

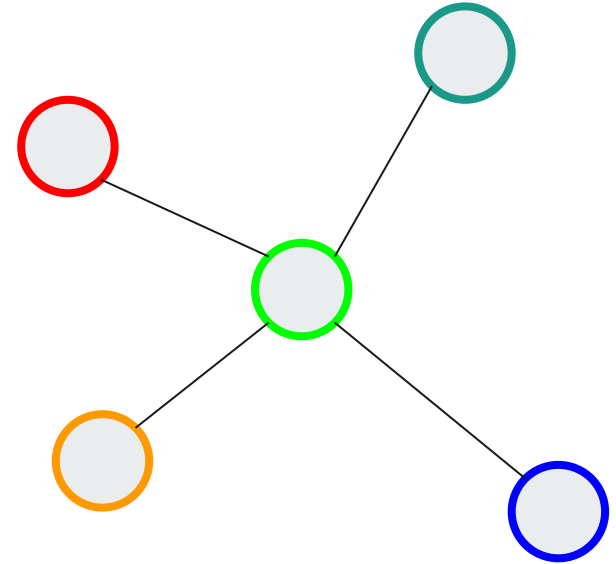
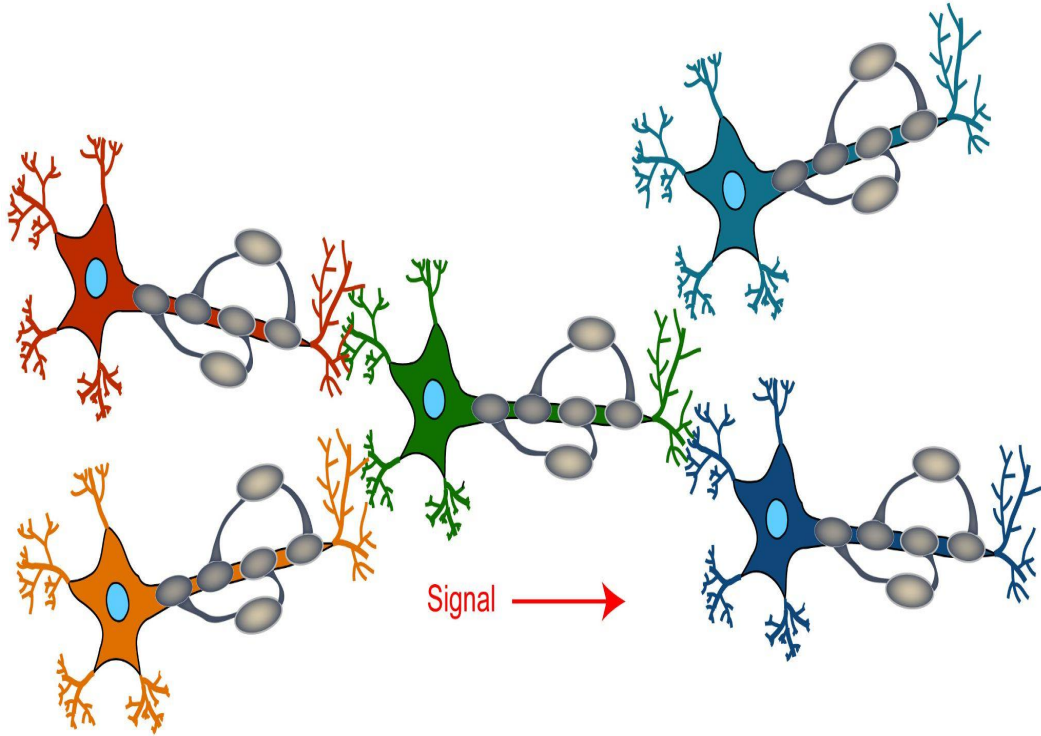


What kind of a graphical model is the brain?

