On Restrict Boltzmann Machine Learning

Yingzhen Li

University of Cambridge

June 10, 2014

Overview

- Restricted Boltzmann Machine
- Contrastive Divergence
- Expectation Propagation
  - paper in preparation

Deep Learning

- From feature engineering to feature learning
- Layer-wise training of very deep networks
- Promising for AI?

Commercialization

- Microsoft's breakthrough in speech recognition
- 'Google Brain'
- Baidu's Institute of Deep Learning

Neural Networks

- Discriminative models
  - feed-forward networks (1950 - 1980s)
- Generative models
  - Bayes belief networks (1985)
  - Sigmoid belief nets (1996)
- Helmholtz machine (1995)
- Undirected graphical models
  - Markov random field (1980)
  - Boltzmann machine (1986)

Neural Networks

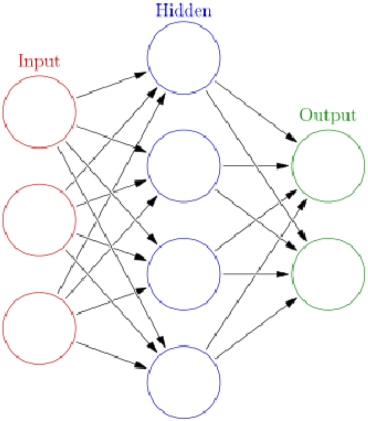
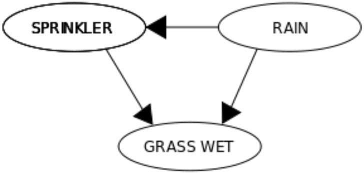


Figure: Feed-forward Nets

Neural Networks

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Figure: Bayes Nets

Neural Networks

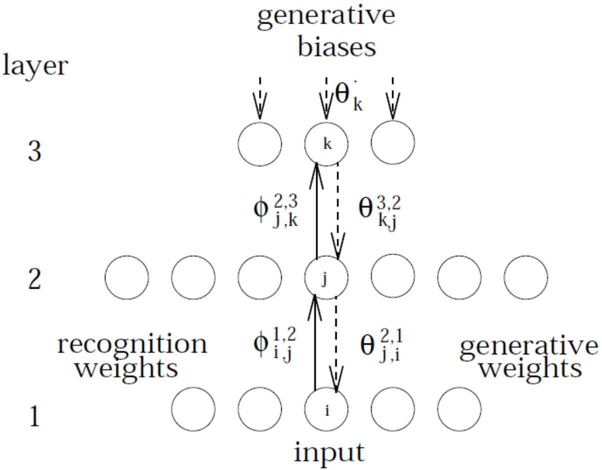


Figure: Helmholtz machine

Neural Networks

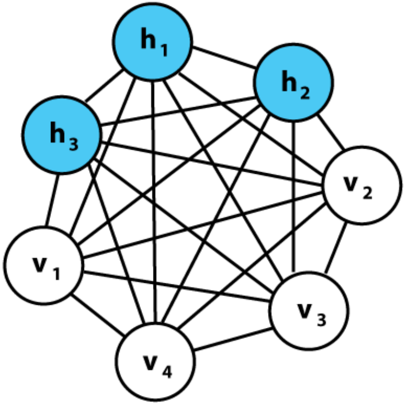
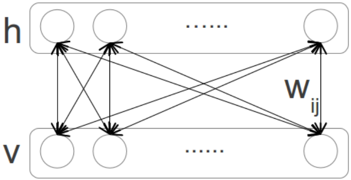


Figure: Boltzmann machine

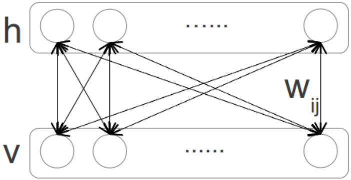
Restricted Boltzmann Machine



$$P(\mathbf{x}, \mathbf{h}|\Theta) = \frac{1}{Z(\Theta)} \exp\left(-E(\mathbf{x}, \mathbf{h}; \Theta)\right) \tag{1}$$

- $E(\mathbf{x}, \mathbf{h}; \Theta) = -\mathbf{h}^T W \mathbf{x} - \mathbf{b}^T \mathbf{x} - \mathbf{c}^T \mathbf{h}$
- $\Theta = \{W, \mathbf{b}, \mathbf{c}\}$
- $Z(\Theta) = \sum_{\mathbf{x},\mathbf{h}} \exp\left(-E(\mathbf{x}, \mathbf{h}; \Theta)\right)$ is often intractable

Restricted Boltzmann Machine



- Maximum likelihood learning

$$\Theta^* = \arg\max_{\Theta} \log P(\mathbf{x}|\Theta), \quad \mathbf{x} \sim P_D \tag{2}$$

