# BOLTZMANN MACHINES

**Regression, CNN,RNN,LSTM-Supervised Learning**

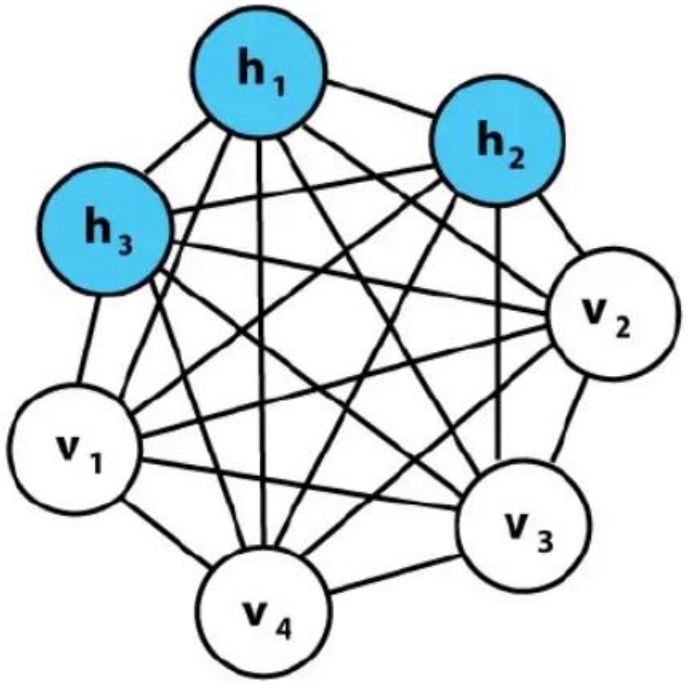
- We have Inputs, outputs, train the model till we get a desired output.
- Our model generates output, then we have comparison with the actual output.
- Then we decide, whether model has given the correct output(close to correct output) or not.
- If Predicted output not equal to correct output, we use gradient descent(weight adjustment)

**The model is directed.**

- Input-> layer 1 ->Layer 2 -> output.

## WHAT IS A BOLTZMANN MACHINE

- Boltzmann machine- Part of **Unsupervised Learning**.
- We provide some **input** to the **model**, we **let the model** decide the **relationship between the features** in the data.
- Here, we do not provide the model with any output.



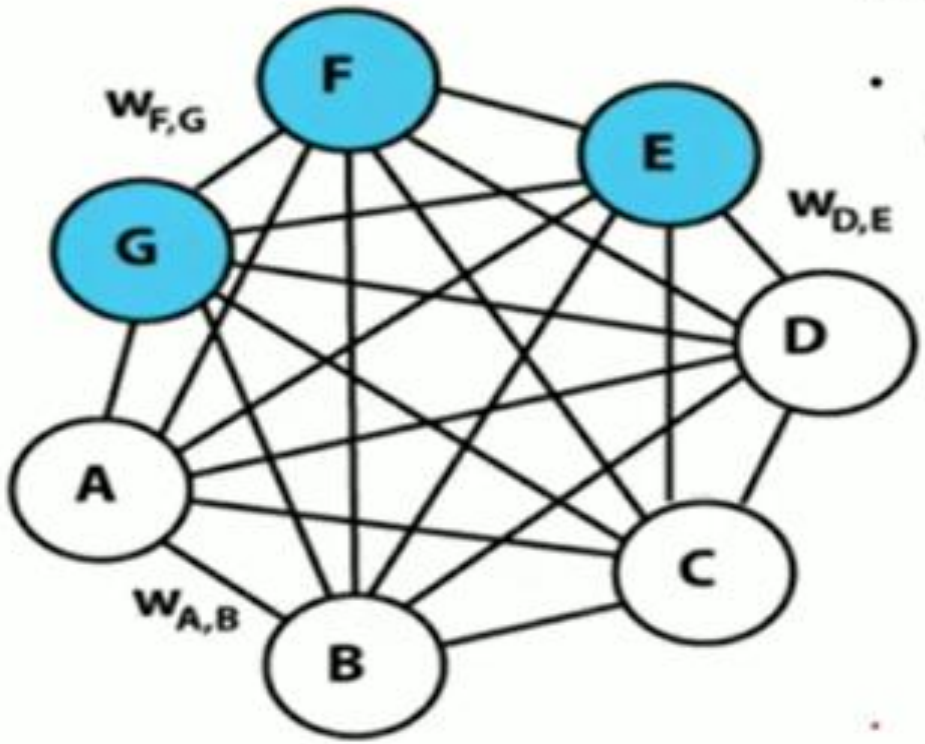

A Boltzmann Machine is a type of artificial neural network that was introduced by Geoffrey Hinton and Terry Sejnowski in the 1980s. It is a stochastic, generative model that can learn and represent the probability distribution of a given set of binary or real-valued data.

