AI VIETNAM
All-in-One Course

# Insight into Logistic Regression

Quang-Vinh Dinh
Ph.D. in Computer Science

# Outline

- Vectorization
- Optimiztion for 1+ samples
- Logistic Regression – Mini-batch
- Logistic Regression – Batch
- BCE and MSE Loss Functions
- Sigmoid and Tanh Function (Optional)

demo

# Implementation - One Sample

| Feature | Label |
|---|---|
| Petal_Length | Label |
| 1.4 | 0 |
| 1.5 | 0 |
| 3 | 1 |
| 4.1 | 1 |

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

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

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

def predict(x, w, b):
    z = w*x + b
    y_hat = sigmoid_function(z)

    return y_hat

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

def compute_gradient(x, y_hat, y):
    dw = x*(y_hat - y)
    db = (y_hat - y)

    return dw, db

def update(w, b, dw, db, lr):
    w = w - lr*dw
    b = b - lr*db

    return w, b
```

if #features changes, which functions are affected?

demo

| 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 the 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

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

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

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

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

def predict(x1, x2, b, w1, w2):
    z = x1*w1 + x2*w2 + b
    y_hat = sigmoid_function(z)

    return y_hat

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

def compute_gradient(x1, x2, y_hat, y):
    db  = (y_hat - y)
    dw1 = x1*(y_hat - y)
    dw2 = x2*(y_hat - y)

    return (db, dw1, dw2)

def update(b, w1, w2, lr, db, dw1, dw2):
    b  = b - lr*db
    w1 = w1 - lr*dw1
    w2 = w2 - lr*dw2

    return (b, w1, w2)
```

How to solve the problem?

# Vector/Matrix Operations

## 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

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

$$\begin{array}{ccccc} \textbf{data} & & & & \textbf{result} \\ \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} & * & \begin{bmatrix} 2 \end{bmatrix} & = & \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} \end{array}$$

# Vector/Matrix Operations

## Dot product

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

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

$$\underset{\textbf{v}}{\begin{bmatrix} 1 & 2 \end{bmatrix}} \bullet \underset{\textbf{w}}{\begin{bmatrix} 2 \\ 3 \end{bmatrix}} = \underset{\textbf{result}}{\begin{bmatrix} 8 \end{bmatrix}}$$

```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 |
| $x$ | $y$ |

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

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

$\eta$ 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

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

$\eta$ is learning rate

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

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

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

numbers

What will we do?

# Vectorization

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

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

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

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

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

common factor

# Vectorization

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

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

$\eta$ is learning rate

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

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad \nabla_{\boldsymbol{\theta}} L = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \end{bmatrix}$$

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

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

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

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

# 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

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

$\eta$ is learning rate

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

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

$\eta$ is learning rate

# Vectorization

## ❖ Implementation (using Numpy)

1) Pick a sample $(\boldsymbol{x}, y)$ 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

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

4) Compute derivative

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

5) Update parameters

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

$\eta$ 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
# 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)
```

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

**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 $(\boldsymbol{x}, y)$ 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 $\boldsymbol{\theta} = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.5 \\ -0.1 \end{bmatrix}$

$\eta = 0.01$

Input $\boldsymbol{x}$

Model

$$\boldsymbol{\theta} = \begin{bmatrix} 0.1 \\ 0.5 \\ -0.1 \end{bmatrix}$$

$$\hat{y} = \sigma(\boldsymbol{\theta}^T \boldsymbol{x}) = 0.6856$$

Label

$y = 0$

3 Loss

$$L = 1.1573$$

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 $(\boldsymbol{x}, y)$ 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

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

$\eta$ is learning rate

# Outline

- Vectorization
- Optimiztion for 1+ samples
- Logistic Regression – Mini-batch
- Logistic Regression – Batch
- BCE and MSE Loss Functions
- Sigmoid and Tanh Function (Optional)

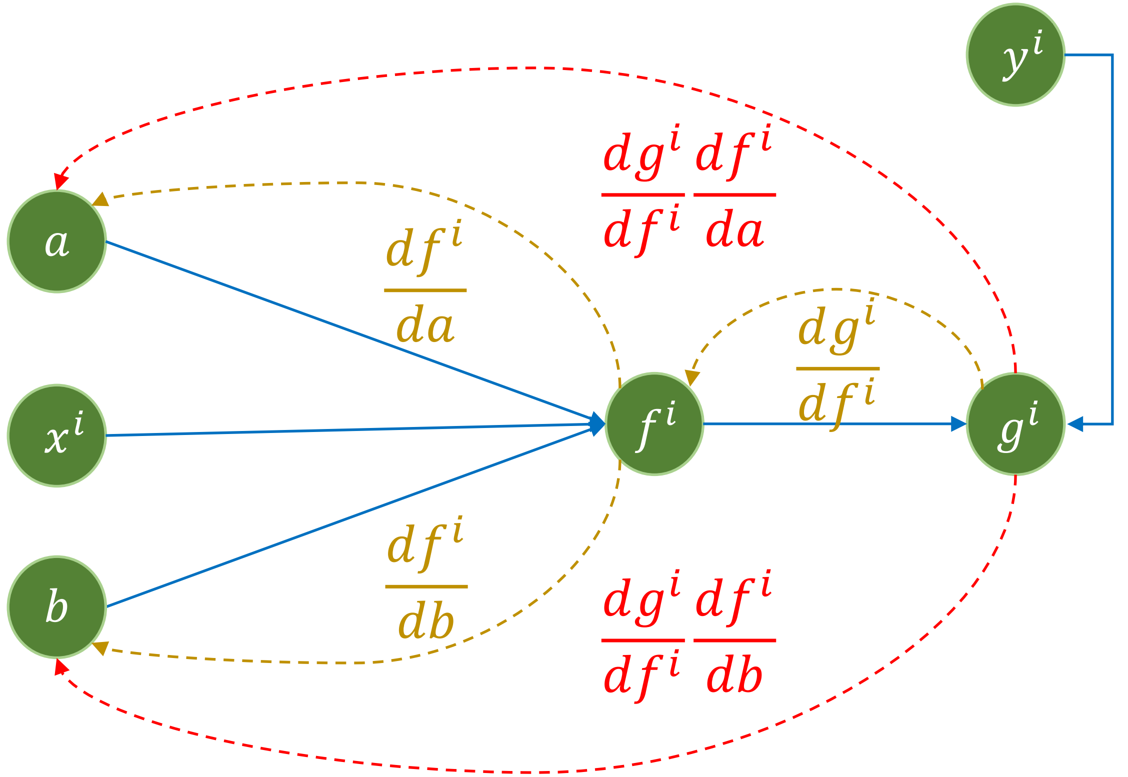
# Optimization for One+ Samples

## ❖ Equations for partial gradients

$f(x^i) = ax^i + b \qquad (x^1{=}1, y^1{=}5)$

$g(f^i) = (f^i - y^i)^2 \qquad (x^2{=}2, y^2{=}7)$

illustration




$$\frac{df}{da} = x \qquad \frac{df}{db} = 1$$

$$\frac{dg}{df} = 2(f - y)$$

$$\frac{dg}{da} = \frac{dg}{df}\frac{df}{da} = 2x(f - y)$$

$$\frac{dg}{db} = \frac{dg}{df}\frac{df}{db} = 2(f - y)$$

During looking for optimal a and b, at a given time, a and b have concrete values

## ❖ Optimization for a composite function

Find a and b so that g(f(x)) is minimum

$f(x^i) = ax^i + b$ $\quad (x^1=1, y^1=5)$

$g(f^i) = (f^i - y^i)^2$ $\quad (x^2=2, y^2=7)$

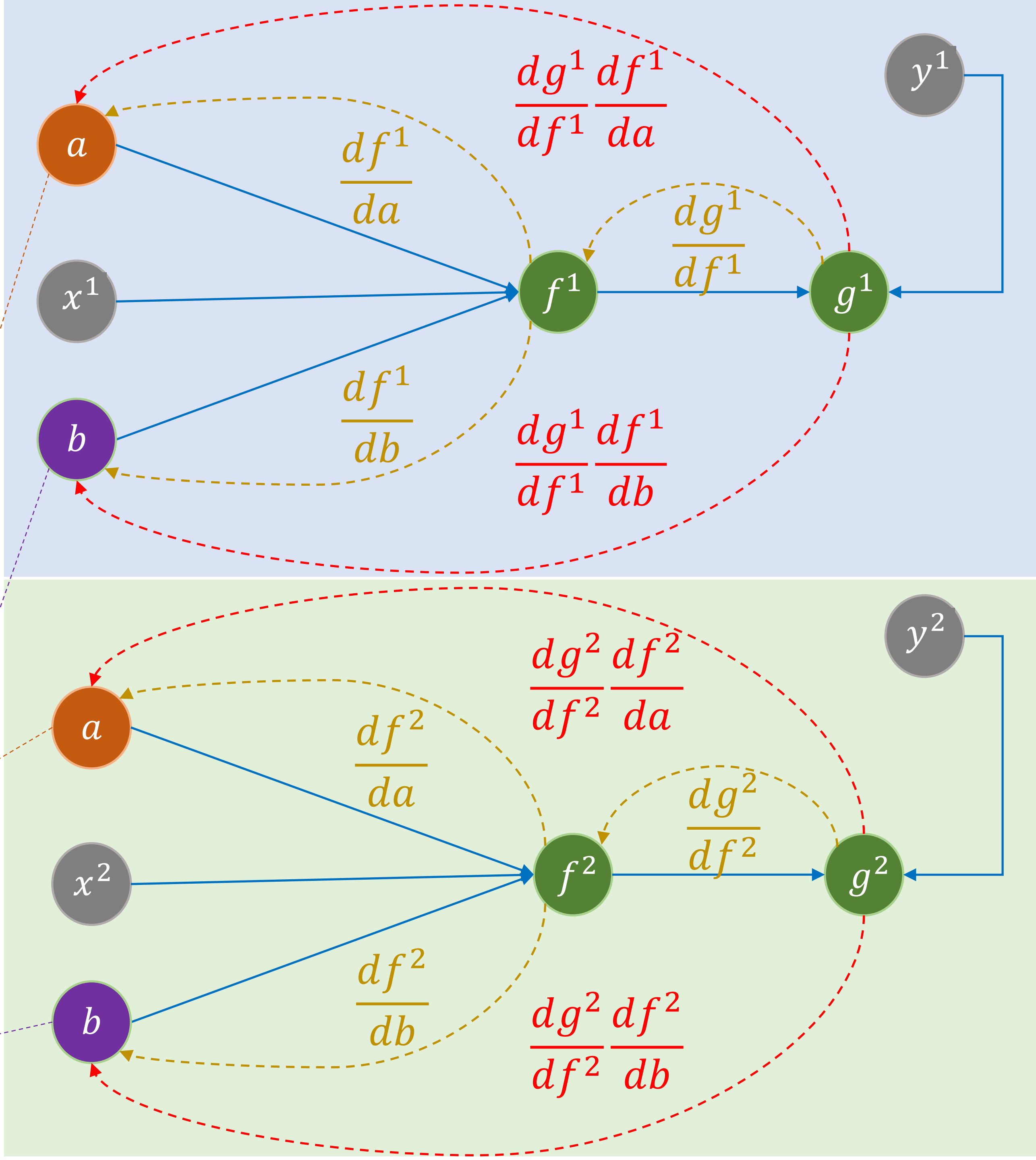
Partial derivative functions

$$\frac{dg}{da} = \frac{dg}{df}\frac{df}{da} = 2x(f-y)$$

$$\frac{dg}{db} = \frac{dg}{df}\frac{df}{db} = 2(f-y)$$

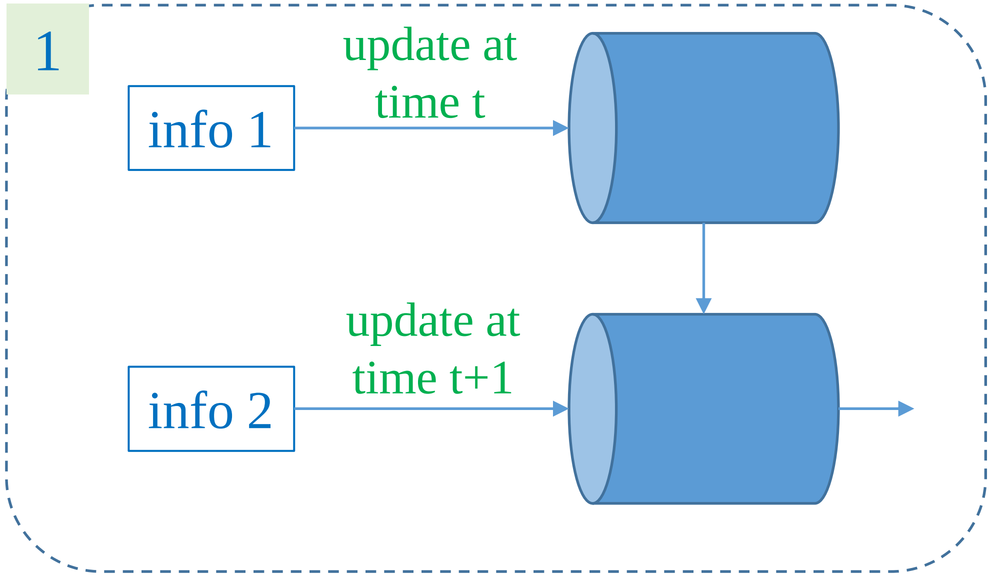
$$\sum_i \frac{dg^i}{da} = \frac{dg^1}{df^1}\frac{df^1}{da} + \frac{dg^2}{df^2}\frac{df^2}{da}$$

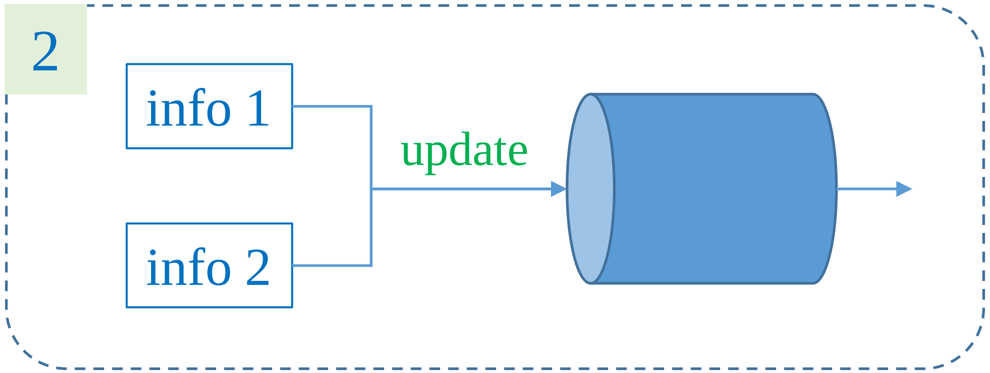
$$\sum_i \frac{dg_i}{db} = \frac{dg^1}{df^1}\frac{df^1}{db} + \frac{dg^2}{df^2}\frac{df^2}{db}$$

