ENGR 3321: Introduction to Deep Learning for Robotics

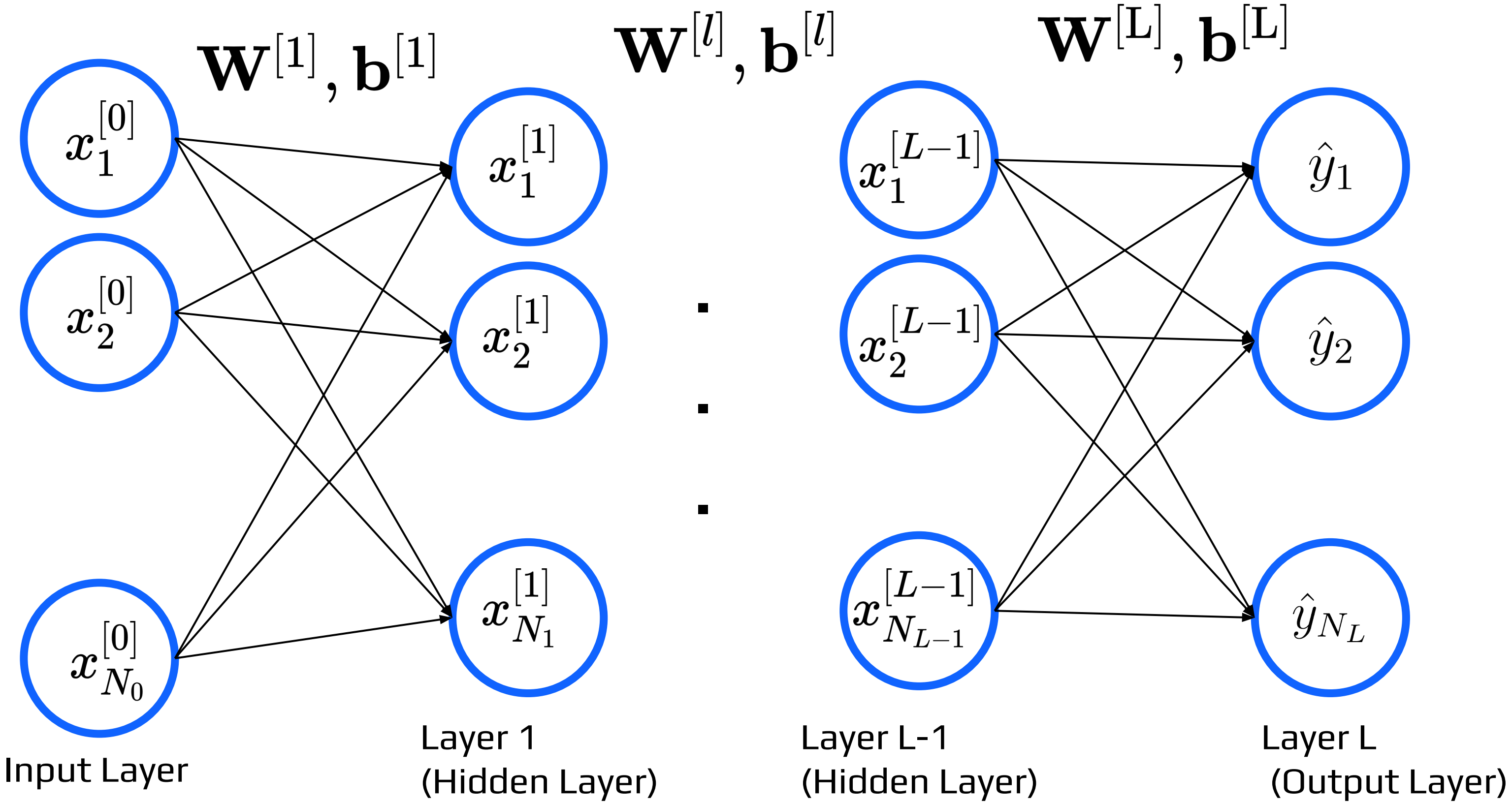
Neural Network NNN: Multi-Layer Perceptron Model

10/13/2025

Outline

- Generic Neural Network Model
- ReLU Activation
- Softmax Activation
- One-Hot Encoding
- Multi-Class Classification
- Cross Entropy Loss
- Stochastic Gradient Descent

