ENGR 3321: Introduction to Deep Learning for Robotics

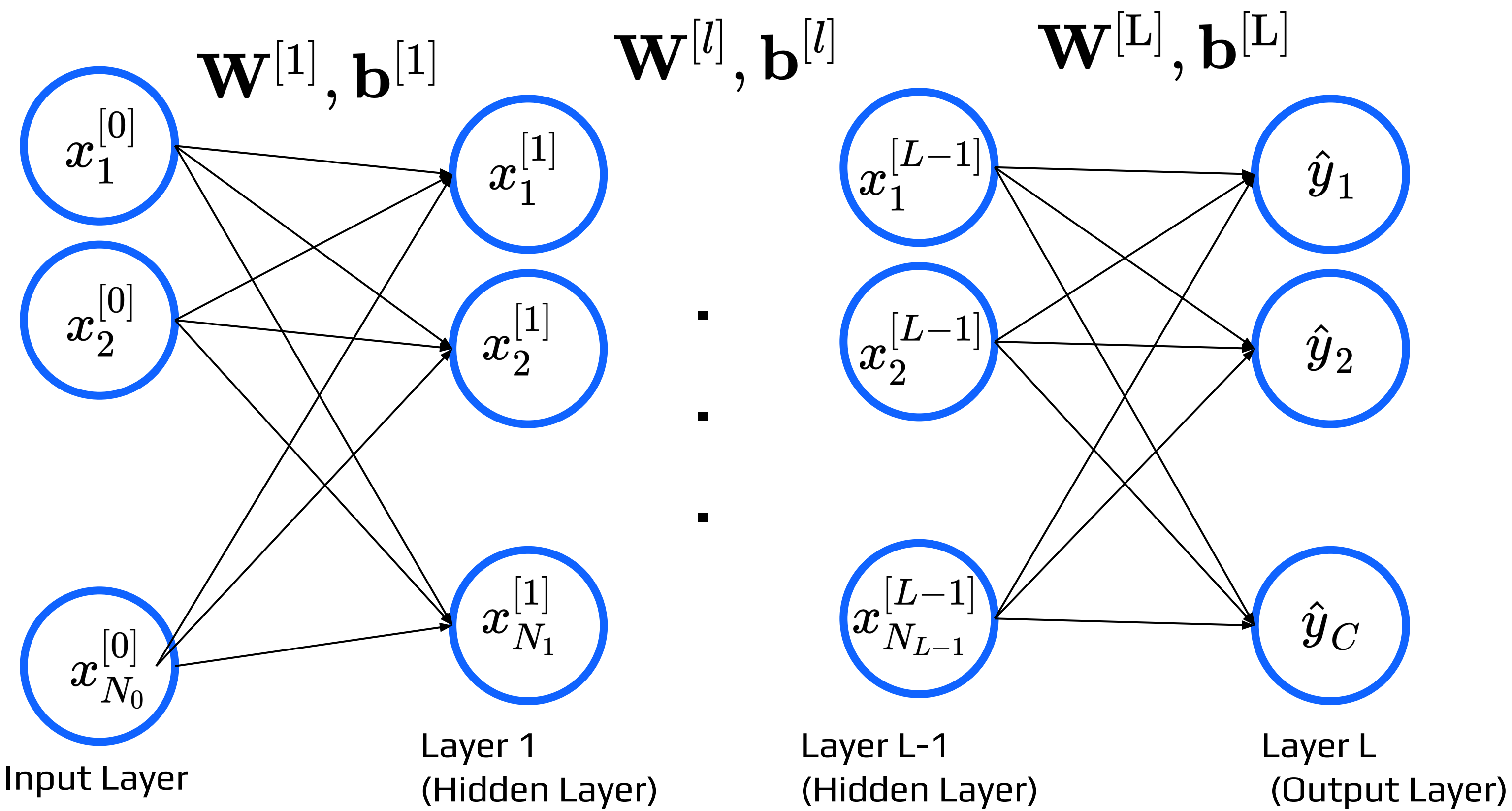
Neural Network NNN: Multi-Layer Perceptron Model

11/06/2023

Outline

- Representations
- ReLU Activation
- One-Hot Encoding
- Softmax Activation
- Multi-Class Cross Entropy

Multi-Layer Perceptron Model



Individual Representation

$$x_n^{[l]} = \sigma\left(w_{1n}^{[l]} x_1^{[l-1]} + w_{2n}^{[l]} x_2^{[l-1]} + \ldots + w_{N_{l-1}n}^{[l]} x_{N_{l-1}}^{[l-1]} + b_n^{[L]}\right)$$

