Deep Boltzmann Machines I
Unsupervised Learning

Non-probabilistic Models
- Sparse Coding
- Autoencoders
- Others (e.g. k-means)

Probabilistic (Generative) Models

Tractable Models
- Fully observed Belief Nets
- NADE
- PixelRNN

Non-tractable Models
- Boltzmann Machines
- Variational Autoencoders
- Helmholtz Machines
- DBNs, many others...

Explicit Density $p(x)$

Implicit Density

Generative Adversarial Networks
Moment Matching Networks
DBNs vs. DBMs

Deep Belief Network

Deep Boltzmann Machine

DBNs are hybrid models:

- Inference in DBNs is problematic due to **explaining away**.
- Only greedy pretraining, **no joint optimization over all layers**.
- Approximate inference is feed-forward: **no bottom-up and top-down**.

Introduce a new class of models called Deep Boltzmann Machines.
25,000 characters from 50 alphabets around the world.

- 3,000 hidden variables
- 784 observed variables (28 by 28 images)
- About 2 million parameters

Model $P(\text{image})$

Bernoulli Markov Random Field
Deep Generative Model

Conditional Simulation

\[ P(\text{image} | \text{partial image}) \]

Bernoulli Markov Random Field
Deep Generative Model

Conditional Simulation

Why so difficult?

$2^{28 \times 28}$ possible images!

Bernoulli Markov Random Field

$P(\text{image} | \text{partial image})$
Fully Observed Models

• Explicitly model conditional probabilities:

\[ p_{\text{model}}(\mathbf{x}) = p_{\text{model}}(x_1) \prod_{i=2}^{n} p_{\text{model}}(x_i \mid x_1, \ldots, x_{i-1}) \]

• A number of successful models, including

  ➢ NADE, RNADE (Larochelle, et.al. 20011)
  ➢ Pixel CNN (van den Ord et. al. 2016)
  ➢ Pixel RNN (van den Ord et. al. 2016)
Restricted Boltzmann Machines

\[ P_\theta(v, h) = \frac{1}{Z(\theta)} \exp \left( \sum_{i=1}^{D} \sum_{j=1}^{F} W_{ij} v_i h_j + \sum_{i=1}^{D} v_i b_i + \sum_{j=1}^{F} h_j a_j \right) \]
\[ \theta = \{W, a, b\} \]

\[ P_\theta(v|h) = \prod_{i=1}^{D} P_\theta(v_i|h) = \prod_{i=1}^{D} \frac{1}{1 + \exp(-\sum_{j=1}^{F} W_{ij} v_i h_j - b_i)} \]

RBM is a Markov Random Field with:

- Stochastic binary visible variables \( v \in \{0, 1\}^D \).
- Stochastic binary hidden variables \( h \in \{0, 1\}^F \).
- Bipartite connections.

Markov random fields, Boltzmann machines, log-linear models.
Model Learning

\[ P_{\theta}(v) = \frac{P^*(v)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_h \exp \left[ v^T W h + a^T h + b^T v \right] \]

Given a set of \textit{i.i.d.} training examples \( \mathcal{D} = \{v^{(1)}, v^{(2)}, \ldots, v^{(N)}\} \), we want to learn model parameters \( \theta = \{W, a, b\} \).

Maximize log-likelihood objective:

\[ L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_{\theta}(v^{(n)}) \]

Derivative of the log-likelihood:

\[
\frac{\partial L(\theta)}{\partial W_{ij}} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial W_{ij}} \log \left( \sum_h \exp \left[ v^{(n) \top} W h + a^\top h + b^\top v^{(n)} \right] \right) - \frac{\partial}{\partial W_{ij}} \log Z(\theta)
\]

\[ = E_{P_{data}}[v_i h_j] - E_{P_{\theta}}[v_i h_j] \]

\[ P_{data}(v, h; \theta) = P(h|v; \theta) P_{data}(v) \]

\[ P_{data}(v) = \frac{1}{N} \sum_n \delta(v - v^{(n)}) \]

