Deep Belief Nets

Sargur N. Srihari
srihari@cedar.buffalo.edu

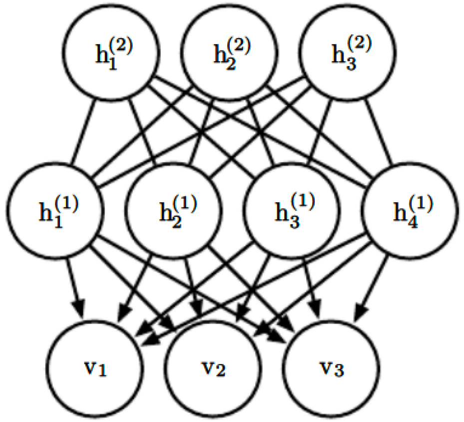
Topics

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 graphical models
3. Probability distribution represented by a DBN
4. Sampling from a DBN
5. Inference with a DBN
6. Training a DBN
7. Using the DBN

Deep Belief Net Architecture



A hybrid graphical model involving both directed and undirected connections.
Like an RBM it has no intralayer connections.

However a DBN has multiple hidden layers and thus connections between hidden units are in separate layers.

All the local CPDs needed by the DBN are copied directly from the local CPDs of Its constituent RBMs.

Alternatively, we could represent the DBN with a completely undirected graph, but it would need intralayer connections to capture the dependencies between parents.

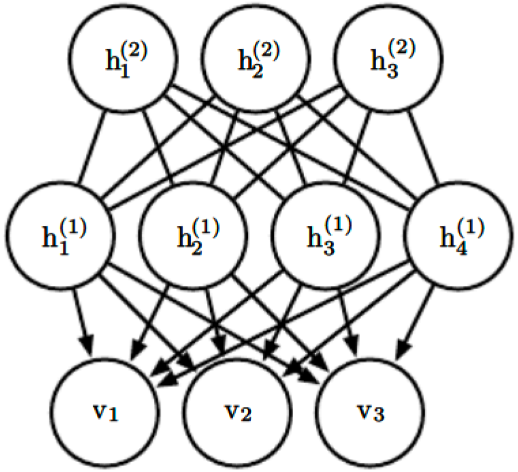
History of Deep Belief Networks

- 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 models were considered too difficult to optimize
      - Kernel machines with convex objective functions dominated the landscape
      - DBNs Demonstrated that deep architectures outperformed kernelized SVM on MNIST
- Today deep belief networks have fallen out of favor and rarely used
  - But still studied due to their role in deep learning

DBNs as generative models

- DBNs 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



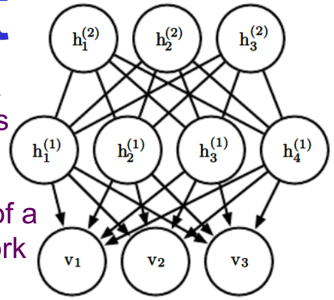
- It is a hybrid graphical model involving both directed and undirected connections
    - Like an RBM it has no intra-layer connections
    - Has multiple hidden layers
    - Connection between top two layers are undirected
    - Connections between all other layers are directed

Distribution represented by a DBN

- A DBN with l hidden layers has l weight matrices $W^{(1)},\ldots,W^{(l)}$
  - It contains $l+1$ bias vectors $\boldsymbol{b}^{(1)},\ldots,\boldsymbol{b}^{(l)}$
  - Bias $\boldsymbol{b}^{(0)}$ provides biases for the visible layer
  - Probability distribution represented (binary $\boldsymbol{v_i},\boldsymbol{h_i}$) is:

A Markov Network over top two layers

Several layers of a Bayesian network



$$P(\boldsymbol{h}^{(l)},\boldsymbol{h}^{(l-1)}) \propto \exp\left(\boldsymbol{b}^{(l)\top}\boldsymbol{h}^{(l)} + \boldsymbol{b}^{(l-1)\top}\boldsymbol{h}^{(l-1)} + \boldsymbol{h}^{(l-1)\top}\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l)}\right)$$

Joint probabilities of a pair of hidden layers MN expressed in terms of energy function

$$P(h_i^{(k)}=1 \mid \boldsymbol{h}^{(k+1)}) = \sigma\left(b_i^{(k)} + \boldsymbol{W}_{:,i}^{(k+1)\top}\boldsymbol{h}^{(k+1)}\right) \forall i, \forall k \in 1,\ldots,l-2,$$

CPDs of lower hidden layers, given layer above

$$P(v_i=1 \mid \boldsymbol{h}^{(1)}) = \sigma\left(b_i^{(0)} + \boldsymbol{W}_{:,i}^{(1)\top}\boldsymbol{h}^{(1)}\right) \forall i.$$

CPDs of visible layers, given first hidden layer

  - In the case of real-valued variables, substitute

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

With β diagonal for tractability

  - A DBN with only one hidden layer is an RBM

Sampling from a DBN

- To generate a sample from a DBN:
  - (say, to compute expected values of features represented by top latent layer $\boldsymbol{h}^{(l)}$
- 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
    - Say to generate a visible sample $\boldsymbol{v}$ from the model

Inference in a DBN

- Inference is the task of determining a probability from a model, e.g., $P(\boldsymbol{h}|\boldsymbol{v})$ given $P(\boldsymbol{h},\boldsymbol{v})$
- Inference in a DBN is intractable 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
  - Because evidence bound takes the expectation of cliques whose size is equal to network width

Evaluating log-likelihood

- Evaluating or maximizing log-likelihood for training (parameter estimation) requires solving two intractable problems:
  1. Inference: marginalizing out latent variables
     - (likelihood is product of probabilities of visible samples)
  2. Partition function
     - Within the undirected model of the top two layers

To Train 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 ML
    - RBM Parameters 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 $p^{(1)}$, $p^{(2)}$: distributions represented by the two RBMs
    - In other words, second RBM is trained to model the distribution defined by sampling the hidden units of the first RBM, when the first RBM is driven by the data
  - This procedure can be repeated indefinitely to add as many layers to the DBN as desired
    - With each RBM modeling the samples of the previous one (justified as increasing the variational lower bound)

Using a DBN

- The trained DBN may be directly used as a generative model
  - But most interest in DBNs arose from classification
    - To do this, take the weights from the DBN to define an MLP

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

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

- After initializing this MLP with the weights and biases learned via generative training of the DBN, we can train the MLP to perform a classification task
  - This additional training of the MLP is an example of discriminative fine-tuning
- Choice of MLP is arbitrary