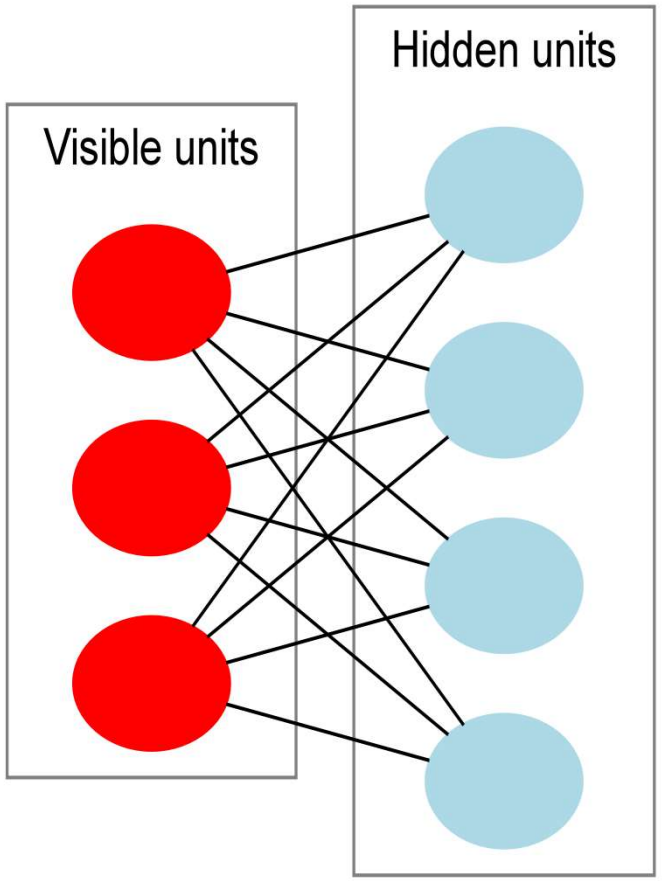
Restricted Boltzmann Machine (RBM)

Restricted Boltzmann Machine (RBM):

- A type of ANN used for unsupervised learning, meaning it learns patterns without labeled data.

- Consists of 2 layers of neurons:
  1. **Visible layer:** represents the input data.
  2. **Hidden layer:** represents a set of features that are learned by the network/model.

- Why called "**restricted**"?
  - Neurons within the same layer are not connected; connections exist only between the visible and hidden layers, enabling the network to learn a compressed representation of the input data by reducing input dimensionality.



Restricted Boltzmann Machine (RBM):

- A generative model that learns probability distributions over input data and, through internal representations, can address complex combinatorial problems.

- RBM is trained using a method called **contrastive divergence** (a variant of the stochastic gradient descent algorithm), helping it learn to represent input data more compactly.

- During training, the network adjusts the weights of the connections between the neurons in order to maximize the likelihood of the training data.

- Once trained, it can be used to generate new samples from the learned probability distribution.

- RBM is useful for Collaborative Filtering, dimensionality reduction, classification, regression, feature learning, topic modeling and even Deep Belief Networks.

How do Restricted Boltzmann Machines work?

In RBM there are 2 phases through which the entire RBM works:

1. **1st Phase: Feed Forward Pass:**
   - **Purpose**: Activates the hidden layer based on the visible layer input.
   - **Process**:
     1. Calculate positive associations (positive correlations b/w visible and hidden units).
     2. Calculate negative associations (negative correlations b/w visible and hidden units).

2. **2nd Phase: Feed Backward Pass:**
   - **Purpose**: Reconstruct the input layer from the activated hidden state.
   - **Process**:
     1. Reconstruct the visible layer from the hidden layer.
     2. Calculate the error: Difference b/w the reconstructed and actual input layers.
     3. Adjust weights: Based on the error, using the formula:

Adjust Weight = Input × Error × Learning Rate (0.1)

Example Illustration

Assumption:

- Visible unit **V1** activates hidden units **h1** and **h2**.
- Visible unit **V2** activates hidden units **h2** and **h3**.

Now when any new visible unit let **V5** has come into the machine and it also activates the **h1** and **h2** unit.

So, we can back trace the hidden units easily and also identify that the characteristics of the new **V5** neuron is matching with that of **V1**. This is because **V1** also activated the same hidden unit earlier.

Restricted Boltzmann Machine
Working example as recommendation system

1	0		1	1	1
0	1	0	0	1	0
	1	1	0	0	
1	0	1	1	0	1
0		1	1		1
0	0	0	0	1	
1	0	1	1	0	1
0	1	1		0	1
	0	1	1	1	1
1		0	0		0
0	1	1	1	0	1