Deep Generative Models for Sequence Learning

Zhe Gan
Advisor: Dr. Lawrence Carin

Ph.D. Qualifying Exam

December 2nd, 2015
Outline

1. Literature Review
   - State Space Models
   - Recurrent Neural Networks
   - Temporal Restricted Boltzmann Machines

2. Deep Temporal Sigmoid Belief Networks
   - Model Formulation
   - Scalable Learning and Inference
   - Experiments

3. Conclusion and Future work
Table of Contents

1 Literature Review
   • State Space Models
   • Recurrent Neural Networks
   • Temporal Restricted Boltzmann Machines

2 Deep Temporal Sigmoid Belief Networks
   • Model Formulation
   • Scalable Learning and Inference
   • Experiments

3 Conclusion and Future work
State Space Models

- We are interested in developing probabilistic models for sequential data.
- Hidden Markov Models (HMMs) [12]: a mixture model, multinomial latent variables
- Linear Dynamical Systems (LDS) [8]: continuous latent variables

Figure: Graphical model for HMM and LDS.
Table of Contents

1 Literature Review
   - State Space Models
   - Recurrent Neural Networks
   - Temporal Restricted Boltzmann Machines

2 Deep Temporal Sigmoid Belief Networks
   - Model Formulation
   - Scalable Learning and Inference
   - Experiments

3 Conclusion and Future work
Recurrent Neural Networks

- Recurrent Neural Networks (RNNs) takes a sequence as input.
- Recursively processing each input $v_t$ while maintaining its internal hidden state $h_t$.

$$h_t = f(v_{t-1}, h_{t-1})$$  \hspace{1cm} (1)

where $f$ is a deterministic non-linear transition function.

Figure: Illustration for RNNs.
Recurrent Neural Networks

- RNN defines the likelihood as

\[
p(x_1, \ldots, x_T) = \prod_{t=1}^{T} p(x_t|x_{<t}), \quad p(x_t|x_{<t}) = g(h_t) \tag{2}
\]

- How to define \( f \)?
  - Logistic function
    \[
    h_t = \sigma(Wx_{t-1} + Uh_{t-1} + b) \tag{3}
    \]
  - Long Short-Term Memory (LSTM) [7]
  - Gated Recurrent Units (GRU) [3]

- All the randomness inside the model lies in the usage of the conditional probability \( p(x_t|x_{<t}) \).
Table of Contents

1 Literature Review
   • State Space Models
   • Recurrent Neural Networks
   • Temporal Restricted Boltzmann Machines

2 Deep Temporal Sigmoid Belief Networks
   • Model Formulation
   • Scalable Learning and Inference
   • Experiments

3 Conclusion and Future work
Restricted Boltzmann Machines

RBM [6] is an *undirected* graphical model that represents a joint distribution over binary visible units $v \in \{0, 1\}^M$ and binary hidden units $h \in \{0, 1\}^J$ as

$$P(v, h) = \exp\{-E(v, h)/Z\}$$

(4)

$$E(v, h) = -(h^T W v + c^T v + b^T h)$$

(5)
Restricted Boltzmann Machines

- The posterior for $v$ and $h$ are factorized.

\[
P(h_j = 1|v) = \sigma \left( \sum_m w_{jm} v_m + b_j \right) \tag{6}
\]

\[
P(v_i = 1|h) = \sigma \left( \sum_m w_{jm} h_j + c_m \right) \tag{7}
\]

- **Inference**: block Gibbs sampling.
- **Learning**: Contrastive-Divergence (CD).
- **Application**:
  1. Deep Belief Networks (DBNs) and Deep Boltzmann Machines (DBMs)
  2. Learn the sequential dependencies in time-series data [9, 13, 14, 15, 16].
Temporal Restricted Boltzmann Machines

- TRBM [13] is defined as a sequence of RBMs

\[ p(V, H) = p_0(v_1, h_1) \prod_{t=2}^{T} p(v_t, h_t|h_{t-1}) \]  \hspace{1cm} (8)

