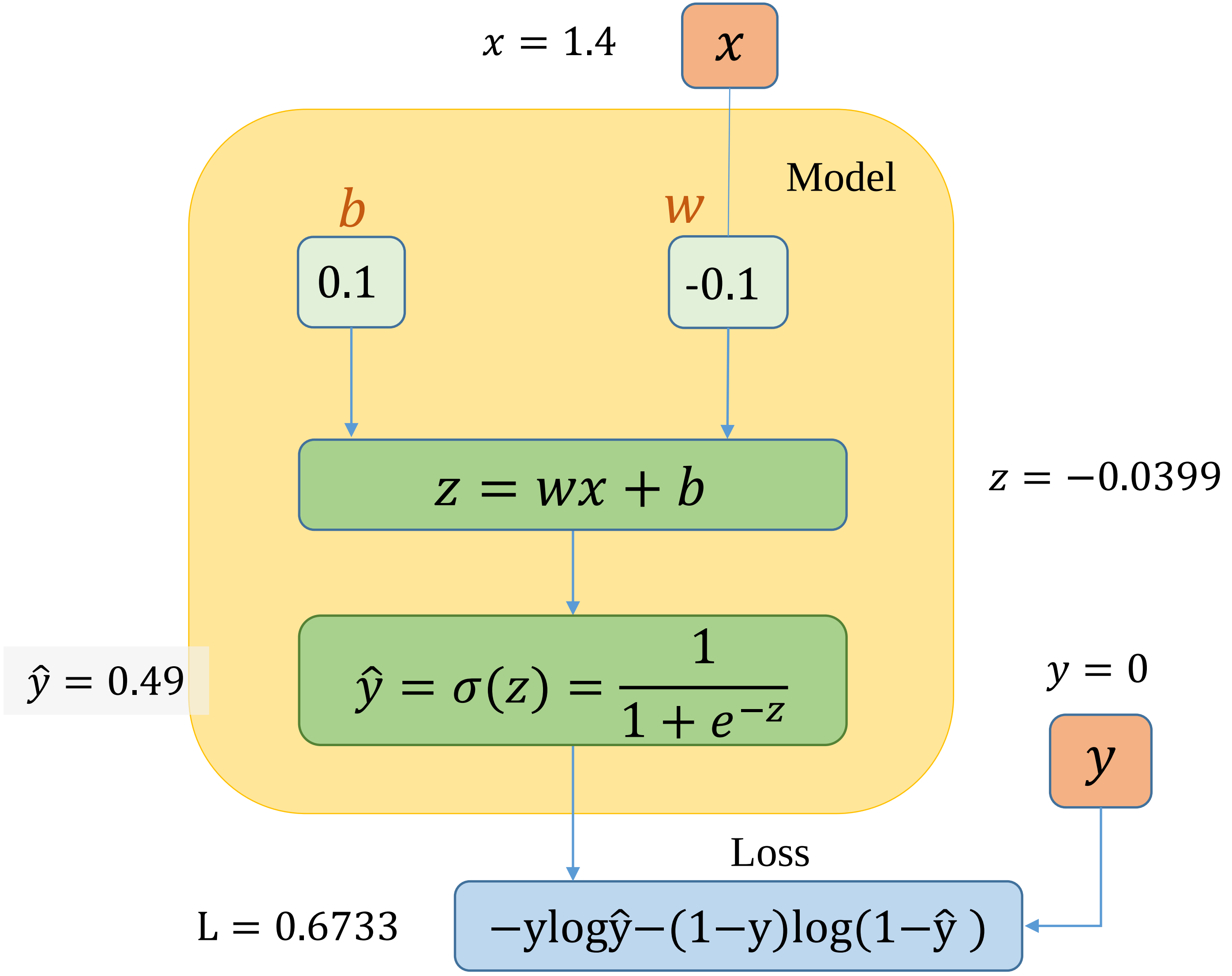


Logistic Regression-Stochastic

Dataset

Petal_Length	Label
1.4	0
1.5	0
3	1
4.1	1

$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1.4 \end{bmatrix}$ $\quad \boldsymbol{y} = [0]$



