SOFT COMPUTING ASSIGNMENT

Paper Report On Matlab Assignment

SUBMITTED TO-

Prof Lavanya K

By-

Naga Malleswar Rao (10BCE0114)

SAKETH P (10BCE0495)

G VINEEL KUMAR (10BCE0131)

KOLLA AJITH BABU(10BCE0139)

SLOT: C2 DATE OF SUBMISSION: 23 APRIL 2013

INTRODUCTION:-

First of all, let us what MATLAB is all about and how it works to make applications in real life and using the field of soft computing. Termed as a MATrix LABaratory, MATLAB is used to alter and program matrices to work and form mathematical applications. It is a programming language developed my MathWorks. It started out as a matrix programming language where linear algebra programming was simple. It can be run both under interactive sessions and as a batch job. [GNU Octave](https://en.wikipedia.org/wiki/GNU_Octave) and [LabVIEW MathScript](https://en.wikipedia.org/wiki/LabVIEW" \o "w:LabVIEW" \t "_blank) are systems for numerical computations with an m-file script language that is mostly compatible with MATLAB. Both alternatives can replace MATLAB in many circumstances. While a good deal of the content of this book will also apply to both Octave and LabVIEW MathScript, it is not guaranteed to work in exactly the same manner. Differences and comparison between MATLAB and [Octave](https://en.wikipedia.org/wiki/GNU_Octave) are presented in [Comparing Octave and MATLAB](https://en.wikibooks.org/wiki/MATLAB_Programming/An_alternative_to_MATLAB:_Octave). Now, since we are going to restrict our study on working on MATLAB, we will not study about the other applications.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of [engineering](https://en.wikipedia.org/wiki/Engineering), [science](https://en.wikipedia.org/wiki/Science), and [economics](https://en.wikipedia.org/wiki/Economics). MATLAB is widely used in academic and research institutions as well as industrial enterprises.

Syntax:-

It follows a basic syntax of the MATLAB language. The MATLAB application is built around the MATLAB language, and most use of MATLAB involves typing MATLAB code into the Command Window (as an interactive mathematical [shell](https://en.wikipedia.org/wiki/Shell_(computing))), or executing text files containing MATLAB code and [functions](https://en.wikipedia.org/wiki/Functional_programming).

A simple assignment is also shown below:

Variables are defined using the assignment operator, =. MATLAB is a [weakly typed](https://en.wikipedia.org/wiki/Type_system#Strong_and_weak_typing) programming language. It is a weakly typed language because types are implicitly converted.[[7]](https://en.wikipedia.org/wiki/MATLAB#cite_note-7) It is a dynamically typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects,[[8]](https://en.wikipedia.org/wiki/MATLAB" \l "cite_note-8) and that their type can change. Values can come from [constants](https://en.wikipedia.org/wiki/Constant_(computer_science)), from computation involving values of other variables, or from the output of a function. For example:

>> x = 17

x =17

>> x = 'hat'

x =hat

>> y = x + 0

y = 104 97 116

>> x = [3\*4, pi/2]

x = 12.0000 1.5708

>> y = 3\*sin(x)

y = -1.6097 3.0000

These values will be assigned to the matrices directly.

Let us stop the introduction of MATLAB here and proceed into our applications that are developed in the course of the project. First, to execute neural networks in MATLAB we have to run an alternate patch called Neuro Solutions for MATLAB which makes us compile the MATLAB programs using neural networks.

1. Printing alphabets comfortably using the software in a MATLAB generator for easy usage in soft computing:-

This method is achieved by using a soft computing function called as [X,T] = prprob;

To execute this a separate editor must be opened for the neural solutions and can be saved and in the normal page that you have run the code-

plotchar(X(:,1)) for A

plotchar(X(:,2)) for B

plotchar(X(:,3)) for C

and so on till 26 alphabets.

This gives us a sketch using a matrix format and makes us use MATLAB correctly for specifying input data in the set.

Logic:-

The script prprob defines a matrix X with 26 columns, one for each letter of alphabet. Each column has 35 values which can either be 1 or 0. Each column of 35 values defines a 5X7 bitmap of letters. This bitmap shows completely how we can toggle with alphabets as the matrix T is a 26X26 identity matrix which maps the 26 inputs to the 26 classes.

