Restricted Boltzmann Machines and Deep Learning

Preston Engstrom

March 22, 2016

Table of contents

Motivations for Deep Learning

Applying a broader model of how the human brain works to extract features and recognize data

- Your brain has around 10^{14} neural connections
- Those connections must be fit in about 10^{9} seconds
- Labels cannot provide enough information

Issues With Classical Methods (Back Propagation)

- Requires Labeled Data (Hard to find, hard to generate)
- The learning time is very slow with multiple hidden layers
- The problem of initialization:

"It is necessary to choose initial random values for all the weights. If these values are small, it is very difficult to learn deep networks because the gradients decrease multiplicatively as we backpropagate through each hidden layer. If the initial values are large, we have randomly chosen a particular region of the weight-space and we may well become trapped in a poor local optimum within this region."

Hinton[1]

A Generative Model for Learning

- Keep the efficiency of using a gradient method for adjusting model weights, but model the structure of the input.
- Rather than develop a model for $P(label \| object)$, Calculate P(object)
  - if you want to do computer vision, learn computer graphics.
- We can then recognize an object and apply labels to data after training has completed.

Boltzmann Machines

- A Boltzmann Machine (BM) is a parameterized generative model representing a probability distribution.
- BM's extract complex aspects (features) of a distribution based on samples of that distribution.
- A BM is constructed of a hidden layer and a visible layer of nodes, interconnected by weights.
- Training is done by adjusting the weights such that the probability of distribution represented by the BM fits the training data

