Restricted Boltzmann Machines

Tushar B. Kute,
http://tusharkute.com

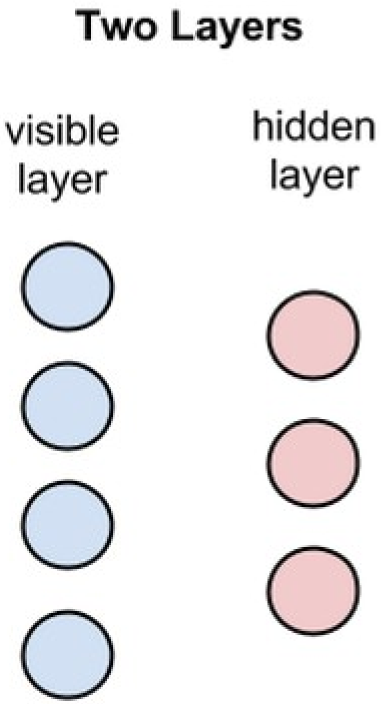
Restricted Boltzmann Machines

- Invented by Geoffrey Hinton, a Restricted Boltzmann machine is an algorithm useful for dimensionality reduction, classification, regression, collaborative filtering, feature learning and topic modeling.
- Given their relative simplicity and historical importance, restricted Boltzmann machines are the first neural network we'll tackle.

Restricted Boltzmann Machines

- RBMs are shallow, two-layer neural nets that constitute the building blocks of deep-belief networks.
- The first layer of the RBM is called the visible, or input, layer, and the second is the hidden layer.

Restricted Boltzmann Machines



Restricted Boltzmann Machines

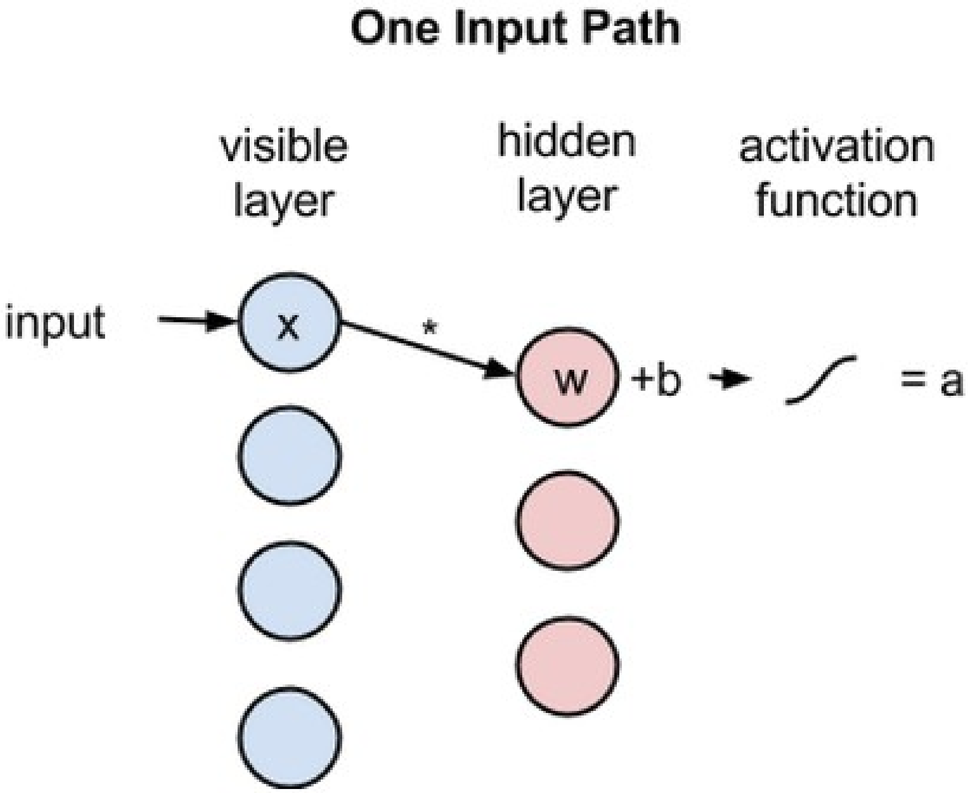
- Each circle in the graph above represents a neuron-like unit called a node, and nodes are simply where calculations take place.
- The nodes are connected to each other across layers, but no two nodes of the same layer are linked.
- That is, there is no intra-layer communication – this is the restriction in a restricted Boltzmann machine.
- Each node is a locus of computation that processes input, and begins by making stochastic decisions about whether to transmit that input or not.

