An Overview on Deep Neural Networks: Part 1

Introduction

Common Tasks in Deep Learning

- Classification
  - Binary
  - Multi-Class
  - Multi-Label
- Regression

Binary Classification

- Class 1: “cat”
- Class 0: “not cat”

🐈 → [0.98] 🐎 → [0.03] 🐶 → [0.1]

Multi-Class Classification

"cat", "dog", "neither"

 $\rightarrow \begin{bmatrix} 0.97 \\ 0.02 \\ 0.01 \end{bmatrix}$

 $\rightarrow \begin{bmatrix} 0.09 \\ 0.86 \\ 0.05 \end{bmatrix}$

 $\rightarrow \begin{bmatrix} 0.01 \\ 0.03 \\ 0.96 \end{bmatrix}$

 $\rightarrow \begin{bmatrix} 0.04 \\ 0.03 \\ 0.93 \end{bmatrix}$

Multi-Label Classification

“cat”, “dog”, “elephant”

 $\rightarrow \begin{bmatrix} 0.98 \\ 0.1 \\ 0.01 \end{bmatrix}$  $\rightarrow \begin{bmatrix} 0.95 \\ 0.1 \\ 0.9 \end{bmatrix}$  $\rightarrow \begin{bmatrix} 0.93 \\ 0.95 \\ 0.2 \end{bmatrix}$  $\rightarrow \begin{bmatrix} 0.2 \\ 0.1 \\ 0.02 \end{bmatrix}$  $\rightarrow \begin{bmatrix} 0.01 \\ 0.04 \\ 0.03 \end{bmatrix}$