Binary Stochastic Neurons

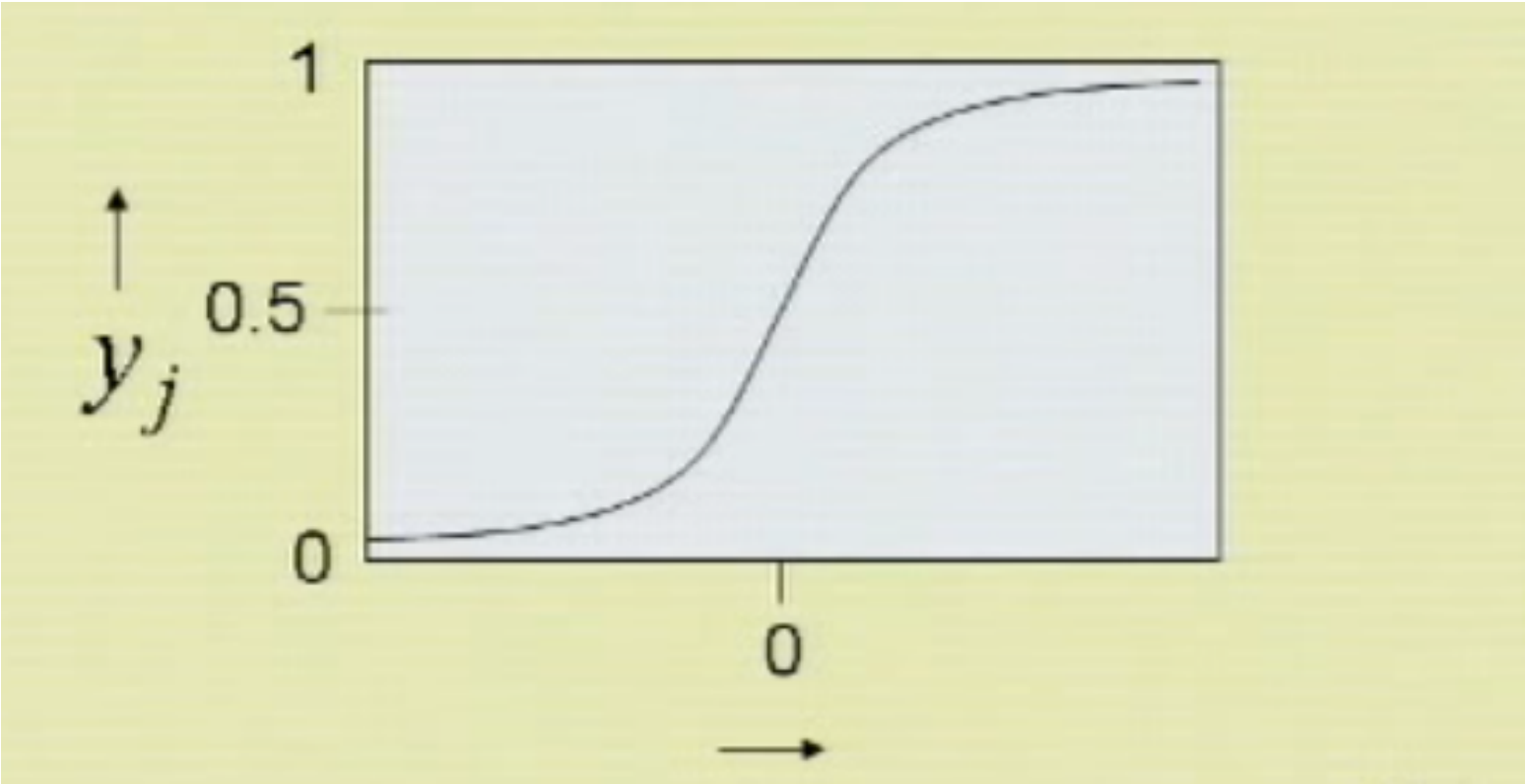


Figure: input to neuron j = external input $+ \sum y_i w_{ij}$

Boltzmann Machine Graph

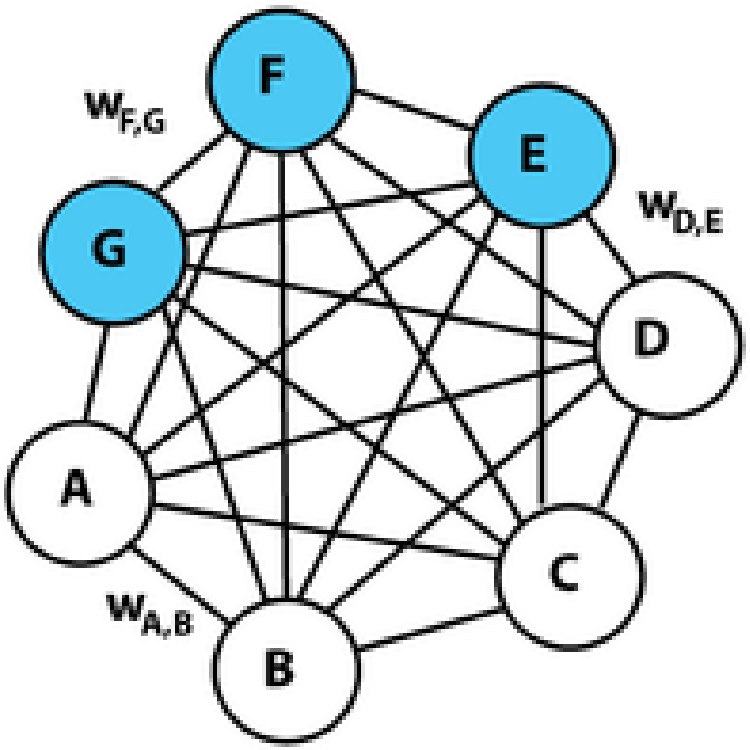


Figure: A Boltzmann Machine

A Simplified Boltzmann Machine As A Learning Module

The issue with Boltzmann Machines: Learning takes a long time

A Solution

We can restrict the topology of the network to speed training, and create useful properties of the machine. Represent the machine as a bipartite graph, where the visible layer and hidden layer only connect between each other, and never to themselves.

