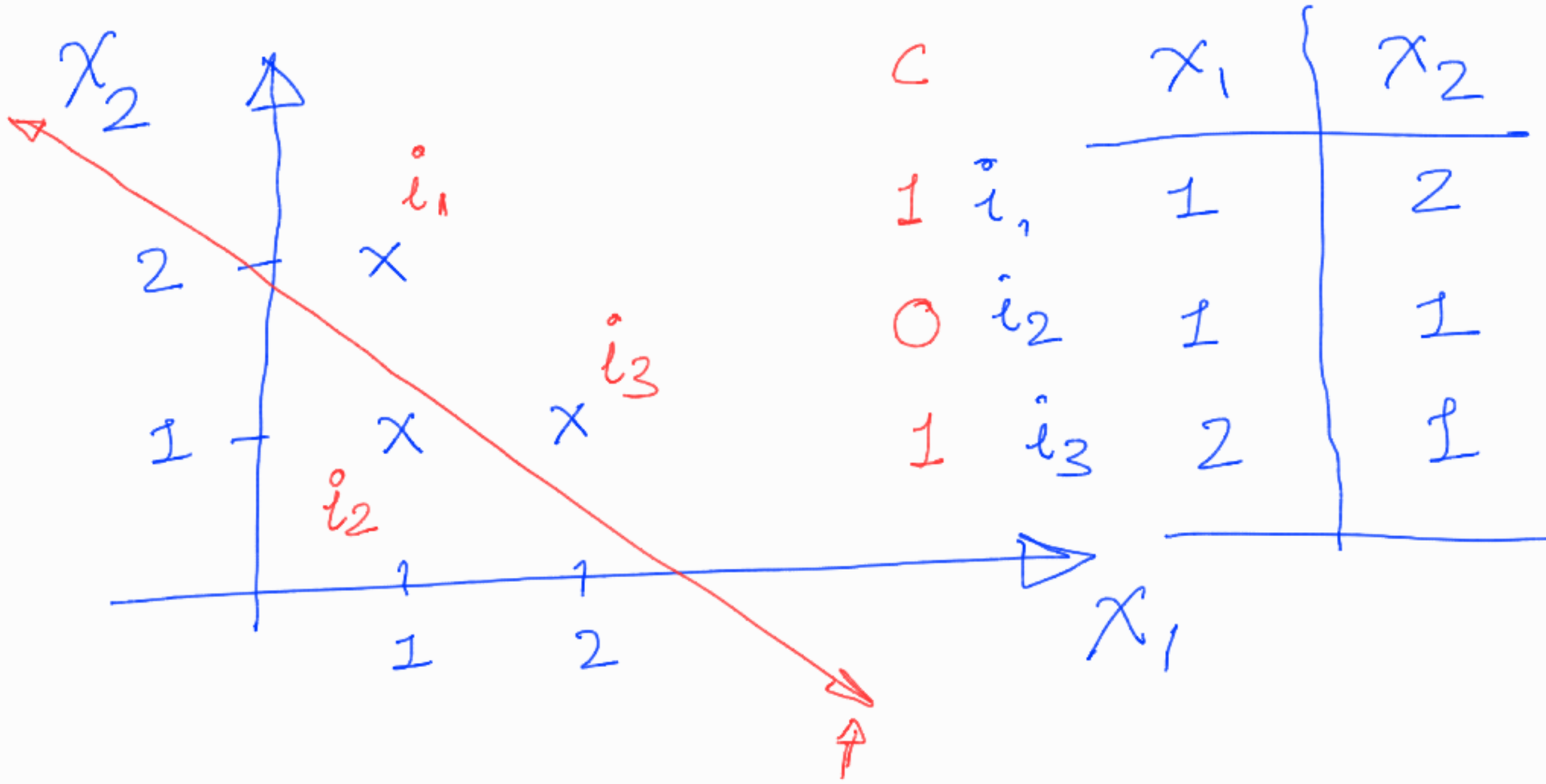
