Learning Deep Boltzmann Machines

Ruslan Salakhutdinov
CSAIL, MIT
Outline

- Boltzmann Machines, Restricted Boltzmann Machines.
- Learning: MCMC + Variational Inference.
- Deep Boltzmann Machines.
- Results.
Boltzmann Machines

\[
P(v, h; \theta) = \frac{1}{Z(\theta)} \exp \left[ v^\top W h + \frac{1}{2} v^\top L v + \frac{1}{2} h^\top J h \right].
\]

\[
P(v; \theta) = \sum_h P(v, h; \theta).
\]

Set of visible \(v\) and hidden \(h\) binary stochastic units.

\(\theta = \{W, L, J\}\) are model parameters.

Inference and maximum likelihood learning are hard.

This talk: Learning \(\theta\).
Restricted Boltzmann Machines

\[ P(v) = \frac{1}{Z} \sum_h \exp [v^\top W h]. \]

\( P^*(v) \), tractable

Computing \( P(h|v) \) is easy. Maximum likelihood learning is hard.
Boltzmann Machines: Learning

\[ P_{\text{model}}(v, h) = \frac{1}{Z} \exp \left[ v^\top Wh + \frac{1}{2} v^\top Lv + \frac{1}{2} h^\top Jh \right]. \]

Maximum Likelihood Learning:

\[ \frac{\partial \ln P(v)}{\partial W} = E_{P_{\text{data}}}[vh^\top] - E_{P_{\text{model}}}[vh^\top]. \]

\[ P_{\text{data}}(h, v) = P(h|v)P_{\text{data}}(v). \]

Previous Approach: For each iteration of learning:
1. A separate Markov chain is run for every data point to approximate \( E_{P_{\text{data}}}[\cdot] \).
2. An additional chain is run to approximate \( E_{P_{\text{model}}}[\cdot] \).
Learning

\[ P_{\text{model}}(v, h) = \frac{1}{Z} \exp \left[ v^\top W h + \frac{1}{2} v^\top L v + \frac{1}{2} h^\top J h \right]. \]

Maximum Likelihood Learning:

\[
\frac{\partial \ln P(v)}{\partial W} = E_{P_{\text{data}}}[vh^\top] - E_{P_{\text{model}}}[vh^\top].
\]

\[
P_{\text{data}}(h, v) = P(h|v)P_{\text{data}}(v).
\]

Key Idea:

- Variational Inference: Approximate \( E_{P_{\text{data}}} [\cdot] \).
- Persistent MCMC: Approximate \( E_{P_{\text{model}}} [\cdot] \).
Learning

\[ P_{\text{model}}(v, h) = \frac{1}{Z} \exp \left[ v^\top W h + \frac{1}{2} v^\top L v + \frac{1}{2} h^\top J h \right]. \]

Maximum Likelihood Learning:

\[ \frac{\partial \ln P(v)}{\partial W} = E_{P_{\text{data}}}[vh^\top] - E_{P_{\text{model}}}[vh^\top]. \]

\[ P_{\text{data}}(h, v) = P(h|v)P_{\text{data}}(v). \]

Key Idea:

- Variational Inference: Approximate \( E_{P_{\text{data}}}[\cdot]. \)
- Persistent MCMC: Approximate \( E_{P_{\text{model}}}[\cdot]. \) most difficult
Stochastic Approximation

Let $p_{\theta}(x) = \frac{1}{Z(\theta)} \exp (\theta^\top \Phi(x))$

Update $\theta_0$

Update $\theta_1$

Update $\theta_2$

$X_1 \sim p_{\theta_0}(X)$

$X_2 \sim T_{\theta_1}(X|X_1)$

$X_3 \sim T_{\theta_2}(X|X_2)$

Update $x_t$ and $\theta_t$ sequentially:

- Sample $x_{t+1} \sim T_{\theta_t}(x|x_t)$ that leaves $p_{\theta_t}$ invariant.
- Update $\theta_t$ by replacing $E_{p_{\text{model}}}[\Phi(x)]$ with a point estimate $\Phi(x_{t+1})$
Stochastic Approximation

Stochastic approximation procedure: estimate $E_{P_{\text{model}}}[\cdot]$. For sufficiently small learning rate: chain will stay very close to its stationary distribution, even if it is only run for a few MCMC updates.

Almost sure convergence guarantees.
Stochastic Approximation

Stochastic approximation procedure: estimate $E_{P_{\text{model}}}[\cdot]$. 

