**Experiment No. 5**

**Aim :** Implementation of EX-OR gate using Backpropagation algorithm

**Theory :**

**Backpropagation :** In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), backpropagation (backprop, BP) is a widely used [algorithm](https://en.wikipedia.org/wiki/Algorithm) in training [feedforward neural networks](https://en.wikipedia.org/wiki/Feedforward_neural_network) for [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning). Generalizations of backpropagation exist for other [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) (ANNs), and for functions generally – a class of algorithms referred to generically as "backpropagation". In [fitting a neural network](https://en.wikipedia.org/wiki/Artificial_neural_network#Learning), backpropagation computes the [gradient](https://en.wikipedia.org/wiki/Gradient) of the [loss function](https://en.wikipedia.org/wiki/Loss_function) with respect to the [weights](https://en.wikipedia.org/wiki/Glossary_of_graph_theory_terms#weight) of the network for a single input–output

**Functions used:**

1. X = net(p,n) returns the first n points from the point set p, which is either a haltonset or sobolset object. X is an n-by-d matrix, where d is the number of dimensions of the points in p. The object p encapsulates properties of a specified quasirandom sequence. Values of the point set are generated whenever you access p using net or parenthesis indexing. Values are not stored within p.

2. BP\_TB: this function allows us to train a multi-layer perceptron based on back propagation of the gradient "for regression".

**Code :**

%Experiment 4 Backpropagation using EX-OR gate

% Arjun Thakur 17070123023

clear all ;

clc;

%% Loading Truth Table for EX-OR gate

x=[0 0;0 1;1 0;1 1]';

y=[0 1 1 0]';

%% Initialize parameters

desired\_error=1e-2;

Learning\_Rate=0.9;

hidden\_layers=[1];

plotting='yes';

%% Training

[net]=BP\_TB(x,y,desired\_error,Learning\_Rate,hidden\_layers,plotting);

%%%%%%%%%%% prediction

%% Prediction

[outputs]=predict(net,x);

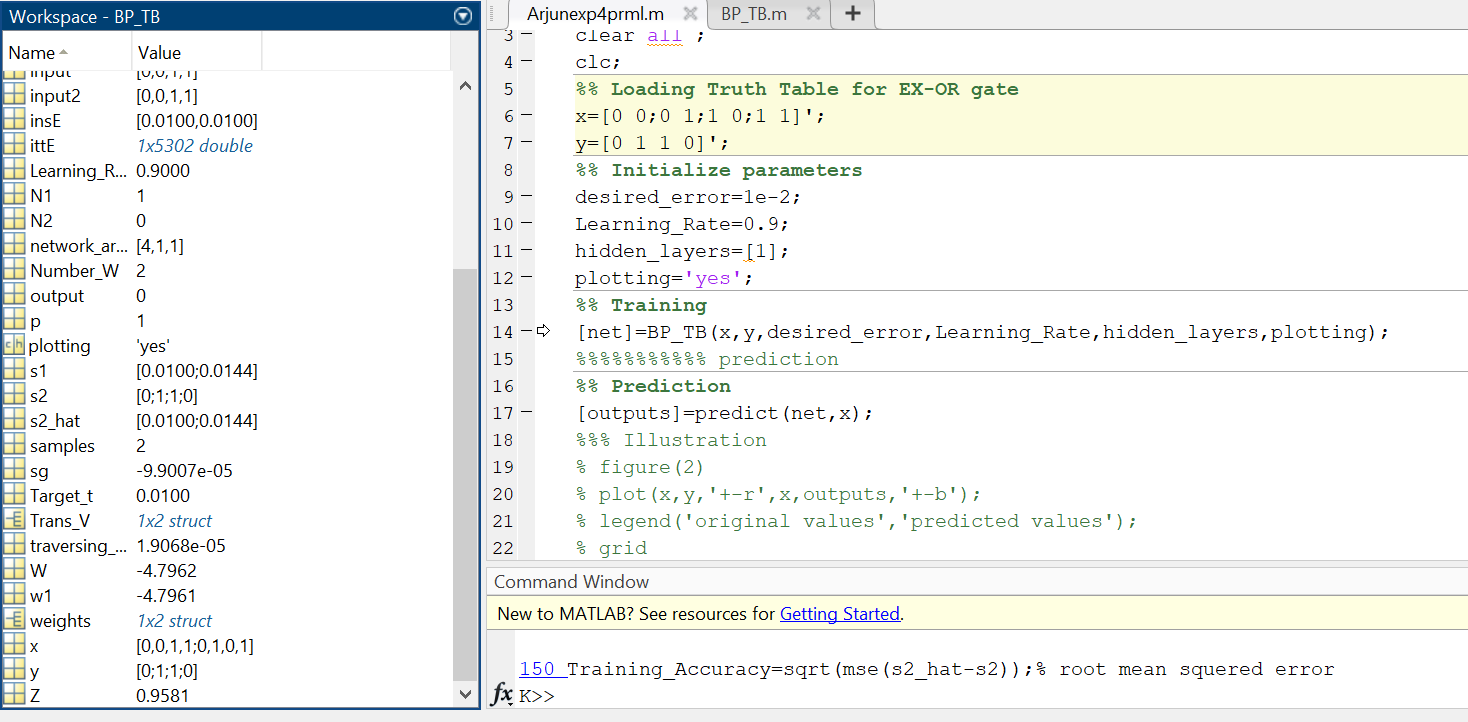
%%% Illustration

% figure(2)

