

# Session 7&8

## Logistic Regression

**Machine Learning | Zahra Amini**

Telegram: @zahraamini\_ai & Instagram:@zahraamini\_ai & LinkedIn: @zahraamini-ai

<https://zil.ink/zahraamini>

# Gradient Descent

## Stochastic Gradient Descent (SGD)

در این روش، گرادیان برای هر نمونه داده به صورت جداگانه محاسبه می‌شود

$$W_{\text{new}} = W_{\text{old}} - \alpha \frac{\delta L}{\delta W_{\text{old}}}$$

## Batch Gradient Descent

در این روش، گرادیان کل مجموعه داده برای هر گام محاسبه می‌شود

## Mini-batch Gradient Descent

این روش ترکیبی از دو روش قبلی است، به این صورت که گرادیان برای دسته‌های کوچکی از داده‌ها (مینی‌بچ‌ها) محاسبه می‌شود

Epoch  
یک دوره زمانی است که الگوریتم یادگیری ماشین یک بار کامل بر روی کل مجموعه داده‌ها آموخت می‌بیند

Batch  
یک بچ مجموعه‌ای از نمونه‌های داده است که در یک مرحله از الگوریتم گرادیان دیسنست استفاده می‌شود

# Stochastic Gradient Descent (SGD)

```
# Set number of epochs and constant learning rate
n_epochs = 50
alpha = 0.1

# Initialize parameters randomly
w = np.random.randn(2,1)

# Training loop
for epoch in range(n_epochs):
    for i in range(m): # m should be defined as the number of samples in the dataset
        random_index = np.random.randint(m) # Select a random index
        xi = X_b[random_index:random_index+1] # Extract features for selected sample
        yi = y[random_index:random_index+1] # Extract target for selected sample

        # Compute gradient of loss function
        gradients = 2 * xi.T.dot(xi.dot(w) - yi)

        # Update parameters
        w = w - alpha * gradients # Apply gradient descent step
```

# Batch Gradient Descent

```
n_epochs = 50
alpha = 0.1 # learning rate

w = np.random.randn(2,1) # random initialization

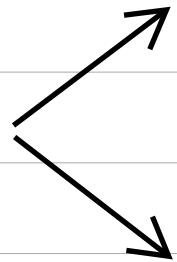
for epoch in range(n_epochs):
    gradients = 2/m * X_b.T.dot(X_b.dot(w) - y)
    w = w - alpha * gradients
```

# Mini-batch Gradient Descent

```
n_epochs = 50  
batch_size = 20 # size of mini-batch  
alpha = 0.1 # learning rate  
  
w = np.random.randn(2,1) # random initialization  
  
for epoch in range(n_epochs):  
    shuffled_indices = np.random.permutation(m)  
    X_b_shuffled = X_b[shuffled_indices]  
    y_shuffled = y[shuffled_indices]  
    for i in range(0, m, batch_size):  
        xi = X_b_shuffled[i:i+batch_size]  
        yi = y_shuffled[i:i+batch_size]  
        gradients = 2/batch_size * xi.T.dot(xi.dot(w) - yi)  
        w = w - alpha * gradients
```

Supervised  
learning

Regression



Classification → Email Spam

Classification: Breast Cancer Detection

✗ Malignant بُدخن

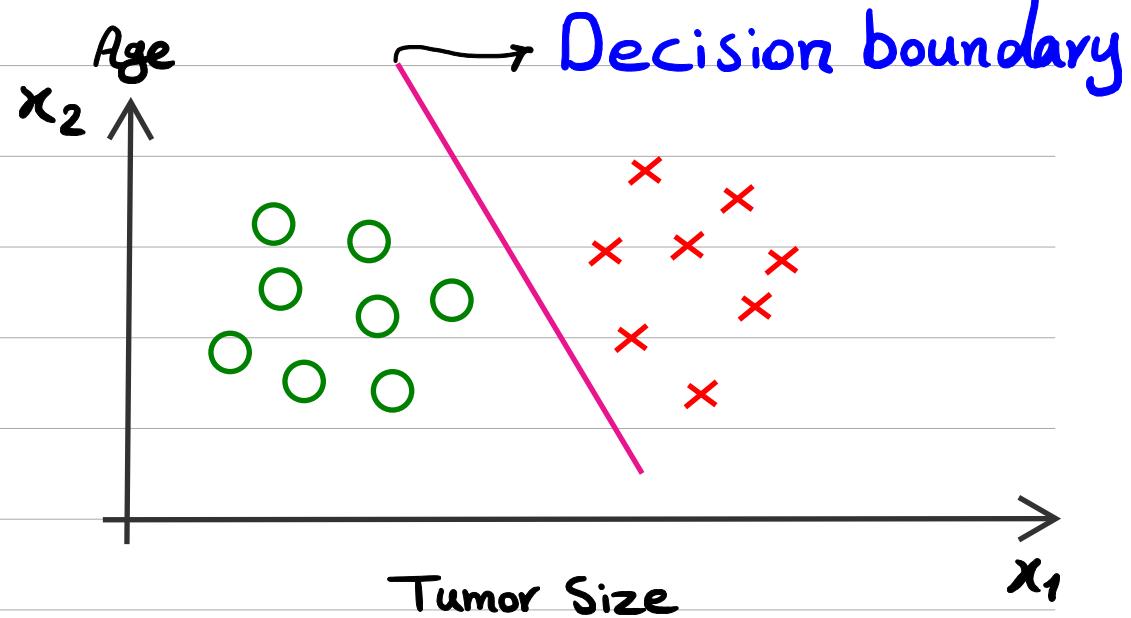
○ Benign خوش



Tumor Size

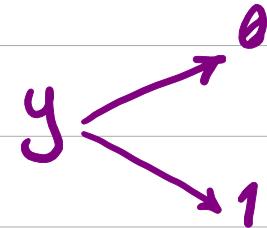
Transaction Fraudulent

Tumor Malignant / Cancer



# چیست؟ Regression و classification

در classification مسی داریم تعداد کی از خروجی ها را پیش بینی کنیم. به عبارتی اعداد محدود کی داریم.

