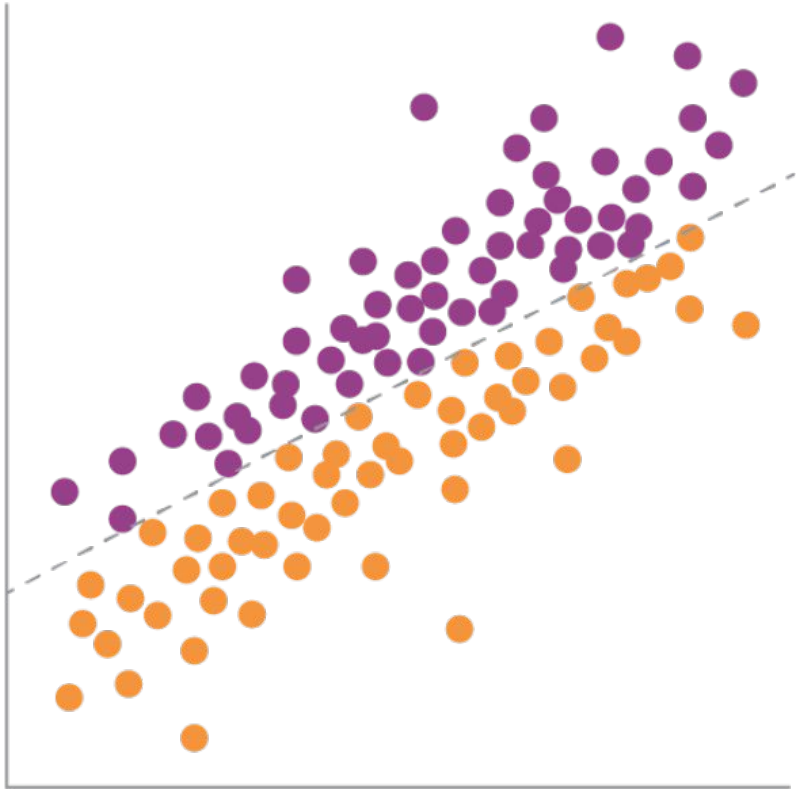
ENGR 3321: Introduction to Deep Learning for Robotics

Binary Classification

09/22/2025

Binary Classifcation



Review: Model Training

1. Load dataset: X (features), y (labels)
2. (Randomly) Initialize model parameters: w, b.
3. Evaluate the model with a metric (e.g. BCE).
4. Calculate gradient of loss.
5. Update parameters a small step on the directions descending the gradient of loss.
6. Repeat 3 to 5 until converge.

Load Dataset

A dataset with M samples:

- Each sample has N features: $x_1, x_2, \ldots, x_N$
- Each sample is labeled: y

$$\mathcal{D} = \{({}^{(1)}x_1, {}^{(1)}x_2, \ldots, {}^{(1)}x_N, {}^{(1)}y), ({}^{(2)}x_1, {}^{(2)}x_2, \ldots, {}^{(2)}x_N, {}^{(2)}y), \ldots, ({}^{(M)}x_1, {}^{(M)}x_2, \ldots, {}^{(M)}x_N, {}^{(M)}y)\}$$

