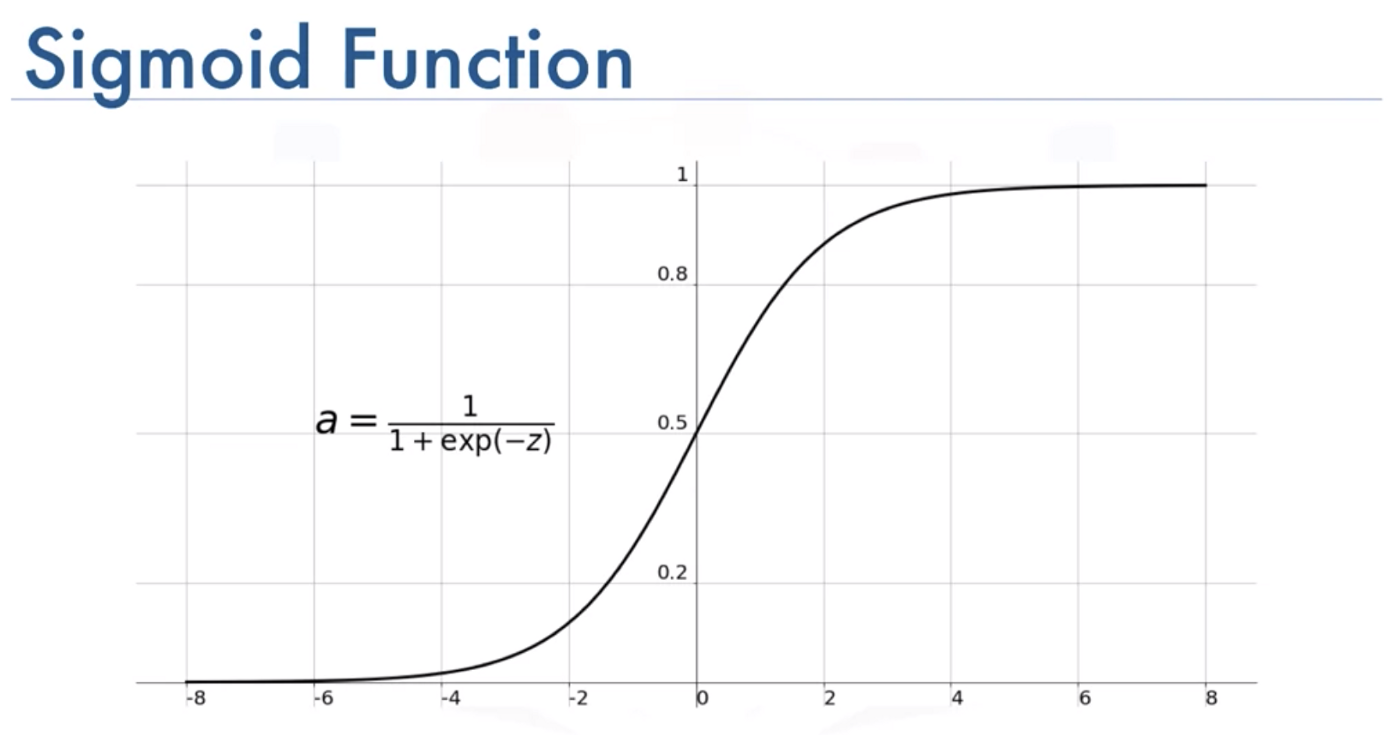
1. According to the task the neural network consists of two neurons, so

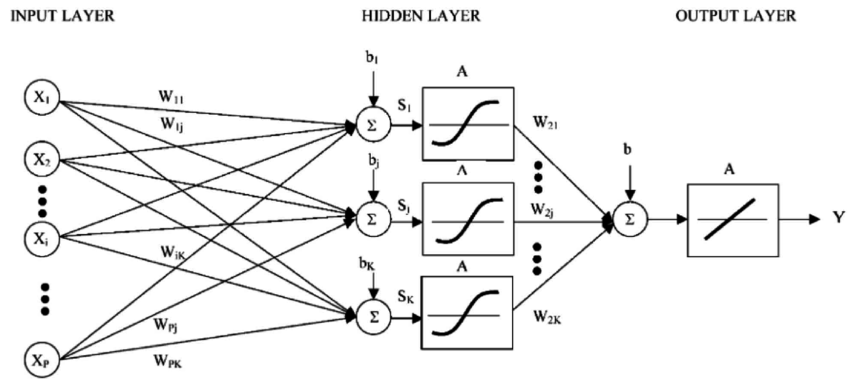
inputs of the one-layer neural network will be equal 2 and outputs = 2, and the

number of synaptic weights = 4. Each neuron has a sigmoid activation

function.