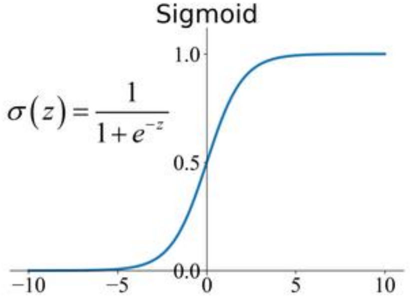
Matrix Form

$$\mathbf{X}^{[l]} = a\left(\mathbf{Z}^{[l]}\right) = a\left(\mathbf{X}^{[l-1]} \cdot \mathbf{W}^{[\mathrm{l}]\mathrm{T}} + \mathbf{b}^{[l]}\right)$$

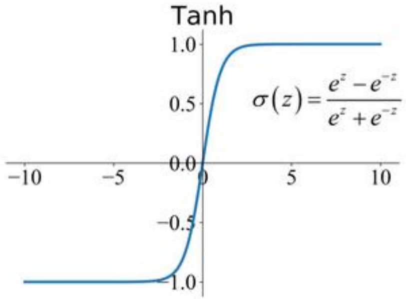
$a(\cdot)$ activation function

Activation Functions

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

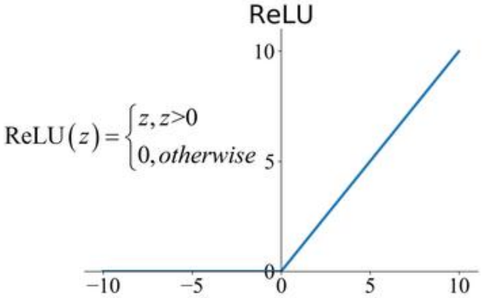


(a)



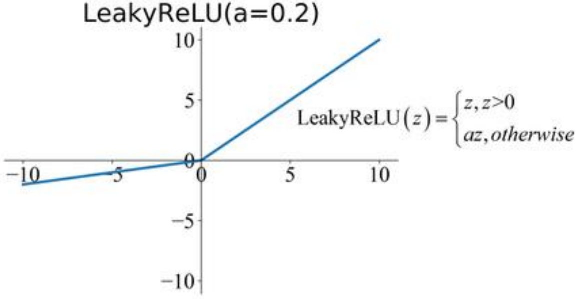
(b)

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



(c)

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



(d)

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

Feature (Input) Matrix

$$\mathbf{X}^{[0]} = \begin{bmatrix} {}^{(1)}x_1^{[0]} & {}^{(1)}x_2^{[0]} & \dots & {}^{(1)}x_{N_0}^{[0]} \\ {}^{(2)}x_1^{[0]} & {}^{(2)}x_2^{[0]} & \dots & {}^{(2)}x_{N_0}^{[0]} \\ & & \dots & \\ {}^{(M)}x_1^{[0]} & {}^{(1)}x_2^{[0]} & \dots & {}^{(M)}x_{N_0}^{[0]} \end{bmatrix}_{(M,N_0)}$$

Trainable Parameters

$$\mathbf{W}^{[l]} = \begin{bmatrix} w_{11}^{[l]} & w_{21}^{[l]} & \cdots & w_{N_{l-1}1}^{[l]} \\ w_{12}^{[l]} & w_{22}^{[l]} & \cdots & w_{N_{l-1}2}^{[l]} \\ & & \cdots & \\ w_{1N_l}^{[l]} & w_{2N_l}^{[l]} & \cdots & w_{N_{l-1}N_l}^{[l]} \end{bmatrix}_{(N_l, N_{l-1})}$$