Restricted Boltzmann Machines

- Each visible node takes a low-level feature from an item in the dataset to be learned. For example, from a dataset of grayscale images, each visible node would receive one pixel-value for each pixel in one image. (MNIST images have 784 pixels, so neural nets processing them must have 784 input nodes on the visible layer.)
- Now let's follow that single pixel value, x, through the two-layer net. At node 1 of the hidden layer, x is multiplied by a weight and added to a so-called bias.
- The result of those two operations is fed into an activation function, which produces the node's output, or the strength of the signal passing through it, given input x.

activation f((weight w * input x) + bias b) = output a

Restricted Boltzmann Machines

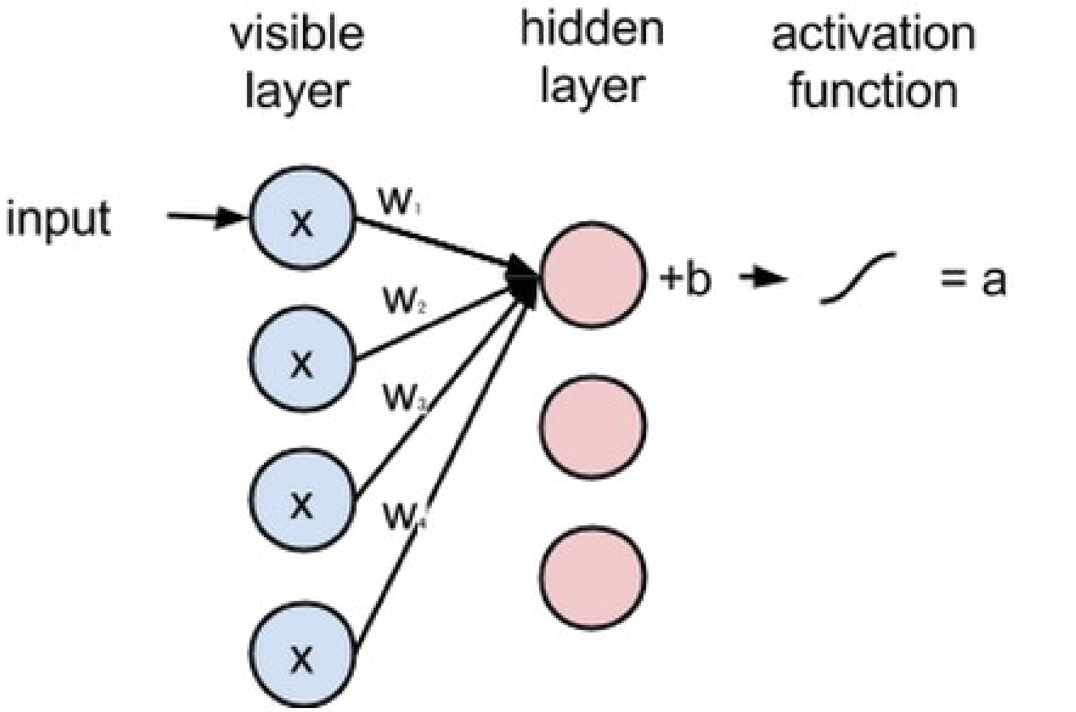


Restricted Boltzmann Machines

- Next, let’s look at how several inputs would combine at one hidden node.
- Each x is multiplied by a separate weight, the products are summed, added to a bias, and again the result is passed through an activation function to produce the node’s output.