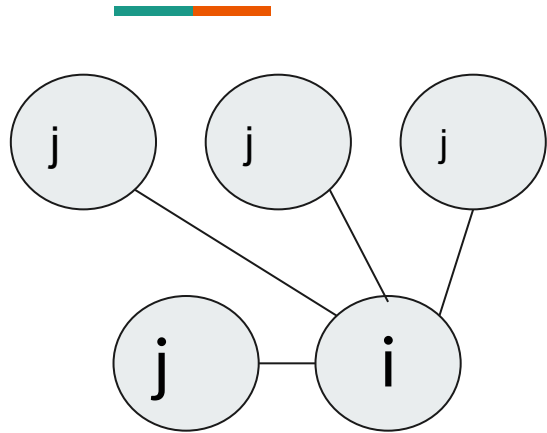
Author - Geoffrey Hinton

Presentor - Vrushali Pandit B20BB047

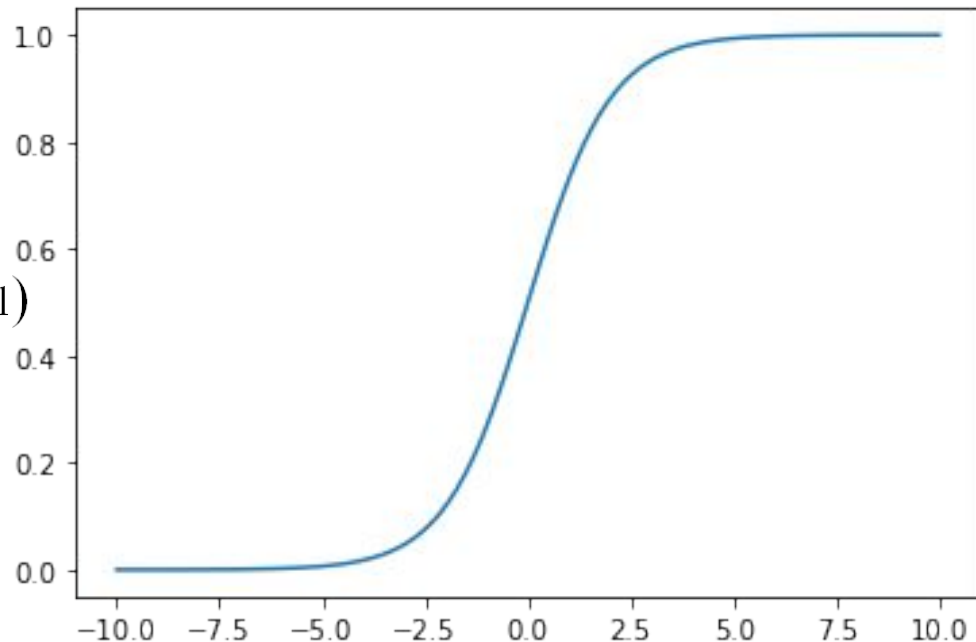
From neuron to nodes



Nodes - stochastic binary units



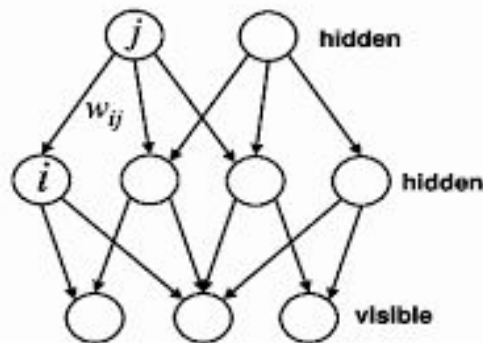
$$p(s_i = 1)$$



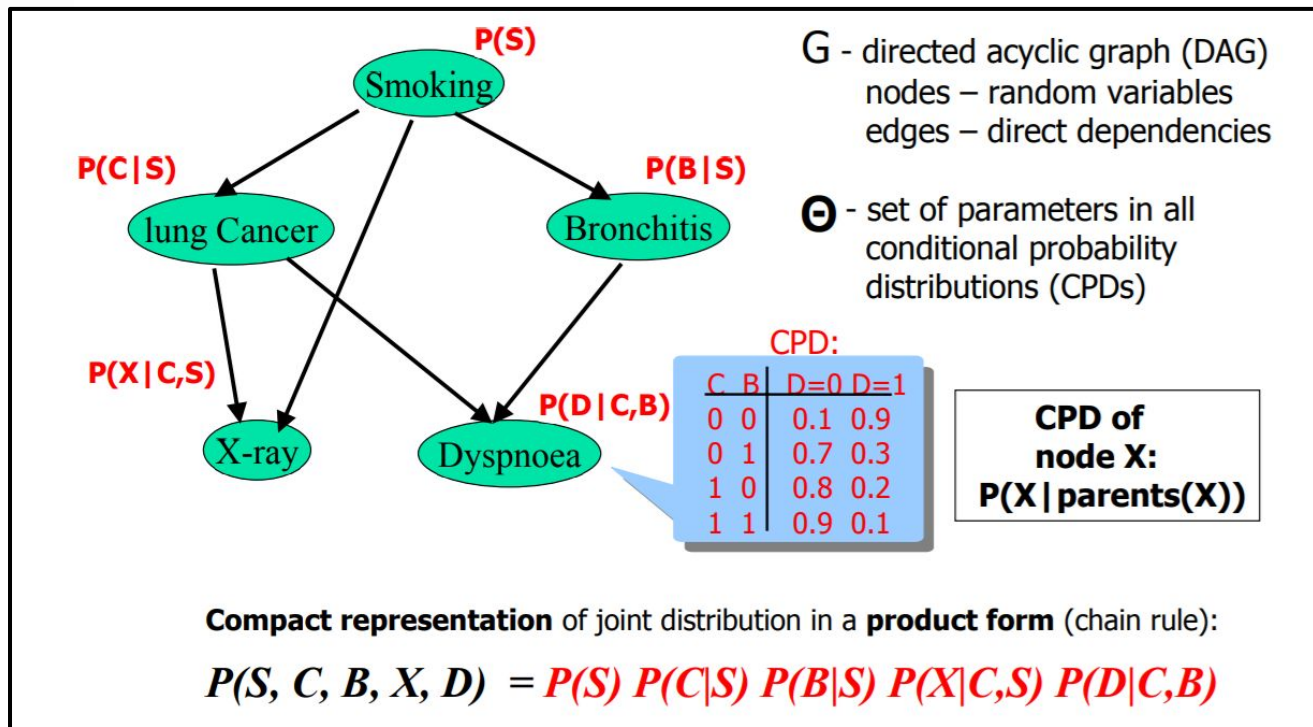
$$p(s_i = 1) = \frac{1}{1 + \exp(-b_i - \sum_j s_j w_{ji})}$$

$$b_i + \sum_j s_j w_{ji}$$

Sigmoid Belief Networks - Directed Model