- Each \( p(v_t, h_t|h_{t-1}) \) is defined as a conditional RBM.

\[ p(v_t, h_t|h_{t-1}) \propto \exp(v_t^\top W h_t + v_t^\top c + h_t^\top (b + U^\top h_{t-1})) \]

**Figure:** Graphical model for TRBM.
Model Reviews

<table>
<thead>
<tr>
<th>Properties</th>
<th>HMM</th>
<th>LDS</th>
<th>RNN</th>
<th>TRBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directed</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Latent</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Distributed$^1$</td>
<td>✗</td>
<td>—</td>
<td>—</td>
<td>✓</td>
</tr>
</tbody>
</table>

Can we find a model that has all the properties listed above? A deep directed latent variable model.

$^1$Each hidden state in HMM is a one-hot vector. To represent $2^N$ distinct states, an HMM requires a length-$2^N$ one-hot vector, while for TRBM, only a length-$N$ vector is needed.
Table of Contents

1 Literature Review
   • State Space Models
   • Recurrent Neural Networks
   • Temporal Restricted Boltzmann Machines

2 Deep Temporal Sigmoid Belief Networks
   • Model Formulation
   • Scalable Learning and Inference
   • Experiments

3 Conclusion and Future work
Overview

- **Problem of interest:** Developing deep directed latent variable models for sequential data.

- **Main idea:**
  1. Constructing a hierarchy of Temporal Sigmoid Belief Networks (TSBNs).
  2. TSBN is defined as a sequential stack of Sigmoid Belief Networks (SBNs).

- **Challenge:** Designing scalable learning and inference algorithms.

- **Solution:**
  1. Stochastic Variational Inference (SVI).
  2. Design a recognition model for fast inference.
An SBN [11] models a visible vector $v \in \{0, 1\}^M$, in terms of hidden variables $h \in \{0, 1\}^J$ and weights $W \in \mathbb{R}^{M \times J}$ with

$$p(v_m = 1|h) = \sigma(w_m^\top h + c_m), \quad p(h_j = 1) = \sigma(b_j)$$

(9)

where $\sigma(x) \triangleq 1/(1 + e^{-x})$. 

$$\begin{align*}
\text{Sigmoid Belief Networks} \\
\text{An SBN [11] models a visible vector } v \in \{0, 1\}^M, \text{ in terms of} \\
\text{hidden variables } h \in \{0, 1\}^J \text{ and weights } W \in \mathbb{R}^{M \times J} \text{ with} \\
p(v_m = 1|h) = \sigma(w_m^\top h + c_m), \quad p(h_j = 1) = \sigma(b_j) \quad (9)
\end{align*}$$
SBN vs. RBM

- **Graphical Model**
  
  ![Graphical Model Diagram]

- **Energy function**

  \[-E_{SBN}(v, h) = v^\top c + v^\top Wh + h^\top b - \sum_m \log(1 + e^{w_m^\top h + c_m})\]

  \[-E_{RBM}(v, h) = v^\top c + v^\top Wh + h^\top b\]

- **How to generate data**
  1. **SBN**: ancestral sampling
  2. **RBM**: iterative Gibbs sampling
SBN vs. RBM

- **Inference methods**
  1. **SBN**: Gibbs sampling, mean-field VB, Recognition model
  2. **RBM**: Gibbs sampling

- **Learning methods**
  1. **SBN**: SGD, Gibbs sampling, mean-field VB, MCEM
     (Polya-Gamma data augmentation)
  2. **RBM**: Contrastive Divergence

- SBN has been shown potential to build deep models [4, 5].
  1. Binary image modeling
  2. Topic modeling
Temporal Sigmoid Belief Networks

- TSBN is defined as a sequence of SBNs.

\[
p_\theta(V, H) = p(h_1)p(v_1|h_1) \cdot \prod_{t=2}^{T} p(h_t|h_{t-1}, v_{t-1}) \cdot p(v_t|h_t, v_{t-1})
\]

