Neural Networks
Neural Networks

- Neural network is a network or circuit of neurons

- Neurons can be
  - Biological neurons
  - Artificial neurons
Biological neurons

- Building block of the brain
- Human brain contains over 10 billion neurons
- Each of them connected to several thousand neurons
- 60 trillion connections
Biological Neurons

- Individual neuron
  - Extremely simple

- Complex network of neurons
  - Process information at great rate and of extraordinary complexity
  - Exceeds “Anything” else in the world
How do we learn?

- We learn because the brain learns!

- Plasticity – Property of a neuron to change the nature and number of their connections to other neurons “in response” to events that occur.
  
  Strong connection between neurons $\rightarrow$ correct solutions
  
  Weak connection between neurons $\rightarrow$ incorrect solutions
Artificial Neurons

- Modeled on the biological neuron
- Used to build artificial neural networks
- Invented by McCulloch and Pitts in 1943
- Currently smaller in terms on number and connections when compared to biological neurons
Working of Artificial Neurons

- Each Neuron (or node) receives a lot of input
- Inputs can have different weights
- A function called as activation function is applied to these input values
- The combination of the inputs on the activation function results in activation level
- Activation level is the output of the neuron
Activation function – Step function

- X-axis - combination of inputs (weighted sum)
- Y-axis - Activation level
- Activation level is chosen by comparing the input to a threshold
Other activation functions

- Sigmoid function
- Linear function
Working of Artificial Neurons

- Output of one neuron serve as input to other neurons
- No central controlling mechanism
- There will be a time lag between input and output due to the passing of information from one neuron to another
- Parallel nature of human brain enables it to calculate quickly
Perceptrons

- A simple neuron that is used to classify its inputs into one of two categories
  - Yes or No
  - True or False
- Perceptron can have any number of inputs
- Input can be arranged in a grid
  - Image
  - Field of vision
- Used for image classification or recognition tasks
Learning of a perceptron

- Inputs are assigned random weights between -0.5 and 0.5
- Training data is given and output is observed
  - Incorrect output $\rightarrow$ Weights are adjusted
- Function for modification
  $$ w_i \leftarrow w_i + (a \times x_i \times e) $$
  a – learning rate (0 < a < 1)
  e – error produced
- Each iteration is known as a epoch
Limitation of Perceptron

- Can only learn functions that are linearly separable
- A linearly separable function – a function that can be drawn in a two dimensional graph and a single straight line can be drawn to classify values
Multilayer neural networks

- Solve problem that are not linearly separable
- Combine layers of Perceptrons
- Feed forward network consists of input layer, hidden layer and output layer
Backpropagation

- Multilayer networks learn in the same way as single perceptron – Just more weights to alter
- Use sigmoid function instead of step function
- Weights are normally distributed over the range \(-2.4/n\) to \(2.4/n\), \(n\) \(\rightarrow\) number of inputs in the layer
- First phase – Data is fed from input to output
- Second phase – Feeding back error from output to input
Backpropagation

- Algorithm is iterated till the error values are sufficiently small
- Inefficient and too slow to model real-world problems
- Does human brain use this? NO
Improving Backpropagation

- Improve performance by including a value called as momentum in the formula to modify the weights (generalized delta rule)
- Momentum takes into account the extent to which the weight was changed in previous iteration
- Avoids local minima and move quickly through areas where the error space is not changing
Improving Backpropagation

- Alternative method to improve is to use hyperbolic tangent function instead of sigmoid function.
- Another alternative is to change the learning rate during the course of training the network.
- By combining the above with the generalized data rule the performance of backpropagation can be improved.
Recurrent Networks

- Feed forward networks
  - acyclic (no cycles in network)
  - Once trained their state is fixed
  - Doesn’t alter when new data is presented
- Recurrent networks
  - Arbitrary connection between any nodes
  - Internal state is altered when new data is given
  - Basically it has memory
Recurrent Networks

- Learning – Feeds its inputs through the network, including feeding back data from outputs to inputs and repeats the process until the values of output do not change. (equilibrium or stability state)
- The stable values of the network (known as fundamental memories) are output values used as response to the input received.
- Once trained, for any given input it will output the closest attractor to it.
The activation function used is sign activation function

\[ \text{Sign}(X) = \begin{cases} +1 & \text{for } X > 0 \\ -1 & \text{for } X < 0 \end{cases} \]

The weights of the network are represented by a matrix \( W \), which is calculated as,

\[ W = \sum_{i=1}^{N} X_i X_i^t - N I \]
Recurrent Networks - Hopsfield Networks

- First stage – Train the network to learn the attractor states (memorization state)
- Second stage – Checking the network by giving attractors as input
- Third stage – Using the network (getting data from memory)
Recurrent Networks – Bidirectional associative memory

- Similar to Hopsfield
- Used to associate items from one set to items in another set
- Extremely useful but its capabilities and limitations are currently not fully understood
Unsupervised learning algorithms

- These algorithms learn to classify without being presented any preclassified training data
- Kohonen Maps
- Hebbian learning
Kohonen Maps

- Winner-take-all algorithm
  - Only one neuron provides the output
  - The neuron which has highest activation level
- Two layers
  - Input layer
  - Cluster layer (output layer)
- Each input layer node is connected to every output layer node
Hebbian learning

- Based on Hebb’s law which was stated by D.O. Hebb in 1949.
- If two neurons connected to each other fire at same time, the weight of the connection between them is increased.
- Conversely, if those neurons fire at different times, the weight of the connection between them is decreased.
- Example: Parlov’s experiment – Dog ➔ Bell and food
That’s it!