Restricted Boltzmann Machines

Weighted Inputs Combine @Hidden Node



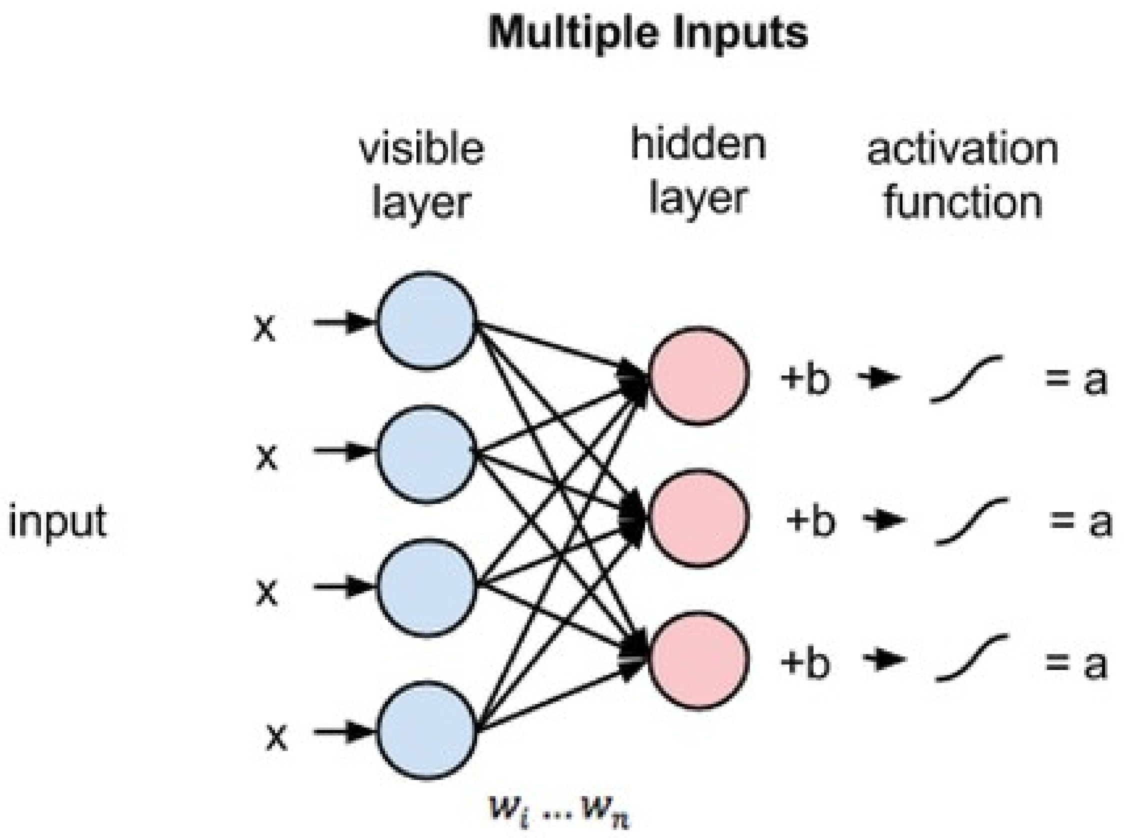
Restricted Boltzmann Machines

- Because inputs from all visible nodes are being passed to all hidden nodes, an RBM can be defined as a symmetrical bipartite graph.
- Symmetrical means that each visible node is connected with each hidden node (see below).
- Bipartite means it has two parts, or layers, and the graph is a mathematical term for a web of nodes.

Restricted Boltzmann Machines

- At each hidden node, each input x is multiplied by its respective weight w. That is, a single input x would have three weights here, making 12 weights altogether (4 input nodes x 3 hidden nodes).
- The weights between two layers will always form a matrix where the rows are equal to the input nodes, and the columns are equal to the output nodes.
- Each hidden node receives the four inputs multiplied by their respective weights.
- The sum of those products is again added to a bias (which forces at least some activations to happen), and the result is passed through the activation algorithm producing one output for each hidden node.

