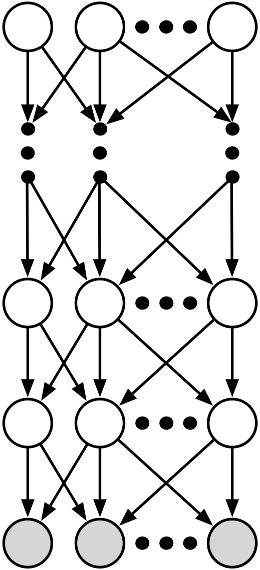
Setting the Stage: Complementary Priors and Variational Bounds

Yee Whye Teh[Gatsby Unit, UCL]
Geoffrey E. Hinton[Toronto]
Simon Osindero[Toronto]

December 6, 2007
Deep Learning Workshop
NIPS

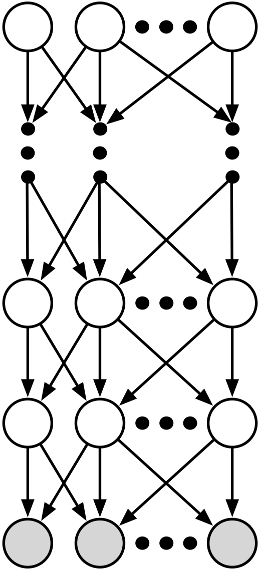
Deep Belief Networks

- Say we have a layered directed graphical model.
- Can we do efficient inference in this model?
- Just from the structure of the graphical model: no.

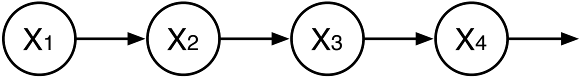


Deep Belief Networks

- Say we have a layered directed graphical model.
- Can we do efficient inference in this model?
- Just from the structure of the graphical model: no.
- But perhaps there are settings of the conditional probabilities in the model allowing for efficient inference…



Markov Chains



- A **Markov chain** is a sequence of variables $X_1, X_2, \ldots$ with the Markov property

$$p(X_t | X_1, \ldots, X_{t-1}) = p(X_t | X_{t-1})$$

- A Markov chain is **stationary** if the transition probabilities do not depend on time

$$p(X_t = x' | X_{t-1} = x) = T(x \rightarrow x')$$

  $T(x \rightarrow x')$ is called the **transition matrix.**

- If a Markov chain is **ergodic** it has a unique equilibrium distribution

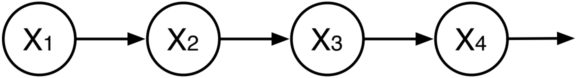
$$p_t(X_t = x) \rightarrow p_\infty(X = x) \quad \text{as } t \rightarrow \infty$$

Markov Chains

- Most Markov chains used in practice satisfy **detailed balance**

$$p_\infty(X)T(X \to X') = p_\infty(X')T(X' \to X)$$

  e.g. Gibbs, Metropolis-Hastings, slice sampling...
- Such Markov chains are **reversible**



$$p_\infty(X_1)T(X_1 \to X_2)T(X_2 \to X_3)T(X_3 \to X_4)$$

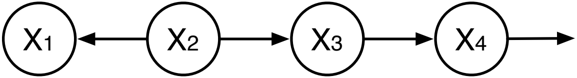
Markov Chains

- Most Markov chains used in practice satisfy **detailed balance**

$$p_\infty(X)T(X \to X') = p_\infty(X')T(X' \to X)$$

e.g. Gibbs, Metropolis-Hastings, slice sampling...

- Such Markov chains are **reversible**



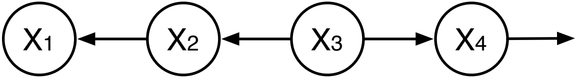
$$T(X_1 \leftarrow X_2)p_\infty(X_2)T(X_2 \to X_3)T(X_3 \to X_4)$$

Markov Chains

- Most Markov chains used in practice satisfy **detailed balance**

$$p_\infty(X)T(X \to X') = p_\infty(X')T(X' \to X)$$

  e.g. Gibbs, Metropolis-Hastings, slice sampling...
- Such Markov chains are **reversible**