- For \( t = 1, \ldots, T \), each conditional distribution is expressed as

\[
p(h_{jt} = 1|h_{t-1}, v_{t-1}) = \sigma(w_{1j}h_{t-1} + w_{3j}v_{t-1} + b_j) \quad (10)
\]

\[
p(v_{mt} = 1|h_t, v_{t-1}) = \sigma(w_{2m}h_t + w_{4m}v_{t-1} + c_m) \quad (11)
\]

Figure: Graphical model for TSBN.
Our TSBN model can be viewed as:

1. an HMM with distributed hidden state representations;
2. a LDS with non-linear dynamics;
3. a probabilistic construction of RNN;
4. a directed-graphical-model counterpart of TRBM.

Each hidden state in the HMM is represented as a *one-hot* length-$J$ vector, while in the TSBN, the hidden states can be any length-$J$ binary vector.
TSBN Variants: Modeling Different Data Types

- **Real-valued data:** \( p(\mathbf{v}_t | \mathbf{h}_t, \mathbf{v}_{t-1}) = \mathcal{N}(\mu_t, \text{diag}(\sigma_t^2)) \), where
  \[
  \mu_{mt} = \mathbf{w}_{2m}^\top \mathbf{h}_t + \mathbf{w}_{4m}^\top \mathbf{v}_{t-1} + c_m, \tag{12}
  \]
  \[
  \log \sigma_{mt}^2 = (\mathbf{w}'_{2m})^\top \mathbf{h}_t + (\mathbf{w}'_{4m})^\top \mathbf{v}_{t-1} + c'_m \tag{13}
  \]

- **Count data:** \( p(\mathbf{v}_t | \mathbf{h}_t, \mathbf{v}_{t-1}) = \prod_{m=1}^M y_{vmt} \), where
  \[
  y_{mt} = \frac{\exp(\mathbf{w}_{2m}^\top \mathbf{h}_t + \mathbf{w}_{4m}^\top \mathbf{v}_{t-1} + c_m)}{\sum_{m'=1}^M \exp(\mathbf{w}_{2m'}^\top \mathbf{h}_t + \mathbf{w}_{4m'}^\top \mathbf{v}_{t-1} + c_{m'})}. \tag{14}
  \]
TSBN Variants: Boosting Performance

- **High Order:** $h_t, v_t$ depends on $h_{t-n:t-1}, v_{t-n:t-1}$.
- **Going Deep:**
  1. adding stochastic hidden layers;
  2. adding deterministic hidden layers.
- Conditional independent “becomes” conditional dependent.

(Top) Shallow TSBN
(Right) A 2-layer TSBN
Table of Contents

1 Literature Review
   • State Space Models
   • Recurrent Neural Networks
   • Temporal Restricted Boltzmann Machines

2 Deep Temporal Sigmoid Belief Networks
   • Model Formulation
   • Scalable Learning and Inference
   • Experiments

3 Conclusion and Future work
Scalable Learning and Inference

- By using Polya-Gamma, Gibbs sampling or mean-field VB can be implemented [5, 11], but are inefficient.
- To allow for
  1. tractable and scalable inference and parameter learning
  2. without loss of the flexibility of the variational posterior

We apply recent advances in stochastic variational Bayes for non-conjugate inference [10].
Recognition model

- Variational Lower Bound

\[ \mathcal{L}(\mathbf{V}, \theta, \phi) = \mathbb{E}_{q_{\phi}(\mathbf{H}|\mathbf{V})} \left[ \log p_{\theta}(\mathbf{V}, \mathbf{H}) - \log q_{\phi}(\mathbf{H}|\mathbf{V}) \right]. \quad (15) \]

- How to design \( q_{\phi}(\mathbf{H}|\mathbf{V}) \)? No mean-field VB!
- **Recognition model:** introduce a fixed-form distribution \( q_{\phi}(\mathbf{H}|\mathbf{V}) \) to approximate the true posterior \( p(\mathbf{H}|\mathbf{V}) \).
  1. Fast inference
  2. Utilization of a global set of parameters
  3. Potential better fit of the data
Recognition model is also defined as a TSBN.

