Name: Tania Rajabally

Roll No: 43

UID: 2017130047

Batch: C

Course Code: CEL71 (AI and Soft Computing Lab)

Date:26/08/2020

**Experiment No 3**

**To implement Basic Neural Network learning rules**

**Aim:** To implement Basic Neural Network learning rules using Joone to distinguish between apple and orange.

**Theory:**

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria.

A neural network works similarly to the human brain’s neural network. A “neuron” in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.

A neural network contains layers of interconnected nodes. Each node is a perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

In a multi-layered perceptron (MLP), perceptrons are arranged in interconnected layers. The input layer collects input patterns. The output layer has classifications or output signals to which input patterns may map. For instance, the patterns may comprise a list of quantities for technical indicators about a security; potential outputs could be “buy,” “hold” or “sell.”

Hidden layers fine-tune the input weightings until the neural network’s margin of error is minimal. It is hypothesized that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs. This describes feature extraction, which accomplishes a utility similar to statistical techniques such as principal component analysis.

Joone is a FREE Neural Network framework to create, train and test artificial neural networks. The aim is to create a powerful environment both for enthusiastic and professional users, based on the newest Java technologies.

Joone is composed of a central engine that is the fulcrum of all applications that are developed with Joone. Joone's neural networks can be built on a local machine, be trained on a distributed environment and run on whatever device.

Everyone can write new modules to implement new algorithms or new architectures starting from the simple components distributed with the core engine. The main idea is to create the basis to promote a zillion of AI applications that revolve around the core framework.

**Procedure:**

We take the file as an input and put the path of the input file. Here it only reads the first two parameters from the file.

We create three sigmoid layers.

We then have a teacher that trains the network.

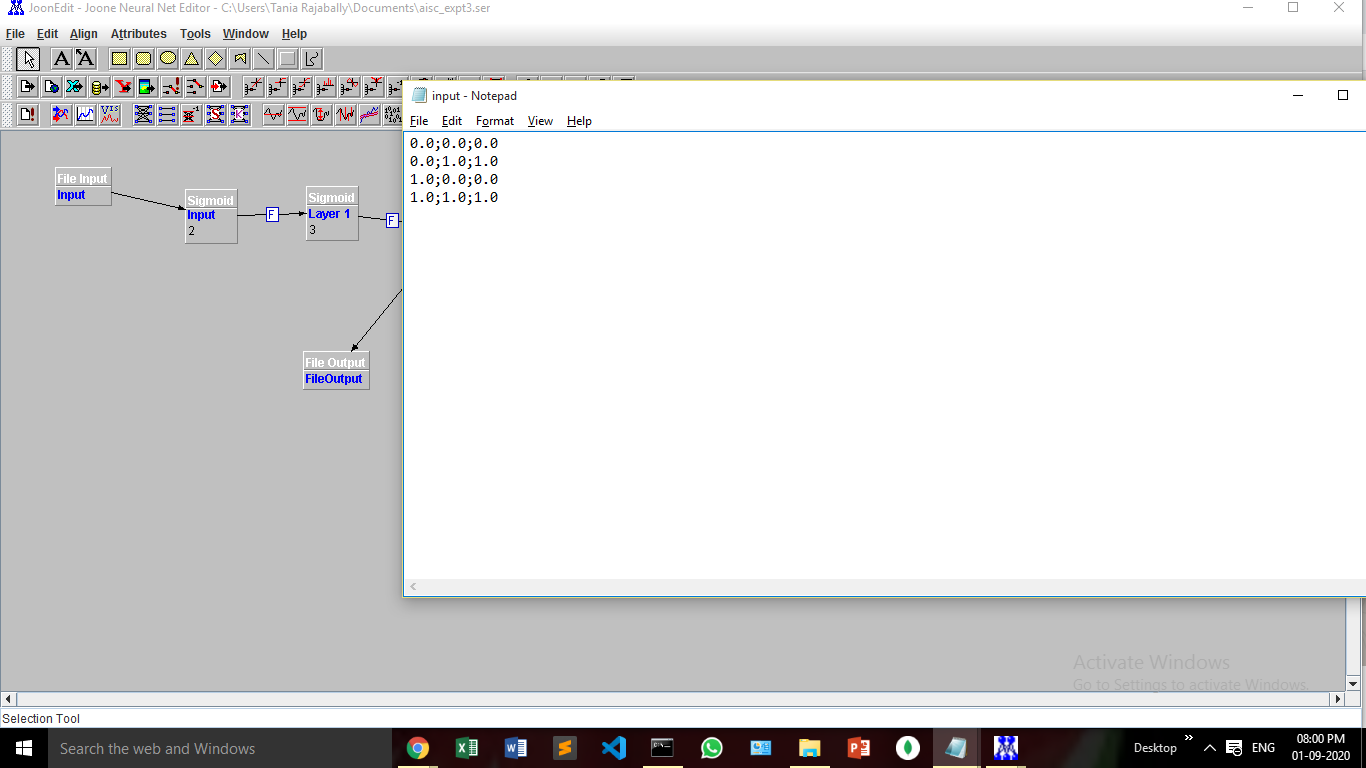
We store the output in a file name specified.

