


# 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

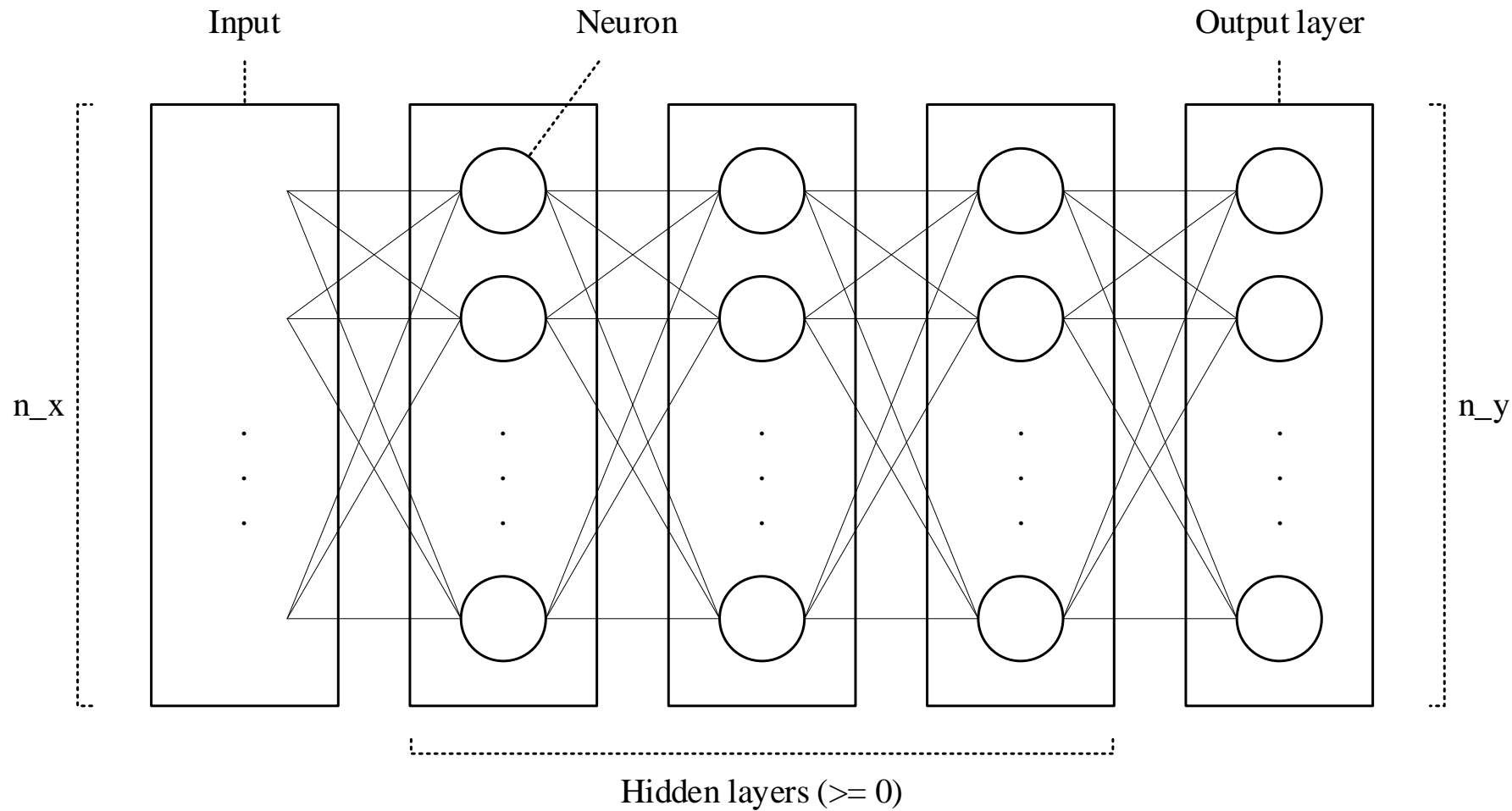
Housing:

(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

$$z = w_1x_1 + w_2x_2 + w_3x_3 + \dots + b$$

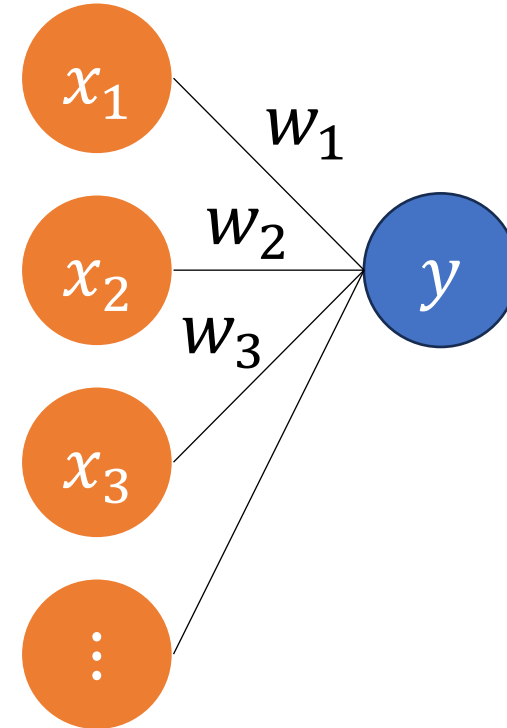
$$y = \sigma(z)$$

Vectorized form:

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

$$z = \vec{w} \cdot \vec{x} + b$$

$$y = \sigma(z)$$



# Forward Propagation: A Layer of Neurons

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

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

...

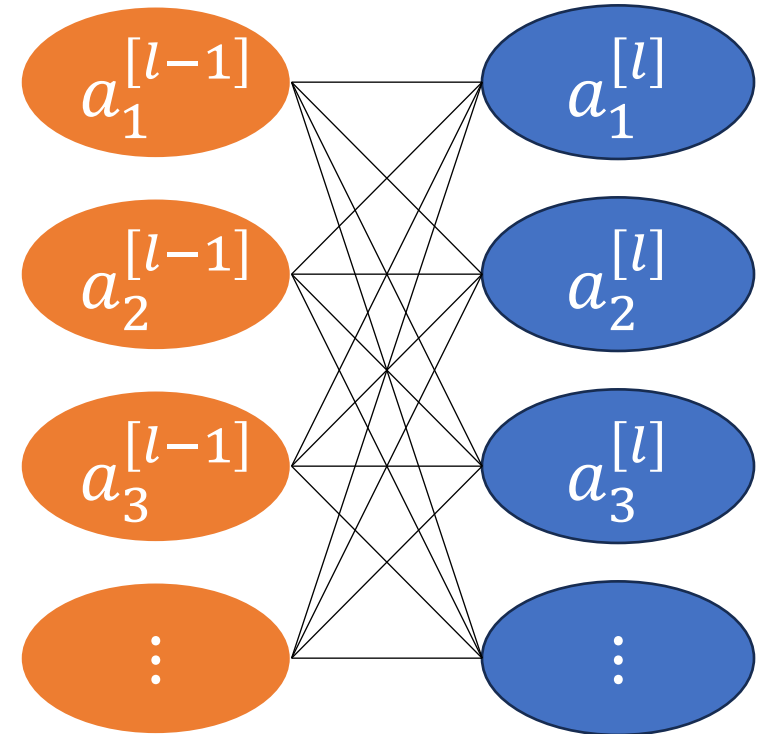
Vectorized form:

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

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

$$\vec{b}^{[l]} = \begin{bmatrix} b_1^{[l]} \\ b_2^{[l]} \\ \vdots \end{bmatrix}$$

$$\vec{a}^{[l]} = \sigma(\vec{z}^{[l]})$$



# Forward Propagation:

## A Dataset with $m$ items

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

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

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

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

# Forward Propagation:

A Dataset with  $m$  items

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

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

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

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

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

# 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.

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



# Sigmoid

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

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

# Softmax

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

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