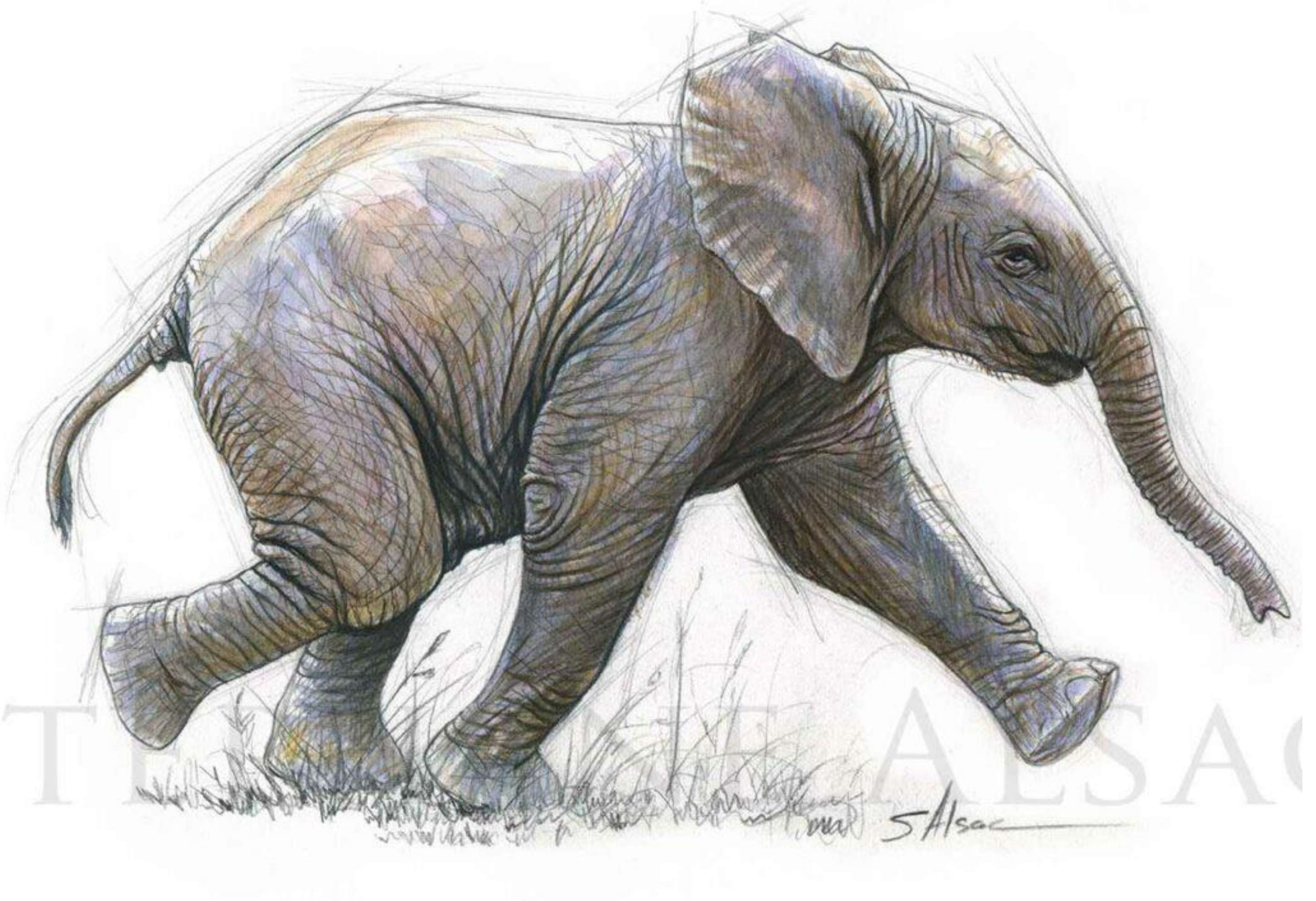


# Optimization

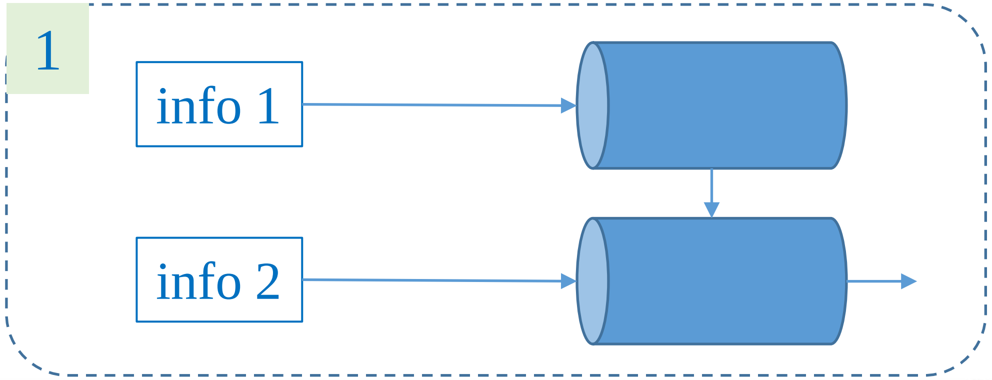
## ❖ How to use gradient information



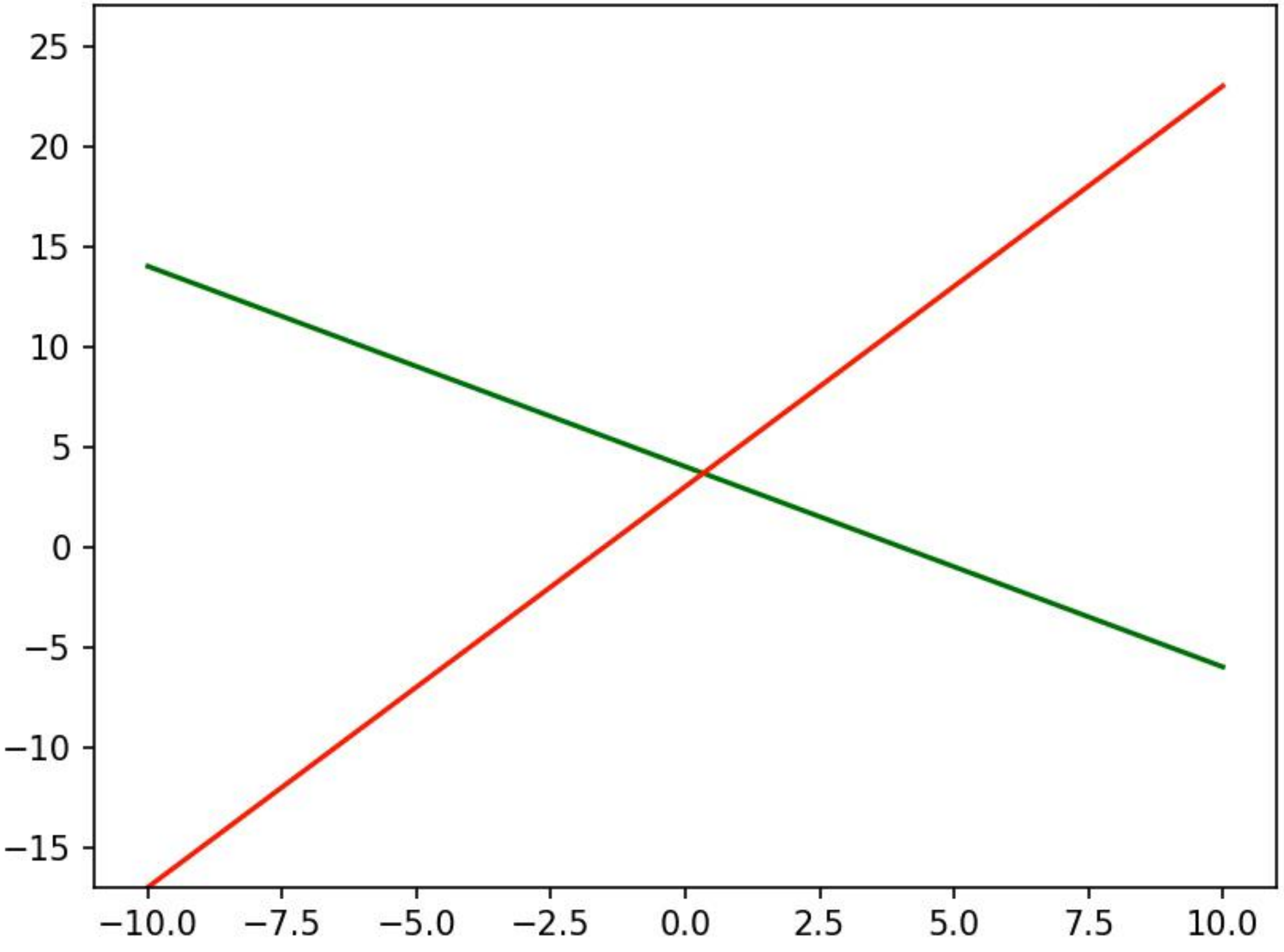



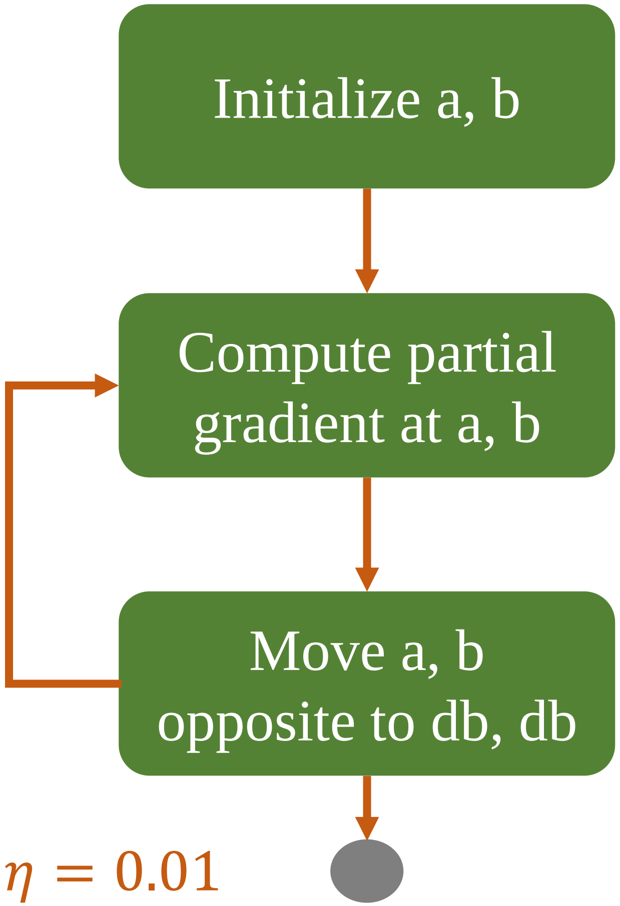
# Summary




$$\frac{dg}{da} = \frac{dg}{df}\frac{df}{da} = 2x(f - y)$$

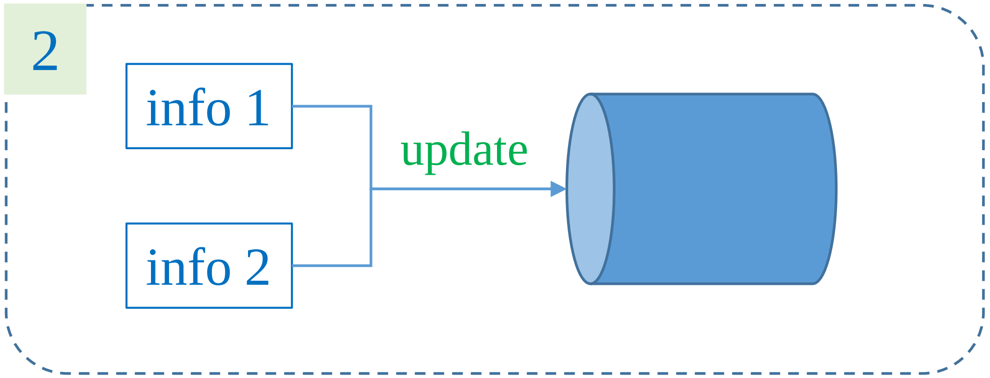
$$\frac{dg}{db} = \frac{dg}{df}\frac{df}{db} = 2(f - y)$$