\[ q_\phi(H|V) = q(h_1|v_1) \cdot \prod_{t=2}^{T} q(h_t|h_{t-1}, v_t, v_{t-1}), \quad (16) \]
Parameter Learning

- **Gradients:**

\[ \nabla_\theta \mathcal{L}(V) = \mathbb{E}_{q_\phi} [\nabla_\theta \log p_\theta(V, H)] \]
\[ \nabla_\phi \mathcal{L}(V) = \mathbb{E}_{q_\phi} [(\log p_\theta(V, H) - \log q_\phi(H|V)) \times \nabla_\phi \log q_\phi(H|V)] \]

- **Variance Reduction:**
  1. centering the learning signal
  2. variance normalization
Table of Contents

1 Literature Review
   • State Space Models
   • Recurrent Neural Networks
   • Temporal Restricted Boltzmann Machines

2 Deep Temporal Sigmoid Belief Networks
   • Model Formulation
   • Scalable Learning and Inference
   • Experiments

3 Conclusion and Future work
Experiments

- **Datasets:**
  - Bouncing balls: **Binary**
  - Motion capture: **Real-valued**
  - Polyphonic music: **Binary**
  - State of the Union: **Count**

- **Notations:**
  1. **HMSBN:** TSBN model with $W_3 = 0$ and $W_4 = 0$;
  2. **DTSBN-S:** Deep TSBN with stochastic hidden layer;
  3. **DTSBN-D:** Deep TSBN with deterministic hidden layer.

- **Experimental Setup:**
  - Random initialization
  - RMSprop optimization
  - Monte Carlo sampling using only a single sample
  - Weight decay regularization
Experiments

- **Prediction** of $v_t$ given $v_{1:t-1}$
  1. first obtain a sample from $q_\phi(h_{1:t-1}|v_{1:t-1})$;
  2. calculate the conditional posterior $p_\theta(h_t|h_{1:t-1}, v_{1:t-1})$ of the current hidden state;
  3. make a prediction for $v_t$ using $p_\theta(v_t|h_{1:t}, v_{1:t-1})$.

- **Generation**: ancestral sampling.
Bouncing balls dataset

Each video is of length 100 and of resolution $30 \times 30$. 
Our learned dictionaries are spatially localized.
Bouncing balls dataset: Prediction

1. Our approach achieves state-of-the-art.
2. A high-order TSBN reduces the prediction error significantly.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DTSBN-s</td>
<td>100-100</td>
<td>2</td>
<td>2.79 ± 0.39</td>
<td>69.29 ± 1.52</td>
</tr>
<tr>
<td>DTSBN-d</td>
<td>100-100</td>
<td>2</td>
<td>2.99 ± 0.42</td>
<td>70.47 ± 1.52</td>
</tr>
<tr>
<td>TSBN</td>
<td>100</td>
<td>4</td>
<td>3.07 ± 0.40</td>
<td>70.41 ± 1.55</td>
</tr>
<tr>
<td>TSBN</td>
<td>100</td>
<td>1</td>
<td>9.48 ± 0.38</td>
<td>77.71 ± 0.83</td>
</tr>
<tr>
<td>RTRBM°</td>
<td>3750</td>
<td>1</td>
<td>3.88 ± 0.33</td>
<td>—</td>
</tr>
<tr>
<td>SRTRBM°</td>
<td>3750</td>
<td>1</td>
<td>3.31 ± 0.33</td>
<td>—</td>
</tr>
</tbody>
</table>
Motion capture dataset: Prediction

We used 33 running and walking sequences of subject 35 in the CMU motion capture dataset. TSBN-based models improve over the RBM-based models significantly.