[X,T] = prprob;

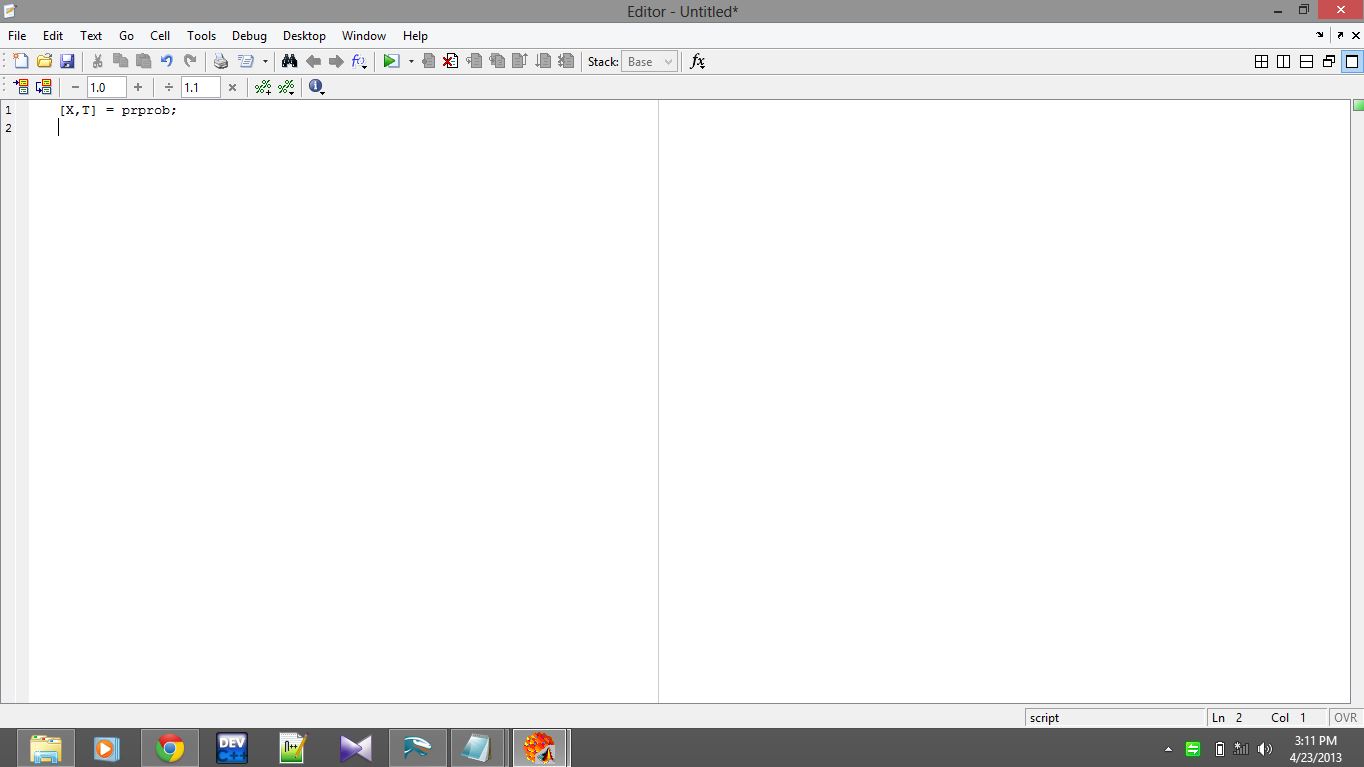
Plotchar(X(:,1))