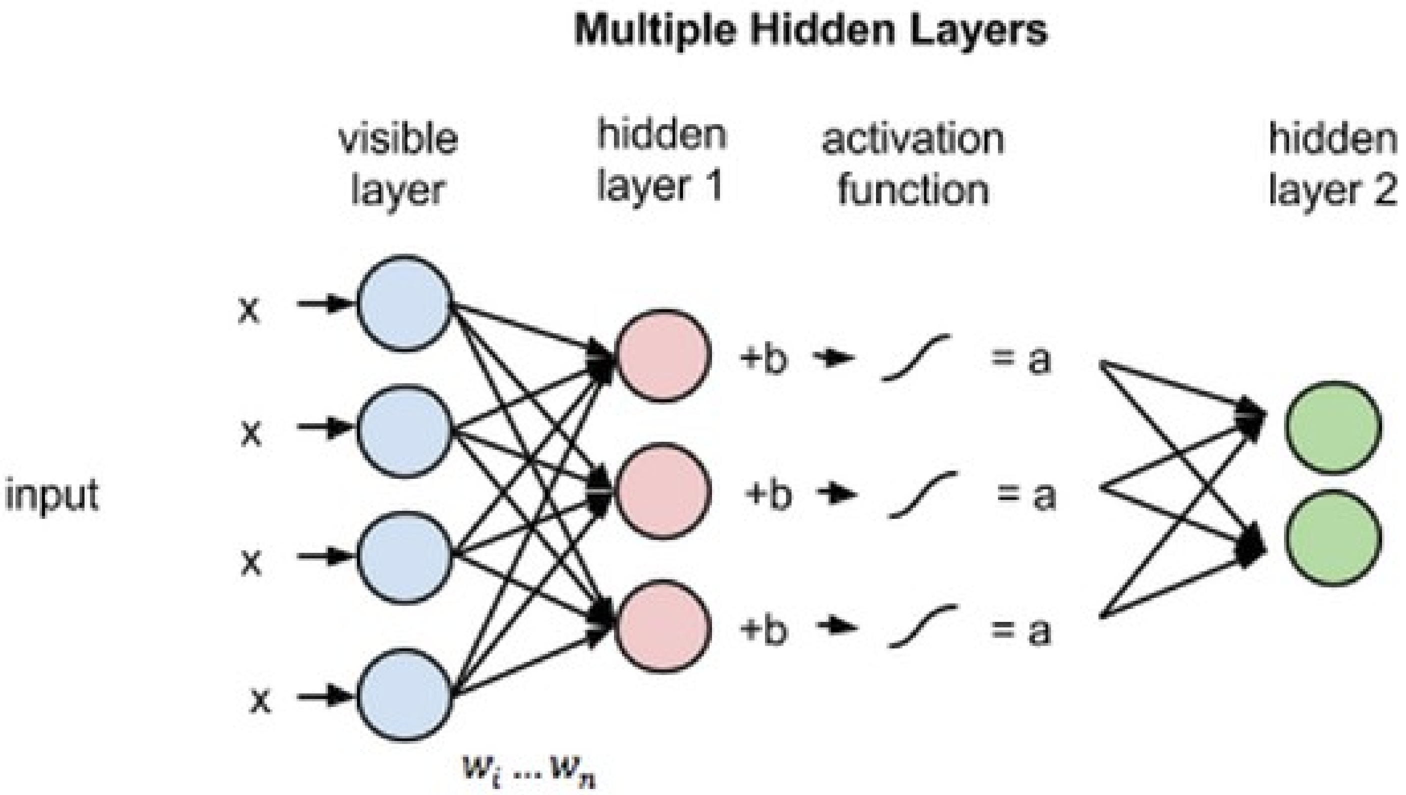
Restricted Boltzmann Machines



Restricted Boltzmann Machines

- If these two layers were part of a deeper neural network, the outputs of hidden layer no. 1 would be passed as inputs to hidden layer no. 2, and from there through as many hidden layers as you like until they reach a final classifying layer. (For simple feed-forward movements, the RBM nodes function as an autoencoder and nothing more.)