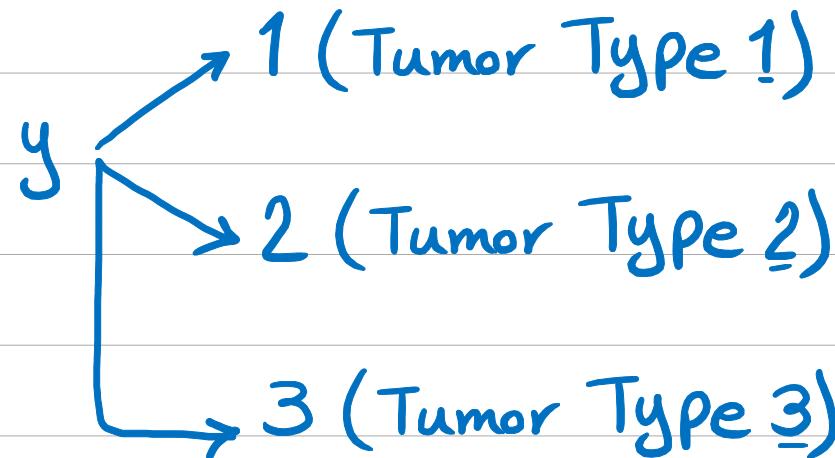


مثل اینجا که فقط دو خردمندی! (خوش خشم) و ۰ (بدخشم) را داریم.

اما در Regression مابینی داریم هر عددی را پیش بینی کنیم.

$$y \in [0, +\infty)$$

۱۰۱۲۳۴ توانیم بیش از دو عدد داشته باشیم مثلاً اما نمی‌توانیم classification \*



بلویم تمام اعداد بین ۰ تا ۱ (۰, ۰.۰۰۰۱, ۰.۲, ... ۱).

## Classification :

Question	Answer	
Is this Email Spam?	No	Yes
IS the transaction fraudulent?	No	Yes
IS the tumor malignant?	No	Yes

