Problem:

In an avionic system, there is an internal testing mechanism that determines whether the navigation sensor is flight-ready. One of the measurements taken by this mechanism is the position error value in degrees, along with the heading error data (the main direction of flight), also in degrees.

This device uses a single output and two-input neural cell model to decide whether it is flight-ready. For the GO (pass) state, the input is [1,0] with an output of 1, and for the NOGO (fail) state, the input is [0,1] with an output of -1.

In this model, the synaptic weight corresponding to the first input parameter is $w_1 = 0.5$ and the synaptic weight for the second input parameter is $w_2 = 0.2$. The system's learning rate (μ) is 0.9, and the threshold value (Ω) is 0.4.

Finally, the activation function is defined as follows for the model to classify correctly:

$$f(net) = \begin{cases} -1, & net < 0 \\ 1, & net \geq 0 \end{cases}$$

Error (E) is calculated as the expected output minus the actual output.

Solution:

$$net_1 = w_1 * x_1 + w_2 * x_2 + \Omega = (0.5 * 1) + (0.2 * 0) + 0.4 = 0.9$$

$$f(net_1) = f(0.9) = 1 \ => \ \textit{The expected and actual output values are equal}$$

$$net_2 = w_1 * x_1 + w_2 * x_2 + \Omega = (0.5 * 0) + (0.2 * 1) + 0.4 = 0.6$$

$$f(net_2) = f(0.6) = 1 \ => \ \textit{The expected and actual output values are not equal}$$

$$w_{1yeni} = w_{1eski} + \mu * \mathrm{E} * x_1 = 0.5 + 0.9 * 0 * 1 = 0.5$$

$$w_{2yeni} = w_{2eski} + \mu * \mathrm{E} * x_2 = 0.2 + 0.9 * -2 * 1 = -1.6$$

$$\Omega_{yeni} = \Omega_{eski} + \mu * \mathrm{E} = 0.4 + 0.9 * -2 = -1.4$$

It recalculates the actual outputs corresponding to the inputs with the new values and compares them to the expected output.

$$net_1 = w_1 * x_1 + w_2 * x_2 + \Omega = (0.5 * 1) + (-1.6 * 0) + (-1.4) = -0.9$$

$$f(net_1) = f(-0.9) = -1 \ => \ \textit{The expected and actual output values are not equal}$$

$$net_2 = w_1 * x_1 + w_2 * x_2 + \Omega = (0.5 * 0) + (-1.6 * 1) + 0.4 = -1.2$$

$$f(net_2) = f(-1.2) = -1 \ => \ \textit{The expected and actual output values are equal}$$

$$w_{1yeni} = w_{1eski} + \mu * \mathrm{E} * x_1 = 0.5 + 0.9 * 2 * 1 = 2.3$$

$$w_{2yeni} = w_{2eski} + \mu * \mathrm{E} * x_2 = -1.6 + 0.9 * 0 * 1 = -1.6$$

$$\Omega_{yeni} = \Omega_{eski} + \mu * \mathrm{E} = -1.4 + 0.9 * 2 = 0.4$$

It recalculates the actual outputs corresponding to the inputs with the new values and compares them to the expected output.

$$net_1 = w_1 * x_1 + w_2 * x_2 + \Omega = (2.3 * 1) + (-1.6 * 0) + (0.4) = 2.7$$

$$f(net_1) = f(2.7) = 1 \;\; => \;\; The\ expected\ and\ actual\ output\ values\ are\ equal$$

$$net_2 = w_1 * x_1 + w_2 * x_2 + \Omega = (2.3 * 0) + (-1.6 * 1) + 0.4 = -1.2$$

$$f(net_2) = f(-1.2) = -1 \;\; => \;\; The\ expected\ and\ actual\ output\ values\ are\ equal$$

In this scenario, the calculations are performed to find the new weight values and threshold after training the network. After several iterations, we find the updated weights and threshold values.