A Simplified Boltzmann Machine As A Learning Module

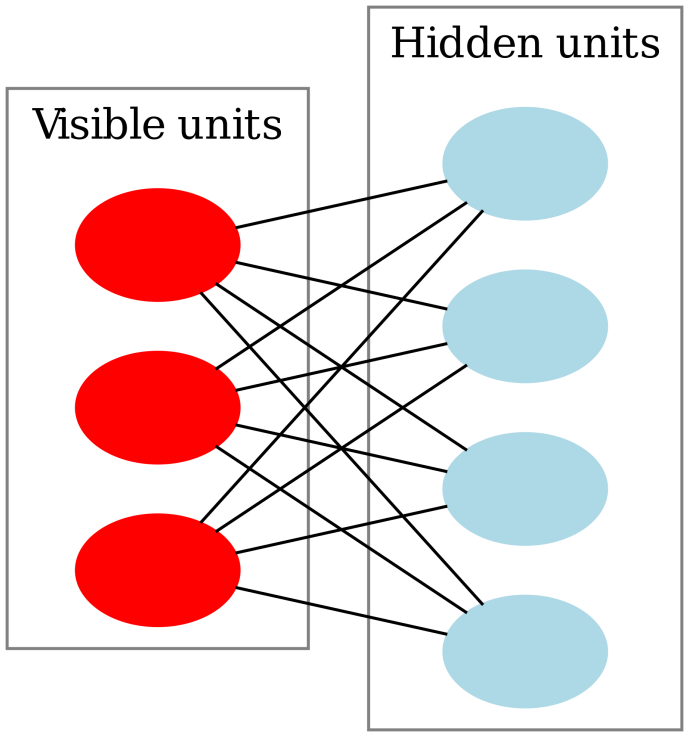


Figure: A Restricted Boltzmann Machine

A Simplified Boltzmann Machine As A Learning Module

- One layer of hidden units, meaning one layer of features?
- No connections between hidden units, visible restriction may be lifted later.

Independence of Hidden States

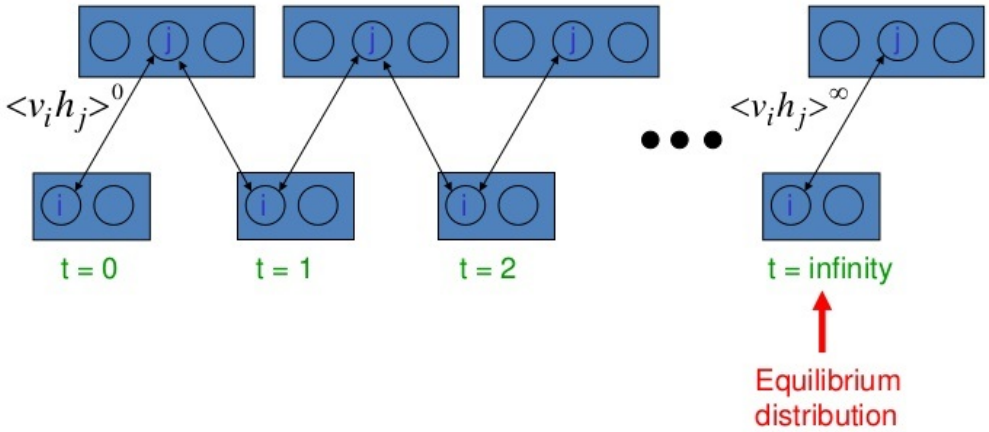
The hidden units are independent given the visible states, so sampling the posterior over "causes" of data vector are quick and unbiased.

The Energy Function

- RBMs are governed by an energy function based on natural systems. Energy is lost from a system as it seeks a stable state.
- The energy is determined by the weights and biases.
- The energy of a connection from visible to hidden units determines the probability the network will choose a particular configuration.
- The weights determine the energies linearly. The probabilities are an exponential function of the weights. Thus the log-probabilities are a linear function of the weights.

The Maximum Likelihood Learning Algorithm

A picture of the maximum likelihood learning algorithm for an RBM



Slide Credit: Geoff Hinton

$$\frac{\delta \log P(v)}{\delta w_{ij}} =< v_i h_j >^0 - < v_i h_j >^\infty \quad (1)$$

