ENGR 3321: Introduction to Deep Learning for Robotics

Neural Network N01: Multi-Input One-Output Model

09/17/2025

Outline

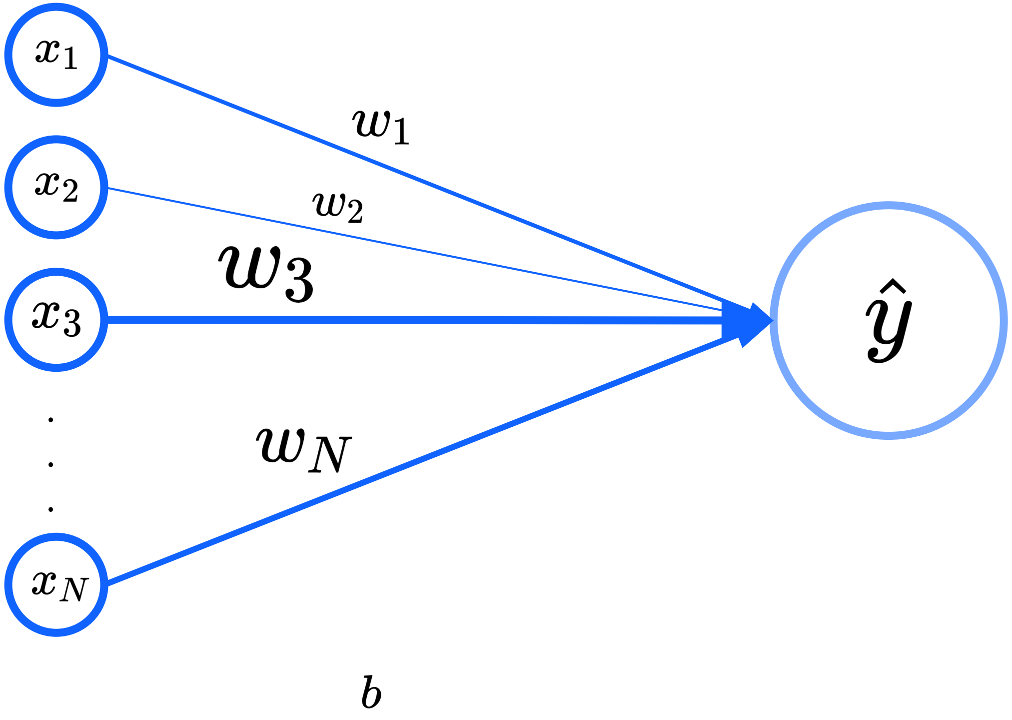
- Multi-Feature Input
- Sigmoid Activation

N-Input One-Output Linear Function

A Linear model predicts a sample's property using multiple features.

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

Neural Network Form



Feature (Input) Matrix

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

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1} & \mathbf{x_2} & \dots & \mathbf{x_N} \end{bmatrix}_{(M,N)}$$

Model Parameters

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \dots & w_N \end{bmatrix}_{(1,N)}$$

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

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. MSE).
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

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

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

Initialize Linear Model

$$\hat{\mathbf{y}} = \mathbf{X} \cdot \mathbf{w}^T + \mathbf{b}$$

$(M, 1)$ (M, N) $(N, 1)$ $(M, 1)$

Matrix Multiplication

$$^{(1)}\hat{y} = {}^{(1)}x_1 w_1 + {}^{(1)}x_2 w_2 + \cdots + {}^{(1)}x_N w_N + b$$

$$^{(2)}\hat{y} = {}^{(2)}x_1 w_1 + {}^{(2)}x_2 w_2 + \cdots + {}^{(2)}x_N w_N + b$$

$$\vdots$$

$$^{(M)}\hat{y} = {}^{(M)}x_1 w_1 + {}^{(M)}x_2 w_2 + \cdots + {}^{(M)}x_N w_N + b$$

Evaluate Model (MSE)

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{M} \sum_{i=1}^{M} ({}^{(i)}\hat{y} - {}^{(i)}y)^2 = \overline{(\hat{\mathbf{y}} - \mathbf{y})^2}$$