<table>
<thead>
<tr>
<th>Model</th>
<th>Walking</th>
<th>Running</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTSBN-s</td>
<td>4.40 ± 0.28</td>
<td>2.56 ± 0.40</td>
</tr>
<tr>
<td>DTSBN-d</td>
<td>4.62 ± 0.01</td>
<td>2.84 ± 0.01</td>
</tr>
<tr>
<td>TSBN</td>
<td>5.12 ± 0.50</td>
<td>4.85 ± 1.26</td>
</tr>
<tr>
<td>HMSBN</td>
<td>10.77 ± 1.15</td>
<td>7.39 ± 0.47</td>
</tr>
<tr>
<td>ss-SRTRBM</td>
<td>8.13 ± 0.06</td>
<td>5.88 ± 0.05</td>
</tr>
<tr>
<td>g-RTRBM</td>
<td>14.41 ± 0.38</td>
<td>10.91 ± 0.27</td>
</tr>
</tbody>
</table>
Motion capture dataset: Generation

1. These generated data are readily produced from the model and demonstrate realistic behavior.

2. The smooth trajectories are walking movements, while the vibrating ones are running.

Figure: Generated motion trajectories. (Left) Walking. (Middle) Running-running-walking. (Right) Running-walking.
Polyphonic music dataset: Prediction

Four polyphonic music sequences of piano [2]: Piano-midi.de, Nottingham, MuseData and JSB chorales.

Table: Test log-likelihood for music datasets. (◊) taken from [2].

<table>
<thead>
<tr>
<th>Model</th>
<th>Piano.</th>
<th>Nott.</th>
<th>Muse.</th>
<th>JSB.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSBN</td>
<td>-7.98</td>
<td>-3.67</td>
<td>-6.81</td>
<td>-7.48</td>
</tr>
<tr>
<td>RNN-NADE◊</td>
<td>-7.05</td>
<td>-2.31</td>
<td>-5.60</td>
<td>-5.56</td>
</tr>
<tr>
<td>RTRBM◊</td>
<td>-7.36</td>
<td>-2.62</td>
<td>-6.35</td>
<td>-6.35</td>
</tr>
<tr>
<td>RNN◊</td>
<td>-8.37</td>
<td>-4.46</td>
<td>-8.13</td>
<td>-8.71</td>
</tr>
</tbody>
</table>
Polyphonic music dataset: Generation

We can generate different styles of music based on different training datasets.

**Figure:** Generated samples.
Prediction is concerned with estimating the held-out words.

MP: *mean precision* over all the years that appear.

PP: *predictive precision* for the final year.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dim</th>
<th>MP</th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMSBN</td>
<td>25</td>
<td>0.327 ± 0.002</td>
<td>0.353 ± 0.070</td>
</tr>
<tr>
<td>DHMSBN-s</td>
<td>25-25</td>
<td>0.299 ± 0.001</td>
<td>0.378 ± 0.006</td>
</tr>
<tr>
<td>GP-DPFA°</td>
<td>100</td>
<td>0.223 ± 0.001</td>
<td>0.189 ± 0.003</td>
</tr>
<tr>
<td>DRFM°</td>
<td>25</td>
<td>0.217 ± 0.003</td>
<td>0.177 ± 0.010</td>
</tr>
</tbody>
</table>
The learned trajectory exhibits different temporal patterns across the topics.

- **Topic 29**: Nicaragua v. U.S.
- **Topic 30**: War of 1812, World War II, Iraq War
- **Topic 130**: The age of American revolution
State of the Union dataset: Dynamic Topic Modeling