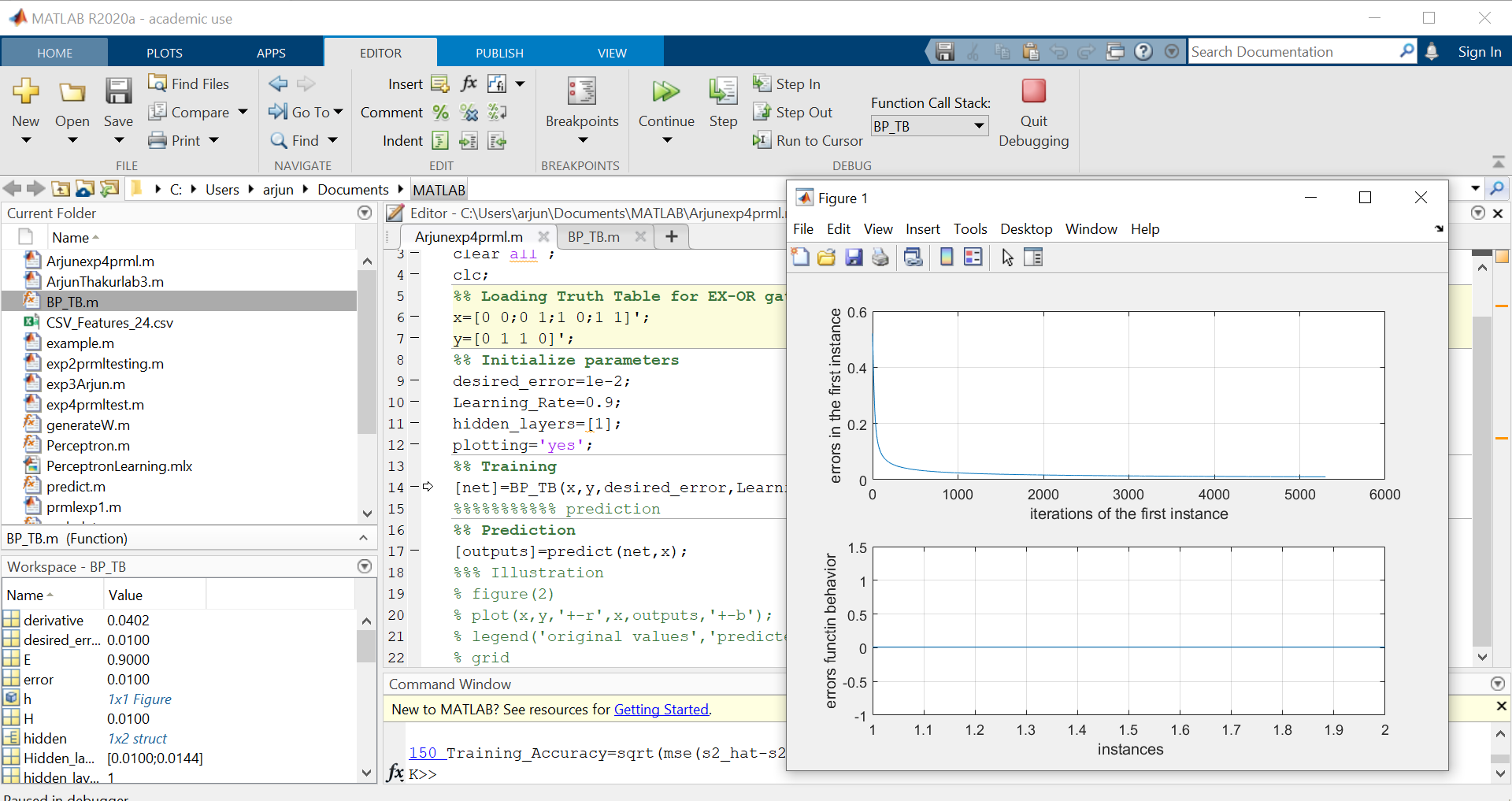
% plot(x,y,'+-r',x,outputs,'+-b');

% legend('original values','predicted values');

% grid



**Output :**



**Conclusion :** Thus I was able to perform Backpropagation for supervised learning using 1 hidden layer for EX-OR Gate and observe the plot for error in first instance and error function behaviour wrt iterations of 1st instance and instances.