## WHAT IS A BOLTZMANN MACHINE(Contd..)

A Boltzmann Machine is a network of **symmetrically connected**, neuron like units that make **stochastic**(Random Probability Distribution) **decisions** about whether to be **on or off**

- Undirected Model-Connection goes both the ways.
- Non-Deterministic model
- Purpose-To optimize the solutions(Optimizing Travelling Sales Man Problem)
- Discovers features from datasets composed of binary vectors.

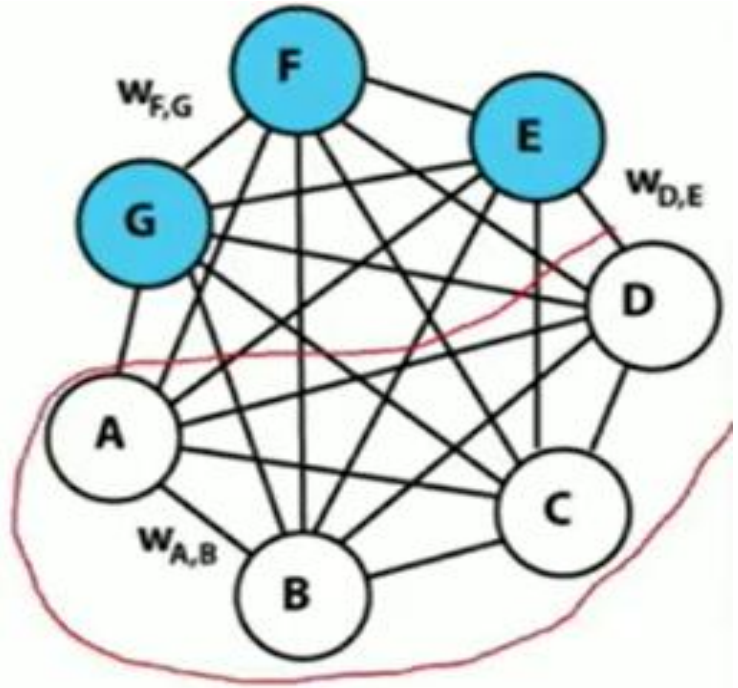
**A graphical representation of a Boltzmann machine**

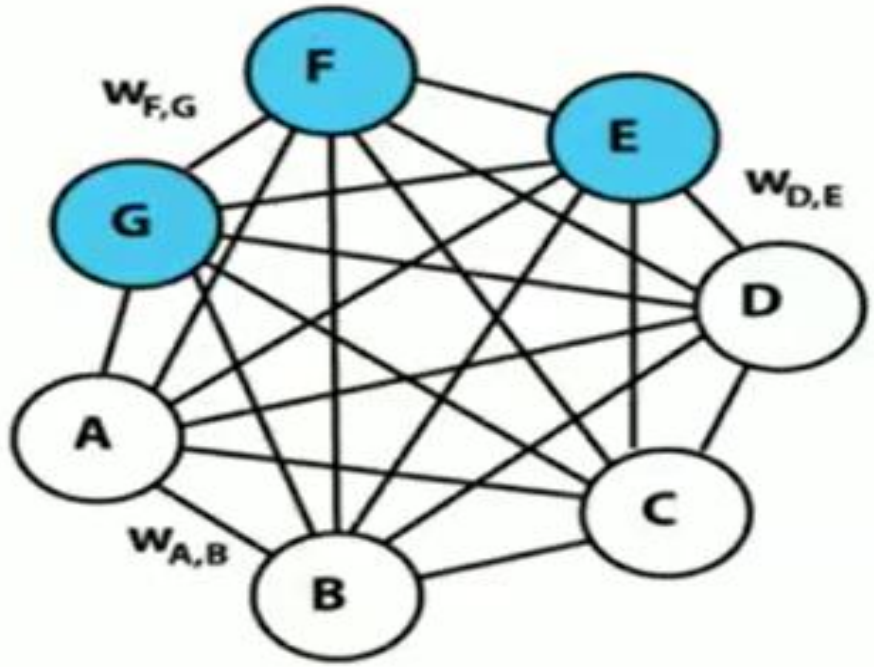



- A few weights labelled.
- Each undirected edge represents dependency and is weighted with weight .

