# Boltzmann Machines

075BCT080 (Sanskar Amgain)
075BCT094 (Tilak Chad)
075BCT095 (Udeshya Dhungana)

# Boltzmann Machine

- A kind of Recurrent Neural Network
- Unsupervised advanced DL model
- Energy based model
- System stable in its lowest energy state
- Not a deterministic model but a stochastic model

Named after **Boltzmann distribution**

Boltzmann distribution are used in Boltzmann machine sampling function

Boltzmann distribution aka Gibbs distribution given by :

$$p_i = \frac{1}{Q} e^{-\varepsilon_i/(kT)} = \frac{e^{-\varepsilon_i/(kT)}}{\sum_{j=1}^{M} e^{-\varepsilon_j/(kT)}}$$

(where symbols have their usual meaning)

gives the probability of the certain state as a function of state's energy and temperature of the system.

# Components of Boltzmann Machine

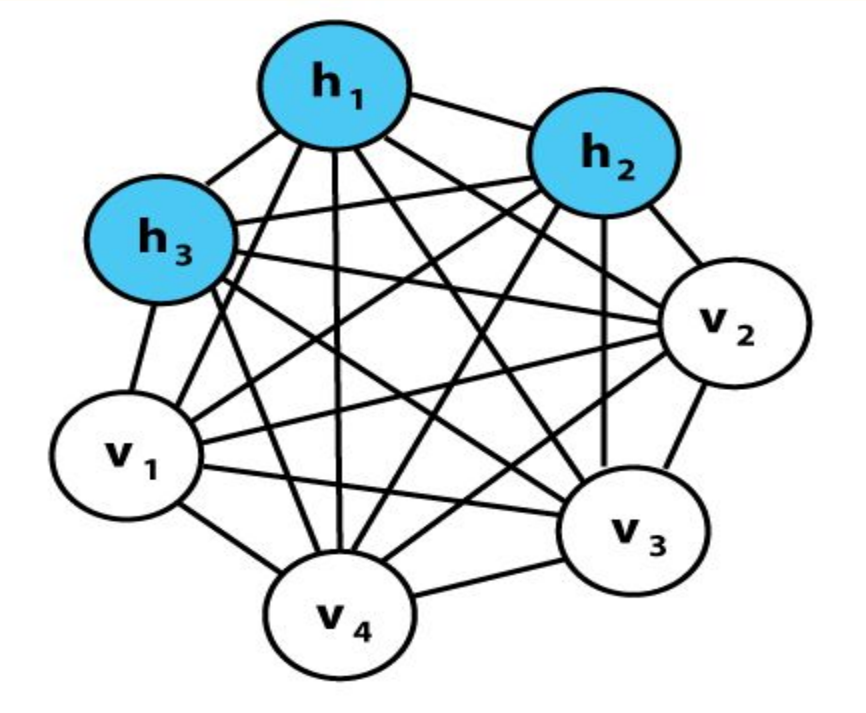



Fig. A sample Boltzmann machine (v - visible layers, h - hidden layers)

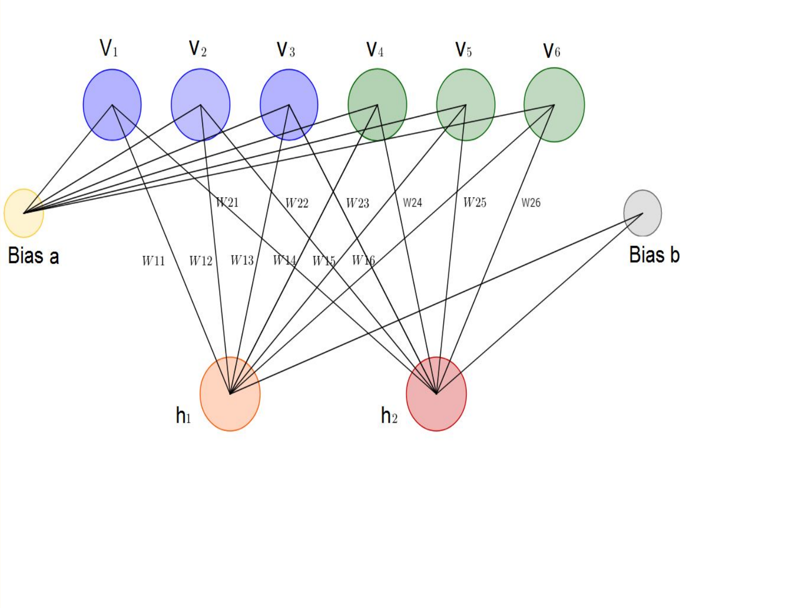
- Composed of two basic layers :
  i) Visible layer

  ii) Hidden layer
- Consists of neuron like unit called nodes, interconnected
- Nodes makes binary decisions and are presented with certain biases

# Applications

- For solving different computation problems
- Optimize the solution of the problem
- Used for unsupervised learning like :
    - Clustering
    - Dimensionality Reduction
    - Anomaly Detection
- Identifying :
    - Latent grouping
    - Irregularities in data
- Generating new samples from the available data