Hinton's Improved Algorithm

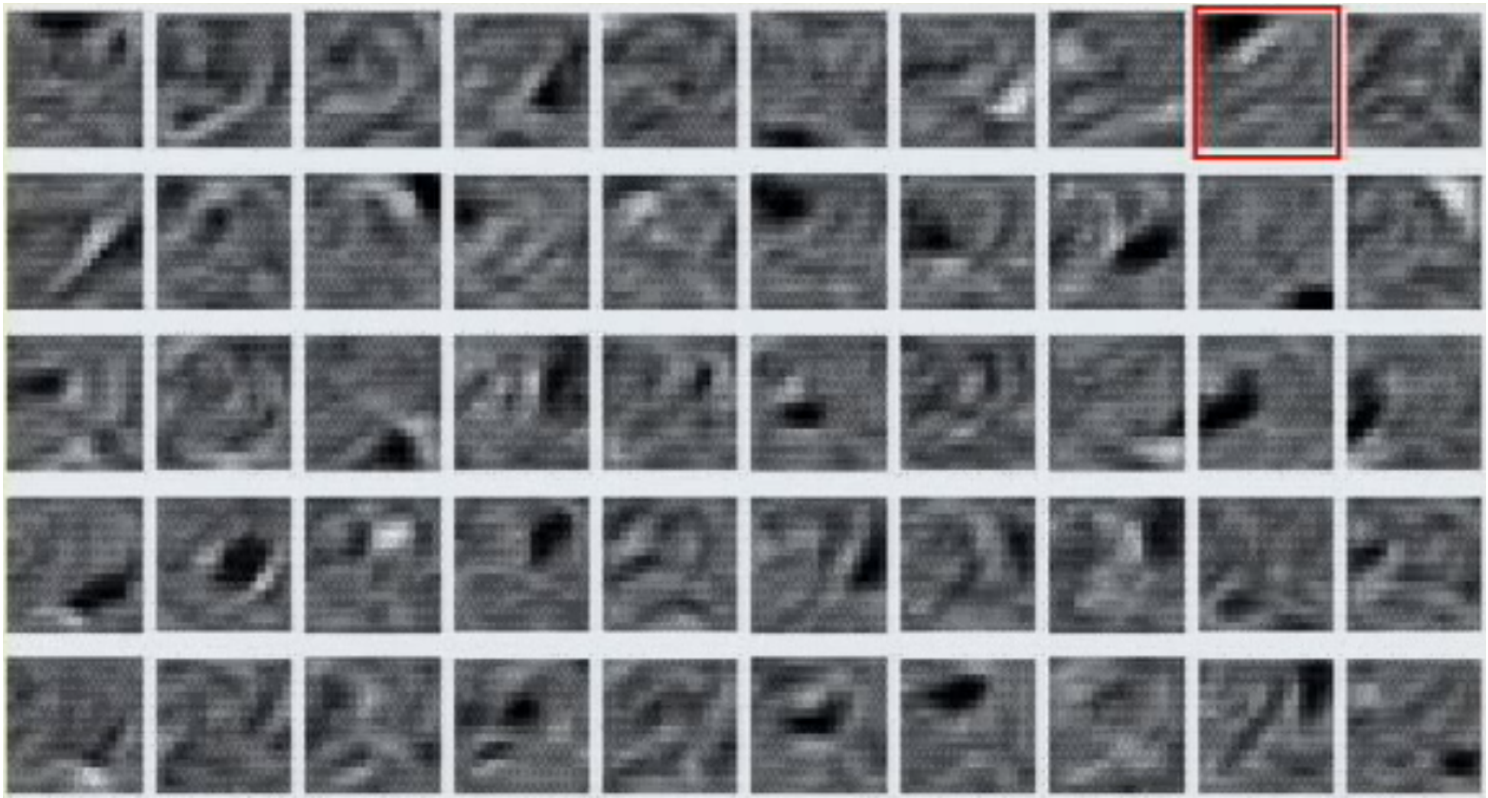
- Start with a training vector on the visible units.
- Update the hidden units, Increment weights from an active pixel to an active feature.
- Create a reconstruction on the visible units.
- Update the hidden units again Decrement weights from an active pixel to an active feature.