- Nodes are rv's and weights are conditional probabilities
- Also called a bayesian network



Advantages:

- Interpretable
- Local Markov property

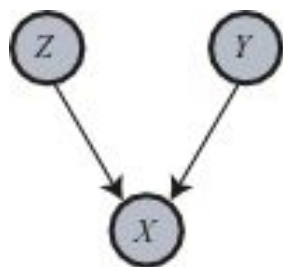
Problems



No efficient learning algorithm for Sigmoid Belief Net

"Explaining Away" directed graph

Explaining Away



X -> car engine fails

Y -> dead battery

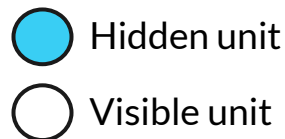
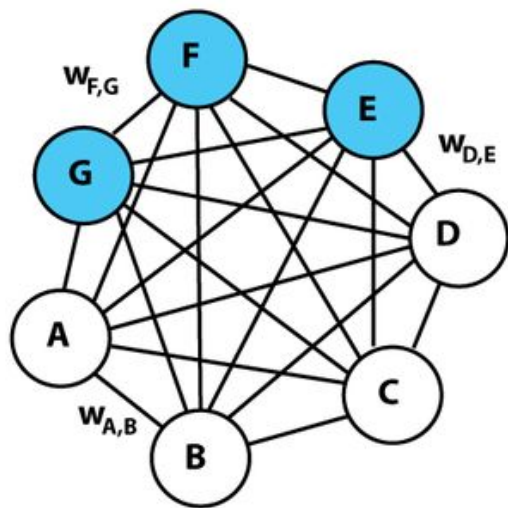
Z -> blocked fuel pump

