

# ENGR 3321: Introduction to Deep Learning for Robotics

Neural Network NO1:  
Multi-Input One-Output Model

10/16/2023



# Outline

- Binary Classification
- Multi-Feature Input
- Binary Cross Entropy Loss
- Sigmoid Activation
- Gradient Descent

# Multi-Input One-Output Model

Assume a dataset contains  $M$  objects

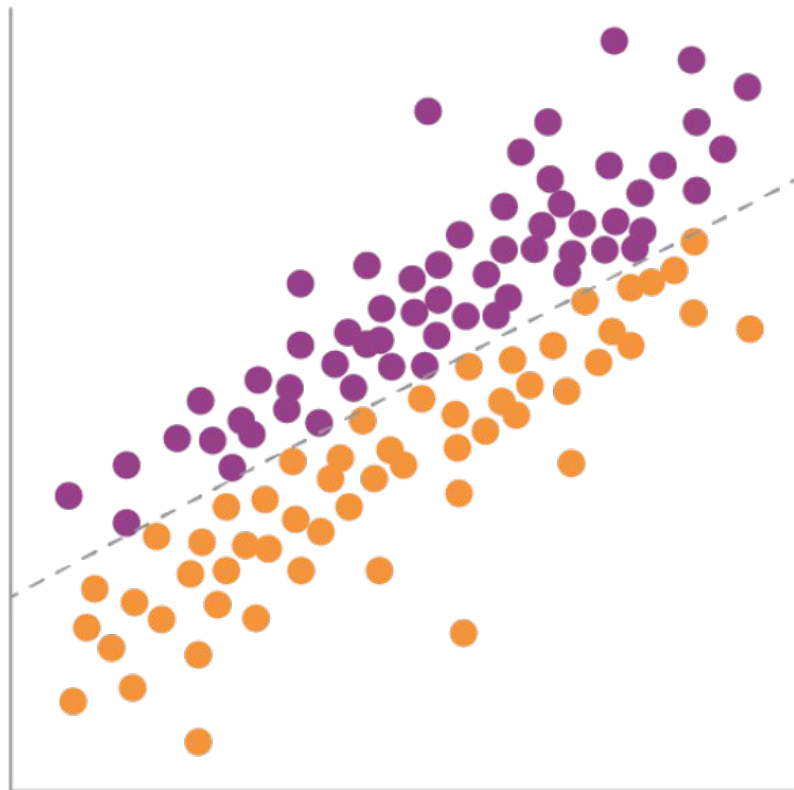
Each object can be described by multiple features:  $x_1, x_2, \dots, x_N$

Each object has a known property/class:  $y$

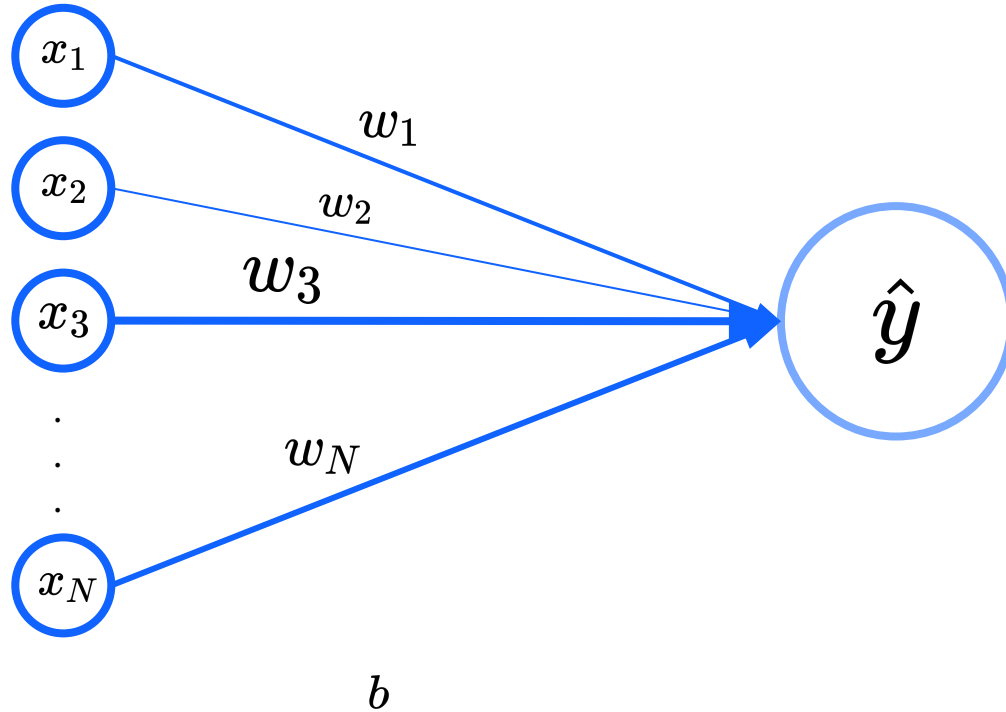
We would like to know the relationship between these features and the known property