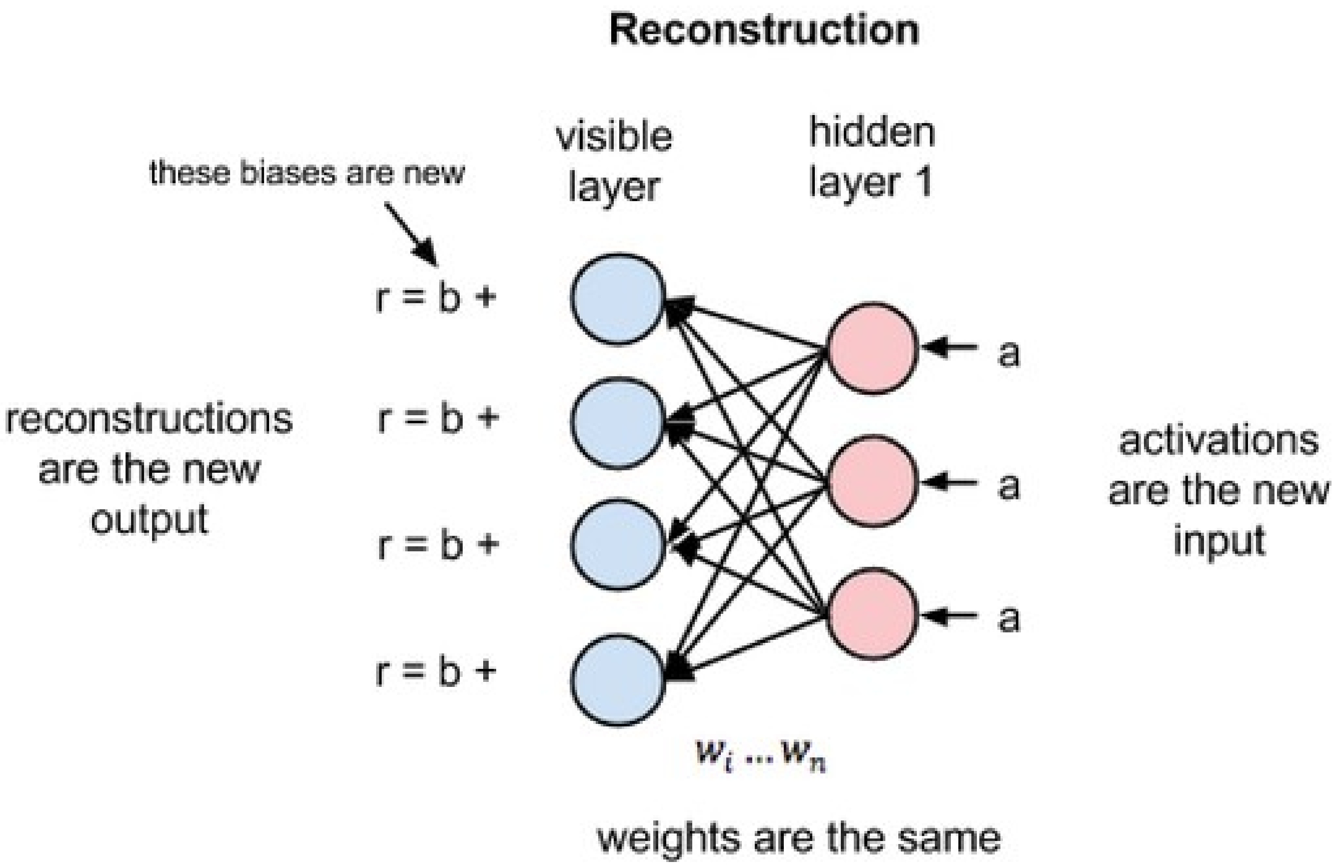
Restricted Boltzmann Machines



RBM – Reconstructions

- In the reconstruction phase, the activations of hidden layer no. 1 become the input in a backward pass.
- They are multiplied by the same weights, one per internode edge, just as x was weight-adjusted on the forward pass.
- The sum of those products is added to a visible-layer bias at each visible node, and the output of those operations is a reconstruction; i.e. an approximation of the original input.