MLP - Graphical Representation



Input Feature 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

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

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

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:

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

Prediction (output) Matrix

$$\hat{\mathrm{Y}} = \begin{bmatrix} {}^{(1)}y_1 & {}^{(1)}y_2 & \cdots & {}^{(1)}y_{N_L} \\ {}^{(2)}y_1 & {}^{(2)}y_2 & \cdots & {}^{(2)}y_{N_L} \\ & & \cdots & \\ {}^{(M)}y_1 & {}^{(M)}y_2 & \cdots & {}^{(M)}y_{N_L} \end{bmatrix}_{(M, N_L)}$$

MLP - Mathematical Representation

$$\underset{(M, N_l)}{\mathbf{X}^{[l]}} = a(\underset{(M, N_l)}{\mathbf{Z}^{[l]}}) = a(\underset{(M, N_{l-1})}{\mathbf{X}^{[l-1]}} \cdot \underset{(N_{l-1}, N_l)}{\mathbf{W}^{[l]T}} + \underset{(1, N_l)}{\mathbf{b}^{[l]}})$$

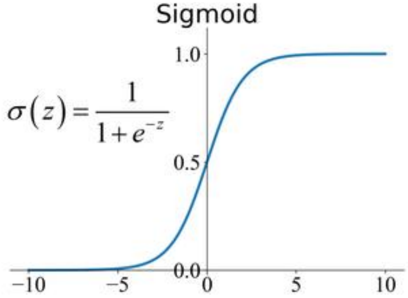
$a(\cdot)$ activation function

$$x_n^{[l]} = a(w_{1n}^{[l]} x_1^{[l-1]} + w_{2n}^{[l]} x_2^{[l-1]} + \cdots + w_{N_{l-1}n}^{[l]} x_{N_{l-1}}^{[l-1]} + b_n^{[l]})$$

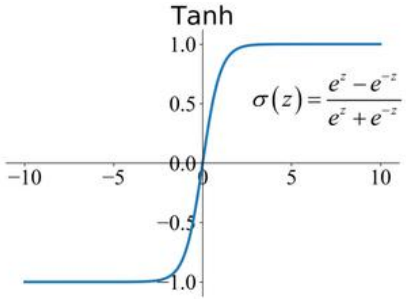
Individual Feature

ReLU Activation Functions

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

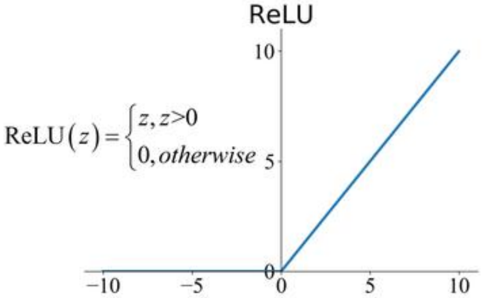


(a)



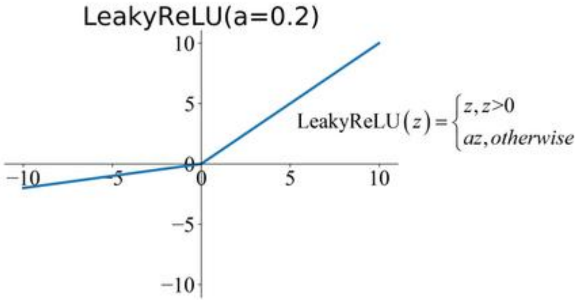
(b)

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



(c)