two classes → Binary Classification

category

False      True

$\emptyset$   
∅

1

Negative class

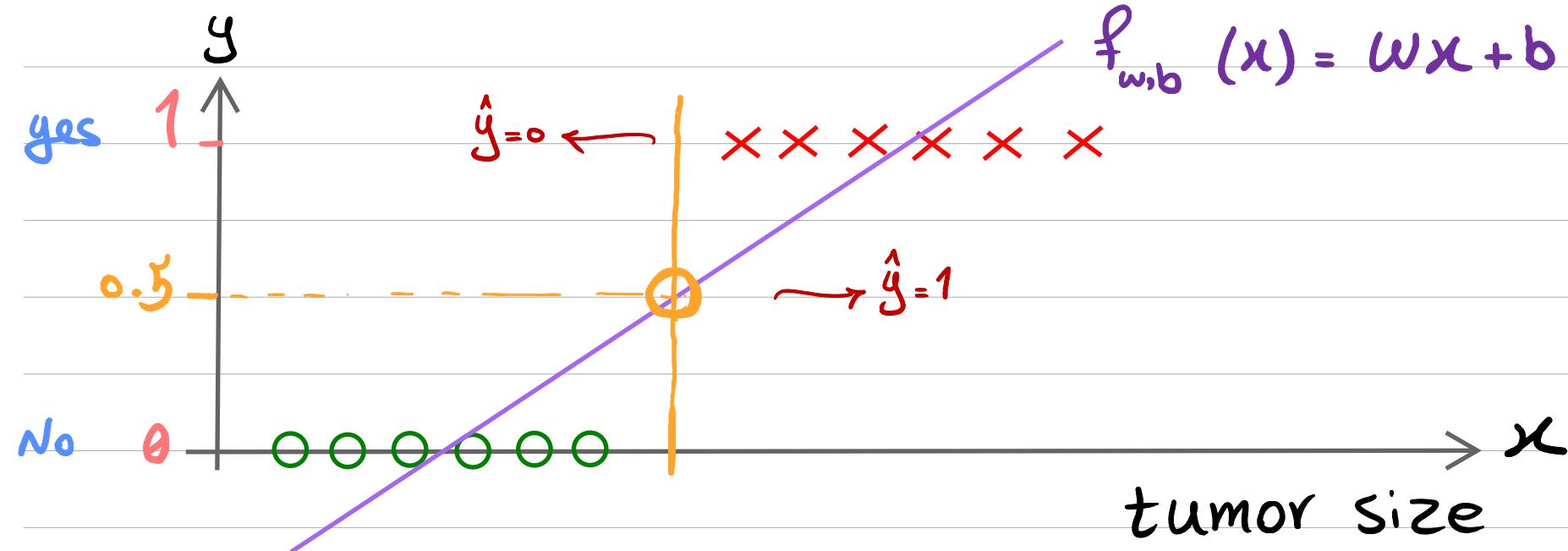
Positive class

اگر بخواهیم بین سؤال پاسخ دهیم که "آیا تصور سرطانی مبتداست یا" چه باید کرد؟

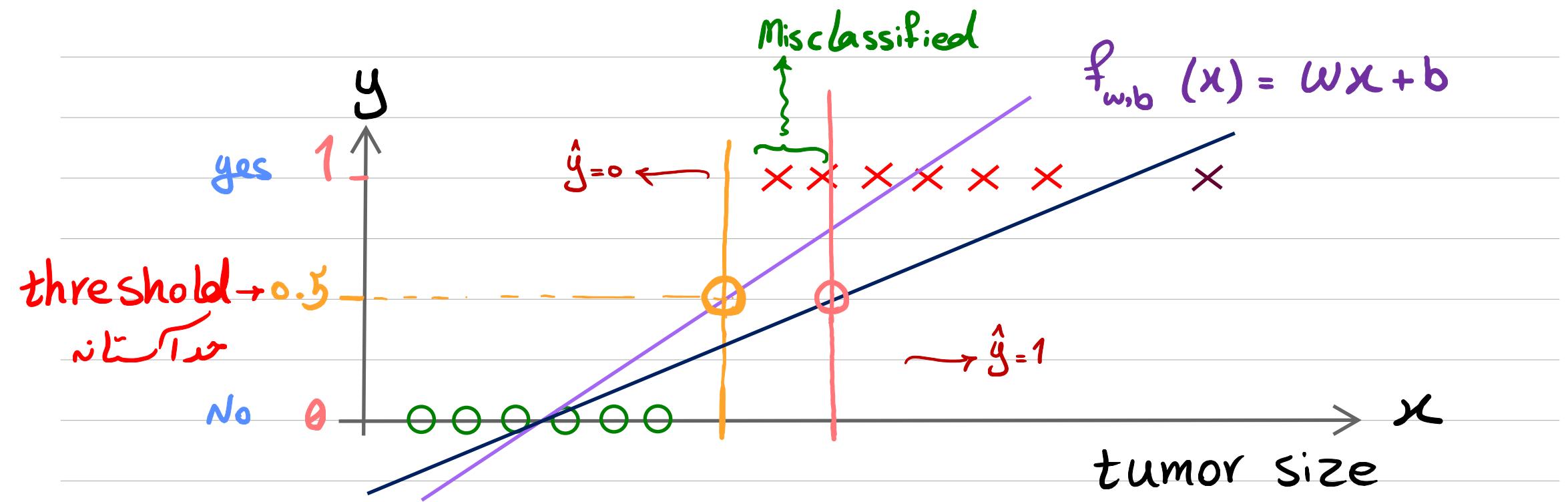
Malignant بدشیم

Being خوششیم

ج: چه طور داده هایمان را کلاس بندی کنیم؟ classification



$$\text{if } \begin{cases} f_{w,b}(x) < 0.5 & \hat{y} = 0 \\ f_{w,b}(x) \geq 0.5 & \hat{y} = 1 \end{cases}$$



