Certainly! Let's walk through a simple numerical example of a perceptron on a dataset. Suppose we have a binary classification problem with two input features (x_1 and x_2), and we want to train a perceptron to classify instances into two classes (0 or 1).

Dataset:

x_1	x_2	Expected Output
2	3	0
1	4	1
3	1	0

Perceptron Initialization:

Let's initialize the weights and the threshold for the perceptron:

- $w_1 = 0.5$
- $w_2 = -0.5$

- Threshold = 1

Training:

We'll use the perceptron learning rule to update the weights. The learning rate is set to 0.1.

1. **For the first instance ($x_1 = 2, x_2 = 3$, Expected Output = 0):**

   Calculate the weighted sum:

   $$\text{weighted\_sum} = 0.5 \cdot 2 + (-0.5) \cdot 3 = 1.5 - 1.5 = 0$$

   Apply the step activation function:

   $$\text{output} = \begin{cases} 1, & \text{if weighted\_sum} \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} = 0$$

   The actual output (0) matches the expected output, so no weight updates are needed.

2. **For the second instance ($x_1 = 1, x_2 = 4$, Expected Output = 1):**

   Calculate the weighted sum:

   $$\text{weighted\_sum} = 0.5 \cdot 1 + (-0.5) \cdot 4 = 0.5 - 2 = -1.5$$

$$\text{weighted\_sum} = 0.5 \cdot 1 + (-0.5) \cdot 4 = 0.5 - 2 = -1.5$$

Apply the step activation function:

$$\text{output} = \begin{cases} 1, & \text{if weighted\_sum} \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} = 0$$

The actual output (0) does not match the expected output (1), so update the weights:

$$w_1 = 0.5 + 0.1 \cdot (1 - 0) \cdot 1 = 0.6$$

$$w_2 = -0.5 + 0.1 \cdot (1 - 0) \cdot 4 = -0.1$$

3. **For the third instance ($x_1 = 3, x_2 = 1$, Expected Output = 0):**

   Calculate the weighted sum:

   $$\text{weighted\_sum} = 0.6 \cdot 3 + (-0.1) \cdot 1 = 1.7$$

   Apply the step activation function:

   $$\text{output} = \begin{cases} 1, & \text{if weighted\_sum} \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} = 1$$

   The actual output (1) does not match the exp ed output (0), so update the weights:

   $$w_1 = 0.6 + 0.1 \cdot (0 - 1) \cdot 3 = 0.3$$

   $$w_2 = -0.1 + 0.1 \cdot (0 - 1) \cdot 1 = -0.2$$

Final Weights:

- $w_1 = 0.3$
- $w_2 = -0.2$
- Threshold = 1

This process would be repeated for multiple iterations until the perceptron converges to a set of weights that correctly classify the given dataset. Keep in mind that this is a simplified example, and in practice, more sophisticated algorithms, such as those used in multilayer perceptrons, are often employed for complex tasks.