We take the third parameter of the file as an input.

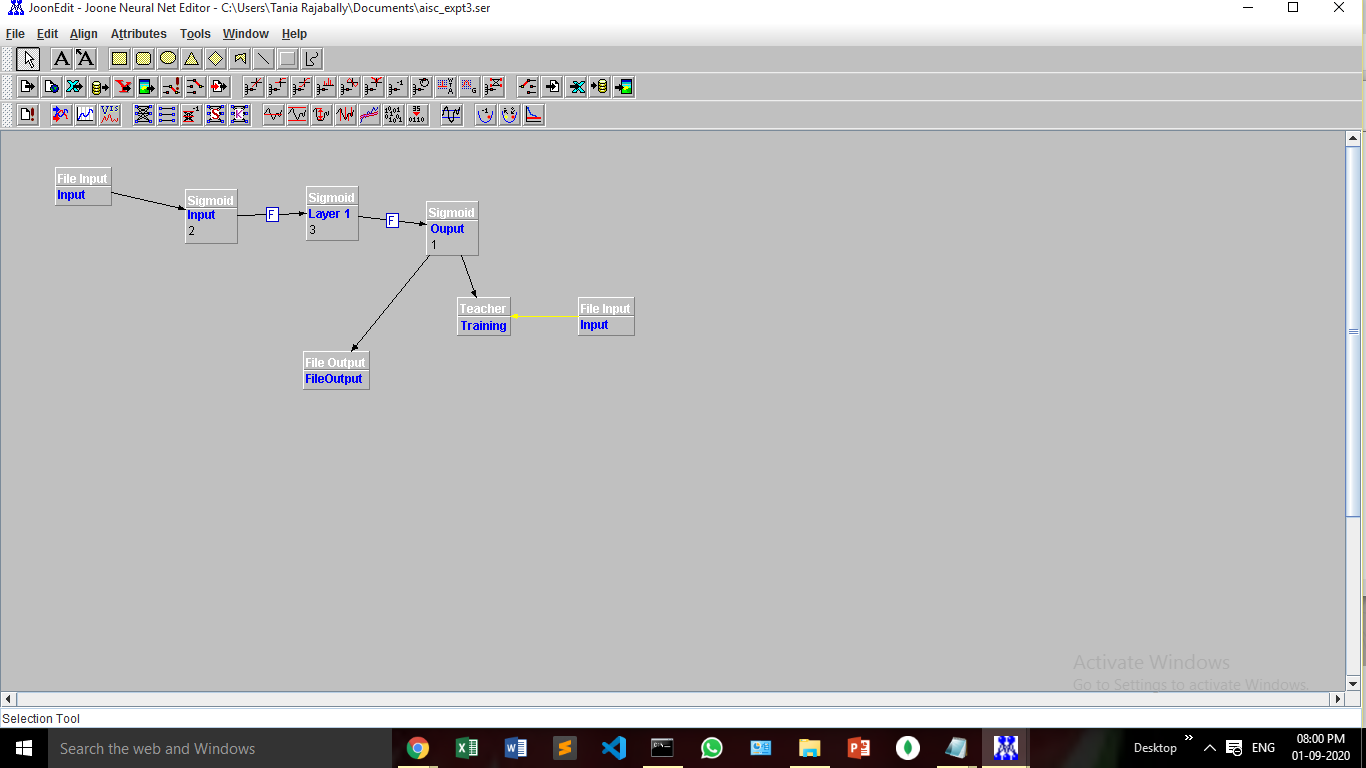
This network is supervised.

