Restricted Boltzmann Machines

A "Light" Introduction

Brian McMahon
26 March 2018

What is a Restricted Boltzmann Machine?

- ➔ Generative, stochastic, undirected and shallow ANN
- ➔ Find patterns (probability distribution) in data by reconstructing input
- ➔ Invented as “Harmonium” by Paul Smolensky in 1986; popularized by Geoffrey Hinton in early 2000s
- ➔ Nodes connected across layers, but no two nodes of same layer are linked (the “restriction” in RBM; simplifies learning)
- ➔ Common building block of deep probabilistic models such as deep belief networks (DBNs)

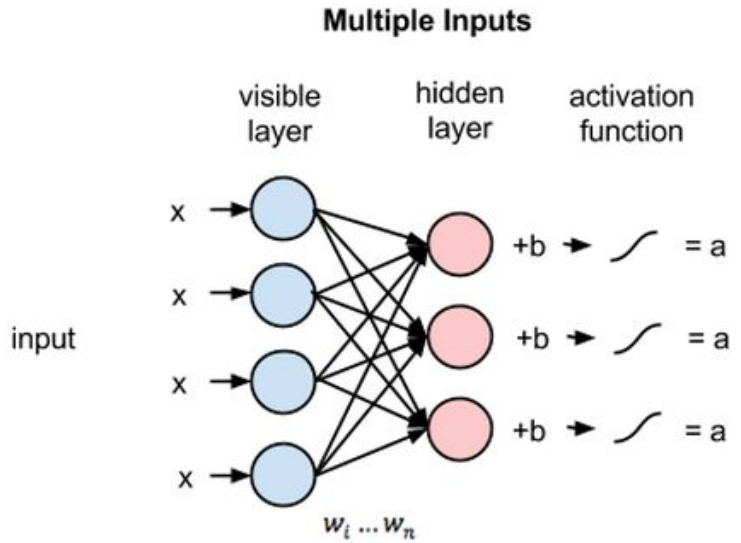
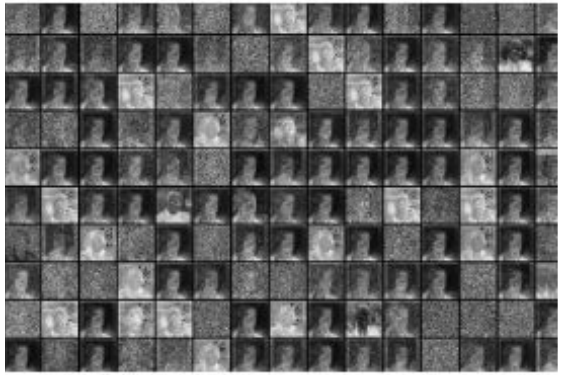


Image courtesy of deeplearning4j

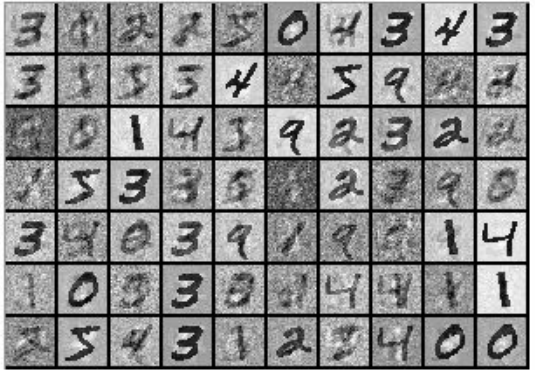
RBM Applications

- ➔ Dimensionality reduction
- ➔ Feature extraction
- ➔ Collaborative Filtering
- ➔ Classification
- ➔ Regression
- ➔ Topic modeling