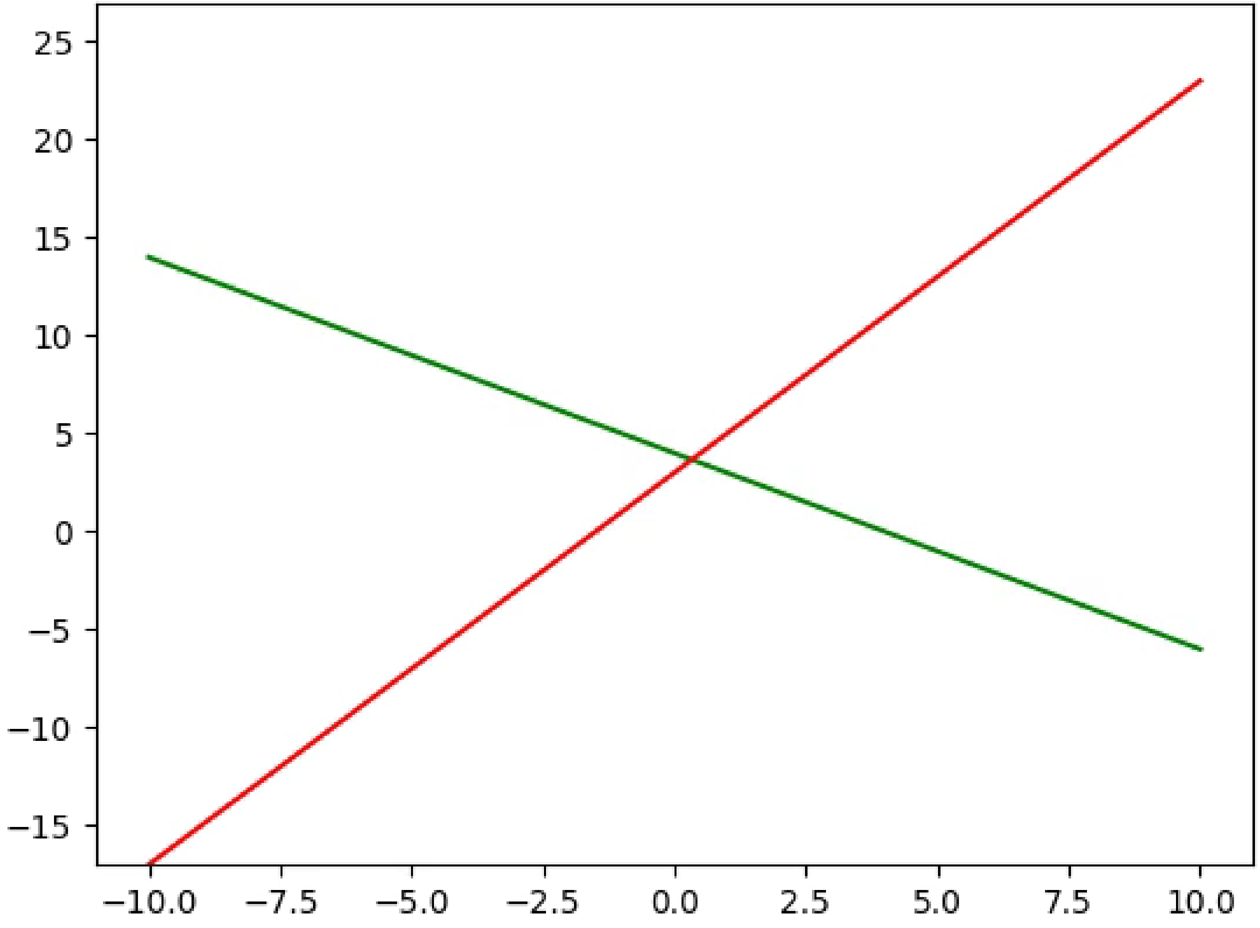

$\eta = 0.01$

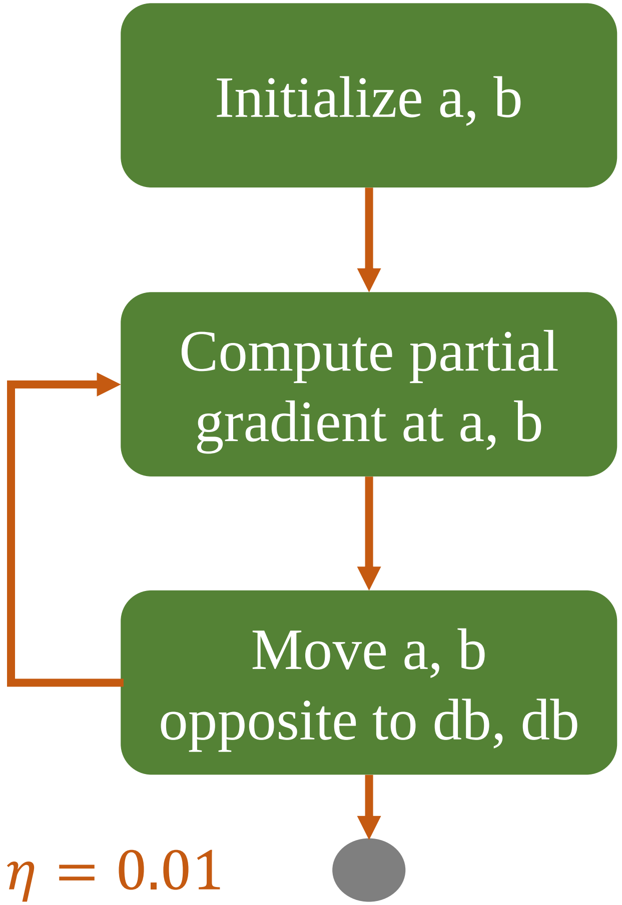
# Summary



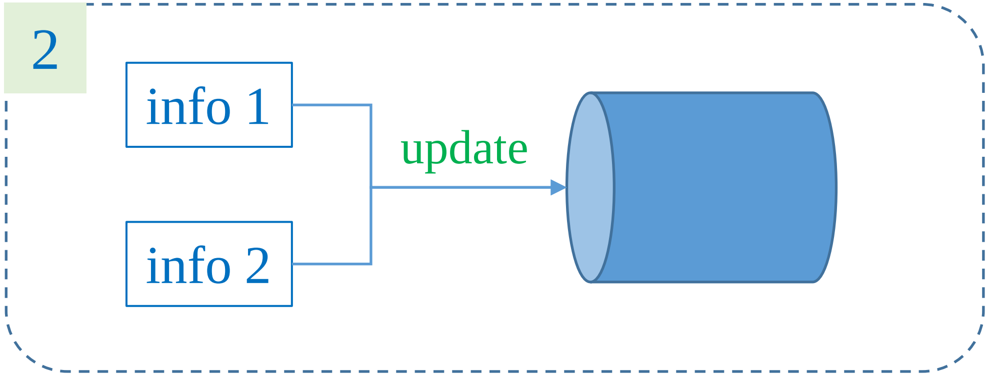

$$\frac{dg}{da} = \frac{dg}{df}\frac{df}{da} = 2x(f-y)$$

$$\frac{dg}{db} = \frac{dg}{df}\frac{df}{db} = 2(f-y)$$



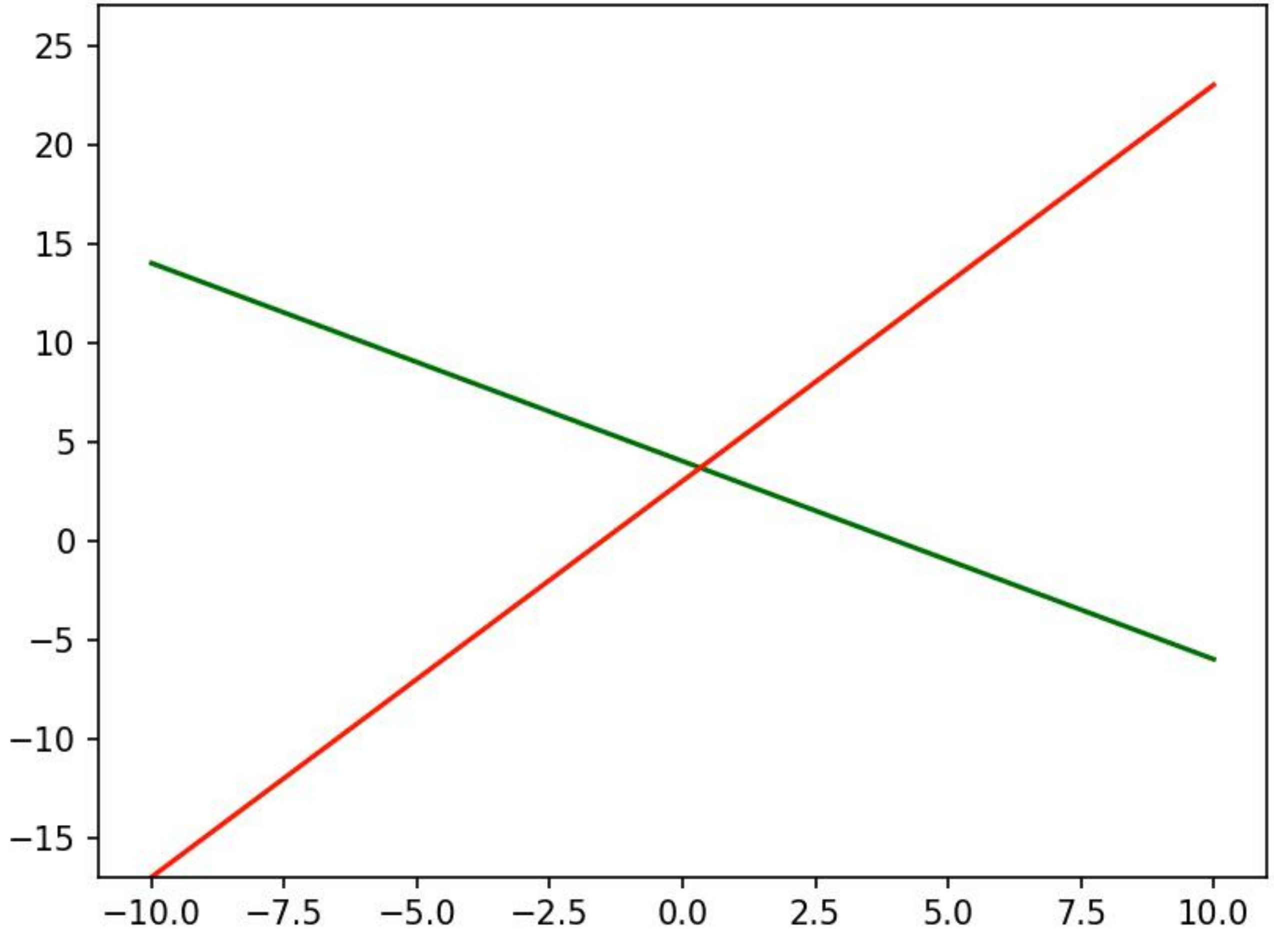



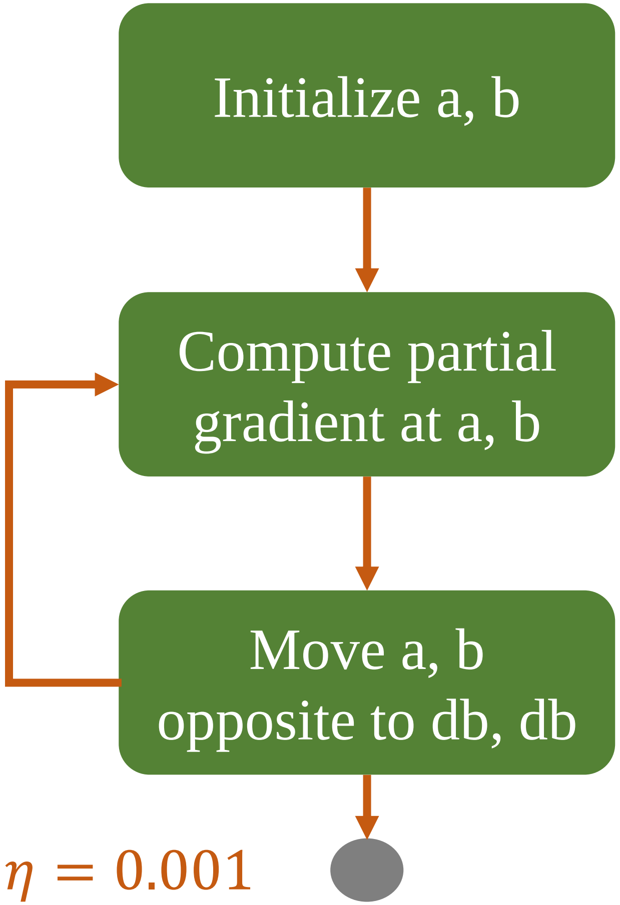
# Summary




$$\frac{dg}{da} = \frac{dg}{df}\frac{df}{da} = 2x(f-y)$$

$$\frac{dg}{db} = \frac{dg}{df}\frac{df}{db} = 2(f-y)$$




$\eta = 0.001$

# Outline

- Vectorization
- Optimiztion for 1+ samples
- Logistic Regression – Mini-batch
- Logistic Regression – Batch
- BCE and MSE Loss Functions
- Sigmoid and Tanh Function (Optional)

# Linear Regression (m-samples)

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

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

## ❖ Construct formulas

**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 |

$$\boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 1.5 & 0.2 \\ 1 & 4.1 & 1.3 \end{bmatrix}$$

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

2) Compute output $\hat{y}$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 1.5 & 0.2 \\ 1 & 4.1 & 1.3 \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \end{bmatrix}$$

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

$$\boldsymbol{z} = \begin{bmatrix} z^{(1)} \\ z^{(2)} \end{bmatrix} = \begin{bmatrix} w_1 x_1^{(1)} + w_2 x_2^{(1)} + b \\ w_1 x_1^{(2)} + w_2 x_2^{(2)} + b \end{bmatrix}$$

$$= \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \end{bmatrix} \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} = \boldsymbol{x}\boldsymbol{\theta} = \begin{bmatrix} 0.83 \\ 2.02 \end{bmatrix}$$

# Linear Regression (m-samples)

$$z = \theta^T x$$

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

## ❖ Construct formulas

**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 |

$$y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$x = \begin{bmatrix} 1 & 1.5 & 0.2 \\ 1 & 4.1 & 1.3 \end{bmatrix}$$

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

2) Compute output $\hat{y}$

$$x = \begin{bmatrix} 1 & 1.5 & 0.2 \\ 1 & 4.1 & 1.3 \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \end{bmatrix}$$

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

$$z = x\theta = \begin{bmatrix} 0.83 \\ 2.02 \end{bmatrix}$$

$$\hat{y} = \sigma(z) = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-z^{(1)}}} \\ \frac{1}{1+e^{-z^{(2)}}} \end{bmatrix} = \frac{1}{1+e^{-z}} = \begin{bmatrix} 0.69 \\ 0.88 \end{bmatrix}$$

Numpy perspective

# Linear Regression (m-samples)

## ❖ Construct formulas

**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 |

$$\boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 1.5 & 0.2 \\ 1 & 4.1 & 1.3 \end{bmatrix}$$

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

3) Compute loss

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

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{L^{(1)}\left(\hat{y}^{(1)}, y^{(1)}\right) + L^{(2)}\left(\hat{y}^{(2)}, y^{(2)}\right)}{\mathrm{m}}$$

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

$+$

$$L^{(2)}\left(\hat{y}^{(2)}, y^{(2)}\right) = -y^{(2)}\log\hat{y}^{(2)} - (1-y^{(2)})\log(1-\hat{y}^{(2)})$$

$\mathbf{y}^{\mathrm{T}}\log\hat{\mathbf{y}}$ $\quad$ $(1-\mathbf{y})^{\mathrm{T}}\log(1-\hat{\mathbf{y}})$

## 4) Compute derivative

sample 1

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

$$\frac{\partial L^{(1)}}{\partial w_1} = x_1^{(1)}(\hat{y}^{(1)} - y^{(1)})$$

$$\frac{\partial L^{(1)}}{\partial w_2} = x_2^{(1)}(\hat{y}^{(1)} - y^{(1)})$$

sample 2

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

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

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

$$\frac{\partial L}{\partial b} = \frac{\frac{\partial L^{(1)}}{\partial b} + \frac{\partial L^{(2)}}{\partial b}}{m} = \frac{(\hat{y}^{(1)} - y^{(1)}) + (\hat{y}^{(2)} - y^{(2)})}{m}$$

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

$$= \frac{1}{m}\begin{bmatrix} x_0^{(1)} & x_0^{(2)} \end{bmatrix}\begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \hat{y}^{(2)} - y^{(2)} \end{bmatrix}$$

$x_0^{(1)} = 1$

$x_0^{(2)} = 1$

$$\frac{\partial L}{\partial w_1} = \frac{\frac{\partial L^{(1)}}{\partial w_1} + \frac{\partial L^{(2)}}{\partial w_1}}{m} = \frac{x_1^{(1)}(\hat{y}^{(1)} - y^{(1)}) + x_1^{(2)}(\hat{y}^{(2)} - y^{(2)})}{m}$$

$$= \frac{1}{m}\begin{bmatrix} x_1^{(1)} & x_1^{(2)} \end{bmatrix}\begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \hat{y}^{(2)} - y^{(2)} \end{bmatrix}$$

$$\frac{\partial L}{\partial w_2} = \frac{\frac{\partial L^{(1)}}{\partial w_2} + \frac{\partial L^{(2)}}{\partial w_2}}{m} = \frac{x_2^{(1)}(\hat{y}^{(1)} - y^{(1)}) + x_2^{(2)}(\hat{y}^{(2)} - y^{(2)})}{m}$$

$$= \frac{1}{m}\begin{bmatrix} x_2^{(1)} & x_2^{(2)} \end{bmatrix}\begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \hat{y}^{(2)} - y^{(2)} \end{bmatrix}$$

## 4) Compute derivative

**sample 1**

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

$$\frac{\partial L^{(1)}}{\partial w_1} = x_1^{(1)}(\hat{y}^{(1)} - y^{(1)})$$

$$\frac{\partial L^{(1)}}{\partial w_2} = x_2^{(1)}(\hat{y}^{(1)} - y^{(1)})$$

**sample 2**

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

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

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

$$\frac{\partial L}{\partial b} = \frac{1}{m}\begin{bmatrix} x_0^{(1)} & x_0^{(2)} \end{bmatrix}\begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \hat{y}^{(2)} - y^{(2)} \end{bmatrix}$$

$$\frac{\partial L}{\partial w_1} = \frac{1}{m}\begin{bmatrix} x_1^{(1)} & x_1^{(2)} \end{bmatrix}\begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \hat{y}^{(2)} - y^{(2)} \end{bmatrix}$$

$$\frac{\partial L}{\partial w_2} = \frac{1}{m}\begin{bmatrix} x_2^{(1)} & x_2^{(2)} \end{bmatrix}\begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \hat{y}^{(2)} - y^{(2)} \end{bmatrix}$$

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{L} = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \end{bmatrix} = \frac{1}{m}\begin{bmatrix} x_0^{(1)} & x_0^{(2)} \\ x_1^{(1)} & x_1^{(2)} \\ x_2^{(1)} & x_2^{(2)} \end{bmatrix}\begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \hat{y}^{(2)} - y^{(2)} \end{bmatrix}$$

$\left(\boldsymbol{x}^{(1)}\right)^T$ $\left(\boldsymbol{x}^{(2)}\right)^T$ $\hat{\mathbf{y}}$ $\mathbf{y}$

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

## 5) Update parameters

**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 |

$$\boldsymbol{x} = \begin{bmatrix} 1 & 1.5 & 0.2 \\ 1 & 4.1 & 1.3 \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

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

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{L} = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \end{bmatrix} = \frac{1}{m} \boldsymbol{x}^T (\hat{\boldsymbol{y}} - \boldsymbol{y})$$

$$\underbrace{\begin{matrix} b \\ w_1 \\ w_2 \end{matrix}}_{\boldsymbol{\theta}} = \underbrace{\begin{matrix} b \\ w_1 \\ w_2 \end{matrix}}_{\boldsymbol{\theta}} \begin{matrix} -\ \eta \\ -\ \eta \\ -\ \eta \end{matrix} \underbrace{\begin{matrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \end{matrix}}_{\nabla_{\boldsymbol{\theta}} \boldsymbol{L}}$$