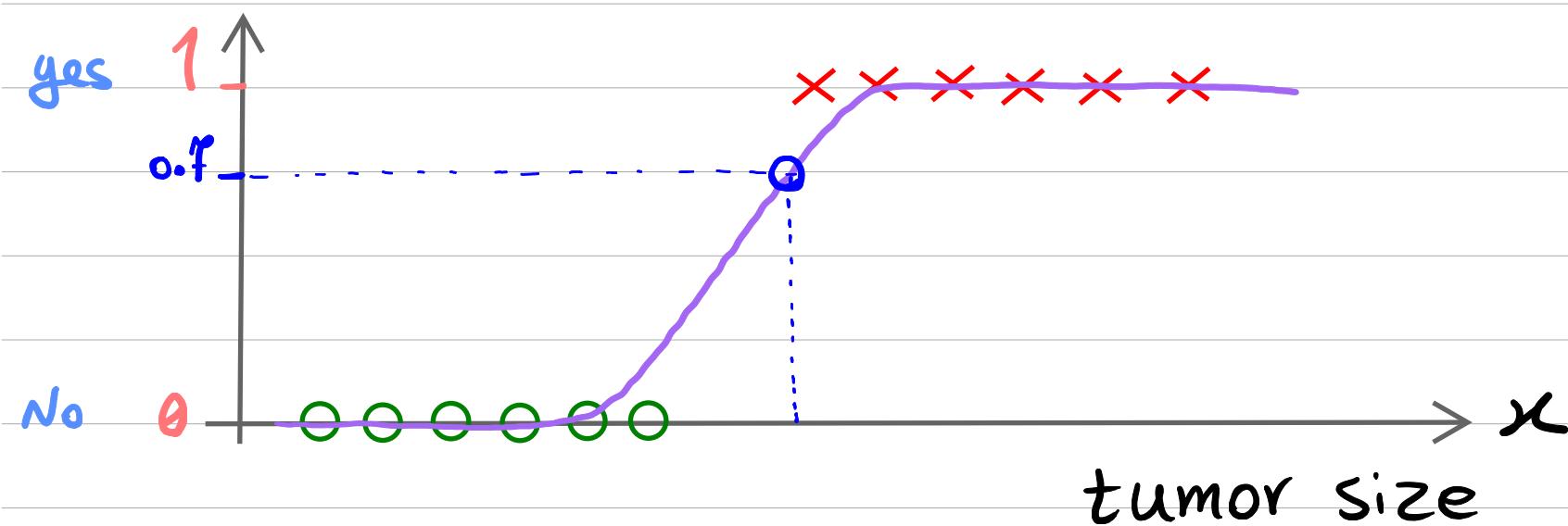
1. Misclassified

: Regression از استفاده کنیم

2.  $F_{w,b}(x) \in (-\infty, +\infty)$

؟ آیا می توان از Classification برای Regression از استفاده نمود ؟

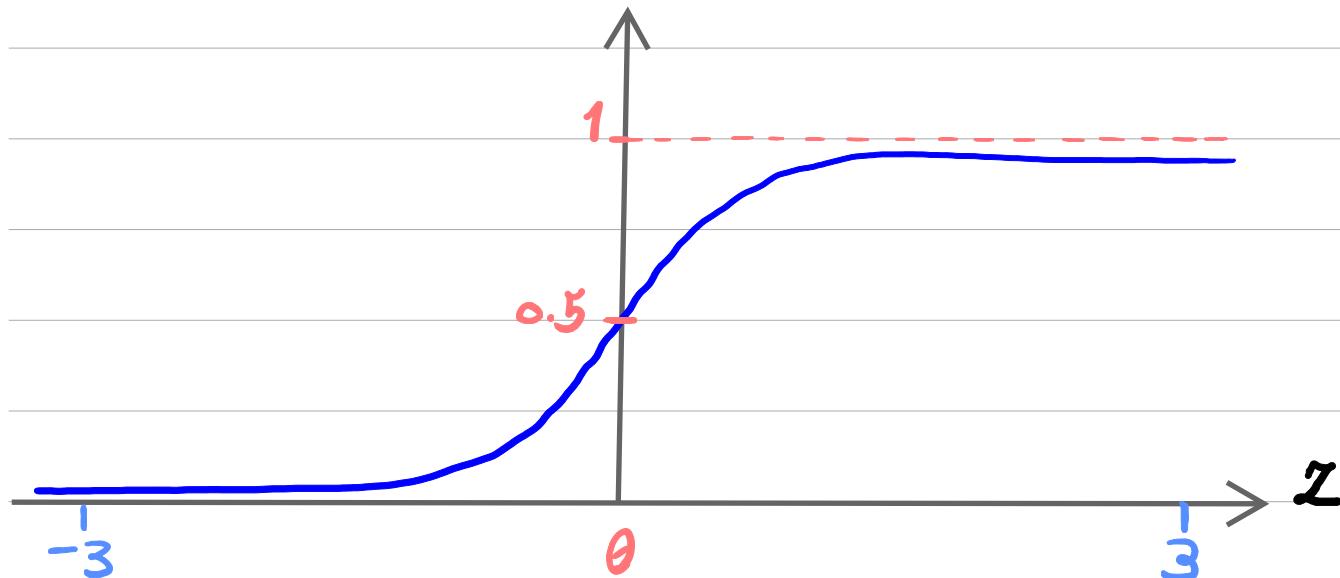
## ۲. برای رفع مشکلات چه باید نمود؟ Regression



از تابع Sigmoid استفاده می‌کنیم. Sigmoid تابعی است که هر عددی را به محدود در در خروجی به مابا احتمالش (یعنی عددی بین ۰ و ۱) راهی دهد.

## Sigmoid Function:

0 < outputs < 1



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

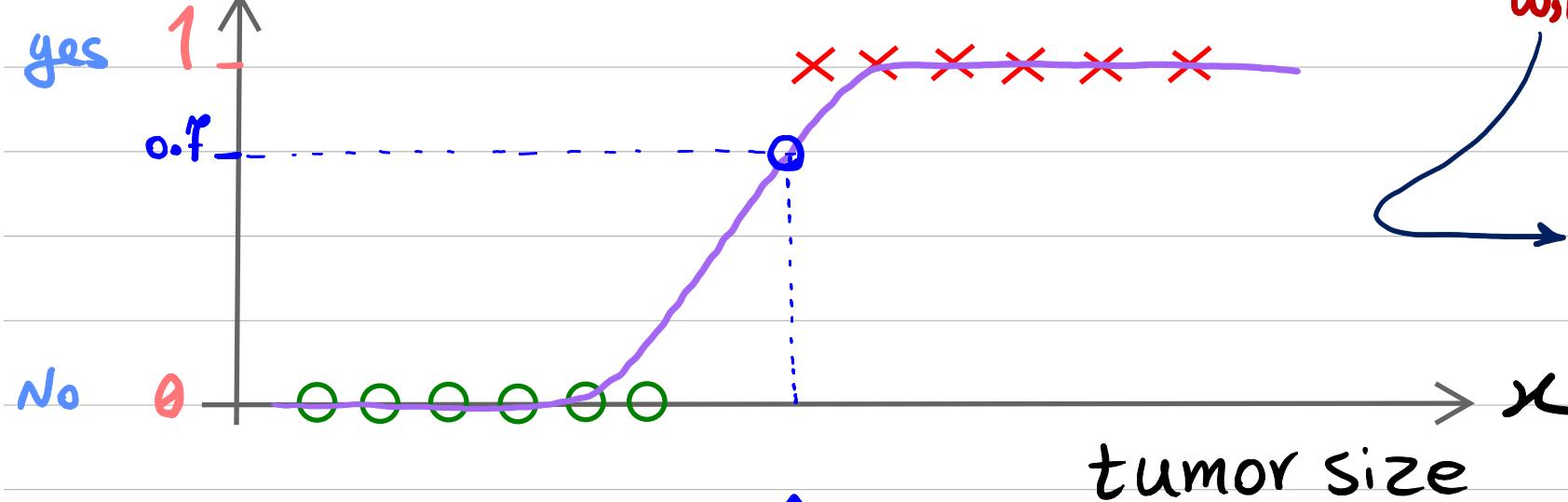
0 < g(z) < 1

$$f_{w,b}(x) = wx + b \rightsquigarrow Z = wx + b \xrightarrow{\text{Sigmoid}} g(z)$$