- Gradient descent

$$\nabla_{W_{ij}} logP(\mathbf{x}|\Theta) = \langle \mathbf{h}_i \mathbf{x}_j \rangle_{P_D} - \langle \mathbf{h}_i \mathbf{x}_j \rangle_P \tag{3}$$

- Difficult to sample from P DIRECTLY

Contrastive Divergence (Geoff Hinton)

- Difficult to sample from P DIRECTLY

  $\Rightarrow$ try to approximate that expectation!
- Gibbs sampling for k sweeps
- $k = 1$ (CD-k) works well

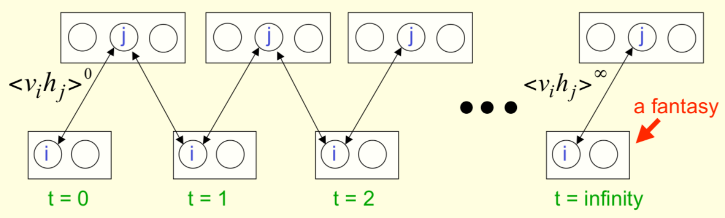
MCMC: Gibbs Sampling

- Iteratively "alternate" between states

$$\mathbf{x}_i^t \sim P(\mathbf{x}_i | \mathbf{x}_1^t, ..., \mathbf{x}_{i-1}^t, \mathbf{x}_{i+1}^{t-1}, ..., \mathbf{x}_n^{t-1})$$

  - no rejection
  - the chain is ergodic (aperiodic + positive recurrent) and irreducible
  - can reach the equilibrium distribution $P_\infty = P$
- denote the sampling procedure as the transition operator T:
  - $\mathbf{x}^k \sim T^k P_D$
  - $P_k = T^k P_D$
  - $P_\infty = T^\infty P_D$

"Truncated" Chain (Geoff Hinton)

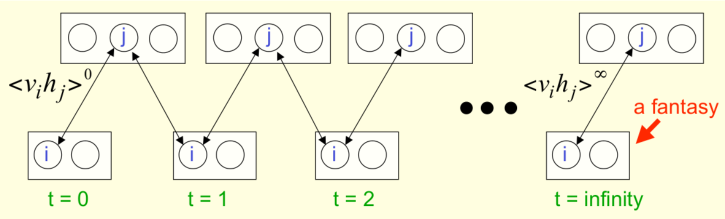


- Run the chain for k transitions (CD-k)
- Block Gibbs sampling:

$$\begin{aligned} P(\mathbf{h}_i|\mathbf{x},\Theta) =& sigm(\mathbf{c}_i + W_{i\cdot}\mathbf{x}) \\ P(\mathbf{x}_j|\mathbf{h},\Theta) =& sigm(\mathbf{b}_j + \mathbf{h}^T W_{\cdot j}) \end{aligned} \tag{4}$$

  - $sigm(x) = (1 + \exp^{-x})^{-1}$

"Truncated" Chain (Geoff Hinton)

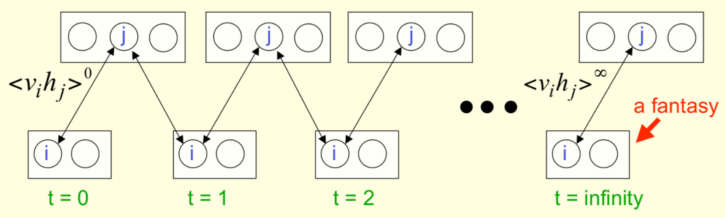


- Run the chain for k transitions (CD-k)

$$\nabla_{W_{ij}} \log P(\mathbf{x}|\Theta) \approx \langle \mathbf{h}_i \mathbf{x}_j \rangle_{P_D} - \langle \mathbf{h}_i \mathbf{x}_j \rangle_{P_k}$$

- $\langle \mathbf{h}_i \mathbf{x}_j \rangle_{P_D} \approx \frac{1}{M} \sum_m \mathbf{h}_i^{(m)} \mathbf{x}_j^{(m)}$
- $\langle \mathbf{h}_i \mathbf{x}_j \rangle_{P_k} \approx \frac{1}{M} \sum_m \mathbf{h}_i^{(m,k)} \mathbf{x}_j^{(m,k)}$

"Truncated" Chain (Geoff Hinton)

