**Experiment No : 4**

**Aim :** Implementing AND Gate and OR Gate using Perceptron

**Theory :** Perceptrons are simple single-layer binary classifiers, which divide the input space with a linear decision boundary. Perceptrons can learn to solve a narrow range of classification problems like Logic Gates. The perceptron is an algorithm for [supervised learning](https://en.wikipedia.org/wiki/Supervised_classification) of [binary classifiers](https://en.wikipedia.org/wiki/Binary_classification). A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class.[[1]](https://en.wikipedia.org/wiki/Perceptron#cite_note-largemargin-1) It is a type of [linear classifier](https://en.wikipedia.org/wiki/Linear_classifier), i.e. a classification algorithm that makes its predictions based on a [linear predictor function](https://en.wikipedia.org/wiki/Linear_predictor_function) combining a set of weights with the [feature vector](https://en.wikipedia.org/wiki/Feature_vector).

**Code 1 :**

**AND gate (AND GATE)**

clear all;clc;

Input=[-1 -1;-1 1;1 -1;1 1];

y=[-1 -1 -1 1];

Initial\_weights=[-1 -1]';

eta=0.7;

weights= Perceptron(Input,y,Initial\_weights,eta)

x\_test = [1 -1];

y\_in = sum(x\_test.\* weights);