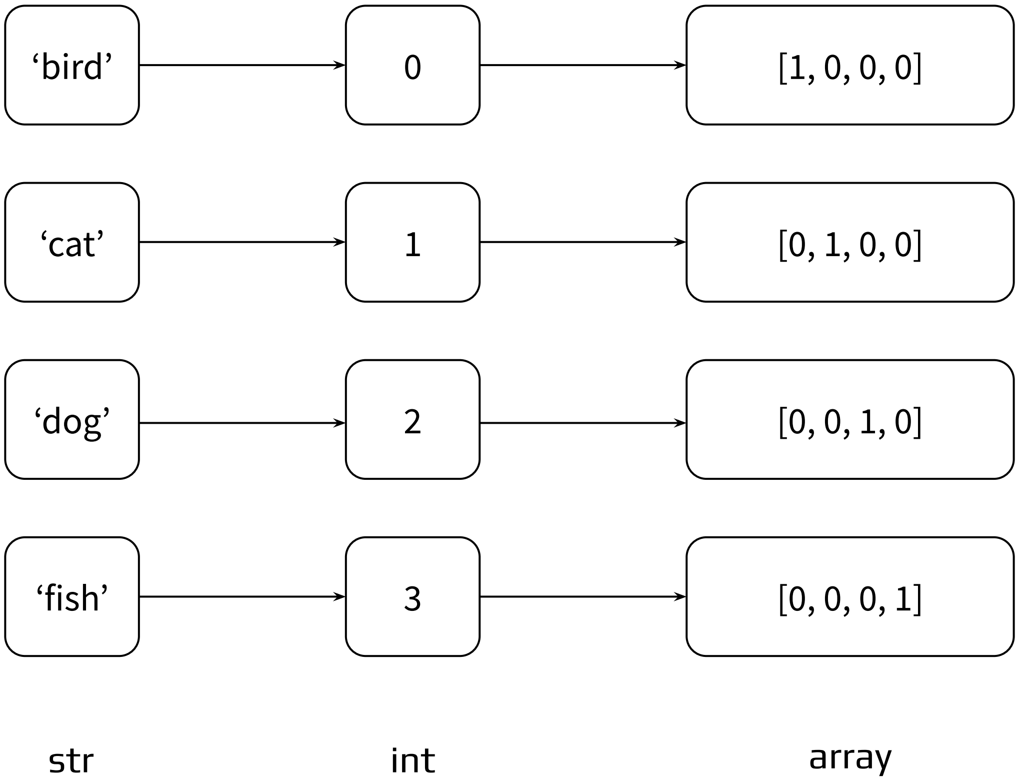
$$\mathbf{b}^{[l]} = \begin{bmatrix} b_1^{[l]} & b_2^{[l]} & \cdots & b_{N_l}^{[l]} \end{bmatrix}_{(1, N_l)}$$

Target and Prediction

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

NOT RECOMMEND

One-Hot Encoding on Targets



Softmax Activation on Predictions

$$\hat{y}_c = \frac{e^{z_c^{[L]}}}{\sum_{c=1}^{C} e^{z_c^{[L]}}}, \forall c = 1, \ldots, C$$

$$\sum \begin{bmatrix} {}^{(m)}\hat{y}_1 & {}^{(m)}\hat{y}_2 & \cdots & {}^{(m)}\hat{y}_C \end{bmatrix} = 1$$

Probability of the m-th sample being predicted as a member in class 1

Multi-Class Classification

$$\mathbf{Y} = \begin{bmatrix} {}^{(1)}y_1 & {}^{(1)}2 & \dots & {}^{(1)}y_C \\ {}^{(2)}y_1 & {}^{(2)}y_2 & \dots & {}^{(2)}y_C \\ & & \dots & \\ {}^{(M)}y_1 & {}^{(M)}y_2 & & {}^{(M)}y_C \end{bmatrix}_{(M,C)}$$

$$\hat{\mathbf{Y}} = \begin{bmatrix} {}^{(1)}\hat{y}_1 & {}^{(1)}\hat{y}_2 & \dots & {}^{(1)}\hat{y}_C \\ {}^{(2)}\hat{y}_1 & {}^{(2)}\hat{y}_2 & \dots & {}^{(2)}\hat{y}_C \\ & & \dots & \\ {}^{(M)}\hat{y}_1 & {}^{(M)}\hat{y}_2 & & {}^{(M)}\hat{y}_C \end{bmatrix}_{(M,C)}$$