Image reconstruction



Headshots, Labeled Faces in the Wild

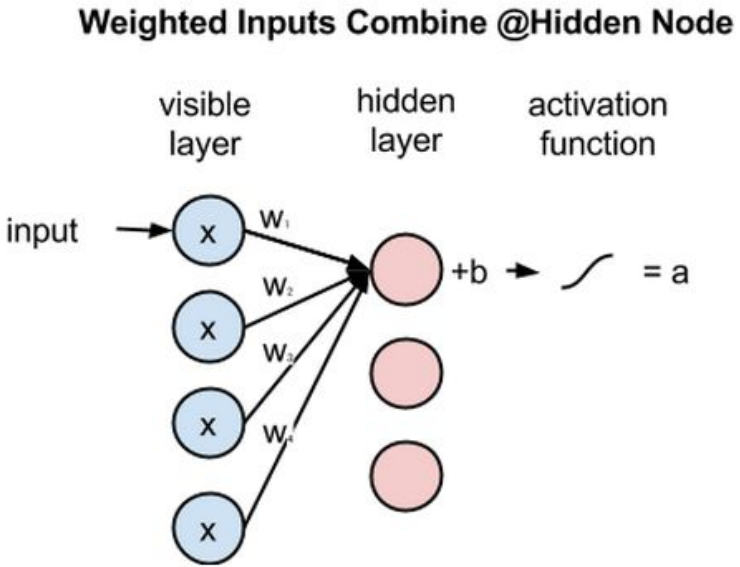


MNIST Handwritten Numerals

Images courtesy of deeplearning4j

How it works - Visible Layer

- Visible layer receives low-level input such as a sample of image pixels
- Each input (x) multiplied by a separate weight
- Products are summed (matrix multiplication) and added to a bias
- Result passed through an activation function to determine the node's output



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

Image courtesy of deeplearning4j

How it works - Hidden Layer

- At each hidden node, each input (x) is multiplied by its respective weight w
  - A single x would have three weights, making 12 weights altogether
- Weights between two layers will always form a matrix:
  - rows equal to the input nodes
  - columns equal to output nodes
- Each hidden node receives four inputs multiplied by weights; sum of products added to bias, result passed through activation function producing one output for each hidden node

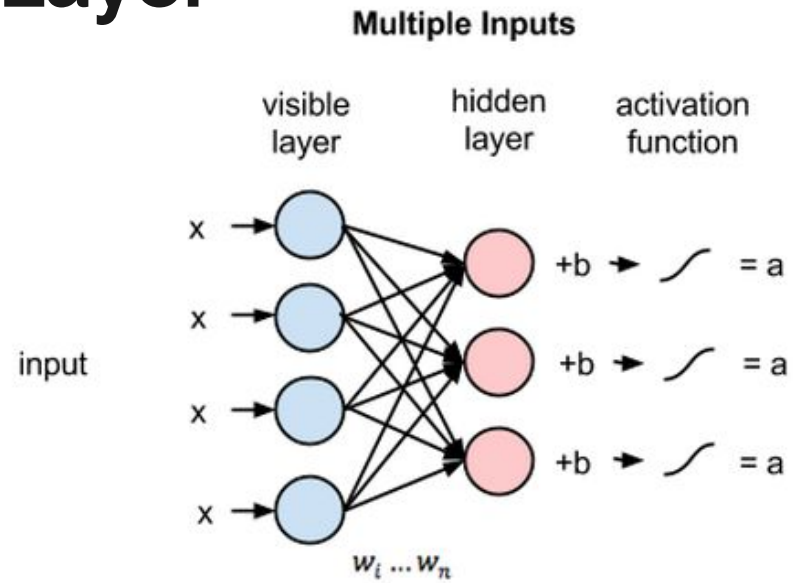


Image courtesy of deeplearning4j

How it works - Reconstructions

- Reconstruct data (unsupervised) by making several forward and backward passes between visible and hidden layers (without involving a deeper network)
- Activations of hidden layer become input in a backward pass to visible layer
- Multiplied by same weights, one per internode edge, just as x was weight-adjusted on forward pass
- Sum of product added to a visible layer bias at each visible node
- Output a reconstruction/ approximation of original input

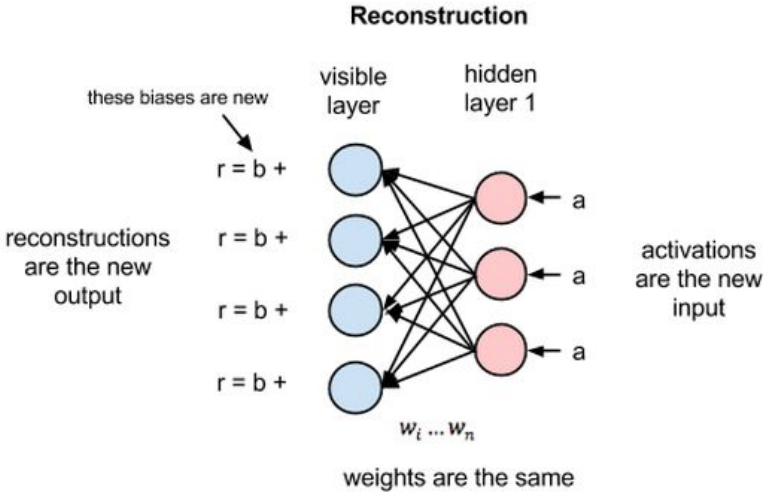


Image courtesy of deeplearning4j

How it works - Part of Network

- If two layers part of a deeper NN (such as DBN), outputs of hidden layer 1 passed as inputs to hidden layer 2, repeating until final classifying layer

