Self-Organizing Maps

Shahmeer

Essam

Talha

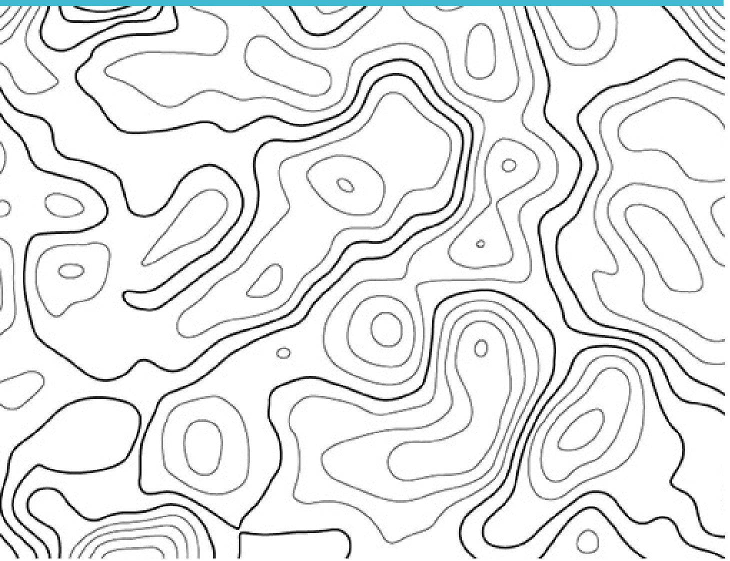
What is an **Artificial Neural Network?**

- Inspired by Biological Neural Networks (ones in animals)
- Such systems "learn" tasks by considering examples, generally without task-specific programming
- Basically, there are 3 different layers in a neural network :
  - **Input Layer** - All the inputs are fed in the model through this layer
  - **Hidden Layers** - There can be more than one hidden layers which are used for processing the inputs received from the input layers
  - **Output Layer** - The data after processing is made available at the output layer
- They are excellent tools for finding patterns far too complex or numerous for a human programmer to extract

What is an **Artificial Neuron?**

- Basic building block of a Neural Network
- In nature, neurons have a number of dendrites (**The Input**), a cell nucleus (**The Processor**) and an axon (**The Output**)
- Intended to replicate the way that we humans learn
- When the neuron activates, it accumulates all its incoming inputs
- The important thing about neurons is that they can **learn**

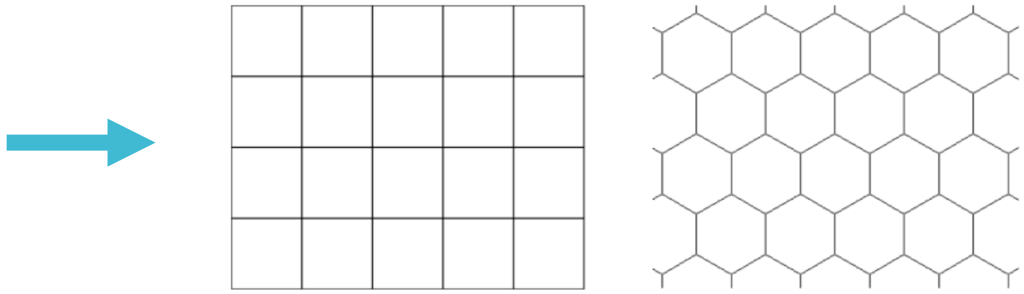
What is a **Topographical Map?**



- In humans, data collected from different senses if mapped onto different regions/neighbourhoods of a brain
- In TM's, each piece of incoming information is kept in its proper context/neighbourhood (at each stage)
- Neurons dealing with closely related pieces of information are kept close together
- Therefore, the location of an output neuron in a topographic map corresponds to a particular domain or feature drawn from the input space

Self Organizing Maps

- Basically, the goal of an SOM is to transform an incoming signal pattern of arbitrary dimension into a one or two dimensional discrete map (usually a **Rectangle** or **Hexagon**)
- This transformation should be performed adaptively in a **topologically** ordered fashion
- Done using **Unsupervised Learning**, in which the networks learn to form their own classifications of the training data without external help