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

Logistic Regression

$$f_{w,b}(x) = g(wx + b) = \frac{1}{1 + e^{-(wx+b)}}$$



$$F_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}}$$

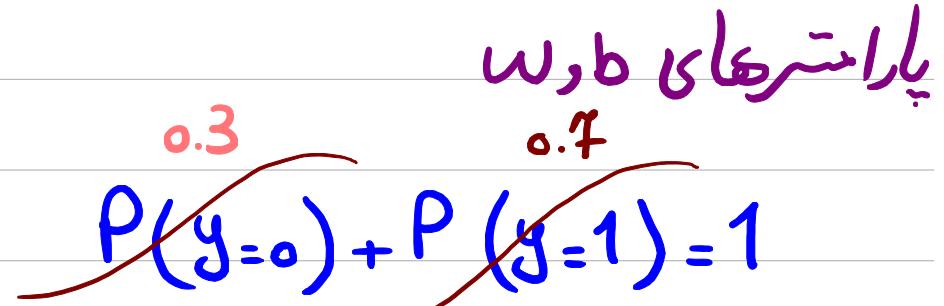
Probability that  
class is 1

$$F_{w,b}(x) = 0.7 \rightsquigarrow \hat{y}=?$$

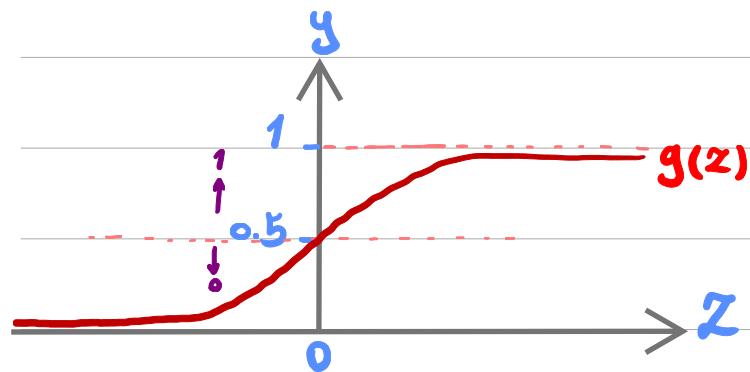
$$f_{w,b}(x) = P(y=1 | x; w, b) \rightarrow \text{احتمال اینکه } \hat{y}=1 \text{ بشود بازای } x \text{ داده شود}$$

$$P(y=0) + P(y=1) = 1$$

$$\begin{cases} F_{w,b}(x) \geq 0.5 & \hat{y}=1 \\ F_{w,b}(x) < 0.5 & \hat{y}=0 \end{cases}$$



پارامترهای  $w, b$



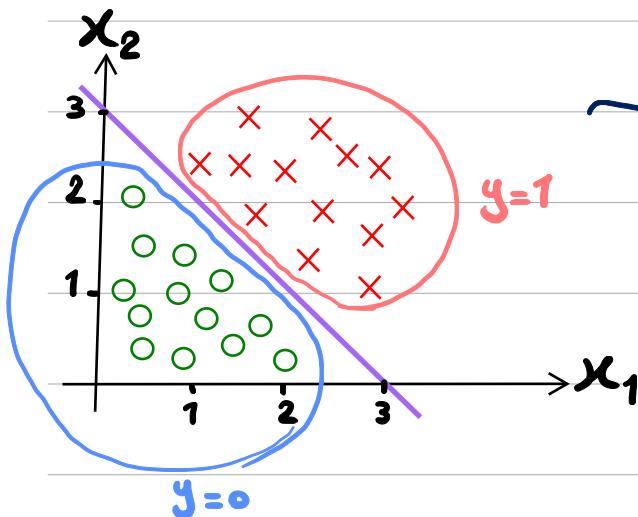
$$z = w \cdot x + b$$

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

$$f_{\vec{w}, b}(\vec{x}) = g(\vec{w} \cdot \vec{x} + b) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}} = P(y=1 | x; \vec{w}, b)$$

## Decision boundary

حالات خواهیم بود که سایر حالتی را  $(x_1, x_2)$  دارد.

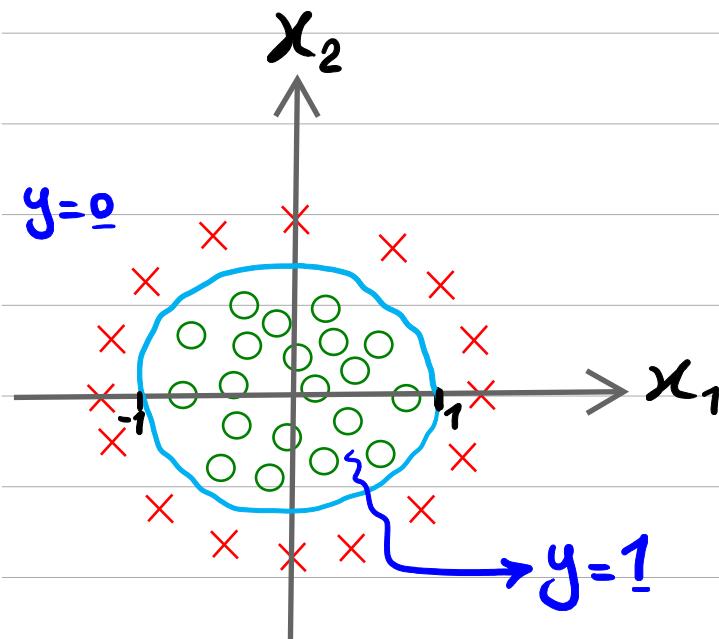


$$F_{\vec{w}, b}(\vec{x}) = g(z) = g(w_1 x_1 + w_2 x_2 + b)$$

$$z = \vec{w} \cdot \vec{x} + b = 0$$

if  $w_1=1, w_2=1, b=-3$

$$w_1 x_1 + w_2 x_2 + b = 0 \rightsquigarrow x_1 + x_2 - 3 = 0 \rightsquigarrow$$