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

# Logistic Regression - Minibatch

1) Pick m samples from training data

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

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

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

3) Compute loss

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

4) Compute derivative

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

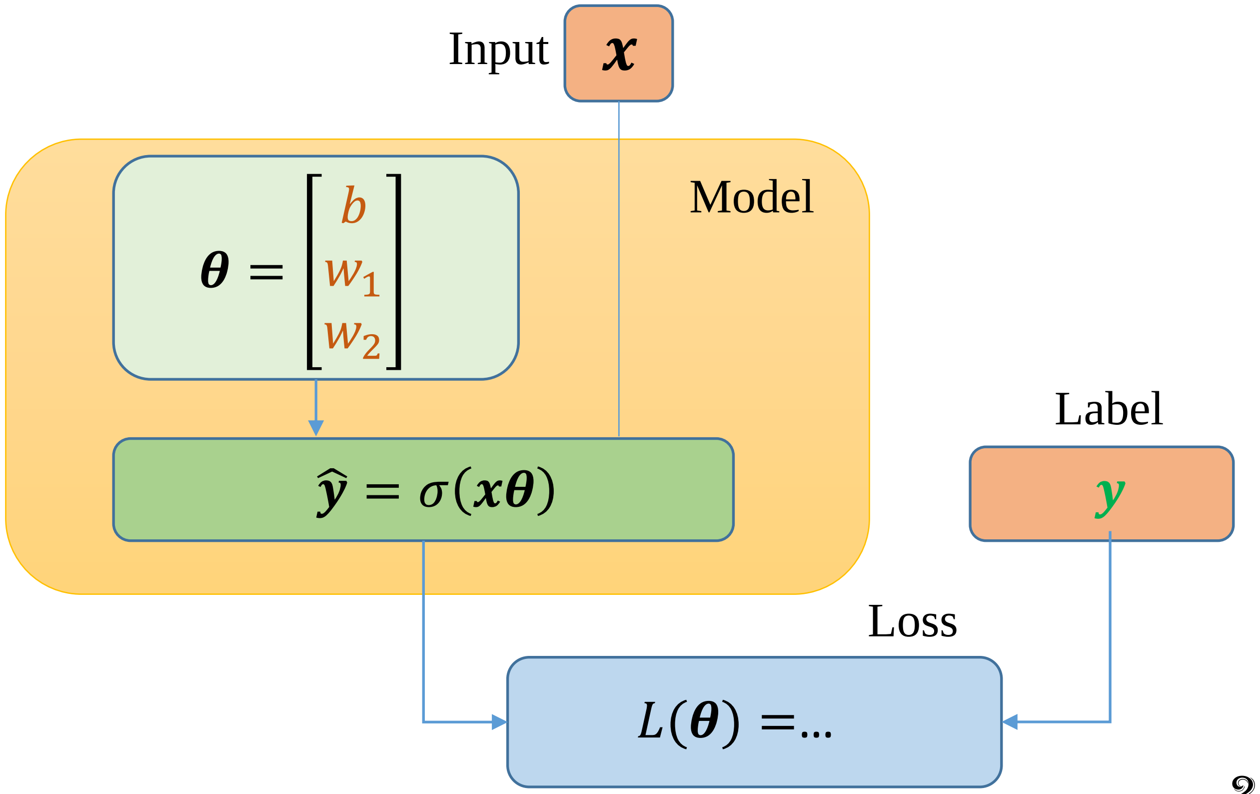
5) Update parameters

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

$\eta$ is learning rate

Mini-batch m=2

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



## 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 m samples from training data

Mini-batch m=2

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

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

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

3) Compute loss

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

4) Compute derivative

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

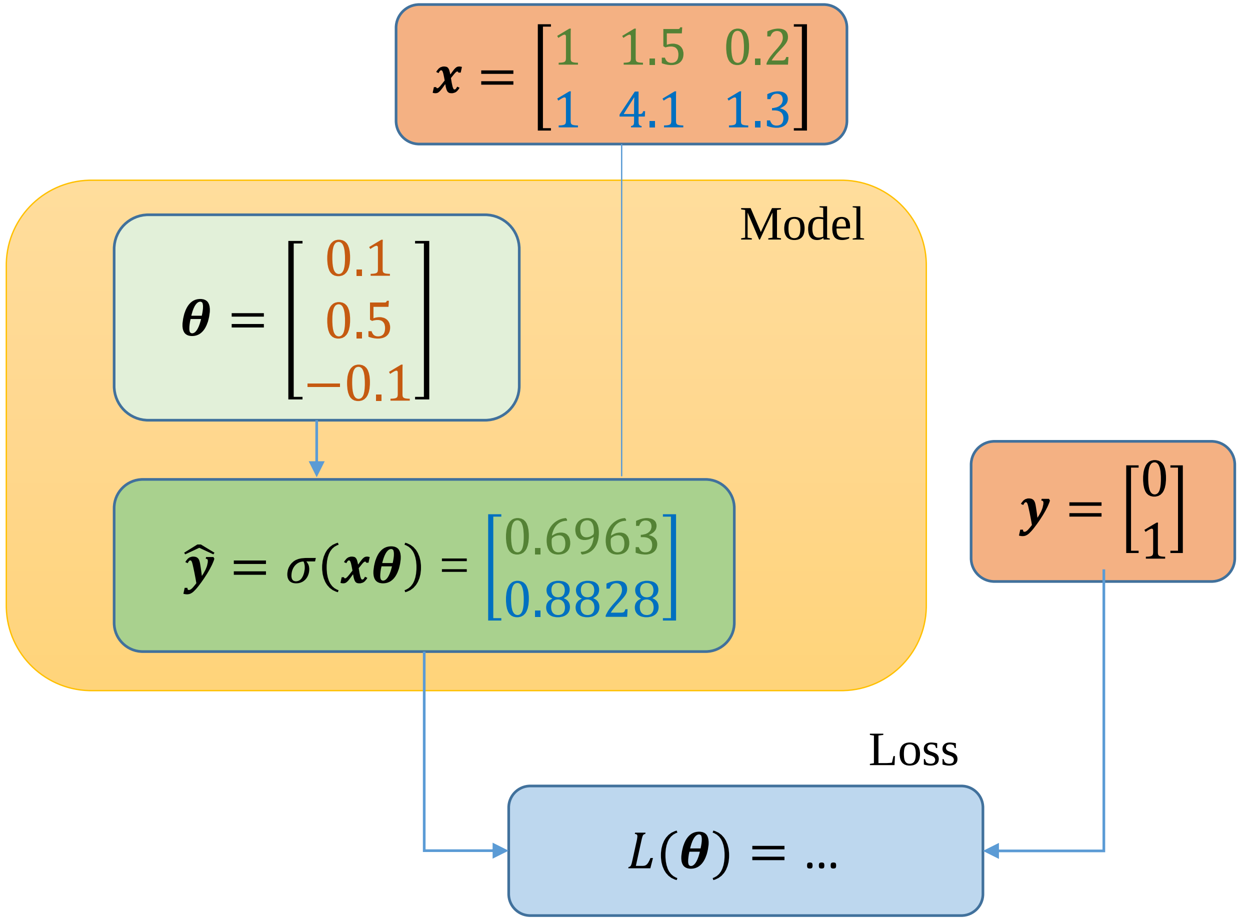
5) Update parameters

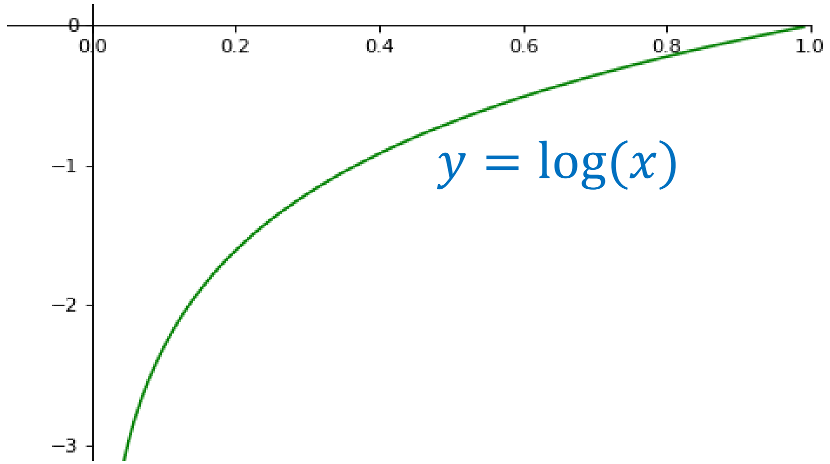
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}} L$$

2

$$\hat{\boldsymbol{y}} = \sigma(\boldsymbol{x}\boldsymbol{\theta}) = \sigma\left(\begin{bmatrix}1 & 1.5 & 0.2\\ 1 & 4.1 & 1.3\end{bmatrix}\begin{bmatrix}0.1\\ 0.5\\ -0.1\end{bmatrix}\right)$$

$$= \sigma\left(\begin{bmatrix}0.83\\ 2.02\end{bmatrix}\right) = \begin{bmatrix}0.6963\\ 0.8828\end{bmatrix}$$






3

$$L(\boldsymbol{\theta}) = \frac{1}{\mathrm{m}}\left(-[0 \quad 1]\begin{bmatrix}\log 0.6963\\ \log 0.8828\end{bmatrix} - [1 \quad 0]\begin{bmatrix}\log(1-0.6963)\\ \log(1-0.8828)\end{bmatrix}\right)$$

$$= \frac{1}{\mathrm{m}}(-\log 0.8828 - \log(1-0.6963))$$

$$= \frac{0.1246 + 1.1917}{\mathrm{m}} = 0.65815$$

1) Pick m samples from training data

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

Mini-batch m=2

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

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

3) Compute loss

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

4) Compute derivative

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

5) Update parameters

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

$$\boldsymbol{x} = \begin{bmatrix}1 & 1.5 & 0.2\\ 1 & 4.1 & 1.3\end{bmatrix}$$

Model

$$\boldsymbol{\theta} = \begin{bmatrix}0.1\\ 0.5\\ -0.1\end{bmatrix}$$

$$\hat{\boldsymbol{y}} = \sigma(\boldsymbol{x\theta}) = \begin{bmatrix}0.6963\\ 0.8828\end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix}0\\ 1\end{bmatrix}$$

Loss

$$L(\boldsymbol{\theta}) = 0.65815$$

## 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 m samples from training data

Mini-batch m=2

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

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

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

3) Compute loss

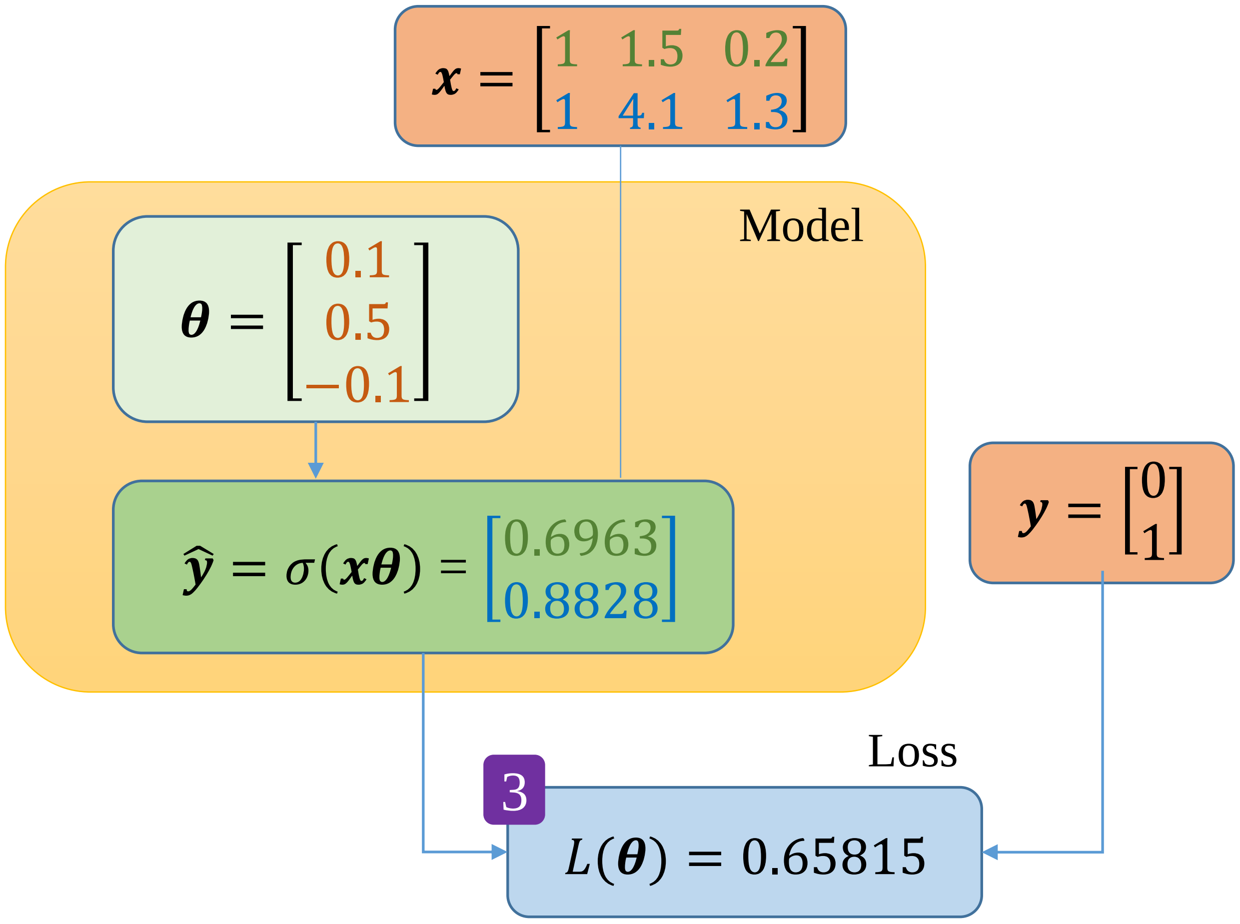
$$L(\hat{\mathbf{y}}, \boldsymbol{y}) = \frac{1}{\mathrm{m}}\left(-\mathbf{y}^{\mathrm{T}}\log\hat{\mathbf{y}} - (\mathbf{1}-\mathbf{y})^{\mathrm{T}}\log(\mathbf{1}-\hat{\mathbf{y}})\right)$$

4) Compute derivative

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

5) Update parameters

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




4

$$\nabla_{\boldsymbol{\theta}} L = \frac{1}{\mathrm{m}}\begin{bmatrix} 1 & 1 \\ 1.5 & 4.1 \\ 0.2 & 1.3 \end{bmatrix}\left(\begin{bmatrix} 0.6963 \\ 0.8828 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)$$

$$= \frac{1}{\mathrm{m}}\begin{bmatrix} 1 & 1 \\ 1.5 & 4.1 \\ 0.2 & 1.3 \end{bmatrix}\begin{bmatrix} 0.6963 \\ -0.1171 \end{bmatrix} = \begin{bmatrix} 0.28961 \\ 0.28217 \\ -0.0064 \end{bmatrix}$$

## 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 m samples from training data

Mini-batch m=2

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

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

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

3) Compute loss

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

4) Compute derivative

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

5) Update parameters

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

$$\boldsymbol{x} = \begin{bmatrix} 1 & 1.5 & 0.2 \\ 1 & 4.1 & 1.3 \end{bmatrix}$$

Model

$$\boldsymbol{\theta} = \begin{bmatrix} 0.1 \\ 0.5 \\ -0.1 \end{bmatrix}$$

$$\hat{\boldsymbol{y}} = \sigma(\boldsymbol{x\theta}) = \begin{bmatrix} 0.6963 \\ 0.8828 \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Loss

3

$$L(\boldsymbol{\theta}) = 0.65815$$

4

$$\nabla_{\boldsymbol{\theta}} L = \begin{bmatrix} 0.28961 \\ 0.28217 \\ -0.0064 \end{bmatrix}$$

5

$$\boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}} = \begin{bmatrix} 0.1 \\ 0.5 \\ -0.1 \end{bmatrix} - \eta \begin{bmatrix} 0.28961 \\ 0.28217 \\ -0.0064 \end{bmatrix} = \begin{bmatrix} 0.0971 \\ 0.4971 \\ -0.099 \end{bmatrix}$$

# Outline

- Vectorization
- Optimiztion for 1+ samples
- Logistic Regression – Mini-batch
- Logistic Regression – Batch
- BCE and MSE Loss Functions
- Sigmoid and Tanh Function (Optional)

# Logistic Regression - Batch

| 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 all the samples from training data

2) Compute output $\hat{y}$

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

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

3) Compute loss

$$L(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{\mathrm{N}}\left(-\mathbf{y}^{\mathrm{T}}\log\hat{\mathbf{y}} - (\mathbf{1}-\mathbf{y})^{\mathrm{T}}\log(\mathbf{1}-\hat{\mathbf{y}})\right)$$

4) Compute derivative

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

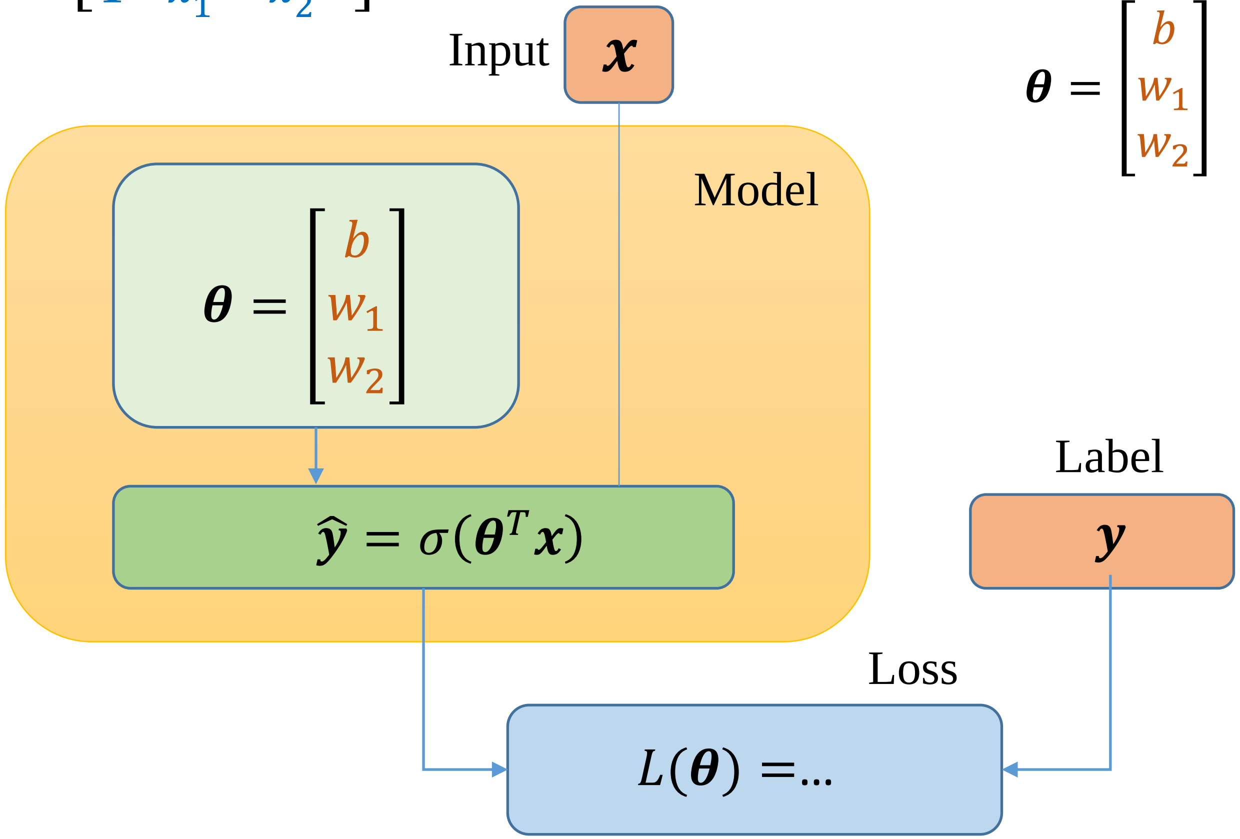
5) Update parameters

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

$\eta$ is learning rate

$$\boldsymbol{x} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \\ 1 & x_1^{(3)} & x_2^{(3)} \\ 1 & x_1^{(4)} & x_2^{(4)} \end{bmatrix}$$

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



# Logistic Regression - Batch

**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 |

$$\boldsymbol{x} = \begin{bmatrix} 1 & 1.4 & 0.2 \\ 1 & 1.5 & 0.2 \\ 1 & 3.0 & 1.1 \\ 1 & 4.1 & 1.3 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}$$