Feed-forward neural network with sigmoid activation function

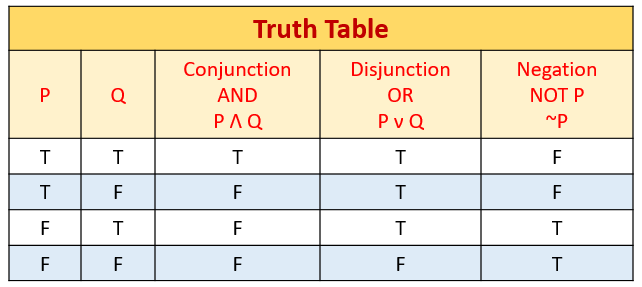


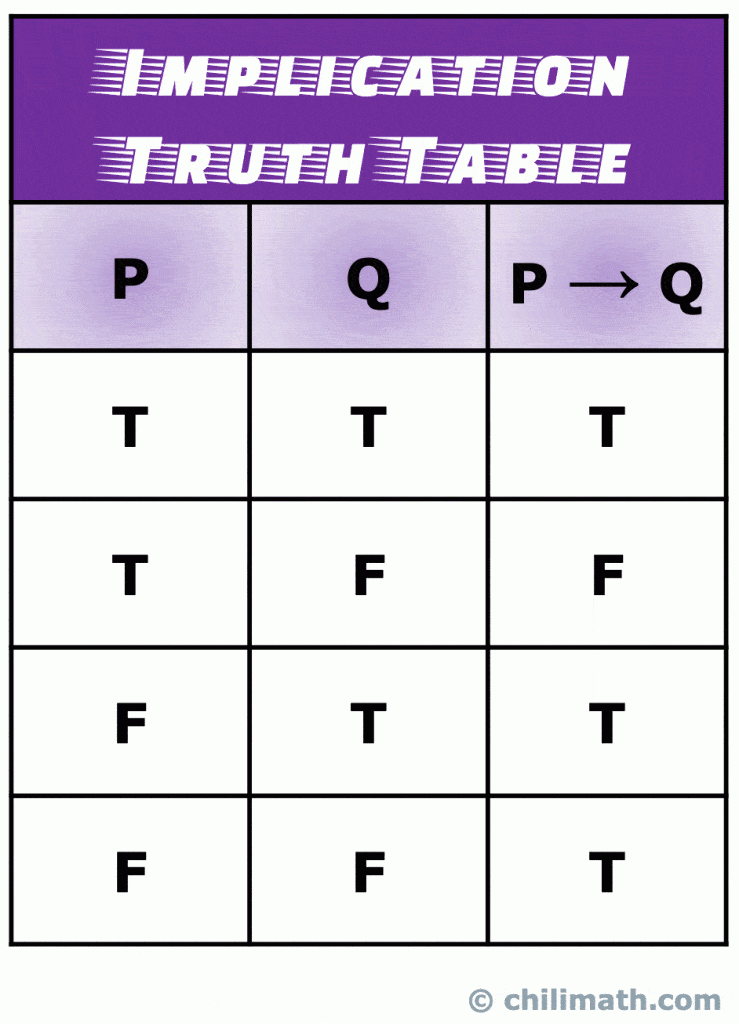
2. According to the task the neural network is binary, so only

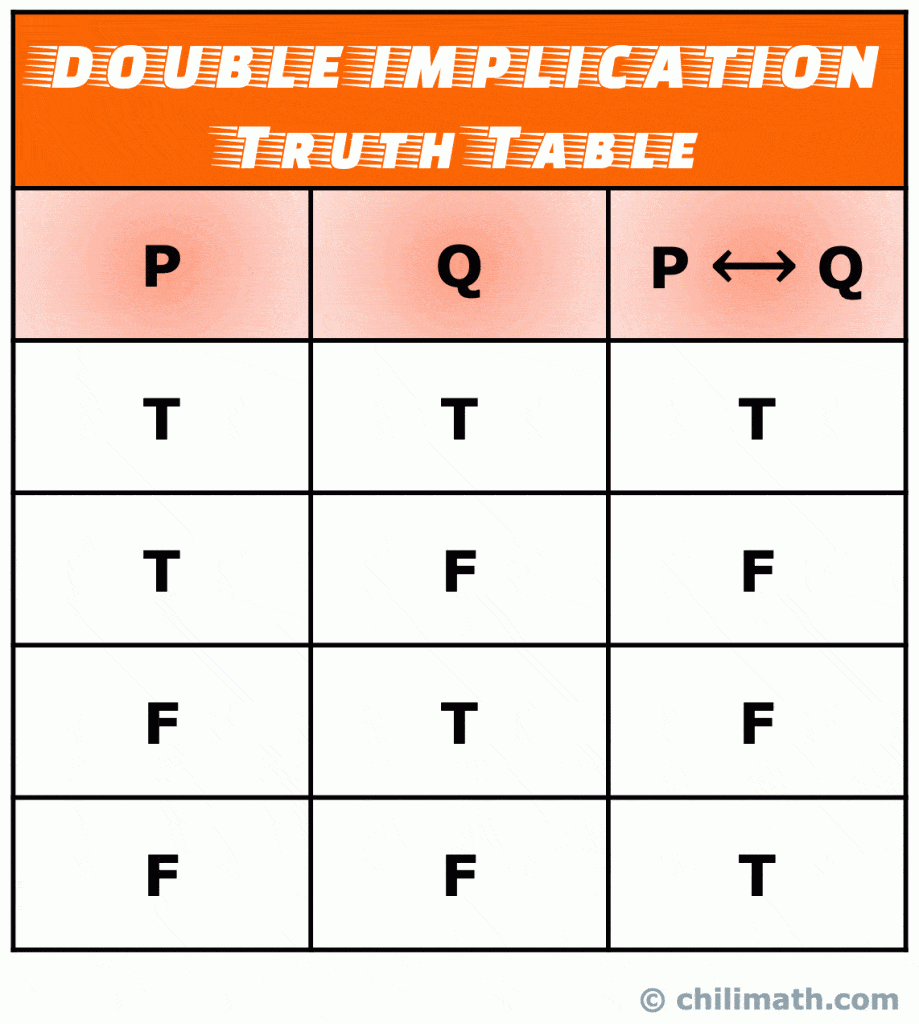
zeros and ones can be fed to its inputs, since there are 2 inputs, so there will be