Self Organizing Maps

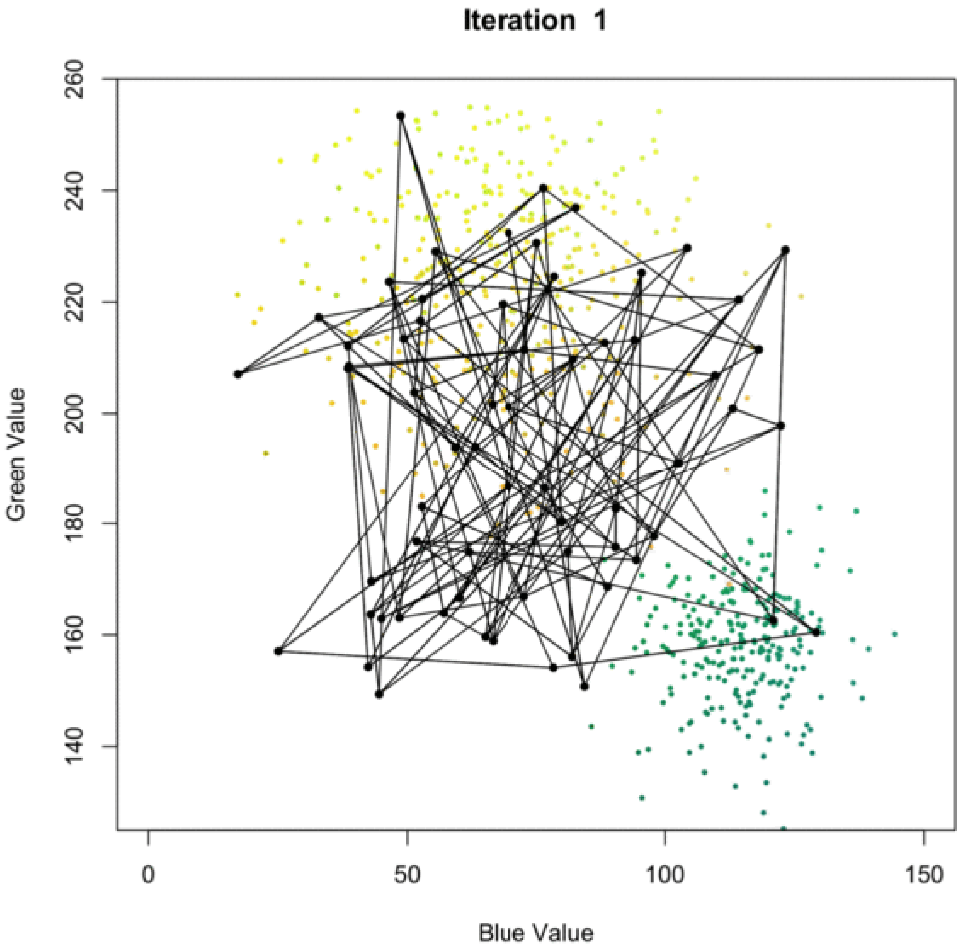
- In SOM's a type of Unsupervised Learning, called **Competitive Learning** is used.
- In Competitive Learning, the output neurons compete amongst themselves to be activated, with the result that only one is activated at any one time
- This activated neuron is called a winner-takes-all neuron or simply the winning neuron
- The result is that the neurons are forced to organise themselves. For obvious reasons, **such a network is called a Self Organizing Map (SOM)**

Self Organizing Maps

- Through multiple iterations, neurons on the grid will gradually gather around areas with a higher density of data points
- Areas with many neurons might reflect underlying clusters in the data

Analysis
of SOMs

Yellow – Green
Example



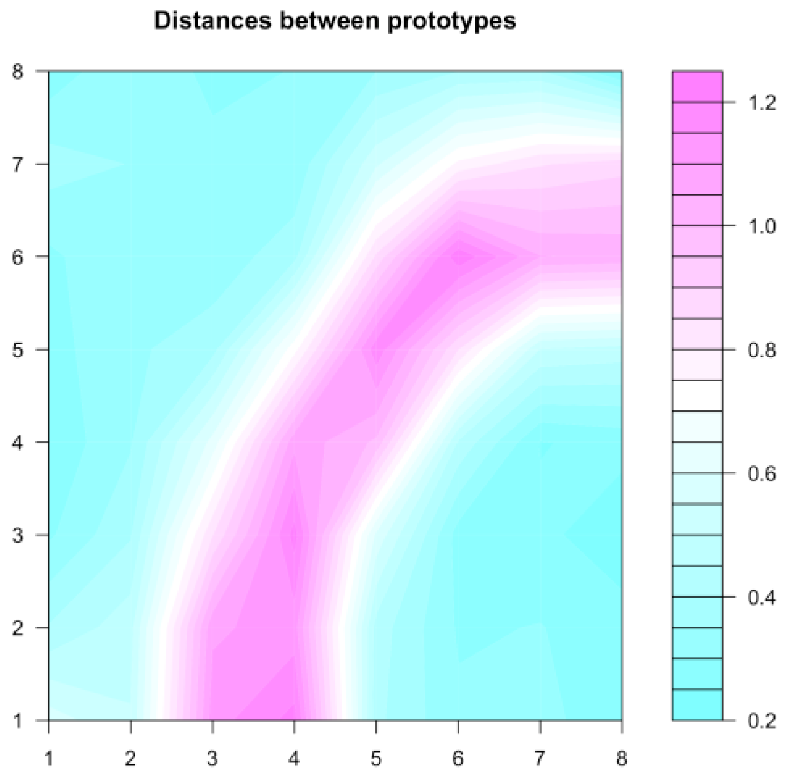
An Example Using Colors:

- x, y and z axis representing RGB
- Analyzing Yellow and Green
- 8x8 Grid, 64 Neurons Used
- Started at random locations
- Each Neuron is activated by a different colour
- By 150 Iterations, the map has taken nearly its final shape

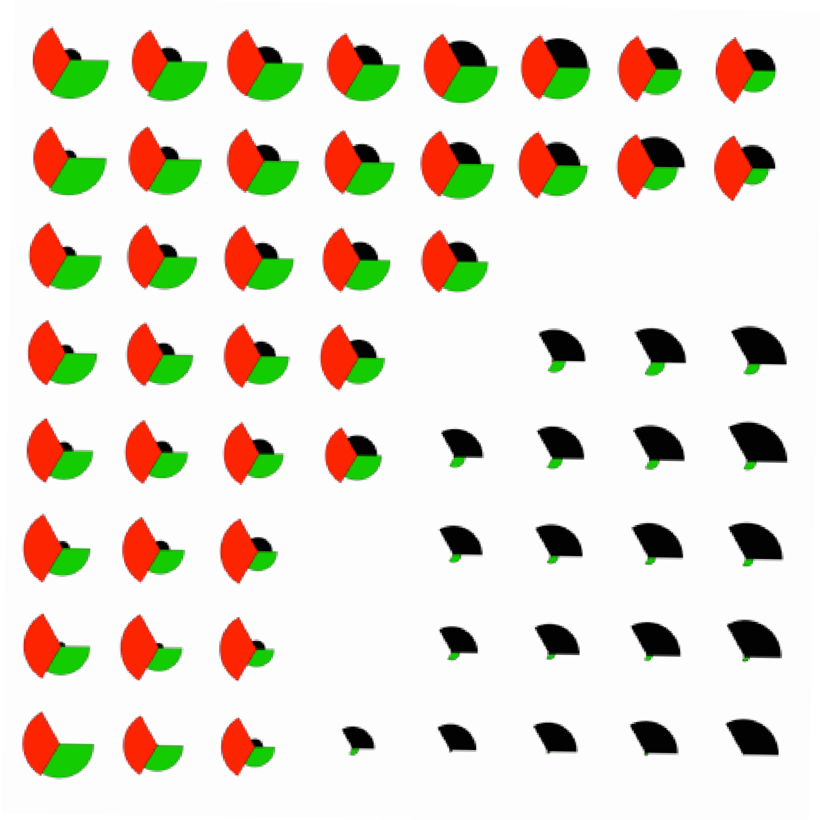
Analysis of SOMs

Yellow – Green Example

U Matrix



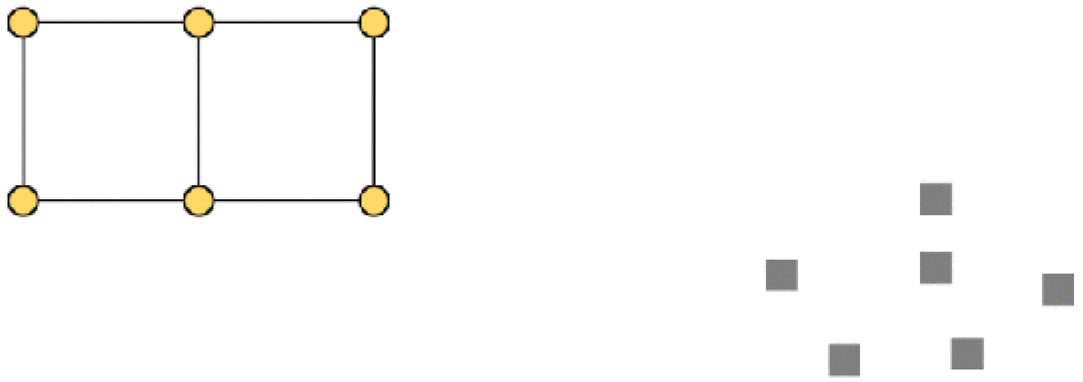
Actual Dataset (For Confirmation)



The **Algorithm**

- **Step 1**: Randomly position the grid's neurons in the data space.
- **Step 2**: Select one data point, either randomly or systematically, cycling through the dataset in order
- **Step 3**: Find the neuron that is closest to the chosen data point. This neuron is called the Best Matching Unit (BMU).
- **Step 4**: Move the BMU closer to that data point. The distance moved by the BMU is determined by a ***learning rate***, which decreases after each iteration.
- **Step 5**: Move the BMU's neighbors closer to that data point as well, with farther away neighbors moving less. Neighbors are identified using a radius around the BMU, and the value for this radius decreases after each iteration.
- **Step 6**: Update the learning rate and BMU radius, before repeating Steps 1 to 4. Iterate these steps until positions of neurons have been stabilized.

The **Algorithm**



Position neurons (orange) in data space.

Limitations

- Hard to Tackle Discrete Problems
- Computationally expensive
- Potentially inconsistent solutions.

Thanks

Shahmeer Khan - 199310

Essam Asif - 199368

Talha Ismail - 199379