1) Pick all the samples from training data

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

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

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

3) Compute loss

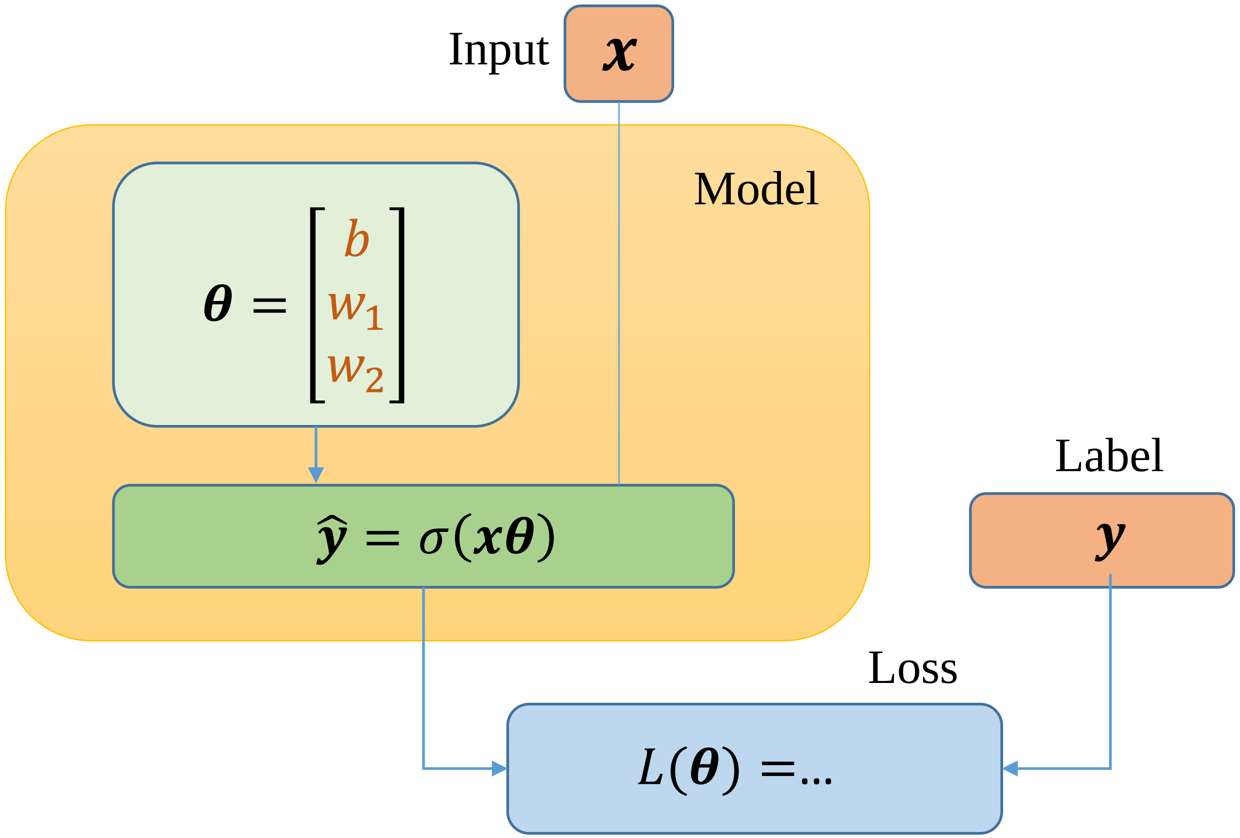
$$L(\hat{\mathbf{y}}, \boldsymbol{y}) = \frac{1}{\mathrm{N}}\left(-\mathbf{y}^{\mathrm{T}}\log\hat{\mathbf{y}} - (\mathbf{1}-\mathbf{y})^{\mathrm{T}}\log(\mathbf{1}-\hat{\mathbf{y}})\right)$$

4) Compute derivative

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

5) Update parameters

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



## 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 |

$$\boldsymbol{x} = \begin{bmatrix} 1 & 1.4 & 0.2 \\ 1 & 1.5 & 0.2 \\ 1 & 3.0 & 1.1 \\ 1 & 4.1 & 1.3 \end{bmatrix}$$

1) Pick all the samples from training data

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

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

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

3) Compute loss

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

4) Compute derivative

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

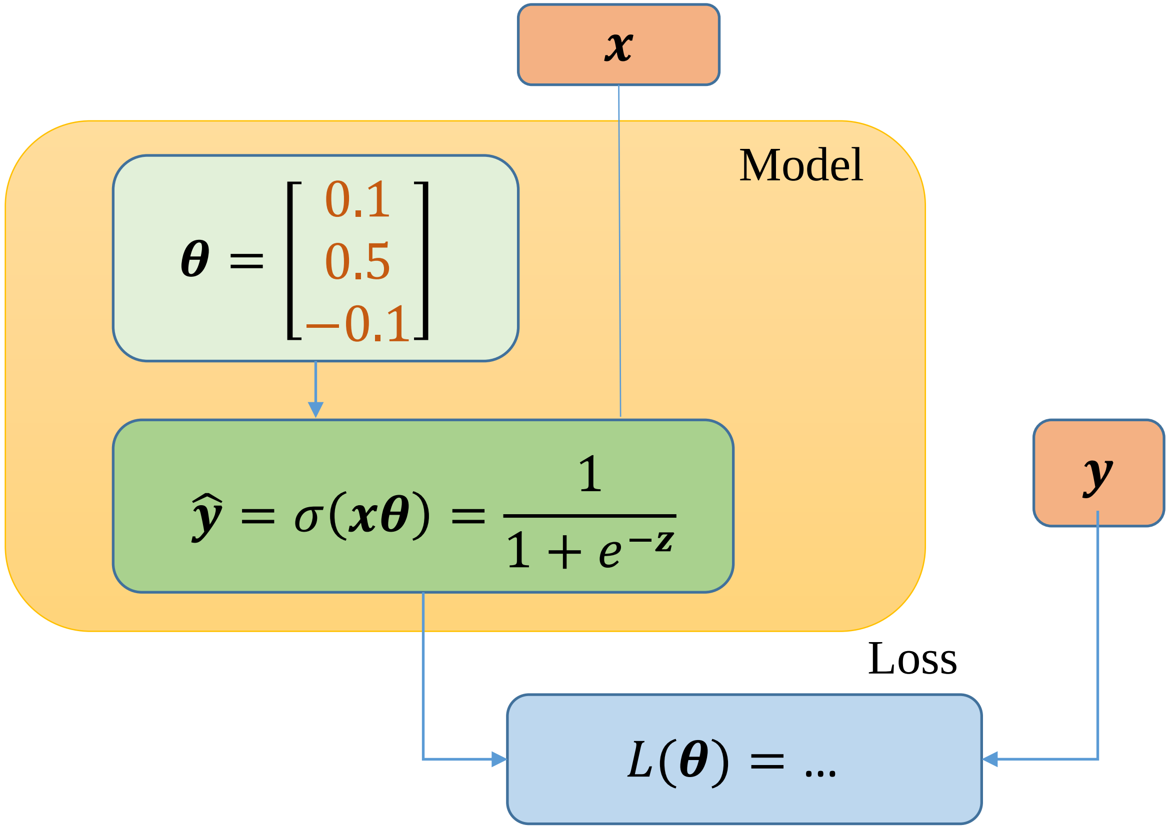
5) Update parameters

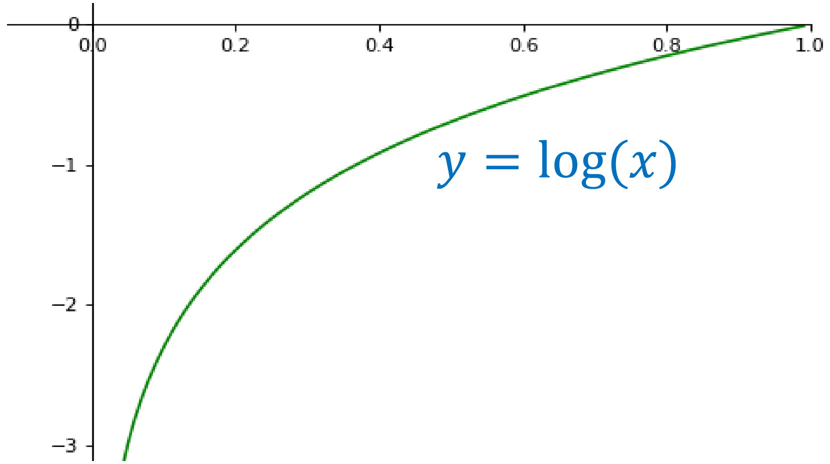
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} L$$

2

$$\hat{\boldsymbol{y}} = \sigma(\boldsymbol{x\theta}) = \sigma\left(\begin{bmatrix} 1 & 1.4 & 0.2 \\ 1 & 1.5 & 0.2 \\ 1 & 3.0 & 1.1 \\ 1 & 4.1 & 1.3 \end{bmatrix}\begin{bmatrix} 0.1 \\ 0.5 \\ -0.1 \end{bmatrix}\right)$$

$$= \sigma\left(\begin{bmatrix} 0.78 \\ 0.83 \\ 1.49 \\ 2.02 \end{bmatrix}\right) = \begin{bmatrix} 0.6856 \\ 0.6963 \\ 0.8160 \\ 0.8828 \end{bmatrix}$$






3

$$L(\boldsymbol{\theta}) = \frac{1}{\text{N}}\left\{-\begin{bmatrix}0\\0\\1\\1\end{bmatrix}^{\text{T}} \log\begin{bmatrix}0.6856\\0.6963\\0.8160\\0.8828\end{bmatrix} - \left(\begin{bmatrix}1\\1\\1\\1\end{bmatrix}-\begin{bmatrix}0\\0\\1\\1\end{bmatrix}\right)^{T} \log\left(\mathbf{1}-\begin{bmatrix}0.6856\\0.6963\\0.8160\\0.8828\end{bmatrix}\right)\right\}$$

1) Pick all the samples from training data

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

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

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

3) Compute loss

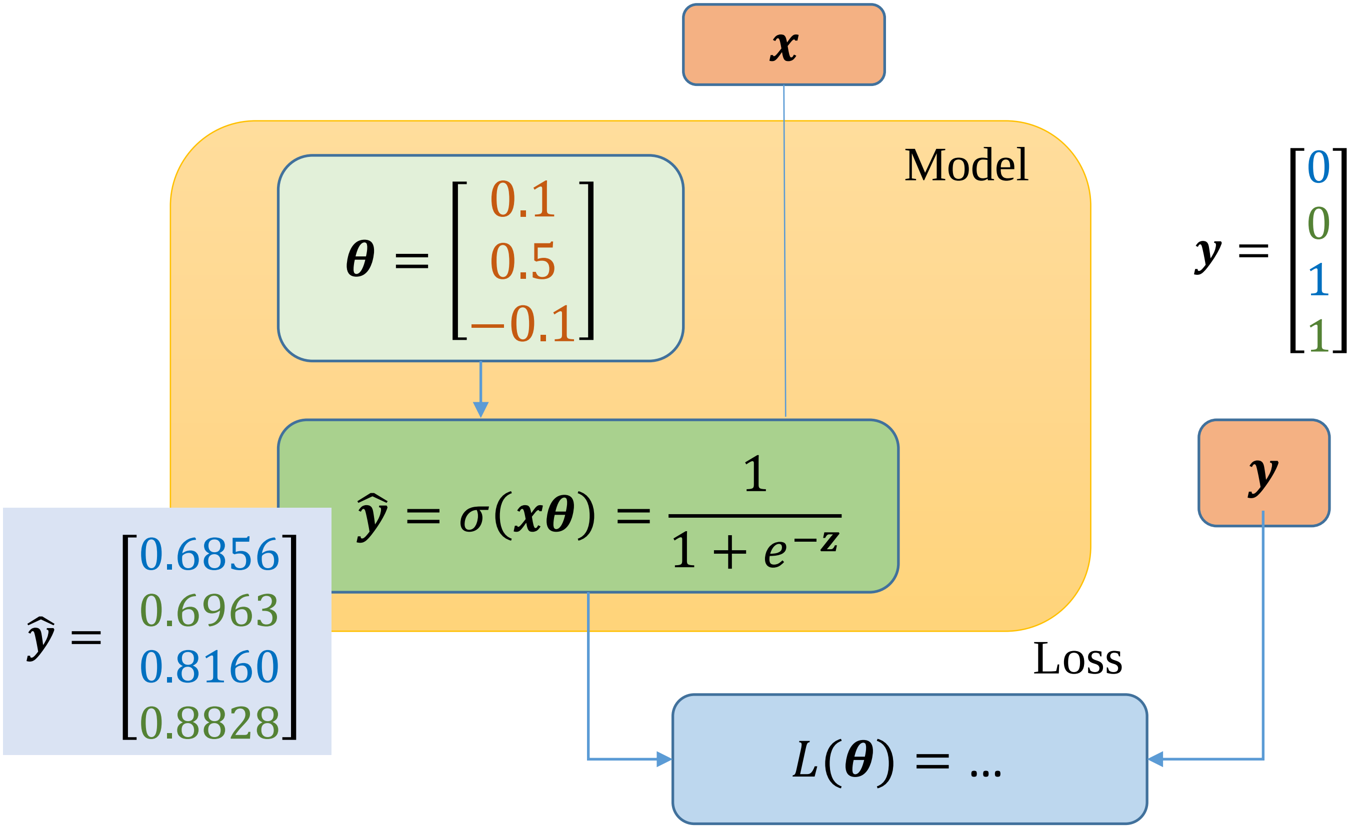
$$L(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{\text{N}}\left(-\mathbf{y}^{\text{T}}\log\hat{\mathbf{y}} - (\mathbf{1}-\mathbf{y})^{\text{T}}\log(\mathbf{1}-\hat{\mathbf{y}})\right)$$

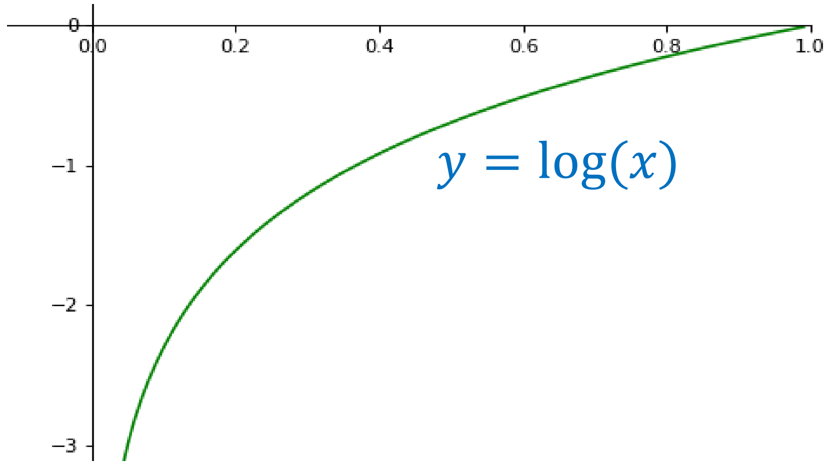
4) Compute derivative

$$\nabla_{\boldsymbol{\theta}} L = \frac{1}{\text{N}}\boldsymbol{x}^{\text{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

5) Update parameters

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






3

$$L(\boldsymbol{\theta}) = \frac{1}{\mathrm{N}}\left\{-\begin{bmatrix}0\\0\\1\\1\end{bmatrix}^{\mathrm{T}}\log\left(\begin{bmatrix}0.6856\\0.6963\\0.8160\\0.8828\end{bmatrix}\right)-\begin{bmatrix}1\\1\\0\\0\end{bmatrix}^{T}\log\left(\begin{bmatrix}0.3144\\0.3037\\0.1840\\0.1172\end{bmatrix}\right)\right\}$$

$$= \frac{1}{\mathrm{N}}(-\log 0.8160 - \log 0.8828 - \log 0.3144 - \log 0.3037)$$

$$= 0.6691$$

1) Pick all the samples from training data

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

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

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

3) Compute loss

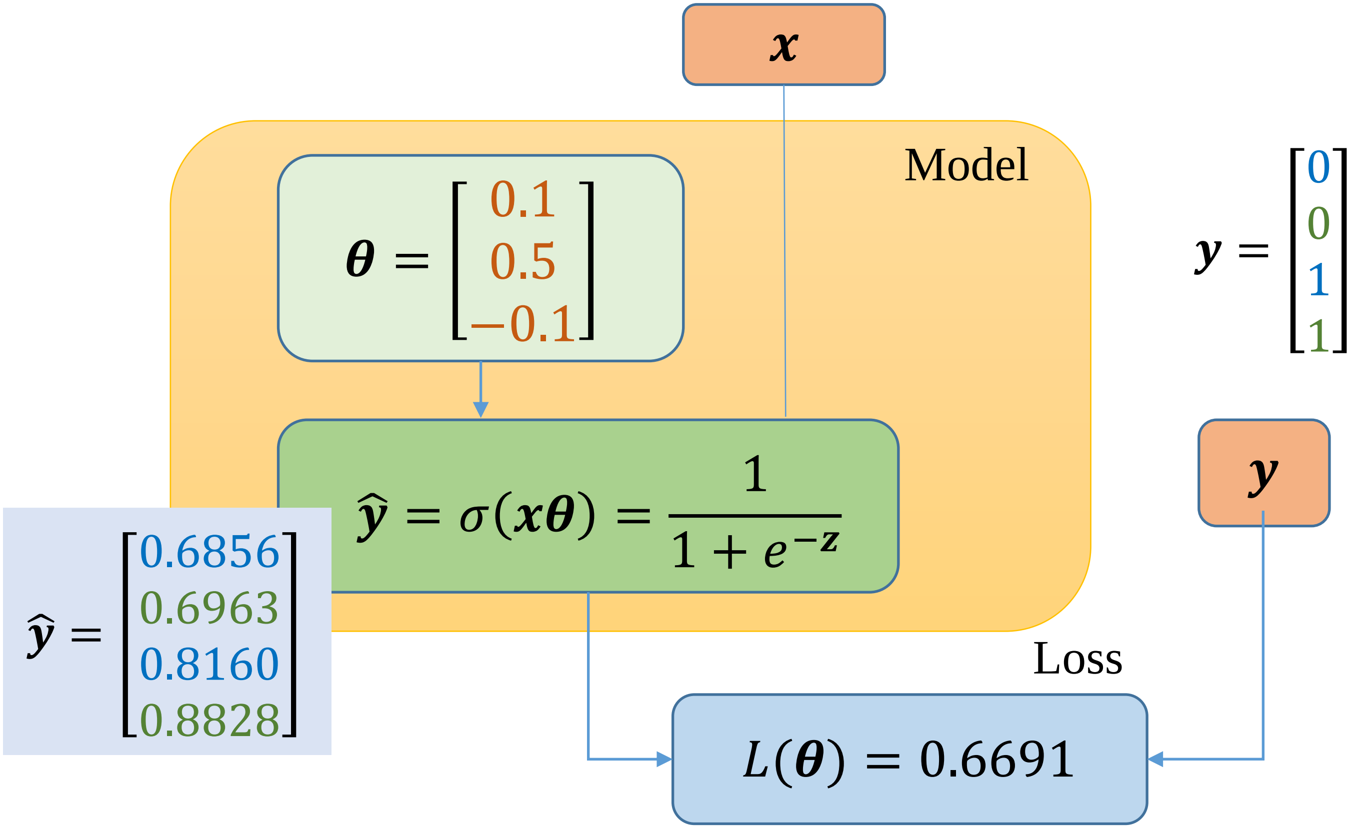
$$L(\hat{\mathbf{y}}, \boldsymbol{y}) = \frac{1}{\mathrm{N}}\left(-\mathbf{y}^{\mathrm{T}}\log\hat{\mathbf{y}} - (\mathbf{1}-\mathbf{y})^{\mathrm{T}}\log(\mathbf{1}-\hat{\mathbf{y}})\right)$$