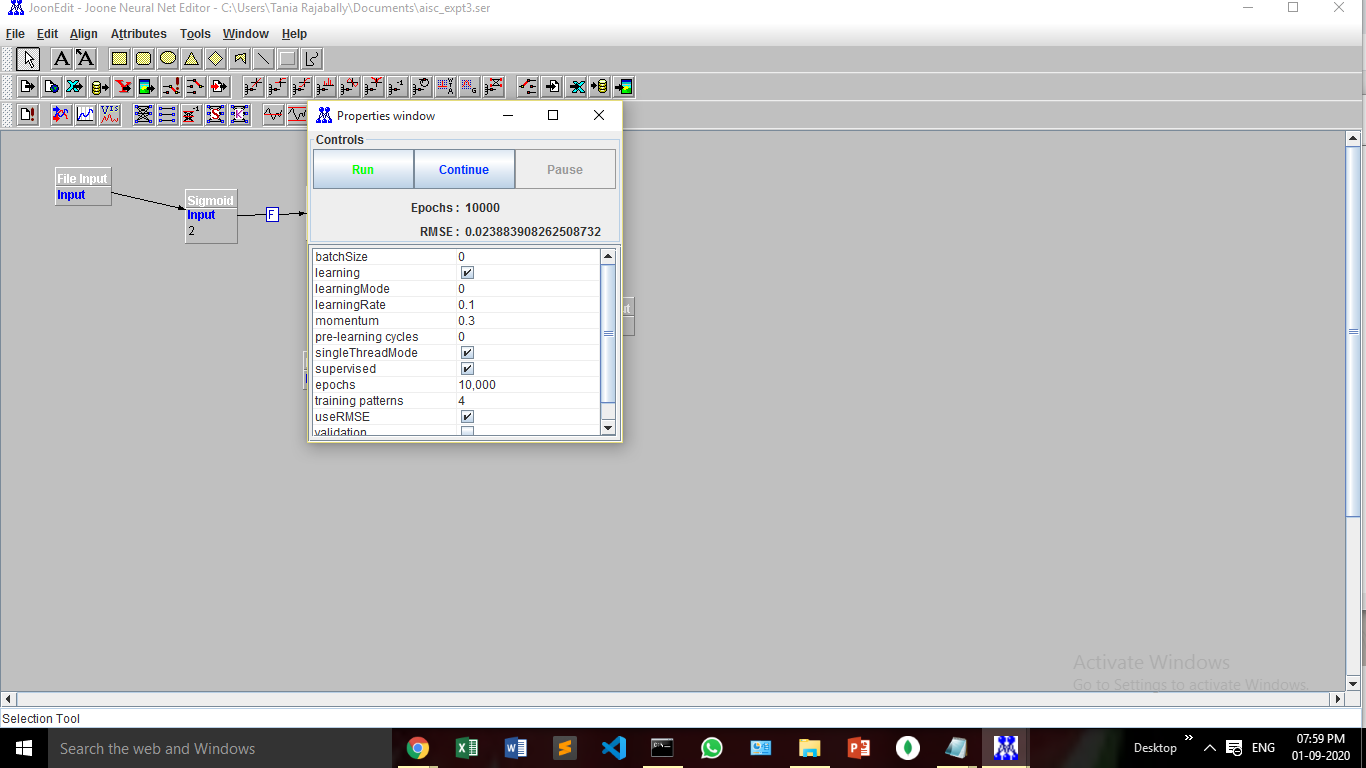
We have to go to tools -> control panel. We select learning and supervised. The number of epochs can be decided. Here the training parameters are 4. We then run this and the output is stored in the file specified.