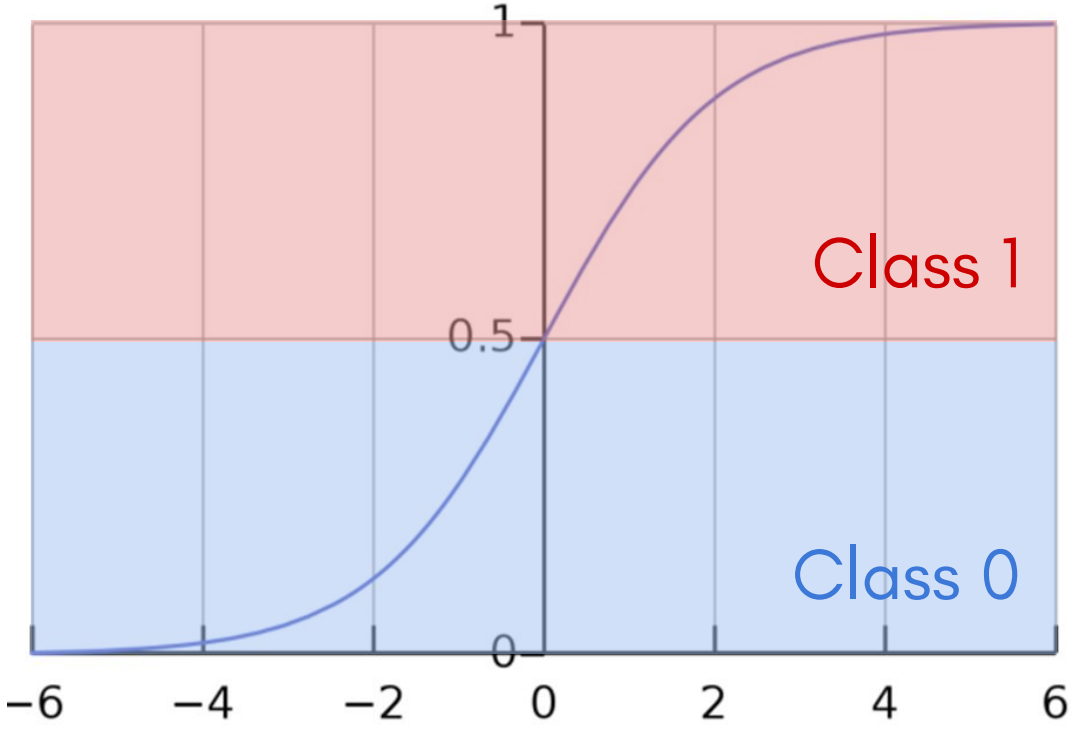
$$= \{({}^{(1)}\mathbf{x}, {}^{(1)}y), ({}^{(2)}\mathbf{x}, {}^{(2)}y), \ldots, ({}^{(M)}\mathbf{x}, {}^{(M)}y)\}$$

$${}^{(i)}y \in \{0, 1\}$$

Initialize Model

$$\underset{(M,1)}{\hat{\mathbf{y}}} = \sigma(\underset{(M,N)}{\mathbf{X}} \cdot \underset{(N,1)}{\mathbf{w}^T} + \underset{(M,1)}{\mathbf{b}}) = \underset{(M,1)}{\sigma(\mathbf{z})}$$

Sigmoid Classifcation



Binary Cross Entropy Loss

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{M} \sum_{i=1}^{M} -{}^{(i)}y \ln {}^{(i)}\hat{y} - (1 - {}^{(i)}y) \ln(1 - {}^{(i)}\hat{y}) = \overline{-\mathbf{y} \ln \hat{\mathbf{y}} - (1 - \mathbf{y}) \ln(1 - \hat{\mathbf{y}})}$$

Gradient of Loss (BCE)

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \underset{(1,N)}{\left[\frac{\partial \mathcal{L}}{\partial w_1} \quad \frac{\partial \mathcal{L}}{\partial w_2} \quad \cdots \quad \frac{\partial \mathcal{L}}{\partial w_N}\right]} = \underset{(1,M)}{\frac{1}{M}(\hat{\mathbf{y}} - \mathbf{y})^T} \cdot \underset{(M,N)}{\mathbf{X}}$$

Matrix Multiplication

$$\frac{\partial \mathcal{L}}{\partial b} = \overline{\hat{\mathbf{y}} - \mathbf{y}}$$

Gradient of Loss (MSE)

Not Recommend

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial w_1} & \frac{\partial \mathcal{L}}{\partial w_2} & \cdots & \frac{\partial \mathcal{L}}{\partial w_N} \end{bmatrix} = \frac{1}{M}[(\hat{\mathbf{y}} - \mathbf{y}) * \hat{\mathbf{y}} * (1 - \hat{\mathbf{y}})]^T \cdot \mathbf{X}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \overline{(\hat{\mathbf{y}} - \mathbf{y}) * \hat{\mathbf{y}} * (1 - \hat{\mathbf{y}})}$$

Gradient Descent

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

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 α is learning rate