\textbf{Difficult to compute: exponentially many configurations}
Derivative of the log-likelihood:

\[
\frac{\partial L(\theta)}{\partial W_{ij}} = \mathbb{E}_{P_{data}[v_i h_j]} - \mathbb{E}_{P_{\theta}[v_i h_j]} - \sum_{v, h} v_i h_j P_{\theta}(v, h)
\]

**Easy to compute exactly**

**Difficult to compute:** exponentially many configurations.

Use MCMC

Approximate maximum likelihood learning

\[
P_{data}(v, h; \theta) = P(h|v; \theta)P_{data}(v)
\]

\[
P_{data}(v) = \frac{1}{N} \sum_n \delta(v - v^{(n)})
\]
Deep Boltzmann Machines

Input: Pixels

Low-level features: Edges

Built from unlabeled inputs.

(Salakhutdinov 2008, Salakhutdinov & Hinton 2012)
Deep Boltzmann Machines

Learn simpler representations, then compose more complex ones

Higher-level features: Combination of edges

Low-level features: Edges

Built from unlabeled inputs.

Input: Pixels

(Salakhutdinov 2008, Salakhutdinov & Hinton 2012)
Mathematical Formulation

\[ P_\theta(v) = \frac{P^*(v)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_{h^1, h^2, h^3} \exp \left[ v^T W^1 h^1 + h^1^T W^2 h^2 + h^2^T W^3 h^3 \right] \]

\[ \theta = \{W^1, W^2, W^3\} \text{ model parameters} \]

- Dependencies between hidden variables.
- All connections are undirected.
- Bottom-up and Top-down:

\[ P(h^2_k = 1|h^1, h^3) = \sigma \left( \sum_j W^2_{jk} h^1_j + \sum_m W^3_{km} h^3_m \right) \]

Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio et.al.), Deep Belief Nets (Hinton et.al.)
Mathematical Formulation

\[ P_\theta(v) = \frac{P^*(v)}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{h^1, h^2, h^3} \exp \left[ v^\top W^1 h^1 + h^1^\top W^2 h^2 + h^2^\top W^3 h^3 \right] \]

Deep Boltzmann Machine

- Conditional Distributions:
  \[ P(h_j^1 = 1|v, h^2) = \sigma \left( \sum_i W_{ij}^1 v_i + \sum_k W_{jk}^2 h_k^2 \right) \]
  \[ P(h_k^2 = 1|h^1, h^3) = \sigma \left( \sum_j W_{jk}^2 h_j^1 + \sum_m W_{km}^3 h_m^3 \right) \]
  \[ P(h_m^3 = 1|h^2) = \sigma \left( \sum_k W_{km}^3 h_k^2 \right) \]

- Note that exact computation of \( P(h^1, h^2, h^3|v) \) is intractable.
Mathematical Formulation

\[
P_\theta(v) = \frac{P^*(v)}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{h^1, h^2, h^3} \exp \left[ v^\top W^1 h^1 + h^1^\top W^2 h^2 + h^2^\top W^3 h^3 \right]
\]

Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio), Deep Belief Nets (Hinton)
Mathematical Formulation

\[ P_\theta(v) = \frac{P^*(v)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_{h^1, h^2, h^3} \exp \left[ v^T W^1 h^1 + h^1^T W^2 h^2 + h^2^T W^3 h^3 \right] \]

Deep Boltzmann Machine

Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio), Deep Belief Nets (Hinton)
Mathematical Formulation

\[ P_\theta(v) = \frac{P^*(v)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_{h^1, h^2, h^3} \exp \left[ v^\top W^1 h^1 + h^{1\top} W^2 h^2 + h^{2\top} W^3 h^3 \right] \]

\( \theta = \{ W^1, W^2, W^3 \} \) model parameters

- Dependencies between hidden variables.