Logistic Regression-Stochastic

$\eta = 0.01$

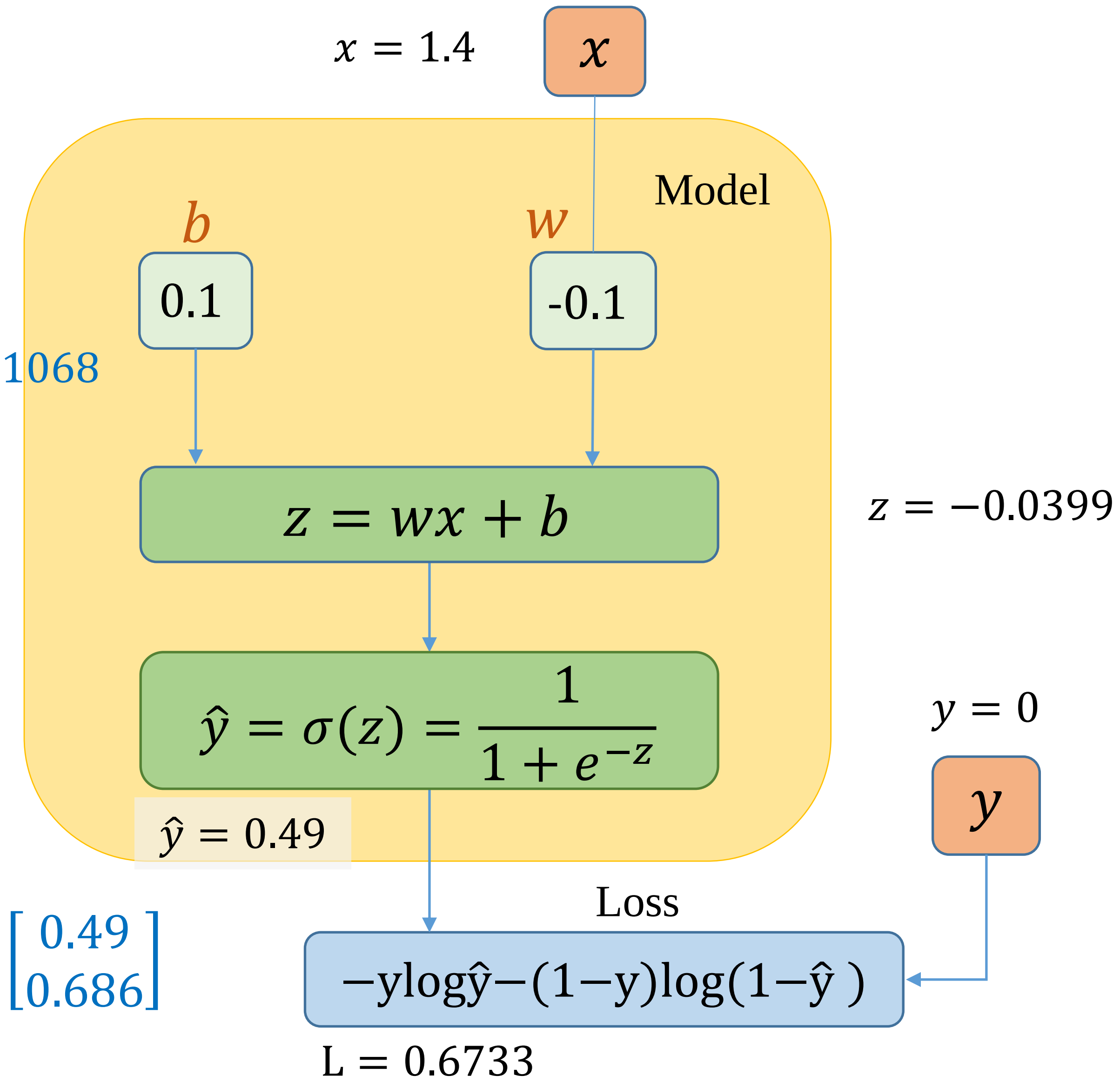
Dataset

Petal_Length	Label
1.4	0
1.5	0
3	1
4.1	1

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1.4 \end{bmatrix} \qquad \boldsymbol{y} = [0]$$

$b = 0.1 - \eta 0.49 = 0.095$

$w = -0.1 - \eta 0.686 = -0.1068$



$$\begin{bmatrix} L'_b \\ L'_w \end{bmatrix} = \begin{bmatrix} 1 * 0.49 \\ 1.4 * 0.49 \end{bmatrix} = \begin{bmatrix} 0.49 \\ 0.686 \end{bmatrix}$$

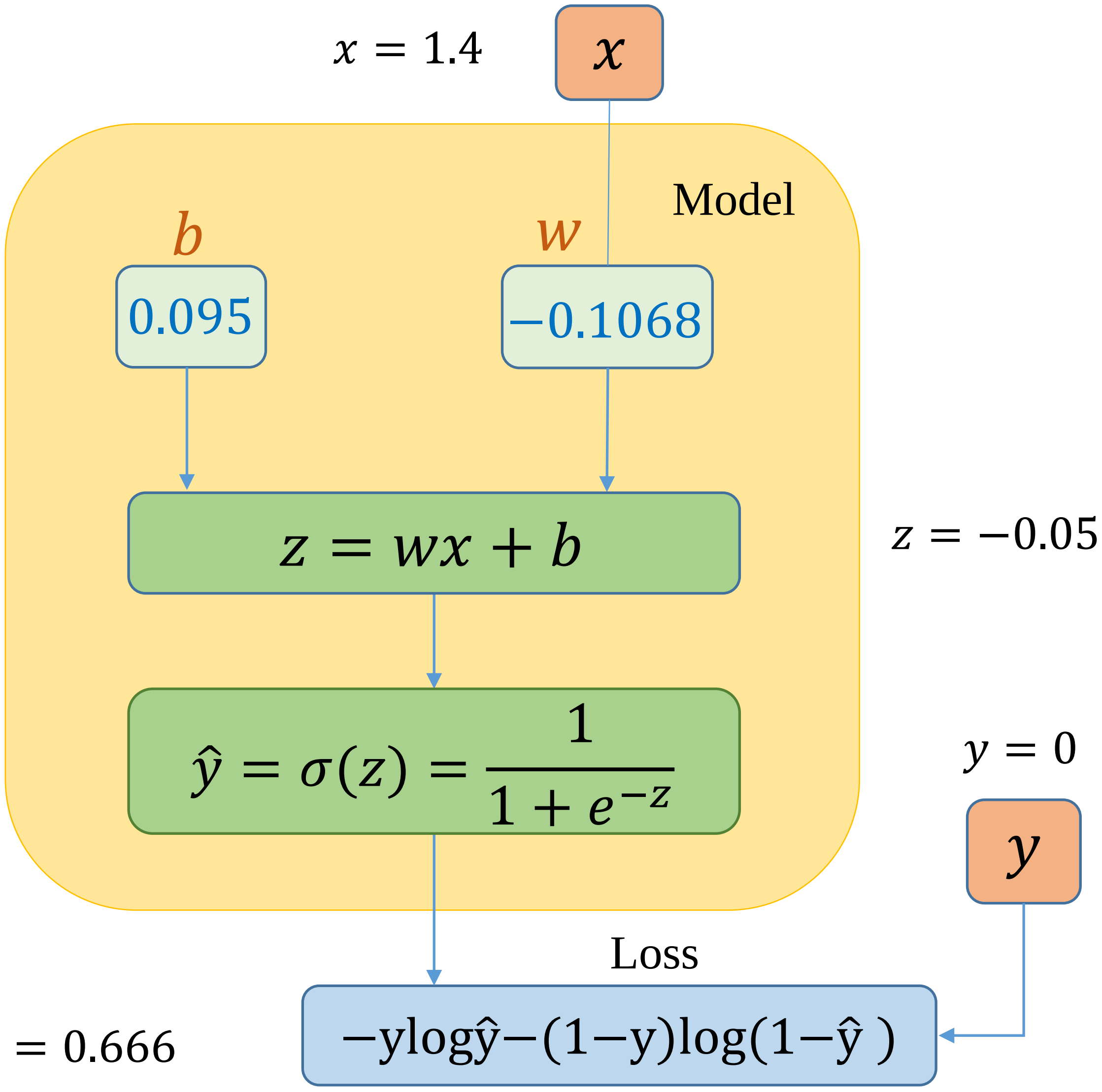
Logistic Regression-Stochastic

Dataset

Petal_Length	Label
1.4	0
1.5	0
3	1
4.1	1

$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1.4 \end{bmatrix}$ $\boldsymbol{y} = [0]$