Let's say the engine works with a probability of 90%

X	Y	Z	P(X,Y,Z)	comment
0	0	0	0.9	Engine works
0	0	1	0	impossible - blocked fuel pump and working engine
0	1	0	0	impossible - dead battery and working engine
0	1	1	0	impossible - dead battery, blocked fuel pump and working engine
1	0	0	0	impossible - a cause we don't have in our graph
1	0	1	a	
1	1	0	b	
1	1	1	c	

$$P(Y = 0, Z = 0 | X = 1) = P(Y = 0 | X = 1) \cdot P(Z = 0 | X = 1)$$

Boltzmann Machine - Undirected Model



- Undirected, fully-connected, generative, unsupervised
 - Non causal
 - No output nodes, instead the model learns what's a normal configuration of the entire model
 - Can predict an abnormal configuration before it is reached
-
- The data is clamped to the visible units
 - Weights and biases are updated so as to drive the energy function to equilibrium

$$-E(\mathbf{v}, \mathbf{h}) = \sum_{i \in \text{vis}} v_i b_i + \sum_{k \in \text{hid}} h_k b_k + \sum_{i < j} v_i v_j w_{ij} + \sum_{i, k} v_i h_k w_{ik} + \sum_{k < l} h_k h_l w_{kl}$$

Boltzmann Machine (learning)



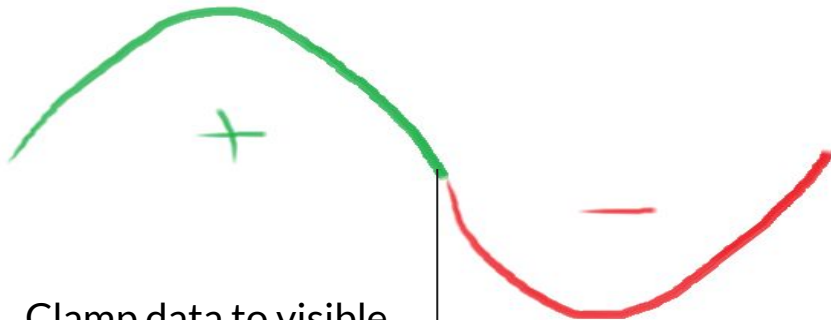
The probability of a particular configuration $BM(v, h)$ is calculated as $\exp(\text{its energy})$ upon such terms for all the possible configurations.

$$p(v, h) = \frac{e^{-E(v, h)}}{\sum_{u, g} e^{-E(u, g)}}$$

For a particular kind of data, the number of visible nodes will be fixed. Hence we can sum up over all the possible configurations of the hidden nodes in the numerator.

$$p(v) = \frac{\sum_h e^{-E(v, h)}}{\sum_{u, g} e^{-E(u, g)}}$$

Boltzmann Machine (learning)



1. Clamp data to visible units
2. Randomly choose hidden units
3. Update them using the stochastic binary unit definition
4. Continue to thermal equilibrium
5. Sample units ij
6. $\langle s_i s_j \rangle^+$ is the correlation b/w them

1. Repeat the same with data vectors unclamped
2. $\langle s_i s_j \rangle^-$ is the correlation b/w them

$$\Delta w_{ij} = \epsilon (\langle s_i s_j \rangle^+ - \langle s_i s_j \rangle^-)$$

Update Rule

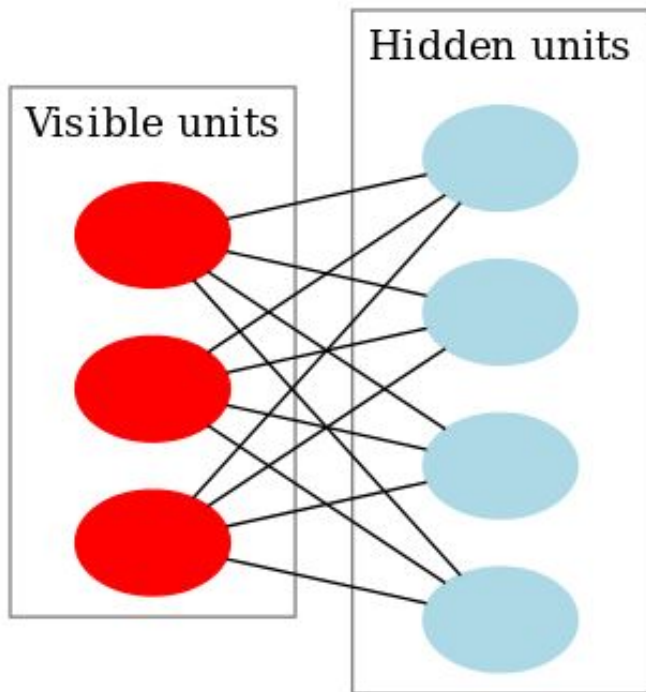
Problems



No efficient learning algorithm for Sigmoid Belief Net
"Explaining Away" Directed Graph
Take too long to reach thermal equilibrium Boltzmann Machine