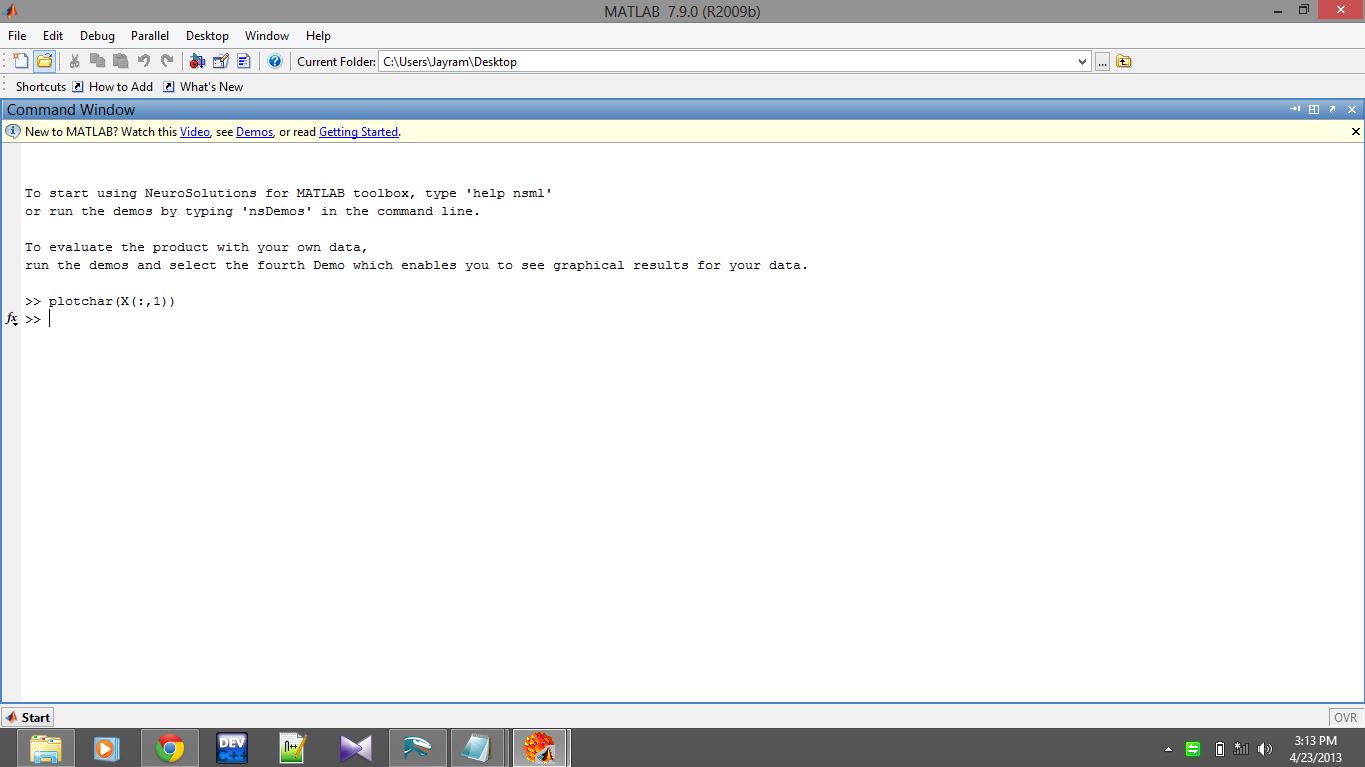
Let us check examples for letters A and B.

To compile-

An alternate editor must be opened parallely and must be run for the patch to be activated.

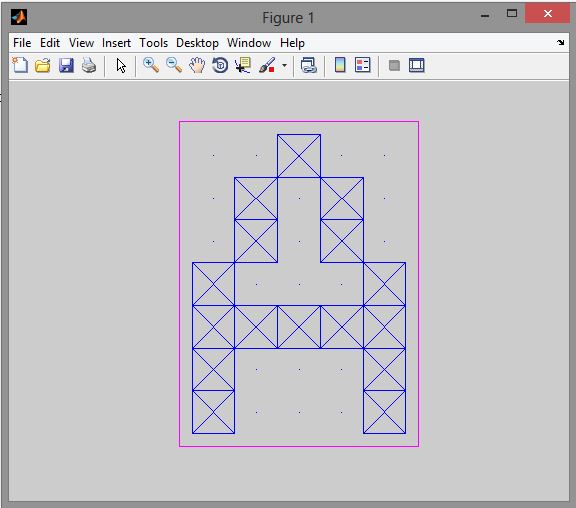
It looks like the following test case.