4) Compute derivative

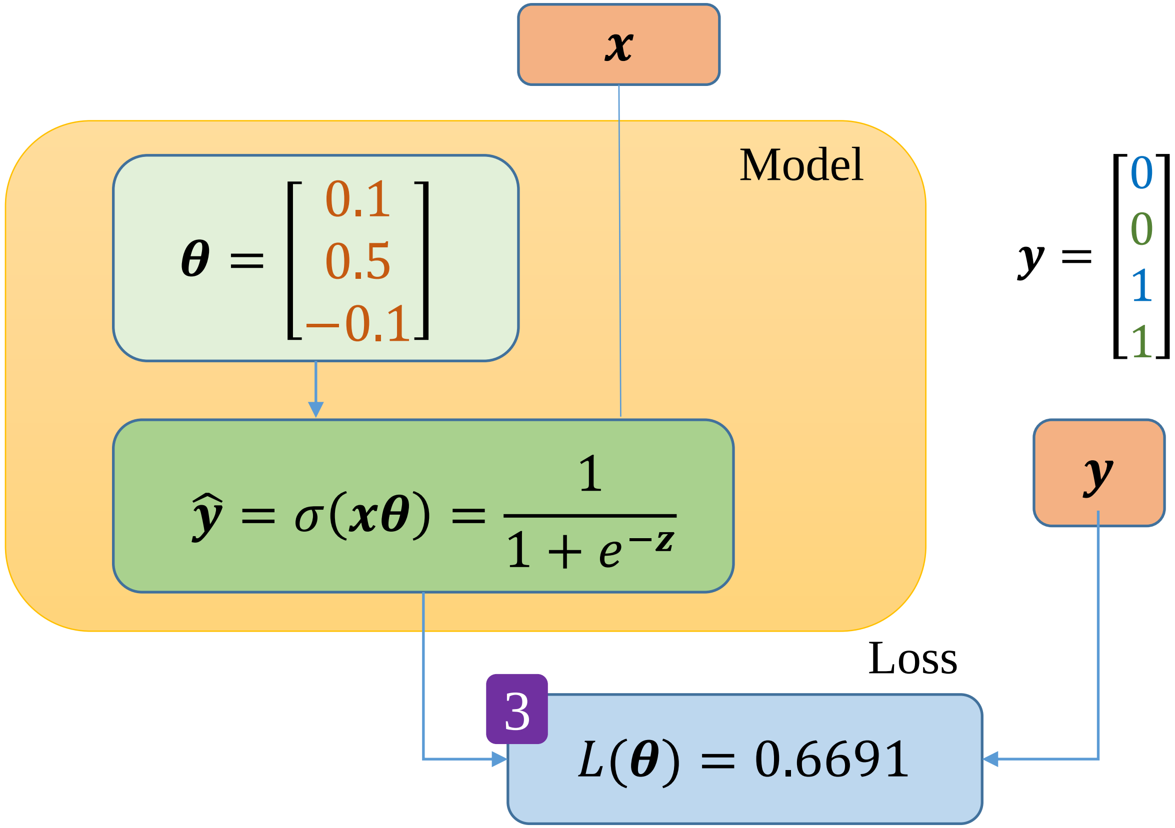
$$\nabla_{\boldsymbol{\theta}} L = \frac{1}{\mathrm{N}}\boldsymbol{x}^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

5) Update parameters

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



$$\boldsymbol{x} = \begin{bmatrix} 1 & 1.4 & 0.2 \\ 1 & 1.5 & 0.2 \\ 1 & 3.0 & 1.1 \\ 1 & 4.1 & 1.3 \end{bmatrix} \qquad \hat{\boldsymbol{y}} = \begin{bmatrix} 0.6856 \\ 0.6963 \\ 0.8160 \\ 0.8828 \end{bmatrix}$$




1) Pick all the samples from training data

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

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

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

3) Compute loss

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

4) Compute derivative

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

5) Update parameters

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

4

$$\nabla_{\boldsymbol{\theta}} L = \frac{1}{\mathrm{N}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1.4 & 1.5 & 3.0 & 4.1 \\ 0.2 & 0.2 & 1.1 & 1.3 \end{bmatrix} \left( \begin{bmatrix} 0.6856 \\ 0.6963 \\ 0.8160 \\ 0.8828 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \right)$$

$$= \frac{1}{\mathrm{N}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1.4 & 1.5 & 3.0 & 4.1 \\ 0.2 & 0.2 & 1.1 & 1.3 \end{bmatrix} \begin{bmatrix} 0.6856 \\ 0.6963 \\ -0.184 \\ -0.117 \end{bmatrix} = \begin{bmatrix} 0.2702 \\ 0.2431 \\ -0.019 \end{bmatrix}$$

## 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 all the samples from training data

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

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

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

3) Compute loss

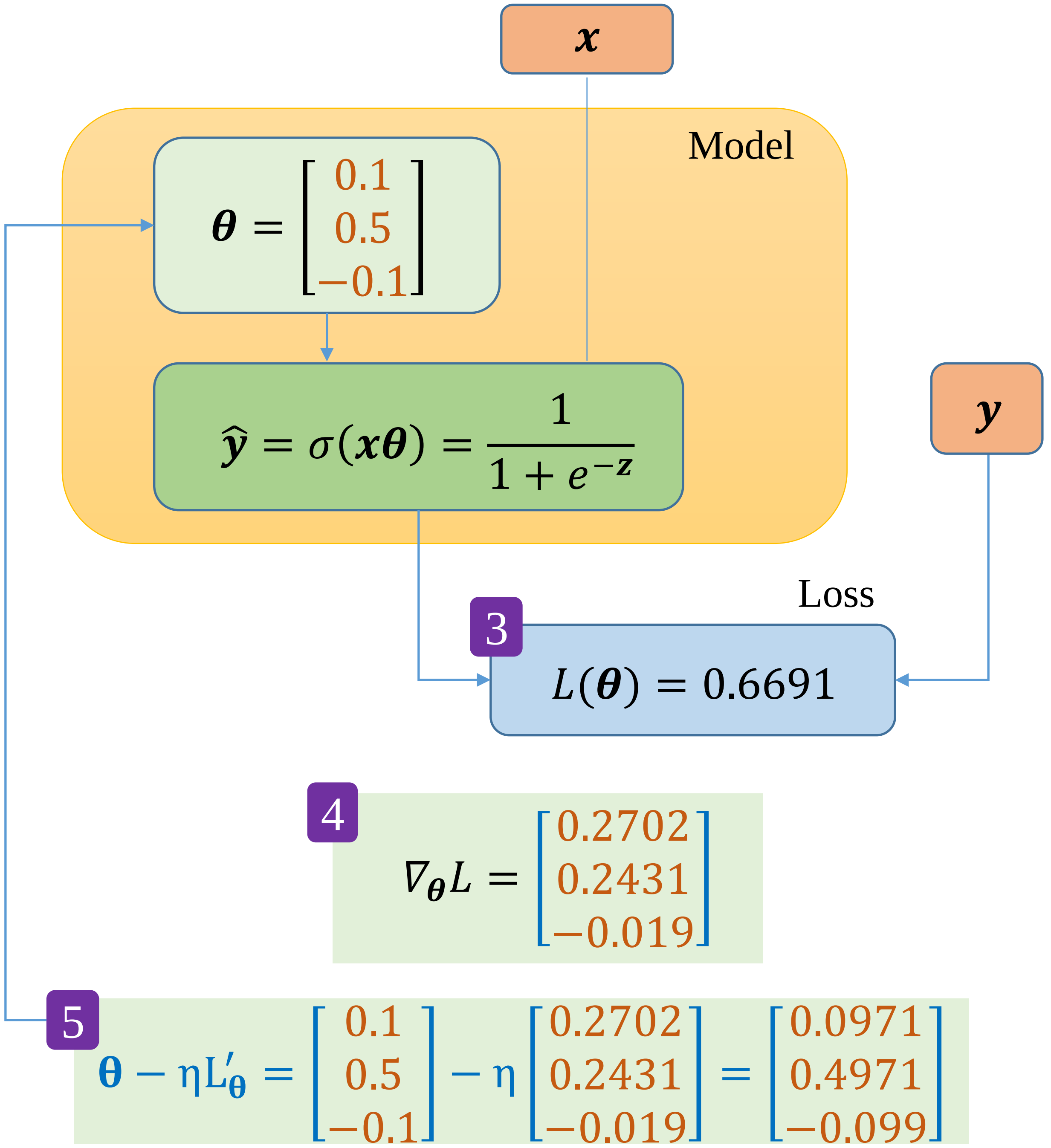
$$L(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{\mathrm{N}}\left(-\mathbf{y}^{\mathrm{T}}\log\hat{\mathbf{y}} - (\mathbf{1}-\mathbf{y})^{\mathrm{T}}\log(\mathbf{1}-\hat{\mathbf{y}})\right)$$

4) Compute derivative

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

5) Update parameters

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



# Outline

- Vectorization
- Optimiztion for 1+ samples
- Logistic Regression – Mini-batch
- Logistic Regression – Batch
- BCE and MSE Loss Functions
- Sigmoid and Tanh Function (Optional)

# Hessian Matrices

## ❖ Definition

The Hessian matrix or Hessian is a square matrix of second-order partial derivatives of a scalar-valued function

https://en.wikipedia.org/wiki/Hessian_matrix

Given $f(x, y)$

$f: R^2 \rightarrow R$

$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

Given $f(x, y) = x^2 + 2x^2y + y^3$

$$\frac{\partial f}{\partial x} = 2x + 4xy$$

$$\frac{\partial f}{\partial y} = 2x^2 + 3y^2$$

$$H_f = \begin{bmatrix} 2 + 4y & 4x \\ 4x & 6y \end{bmatrix}$$

# Binary Cross-entropy

## ❖ Convex function

**Model and Loss**

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

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

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

**Derivative**

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

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

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

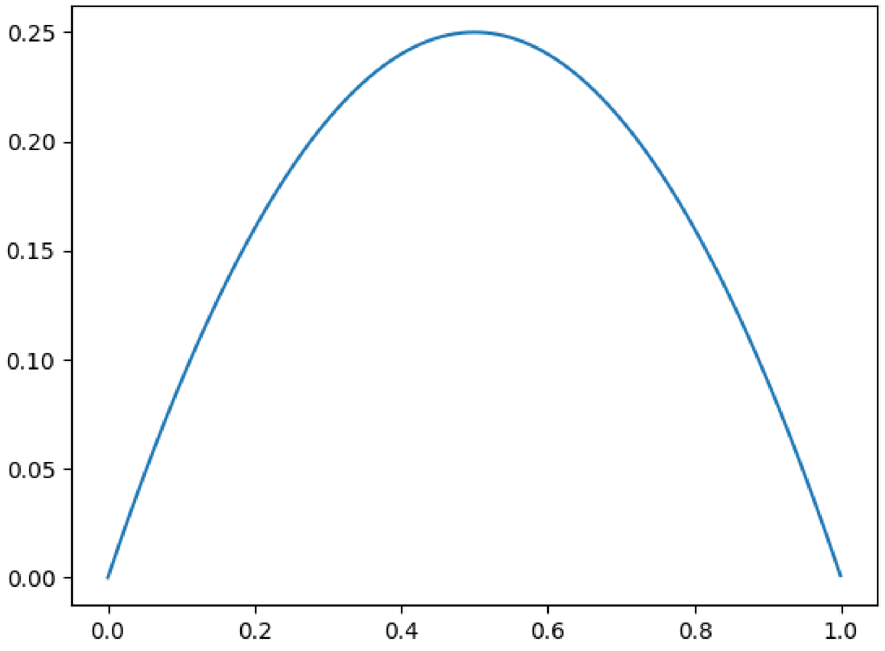
$$\frac{\partial z}{\partial \theta_i} = x_i$$

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

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

$$\frac{\partial^2 L}{\partial \theta_i^2} = \frac{\partial}{\partial \theta_i}[x_i(\hat{y}-y)] = x_i^2(\hat{y}-\hat{y}^2) \geq 0$$

$$x_i^2 \geq 0 \qquad \hat{y}-\hat{y}^2 \in \left[0, \frac{1}{4}\right]$$



# Logistic Regression-MSE

## ❖ Construct loss

### Model and Loss

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

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

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

### Derivative

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

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

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

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

$$\frac{\partial L}{\partial \theta_i} = 2x_i(\hat{y} - y)\hat{y}(1 - \hat{y})$$

# Mean Squared Error

**Model and Loss**

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

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

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

**Derivative**

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

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y) \qquad \frac{\partial z}{\partial \theta_i} = x_i$$

$$\frac{\partial L}{\partial \theta_i} = 2x_i(\hat{y} - y)\hat{y}(1 - \hat{y})$$

$$\frac{\partial L}{\partial \theta_i} = 2x_i(\hat{y} - y)\hat{y}(1 - \hat{y}) = 2x_i\left(-\hat{y}^3 + \hat{y}^2 - y\hat{y} + y\hat{y}^2\right)$$

$$\begin{aligned}
\frac{\partial^2 L}{\partial \theta_i^2} &= \frac{\partial}{\partial \theta_i}[2x_i(-\hat{y}^3 + \hat{y}^2 - y\hat{y} + y\hat{y}^2)] \\
&= 2x_i\left[-3\hat{y}^2 x_i \hat{y}(1 - \hat{y}) + 2x_i\hat{y}\hat{y}(1 - \hat{y}) - yx_i\hat{y}(1 - \hat{y}) + 2x_i y\hat{y}\hat{y}(1 - \hat{y})\right] \\
&= 2x_i^2\hat{y}(1 - \hat{y})\left[-3\hat{y}^2 + 2\hat{y} - y + 2y\hat{y}\right]
\end{aligned}$$

# Mean Squared Error

$$\frac{\partial^2 L}{\partial \theta_i^2} = 2x_i^2 \hat{y}(1-\hat{y})\left[-3\hat{y}^2 + 2\hat{y} - y + 2y\hat{y}\right]$$

$x_i^2 \geq 0$

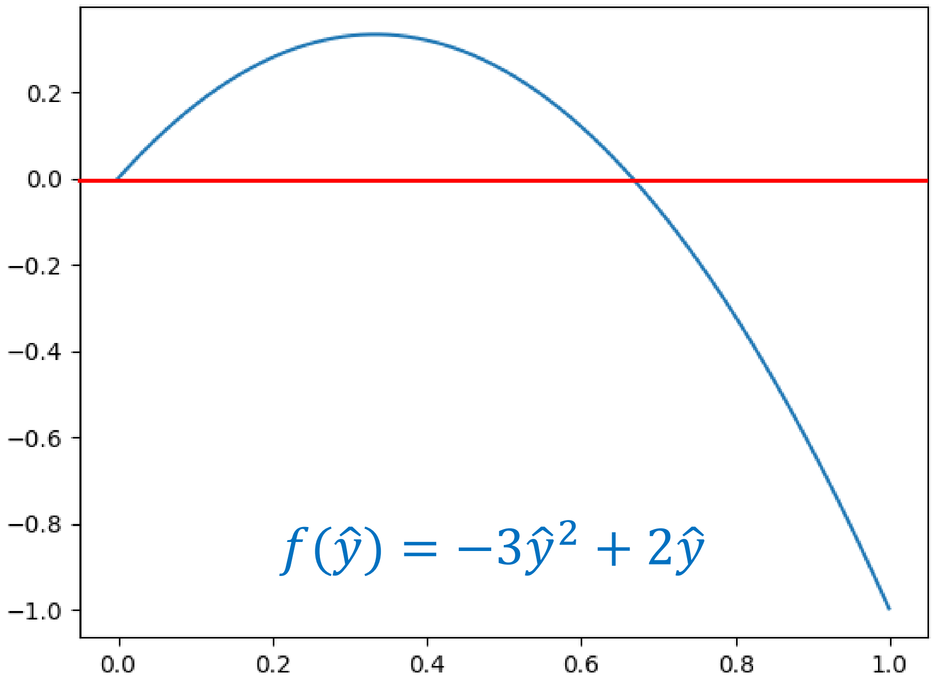
$\hat{y}(1-\hat{y}) \in \left[0, \frac{1}{4}\right]$

