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

State Space Models

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

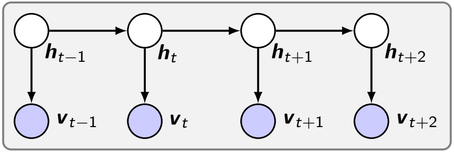


Figure: Graphical model for HMM and LDS.

Table of Contents

Recurrent Neural Networks

- Recurrent Neural Neworks (RNNs) takes a sequence as input.
- Recursively processing each input $\boldsymbol{v}_t$ while maintaining its internal hidden state $\boldsymbol{h}_t$.

$$\boldsymbol{h}_t = f(\boldsymbol{v}_{t-1}, \boldsymbol{h}_{t-1}) \tag{1}$$

where f is a deterministic non-linear transition function.

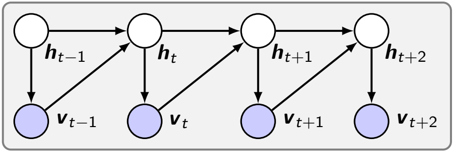


Figure: Illustration for RNNs.

Recurrent Neural Networks

- RNN defines the likelihood as

$$p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T) = \prod_{t=1}^{T} p(\boldsymbol{x}_t | \boldsymbol{x}_{<t}), \quad p(\boldsymbol{x}_t | \boldsymbol{x}_{<t}) = g(\boldsymbol{h}_t) \qquad (2)$$

- How to define f ?
  1. Logistic function

$$\boldsymbol{h}_t = \sigma(\mathbf{W}\boldsymbol{x}_{t-1} + \mathbf{U}\boldsymbol{h}_{t-1} + \boldsymbol{b}) \qquad (3)$$

  2. Long Short-Term Memory (LSTM) [7]
  3. Gated Recurrent Units (GRU) [3]
- All the randomness inside the model lies in the usage of the conditional probability $p(\boldsymbol{x}_t | \boldsymbol{x}_{<t})$.

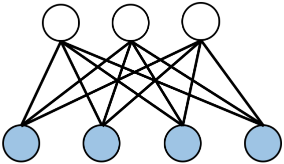
Table of Contents

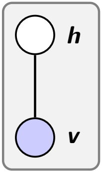
Restricted Boltzmann Machines

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

$$P(\boldsymbol{v}, \boldsymbol{h}) = \exp\{-E(\boldsymbol{v}, \boldsymbol{h})\}/Z \tag{4}$$

$$E(\boldsymbol{v}, \boldsymbol{h}) = -(\boldsymbol{h}^\top \mathbf{W} \boldsymbol{v} + \boldsymbol{c}^\top \boldsymbol{v} + \boldsymbol{b}^\top \boldsymbol{h}) \tag{5}$$





Restricted Boltzmann Machines

- The posterior for $\boldsymbol{v}$ and $\boldsymbol{h}$ are factorized.

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

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

- 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

$$p(\mathbf{V}, \mathbf{H}) = p_0(\boldsymbol{v}_1, \boldsymbol{h}_1) \prod_{t=2}^{T} p(\boldsymbol{v}_t, \boldsymbol{h}_t | \boldsymbol{h}_{t-1}) \tag{8}$$

- Each $p(\boldsymbol{v}_t, \boldsymbol{h}_t | \boldsymbol{h}_{t-1})$ is defined as a conditional RBM.

$$p(\boldsymbol{v}_t, \boldsymbol{h}_t | \boldsymbol{h}_{t-1}) \propto \exp(\boldsymbol{v}_t^\top \mathbf{W} \boldsymbol{h}_t + \boldsymbol{v}_t^\top \boldsymbol{c} + \boldsymbol{h}_t^\top (\boldsymbol{b} + \mathbf{U}^\top \boldsymbol{h}_{t-1}))$$

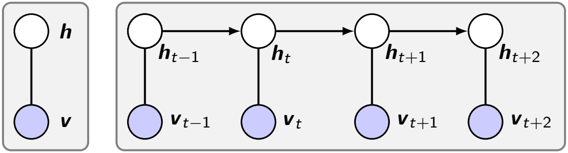


Figure: Graphical model for TRBM.

Model Reviews

Propeties	HMM	LDS	RNN	TRBM
Directed	✓	✓	✓	✗
Latent	✓	✓	✗	✓
Nonlinear	✗	✗	✓	✓
Distributed[1]	✗	—	—	✓

Can we find a model that has all the propeties 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

Overview

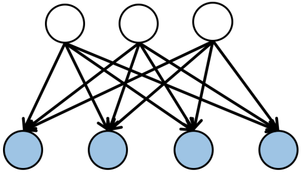
- **Problem of interest:** Developing deep directed latent variable models for sequential data.
- **Main idea:**
  1. Constructing a hierarchy of Temporal Sigmoid Belief Networks (TSBNs).
  2. TSBN is defined as a sequetial stack of Sigmoid Belief Networks (SBNs).
- **Challenge:** Designing scalable learning and inference algorithms.
- **Solution:**
  1. Stochastic Variational Inference (SVI).
  2. Design a recognition model for fast inference.