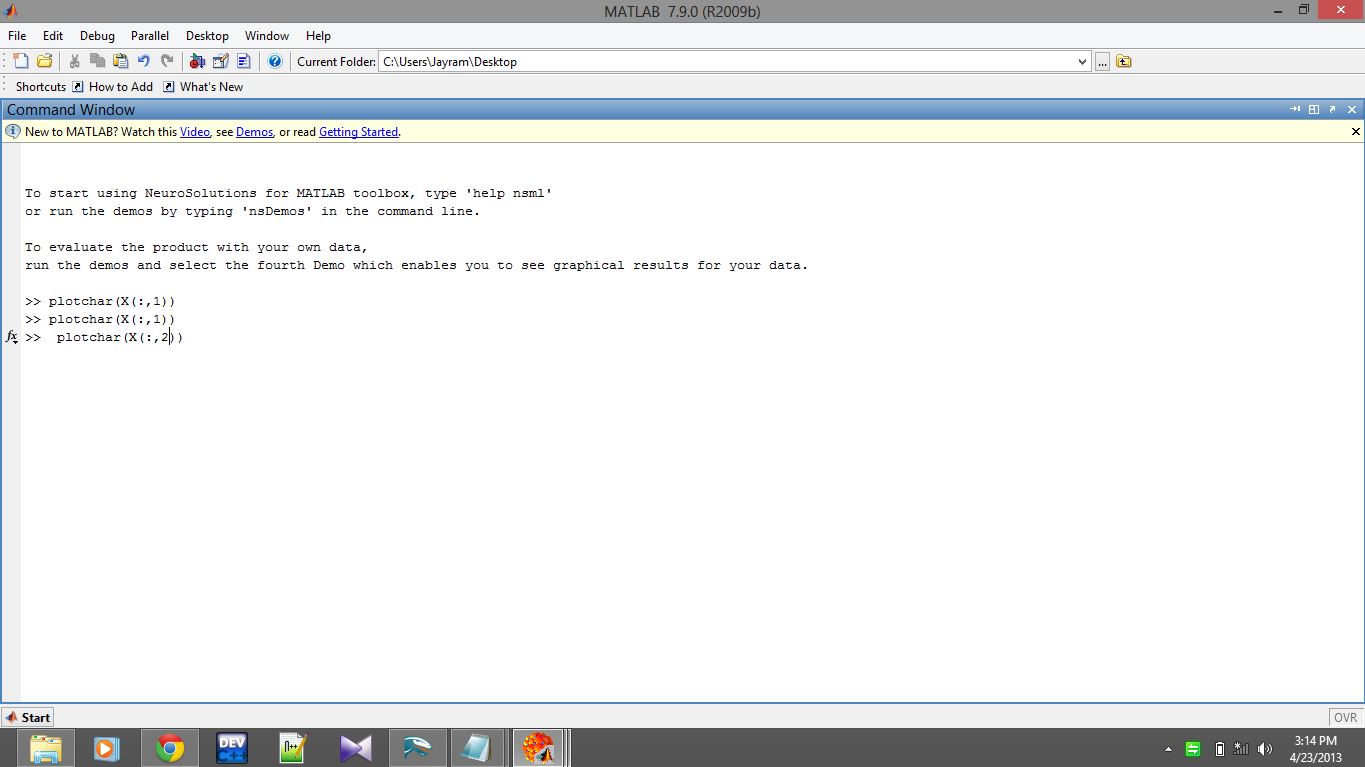
After running this, we have come back to the original file and run the following codes for the required alphabets.

Let’s check for A-This is how it must be run(as shown above)

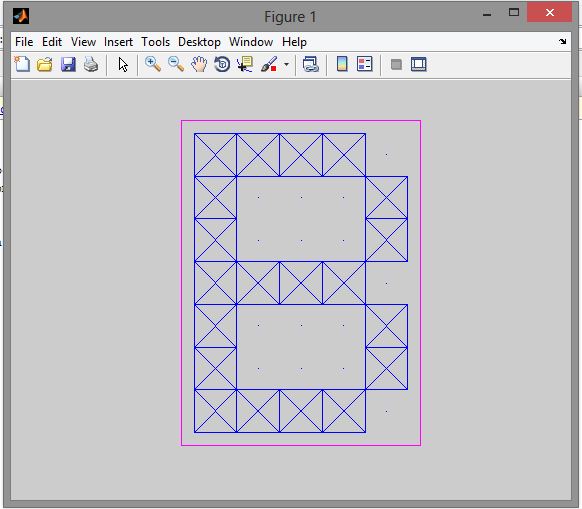
The following output is obtained-



Similarly, let’s try out for B after A and compile it-



The output is obtained as-



Similarly, this can be used to run any kind of information comprehensively.

1. The second type of training that is run is strictly soft computing oriented. Here, we run the perceptron training algorithm using a function called as

net=newp(I,O);

where net is any variable and that is a function that takes any input and output and trains it.

Now, we have the choice to take any input of the format

I=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1;

0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1;

0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0;

0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1;

1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0;]

Where the columns are the number of training variables and rows are the number of times it is trained.