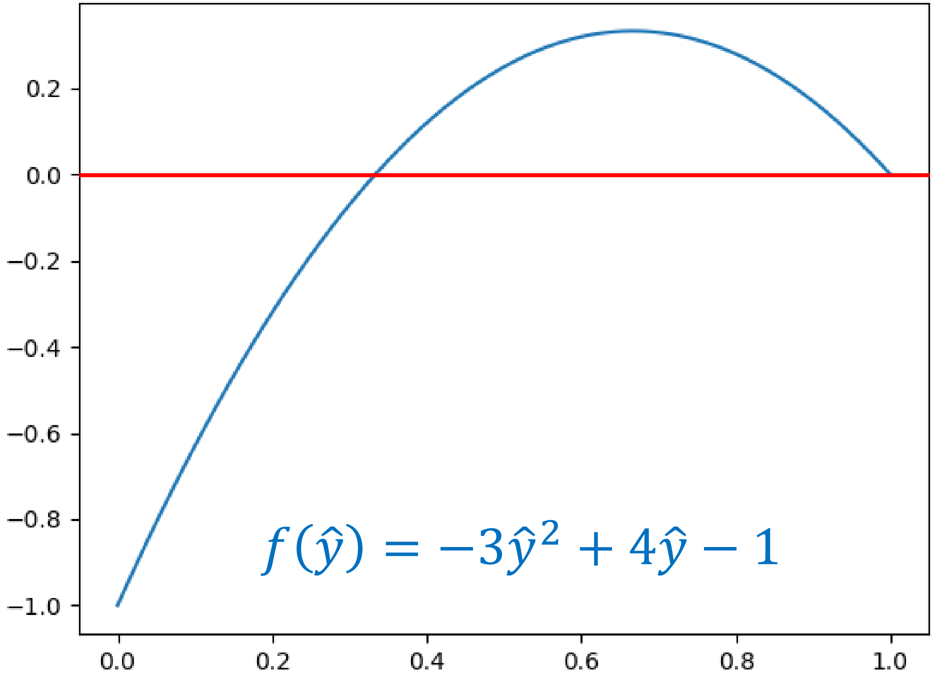
$y = 0$

$f(\hat{y}) = -3\hat{y}^2 + 2\hat{y}$

$y = 1$

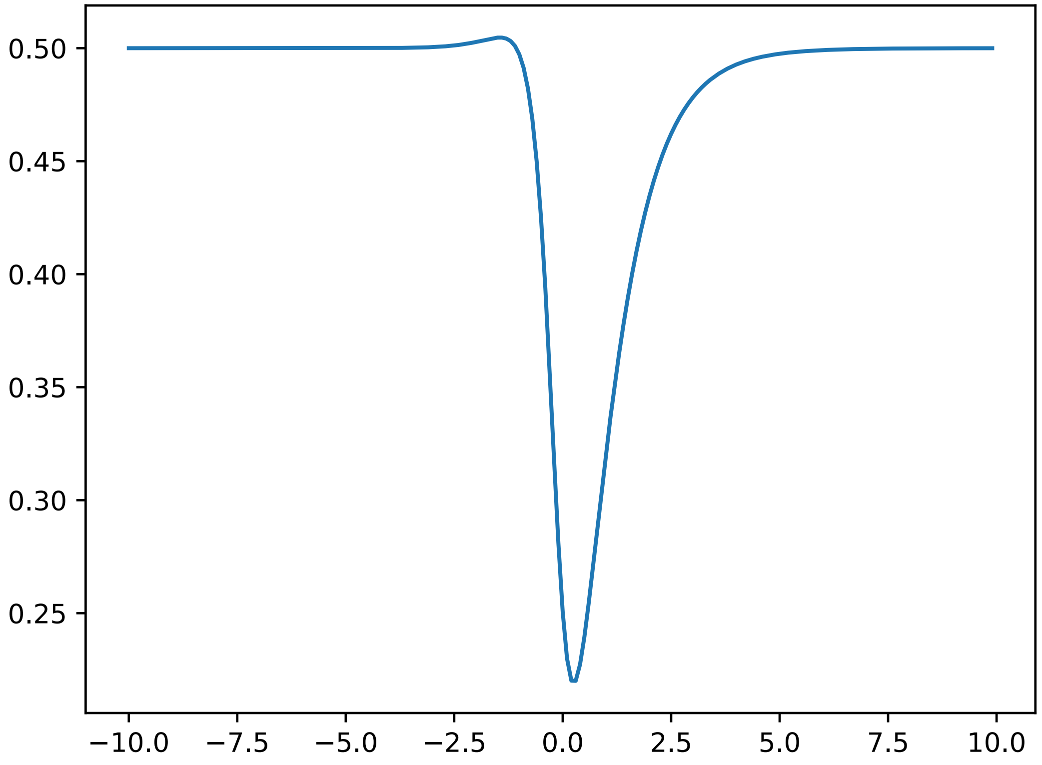
$f(\hat{y}) = -3\hat{y}^2 + 4\hat{y} - 1$



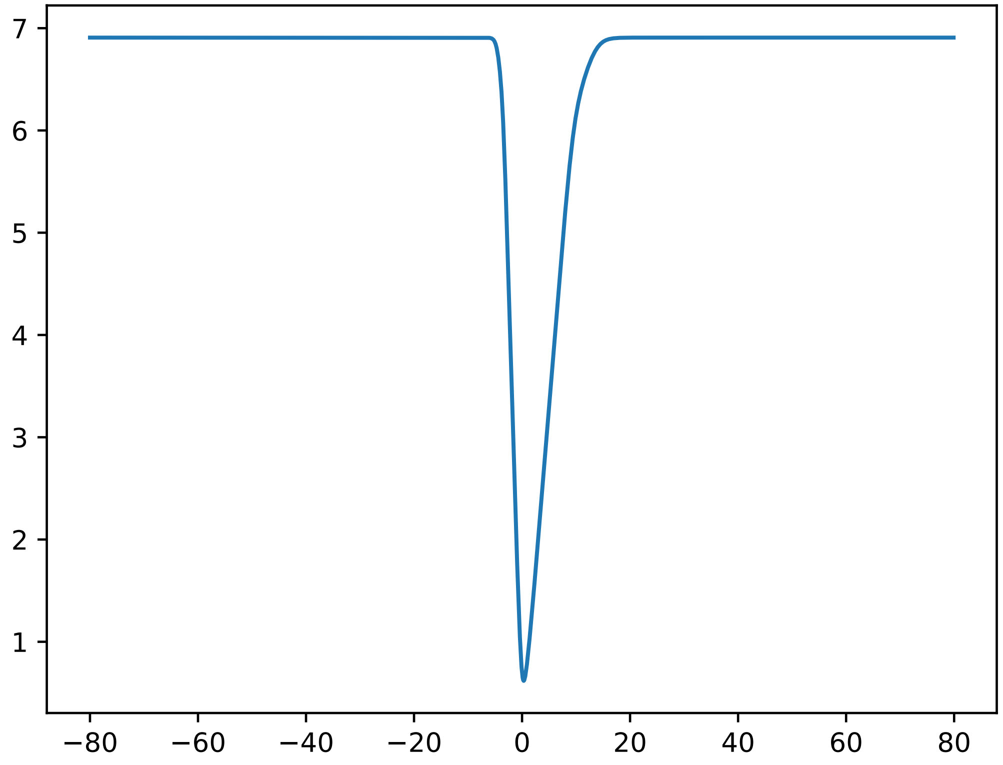



# MSE and BCE

## ❖ Visualization




Mean Squared Error

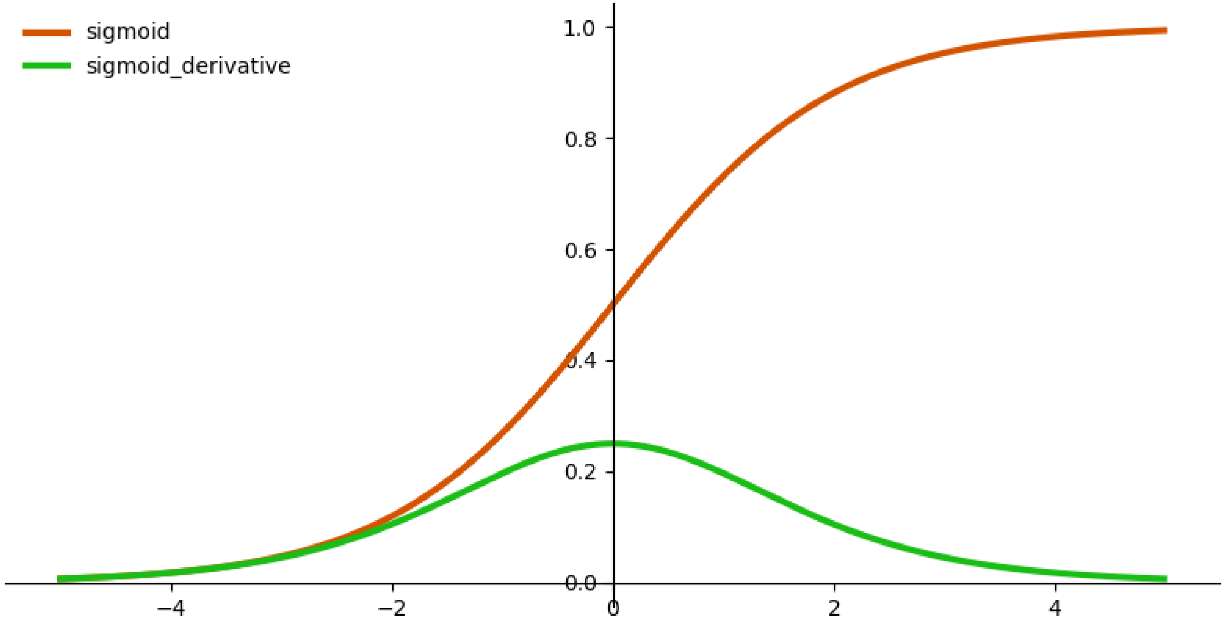


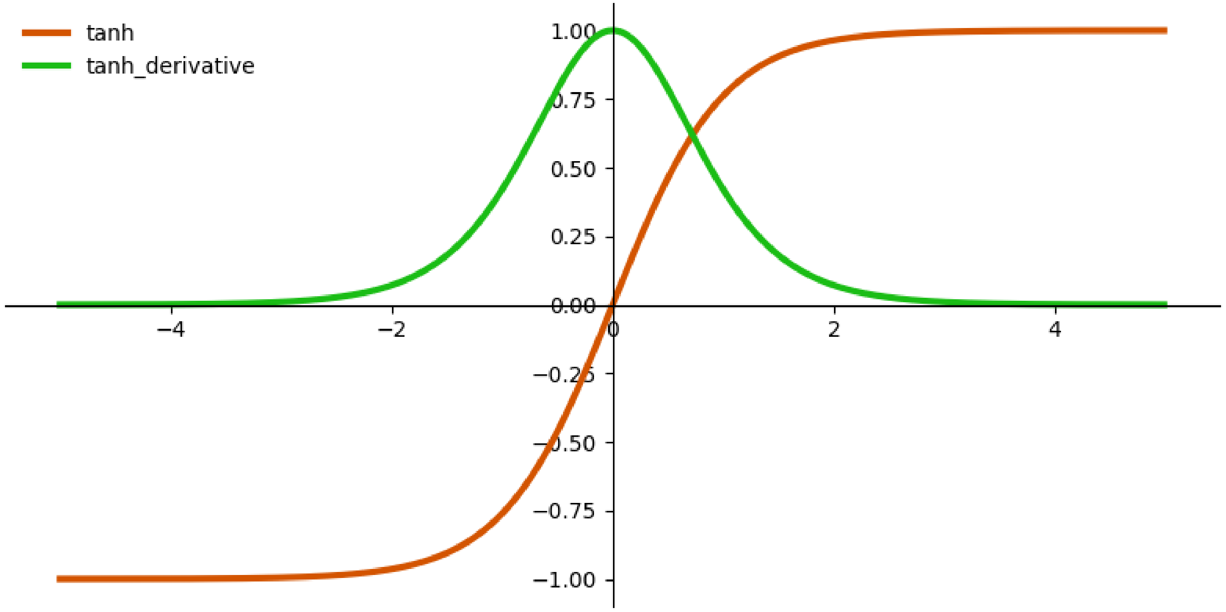

Binary Cross-Entropy

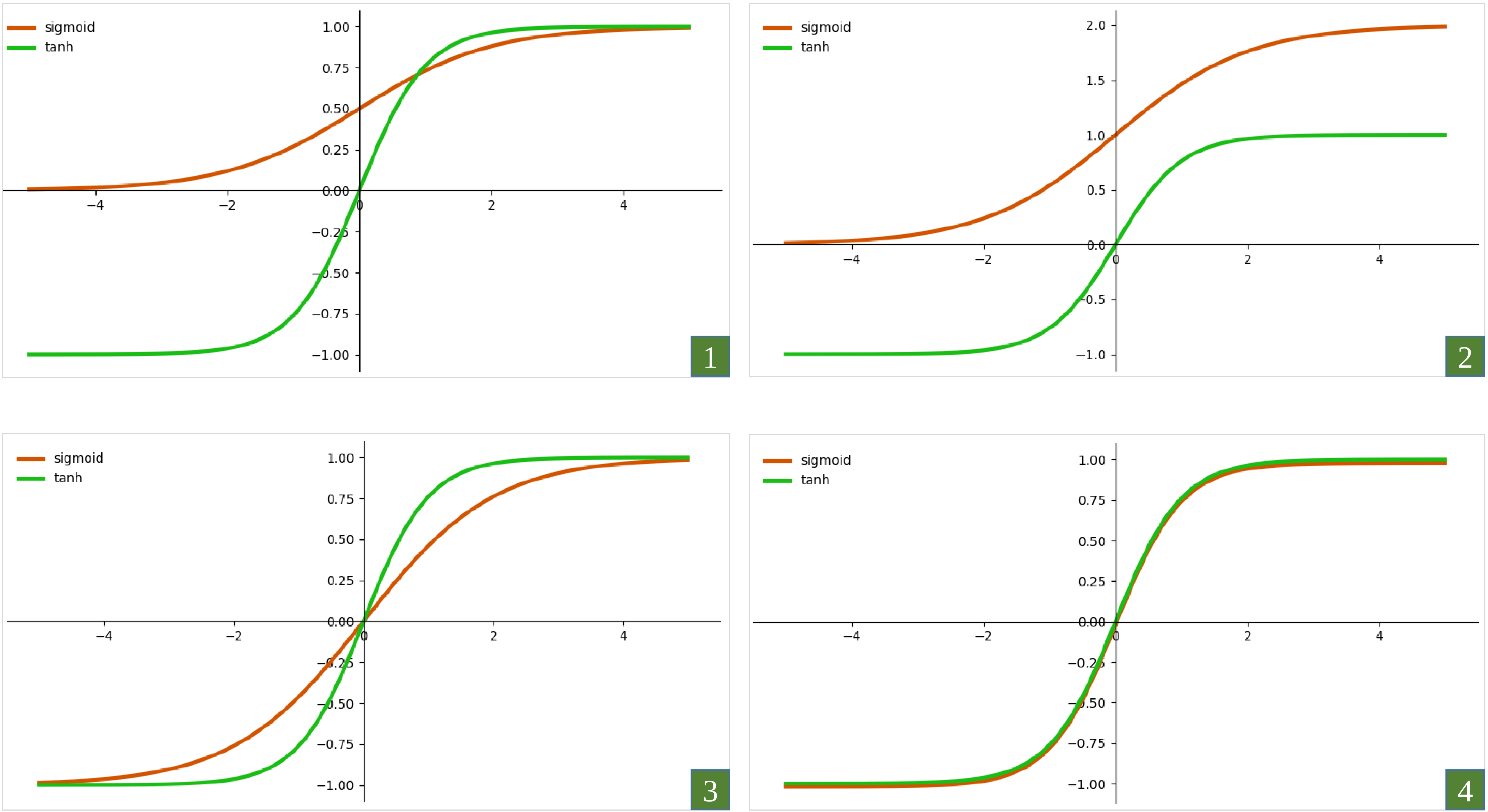
# Sigmoid and Tanh Functions

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



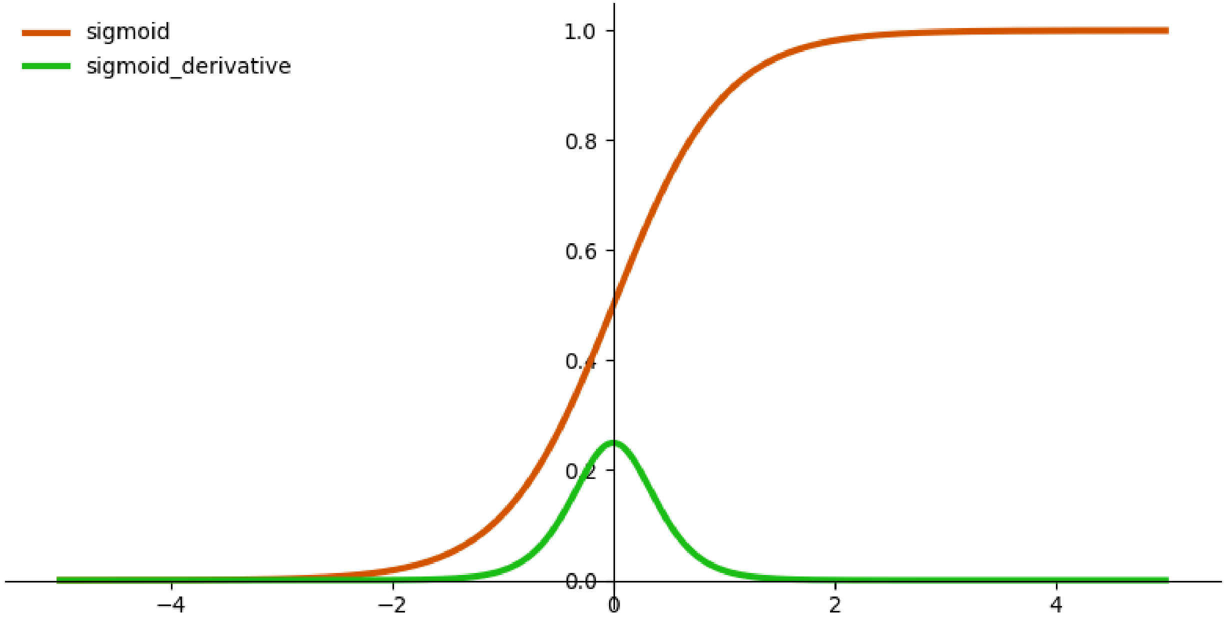

sigmoid
tanh
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−4
−2
0
2
4
1
sigmoid
tanh
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4
2
sigmoid
tanh
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−4
−2
0
2
4
3
sigmoid
tanh
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−4
−2
0
2
4
4

