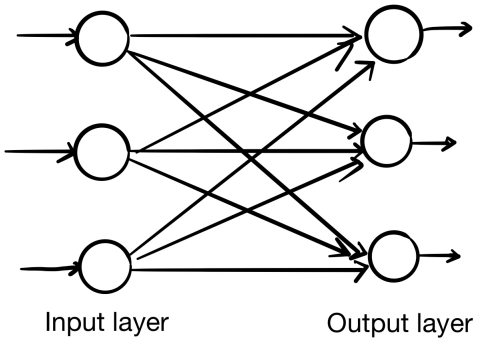
1: The First Problem

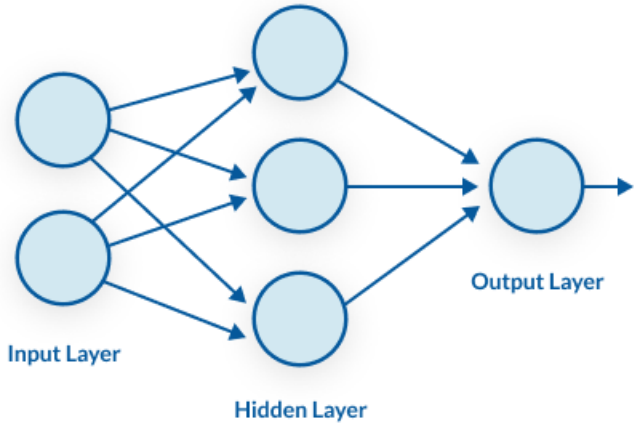
(a) Single-Layer Neural Networks:

A single-layer neural network represents the most simple form of neural network, in which there is only one layer of input nodes that send weighted inputs to a subsequent layer of receiving nodes, or in some cases, one receiving node.



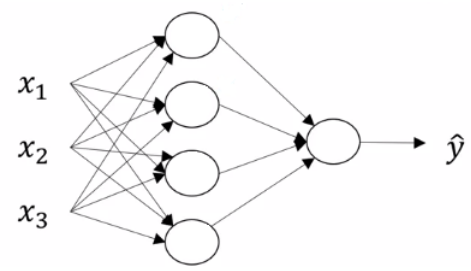
(b) Multi-Layer Neural Networks:

A multi-layer neural network contains more than one layer of artificial neurons or nodes. Typically, they have at least one input layer, which sends weighted inputs to a series of hidden layers, and an output layer at the end.



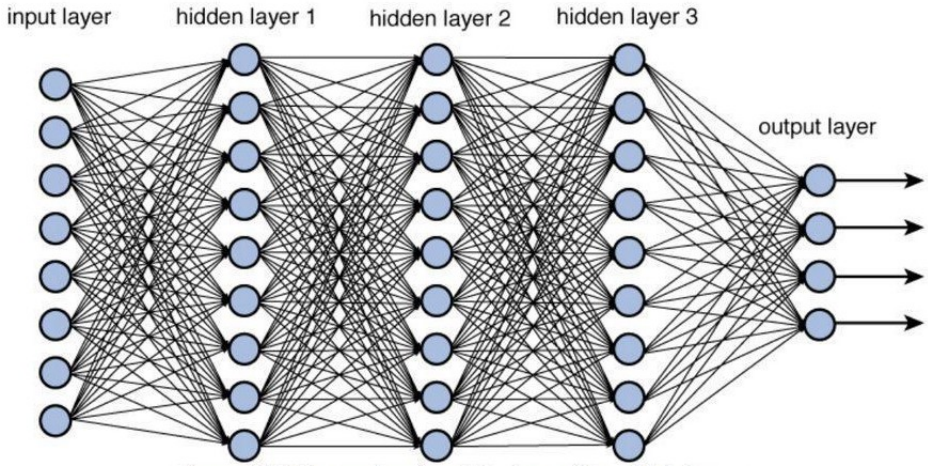
(c) Shallow Neural Networks:

A shallow neural network simply consist of one hidden layer between the input and the output.



(d) Deep Neural Networks:

A deep neural network simply consist of more than more than equal to 2 hidden layers between the input and the output.



2: The Second problem

(a) Algorithm:

```
import numpy as np

#create neurons of neural networks
class Neuron():
    def __init__(self,weights,bias):
        self.weights = weights
        self.bias = bias

#activation function
def sigmoid(x):
    return 1 / (1 + np.exp(-x))


def feedforward(w,b,inputs):
    h1_1 = np.dot(w[0:1],inputs) + b
    h1_2 = np.dot(w[1:2],inputs) + b
```

```
    t_1 = np.array([sigmoid(h1_1),sigmoid(h1_2)])
    h2_1 = np.dot(w[2:3],t_1) + b
    h2_2 = np.dot(w[3:4],t_1) + b
    return sigmoid(h2_1), sigmoid(h2_2)


#parameters w and b
weights = np.array([[3,1],[2,4],[3,2],[5,1]])
bias = 1
n = Neuron(weights,bias)

set_x = input()
set_y = input()

inputs = np.array([int(set_x),int(set_y)])


print('The first ouput is ' + str(feedforward(n.weights,n.bias,inputs)[0]))
print('The second output is ' + str(feedforward(n.weights,n.bias,inputs)[1]))
```

(b) Output:

```
 ▲ Assignments/Computational Intelligence/Lab1 py3 lab1.py
1
2
The first ouput is [[0.99750893]]
The second output is [[0.99907761]]
```