$$z = x_1^2 + x_2^2 - 1 = 0$$

$$x_1^2 + x_2^2 = 1$$

$$\begin{cases} x_1=0, x_2=3 \\ x_1=3, x_2=0 \end{cases}$$

if  $x_1=0$   
 ~~$w_1 x_1 + w_2 x_2 + b = 0$~~   
 $\Rightarrow w_2 x_2 = -b \Rightarrow$   
 $x_2 = \frac{-b}{w_2} = \frac{-(-3)}{1} = 3$

tumor size	...	Patient's age	malignant?
$x_1$		$x_n$	y
10		52	1
2		73	0
5		55	0
12		49	1
:		:	:

$i=1, \dots, m$

$j=1, \dots, n$

target  $y$  is 0 or 1

$$F_{w,b}(x) = \frac{1}{1+e^{-(wx+b)}}$$

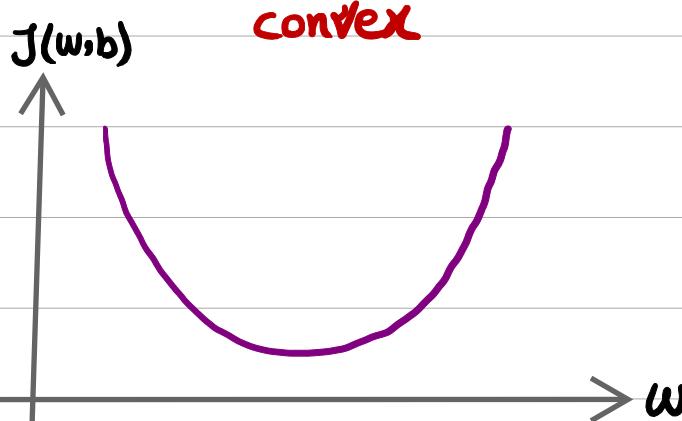
$$\vec{W} = [w_1 \ w_2 \ \dots \ w_n], \ b$$

Linear Regression

Cost Function

چن طور پر کتنے  $w, b$  را اتنا بکشم؟

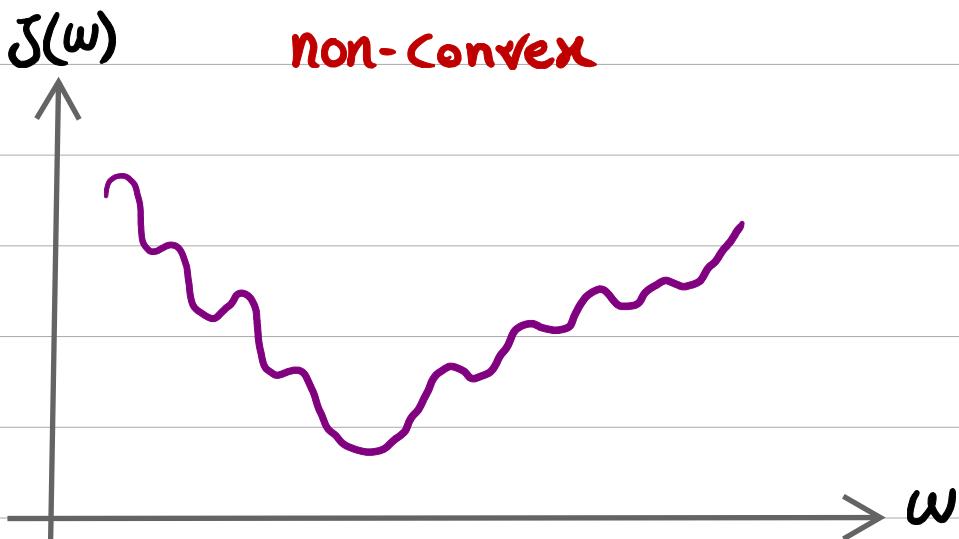
$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m \underbrace{(F_{w,b}(x^{(i)}) - y^{(i)})^2}_{\hat{y}^{(i)}}$$



$$F_{w,b}(x) = w \cdot x + b$$

## Logistic Regression

$$f_{w,b}(x) = \frac{1}{1+e^{-(wx+b)}}$$



## Logistic Loss Function

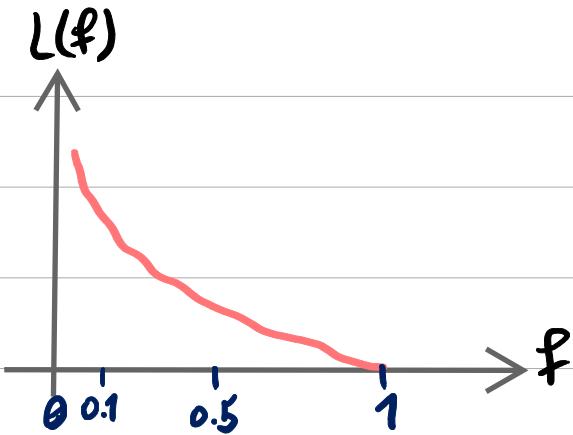
$$L(f_{w,b}(x^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{w,b}(x^{(i)})) & \text{if } y^{(i)}=1 \\ -\log(1-f_{w,b}(x^{(i)})) & \text{if } y^{(i)}=0 \end{cases}$$