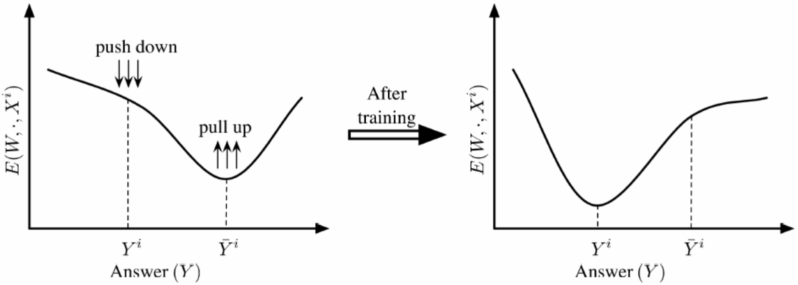


- Run the chain for k transitions (CD-k)
- We are approximately minimizing the objective

$$KL(P_D||P) - KL(P_k||P_\infty)$$

- This tells us the DIRECTION to the (local) optimum

Minimizing the Energy (Yann LeCun)



- High probability at $\mathbf{x}$ = low energy at $\mathbf{x}$
- "wake" gradients $\langle \mathbf{h}_i \mathbf{x}_j \rangle_{P_D}$: reduce the energy at the datapoints
- "sleep" gradients $-\langle \mathbf{h}_i \mathbf{x}_j \rangle_{P_k}$: pull up the energy elsewhere
    - Hopefully $\mathbf{x}^{(m,k)}$ will be far away from $\mathbf{x}^{(m)}$ (when the train mixes quickly)

(Fast) Persistent Contrastive Divergence

- Variance increases with k
- CD-1 can overfit when the chain's mixing rate is low

  $\Rightarrow$ just run the chain without restart!
  - the model changes very slightly between each iterations
- Use fast weights to improve mixing
  - use $\nabla \log P(\mathbf{x}|\Theta + \Theta_{fast})$ instead of $\nabla \log P(\mathbf{x}|\Theta)$
  - use large weight decay of Θ_{fast}

Advanced Models & Techniques

- Models
  - Deep belief nets
  - Deep Boltzmann machines
  - Gaussian RBM, Gated RBM ...
  - Classification RBM
  - Deep Gaussian Process
- Training Methods
  - Annealed importance sampling
  - Dropout
  - Hybrid objective (discriminative + generative)

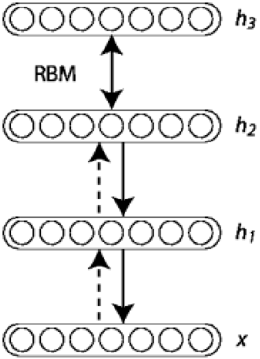


Figure: Deep belief nets

Possible Drawbacks (Literature, Rich & Me)

- Overfitting
  - Hard to remove modes far away from the data
  - No idea about the volume of the mode
- Fail to capture the uncertainty
  - when there's lot of missing data
- Small reconstruction error $\neq$ high data likelihood!
- Can walk away from the true model

Bayesian Inference

- Learning with Bayes Rule:

$$P(\Theta|D) \propto P_0(\Theta)P(D|\Theta)$$

- Bayesian Inference:

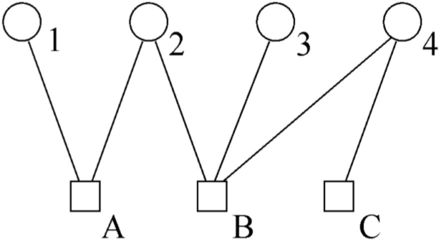
$$P(\mathbf{x}^*) = \int P(\mathbf{x}^*|\Theta)P(\Theta|D)d\Theta$$

- The posterior of an RBM's parameters is intractable

  $\Rightarrow$ approximate that posterior

  - Variational inference
  - **Expectation propagation**

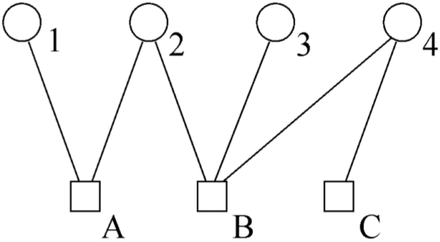
Factor Graph



$$p(x_1, x_2, x_3, x_4) = f_A(x_1, x_2) f_B(x_2, x_3, x_4) f_C(x_4) \tag{5}$$

$$p(\mathbf{x}_S) = \sum_{\mathbf{x} \backslash \mathbf{x}_S} p(\mathbf{x}), \quad \forall S \subset \{x_1, x_2, x_3, x_4\}$$

Expectation Propagation (Tom Minka)



- Define some "simple" $q(\mathbf{x})$ to approximate p:

$$q(x_1, x_2, x_3, x_4) = \tilde{f}_A(x_1, x_2)\tilde{f}_B(x_2, x_3, x_4)\tilde{f}_C(x_4) \tag{6}$$

- Iteratively update $\tilde{f}_i$ by minimizing $KL(q^{\backslash i} f_i || q^{new})$
  - $q^{\backslash i} = q/\tilde{f}_i, i \in \{A, B, C\}$
- Moment Matching: $q^{new} \leftarrow \texttt{moments}[q^{\backslash i} f_i]$