$$T(X_1 \leftarrow X_2)T(X_2 \leftarrow X_3)p_\infty(X_3)T(X_3 \to X_4)$$

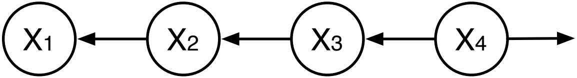
Markov Chains

- Most Markov chains used in practice satisfy **detailed balance**

$$p_\infty(X)T(X \to X') = p_\infty(X')T(X' \to X)$$

e.g. Gibbs, Metropolis-Hastings, slice sampling…

- Such Markov chains are **reversible**



$$T(X_1 \leftarrow X_2)T(X_2 \leftarrow X_3)T(X_3 \leftarrow X_4)p_\infty(X_4)$$

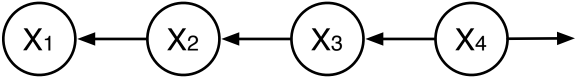
Markov Chains

- Most Markov chains used in practice satisfy **detailed balance**

$$p_\infty(X)T(X \to X') = p_\infty(X')T(X' \to X)$$

  e.g. Gibbs, Metropolis-Hastings, slice sampling. . .

- Such Markov chains are **reversible**

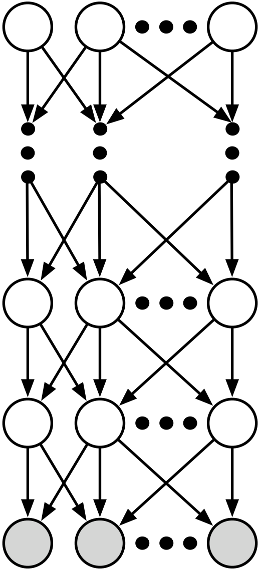


$$T(X_1 \leftarrow X_2)T(X_2 \leftarrow X_3)T(X_3 \leftarrow X_4)p_\infty(X_4)$$

- This is the basic idea of **complementary priors.**

Complementary Priors

- Say we have a layered directed graphical model.
- Can we do efficient inference in this model?

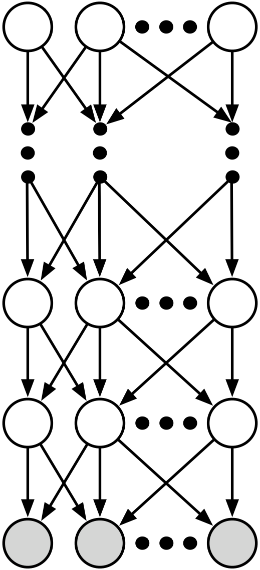


Complementary Priors

- Say we have a layered directed graphical model.
- Can we do efficient inference in this model?
- Consider the following conditional probabilities:

$$p(X_L) = p_\infty(X_L)$$
$$p(X_i|X_{i+1}) = T(X_{i+1} \to X_i) \quad \text{for } i = 1 \dots L$$

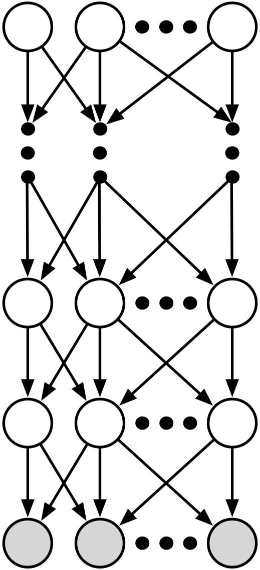
  Note: X_i is a vector of variables in layer i.
- This is just the Markov chain unrolled.
- Detailed balance and the time reversal of the Markov chain comes to our rescue!



Complementary Priors

- We can reverse the arcs in the model:

$$p(X_1 \ldots, X_L) = p_\infty(X_L) \prod_{i=L-1}^{1} T(X_{i+1} \to X_i)$$

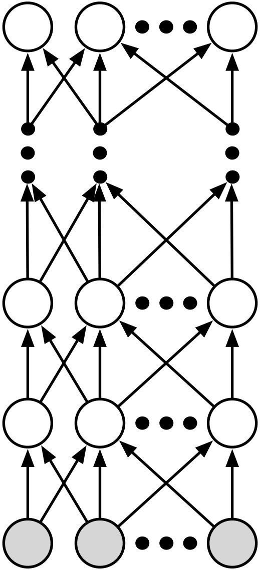


Complementary Priors

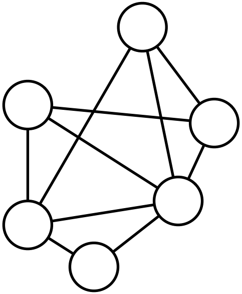
- We can reverse the arcs in the model:

$$
\begin{aligned}
p(X_1 \ldots, X_L) =& p_\infty(X_L) \prod_{i=L-1}^{1} T(X_{i+1} \rightarrow X_i) \\
=& p_\infty(X_1) \prod_{i=2}^{L} T(X_i \rightarrow X_{i+1})
\end{aligned}
$$

- Now inference is trivial!
- To obtain a sample from the posterior given observations we just run the Markov chain upwards.
- The complementary prior is simply the equilibrium distribution of the Markov chain.