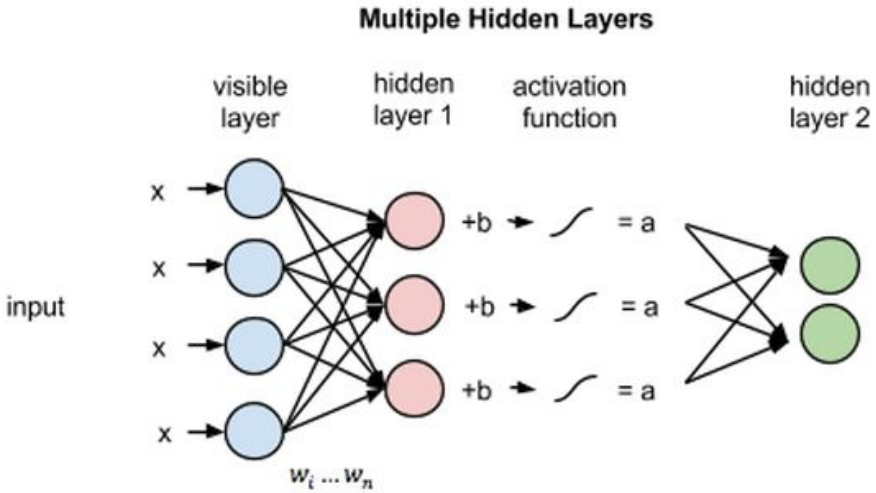


Image courtesy of deeplearning4j

Deep Belief Network

- ➔ “Introduction of DBN in 2006 began current deep learning renaissance.” -Deep Learning, Goodfellow.
- ➔ One of first nonconvolutional models to successfully train deep architectures
- ➔ Stacked RBMS; each RBM layer communicates with both previous and subsequent layers (but not laterally)
- ➔ End with:
  - ◆ Softmax to create classifier; or
  - ◆ Cluster unlabeled data (unsupervised)
- ➔ Used to recognize, cluster and generate images, video sequences and motion-capture data

Why RBMs are Important

- ➔ Unsupervised, discover a “feature-rich” representation of input more efficiently than clustering; small RBMs can capture complicated distributions
- ➔ More efficient at dimensionality reduction than PCA, producing non-linear transformations rather than PCA’s linear
- ➔ Generative; can generate samples from learned hidden representations

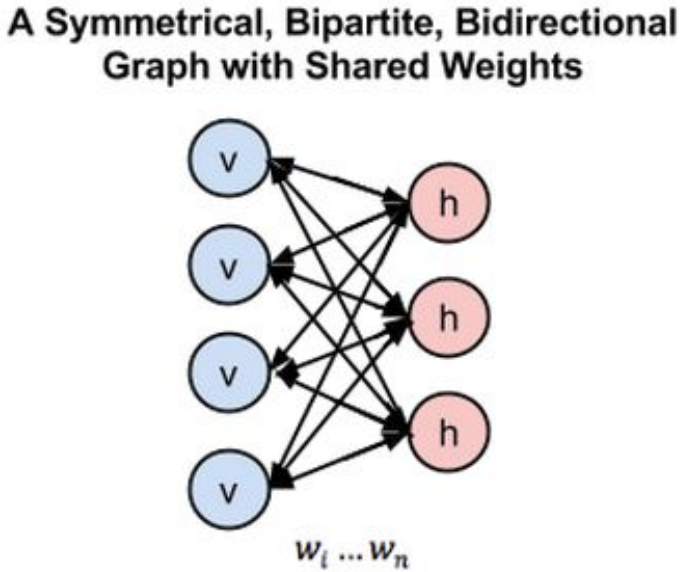


Image courtesy of deeplearning4j

Resources

Hinton, Geoffrey. “A fast learning algorithm for deep belief nets.” University of Toronto. 2006.

Original RBM/DBN whitepaper which kicked off the deep learning renaissance.

Goodfellow, Ian. “Deep Learning.” MIT Press. http://www.deeplearningbook.org/. 2016.