؟ آیا استفاده از این تابع برای Logistic درست است؟

$$L(f_{w,b}(x^{(i)}), y^{(i)})$$

if  $y=1$



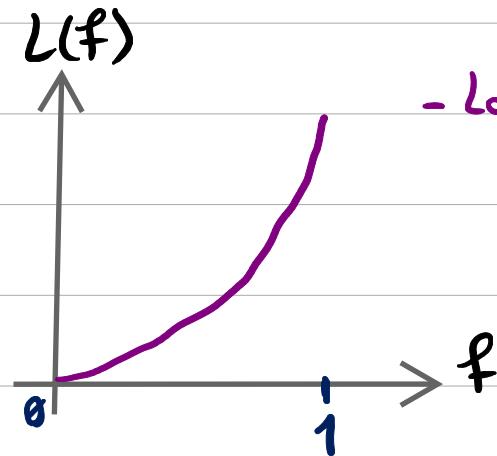
if  $f_{w,b}(x^{(i)}) \rightarrow 1$  then Loss  $\rightarrow 0$

but if  $f_{w,b}(x^{(i)}) \rightarrow 0$  then Loss  $\rightarrow \infty$

$$-\log(1-f_{w,b}(x^{(i)}))$$

if  $y^{(i)}=0$

$$-\log(1-f) \rightarrow [0, 1]$$



if  $f_{w,b}(x^{(i)}) \rightarrow 0$  then Loss  $\rightarrow \infty$

$f_{w,b}(x^{(i)}) \rightarrow 1$  then Loss  $\rightarrow 0$

## Logistic Regression Cost function:

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m \underbrace{L(f_{w,b}(x^{(i)}), y^{(i)})}_{\downarrow}$$

$$\begin{cases} -\log(f_{w,b}(x^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1-f_{w,b}(x^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

$$L(f_{w,b}(x^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{w,b}(x^{(i)})) & \text{if } y^{(i)}=1 \\ -\log(1-f_{w,b}(x^{(i)})) & \text{if } y^{(i)}=0 \end{cases}$$

ج: حالاً الريـك تابع يـكـيـارـجـه بـخـواـصـه بـاـيدـجـه كـنـسـمـ؟

$$L(f_{w,b}(x^{(i)}), y^{(i)}) = -y^{(i)} \log(f_{w,b}(x^{(i)})) - (1-y^{(i)}) \log(1-f_{w,b}(x^{(i)}))$$

**Convex**

if  $y^{(i)} = 1$ :

$$-\underbrace{(1) \log(f_{w,b}(x^{(i)}))}_{\theta} - \cancel{(1-(1))} \cancel{\log(1-f_{w,b}(x^{(i)}))}$$

if  $y^{(i)} = 0$ :

$$\cancel{-(0) \log(f_{w,b}(x^{(i)}))} - \cancel{(1-(0))} \cancel{\log(1-f_{w,b}(x^{(i)}))}$$

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m [L(f_{w,b}(x^{(i)}), y^{(i)})]$$

$$= -\frac{1}{m} \sum_{i=1}^m \underbrace{\left[ y^{(i)} \log(f(x^{(i)})) + (1-y^{(i)}) \log(1-f_{w,b}(x^{(i)})) \right]}_{\text{Log Likelihood}}$$

حالا ما باید به ذیل  $w, b$  باشیم که  $J$  را minimum کند.

چه طور مترادar  $w, b$  را پیدا کنیم؟

## 1. Newton's Method

دروش رایج داریم:

## 2. Gradient Descent

در مدل‌های احتمالی، هدف این است که پارامترهای مدل را طوری تنظیم کنیم که احتمال تولید داده‌های واقعی (که در مجموعه داده داریم) توسط مدل به حداقل برسد. برای این کار، از لایکلیهود استفاده می‌شود. لایکلیهود یک معیار است که بیان می‌کند با توجه به پارامترهای فعلی مدل، چقدر احتمال دارد که داده‌های مشاهده شده رخ دهند

## Gradient Descent:

$$J(w, b) = -\frac{1}{m} \sum_{i=1}^m \left[ y^{(i)} \log(f_{w,b}(x^{(i)})) + (1-y^{(i)}) \log(1-f_{w,b}(x^{(i)})) \right]$$

repeat

{

$$w_j = w_j - \alpha \boxed{\frac{\partial}{\partial w_j} J(w, b)}$$

$$\frac{\partial}{\partial w} J(w, b) =$$

$$\frac{1}{m} \sum_{i=1}^m (f_{w,b}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$b = b - \alpha \boxed{\frac{\partial}{\partial b} J(w, b)}$$

$$\frac{\partial}{\partial b} J(w, b) =$$

$$\frac{1}{m} \sum_{i=1}^m (f_{w,b}(x^{(i)}) - y^{(i)})$$

$$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}}$$

# Step-by-Step Logistic Regression

X1	X2	y(Label)
1	2	1 (Positive)
2	1	0 (Negative)
3	3	1 (Positive)

## 1. Logistic Regression Formula

$$h(x) = \frac{1}{1 + e^{-(w_1x_1 + w_2x_2 + b)}}$$

Here:

w<sub>1</sub>, w<sub>2</sub> Weights for X<sub>1</sub> and X<sub>2</sub>

b: Bias(offset)

z = w<sub>1</sub> X<sub>1</sub> + w<sub>2</sub> X<sub>2</sub> + b is the linear combination of features

## 2. Initialize Weights and Bias

$$w_1 = 0.5, w_2 = 0.5, b = 0$$

### 3. Compute Predictions ( $h(x)$ )

For each sample, calculate  $z$  and apply the sigmoid function to get the probability ( $h(x)$ )