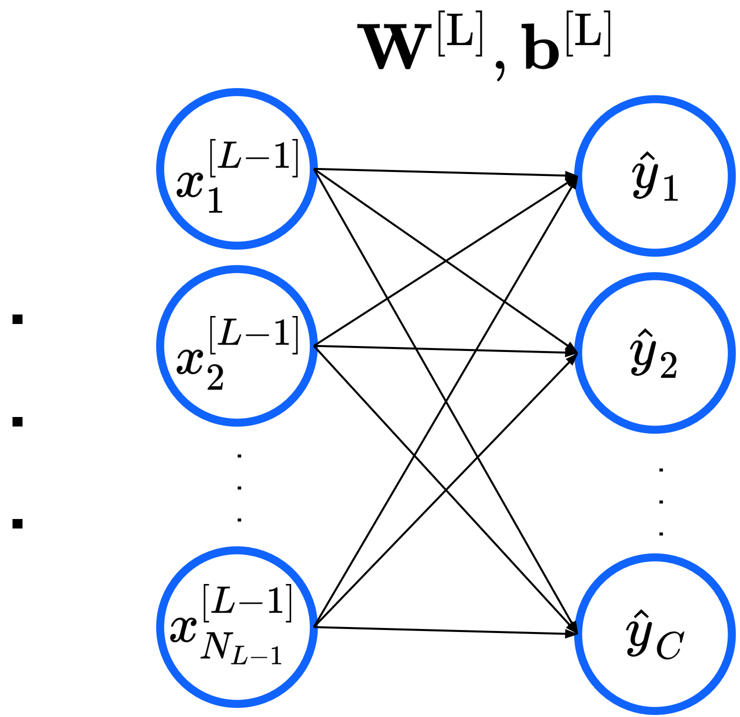
$$\mathrm{ReLU}'(z) = \begin{cases} 1, & z > 0 \\ 0, & otherwise \end{cases}$$



(d)

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

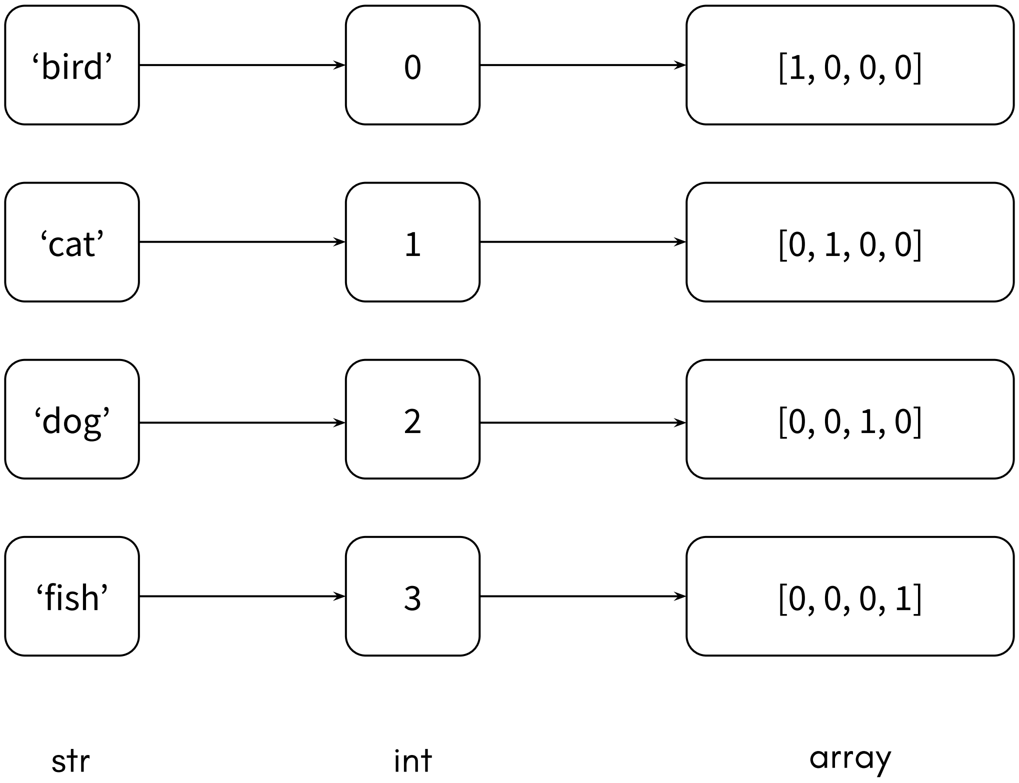
Multi-Class Classification



Multi-Class Classification

$$\mathbf{Y} = \begin{bmatrix} {}^{(1)}y_1 & {}^{(1)}2 & \cdots & {}^{(1)}y_C \\ {}^{(2)}y_1 & {}^{(2)}y_2 & \cdots & {}^{(2)}y_C \\ & & \cdots & \\ {}^{(M)}y_1 & {}^{(M)}y_2 & & {}^{(M)}y_C \end{bmatrix}_{(M,C)} \qquad \hat{\mathbf{Y}} = \begin{bmatrix} {}^{(1)}\hat{y}_1 & {}^{(1)}\hat{y}_2 & \cdots & {}^{(1)}\hat{y}_C \\ {}^{(2)}\hat{y}_1 & {}^{(2)}\hat{y}_2 & \cdots & {}^{(2)}\hat{y}_C \\ & & \cdots & \\ {}^{(M)}\hat{y}_1 & {}^{(M)}\hat{y}_2 & & {}^{(M)}\hat{y}_C \end{bmatrix}_{(M,C)}$$