See Chapter 20 “Deep Generative Models” for an in-depth analysis of various RBM recipes.

“A Beginner’s Tutorial for Restricted Boltzmann Machines.” www.deeplearning4j.org.

Hinton, Geoffrey. “Neural Networks for Machine Learning.” University of Toronto. www.coursera.org.

“Boltzmann machine” and “Restricted Boltzmann machine.” www.wikipedia.org.

“Restricted Boltzmann Machines (RBM).” www.deeplearning.net.

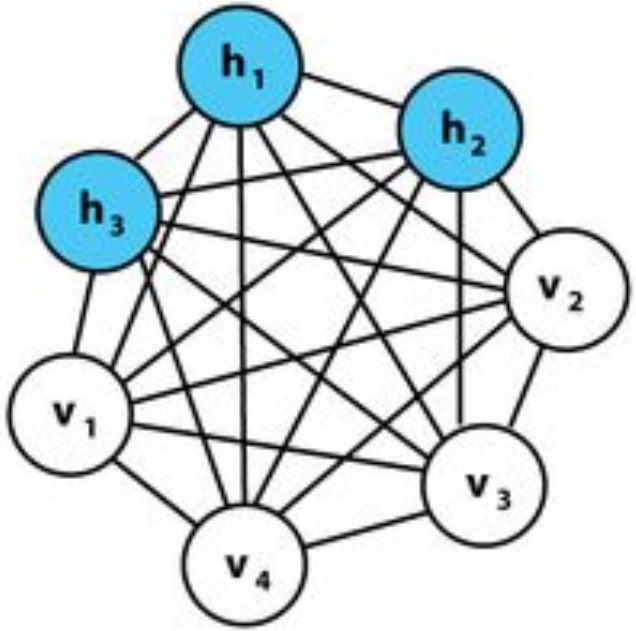
Chen, Edwin. “Introduction to Restricted Boltzmann Machines.” blog.echen.me. 18 July 2011.

Questions?

Appendix

Boltzmann Machine

- ➔ Invented by Geoffrey Hinton and Terry Sejnowski in 1985
- ➔ Unsupervised type of stochastic RNN
- ➔ One of first NNs capable of learning internal representations
- ➔ Given a training set of binary vectors, fit a model that will assign a probability to every possible binary vector
- ➔ May have connections between hidden units



Example Boltzmann machine with 3 hidden and 4 visible units.
Image courtesy of wikipedia.

RBM vs. MLP

	RBM	MLP
Layers	Two	Many
Stacked into	Deep Belief Network	Autoencoder
	Unsupervised	Supervised
	Stochastic / probabilistic Learns statistical distribution	Deterministic Learns representation of inputs
	Generative / reconstruct input	Discriminative / predict label
Layers	Two	Many
Training	Contrastive Divergence	Gradient Descent
Weight-setting	Reconstruction / Minimize error at minimum energy state	Backpropogation / least squares

Application: Movie Reviews

- RBMs perform a binary factor analysis (like/don't like) to discover latent factors that can explain the activation of movie choices
  - V: user movie preferences whose states are set
  - H: latent factors to learn
  - b: adjusts for popularities of each movie
- Consider 6 movies:
  - Latent factors to learn may be sci-fi/fantasy and Oscar winners
  - If a user (Alice) tells us her six binary preferences, the RBM will return with a probability of which latent factors activate

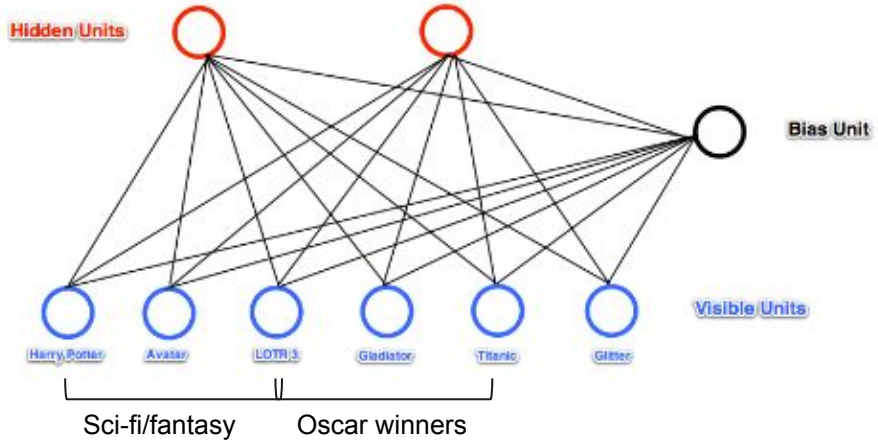


Image courtesy of Edwin Chen.