For ( $X_1=1, X_2=2$ ):

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + b = 0.5 \cdot 1 + 0.5 \cdot 2 + 0 = 1.5$$

$$h(x) = \frac{1}{1 + e^{-1.5}} \approx 0.817$$

For ( $X_1=2, X_2=1$ ):

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + b = 0.5 \cdot 2 + 0.5 \cdot 1 + 0 = 1.5$$

$$h(x) = \frac{1}{1 + e^{-1.5}} \approx 0.817$$

For ( $X_1=3, X_2=3$ ):

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + b = 0.5 \cdot 3 + 0.5 \cdot 3 + 0 = 3$$

$$h(x) = \frac{1}{1 + e^{-3}} \approx 0.953$$

#### 4. Compute the Cost Function (Log Loss)

$$J(w, b) = -\frac{1}{m} \sum_{i=1}^m \left[ y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \right]$$

$$J(w, b) = -\frac{1}{3} \left[ 1 \cdot \log(0.817) + (1 - 1) \cdot \log(1 - 0.817) + 0 \cdot \log(0.817) + (1 - 0) \cdot \log(1 - 0.817) + 1 \cdot \log(0.953) + (1 - 1) \cdot \log(1 - 0.953) \right]$$

Step-by-step:

- For Sample 1:  $1 \cdot \log(0.817) \approx -0.202$
- For Sample 2:  $(1 - 0) \cdot \log(1 - 0.817) \approx -1.713$
- For Sample 3:  $1 \cdot \log(0.953) \approx -0.048$

$$J(w, b) \approx -\frac{1}{3} \cdot [ -0.202 - 1.713 - 0.048 ] = \frac{1}{3} \cdot 1.963 \approx 0.654$$

## 5. Compute Gradients

To update the weights and bias, calculate the gradients. The gradient for each parameter is:

- For weight  $w_j$ :

$$\frac{\partial J}{\partial w_j} = \frac{1}{m} \sum_{i=1}^m [h(x^{(i)}) - y^{(i)}] x_j^{(i)}$$

- For bias  $b$ :

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^m [h(x^{(i)}) - y^{(i)}]$$

**Compute Gradient for  $w_1$ :**

$$\frac{\partial J}{\partial w_1} = \frac{1}{3} [(0.817 - 1) \cdot 1 + (0.817 - 0) \cdot 2 + (0.953 - 1) \cdot 3]$$

$$\frac{\partial J}{\partial w_1} = \frac{1}{3} [-0.183 + 1.634 - 0.141] = \frac{1}{3} \cdot 1.31 \approx 0.436$$

**Compute Gradient for  $w_2$ :**

$$\frac{\partial J}{\partial w_2} = \frac{1}{3} [(0.817 - 1) \cdot 2 + (0.817 - 0) \cdot 1 + (0.953 - 1) \cdot 3]$$

$$\frac{\partial J}{\partial w_2} = \frac{1}{3} [-0.366 + 0.817 - 0.141] = \frac{1}{3} \cdot 0.31 \approx 0.103$$

**Compute Gradient for  $b$ :**

$$\frac{\partial J}{\partial b} = \frac{1}{3} [(0.817 - 1) + (0.817 - 0) + (0.953 - 1)]$$

$$\frac{\partial J}{\partial b} = \frac{1}{3} [-0.183 + 0.817 - 0.047] = \frac{1}{3} \cdot 0.587 \approx 0.196$$

## 6. Update Weights and Bias

Using gradient descent, update weights and bias:

$$w_j = w_j - \alpha \cdot \frac{\partial J}{\partial w_j}, \quad b = b - \alpha \cdot \frac{\partial J}{\partial b}$$

Assume learning rate  $\alpha=0.1$

پس از تکرار این مراحل به دفعات، مدل مقادیر بهینه‌ای برای وزن‌ها و بایاس یاد می‌گیرد. سپس،  
مدل رگرسیون منطقی می‌تواند با دقت بیشتری برای بررسی داده‌های جدید پیش‌بینی کند که  
آیا نظر مثبت است یا منفی، بر اساس ویژگی‌های آن

Update  $w_1$ :

$$w_1 = 0.5 - 0.1 \cdot 0.436 = 0.456$$

Update  $w_2$ :

$$w_2 = 0.5 - 0.1 \cdot 0.103 = 0.4897$$

Update  $b$ :

$$b = 0 - 0.1 \cdot 0.196 = -0.0196$$

## 7. Make New Predictions

With updated parameters ( $w_1=0.456$ ,  $w_2=0.4897$ ,  $b=-0.0196$ ):

For Sample 1:

$$z = 0.456 \cdot 1 + 0.4897 \cdot 2 - 0.0196 = 1.4158$$

$$h(x) = \frac{1}{1 + e^{-1.4158}} \approx 0.804 \rightarrow \text{Class 1}$$

For Sample 2:

$$z = 0.456 \cdot 2 + 0.4897 \cdot 1 - 0.0196 = 1.3818$$

$$h(x) = \frac{1}{1 + e^{-1.3818}} \approx 0.799 \rightarrow \text{Class 1}$$

For Sample 3:

$$z = 0.456 \cdot 3 + 0.4897 \cdot 3 - 0.0196 = 2.817$$

$$h(x) = \frac{1}{1 + e^{-2.817}} \approx 0.943 \rightarrow \text{Class 1}$$

مدل همچنان تمام نمونه‌ها را به عنوان مثبت پیش‌بینی می‌کند، به این معنی که برای کاهش خطا به بهینه‌سازی بیشتری نیاز است