Restricted Boltzmann Machine

Tries to solve the limitation of a usual BM



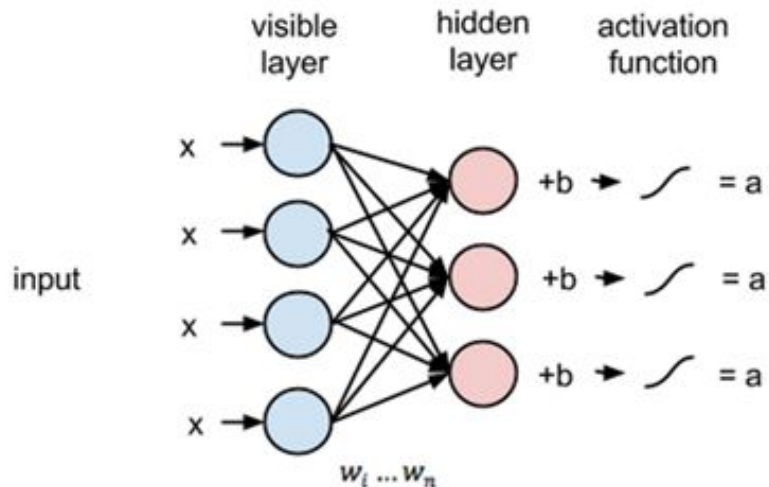
- Only one layer of hidden units
- All hidden units independent
- All visible units independent

How is this helpful?

- In the positive phase, earlier we had to randomly select hidden units and update them to reduce bias, but now because they depend only on the visible units, all of them can be updated in parallel in one go.
- Making the visible units independent has a similar effect.

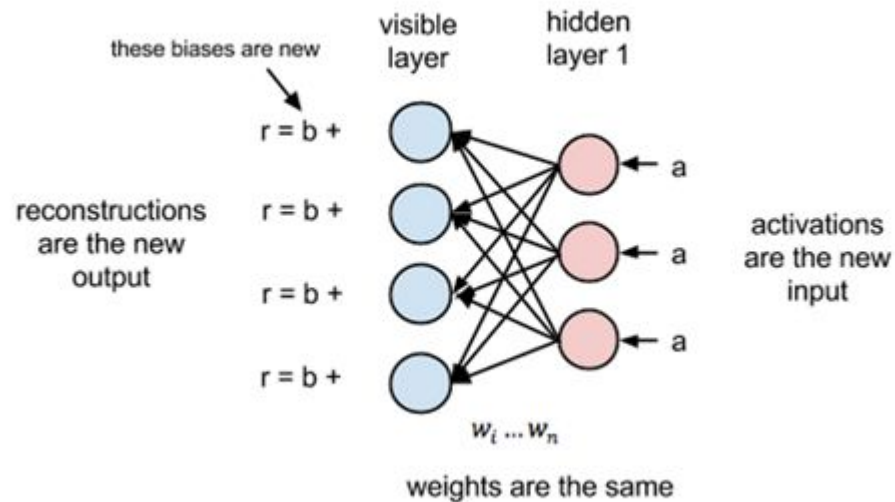
Learning an RBM

Multiple Inputs



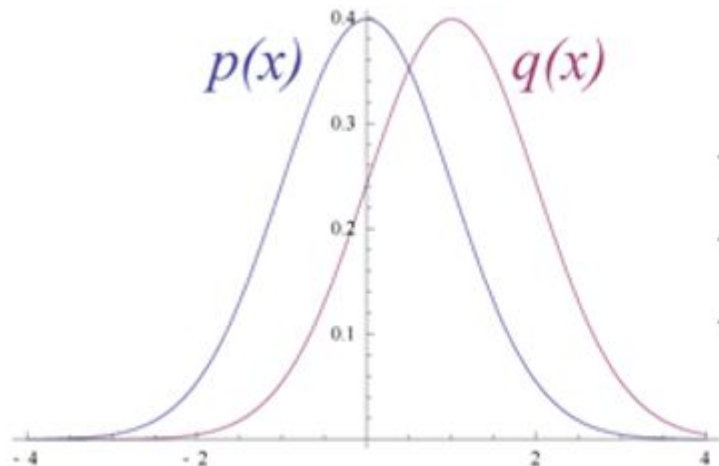
Usual forward pass using hidden layer bias terms
Gives **v0**