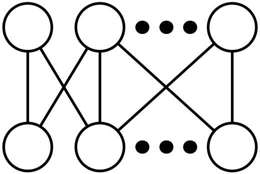
Boltzmann Machines



- A **Boltzmann machine** is a pairwise Markov random field with binary variables

$$p_{BM}(x_1 \ldots x_n) = \frac{1}{Z} e^{\sum_{ij} W_{ij} x_i x_j + \sum_i b_i x_i}$$

- It is an exponential family with natural parameters $\{W_{ij}, b_i\}$, and sufficient statistics $\{E[x_i x_j], E[x_i]\}$ for all i, j.



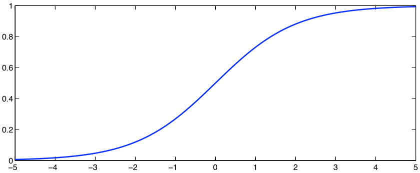
Boltzmann Machines

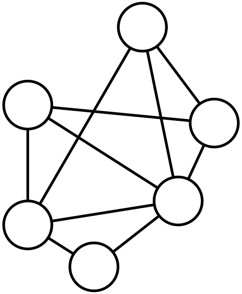
$$p_{BM}(x_1 \dots x_n) = \frac{1}{Z} e^{\sum_{ij} W_{ij} x_i x_j + \sum_i b_i x_i}$$

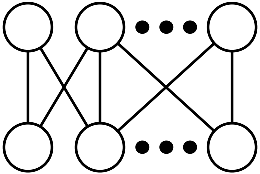
- **Gibbs sampling** in a Boltzmann machine:

$$p(x_i = 1 | x_{\neg i}) = \sigma\left(\sum_j W_{ij} x_j + b_i\right)$$

$$\sigma(y) = \frac{1}{1+\exp(-y)}$$



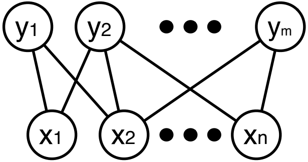




Restricted Boltzmann Machines

$$p_{RBM}(x_{1:n}, y_{1:m}) = \frac{1}{Z} e^{\sum_{ij} W_{ij} x_i y_j + \sum_i b_i x_i + \sum_j c_j y_j}$$

- A **Restricted Boltzmann machine** (RBM) is simply a Boltzmann machine with a bipartite structure.
- In an RBM we can do **blocked Gibbs** sampling, alternating between the layers.



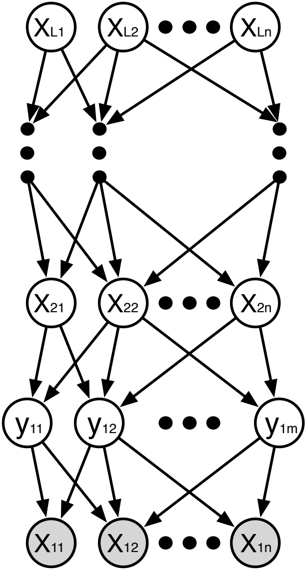
$$p(x_: = 1 | y_:) = \sigma\left(W y_: + b\right)$$
$$p(y_: = 1 | x_:) = \sigma\left(W^\top x_: + c\right)$$

Sigmoid Belief Networks

- We use blocked Gibbs in an RBM as our Markov chain to define a directed graphical model, and use the RBM for the top layer of variables[1],

$$
\begin{aligned}
p(X_{L1} \ldots X_{Ln}) =& p_{RBM}(x_{L1} \ldots X_{Ln}) \\
p(y_{k:} = 1 | x_{k+1\,:}) =& \sigma\left(W^{\top} x_{k+1\,:} + c\right) \\
p(x_{k:} = 1 | y_{k:}) =& \sigma\left(W y_{k:} + b\right)
\end{aligned}
$$

- This is a **sigmoid belief network** with tied parameters.



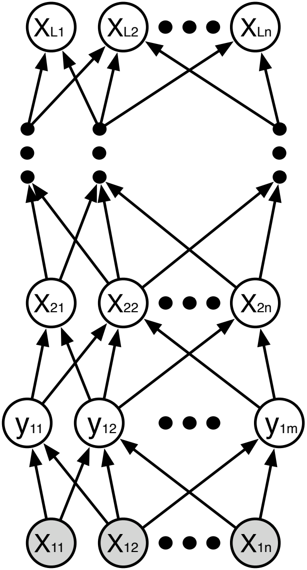
[1] Because of the bipartite structure of the RBM the layers alternate between the x's and y's, but the unrolling and complementary prior argument still holds.