Output can also be specified as-

O=[0 0 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 1 0 1 1 1 0 0 0 0;

0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1;

0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0;

1 0 0 0 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 1 0 1 0 1 1 0;

0 0 1 0 1 1 0 0 1 0 1 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1;

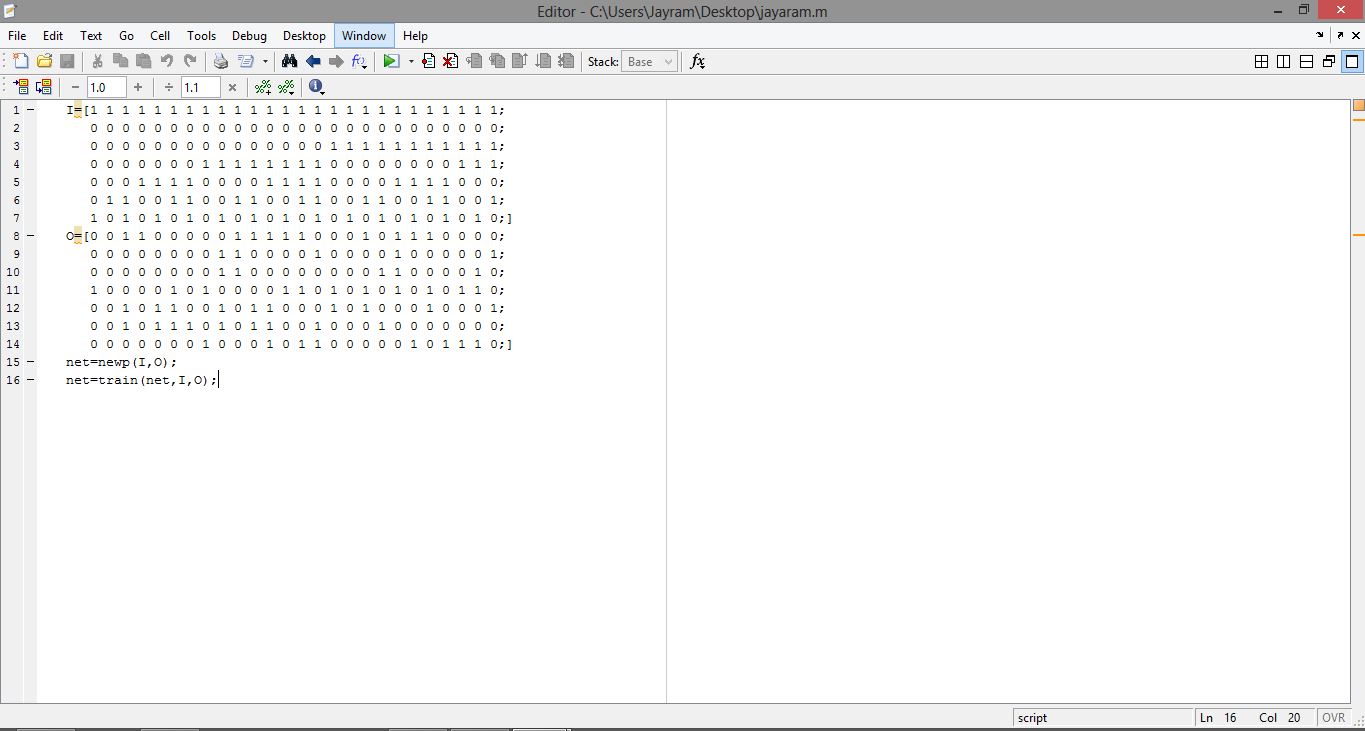
0 0 1 0 1 1 1 0 1 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0;

0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 1 0 1 1 1 0;]

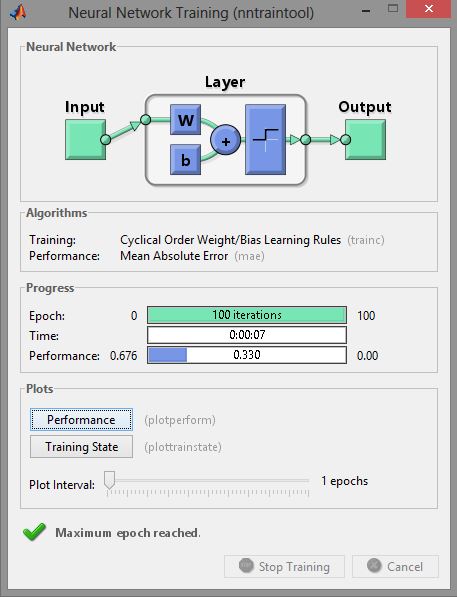
Now after selecting perceptron training using net=newp(); we need to add another function to train input net=train(net(I,O));

If we don’t specify the newp function, it trains in a random algorithm and expected output is not obtained in terms of Performance.