$x = 1.4$



$z = -0.0545$

$\hat{y} = 0.486$

$y = 0$

previous L = 0.6733

L = 0.666

Another example

Logistic Regression-Stochastic

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

2) Compute output $\hat{y}$

$$z = w_1 x_1 + w_2 x_2 + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

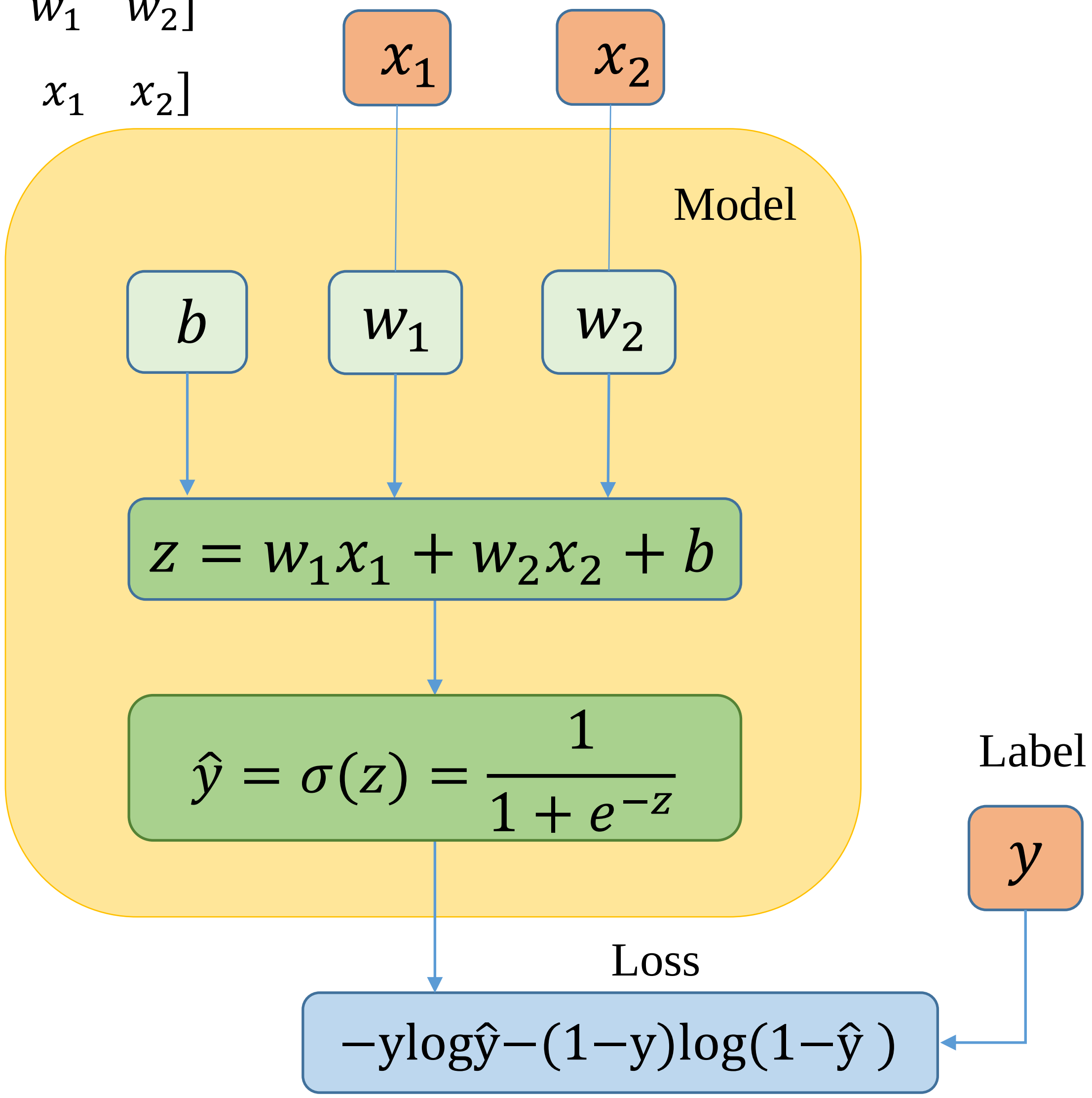
$$\frac{\partial L}{\partial w_i} = x_i(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = (\hat{y} - y)$$