RBM – Reconstructions



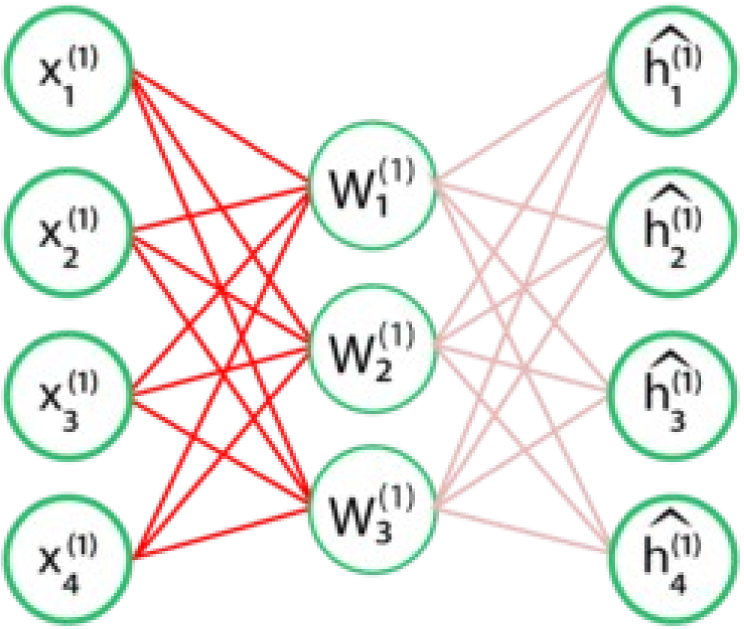
Restricted Boltzmann Machines

- Because the weights of the RBM are randomly initialized, the difference between the reconstructions and the original input is often large.
- You can think of reconstruction error as the difference between the values of r and the input values, and that error is then backpropagated against the RBM’s weights, again and again, in an iterative learning process until an error minimum is reached.

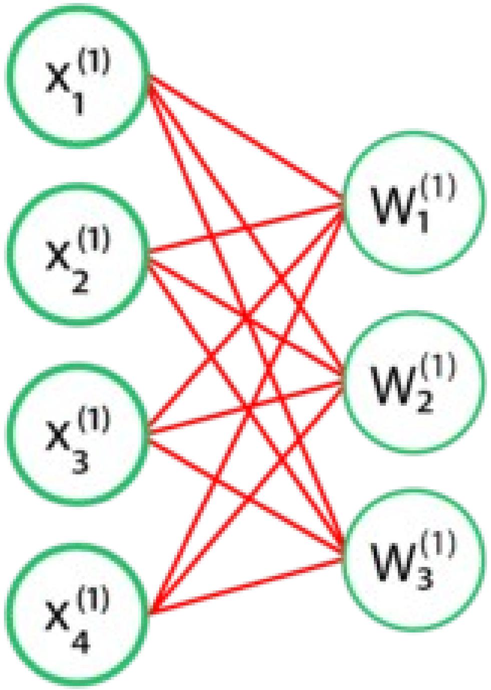
Difference between Autoencoders & RBMs

- Autoencoder is a simple 3-layer neural network where output units are directly connected back to input units.
- Typically, the number of hidden units is much less than the number of visible ones.
- The task of training is to minimize an error or reconstruction, i.e. find the most efficient compact representation for input data.

