An Overview on Deep Neural Networks: Part 1

Preface

Common Tasks in Deep Learning

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

Binary Classification

- Class 1: “cat”
- Class 0: “not cat”

 $\rightarrow [1]$

 $\rightarrow [0]$

 $\rightarrow [0]$

Multi-Class Classification

“cat”, “dog”, “neither”

 $\rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

 $\rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

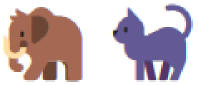
 $\rightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

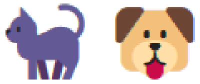
 $\rightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

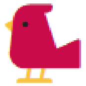
Multi-Label Classification

“cat”, “dog”, “elephant”

 $\rightarrow \begin{bmatrix}1\\0\\0\end{bmatrix}$

 $\rightarrow \begin{bmatrix}1\\0\\1\end{bmatrix}$

 $\rightarrow \begin{bmatrix}1\\1\\0\end{bmatrix}$

 $\rightarrow \begin{bmatrix}0\\0\\0\end{bmatrix}$

 $\rightarrow \begin{bmatrix}0\\0\\0\end{bmatrix}$

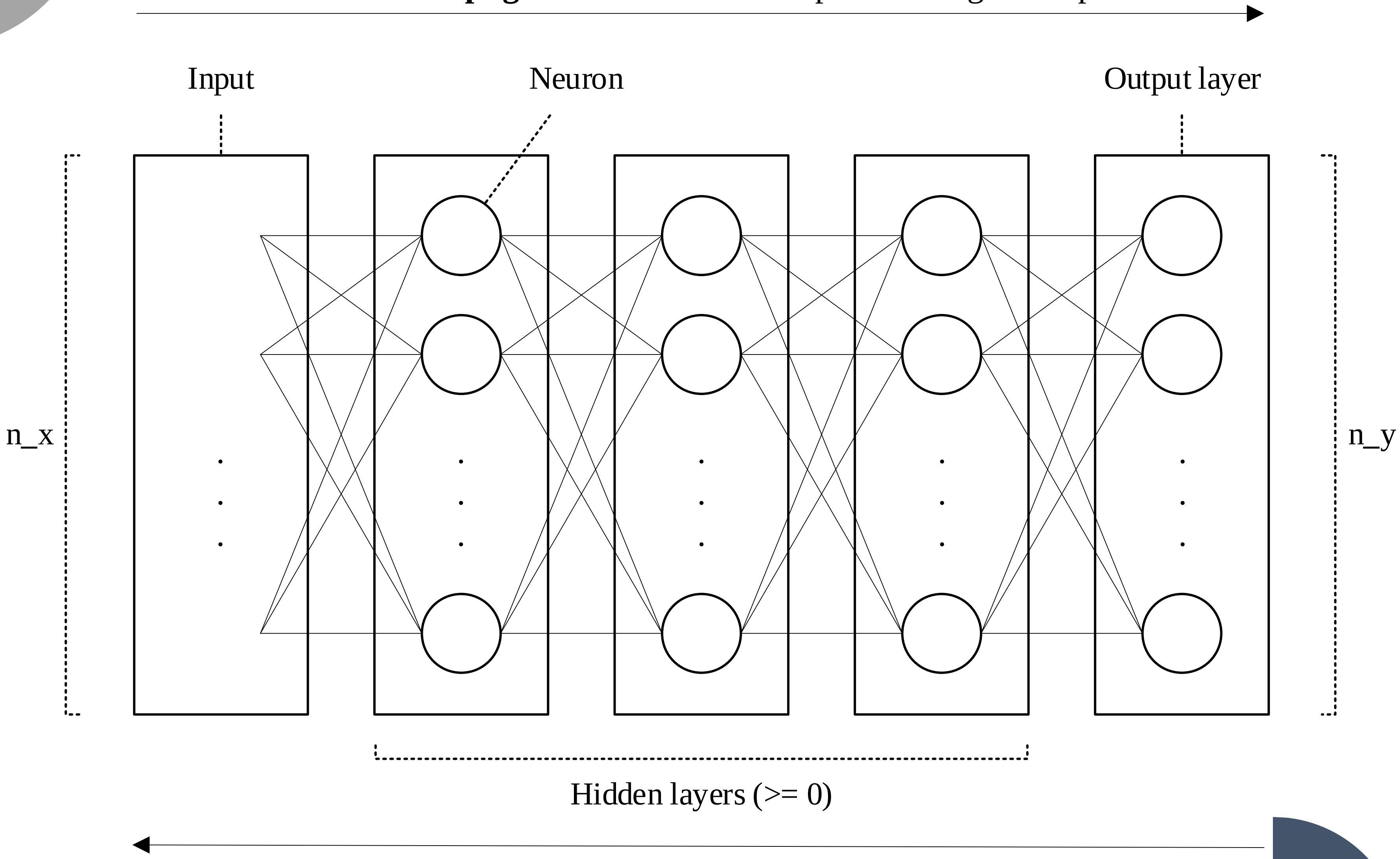
Regression

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

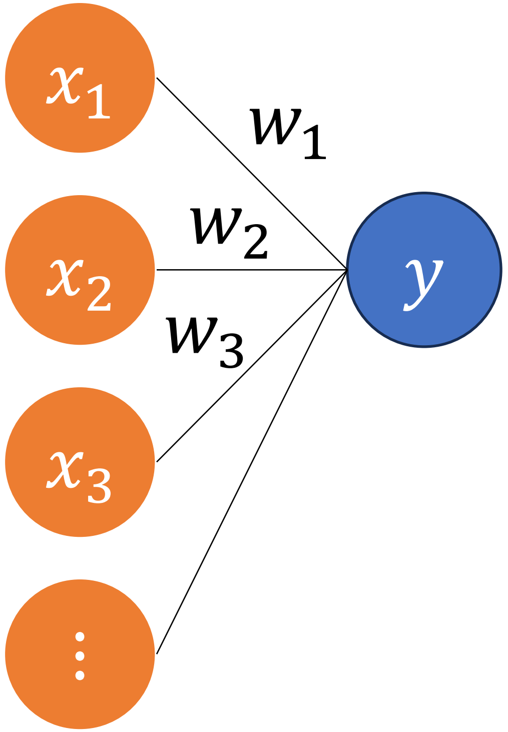
Forward Propagation: A Neuron

$$z = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + b$$
$$y = \sigma(z)$$

Vectorized form:

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

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



Forward Propagation: A Layer of Neurons

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

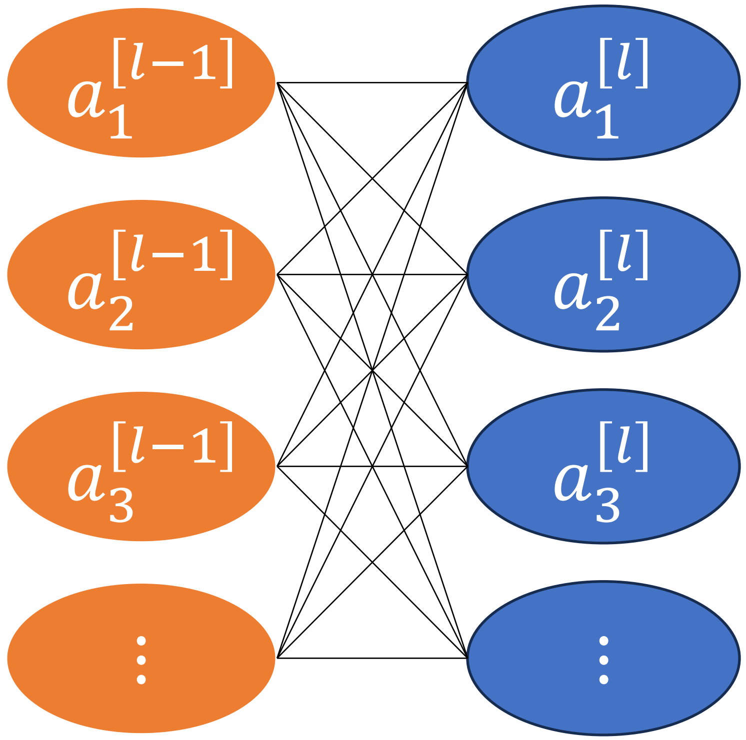
$$z_2^{[l]} = \vec{w}_2^{[l]} . \vec{a}^{[l-1]} + b_2^{[l]}$$

...

Vectorized form:

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

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



Forward Propagation: A Dataset with m items

$$\vec{a}_j^{[0]} = \vec{x}_j \qquad\qquad \forall j = 1,2,\dots,m$$

$$\vec{z}_j^{[l]} = W^{[l]} \times \vec{a}_j^{[l-1]} + \vec{b}^{[l]} \qquad \forall l = 1,2,\dots,L \qquad \forall j = 1,2,\dots,m$$

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

$$\vec{\hat{y}}_j = \vec{a}_j^{[L]} \qquad\qquad \forall j = 1,2,\dots,m$$

Forward Propagation: A Dataset with m items

$$X = \begin{bmatrix} | & | & | \\ \vec{x}_1 & \vec{x}_2 & \vec{x}_3 \\ | & | & | \end{bmatrix}$$

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

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

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

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

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.

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