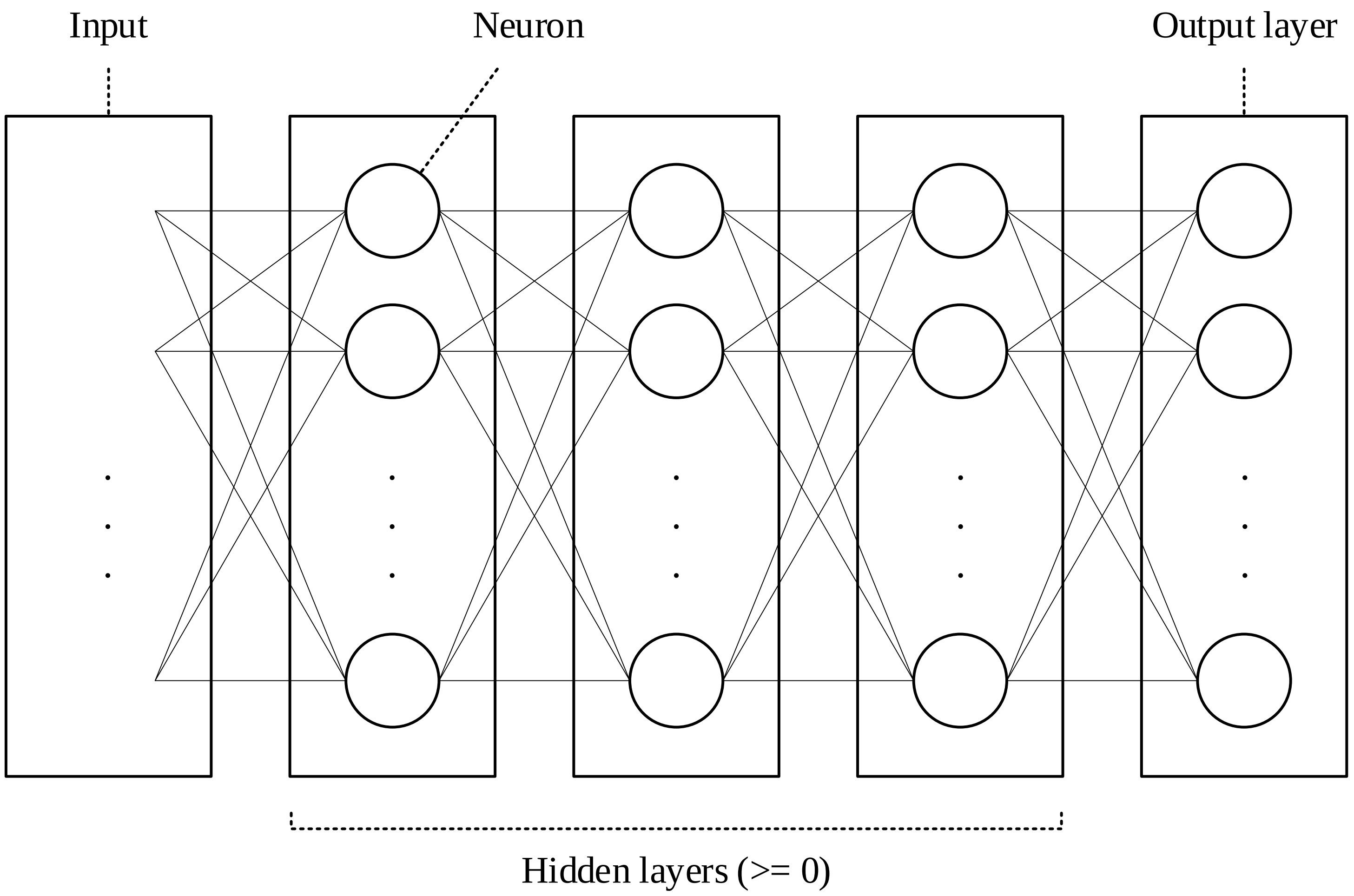
Regression

Housing:

(area, number of bedrooms, location, etc.) -> price

DNN Structure

Forward Propagation: Predict the output for the given input.



Backward Propagation: Updating the network's parameters by comparing the predicted output with the actual output

Forward Propagation

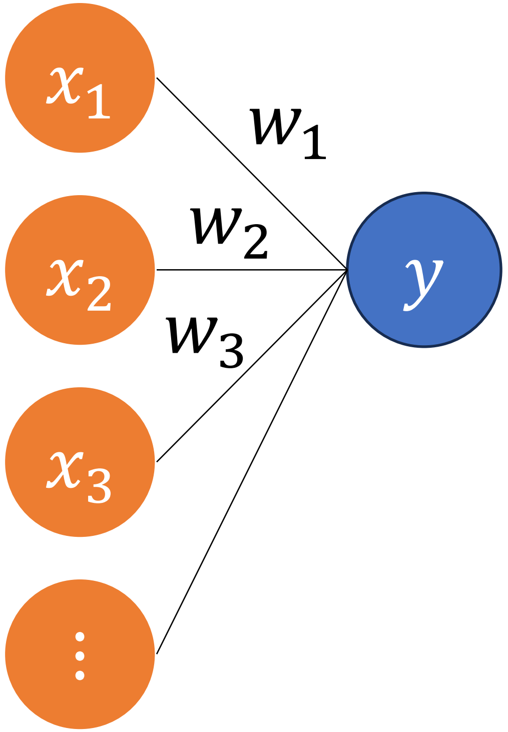
Forward Propagation: A Neuron

$$z = x_1 w_1 + x_2 w_2 + x_3 w_3 + \cdots + b$$
$$y = \sigma(z)$$

Vectorized form:

$$\vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \end{bmatrix} \quad \vec{x} = [x_1 \quad x_2 \quad \ldots]$$