Difference between Autoencoders & RBMs



AUTOENCODERS



RBMs

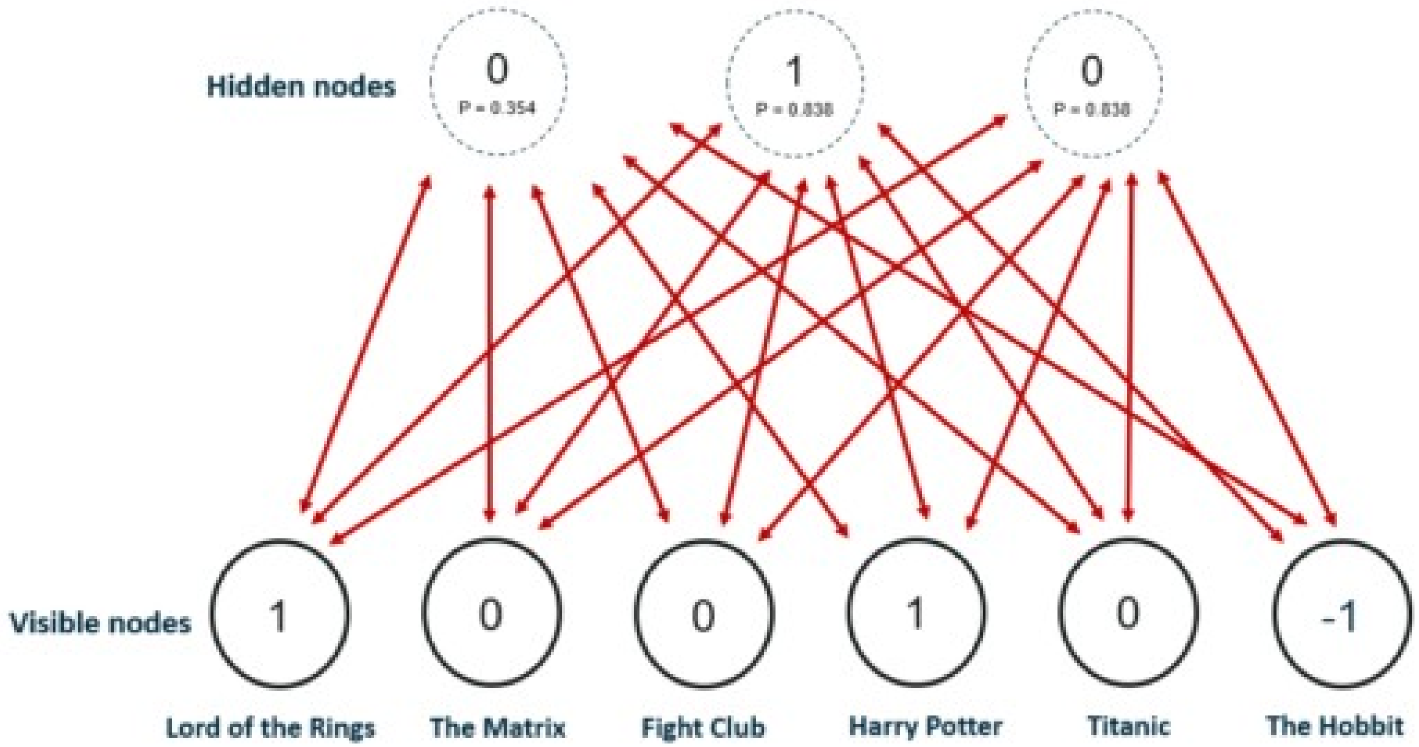
Difference between Autoencoders & RBMs

- RBM shares a similar idea, but it uses stochastic units with particular distribution instead of deterministic distribution.
- The task of training is to find out how these two sets of variables are actually connected to each other.
- One aspect that distinguishes RBM from other autoencoders is that it has two biases.
  - The hidden bias helps the RBM produce the activations on the forward pass, while
  - The visible layer's biases help the RBM learn the reconstructions on the backward pass.

Collaborative Filtering

- Recognizing Latent factors in the Data
  - Let us assume that some people were asked to rate a set of movies in the scale of 1-5 and each movie could be explained in terms of a set of latent factors such as drama, fantasy, action and many more.
  - Restricted Boltzmann Machines are used to analyze and find out these underlying factors.

Collaborative Filtering



Collaborative Filtering

- The analysis of hidden factors is performed in a binary way, i.e, the user only tells if they liked (rating 1) a specific movie or not (rating 0) and it represents the inputs for the input/visible layer.
- Given the inputs, the RMB then tries to discover latent factors in the data that can explain the movie choices and each hidden neuron represents one of the latent factors.

