Deep Belief Nets

Sargur N. Srihari
srihari@cedar.buffalo.edu

Topics

1. Boltzmann machines
   2. Restricted Boltzmann machines
   3. Deep Belief Networks
   4. Deep Boltzmann machines
   5. Boltzmann machines for continuous data
   6. Convolutional Boltzmann machines
   7. Boltzmann machines for structured and sequential outputs
   8. Other Boltzmann machines
9. Backpropagation through random operations
10. Directed generative nets
11. Drawing samples from autoencoders
12. Generative stochastic networks
13. Other generative schemes
14. Evaluating generative models
15. Conclusion

Topics

1. History of Deep Belief Networks (DBNs)
2. What are DBNs?
   - Example of a DBN
   - DBNs as Hybrid PGMs
3. Probability distribution represented by a DBN
4. Inference with a DBN
   - Sampling from a DBN
5. Training a DBN
6. Using the DBN

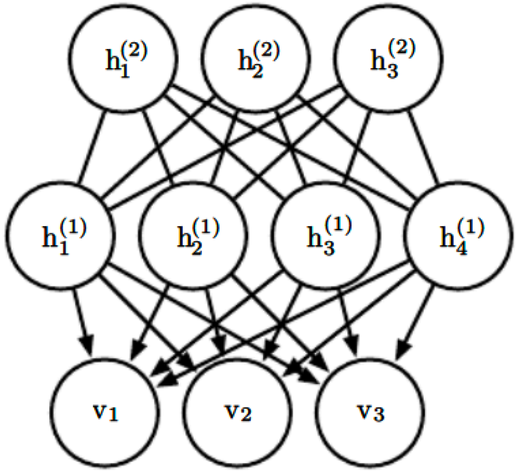
History of Deep Belief Networks

- One of the first non-convolutional models to admit training of deep architectures
  - Deep belief networks started the current deep learning renaissance
  - Prior to this deep modes were considered too difficult to optimize
    - Kernel machines with convex objective functions dominated the landscape
  - Demonstrated that deep architectures outperformed kernelized SVM on MNIST
- Today deep belief networks have fallen out of favor and rarely used

What are deep belief networks?

- They are generative models with several layers of latent variables
  - Latent variables are typically binary
  - Visible layers can be binary or real
  - There are no intra-layer connections
- Connections between top two layers are undirected
- Connections between all other layers is directed, pointing towards data

An example of a DBN



- It is a hybrid graphical model involving both directed and undirected connections
  - No intra-layer connections
  - Has multiple hidden layers

Distribution represented by a DBM

- A DBN with ℓ hidden layers has ℓ weight matrices $\mathrm{W}^{(1)},..,\mathrm{W}^{(\ell)}$.
- It contains $\ell+1$ bias vectors $\boldsymbol{b}^{(1)},..,\boldsymbol{b}^{(\ell)}$
- Bias $\boldsymbol{b}^{(0)}$ provides biases for the visible layer
- The probability distribution represented is:

$$\boxed{\begin{aligned} P(\boldsymbol{h}^{(l)}, \boldsymbol{h}^{(l-1)}) &\propto \exp\left(\boldsymbol{b}^{(l)\top}\boldsymbol{h}^{(l)} + \boldsymbol{b}^{(l-1)\top}\boldsymbol{h}^{(l-1)} + \boldsymbol{h}^{(l-1)\top}\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l)}\right), \\ P(h_i^{(k)} = 1 \mid \boldsymbol{h}^{(k+1)}) &= \sigma\left(b_i^{(k)} + \boldsymbol{W}_{:,i}^{(k+1)\top}\boldsymbol{h}^{(k+1)}\right) \forall i, \forall k \in 1, \ldots, l-2, \\ P(v_i = 1 \mid \boldsymbol{h}^{(1)}) &= \sigma\left(b_i^{(0)} + \boldsymbol{W}_{:,i}^{(1)\top}\boldsymbol{h}^{(1)}\right) \forall i. \end{aligned}}$$

  - In the case of real-valued variables

$$\boxed{\mathbf{v} \sim \mathcal{N}\left(\boldsymbol{v}; \boldsymbol{b}^{(0)} + \boldsymbol{W}^{(1)\top}\boldsymbol{h}^{(1)}, \boldsymbol{\beta}^{-1}\right)}$$

Sampling from a DBN

- To sample from a DBN:
- First run several steps of Gibbs sampling from the top two hidden layers
  - This stage is drawing a sample from the RBM defined by the top two layers
- Then use a single pass of ancestral sampling through rest of the model
  - to draw a sample from the visible units

Inference in a DBN

- Intractability of inference is due to:
  - the explaining away effect within each directed layer
  - Interaction between two hidden layers that have undirected connections
- Evaluating or maximizing standard evidence bound on the log-likelihood is also intractable
  - Beacause evidence bound takes the expectation of cliques
    - whose size is equal to network width

Training a DBN

- Begin by training an RBM to maximize $\mathbb{E}_{\mathbf{v}\sim p_{\text{data}}} \log p(\boldsymbol{v})$
  - Using contrastive divergence or stochastic maximum likelihood
    - Parameters of RBM then define parameters of first layer of DBN
- Next, a second RBM is trained to maximize

$$\mathbb{E}_{\mathbf{v}\sim p_{\text{data}}} \mathbb{E}_{\mathbf{h}^{(1)}\sim p^{(1)}(\boldsymbol{h}^{(1)}|\boldsymbol{v})} \log p^{(2)}(\boldsymbol{h}^{(1)})$$

  - Where $p^{(1)}$ and $p^{(2)}$: probability distributions represented by the two RBMs
    - In effect second RBM is trained to model the distribution defined by sampling the hidden units of the first RBM

Using a DBN

- The trained DBN may be directly used as a generative model
- But most interest arose from classification problems
- We can use weights of DBN to define an MLP

$$\boldsymbol{h}^{(1)} = \sigma\left(b^{(1)} + \boldsymbol{v}^\top \boldsymbol{W}^{(1)}\right).$$

$$\boldsymbol{h}^{(l)} = \sigma\left(b_i^{(l)} + \boldsymbol{h}^{(l-1)\top} \boldsymbol{W}^{(l)}\right) \forall l \in 2, \ldots, m,$$