$$z = \vec{x}\vec{w} + b \qquad\qquad y = \sigma(z)$$



Forward Propagation: A Layer of Neurons

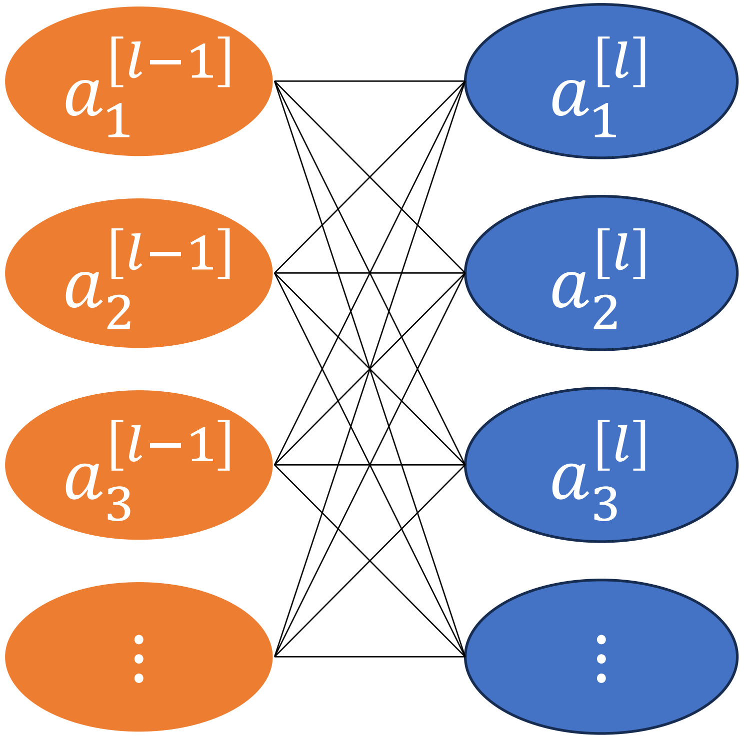
$$z_1^{[l]} = \vec{a}^{[l-1]}\vec{w}_1^{[l]} + b_1^{[l]}$$
$$z_2^{[l]} = \vec{a}^{[l-1]}\vec{w}_2^{[l]} + b_2^{[l]}$$

…

Vectorized form:

$$W^{[l]} = \begin{bmatrix} | & | & | \\ \vec{w}_1^{[l]} & \vec{w}_2^{[l]} & \dots \\ | & | & | \end{bmatrix} \qquad \vec{b}^{[l]} = \begin{bmatrix} b_1^{[l]} & b_2^{[l]} & \dots \end{bmatrix}$$

$$\vec{z}^{[l]} = \vec{a}^{[l-1]}W^{[l]} + \vec{b}^{[l]} \qquad \vec{a}^{[l]} = \sigma\left(\vec{z}^{[l]}\right)$$



Forward Propagation: A Dataset with n items

$$\vec{a}_i^{[0]} = \vec{x}_i \qquad \forall i = 1,2,\ldots,m$$

$$\vec{z}_i^{[l]} = \vec{a}_i^{[l-1]} W^{[l]} + \vec{b}^{[l]} \qquad \forall l = 1,2,\ldots,L \qquad \forall i = 1,2,\ldots,m$$

$$\vec{a}_i^{[l]} = \sigma^{[l]}\left(\vec{z}_i^{[l]}\right) \qquad \forall l = 1,2,\ldots,L \qquad \forall i = 1,2,\ldots,m$$

$$\vec{\hat{y}}_i = \vec{a}_i^{[L]} \qquad \forall i = 1,2,\ldots,m$$

Forward Propagation:
A Dataset with m items

$$X = \begin{bmatrix} - & \vec{x}_1 & - \\ - & \vec{x}_2 & - \\ - & \vdots & - \end{bmatrix}$$

$$A^{[0]} = X$$

$$Z^{[l]} = A^{[l-1]}W^{[l]} + b^{[l]} \qquad \forall l = 1,2,\dots,L$$

$$A^{[l]} = \sigma^{[l]}\left(Z^{[l]}\right) \qquad \forall l = 1,2,\dots,L$$

$$\hat{Y} = A^{[L]}$$

Activation Function Examples

- Rectified Linear Unit (ReLU)
- Sigmoid
- Softmax

ReLU

- Usually applied to the output layer in regression.
- One of the common activation functions in the hidden layers of modern models.

$$\sigma^{[l]}\left(Z^{[l]}\right)_{i,j} = \max\left\{0, Z_{i,j}^{[l]}\right\}$$

Sigmoid

- Usually applied to the output layer in binary or multi-label classification.

$$\sigma^{[l]}\left(Z^{[l]}\right)_{i,j} = \frac{1}{1 + \exp\left(-Z_{i,j}^{[l]}\right)}$$

Softmax

- Usually applied to the output layer in multi-class classification.

$$\sigma^{[l]}\left(Z^{[l]}\right)_{i,j} = \frac{\exp\left(Z_{i,j}^{[l]}\right)}{\sum_{k=1}^{m_h^{[l]}} \exp\left(Z_{i,k}^{[l]}\right)}$$