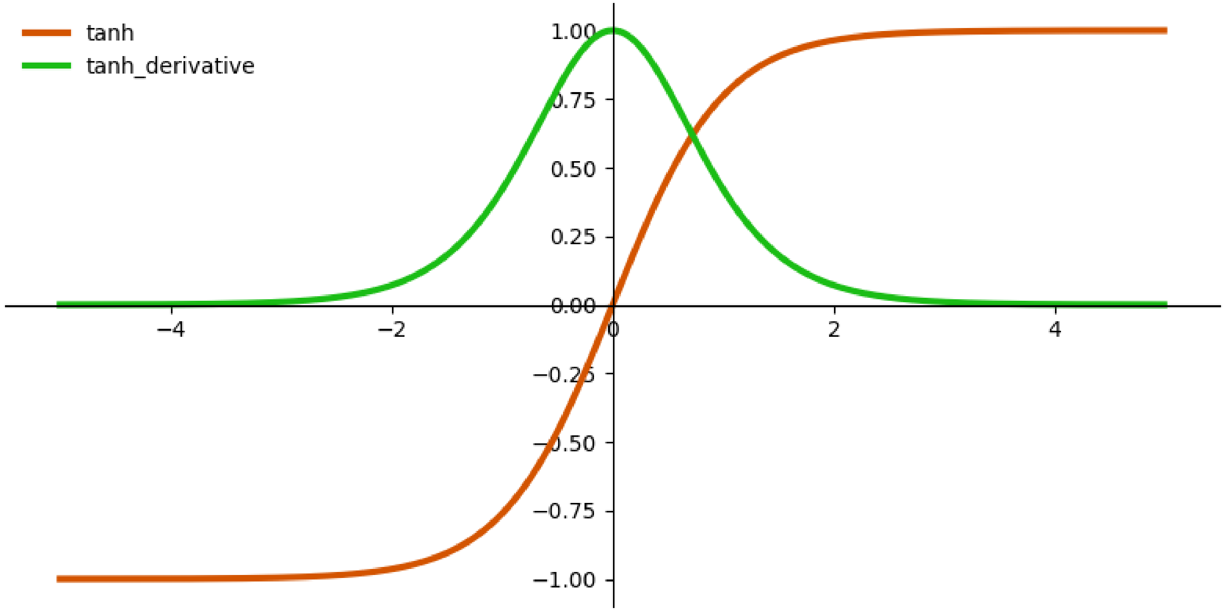
# Sigmoid and Tanh Functions

$$\mathrm{sigmoid}(2x) = \frac{1}{1+e^{-2x}}$$

$$\tanh(x) = 2 \times \frac{1}{1+e^{-2x}} - 1$$

$$\tanh(x) = 2 \times \mathrm{sigmoid}(2x) - 1$$



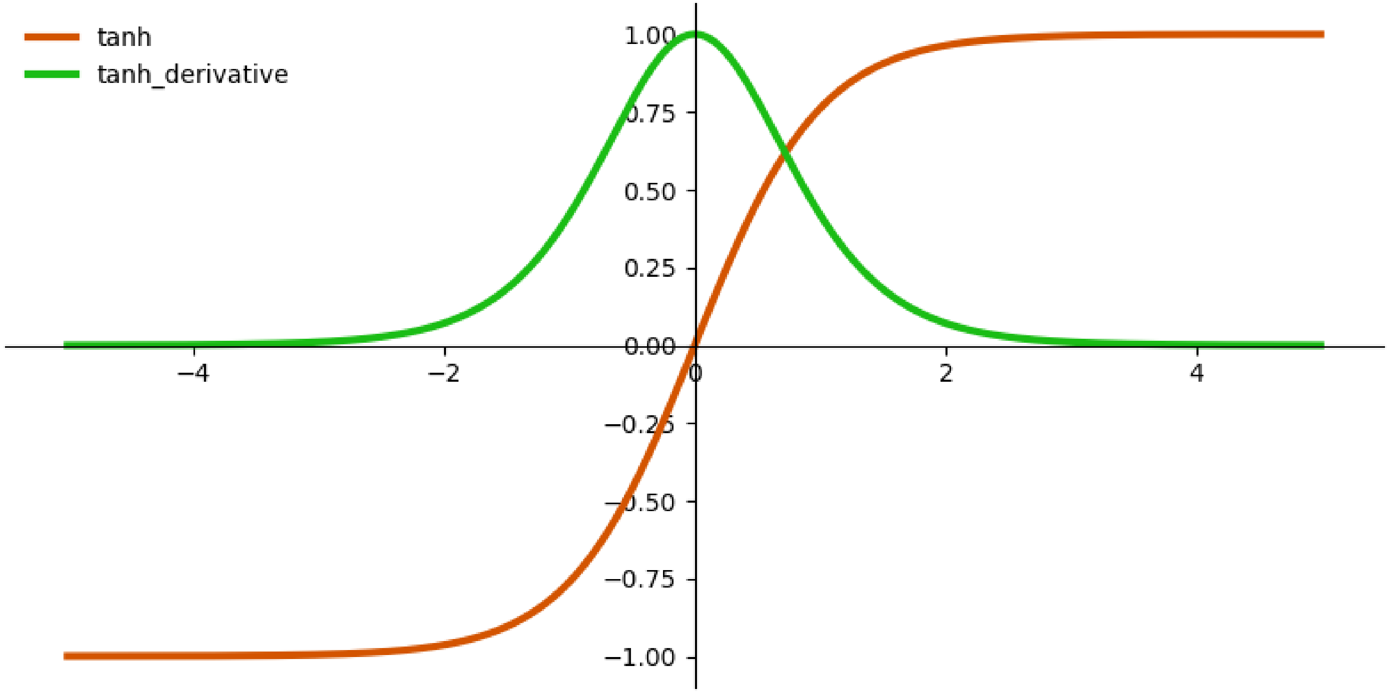
# Outline

- Vectorization
- Optimiztion for 1+ samples
- Logistic Regression – Mini-batch
- Logistic Regression – Batch
- BCE and MSE Loss Functions
- Sigmoid and Tanh Function (Optional)

# Tanh function

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

$$= 1 - \frac{2}{e^{2x} + 1}$$




$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$= -\frac{e^{-2x} - 1}{e^{-2x} + 1} = \frac{2}{e^{-2x} + 1} - 1$$

# Tanh function

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} = 1 - \frac{2}{e^{2x} + 1}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} = -\frac{e^{-2x} - 1}{e^{-2x} + 1} = \frac{2}{e^{-2x} + 1} - 1$$

$$tanh'(x) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)' = \frac{(e^x + e^{-x})(e^x + e^{-x}) - (e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2}$$

$$= \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2}$$

$$= 1 - \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)^2 = 1 - tanh^2(x)$$

AI VIETNAM
All-in-One Course

# Tanh function

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} = 1 - \frac{2}{e^{2x} + 1}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} = -\frac{e^{-2x} - 1}{e^{-2x} + 1} = \frac{2}{e^{-2x} + 1} - 1$$

$$tanh'(x) = \left(\frac{2}{e^{-2x} + 1} - 1\right)' = \frac{4e^{-2x}}{(e^{-2x} + 1)^2} = 4\left(\frac{e^{-2x} + 1 - 1}{(e^{-2x} + 1)^2}\right)$$

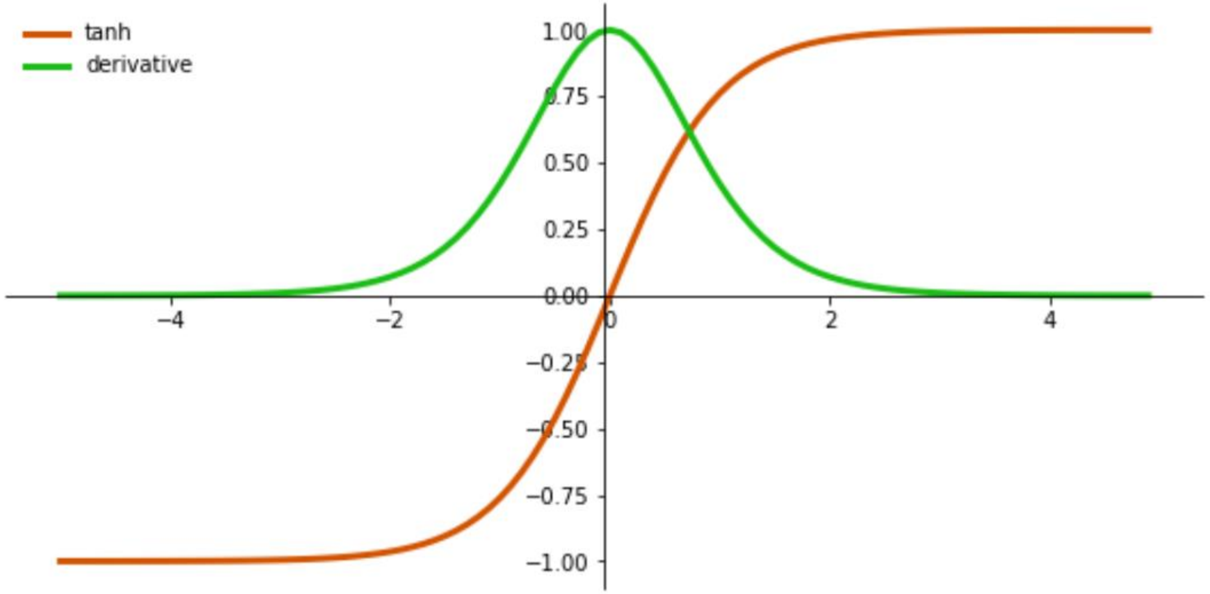
$$= 4\left(\frac{1}{e^{-2x} + 1} - \frac{1}{(e^{-2x} + 1)^2}\right) = -\left(\frac{4}{(e^{-2x} + 1)^2} - \frac{4}{e^{-2x} + 1}\right)$$

$$= -\left(\frac{4}{(e^{-2x} + 1)^2} - \frac{4}{e^{-2x} + 1} + 1 - 1\right) = 1 - \left(\frac{2}{e^{-2x} + 1} - 1\right)^2 = 1 - tanh^2(x)$$

# Logistic Regression
# Tanh

## ❖ Construct loss

$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1$$




**Model and Loss**

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

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

$$\hat{y}_s = \frac{\hat{y}+1}{2}$$

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

**Derivative**

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

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

$$\frac{\partial \hat{y}_s}{\partial \hat{y}} = \frac{1}{2} \qquad \frac{\partial \hat{y}}{\partial z} = 1 - \hat{y}^2 \qquad \frac{\partial z}{\partial \theta_i} = x_i$$

$$\frac{\partial L}{\partial \theta_i} = x_i \frac{(\hat{y}_s - y)(1-\hat{y}^2)}{2\hat{y}_s(1-\hat{y}_s)}$$

# Logistic Regression
# Tanh

## ❖ Construct loss

$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

**Model and Loss**

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

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

$$\hat{y}_s = \frac{\hat{y}+1}{2}$$

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

**Derivative**

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

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

$$\frac{\partial \hat{y}_s}{\partial \hat{y}} = \frac{1}{2} \qquad \frac{\partial \hat{y}}{\partial z} = 1 - \hat{y}^2 \qquad \frac{\partial z}{\partial \theta_i} = x_i$$

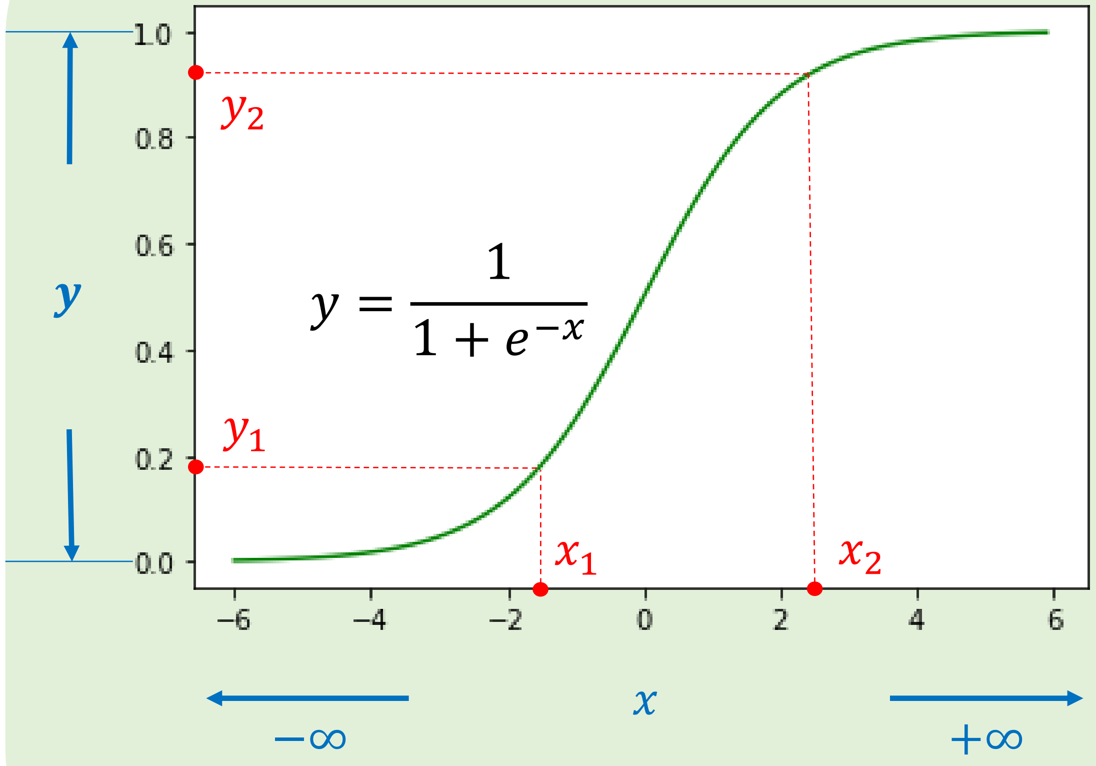
$$\frac{\partial L}{\partial \theta_i} = x_i\frac{(\hat{y}_s - y)(1-\hat{y}^2)}{2\hat{y}_s(1-\hat{y}_s)}$$

$$\frac{\partial L}{\partial \theta_i} = x_i\frac{(\frac{\hat{y}+1}{2} - y)(1-\hat{y}^2)}{2\frac{\hat{y}+1}{2}(1-\frac{\hat{y}+1}{2})}$$

$$\frac{\partial L}{\partial \theta_i} = x_i\frac{(\hat{y}+1-2y)(1-\hat{y}^2)}{(\hat{y}+1)(1-\hat{y})}$$

$$\boxed{\frac{\partial L}{\partial \theta_i} = x_i(\hat{y}+1-2y)}$$

# Summary




Sigmoid function

1) Pick all the samples from training data

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

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

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

3) Compute loss (binary cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

$\eta$ is learning rate