In [26]:
```
import numpy as np
```

In [27]:
```
#Input1 parameters
input_vector1=np.array([1, 1]).reshape(1,2)
target1 = 0.9

input_vector2 = np.array([-1,-1]).reshape(1,2)
target2 = 0.05

weight_layer_1 = np.array([[0.3, 0.3], [0.3, 0.3]])
weight_layer_2 = np.array([0.8, 0.8]).reshape(2,1)

eta = 1

bias = 0
```

In [28]:
```
class Layer:
    #initialize layer class
    def __init__(self, layer_weight, eta = 0,  bias = None):
        self.layer_weight = layer_weight
        self.layer_bias = bias
        self.eta = eta

    def sigmoid(self, x):
        return 1/(1+np.exp(-x))

    def get_layer_output_vector(self, inputs):
        self.activity = np.dot(inputs ,self.layer_weight)+self.layer_bias
        self.output_vector = 1/(1+np.exp(-self.activity))
        return self.output_vector

    def get_error_vector(self, desired_output):
        self.littleE_vector = desired_output - self.output_vector
        return self.littleE_vector

    def calc_delta_k(self, target):
        self.littleE = target - self.output_vector
        self.bigE = 0.5*self.littleE**2
        self.delta_k = self.littleE*(1-self.output_vector)*self.output_vector
        return self.delta_k
```

```
    def calc_delta_j(self, delta_k, next_layer_weights):
        self.delta_j = (1-self.output_vector)*self.output_vector*(delta_k*next_layer_weights).T
        return self.delta_j

    def calc_delta_weights (self, delta, prev_layer_input):
        self.delta_weights = self.layer_weight + self.eta*delta*prev_layer_input.T
        self.delta_bias = self.layer_bias + self.eta*delta
        return self.delta_weights, self.delta_bias

    def update_layer_weights(self):
        self.layer_weight = self.delta_weights
        self.layer_bias = self.delta_bias
```

In [29]:
```
Layer1=Layer(weight_layer_1, eta, bias)
Layer2=Layer(weight_layer_2, eta, bias)
```

Question 1

After training using Method 1, what is the output of the network (the activation value of the output node) when the network is presented with an input of [1.0, 1.0]? Answer to 4 significant decimal digits.

In [30]:
```
#Train network for method 1:
for i in range(0,15):
    #Input1
    Layer1.get_layer_output_vector(input_vector1)
    Layer2.get_layer_output_vector(Layer1.output_vector)
    #print("Input 1, Iteration: ", i, " Layer output ", Layer2.output_vector)
    Layer2.calc_delta_k(target1)
    Layer2.calc_delta_weights(Layer2.delta_k, Layer1.output_vector)
    Layer1.calc_delta_j(Layer2.delta_k, Layer2.layer_weight)
    Layer1.calc_delta_weights(Layer1.delta_j, input_vector1)
    Layer1.update_layer_weights()
    Layer2.update_layer_weights()
    #Input2
    Layer1.get_layer_output_vector(input_vector2)
    Layer2.get_layer_output_vector(Layer1.output_vector)
    #print("Input 2, Iteration: ", i, " Layer output ", Layer2.output_vector)
    Layer2.calc_delta_k(target2)
```

```
    Layer2.calc_delta_weights(Layer2.delta_k, Layer1.output_vector)
    Layer1.calc_delta_j(Layer2.delta_k, Layer2.layer_weight)
    Layer1.calc_delta_weights(Layer1.delta_j, input_vector2)
    Layer1.update_layer_weights()
    Layer2.update_layer_weights()
    #print("---------")
```

In [31]:
```
Layer1.get_layer_output_vector(input_vector1)
Layer2.get_layer_output_vector(Layer1.output_vector)
```

Out[31]:
```
array([[0.65832087]])
```

Question 2

After training the network with Method 1, what is the value of Big E at the end of the training when the network is presented with an input of [1.0, 1.0]? Answer to 4 significant decimal digits.

In [32]:
```
Layer2.calc_delta_k(target1)
Layer2.bigE
```

Out[32]:
```
array([[0.0292044]])
```

Question 3

After training the network with Method 1, what is the output value when the network is presented with an input of [-1.0, -1.0]? Answer to 4 significant decimal digits.

In [33]:
```
Layer1.get_layer_output_vector(input_vector2)
Layer2.get_layer_output_vector(Layer1.output_vector)
```

Out[33]:
```
array([[0.38176596]])
```

Question 4

After training with Method 1, what is the value of Big E when the network is presented with the input [-1.0, -1.0]? Answer to 4 significant decimal digits.

In [34]:
```
Layer2.calc_delta_k(target2)
Layer2.bigE
```

Out[34]:
```
array([[0.05503433]])
```

Question 5

After training the network with Method 2, what is the network output when presented with input [1.0, 1.0]? Answer to 4 significant decimal digits.

In [35]:
```
#Reset the network
Layer1=Layer(weight_layer_1, eta, bias)
Layer2=Layer(weight_layer_2, eta, bias)
```

In [36]:
```
#Train network for method 2:
for i in range(0,15):
    #Input1
    Layer1.get_layer_output_vector(input_vector1)
    Layer2.get_layer_output_vector(Layer1.output_vector)
    #print("Input 1, Iteration: ", i, " Layer output ", Layer2.output_vector)
    Layer2.calc_delta_k(target1)
    Layer2.calc_delta_weights(Layer2.delta_k, Layer1.output_vector)
    Layer1.calc_delta_j(Layer2.delta_k, Layer2.layer_weight)
    Layer1.calc_delta_weights(Layer1.delta_j, input_vector1)
    Layer1.update_layer_weights()
    Layer2.update_layer_weights()
for i in range(0,15):
    #Input2
    Layer1.get_layer_output_vector(input_vector2)
    Layer2.get_layer_output_vector(Layer1.output_vector)
    #print("Input 2, Iteration: ", i, " Layer output ", Layer2.output_vector)
    Layer2.calc_delta_k(target2)
    Layer2.calc_delta_weights(Layer2.delta_k, Layer1.output_vector)
    Layer1.calc_delta_j(Layer2.delta_k, Layer2.layer_weight)
    Layer1.calc_delta_weights(Layer1.delta_j, input_vector2)
    Layer1.update_layer_weights()
    Layer2.update_layer_weights()
    #print("---------")
```

In [37]:
```
Layer1.get_layer_output_vector(input_vector1)
Layer2.get_layer_output_vector(Layer1.output_vector)
```

Out[37]:
```
array([[0.42165763]])
```

Question 6

After training the network with Method 2, what is the value of Big E when the network is presented with the input [1.0, 1.0]? Answer to 4 significant decimal digits.

In [38]:
```
Layer2.calc_delta_k(target1)
Layer2.bigE
```

Out[38]:
```
array([[0.11440571]])
```

Question 7

After training using Method 2, what is the output of the network when presented with an input of [-1.0, -1.0]? Answer to 4 significant decimal digits.

In [39]:
```
Layer1.get_layer_output_vector(input_vector2)
Layer2.get_layer_output_vector(Layer1.output_vector)
```

Out[39]:
```
array([[0.28384481]])
```

Question 8

After training with Method 2, what is the value of Big E when the network is presented with an input of [-1.0, -1.0]? Answer to 4 significant decimal digits.

In [40]:
```
Layer2.calc_delta_k(target2)
Layer2.bigE
```

Out[40]:
```
array([[0.0273417]])
```