$$\Delta w_{ij} = \epsilon(< v_i h_j >^0 - < v_i h_j >^1) \quad (2)$$

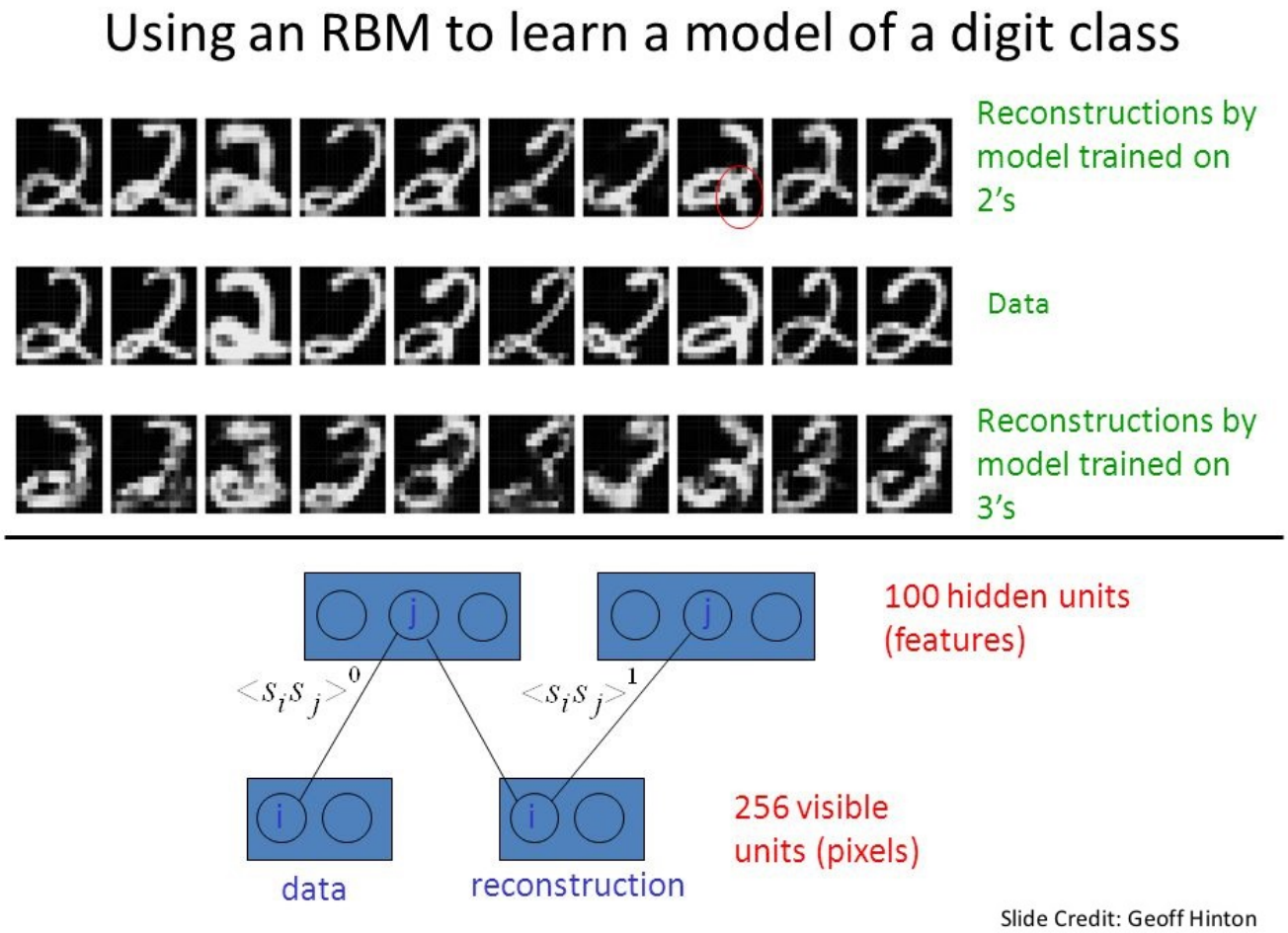
An Example Using Images Of The Digit 2

- Data: 16 by 16 pixel image (256 visible nodes)
- 50 Binary Feature Neurons
- IMPORTANT: Increment connections on the data, decrement them on the reconstruction.

The Maximum Likelihood Learning Algorithm



Visualization Of Features On The Digit 2



Layers Of Features: Training A Deep Network

- Begin by training the first layer as with a single RBM.
- Treat the activations of the trained features as the data vector for the next hidden layer.
- It can be proved that each time a layer is added, we improve our model of the training set.

The proof uses variational free energy, a method used in physics for analyzing complicated non-equilibrium systems.

Layers Of Features: Training A Deep Network