Maximum likelihood learning:

\[ \frac{\partial \log P_\theta(v)}{\partial W^1} = E_{P_{data}}[vh^{1\top}] - E_{P_\theta}[vh^{1\top}] \]

**Problem:** Both expectations are intractable!

Learning rule for undirected graphical models: MRFs, CRFs, Factor graphs.
Approximate Learning

\[ P_\theta(v, h^{(1)}, h^{(2)}, h^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp \left[ v^\top W^{(1)} h^{(1)} + h^{(1)}^\top W^{(2)} h^{(2)} + h^{(2)}^\top W^{(3)} h^{(3)} \right] \]

(Approximate) Maximum Likelihood:

\[ \frac{\partial \log P_\theta(v)}{\partial W^1} = \mathbb{E}_{P_{data}} [vh^{1\top}] - \mathbb{E}_{P_\theta} [vh^{1\top}] \]

- Both expectations are intractable!

\[ P_{data}(v, h^1) = P_\theta(h^1|v)P_{data}(v) \]

\[ P_{data}(v) = \frac{1}{N} \sum_{n=1}^{N} \delta(v - v_n) \]

Not factorial any more!
Approximate Learning

\[ P_\theta(v, h^{(1)}, h^{(2)}, h^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp \left[ v^\top W^{(1)} h^{(1)} + h^{(1)}^\top W^{(2)} h^{(2)} + h^{(2)}^\top W^{(3)} h^{(3)} \right] \]

(Approximate) Maximum Likelihood:

\[ \frac{\partial \log P_\theta(v)}{\partial W^1} = \mathbb{E}_{P_{\text{data}}} [vh^1] - \mathbb{E}_{P_\theta} [vh^1] \]

\[ P_{\text{data}}(v, h^1) = P_\theta(h^1|v)P_{\text{data}}(v) \]

\[ P_{\text{data}}(v) = \frac{1}{N} \sum_{n=1}^{N} \delta(v - v_n) \]

Not factorial any more!
Approximate Learning

\[ P_\theta(v, h^{(1)}, h^{(2)}, h^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp \left[ v^\top W^{(1)} h^{(1)} + h^{(1)}^\top W^{(2)} h^{(2)} + h^{(2)}^\top W^{(3)} h^{(3)} \right] \]

(Approximate) Maximum Likelihood:

\[
\frac{\partial \log P_\theta(v)}{\partial W^1} = \mathbb{E}_{P_{data}}[vh^1\top] - \mathbb{E}_{P_\theta}[vh^1\top]
\]

Variational Inference

Stochastic Approximation (MCMC-based)

Not factorial any more!

\[ P_{data}(v, h^1) = P_\theta(h^1|v)P_{data}(v) \]

\[ P_{data}(v) = \frac{1}{N} \sum_{n=1}^{N} \delta(v - v_n) \]
Previous Work

Many approaches for learning Boltzmann machines have been proposed over the last 20 years:

- Hinton and Sejnowski (1983),
- Peterson and Anderson (1987)
- Galland (1991)
- Kappen and Rodriguez (1998)
- Lawrence, Bishop, and Jordan (1998)
- Tanaka (1998)
- Welling and Hinton (2002)
- Welling and Teh (2003)
- Yasuda and Tanaka (2009)

Real-world applications – thousands of hidden and observed variables with millions of parameters.

Many of the previous approaches were not successful for learning general Boltzmann machines with **hidden variables**.

New Learning Algorithm

**Posterior Inference**

- **Conditional**
  - Approximate conditional $P_{data}(h|v)$

**Simulate from the Model**

- Approximate the joint distribution $P_{model}(h, v)$

---

(Salakhutdinov, 2008; NIPS 2009)
New Learning Algorithm

Posterior Inference

Approximate conditional
\[ P_{data}(h|v) \]
Data-dependent

\[ E_{P_{data}}[vh^T] \]

Simulate from the Model

Approximate the joint distribution
\[ P_{model}(h, v) \]
Data-independent

\[ E_{P_{model}}[vh^T] \]

Unconditional

(Salakhutdinov, 2008; NIPS 2009)
New Learning Algorithm

Key Idea of Our Approach:
Data-dependent: Variational Inference, mean-field theory
Data-independent: Stochastic Approximation, MCMC based
Sampling from DBMs

Sampling from two-hidden layer DBM by running a Markov chain:

$$P(h^1_m = 1|v, h^2) = \frac{1}{1 + \exp(-\sum_i W^1_{im} v_i - \sum_j W^2_{mj} h^2_j)}$$

$$P(h^2_j = 1|h^1) = \frac{1}{1 + \exp(-\sum_m W^2_{mj} h^1_m)}$$

$$P(v_i = 1|h^1) = \frac{1}{1 + \exp(-\sum_m W^1_{im} h^1_m)}$$
Stochastic Approximation