Gradient of Loss

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

$$\frac{\partial \mathcal{L}}{\partial w_1} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial w_1} = \frac{1}{M} \sum_{i=1}^{M} ({}^{(i)}\hat{y} - {}^{(i)}y)^{(i)}x_1$$

$$\frac{\partial \mathcal{L}}{\partial w_2} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial w_2} = \frac{1}{M} \sum_{i=1}^{M} ({}^{(i)}\hat{y} - {}^{(i)}y)^{(i)}x_2$$

$$\cdots$$

$$\frac{\partial \mathcal{L}}{\partial w_N} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial w_N} = \frac{1}{M} \sum_{i=1}^{M} ({}^{(i)}\hat{y} - {}^{(i)}y)^{(i)}x_N$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial b} = \frac{1}{M} \sum_{i=1}^{M} ({}^{(i)}\hat{y} - {}^{(i)}y)$$

Vectorized Gradient

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

$(1, N)$ $(1, M)$ (M, N)

Matrix Multiplication

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

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 {

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

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

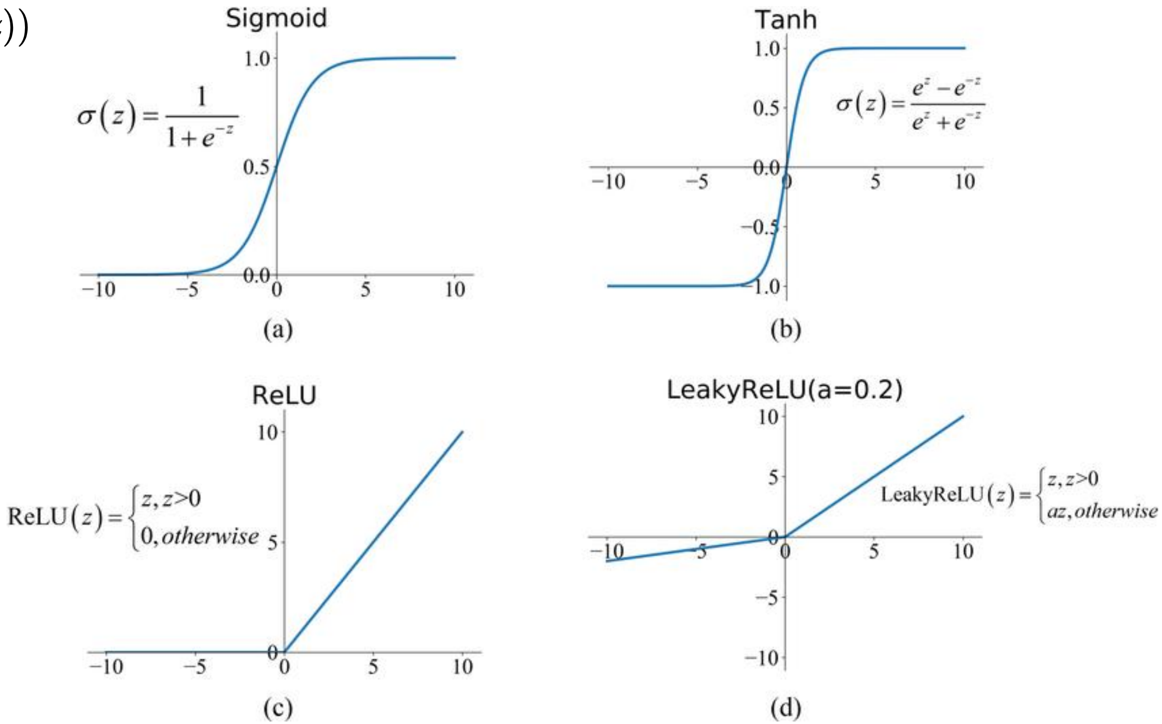
}

where α is learning rate

Activation Functions

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

$$\sigma'(z) = 1 - \sigma^2(z)$$



$$\mathrm{ReLU}'(z) = \begin{cases} 1, & z > 0 \\ 0, & otherwise \end{cases}$$

$$\mathrm{LeakyReLU}'(z) = \begin{cases} 1, & z > 0 \\ a, & otherwise \end{cases}$$

Gradient with Sigmoid Activation

$$\hat{\mathbf{y}} = \sigma(\mathbf{z}) = \sigma(\mathbf{X} \cdot \mathbf{w}^T + \mathbf{b})$$

$$\frac{\partial \mathcal{L}}{\partial w_n} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial w_n} = \frac{1}{M} \sum_{i=1}^{M} ({}^{(i)}\hat{y} - {}^{(i)}y)^{(i)}\hat{y}(1 - {}^{(i)}\hat{y})^{(i)}x_n$$

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

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

Element-wise Multiplication

Matrix Multiplication