5) Update parameters

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

$$\boldsymbol{\theta}^T = [b \quad w_1 \quad w_2]$$

$$\boldsymbol{x}^T = [1 \quad x_1 \quad x_2]$$

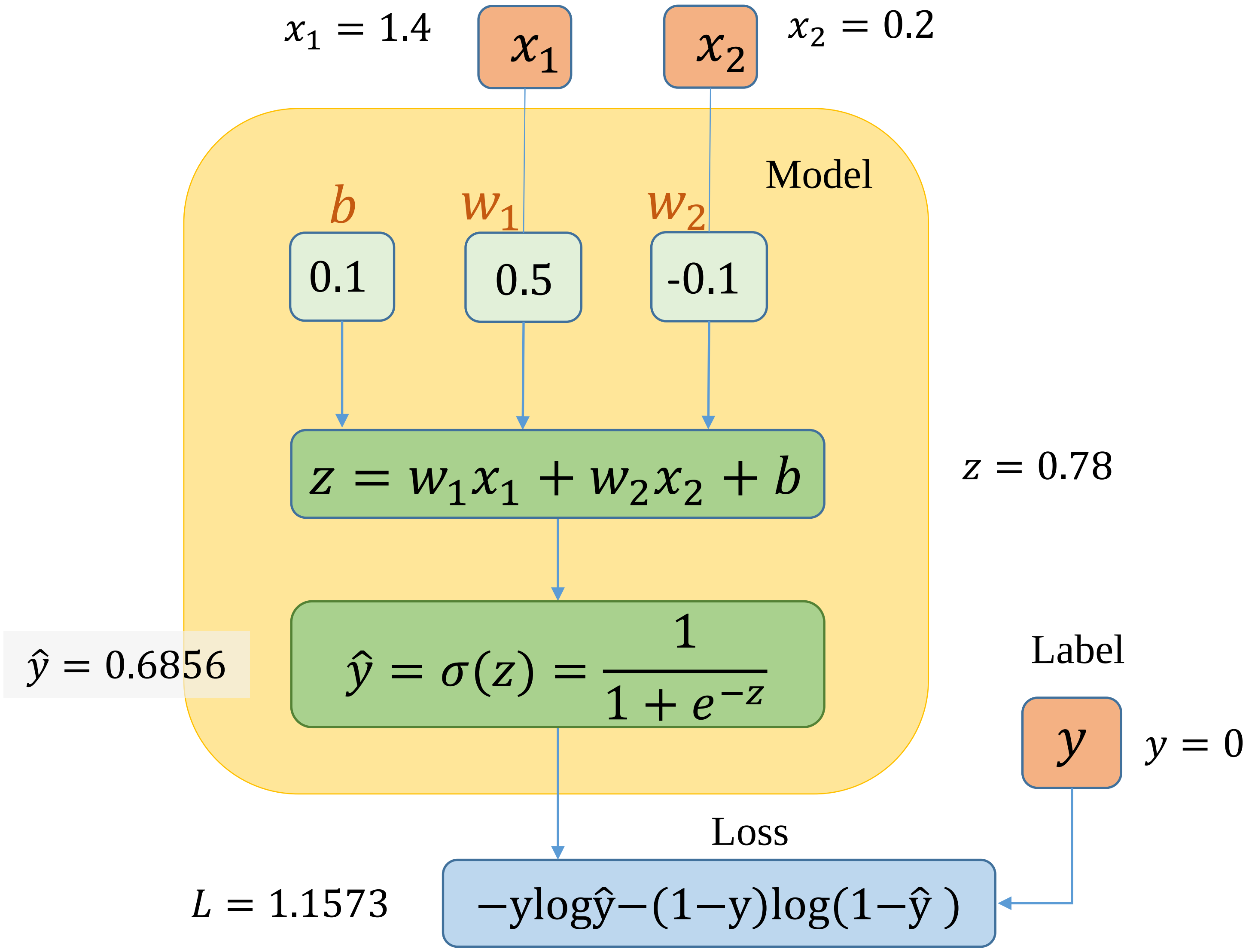


Logistic Regression-Stochastic

Dataset

Petal_Length	Petal_Width	Label
1.4	0.2	0
1.5	0.2	0
3	1.1	1
4.1	1.3	1

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1.4 \\ 0.2 \end{bmatrix} \qquad \boldsymbol{y} = [0]$$



Logistic Regression-Stochastic

Dataset

Petal_Length	Petal_Width	Label
1.4	0.2	0
1.5	0.2	0
3	1.1	1
4.1	1.3	1

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1.4 \\ 0.2 \end{bmatrix} \qquad \boldsymbol{y} = [0]$$