# Restricted Boltzmann Machines (RBMs)




- Bipartite Graph
- An edge is allowed only between a node from hidden and a visible layer
- Faster than traditional BMs
- Efficient

# Features

# Major Operations

1. Initialize weights and biases
2. Sampling
3. Forward Pass
4. Reconstruction
5. Free Energy Calculation

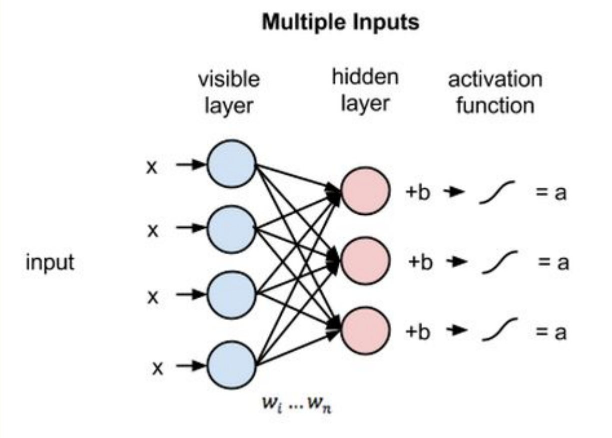
# Sampling Function

1. A stochastic function that, on average, rounds the number smaller than 0.5 to 0 and greater to 1.

2. In more detailed manner, we do the following:
   a. Subtract a vector whose elements belong to Standard Normal Distribution(Or any distribution)
   b. Apply Sign function
   c. Apply Relu

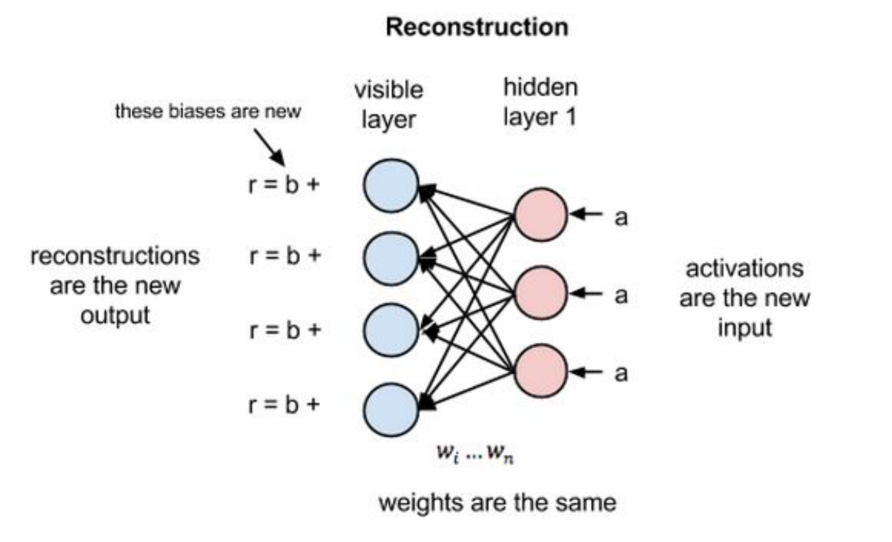
# Forward Pass (Visible Layer to Hidden Layer)

1. Calculates probabilities of hidden layer given weights, biases and sample values in visible layer
2. For each node in hidden layer, the probability is calculated as :
   a. Add the following
      i. sample values connecting a visible node to that hidden node
      ii. it’s corresponding edge value (entry from weight matrix)
      iii. Bias Vector
3. State is sampling function applied to the probability



# Reconstruction (Hidden to Visible Layer)

We repeat the same process from the hidden to the visible layer



# Free Energy Calculation

Free Energy of a system is defined as:

$$
\begin{aligned}
E(v,h) &= -h^T W v - c^T v - b^T h \\
&= -\sum_{j,k} W_{jk} h_j v_k - \sum_k c_k v_k - \sum_j b_j h_j
\end{aligned}
$$

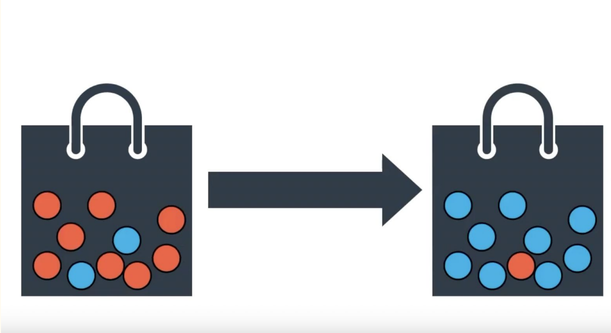
Sum of states, weights and biases of every node and edge in the network (Both visible and hidden layer)

# Gibbs Sampling

One forward pass to hidden layer and one reconstruction of visible layer is called Gibbs Sampling. Generally used to Sample data from a joint distribution

# Updating Parameters

We update parameters (weights and biases in the network) such as way that it mimics the distribution of our input data.