How it works - First Node

- Looking at first node:
  - At hidden layer, input (x) multiplied by a weight, added to a bias and fed into an activation function, producing the node's output

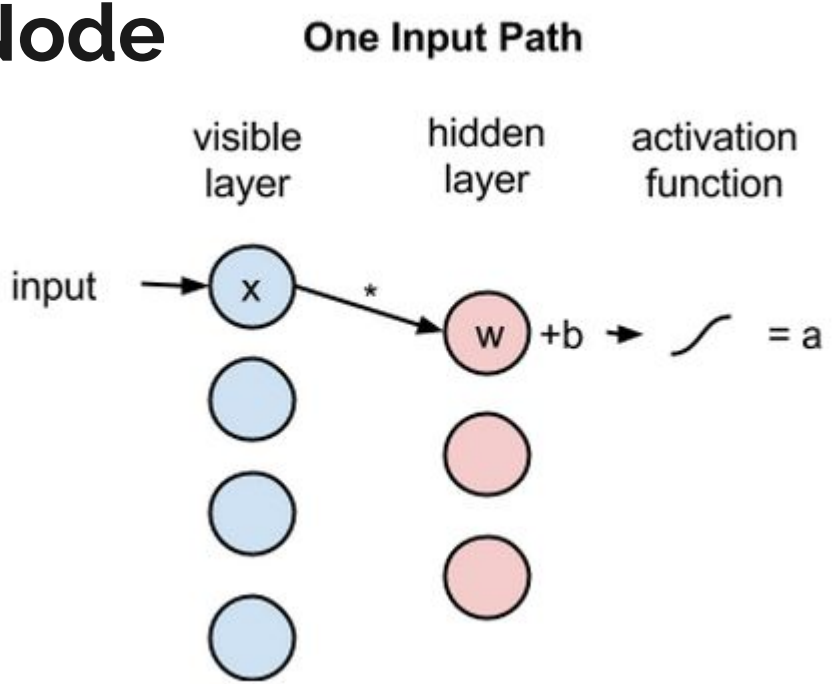


Image courtesy of deeplearning4j

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

Reconstructions (2)

- ➔ Kullback Leibler Divergence measures distance between estimated probability distribution and ground truth distribution of input
  - Measures non-overlapping/diverging areas under two curves
  - RBM's optimization algorithm attempts to minimize areas so shared weights, when multiplied by activations of h, produce a close approximation of original input

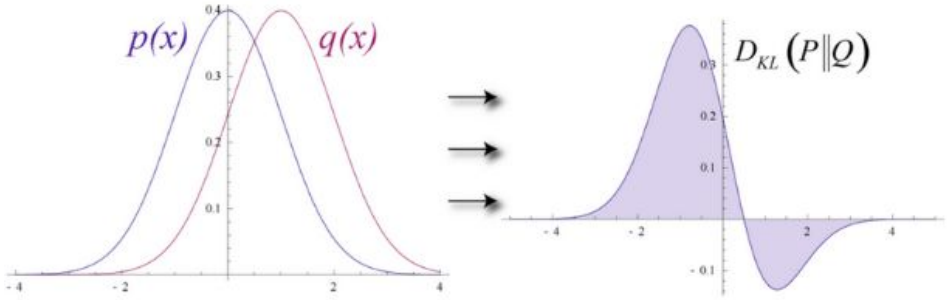


Image courtesy of deeplearning4j

Reconstructions (3)

- ➔ Because weights are randomly initialized, difference between reconstructions and original input often large
- ➔ Reconstruction error is difference between values of reconstruction and input values
- ➔ Error backpropogated against weights iteratively until error minimum reached
- ➔ On forward pass, inputs used to make predictions about node activations, or probability of output given a weighted x: p(a|x;w)
- ➔ On backward pass, when activations fed in and reconstructions (guesses about original data) are spit out, an RBM is attempting to estimate probability of inputs x given activations a

Important Characteristics

- ➔ Energy based model. Associates a scalar energy to each configuration of the variables of interest.
- ➔ Trained via contrastive divergence. “Approximate gradient descent.” - Edwin Chan
- ➔ Key component of Deep Belief Network.
- ➔ Trained via Gibbs sampling.