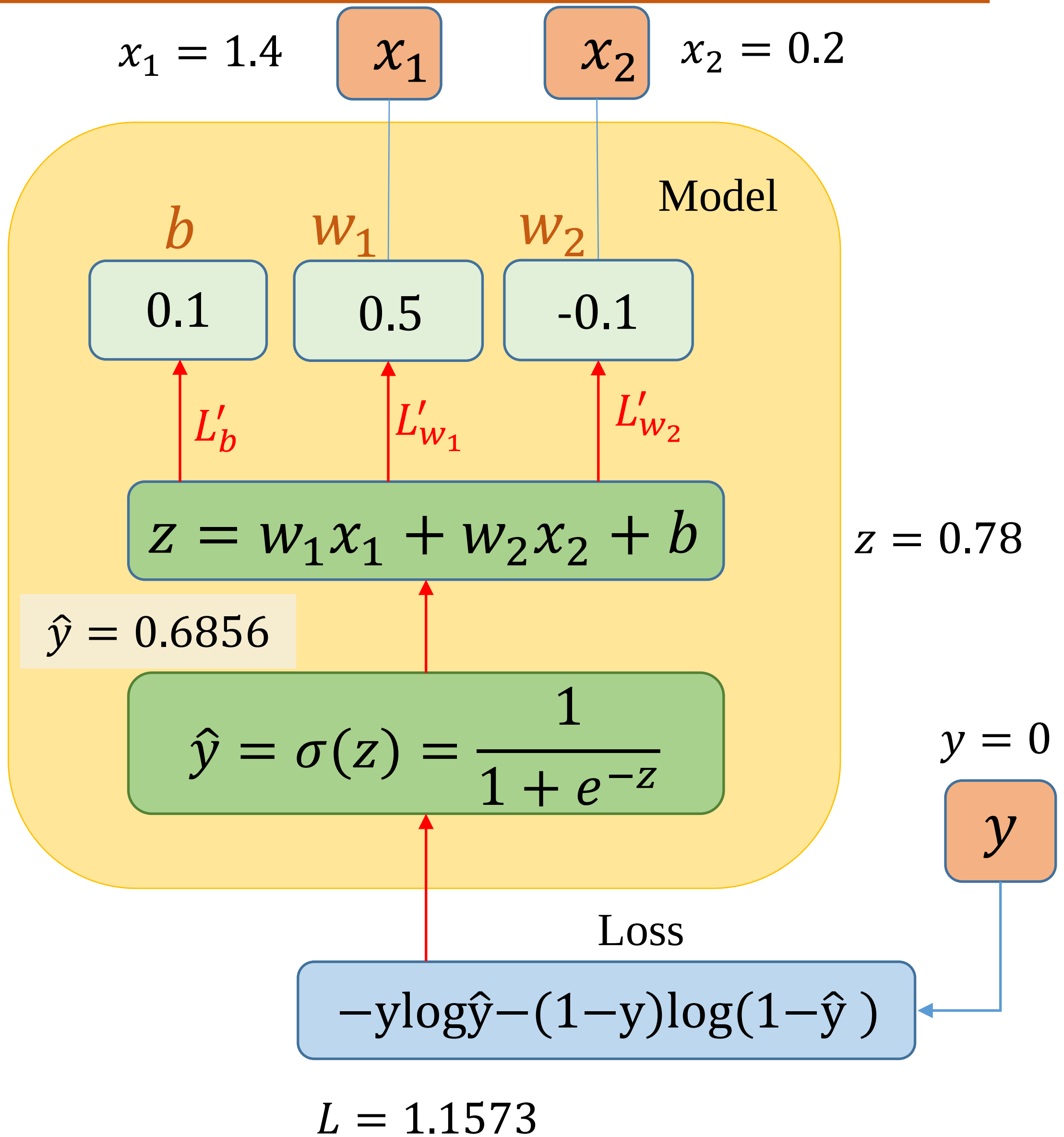
$$\begin{bmatrix} L'_b \\ L'_{w_1} \\ L'_{w_2} \end{bmatrix} = \begin{bmatrix} 1 * 0.6856 \\ 1.4 * 0.6856 \\ 0.2 * 0.6856 \end{bmatrix} = \begin{bmatrix} 0.6856 \\ 0.9599 \\ 0.1371 \end{bmatrix}$$

$\eta = 0.01$

$b = 0.1 - \eta 0.6856$
$= 0.0931$

$w_1 = 0.5 - \eta 0.9598$
$= 0.4990$

$w_2 = -0.1 + \eta 0.1371$
$= -0.1013$



Logistic Regression-Stochastic

Dataset

Petal_Length	Petal_Width	Label
1.4	0.2	0
1.5	0.2	0
3	1.1	1
4.1	1.3	1

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1.4 \\ 0.2 \end{bmatrix} \qquad \boldsymbol{y} = [0]$$

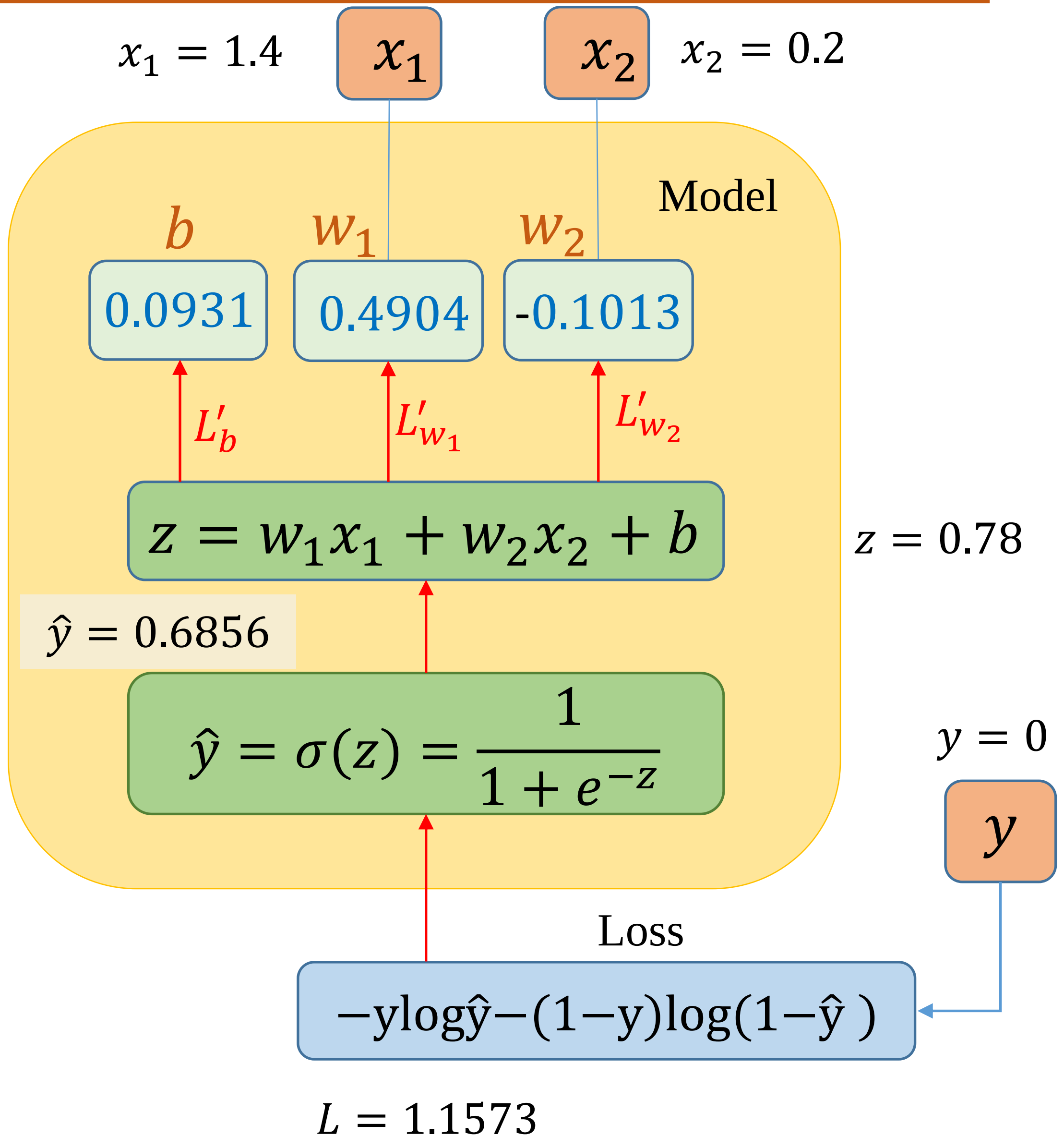
$$\begin{bmatrix} L'_b \\ L'_{w_1} \\ L'_{w_2} \end{bmatrix} = \begin{bmatrix} 1 * 0.6856 \\ 1.4 * 0.6856 \\ 0.2 * 0.6856 \end{bmatrix} = \begin{bmatrix} 0.6856 \\ 0.9599 \\ 0.1371 \end{bmatrix}$$