Let $S(\theta) = \frac{\partial \log p(x;\theta)}{\partial \theta}$, then:

$$
\theta_t = \theta_{t-1} + \alpha_t S(\theta_t) + \alpha_t \left( E_{P_{\text{model}}}[\Phi(x)] - \frac{1}{M} \sum_{m=1}^{M} \Phi(x^{t+1,m}) \right)
$$

ODF $\dot{\theta} = S(\theta)$

noise term $\epsilon_{t+1}$
Variational Inference

**Variational Lower Bound:** (approximate $E_{P_{\text{data}}}[\cdot]$)

\[
\log P(v) = \log \sum_h P(v, h) \\
\geq \sum_h Q(h|v) \log P^*(v, h) - \log \mathcal{Z} + \mathcal{H}[Q(h|v)].
\]

Bound is tight iff $Q(h|v) = P(h|v)$.

Naive mean-field approach. Fully factorized distribution:

\[
Q(h|v) = \prod_j q(h_j).
\]
Learning BM’s

\[ \log P(v; \theta) \geq \sum_h Q(h|v) \log P^*(v, h; \theta) - \log \mathcal{Z}(\theta) + \mathcal{H}[Q(h|v)] \]

\[ + v^\top W h + \frac{1}{2} v^\top L v + \frac{1}{2} h^\top J h \]

For each iteration of learning:

1. **Variational Inference**: Maximize the lower bound w.r.t. variational parameters for fixed \( \theta \).

2. **MCMC**: Apply stochastic approximation algorithm to update the model parameters \( \theta \).
MNIST

500 hidden and 784 visible units (820,000 parameters).

Samples were generated by running the Gibbs sampler for 100,000 steps.
Deep Boltzmann Machines

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

Complex representations.

Fast greedy initialization by training a stack of RBM’s.

High-level representations are built from unlabeled inputs.
Labeled data is used to only slightly fine-tune the model.
MNIST

1000 units

500 units

28 x 28 pixel image

0.9 million parameters,
60,000 training and 10,000 test examples.

Discriminative fine-tuning: test error of 0.95%.
DBN’s get 1.2%, SVM’s get 1.4%, backprop gets 1.6%.
Learning DBM’s

$1^{st}$-layer features

$2^{nd}$-layer features
An experiment


Need an estimate of $Z$. 

DBM samples

Mixture of Bernoulli’s
Annealing

$$P(v; \beta) = \frac{1}{Z(\beta)} P(v; \theta)^\beta \pi(v)^{(1-\beta)}$$

Related to annealing or tempering, $1/\beta = “temperature”$
Evaluating DBM’s

\[ \hat{Z}_{\text{AIS}} \approx Z = \sum_{v, h^1, h^2, h^3} \exp \left[ v^\top W^1 h + h^1^\top W^2 h^2 + h^2^\top W^3 h^3 \right] . \]

Use \( \hat{Z}_{\text{AIS}} \) to estimate variational lower bound on the test cases.

\[
\log P(v) \geq \sum_h Q(h|v) \log P^*(v, h) - \log Z + \mathcal{H}[Q(h|v)].
\]

Other ways of estimating \( Z \)?
Benchmark experiment

MoB, test log-prob: \(-137.64\) (per digit).
DBM, test log-prob (LB): \(-84.62\) (per digit).

Difference of over 50 nats is quite large.
Gaussian-Bernoulli RBM’s

\[ P(v) = \frac{1}{Z} \sum_h \exp \left[ -\frac{1}{2} \sum_i \frac{v_i^2}{\sigma_i^2} + \sum_{ij} \frac{v_i W_{ij} h_j}{\sigma_i} \right] \]

Multinomial, Poisson, Exponential.
NORB data

5 object categories, 5 different objects within each category, 6 lighting conditions, 9 elevations, 18 azimuth.

24,300 training and 24,300 test cases.
Deep Boltzmann Machines

4000 units

4000 units

4000 units

Stereo pair

About 68 million parameters.
Discriminative fine-tuning: test error of 7.2%.

SVM’s get 11.6%, logistic regression gets 22.5%.
Image Completion
The Reuters Corpus: 804,414 newswire stories.
Simple “bag-of-words” representation.
Thank you.
Deterministic Learning

\[
\log P(v; \theta) \geq \sum_h Q(h|v) \log P^*(v, h; \theta) - \log Z(\theta) + \mathcal{H}[Q(h|v)].
\]

\[
\log Z(\theta) \leq \sup_{\mu \in \mathcal{M}_L} (\theta^\top \mu + \tilde{\mathcal{H}}(\mu)).
\]

For each iteration of learning:

1. **Variational Inference:** Maximize the lower bound w.r.t. variational parameters for fixed \( \theta \).

2. **TRBP:** Run TRBP and update the model parameters \( \theta \).

   “Monotonic” increase of the lower bound on the log-prob.
Deterministic Learning

USPS 16 × 16 digits, 20 hidden units.
Deterministic vs. Stochastic

USPS 16 × 16 digits, 20 hidden units.
DBM’s vs. DBN’s

Deep Boltzmann Machine

Deep Belief Network