if y\_in > 0

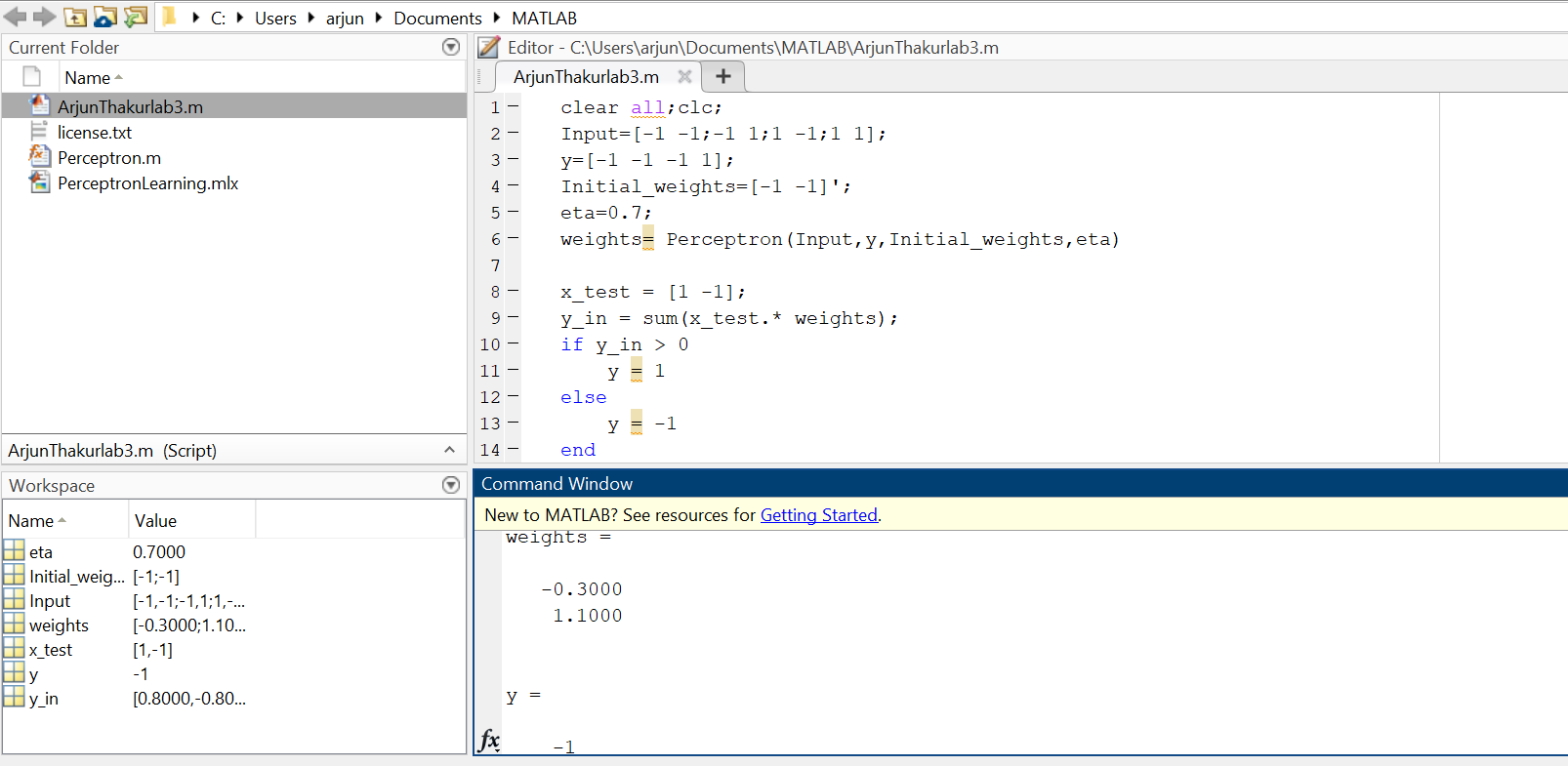
y = 1

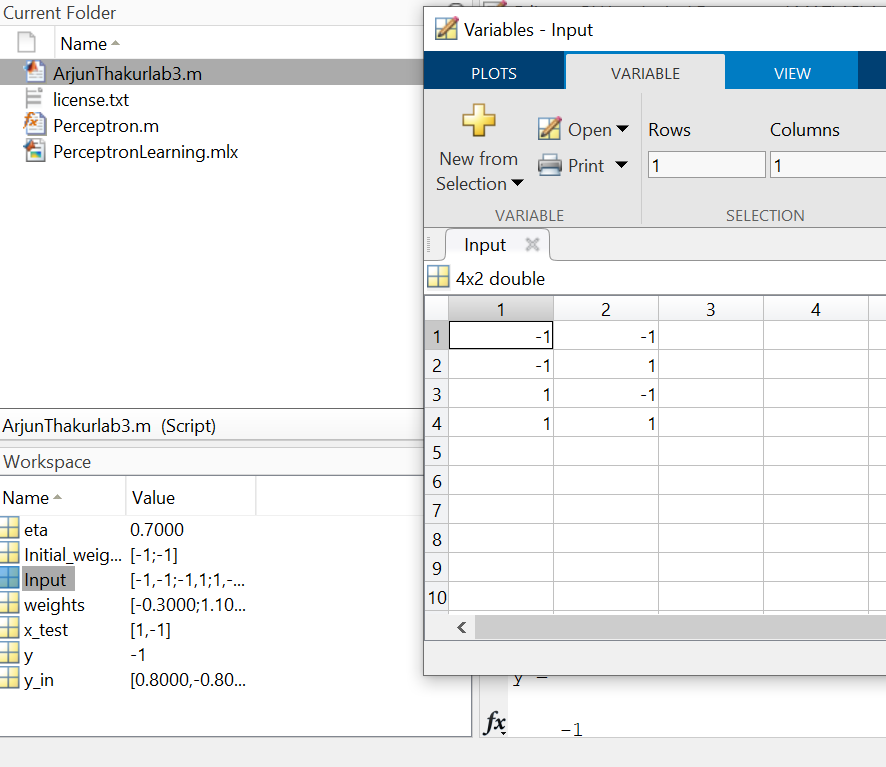
else

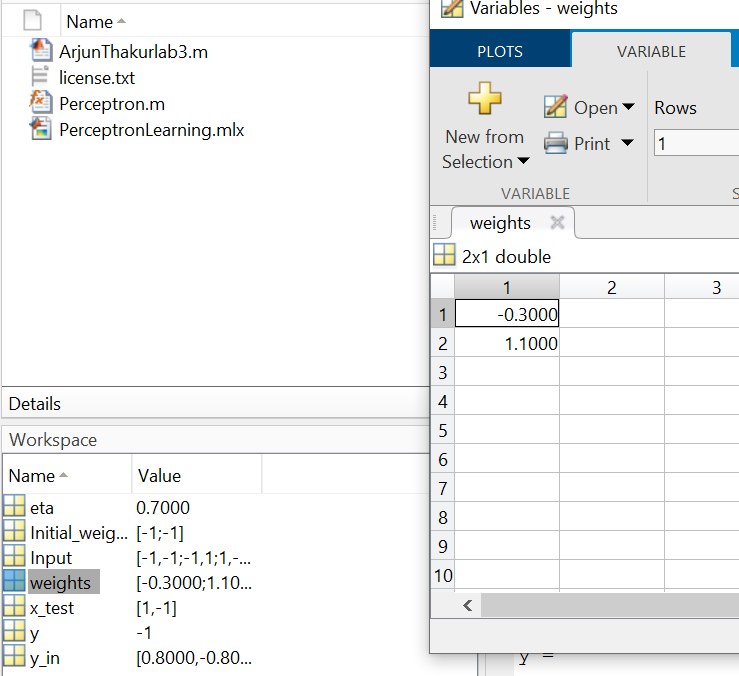
y = -1

end

**Output 1:**







**Code 2 (OR GATE) :**

clear all;clc;

Input=[-1 -1;-1 1;1 -1;1 1];

y=[-1 1 1 1];

