DEVELOPING A PERCEPTRON MODEL and IMPLEMENTATION of CLASSIFICATION

Generate an artificial neural cell (class) and use it to solve the given classification problem.

Artificial Neural Networks (ANNs), which are used in the field of Machine Learning and form the basis of deep learning, are Artificial Neural Cells (Artificial Neurons). ANNs are used to solve many problems such as classification, clustering and prediction.

The structure of an artificial neural cell and an example computation process is shown in Figure 1. The Perceptron model or neuron in the figure has 4 inputs (x) and 1 output (y).

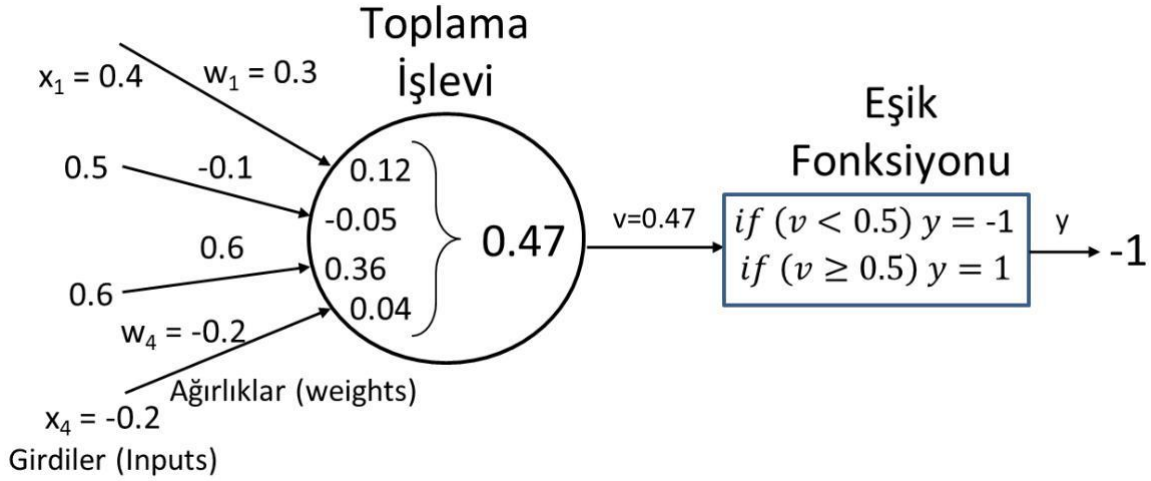

Figure 1: Perceptron Model and Functioning

The Summation Function is performed by taking the sum of the product of the inputs and the weights:

$$v_k = \sum_{i=1}^{n} w_i\, x_i \;= w_1\, x_1 \;+ w_2 x_2 \;+ \;\ldots + w_n x_n \;= [\, w_1\; w_2\; \ldots\; w_n\,] \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}$$

In Supervised Learning, the output values (targets) are given / provided by the system along with the inputs. We have a problem that requires us to classify two positive numbers as 1 and two negative numbers as -1. We exclude numbers whose sum is 0. Let us generate training data to be used in training the network (Table 1):

	Data 1	**Data 2**	**Data 3**	**Data 4**	**Data 5**	**Data 6**	**Data 7**	**Data 8**
x₁	6	2	-3	-1	1	-2	-4	-6
x₂	5	4	-5	-1	1	7	-2	3
Target	1	1	-1	-1	1	1	-1	-1

Table 1: Data Set

Assume that the Learning Rule used to train the network is given as follows:

- If the output produced by the network is different from the expected value, i.e. the target value, increase the weights by $\lambda*(t-o)*x_i$ (t: target, o: output, λ: learning coefficient): So $w_i = w_i + \lambda*(t-o)*x_i$.
- Do not change the weights if the output and target values are the same.

a) Create a Neuron (Neural Cell) class. Choose **appropriate data structures** to hold inputs and weights. **Write methods that perform calculations and necessary operations**. Generate random double values in the range [-1, 1] for the weights.

b) Training: Perform training on the data set formed by dividing all input values in Table 1 by 10. You can take **λ = 0.05**. After **10 epochs** and **100 epochs**, **calculate and print the accuracy of your method** on the data set. An epoch is when all the training data is fed to the system once in sequence and the weights are changed. Accuracy value is the number of correctly classified samples (data) / total number of samples. If 5 of the 8 data you have are classified correctly, the accuracy value is acc = 5/8 = 62.5%.

c) Test: Test the accuracy of your method by creating several different test data other than the training data.