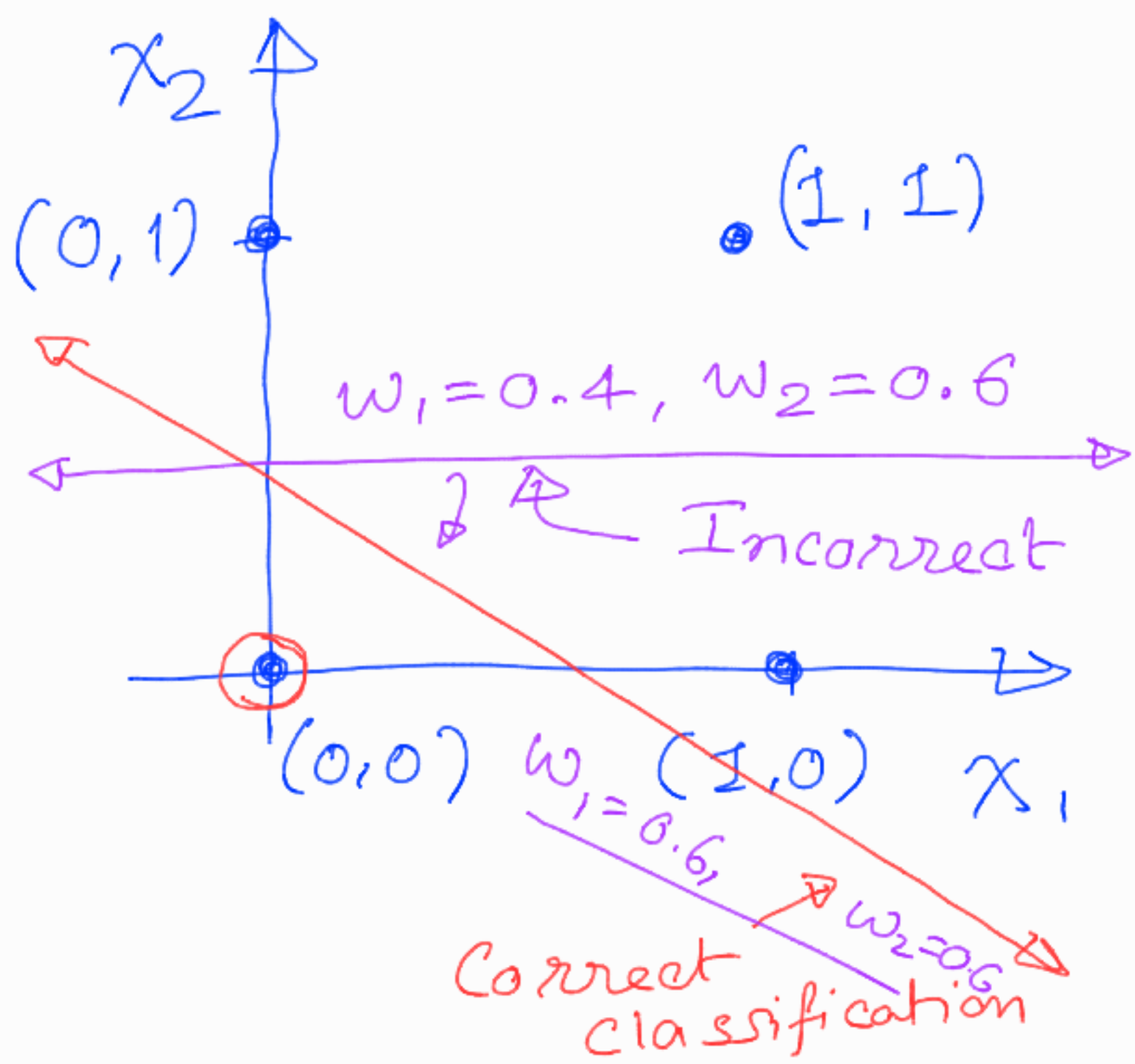