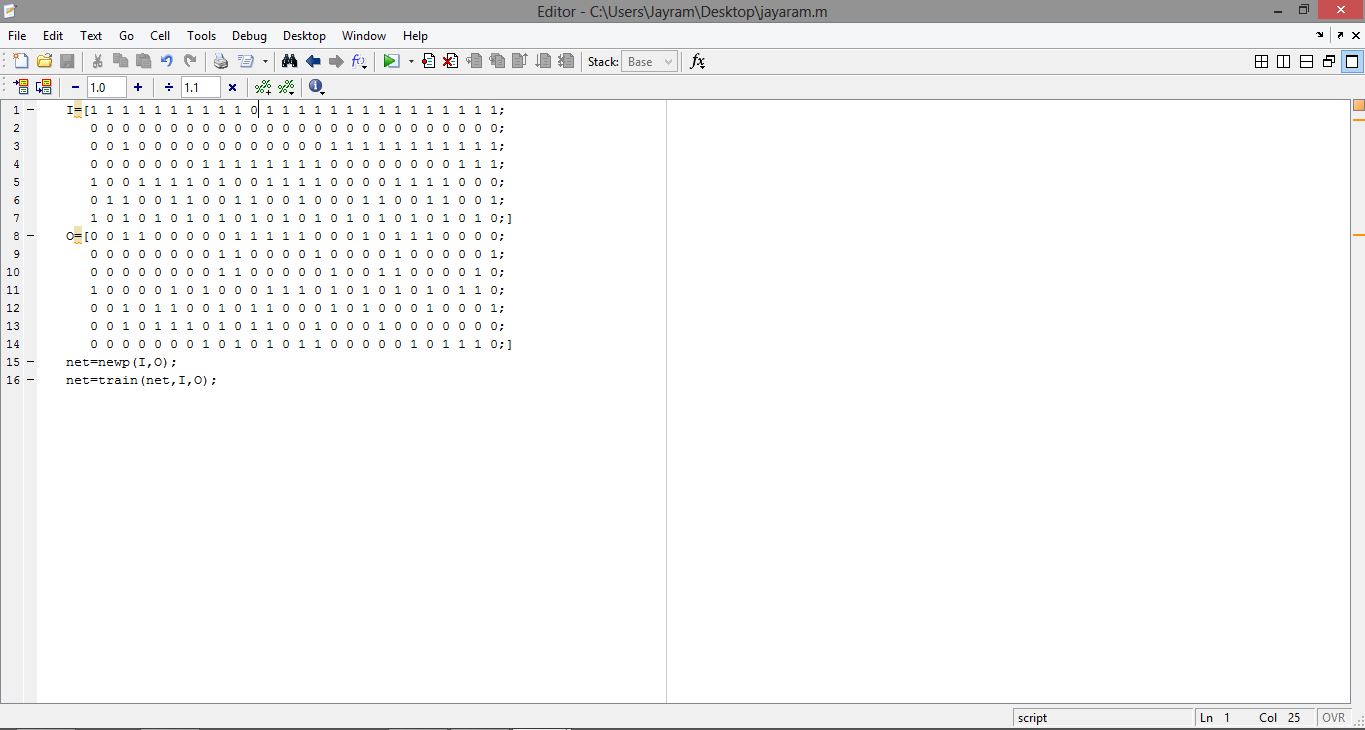
Let us see test cases for this.

Training for above input-

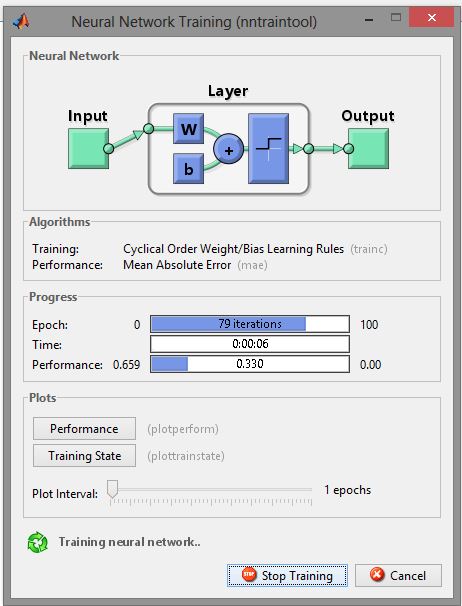
Output is obtained as :



Performance=.676 which is manually true.

Let’s try out for another input-

While training,



While training it shows .330 in the middle and 79 iterations are gone. Next, we will check after it is entirely trained. It comes to .659