**Input File:**

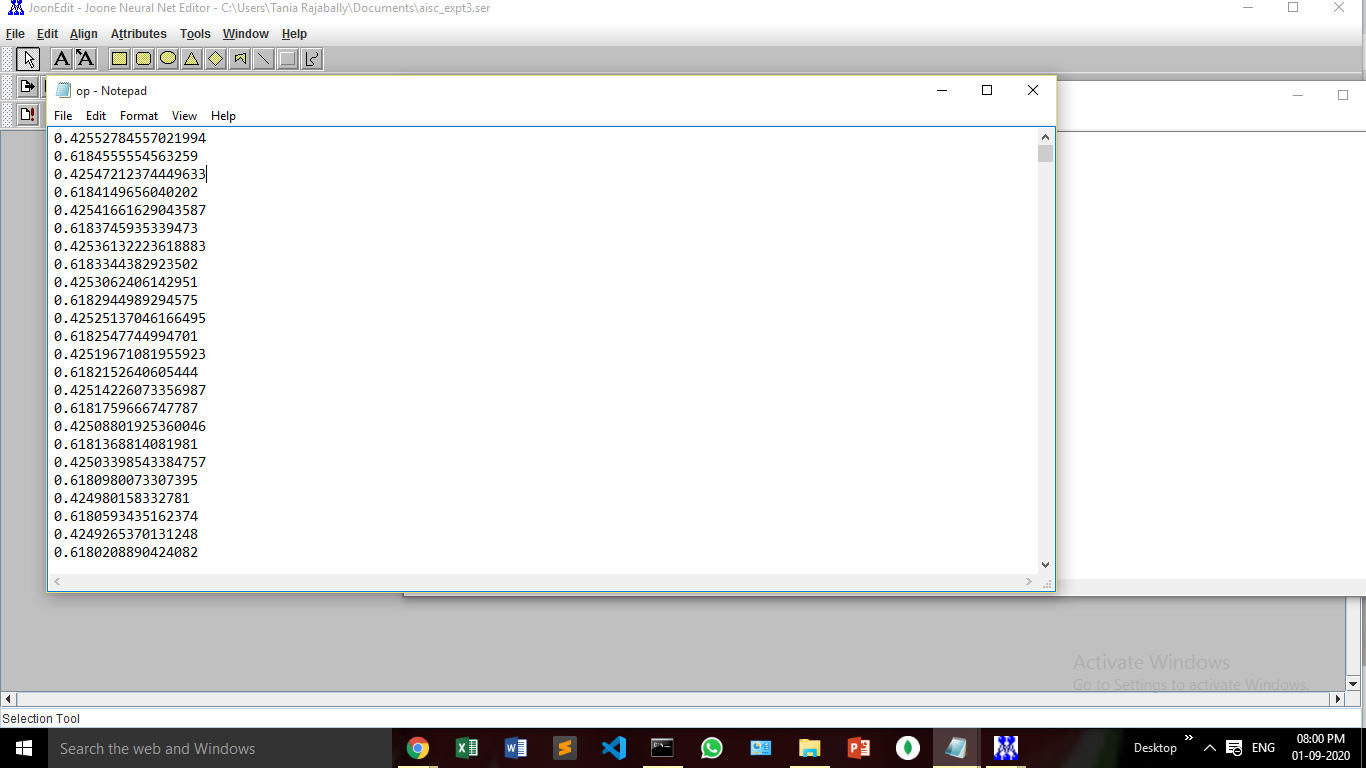
****

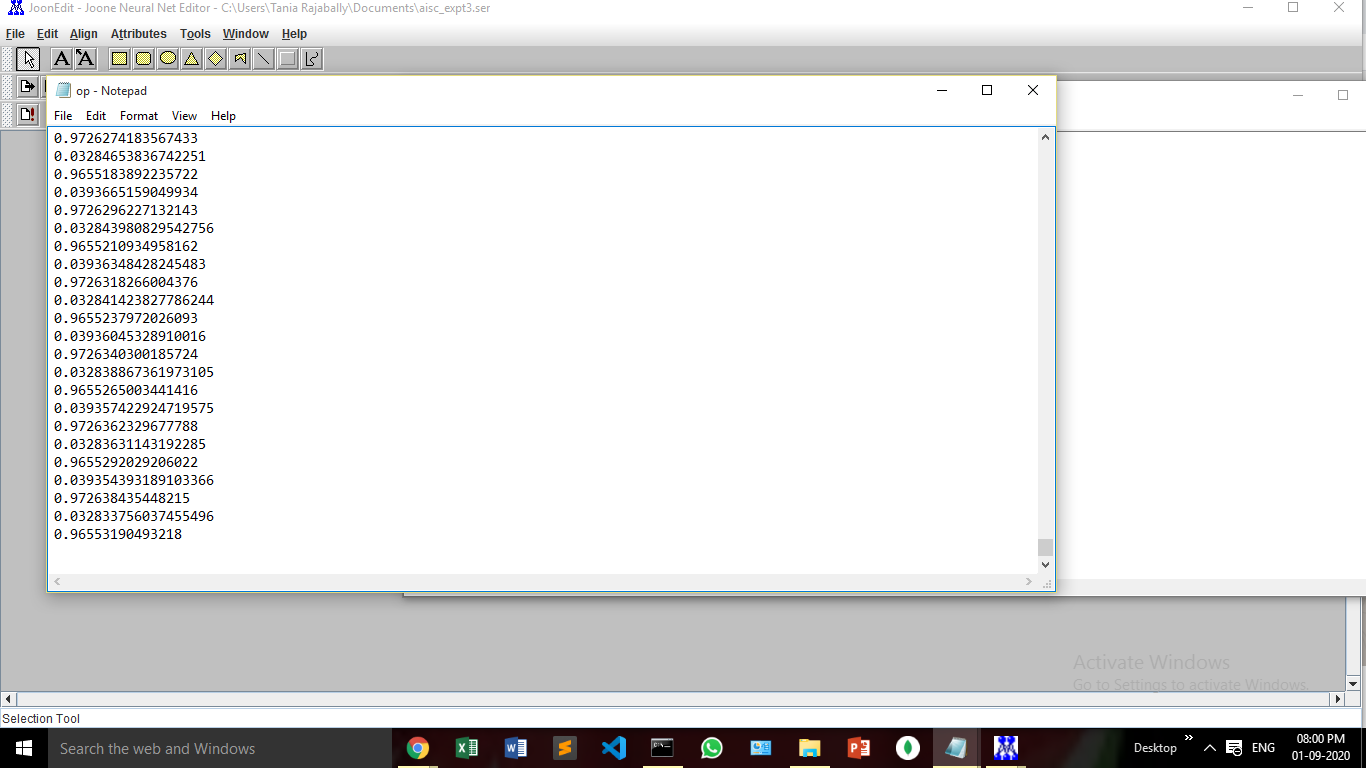
**Joone:**

****

****

**Output File:**





**Conclusion:**

In this experiment, we have distinguished between apples and oranges by implementing a basic neural network using joone. We trained upto 10000 epochs and saw that the network learned.