$\eta = 0.01$

$$b = 0.1 - \eta 0.6856$$
$$= 0.0931$$

$$w_1 = 0.5 - \eta 0.9598$$
$$= 0.4990$$

$$w_2 = -0.1 + \eta 0.1371$$
$$= -0.1013$$

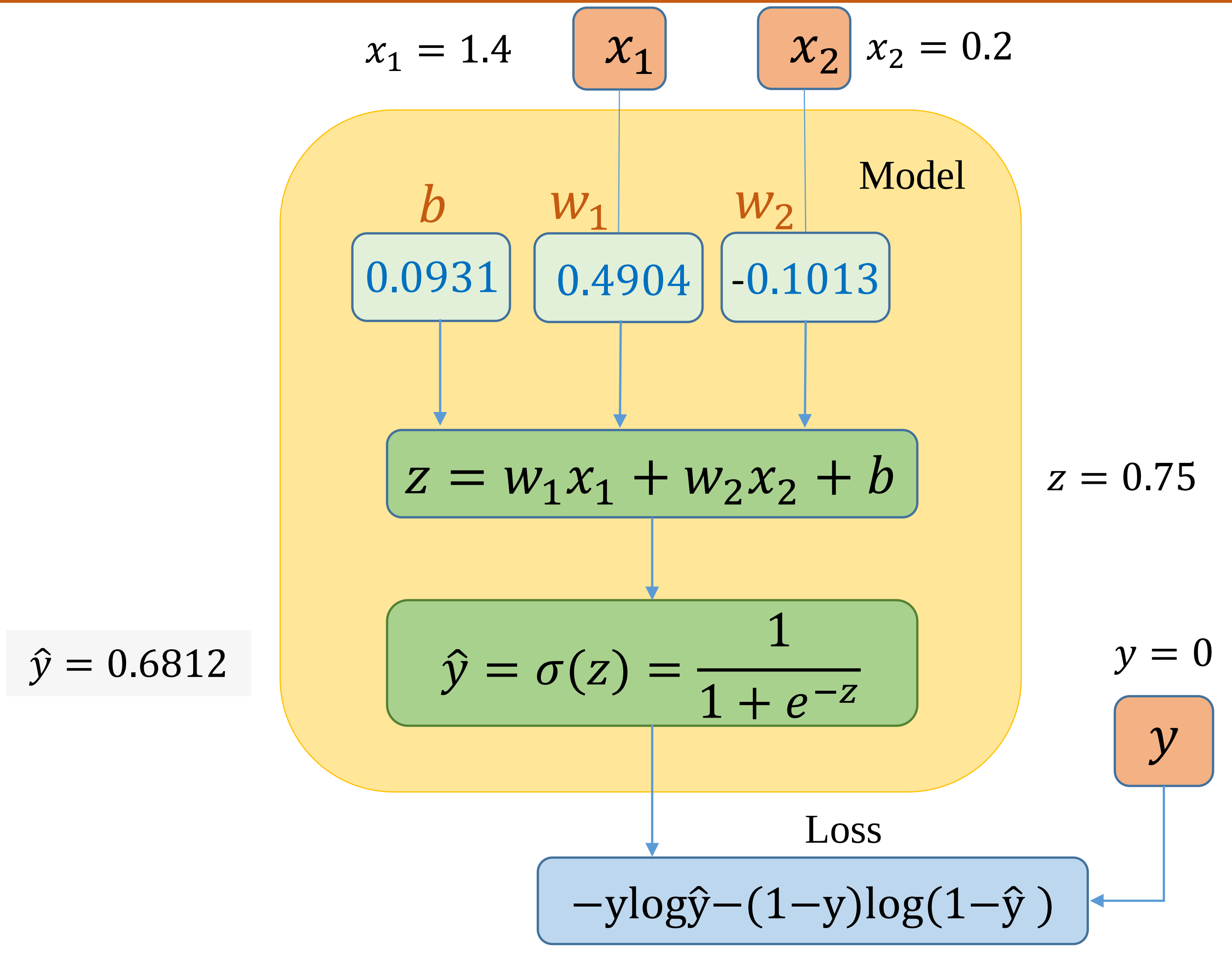


Logistic Regression-Stochastic

Dataset

Petal_Length	Petal_Width	Label
1.4	0.2	0
1.5	0.2	0
3	1.1	1
4.1	1.3	1

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1.4 \\ 0.2 \end{bmatrix} \qquad \boldsymbol{y} = [0]$$



previous $\boldsymbol{L} = [1.1573]$

$L = 1.1432$

Outline

- Optimization Review
- Linear Regression Review
- Logistic Regression
- Examples
- Vectorization
- Implementation (optional)

Review

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

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

$$\underset{\text{data}}{\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}} * 2 = \underset{\text{result}}{\begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}}$$