Reconstruction

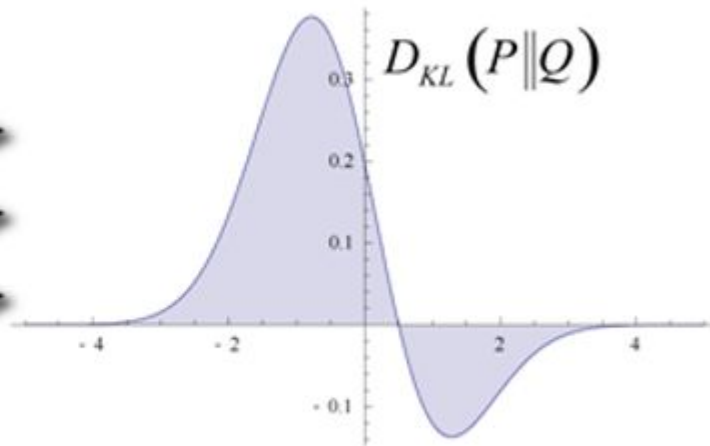
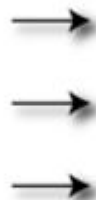


Use **v0** to reconstruct the same input using visible
biases
Gives **v1**

Learning an RBM



Original Gaussian PDF's



KL Area to be Integrated

Assume that we have two normal distributions, one from the input data (denoted by $p(x)$) and one from the reconstructed input approximation (denoted by $q(x)$). The difference between these two distributions is our error in the graphical sense and our goal is to minimize it, i.e., bring the graphs as close as possible

Problems

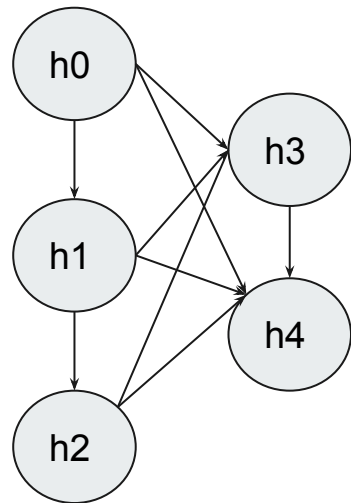


No efficient learning algorithm for Sigmoid Belief Net

"Explaining Away" Directed Graph

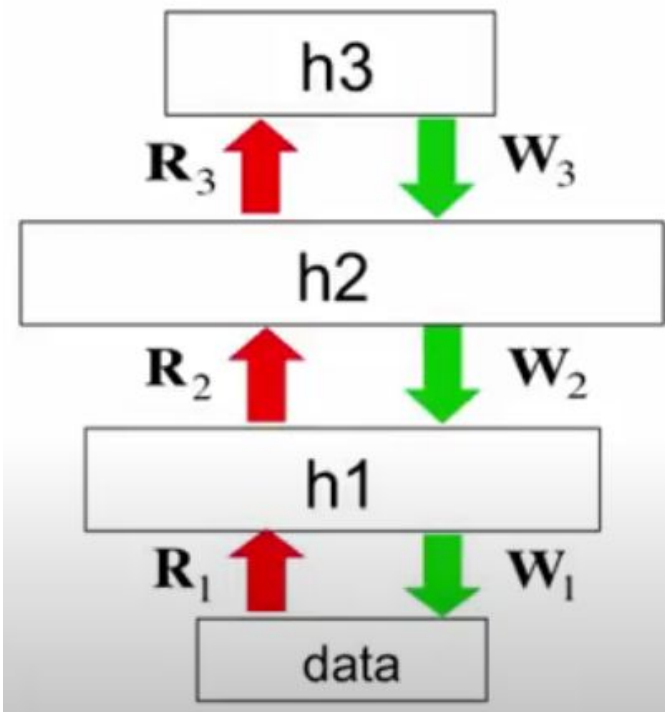
~~Take too long to reach thermal equilibrium Boltzmann Machine~~

Wake-sleep Algorithm



$$\begin{aligned} P(\text{all data used for training}) &= P(L0) * P(L1) \\ &= P(h0)P(h1)P(h2)P(h3)P(h4) \end{aligned}$$

Wake-sleep Algorithm



WAKE PHASE:

- Use **recognition** weights to perform forward pass
- Maximum likelihood learning of the **generative** weights
- Draw samples and move to sleep phase.