c		x_1	x_2
1	i_1	1	2
0	i_2	1	1
1	i_3	2	1

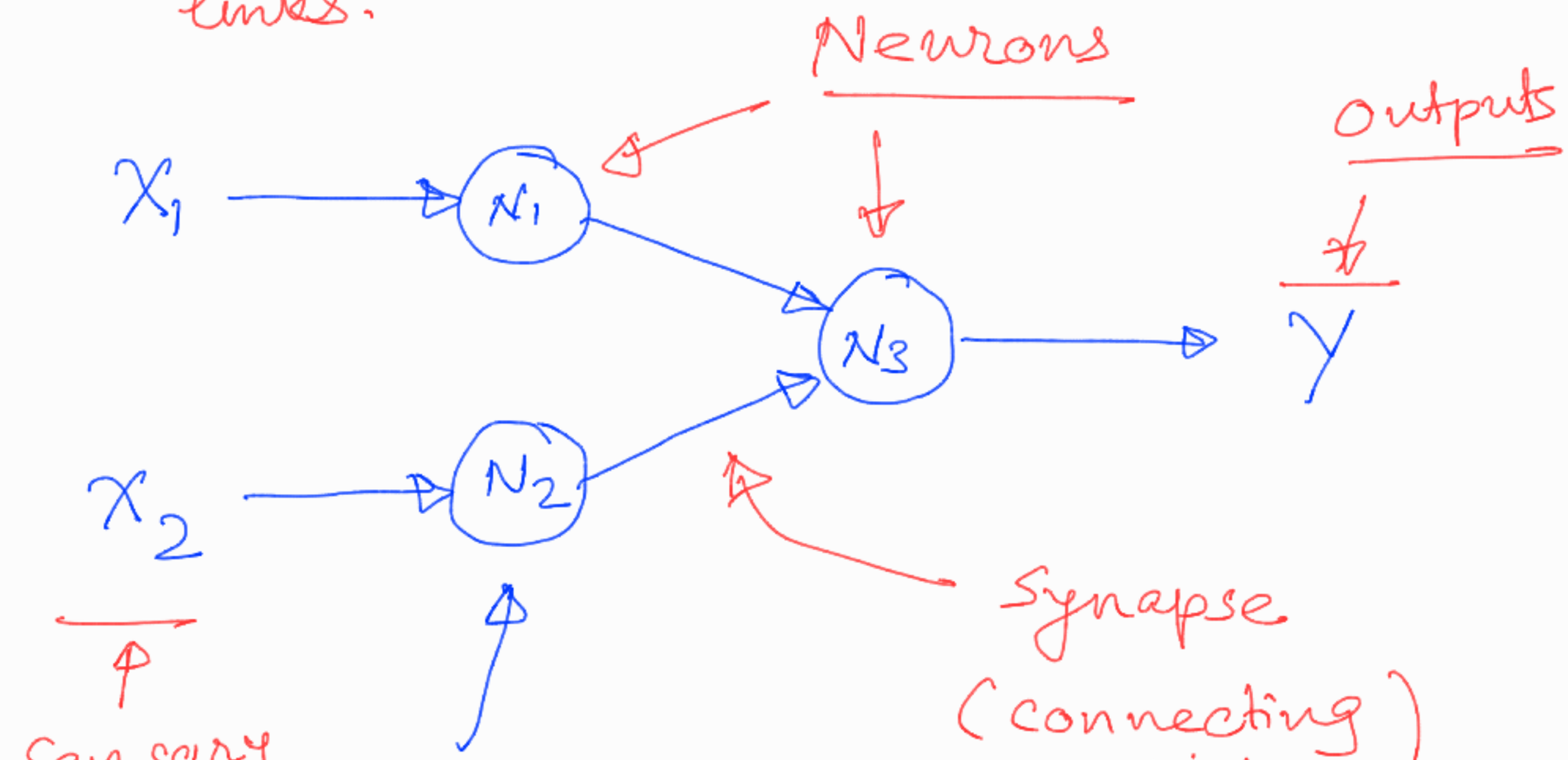
x_1	x_2	c
0	0	0
0	1	1
1	0	1
1	1	1



Perceptron :