Review

Dot product

$$\vec{v} = \begin{bmatrix} v_1 \\ \dots \\ v_n \end{bmatrix} \qquad \vec{u} = \begin{bmatrix} u_1 \\ \dots \\ u_n \end{bmatrix}$$

$$\vec{v} \cdot \vec{u} = v_1 \times u_1 + \cdots + v_n \times u_n$$

v • w = result

$$\begin{bmatrix} 1 & 2 \end{bmatrix} \bullet \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 8 \end{bmatrix}$$

```python
def dot_product(vector1, vector2):
    '''
    Compute dot product between two vectors
    Output is a floating-point number
    '''

    return sum([v1*v2 for v1, v2 in zip(vector1, vector2)])

# test case
vector1 = [1, 2, 3]
vector2 = [2, 3, 4]

ouptut = dot_product(vector1, vector2)
print(ouptut)
```

```
20
```

```python
import numpy as np

v = np.array([1, 2])
w = np.array([2, 3])

# Tính inner product giữa v và w
print('method 1 \n', v.dot(w))
print('method 2 \n', np.dot(v, w))
```

```
method 1
8

method 2
8
```

Vectorization

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 $\hat{y}$

$$z = wx + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

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

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

dot product

Vectorization

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

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

Traditional

$$z = wx + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

$$z = \boldsymbol{\theta}^T \boldsymbol{x} \qquad \hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

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

numbers

What will we do?

Vectorization

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

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

Traditional

$$z = wx + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}\,))$$

4) Compute derivative

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

5) Update parameters

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

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

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

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

common factor

Vectorization

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

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

Traditional

$$z = wx + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

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

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

$$\nabla_{\boldsymbol{\theta}} L = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \end{bmatrix}$$

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

$$\boldsymbol{\theta} \quad \boldsymbol{\theta} \quad \nabla_{\boldsymbol{\theta}} L$$

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

Vectorization

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

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

$$z = wx + b \qquad \hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

4) Compute derivative

Traditional

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

5) Update parameters

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

η is learning rate

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

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta} \qquad \hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

Vectorized

4) Compute derivative

$$\nabla_{\boldsymbol{\theta}} L = \boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} L$$

η is learning rate

Vectorization

❖ Implementation (using Numpy)

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

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta} \qquad \hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

4) Compute derivative

$$\nabla_{\boldsymbol{\theta}} L = \boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} L$$

η is learning rate

```python
def sigmoid_function(z):
    return 1 / (1 + np.exp(-z))

def predict(X, theta):
    return sigmoid_function( np.dot(X.T, theta) )

def loss_function(y_hat, y):
    return -y*np.log(y_hat) - (1 - y)*np.log(1 - y_hat)

def compute_gradient(X, y_hat, y):
    return X*(y_hat - y)

def update(theta, lr, gradient):
    return theta - lr*gradient
```

```python
# Given X and y
```

```python
# compute output
y_hat = predict(X, theta)

# compute loss
loss = loss_function(y_hat, y)

# compute mean of gradient
gradient = compute_gradient(X, y_hat, y)

# update
theta = update(theta, lr, gradient)
```

Dataset

Petal_Length	Petal_Width	Label
1.4	0.2	0
1.5	0.2	0
3	1.1	1
4.1	1.3	1

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

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta} \qquad \hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\hat{y}, y) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

4) Compute derivative

$$\nabla_{\boldsymbol{\theta}} L = \boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

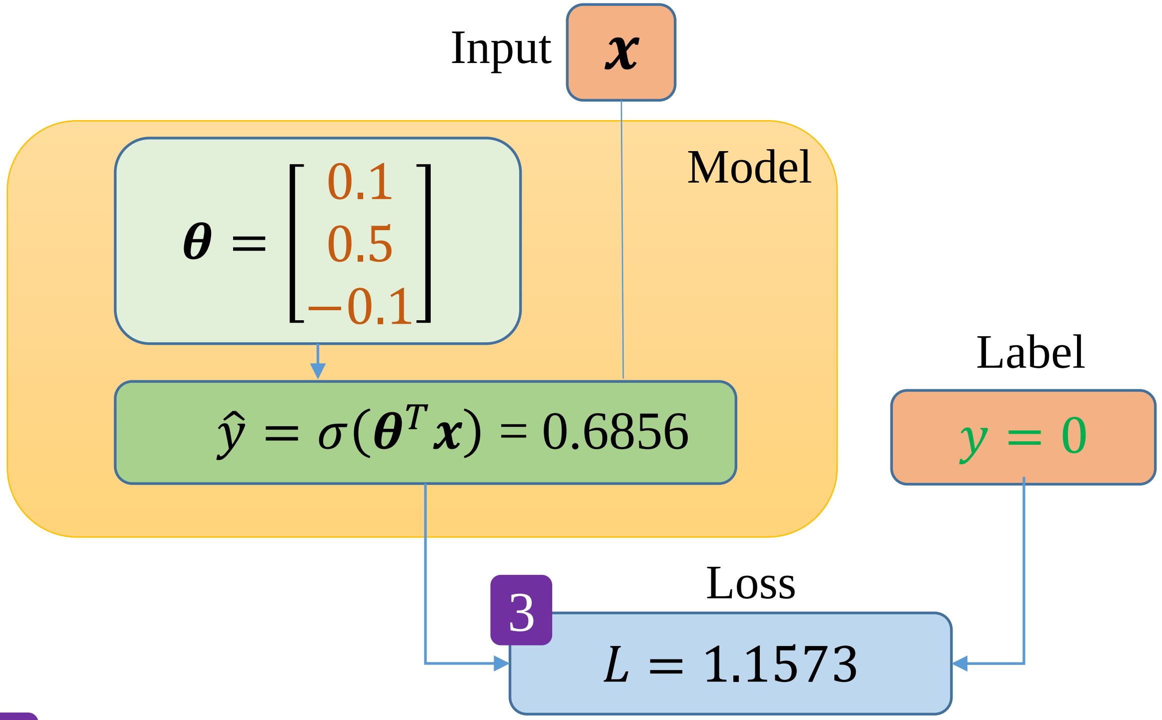
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} L$$