**Table:** Top 10 most probable words associated with the STU topics.

<table>
<thead>
<tr>
<th>Topic #29</th>
<th>Topic #30</th>
<th>Topic #130</th>
<th>Topic #64</th>
<th>Topic #70</th>
<th>Topic #74</th>
</tr>
</thead>
<tbody>
<tr>
<td>family</td>
<td>officer</td>
<td>government</td>
<td>generations</td>
<td>Iraqi</td>
<td>Philippines</td>
</tr>
<tr>
<td>budget</td>
<td>civilized</td>
<td>country</td>
<td>generation</td>
<td>Qaida</td>
<td>islands</td>
</tr>
<tr>
<td>Nicaragua</td>
<td>warfare</td>
<td>public</td>
<td>recognize</td>
<td>Iraq</td>
<td>axis</td>
</tr>
<tr>
<td>free</td>
<td>enemy</td>
<td>law</td>
<td>brave</td>
<td>Iraqis</td>
<td>Nazis</td>
</tr>
<tr>
<td>future</td>
<td>whilst</td>
<td>present</td>
<td>crime</td>
<td>Al</td>
<td>Japanese</td>
</tr>
<tr>
<td>freedom</td>
<td>gained</td>
<td>citizens</td>
<td>race</td>
<td>Saddam</td>
<td>Germans</td>
</tr>
<tr>
<td>excellence</td>
<td>lake</td>
<td>united</td>
<td>balanced</td>
<td>ballistic</td>
<td>mines</td>
</tr>
<tr>
<td>drugs</td>
<td>safety</td>
<td>house</td>
<td>streets</td>
<td>terrorists</td>
<td>sailors</td>
</tr>
<tr>
<td>families</td>
<td>American</td>
<td>foreign</td>
<td>college</td>
<td>hussein</td>
<td>Nazi</td>
</tr>
<tr>
<td>God</td>
<td>militia</td>
<td>gentlemen</td>
<td>school</td>
<td>failures</td>
<td>hats</td>
</tr>
</tbody>
</table>
Conclusion and Future work

Conclusion

1. Proposed Temporal Sigmoid Belief Networks
2. Developed an efficient variational optimization algorithm
3. Extensive applications to diverse data types

Future work

1. Controlled style transitioning
2. Language modeling
3. Multi-modality learning
A. Acharya, J. Ghosh, and M. Zhou.
Nonparametric Bayesian factor analysis for dynamic count matrices.
In *AISTATS*, 2015.

N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent.
Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription.
In *ICML*, 2012.

Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Scalable deep poisson factor analysis for topic modeling.
In *ICML*, 2015.

Learning deep sigmoid belief networks with data augmentation.
In *AISTATS*, 2015.

G. Hinton.
Training products of experts by minimizing contrastive divergence.
In *Neural computation*, 2002.
References II

Sepp Hochreiter and Jürgen Schmidhuber.
Long short-term memory.

R. Kalman.
Mathematical description of linear dynamical systems.

R. Mittelman, B. Kuipers, S. Savarese, and H. Lee.
Structured recurrent temporal restricted boltzmann machines.
In *ICML*, 2014.

A. Mnih and K. Gregor.
Neural variational inference and learning in belief networks.
In *ICML*, 2014.

R. Neal.
Connectionist learning of belief networks.

L. Rabiner and B. Juang.
An introduction to hidden markov models.

Learning multilevel distributed representations for high-dimensional sequences.
In *AISTATS*, 2007.
The recurrent temporal restricted boltzmann machine.
In *NIPS*, 2009.

G. Taylor and G. Hinton.
Factored conditional restricted boltzmann machines for modeling motion style.
In *ICML*, 2009.

Modeling human motion using binary latent variables.
In *NIPS*, 2006.
Contrastive Divergence:

$$\frac{\partial P(v)}{\partial \theta} = \mathbb{E}_{p(v,h)} \left[ \frac{\partial}{\partial \theta} E(v, h) \right] - \mathbb{E}_{p(h|v)} \left[ \frac{\partial}{\partial \theta} E(v, h) \right] \quad (17)$$
The contribution of $\mathbf{v}$ to the log-likelihood can be lower-bounded as follows

\[
\log p_\theta = \log \sum_h p_\theta(\mathbf{v}, \mathbf{h}) \geq \sum_h q_\phi(h|\mathbf{v}) \log \frac{p_\theta(\mathbf{v}, \mathbf{h})}{q_\phi(h|\mathbf{v})} = \mathbb{E}_{q_\phi(h|\mathbf{v})}[\log p_\theta(\mathbf{v}, \mathbf{h}) - \log q_\phi(h|\mathbf{v})] = \mathcal{L}(\mathbf{v}, \theta, \phi)
\]  