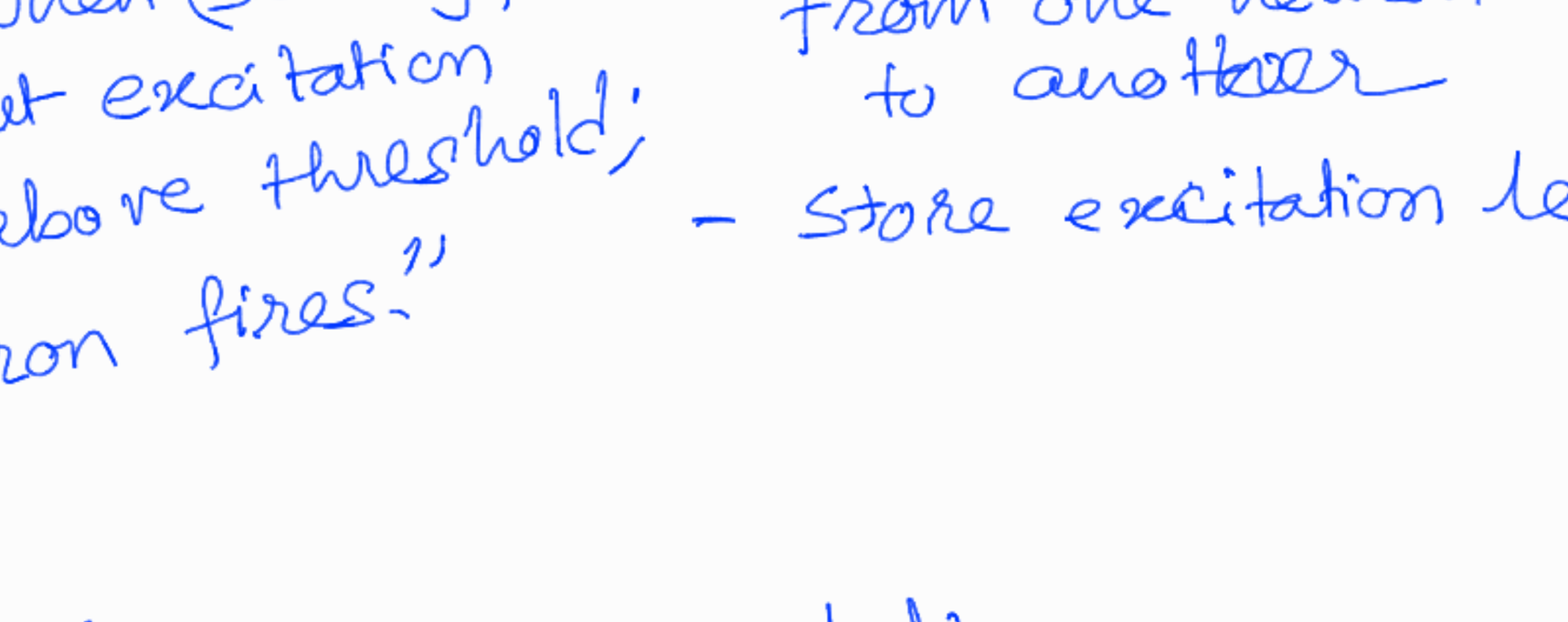
- Inspired by human brain
- In human brain neurons are basic units of computation.

- Neurons are connected via synaptic links.



- when (sensory) input excitation is above threshold; "Neuron fires."

$\Rightarrow$ "Neuron computation : Summation"



x_1, x_2 = inputs signals (0, 1)

w_1, w_2 = strength of synaptic links (Synaptic weight)

y = output signal (+1, −1)
pos. Neg

$y = +1$ if $\sum w_i x_i > b$

$y = -1$ if $\sum w_i x_i \leq b$

Learning

→ Start with randomly initialize weights.

→ Calculate Y, for the input X

→ If Y does not match the the given label; Correct/ update weights.

↘ Error calculation & parameter tuning

(MSE, Sigmoid)

↓

(Mean Squared Error)

/ gradient descent

	w_1	w_2	x_1	x_2	$\sum w_i x_i \geq T$	Y	C
✓	0.4	0.6	0	0	0	0.5 -1	-1 (0)
✓	0.4	0.6	0	1	0.6	0.5 +1	1 (1)
✗ E	0.4	0.6	1	0	0.4	0.5 -1	+1 (1)
	0.4	0.6	1	1	1	0.5 +1	+1

if $E = f(Y - C)$

Update weights

$$w_{i+1} = w_i + \eta \cdot (E)$$

- w_{i+1}: New value of weight
- w_i: Previous value of weight
- η: What portion of error goes in update w_i

$$\boxed{w_1 = 0.6 \quad w_2 = 0.6}$$