- In this example there are **3 hidden units** (blue) and **4 visible units** (white).

- Visible(white) layers- Also called as input layers
    - Input1-A, Input2-B, Input3-C, Input4-D
- Hidden (blue) layers-neurons are Not visible
- **Every** node is connected to **every other node** (even inputs are connected with each other).
- **No output layer**






- The machine tries to find the **relationship** between the **inputs**, using the **features**.
- The features are created by machine, when it starts learning from input data.
- The machine correlates the data on basis of data that we provided.

# User – Movie Rating Matrix

| | User1 | User2 | User3 | User4 | User5 |
|---|---|---|---|---|---|
| Movie 1 | 4 | 5 | 3 | 4 | 4 |
| Movie 2 | 2 | 1 | 5 | 2 | 2 |
| Movie 3 | 5 | 3 | 4 | 4 | |
| Movie 4 | 4 | 5 | 4 | 4 | 4 |

| | | | | |
|---|---|---|---|---|
| 4 | 5 | 3 | 4 | 4 |
| 2 | 1 | 5 | 2 | 2 |
| 5 | 3 | 4 | 4 | |
| 4 | 5 | 4 | 4 | 4 |


Reconstructed Input

| |
|---|
| 1 |
| 0 |
| 1 |

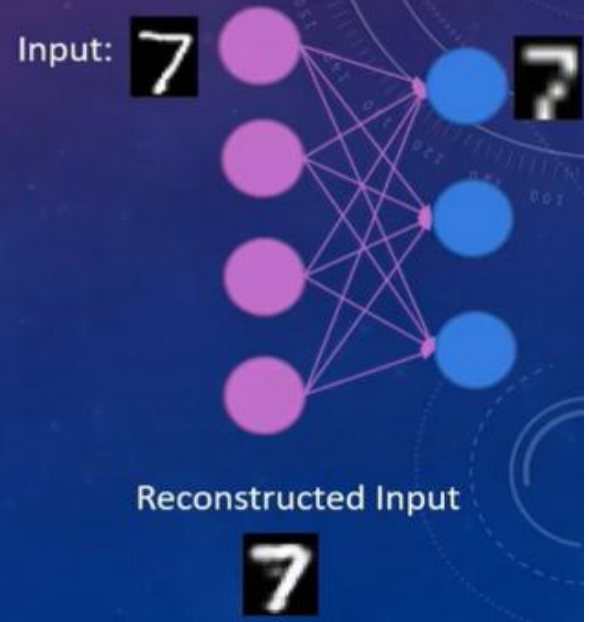
- Based on reconstructed input, we can guess if user5 is interested to watch movie3 & recommend it
- It is a way of solving collaborative filtering, which is a type of recommender system
- The network that can make such a model is called a Restricted Boltzmann Machine.

# Restricted Boltzmann Machine

- RBMs are shallow neural networks  with only two layers.
- Automatically find patterns in data by reconstructing input
  - Extract important features and reconstruct input
  - Combine features to form patterns
- Unlabelled data (real-world data)
  - Photos
  - Videos
  - Voices
  - Sensor data
- First layer is called the visible layer
- Second layer is the hidden layer
- "restricted" because neurons within the same layer are not connected

# Restricted Boltzmann Machine

- Feeding the input data, the network learns its weights.
- Feeding an input image, the values that appear in the hidden layer can be considered as features learnt automatically from the input data.
- There are a smaller number of units in the hidden units of an RBM
  - Good representation of data that are lower in dimensionality (Compared to original data)




- Useful in applications:
  - dimensionality reduction
  - feature extraction
  - collaborative filtering

- RBMs are used as the main block of another type of Deep Neural Network called, Deep Belief Networks

- Each visible node takes a low-level feature from an item in the dataset to be learned.
- At node 1 of the hidden layer, x is multiplied by a *weight* and added to a *bias*.
- The result of those two operations is fed into an *activation function*
- This produces the node's output, or the strength of the signal passing through it


- The learning process consists of several forward and backward passes

- In the forward pass,

  RBM takes the inputs and translates them into a set of numbers that encode the inputs.

- In the backward pass,

  it takes this set of numbers and translates them back to form the re-constructed inputs.

# Deep Belief Networks