We can also rewrite the bound as

\[
\mathcal{L}(\mathbf{v}, \theta, \phi) = \log p_\theta(\mathbf{v}) - KL(q_\phi(h|\mathbf{v}), p_\theta(h|\mathbf{v}))
\]
Differentiating the variational lower bound w.r.t. to the recognition model parameters gives

$$\nabla_\phi \mathcal{L}(v) = \nabla_\phi \mathbb{E}_q[\log p_\theta(v, h) - \log q_\phi(h|v)]$$  \hspace{1cm} (23)

$$= \nabla_\phi \sum_h q_\phi(h|v) \log p_\theta(v, h) - \nabla_\phi \sum_h q_\phi(h|v) \log q_\phi(h|v)$$

$$= \sum_h \log p_\theta(v, h) \nabla_\phi q_\phi(h|v) - \sum_h (\log q_\phi(h|v) + 1) \nabla_\phi q_\phi(h|v)$$

$$= \sum_h (\log p_\theta(v, h) - \log q_\phi(h|v)) \nabla_\phi q_\phi(h|v)$$  \hspace{1cm} (24)

where we have used the fact that

$$\sum_h \nabla_\phi q_\phi(h|v) = \nabla_\phi \sum_h q_\phi(h|v) = \nabla_\phi 1 = 0.$$  

Using the identity

$$\nabla_\phi q_\phi(h|v) = q_\phi(h|v) \nabla_\phi \log q_\phi(h|v),$$

then gives

$$\nabla_\phi \mathcal{L}(v) = \mathbb{E}_q[(\log p_\theta(v, h) - \log q_\phi(h|v)) \times \nabla_\phi \log q_\phi(h|v)]$$  \hspace{1cm} (25)
\[ E_q[\nabla \phi \log q_\phi(h|v)] = E_q \left[ \frac{\nabla \phi q_\phi(h|v)}{q_\phi(h|v)} \right] \] 

(26)

\[ = \sum_h \nabla \phi q_\phi(h|v) \] 

(27)

\[ = \nabla \phi \sum_h q_\phi(h|v) \] 

(28)

\[ = \nabla \phi 1 \] 

(29)

\[ = 0 \] 

(30)
Algorithm 1 Compute gradient estimates.

\[
\begin{align*}
\Delta \theta & \leftarrow 0, \Delta \phi \leftarrow 0, \Delta \lambda \leftarrow 0, \mathcal{L} \leftarrow 0 \\
\text{for } t \leftarrow 1 \text{ to } T & \text{ do} \\
& \quad h_t \sim q_\phi(h_t|v_t) \\
& \quad l_t \leftarrow \log p_\theta(v_t, h_t) - \log q_\phi(h_t|v_t) \\
& \quad \mathcal{L} \leftarrow \mathcal{L} + l_t \\
& \quad l_t \leftarrow l_t - C_\lambda(v_t) \\
\text{end for} \\
& c_b \leftarrow \text{mean}(l_1, \ldots, l_T) \\
& v_b \leftarrow \text{variance}(l_1, \ldots, l_T) \\
& c \leftarrow \alpha c + (1 - \alpha)c_b \\
& v \leftarrow \alpha v + (1 - \alpha)v_b \\
\text{for } t \leftarrow 1 \text{ to } T & \text{ do} \\
& \quad l_t \leftarrow \frac{l_t - c}{\max(1, \sqrt{v})} \\
& \quad \Delta \theta \leftarrow \Delta \theta + \nabla_\theta \log p_\theta(v_t, h_t) \\
& \quad \Delta \phi \leftarrow \Delta \phi + l_t \nabla_\phi \log q_\phi(h_t|v_t) \\
& \quad \Delta \lambda \leftarrow \Delta \lambda + l_t \nabla_\lambda C_\lambda(v_t) \\
\text{end for}
\]