One-Hot Encoding on Labels



Softmax Activation on Predictions

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

$$\sum \left[{}^{(m)}\hat{y}_1 \quad {}^{(m)}\hat{y}_2 \quad \cdots \quad {}^{(m)}\hat{y}_C \right] = 1$$

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

Review: Model Training

1. Prepare datasets: train, validation
2. (Randomly) Initialize model parameters: weights, biases.
3. Evaluate the model with a metric (e.g. CE, MSE).
4. Calculate gradients of loss.
5. Update parameters a small step on the directions descending the gradient of loss.
6. Repeat 3 to 5 until converge.

Prepare Datasets: Training

A dataset with M_{tr} samples:

- Each sample has N_0 features: $\mathbf{x} = x_1, x_2, \ldots, x_{N_0}$
- Each sample is labeled: $\mathbf{y} = y_1, y_2, \ldots, y_{N_L}$

$$\mathcal{D} = \{({}^{(1)}\mathbf{x}, {}^{(1)}\mathbf{y}), ({}^{(2)}\mathbf{x}, {}^{(2)}\mathbf{y}), \ldots, ({}^{(M_{tr})}\mathbf{x}, {}^{(M_{tr})}\mathbf{y})\}$$

Prepare Datasets: Validation

A dataset with M_{va} ($M_{va} < M_{tr}$) samples:

- Each sample has N_0 features: $\tilde{\mathbf{x}} = \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{N_0}$
- Each sample is labeled: $\tilde{\mathbf{y}} = \tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{N_L}$
- Validation dataset can be used to evaluate model.
- Validation dataset does not participate into model updating

$$\tilde{\mathcal{D}} = \{({}^{(1)}\tilde{\mathbf{x}}, {}^{(1)}\tilde{\mathbf{y}}), ({}^{(2)}\tilde{\mathbf{x}}, {}^{(2)}\tilde{\mathbf{y}}), \ldots, ({}^{(M_{va})}\tilde{\mathbf{x}}, {}^{(M_{va})}\tilde{\mathbf{y}})\}$$

Model Evaluation Metrics

Mean Squared Error (MSE)

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

Binary Cross Entropy (BCE)

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

Cross Entropy (CE)

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

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. Cross Entropy Loss + Softmax Activation;
2. Mean Squared Error Loss + Linear Activation;
3. Binary Cross Entropy Loss + Sigmoid Activation

For l from L to 1

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

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

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

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

Stochastic Gradient Descent Optimization

- Given dataset: $\mathcal{D} = \{({}^{(1)}\mathbf{x}, {}^{(1)}\mathbf{y}), ({}^{(2)}\mathbf{x}, {}^{(2)}\mathbf{y}), \dots, ({}^{(M)}\mathbf{x}, {}^{(M)}\mathbf{y})\}$
- Set hyper-parameters: number of iterations/epochs, learning rate(α)
- Initialize model parameters: $\mathbf{W}^{[l]}, \mathbf{b}^{[l]}$
- Repeat until converge
  - Extract a batch of data: $\mathcal{B} = \{({}^{(1)}\mathbf{x}, {}^{(1)}\mathbf{y}), ({}^{(2)}\mathbf{x}, {}^{(2)}\mathbf{y}), \dots, ({}^{(M_b)}\mathbf{x}, {}^{(M_b)}\mathbf{y})\}, \ M_b <= M$
  - Make predictions
  - Evaluate model
  - Compute gradients
  - Update model parameters (back-propagation) with batch

$$\boldsymbol{W}^{[l]} = \boldsymbol{W}^{[l]} - \alpha \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^{[l]}}$$

$$\boldsymbol{b}^{[l]} = \boldsymbol{b}^{[l]} - \alpha \frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^{[l]}}$$