$$\hat{y} = f(x_1, x_2, \dots, x_N) = \sigma(w_1x_1 + w_2x_2 + \dots + w_Nx_N + b)$$

# Binary Classification



# Neural Network Form



# Matrix Form

$$\begin{aligned}\hat{\mathbf{y}} &= \sigma(\mathbf{Z}) \\ &= \sigma(\mathbf{X} \cdot \mathbf{w}^{\mathbf{T}} + \mathbf{b})\end{aligned}$$

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

# Feature (Input) Matrix

$$\mathbf{x}_n = \begin{bmatrix} {}^{(1)}x_n \\ {}^{(2)}x_n \\ \cdot \\ \cdot \\ \cdot \\ {}^{(M)}x_n \end{bmatrix}_{(M,1)}$$

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_N]_{(M,N)}$$

# Target (Output) Vector

$$\mathbf{y} = \begin{bmatrix} {}^{(1)}y \\ {}^{(2)}y \\ \cdot \\ \cdot \\ \cdot \\ {}^{(M)}y \end{bmatrix}_{(M,1)}$$



# Model Parameters

$$\mathbf{w} = [w_1 \quad w_2 \quad \dots \quad w_N]_{(1,N)}$$

$$\mathbf{b} = \begin{bmatrix} b \\ b \\ \cdot \\ \cdot \\ \cdot \\ b \end{bmatrix}_{(M,1)}$$

# Binary Cross Entropy Loss

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{M} \sum (-\mathbf{y} \log_e (\hat{\mathbf{y}}) - (1 - \mathbf{y}) \log_e (1 - \hat{\mathbf{y}}))$$

# Gradient of Loss

$$\nabla \mathcal{L} = \left[ \frac{\partial \mathcal{L}}{\partial w_1} \quad \frac{\partial \mathcal{L}}{\partial w_2} \quad \dots \quad \frac{\partial \mathcal{L}}{\partial w_N} \quad \frac{\partial \mathcal{L}}{\partial b} \right]$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \cdot \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{Z}} \cdot \frac{\partial \mathbf{Z}}{\partial \mathbf{w}} = \frac{1}{M} \sum (\hat{\mathbf{y}} - \mathbf{y})^T \cdot \mathbf{X}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \cdot \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{Z}} \cdot \frac{\partial \mathbf{Z}}{\partial b} = \frac{1}{M} \sum (\hat{\mathbf{y}} - \mathbf{y})$$

# Vectorized Gradient Descent

Given dataset:  $\left\{ \left( {}^{(1)}\mathbf{x}, {}^{(1)}y \right), \left( {}^{(2)}\mathbf{x}, {}^{(2)}y \right), \dots, \left( {}^{(M)}\mathbf{x}, {}^{(M)}y \right) \right\}$

Initialize  $\mathbf{w}$  and  $b$

Repeat until converge {

$$\mathbf{w} := \mathbf{w} - \alpha \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$$

$$b := b - \alpha \frac{\partial \mathcal{L}}{\partial b}$$

}

where  $\alpha$  is learning rate