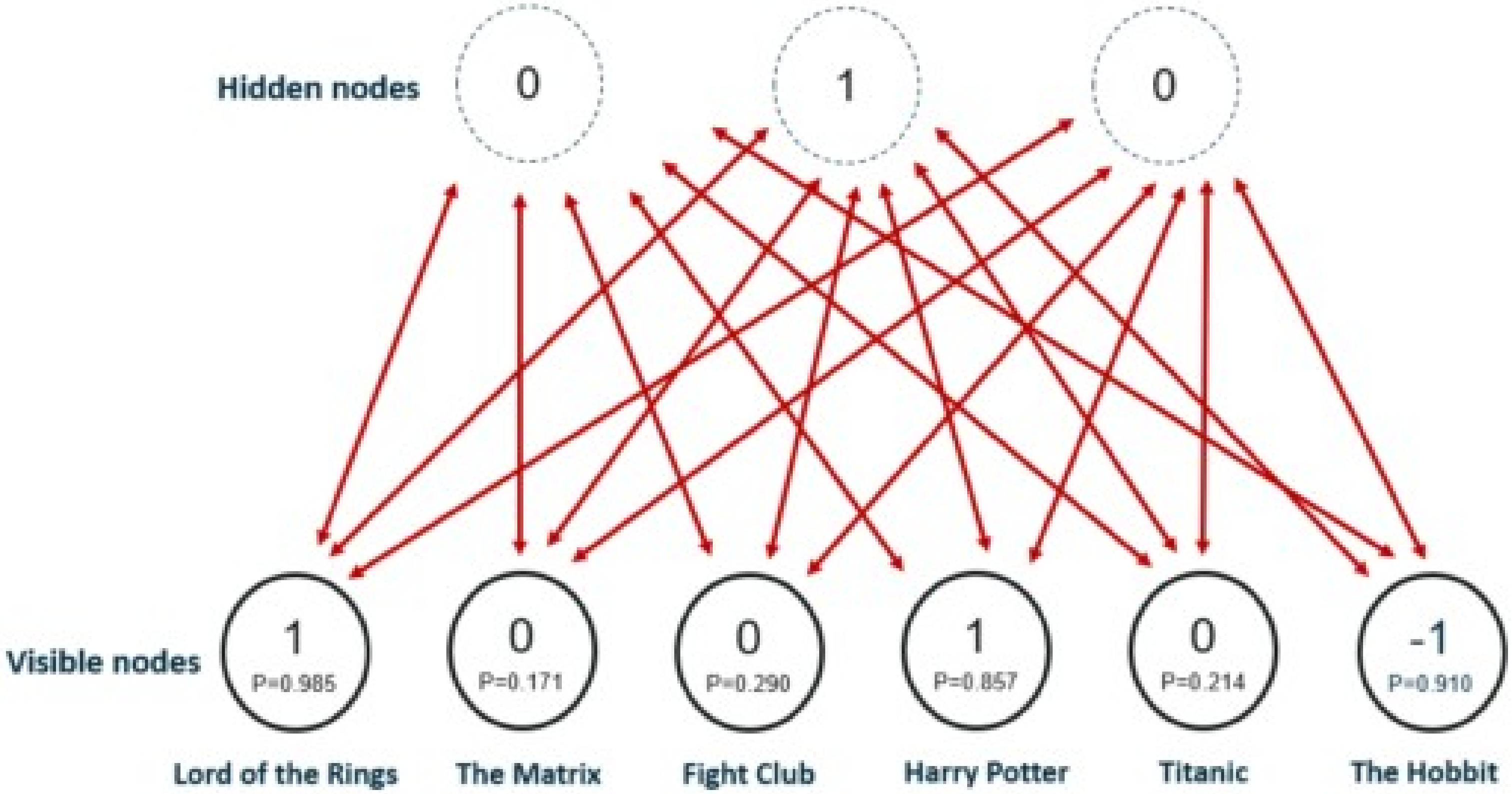
Collaborative Filtering

- Let us consider the following example where a user likes Lord of the Rings and Harry Potter but does not like The Matrix, Fight Club and Titanic.
- The Hobbit has not been seen yet so it gets a -1 rating. Given these inputs, the Boltzmann Machine may identify three hidden factors Drama, Fantasy and Science Fiction which correspond to the movie genres.

Collaborative Filtering

- Using Latent Factors for Prediction
  - After the training phase, the goal is to predict a binary rating for the movies that had not been seen yet.
  - Given the training data of a specific user, the network is able to identify the latent factors based on the user's preference and sample from Bernoulli distribution can be used to find out which of the visible neurons now become active.

Collaborative Filtering



Collaborative Filtering

- The image shows the new ratings after using the hidden neuron values for the inference. The network identified Fantasy as the preferred movie genre and rated The Hobbit as a movie the user would like.
- The process from training to the prediction phase goes as follows:
  - Train the network on the data of all users
  - During inference-time, take the training data of a specific user
  - Use this data to obtain the activations of hidden neurons
  - Use the hidden neuron values to get the activations of input neurons
  - The new values of input neurons show the rating the user would give yet unseen movies

Thank you

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

MITUSkillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com