Forward Propagation

$$\mathbf{Z}^{[l]} = \mathbf{X}^{[l-1]} \cdot \mathbf{W}^{[l]\mathrm{T}} + \mathbf{b}^{[l]}$$

$$\mathbf{Z}^{[l]} = \begin{bmatrix} {}^{(1)}x_1^{[l-1]} & {}^{(1)}x_2^{[l-1]} & \dots & {}^{(1)}x_{N_{l-1}}^{[l-1]} \\ {}^{(2)}x_1^{[l-1]} & {}^{(2)}x_2^{[l-1]} & \dots & {}^{(2)}x_{N_{l-1}}^{[l-1]} \\ & & \dots & \\ {}^{(M)}x_1^{[l-1]} & {}^{(M)}x_2^{[l-1]} & \dots & {}^{(M)}x_{N_{l-1}}^{[l-1]} \end{bmatrix} \cdot \begin{bmatrix} w_{11}^{[l]} & w_{12}^{[l]} & \dots & w_{1N_l}^{[l]} \\ w_{21}^{[l]} & w_{22}^{[l]} & \dots & w_{2N_l}^{[l]} \\ & & \dots & \\ w_{N_{l-1}1}^{[l]} & w_{N_{l-1}2}^{[l]} & \dots & w_{N_{l-1}N_l}^{[l]} \end{bmatrix} + \begin{bmatrix} b_1^{[l]} & b_2^{[l]} & \dots & b_{N_l}^{[l]} \\ b_1^{[l]} & b_2^{[l]} & \dots & b_{N_l}^{[l]} \\ & & \dots & \\ b_1^{[l]} & b_2^{[l]} & \dots & b_{N_l}^{[l]} \end{bmatrix}$$

$$\mathbf{X}^{[l]} = a\left(\mathbf{Z}^{[l]}\right)$$

Special Case:

$$\hat{\mathbf{Y}} = a\left(\mathbf{X}^{[\mathrm{L}-1]}\mathbf{W}^{[\mathrm{L}]\mathrm{T}} + \mathbf{b}^{[\mathrm{L}]}\right) = a\left(\mathbf{Z}^{[\mathrm{L}]}\right) = \mathbf{X}^{[\mathrm{L}]}$$

Multi-Class Cross Entropy Loss

$$\mathcal{L}\left(\hat{\mathbf{Y}}, \mathbf{Y}\right) = \frac{1}{M} \sum_{m=1}^{M} \left[\sum_{c=1}^{C} \left(-^{(m)} y_c \ln^{(m)} \hat{y}_c\right)\right]$$

Back-Propagation

$$\nabla \mathcal{L} = \begin{bmatrix} \cdots & \frac{\partial \mathcal{L}}{\partial w_{l-1,l}{}^{[l]}} & \cdots & \frac{\partial \mathcal{L}}{\partial b_l{}^{[l]}} & \cdots \end{bmatrix}$$

$$d\mathbf{Z}^{[L]} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{Y}}} \cdot \frac{\partial \hat{\mathbf{Y}}}{\partial \mathbf{Z}^{[L]}} = \underline{\hat{\mathbf{Y}} - \mathbf{Y}}$$

NOTE: Only valid if

1. last layer is softmax activated;
2. Prediction is evaluated by cross entropy loss

For l from L to 1

$$d\mathbf{W}^{[l]} = d\mathbf{Z}^{[l]} \cdot \frac{\partial \mathbf{Z}^{[l]}}{\partial \mathbf{W}^{[\mathbf{l}]}} = d\mathbf{Z}^{[l]T} \cdot \mathbf{X}^{[l-1]}$$

$$d\mathbf{b}^{[l]} = d\mathbf{Z}^{[l]} \cdot \frac{\partial \mathbf{Z}^{[l]}}{\partial \mathbf{b}^{[\mathbf{l}]}} = mean\left(d\mathbf{Z}^{[l]}, \text{ axis=0, keepdims=True}\right)$$

$$d\mathbf{X}^{[l-1]} = d\mathbf{Z}^{[l]} \cdot \frac{\partial \mathbf{Z}^{[l]}}{\partial \mathbf{X}^{[l-1]}} = d\mathbf{Z}^{[l]} \cdot \mathbf{W}^{[l]}$$

$$d\mathbf{Z}^{[l-1]} = d\mathbf{X}^{[l-1]} * a'\left(\mathbf{Z}^{[l-1]}\right)$$

Gradient Descent Optimization

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

Initialize $\mathbf{W}^{[l]},\ \mathbf{b}^{[l]}$

Repeat until converge {

compute $\mathcal{L}\left(\hat{\mathbf{Y}}, \mathbf{Y}\right)$

compute $\nabla\mathcal{L}$

$\mathbf{W}^{[l]} := \mathbf{W}^{[l]} - \alpha \cdot d\mathbf{W}^{[l]}$

$\mathbf{b}^{[l]} := \mathbf{b}^{[l]} - \alpha \cdot d\mathbf{b}^{[l]}$

}

where α is learning rate