1

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.4 \\ 0.2 \end{bmatrix}$$

Given $\boldsymbol{\theta} = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.5 \\ -0.1 \end{bmatrix}$

$\eta = 0.01$



4

$$\nabla_{\boldsymbol{\theta}} L = \boldsymbol{x}(\hat{y} - y) = \begin{bmatrix} 1 \\ 1.4 \\ 0.2 \end{bmatrix} [0.6856] = \begin{bmatrix} 0.6856 \\ 0.9599 \\ 0.1371 \end{bmatrix} = \begin{bmatrix} L'_b \\ L'_{w_1} \\ L'_{w_2} \end{bmatrix}$$

5

$$\boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}} = \begin{bmatrix} 0.1 \\ 0.5 \\ -0.1 \end{bmatrix} - \eta \begin{bmatrix} 0.6856 \\ 0.9599 \\ 0.1371 \end{bmatrix} = \begin{bmatrix} 0.093 \\ 0.499 \\ -0.101 \end{bmatrix}$$

Logistic Regression-Stochastic

Dataset

Petal_Length	Petal_Width	Label
1.4	0.2	0
1.5	0.2	0
3	1.1	1
4.1	1.3	1

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1.4 \\ 0.2 \end{bmatrix} \qquad \boldsymbol{y} = [0]$$

Demo

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

2) Compute output $\hat{y}$

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

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\boldsymbol{\theta}) = -\text{y}\log\hat{\text{y}} - (1-\text{y})\log(1-\hat{\text{y}})$$

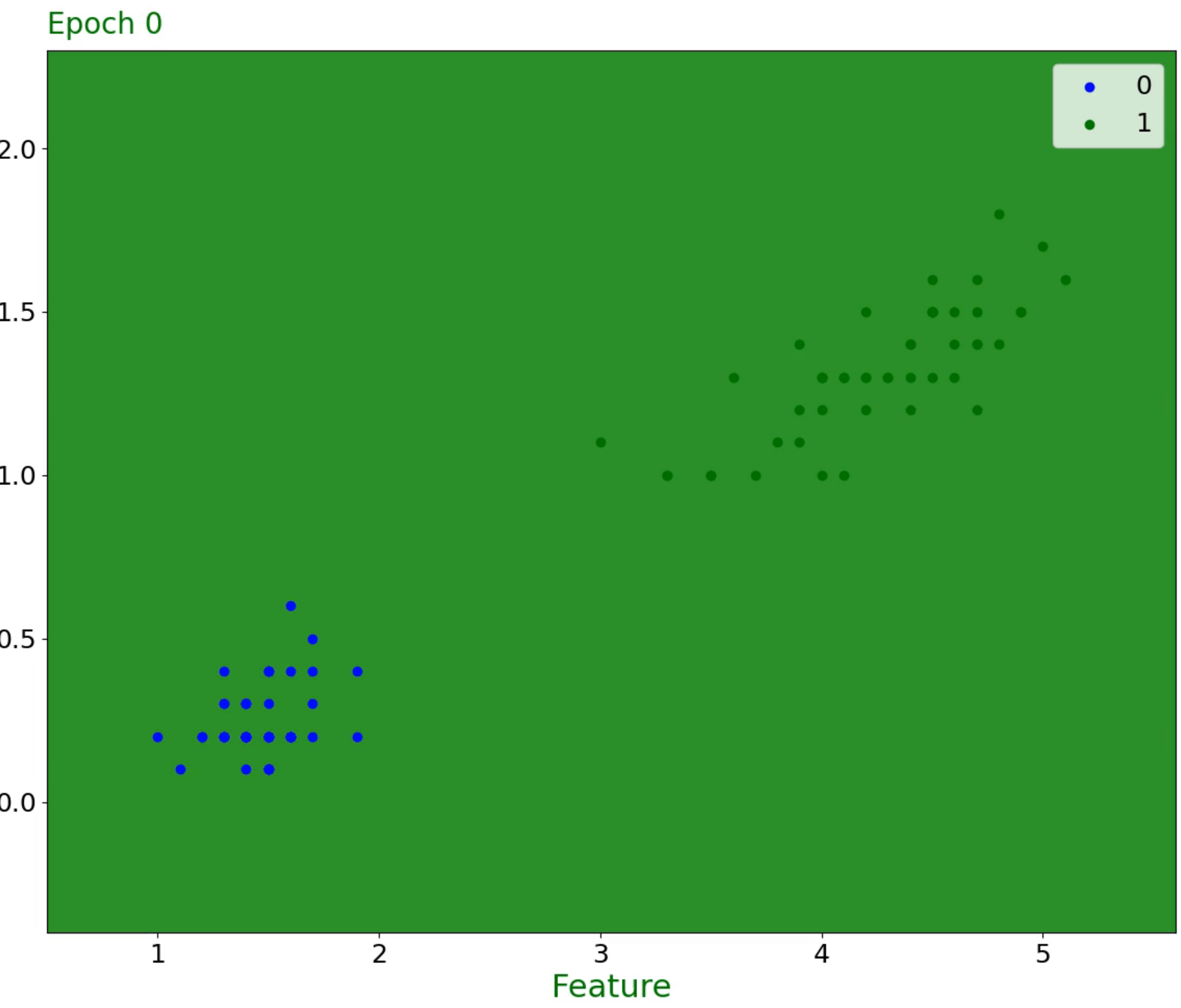
4) Compute derivative

$$\nabla_{\boldsymbol{\theta}} L = \mathbf{x}(\hat{\text{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} L$$

η is learning rate



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

2) Compute output $\hat{y}$

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

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3) Compute loss

$$L(\boldsymbol{\theta}) = -\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}})$$

4) Compute derivative

$$\nabla_{\boldsymbol{\theta}} L = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} L$$

η is learning rate