Expectation Propagation (Tom Minka)

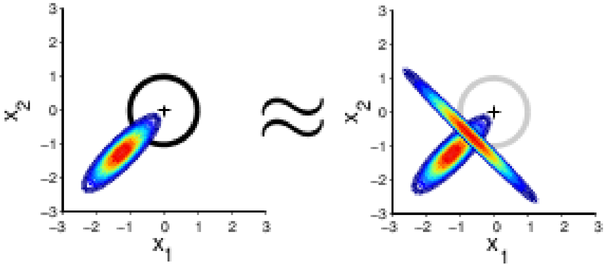


Figure: EP moment matching (fig by Rich Turner)

Bayesian Inference (RBM)

- True posterior:

$$
\begin{aligned}
P(H, \Theta | D) \propto & P_0(\Theta) \prod_m P(\mathbf{x}^{(m)}, \mathbf{h}^{(m)} | \Theta) \\
= & P_0(\Theta) \prod_m \frac{1}{Z(\Theta)} \exp\left(-E(\mathbf{x}^{(m)}, \mathbf{h}^{(m)}; \Theta)\right)
\end{aligned} \tag{7}
$$

- Approximated posterior:

$$
Q(H, \Theta) = P_0(\Theta) \prod_m \frac{1}{\tilde{Z}(\Theta)} \tilde{f}(\mathbf{x}^{(m)}, \mathbf{h}^{(m)}; \Theta) \tag{8}
$$

EP-k (Rich & Me)

- Don't touch $Q(\Theta)$ until a good estimation of $Q(H)$
  - ...by updating $Q(H)$ with EP for k times
- An analogy to CD-k

Bayesian Inference (RBM)

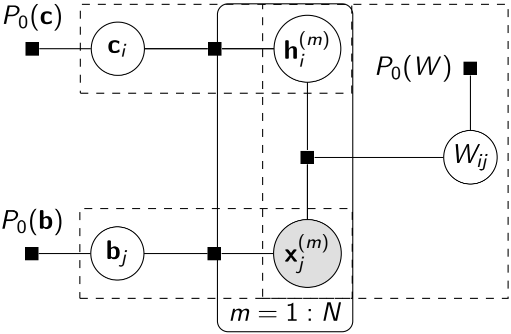


Figure: Restricted Boltzmann Machine as a factor graph. We separate the graph into three subgraph (dashed rectangles) for EP approximation.

Bayesian Inference (RBM)

- Bayesian point estimate (BPE)

$$P(\mathbf{h}^*|D, \mathbf{x}^*) \approx P(\mathbf{h}^*|\mathbf{x}^*; \Theta_{post}), \quad \Theta_{post} \sim Q(\Theta) \tag{9}$$

- Approximate model averaging

$$P(\mathbf{h}^*|D, \mathbf{x}^*) \approx \int_\Theta P(\mathbf{h}^*|\mathbf{x}^*; \Theta_{post}) Q(\Theta) d\Theta \tag{10}$$

$\Rightarrow$ approximate this predictive distribution by EP again!

Future Works

- Finish the experiments on biased RBM and get it published!
- My PhD thesis would be:
    - theoretical analysis of deep learning in a Bayesian flavour
    - denoising & filling the missing data
    - fast & parallel algorithms (like that of TrueSkill™)
    - extension to continuous hidden states deep models
        - Deep Gaussian Process
        - Boltzmann machines with other continuous energy functions, e.g. Gaussian CDF

Bengio, Y. (2009), 'Learning Deep Architectures for AI.', Foundations and Trends in Machine Learning 2 (1) , 1-127.
Bengio, Y.; Courville, A. C. & Vincent, P. (2013), 'Representation Learning: A Review and New Perspectives.', IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) , 1798-1828.
Dayan, P.; Hinton, G. E.; Neal, R. N. & Zemel, R. S. (1995), 'The Helmholtz Machine', Neural Computation 7 , 889-904.
Hinton, G. E.; Osindero, S. & Teh, Y. W. (2006), 'A Fast Learning Algorithm for Deep Belief Nets', Neural Computation 18 , 1527-1554.

Minka, T. (2001), 'A family of algorithms for approximate Bayesian inference', PhD thesis, Massachusetts Institute of Technology.

Qi, Y.; Szummer, M. & Minka, T. (2005), 'Bayesian Conditional Random Fields', Proc. AISTATS 2005.

Tieleman, T. (2008), Training restricted Boltzmann machines using approximations to the likelihood gradient., in 'ICML' , ACM, , pp. 1064-1071.

Tieleman, T. & Hinton, G. E. (2009), Using fast weights to improve persistent contrastive divergence., in 'ICML' , ACM, , pp. 130.