Update $\theta_t$ and $x_t$ sequentially, where $x = \{v, h^1, h^2\}$

- Generate $x_t \sim T_{\theta_t}(x_t \leftarrow x_{t-1})$ by simulating from a Markov chain that leaves $P_{\theta_t}$ invariant (e.g. Gibbs or M-H sampler)
- Update $\theta_t$ by replacing intractable $E_{P_{\theta_t}}[vh^\top]$ with a point estimate $[v_t h_t^\top]$

In practice we simulate several Markov chains in parallel.

L. Younes, Probability Theory 1989
Learning Algorithm

Update rule decomposes:

$$\theta_{t+1} = \theta_t + \alpha_t \left( \mathbb{E}_{P_{data}}[v_h^\top] - \mathbb{E}_{P_{\theta_t}}[v_h^\top] \right) + \alpha_t \left( \mathbb{E}_{P_{\theta_t}}[v_h^\top] - \frac{1}{M} \sum_{m=1}^{M} v_t^{(m)} h_t^{(m)^\top} \right)$$

- True gradient
- Perturbation term $\epsilon_t$

Almost sure convergence guarantees as learning rate $\alpha_t \to 0$

**Problem:** High-dimensional data: the probability landscape is highly multimodal.

**Key insight:** The transition operator can be any valid transition operator – Tempered Transitions, Parallel/Simulated Tempering

Connections to the theory of stochastic approximation and adaptive MCMC.
Variational Inference

Approximate intractable distribution \( P_\theta(h|v) \) with simpler, tractable distribution \( Q_\mu(h|v) \):

\[
\log P_\theta(v) = \log \sum_h P_\theta(h, v) = \log \sum_h Q_\mu(h|v) \frac{P_\theta(h, v)}{Q_\mu(h|v)} \\
\geq \sum_h Q_\mu(h|v) \log \frac{P_\theta(h, v)}{Q_\mu(h|v)} \\
= \sum_h Q_\mu(h|v) \log P_\theta^*(h, v) - \log Z(\theta) + \sum_h Q_\mu(h|v) \log \frac{1}{Q_\mu(h|v)} \\
v^T W^1 h^1 + h^1^T W^2 h^2 + h^2^T W^3 h^3
\]

Variational Lower Bound

\[
= \log P_\theta(v) - \text{KL} \left( Q_\mu(h|v) \| P_\theta(h|v) \right)
\]

Minimize KL between approximating and true distributions with respect to variational parameters \( \mu \).

(Salakhutdinov, 2008; Salakhutdinov & Larochelle, AI & Statistics 2010)
Variational Inference

Approximate intractable distribution $P_\theta(h|v)$ with simpler, tractable distribution $Q_\mu(h|v)$:

$$\log P_\theta(v) \geq \log P_\theta(v) - KL(Q_\mu(h|v)||P_\theta(h|v))$$

$KL(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$

**Mean-Field:** Choose a fully factorized distribution:

$$Q_\mu(h|v) = \prod_{j=1}^{F} q(h_j|v) \text{ with } q(h_j = 1|v) = \mu_j$$

**Variational Inference:** Maximize the lower bound w.r.t. Variational parameters $\mu$.

Nonlinear fixed-point equations:

$$\mu_j^{(1)} = \sigma \left( \sum_i W_{ij}^1 v_i + \sum_k W_{jk}^2 \mu_k^{(2)} \right)$$

$$\mu_k^{(2)} = \sigma \left( \sum_j W_{jk}^2 \mu_j^{(1)} + \sum_m W_{km}^3 \mu_m^{(3)} \right)$$

$$\mu_m^{(3)} = \sigma \left( \sum_k W_{km}^3 \mu_k^{(2)} \right)$$
Variational Inference