Sigmoid Belief Networks

- We use blocked Gibbs in an RBM as our Markov chain to define a directed graphical model, and use the RBM for the top layer of variables[1],

$$\begin{aligned} p(X_{L1} \dots X_{Ln}) &= p_{RBM}(x_{L1} \dots X_{Ln}) \\ p(y_{k:} = 1 | x_{k+1\,:}) &= \sigma\left(W^\top x_{k+1\,:} + c\right) \\ p(x_{k:} = 1 | y_{k:}) &= \sigma\left(W y_{k:} + b\right) \end{aligned}$$

- This is a **sigmoid belief network** with tied parameters.
- Inference just involves reversing all the arcs.



[1] Because of the bipartite structure of the RBM the layers alternate between the x's and y's, but the unrolling and complementary prior argument still holds.

Stagewise Variational Bound

- Say we trained a RBM on a dataset $\{x^{(1)}, \ldots, x^{(D)}\}$, obtaining a set of weights W_{train} (also includes the biases).
- The variational lower bound is exact when $q(y|x) = p(y|x)$:

$$
\begin{aligned}
&\log p(x) \\
=&E_{\log q(y|x)} [\log p(x, y) - \log q(y|x)]
\end{aligned}
$$

Stagewise Variational Bound

- Say we trained a RBM on a dataset $\{x^{(1)}, \ldots, x^{(D)}\}$, obtaining a set of weights W_{train} (also includes the biases).
- The variational lower bound is exact when $q(y|x) = p(y|x)$:

$$
\begin{aligned}
&\log p(x) \\
=&E_{\log q(y|x)} \left[\log p(x, y) - \log q(y|x)\right] \\
=&E_{\log q(y|x)} \left[\log p(y) + \log p(x|y) - \log q(y|x)\right]
\end{aligned}
$$

Stagewise Variational Bound

- Say we trained a RBM on a dataset $\{x^{(1)}, \ldots, x^{(D)}\}$, obtaining a set of weights W_{train} (also includes the biases).
- The variational lower bound is exact when $q(y|x) = p(y|x)$:

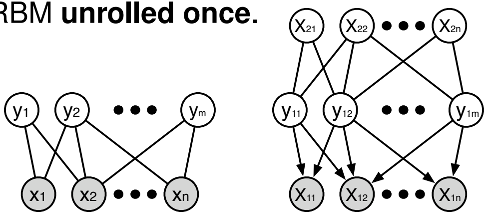
$$
\begin{aligned}
&\log p(x) \\
=&E_{\log q(y|x)}\left[\log p(x, y) - \log q(y|x)\right] \\
=&E_{\log q(y|x)}\left[\log p(y) + \log p(x|y) - \log q(y|x)\right] \\
=&E_{\log q(y|x)}\left[\log p_{RBM}(y) + \log T(y \to x) - \log q(y|x)\right]
\end{aligned}
$$

Stagewise Variational Bound

- Say we trained a RBM on a dataset $\{x^{(1)}, \ldots, x^{(D)}\}$, obtaining a set of weights W_{train} (also includes the biases).
- The variational lower bound is exact when $q(y|x) = p(y|x)$:

$$
\begin{aligned}
&\log p(x) \\
=&E_{\log q(y|x)}\left[\log p(x,y) - \log q(y|x)\right] \\
=&E_{\log q(y|x)}\left[\log p(y) + \log p(x|y) - \log q(y|x)\right] \\
=&E_{\log q(y|x)}\left[\log p_{RBM}(y) + \log T(y \rightarrow x) - \log q(y|x)\right]
\end{aligned}
$$

- This is the RBM **unrolled once.**



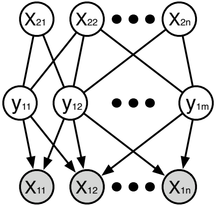
Stagewise Variational Bound

$$\log p(x) = E_{\log q(y|x)}\left[\log p_{RBM}(y) + \log T(y \to x) - \log q(y|x)\right]$$

- Note at this point both

$$p_{RBM}(y) = p_{RBM}(y|W_{train})$$
$$T(y \to x) = T(y \to x|W_{train})$$

are parametrized by the same W_{train} and the variational bound is tight.



