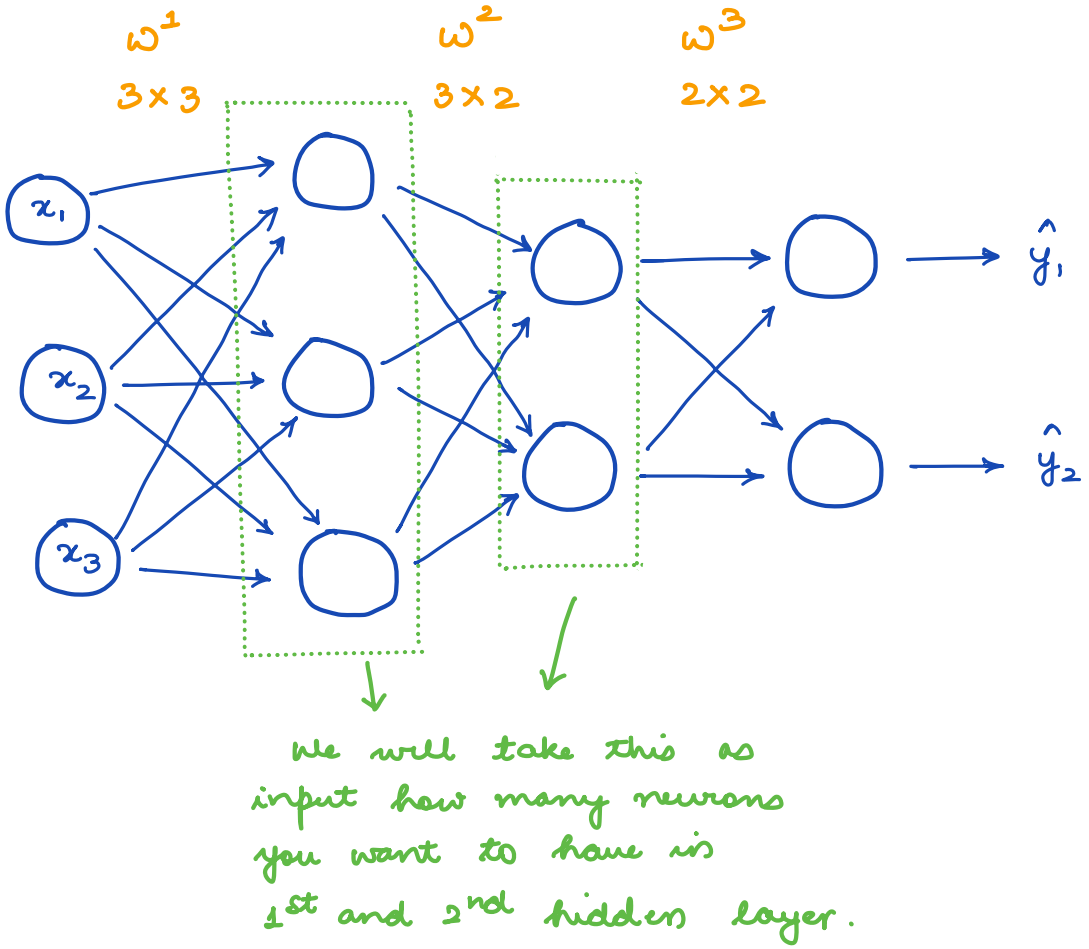
Part 1 - Implementation

In this video we will implement a simple 3 layer network and in the next part we will do forward propagation.



CODE :

```
import numpy as np

# Model Parameters
input_size = 2      # no_of_features
layers = [4,3]      # no of neurons in 1st and 2nd layer
output_size = 2
```

W1 = np.random.randn(input_size, layers[0])
print(W1)

} Create a 2x4 matrix

```
import numpy as np
```

```
# Model Parameters
input_size = 2
layers = [4,3]
output_size = 2
```

```
W1 = np.random.randn(input_size, layers[0])
print(W1)
```

```
[[-0.6118095  -0.36986931  2.29482804 -0.52029743]
 [ 0.38156869  1.20296949 -0.47847123 -0.44714696]]
```

```
class NeuralNetwork:
    def __init__(self, input_size, layers, output_size):
        np.random.seed(0)
        model = {}  #Dictionary
        #First Layer
        model['W1']= np.random.randn(input_size, layers[0])
        model['b1'] = np.zeros((1, layers[0]))
        #Second Layer
        model['W2']= np.random.randn(layers[0], layers[1])
        model['b2'] = np.zeros((1, layers[1]))
        #Third/Output Layer
        model['W3']= np.random.randn(layers[1],output_size)
        model['b3'] = np.zeros((1, output_size))

        self.model = model
```

```python
class NeuralNetwork :

    def __init__(self, input_size, layers, output_size) :

        np.random.seed(0)

        model = {} # Dictionary

        # First Layer
        model['W1'] = np.random.randn(input_size, layers[0])
        model['b1'] = np.zeros((1, layers[0]))

        # Second Layer
        model['W2'] = np.random.randn(layers[0], layers[1])
        model['b2'] = np.zeros((1, layers[1]))

        # Third Layer
        model['W3'] = np.random.randn(layers[1], output_size)
        model['b3'] = np.zeros((1, layers[2]))

        self.model = model
```

} creates a matrix of size 2×4

} creates a matrix of size 4×3

} creates a matrix of size 3×2

output_size