Approximate intractable distribution $P_\theta(h|v)$ with simpler, tractable distribution $Q_\mu(h|v)$:

$$
\log P_\theta(v) \geq \log P_\theta(v) - KL(Q_\mu(h|v)\|P_\theta(h|v))
$$

$KL(Q\|P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$

1. **Variational Inference:** Maximize the lower bound w.r.t. variational parameters

2. **MCMC:** Apply stochastic approximation to update model parameters

Almost sure convergence guarantees to an asymptotically stable point.
Variational Inference

Approximate intractable distribution $P_\theta(h|v)$ with simpler, tractable distribution $Q_\mu(h|v)$:

$$
\log P_\theta(v) \geq \log P_\theta(v) - KL(Q_\mu(h|v)||P_\theta(h|v))
$$

$$
KL(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx
$$

1. Variational Inference: Maximize the lower bound w.r.t. variational parameters.

2. MCMC: Apply stochastic approximation to update model parameters. Almost sure convergence guarantees to an asymptotically stable point.

Fast Inference

Learning can scale to millions of examples

Markov Chain Monte Carlo

Unconditional Simulation
Good Generative Model?

Handwritten Characters
Good Generative Model?

Handwritten Characters
Good Generative Model?

Handwritten Characters

Simulated Real Data
Good Generative Model?

Handwritten Characters

Real Data  Simulated
Good Generative Model?

Handwritten Characters
Good Generative Model?

MNIST Handwritten Digit Dataset
## Handwriting Recognition

### MNIST Dataset
60,000 examples of 10 digits

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic regression</td>
<td>12.0%</td>
</tr>
<tr>
<td>K-NN</td>
<td>3.09%</td>
</tr>
<tr>
<td>Neural Net (Platt 2005)</td>
<td>1.53%</td>
</tr>
<tr>
<td>SVM (Decoste et.al. 2002)</td>
<td>1.40%</td>
</tr>
<tr>
<td>Deep Autoencoder (Bengio et. al. 2007)</td>
<td>1.40%</td>
</tr>
<tr>
<td>Deep Belief Net (Hinton et. al. 2006)</td>
<td>1.20%</td>
</tr>
<tr>
<td>DBM</td>
<td>0.95%</td>
</tr>
</tbody>
</table>

### Optical Character Recognition
42,152 examples of 26 English letters

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic regression</td>
<td>22.14%</td>
</tr>
<tr>
<td>K-NN</td>
<td>18.92%</td>
</tr>
<tr>
<td>Neural Net</td>
<td>14.62%</td>
</tr>
<tr>
<td>SVM (Larochelle et.al. 2009)</td>
<td>9.70%</td>
</tr>
<tr>
<td>Deep Autoencoder (Bengio et. al. 2007)</td>
<td>10.05%</td>
</tr>
<tr>
<td>Deep Belief Net (Larochelle et. al. 2009)</td>
<td>9.68%</td>
</tr>
<tr>
<td>DBM</td>
<td>8.40%</td>
</tr>
</tbody>
</table>

Permutation-invariant version.
Generative Model of 3-D Objects

24,000 examples, 5 object categories, 5 different objects within each category, 6 lightning conditions, 9 elevations, 18 azimuths.
3-D Object Recognition

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic regression</td>
<td>22.5%</td>
</tr>
<tr>
<td>K-NN (LeCun 2004)</td>
<td>18.92%</td>
</tr>
<tr>
<td>SVM (Bengio &amp; LeCun 2007)</td>
<td>11.6%</td>
</tr>
<tr>
<td>Deep Belief Net (Nair &amp; Hinton 2009)</td>
<td>9.0%</td>
</tr>
<tr>
<td>DBM</td>
<td>7.2%</td>
</tr>
</tbody>
</table>

Pattern Completion

Permutation-invariant version.
Learning Hierarchical Representations

Deep Boltzmann Machines:

Learning Hierarchical Structure in Features: edges, combination of edges.

• Performs well in many application domains
• Fast Inference: fraction of a second
• Learning scales to millions of examples
Learning Hierarchical Representations

Deep Boltzmann Machines:

- Need more structured and robust models


Hallucinations in Charles Bonnet Syndrome Induced by Homeostasis: a Deep Boltzmann Machine Model (Reichert, Series, Storkey, NIPS 2012)

Demo DBM