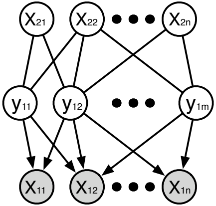
Stagewise Variational Bound

$$\log p(x) = E_{\log q(y|x)} \left[\log p_{RBM}(y) + \log T(y \rightarrow x) - \log q(y|x)\right]$$

- Note at this point both

$$p_{RBM}(y) = p_{RBM}(y|W_{train})$$
$$T(y \rightarrow x) = T(y \rightarrow x|W_{train})$$

  are parametrized by the same W_{train} and the variational bound is tight.
- If we now continue to optimize only $p_{RBM}(y|W)$, we will increase this lower bound on the log likelihood.



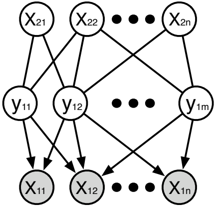
Stagewise Variational Bound

$$\log p(x) = E_{\log q(y|x)} [\log p_{RBM}(y) + \log T(y \to x) - \log q(y|x)]$$

- Note at this point both

$$p_{RBM}(y) = p_{RBM}(y|W_{train})$$
$$T(y \to x) = T(y \to x|W_{train})$$

  are parametrized by the same W_{train} and the variational bound is tight.
- If we now continue to optimize only $p_{RBM}(y|W)$, we will increase this lower bound on the log likelihood.
- Note: the “training set” used to train $p_{RBM}(y|W)$ can be drawn from $q(y|x^{(d)})$ with $x^{(d)}$ a training data point.

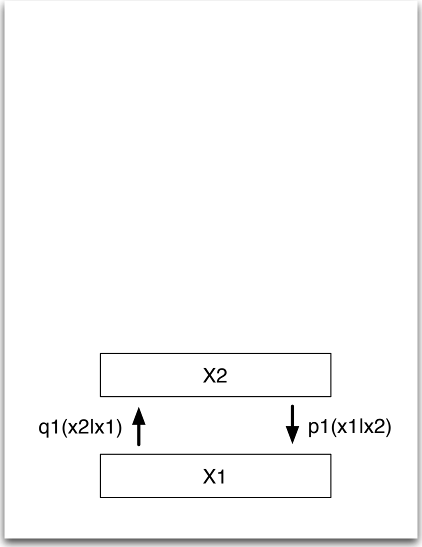


Stagewise Variational Bound

- At stage k learn an RBM, producing a variational posterior

$$q_k(x_{k+1}|x_k)$$
$$p_k(x_k|x_{k+1})$$

- q_k used to “represent” training data points up the stages.
- p_k used to “model” data at the previous stage given higher level representations.
- Each stage of this process increases a variational lower bound on the log likelihood.

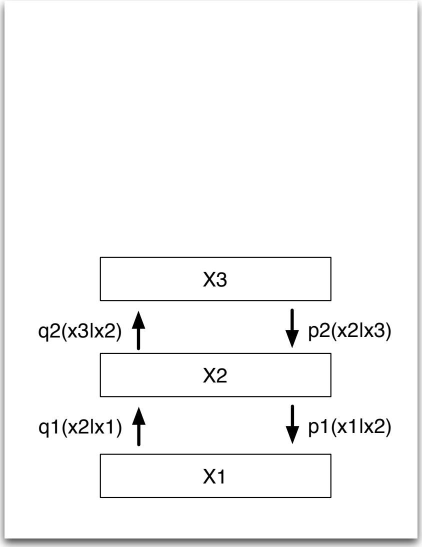


Stagewise Variational Bound

- At stage k learn an RBM, producing a variational posterior

$$q_k(x_{k+1}|x_k)$$
$$p_k(x_k|x_{k+1})$$

- q_k used to “represent” training data points up the stages.
- p_k used to “model” data at the previous stage given higher level representations.
- Each stage of this process increases a variational lower bound on the log likelihood.

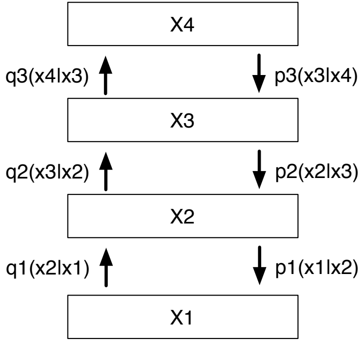


Stagewise Variational Bound

- At stage k learn an RBM, producing a variational posterior

$$q_k(x_{k+1}|x_k)$$
$$p_k(x_k|x_{k+1})$$

- q_k used to “represent” training data points up the stages.
- p_k used to “model” data at the previous stage given higher level representations.
- Each stage of this process increases a variational lower bound on the log likelihood.



Thank You

Thank you!

Thank You

Thank you!
Thank you, Geoff!

Thank You

Thank you!
Thank you, Geoff!
Happy Birthday, Geoff!