Initial\_weights=[-1 -1]';

eta=0.7;

weights= Perceptron(Input,y,Initial\_weights,eta)

x\_test = [1 -1];

y\_in = sum(x\_test.\* weights);

if y\_in > 0

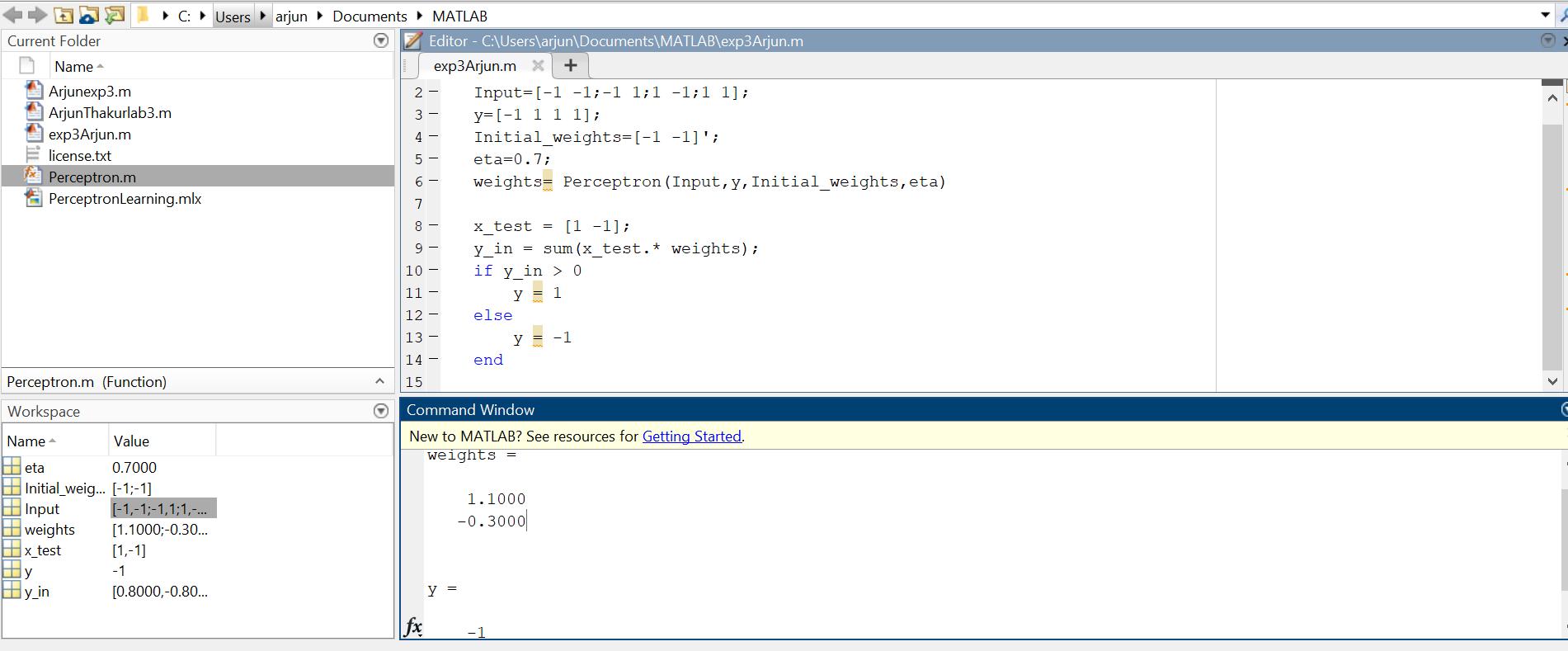
y = 1

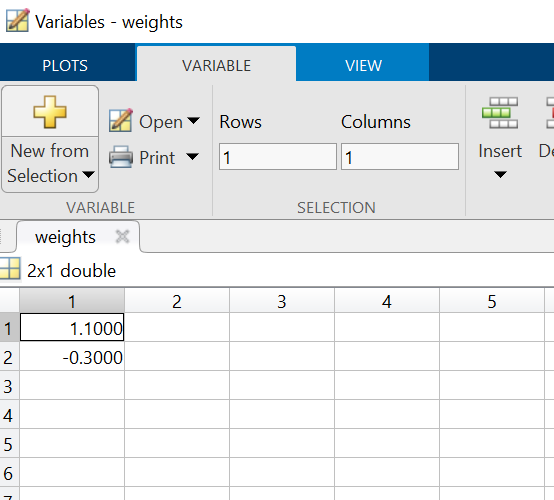
else

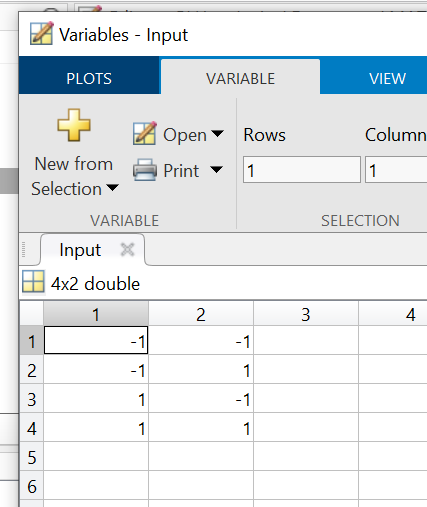
y = -1

end

**Output 2:**







**Conclusion :** Therefore I performed the experiment for AND &OR gate perceptron using the perceptron function from Mathworks and observed the final weights in the workspace. I also saw that threshold, activation function were performed itself with the help of perceptron function.