4 possible combinations of input values (the training sample will consist of 4

vectors).







Training sample is randomly selected. The output of the first neuron corresponds to the **conjunction** operator, and the output of the second neuron corresponds to the

**disjunction** operator.

Truth table is created below:

|  |  |  |  |
| --- | --- | --- | --- |
| X1 | X2 | D1 | D2 |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |

X1 – input of 1st neuron

X2 – input of 2nd neuron

D1 – Output of 1st neuron calculated by conjunction operator

D2 – Output of 2nd neuron calculated by conjunction operator

3. 3rd row of the table will be the learning vector.

4. Following the Δ-rule learning algorithm, 6 steps are performed.

**Step 1:** from the interval [0,1] is set the matrix of weights randomly:

|  |  |  |
| --- | --- | --- |
| Wij(1) | 1 | 2 |
| 1 | 0.6 | 0.7 |
| 2 | 0.5 | 0.3 |

Wij – Weight of connection of j – input with i – neuron

**Step 2**: vector X= {1,0}, vector D= {0,1}.

**Step 3**: Calculation of the output values of the neural network (vector Y):

k = 0.8 → const

**S1**= x1\*w11 + x2\*w21 = 1\*0.6+0\*0.5 = 0.6

(S1 – weighted sum of 1st neuron input value)

**Y1**= 1/1+e^(-S\*k) = 1/1+e^(-0.6 \* 0.8) = 1/1+e^(-0.48) = 1/1+0.62 = 1/1.62 ≈ 0.62

(e^(-0.48) ≈ 0.62)

(Y1 – output value of 1st neural network)

**S2**= x1\*w12 + x2\*w22 = 1\*0.7+0\*0.3 = 0.7

(S2 – weighted sum of 2nd neuron input value)

**Y2**= 1/1+e^(-S\*k) = 1/1+e^(-0.7 \* 0.8) = 1/1+e^(-0.56) = 1/1+0.57 = 1/1.57 ≈ 0.64

(e^(-0.56) ≈ 0.57)

(Y2 – output value of 2nd neural network)

**Step 4**: Calculation for each neuron the difference between the real result and the desired one:

= 0 - 0.62 = -0.62 = 1- 0.64 = 0.36

d1 – desired output value on the 1st neuron

y1 – real value on the 1st neuron

d2 – desired output value on the 2nd neuron

y2 – real value on the 2nd neuron

**Step 5**: Setting η - the coefficient of learning from 0 to 1 and changing the weights:

η =0.6

w11(2) = w11(1) – η \*\*x1 = 0.6 - 0.6\* (-0.62)\*1 = 0.6 + 0.372 = 0.972

w21(2) = w21(1) – η \*\*x2 = 0.5 - 0.6\*(-0.62)\*0 = 0.5

w12(2) = w12(1) – η \*\*x1 = 0.7 - 0.6\*0.36\*1 = 0.7 – 0.216 = 0.484

w22(2) = w22(1) – η \*\*x2 =0.3- 0.6\*0.36\*0 = 0.3

|  |  |  |
| --- | --- | --- |
| wij(2) | 1 | 2 |
| 1 | 0.972 | 0.484 |
| 2 | 0.5 | 0.3 |

**Step 6:** Calculating the ***root mean square error***:

A picture containing watch, clock

Description automatically generated

= (-0.62)2 + (0.36)2 = 0.3844 + 0.1296 = 0.514

H – the number of neurons

One iteration of the learning loop is considered, and the loop is quitted.