SLEEP PHASE:

- Do the exact opposite
- Use the **generative** weights to perform backward pass
- And learn **recognition** weights

Problems



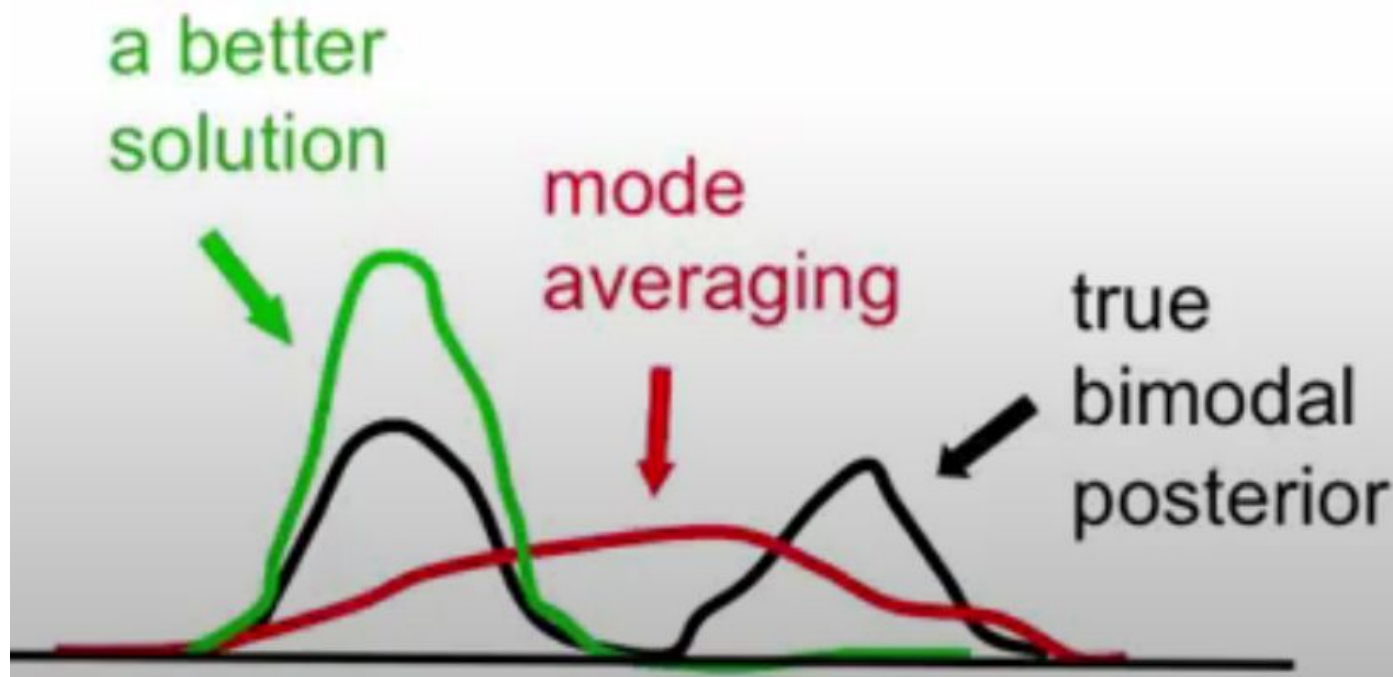
No efficient learning algorithm for ~~Sigmoid Belief Net~~


"Explaining Away" ~~Directed Graph~~ (not really)

Take too long to reach thermal equilibrium ~~Boltzmann Machine~~

Mode averaging ~~wake-sleep algorithm~~

Mode averaging



- 
- The paper further introduces a new hybrid generative model. The top two layers of this model form an undirected graph (a restricted boltzmann machine) which can extract associative memory from the data. This memory is used by a directed cyclic graph (a sigmoid belief network) to convert it to observables such as pixels of an image.
 - This model provides 1.25% of an error rate on MNIST dataset.
 - The only discriminative model which provides a lower error rate than this hybrid one is SVM.
 - But this lower error rate is obtained by doing stuff like weight-sharing (reduced the time of computation because most of the data is similar and learning similar weights again and again might be redundant. And sub-sampling(