$$\frac{\partial(-\ln p(v))}{\partial\theta} = \mathbb{E}_h\left[\frac{\partial E(v,h)}{\partial\theta}|v\right] - \mathbb{E}_h\left[\frac{\partial E(\tilde{v},h)}{\partial\theta}|\tilde{v}\right]$$

$$\frac{\partial E(v,h)}{\partial W_{jk}} = \frac{\partial}{\partial W_{jk}}[-\sum_{j,k} W_{jk}h_jv_k - \sum_k c_kv_k - \sum_j b_jh_j]$$

# Algorithm

For each training example:

X_sample := Gibbs_Sampling(example, steps)

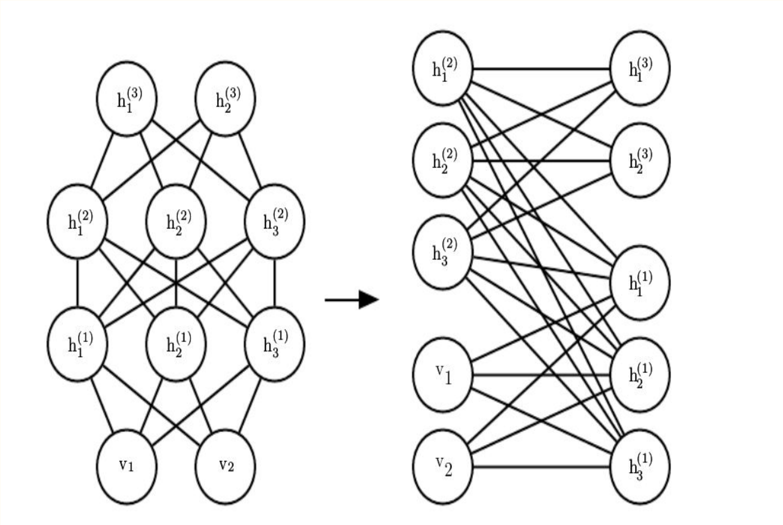
Adjust Weights

$$W \leftarrow W + \alpha(-h(x^{(t)})(x^{(t)})^T - h(\tilde{x})\tilde{x}^T)$$

Adjust biases

$$b \leftarrow b + \alpha(-h(x^{(t)}) - h(\tilde{x}))$$
$$c \leftarrow c + \alpha(x^{(t)} - \tilde{x})$$

# Deep Boltzmann Machine(DBM)




# Features

- Deep Generative Model
- More than one hidden layers with directionless connection
- Bipartite as RBM
- Training process computationally expensive than RBMs

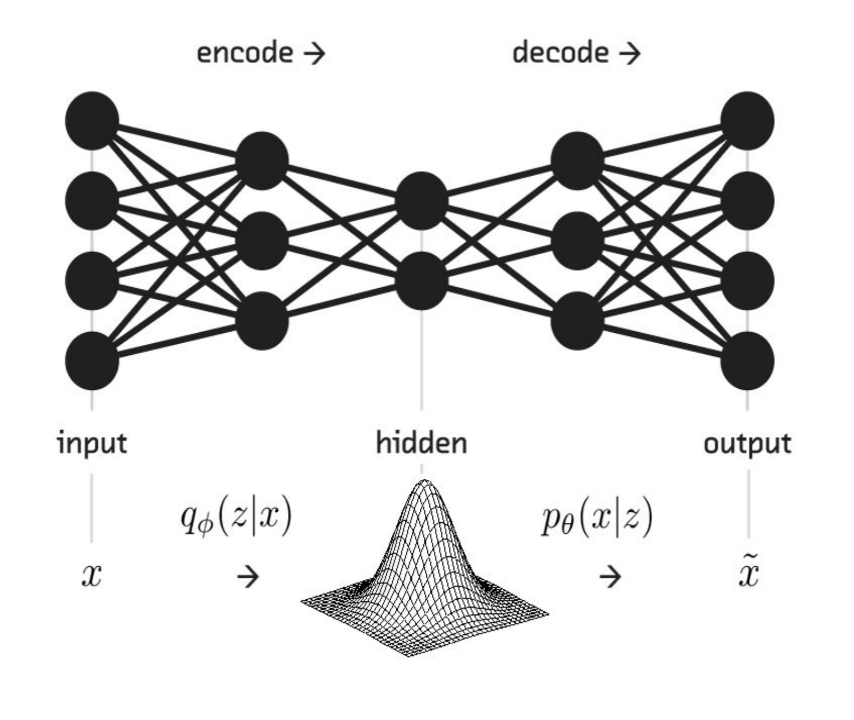
# Limitations and Current Scenario

# Limitations

- Training is more difficult than due to energy gradient function
- CD-k algorithm less intuitive than backpropagation
- Cannot be scaled up for bigger applications

# Current Scenario

- Occasionally used to pre-training or dimensionality reduction
- Replaced with more complex Generative Adversarial Networks(GANs) and Variational Autoencoders (VAE)



# Demo

Result