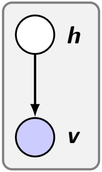
Sigmoid Belief Networks

An SBN [11] models a visible vector $\boldsymbol{v} \in \{0,1\}^M$, in terms of hidden variables $\boldsymbol{h} \in \{0,1\}^J$ and weights $\mathbf{W} \in \mathbb{R}^{M \times J}$ with

$$p(v_m = 1|\boldsymbol{h}) = \sigma(\boldsymbol{w}_m^\top \boldsymbol{h} + c_m), \qquad p(h_j = 1) = \sigma(b_j) \tag{9}$$

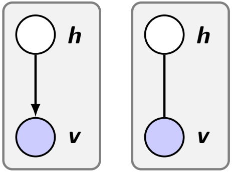
where $\sigma(x) \triangleq 1/(1 + e^{-x})$.





SBN vs. RBM

- Graphical Model



- Energy function

$$-E_{SBN}(\boldsymbol{v},\boldsymbol{h}) = \boldsymbol{v}^\top \boldsymbol{c} + \boldsymbol{v}^\top \mathbf{W}\boldsymbol{h} + \boldsymbol{h}^\top \boldsymbol{b} - \sum_m \log(1 + e^{\boldsymbol{w}_m^\top \boldsymbol{h} + c_m})$$

$$-E_{RBM}(\boldsymbol{v},\boldsymbol{h}) = \boldsymbol{v}^\top \boldsymbol{c} + \boldsymbol{v}^\top \mathbf{W}\boldsymbol{h} + \boldsymbol{h}^\top \boldsymbol{b}$$

- 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.

$$p_\theta(\mathbf{V}, \mathbf{H}) = p(\boldsymbol{h}_1)p(\boldsymbol{v}_1|\boldsymbol{h}_1) \cdot \prod_{t=2}^{T} p(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{v}_{t-1}) \cdot p(\boldsymbol{v}_t|\boldsymbol{h}_t, \boldsymbol{v}_{t-1})$$

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

$$p(h_{jt} = 1|\boldsymbol{h}_{t-1}, \boldsymbol{v}_{t-1}) = \sigma(\boldsymbol{w}_{1j}^\top \boldsymbol{h}_{t-1} + \boldsymbol{w}_{3j}^\top \boldsymbol{v}_{t-1} + b_j) \quad (10)$$

$$p(v_{mt} = 1|\boldsymbol{h}_t, \boldsymbol{v}_{t-1}) = \sigma(\boldsymbol{w}_{2m}^\top \boldsymbol{h}_t + \boldsymbol{w}_{4m}^\top \boldsymbol{v}_{t-1} + c_m) \quad (11)$$

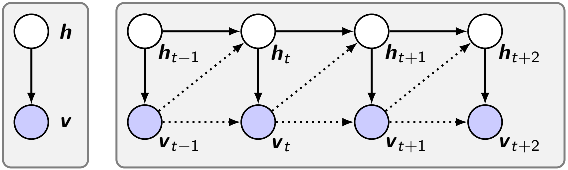


Figure: Graphical model for TSBN.

TSBN Properties

- Our TSBN model can be viewed as :
  1. an HMM with distributed hidden state representations[2];
  2. a LDS with non-linear dynamics;
  3. a probabilistic construction of RNN;
  4. a directed-graphical-model counterpart of TRBM.

[2] 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: $p(\boldsymbol{v}_t|\boldsymbol{h}_t, \boldsymbol{v}_{t-1}) = \mathcal{N}(\boldsymbol{\mu}_t, \mathrm{diag}(\boldsymbol{\sigma}_t^2))$, where

$$\mu_{mt} = \boldsymbol{w}_{2m}^\top \boldsymbol{h}_t + \boldsymbol{w}_{4m}^\top \boldsymbol{v}_{t-1} + c_m, \tag{12}$$

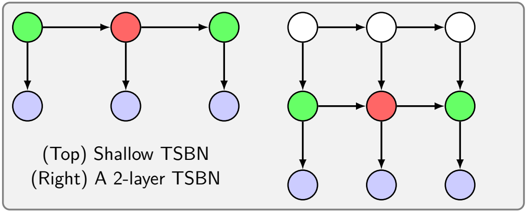
$$\log \sigma_{mt}^2 = (\boldsymbol{w}_{2m}')^\top \boldsymbol{h}_t + (\boldsymbol{w}_{4m}')^\top \boldsymbol{v}_{t-1} + c_m' \tag{13}$$

- Count data: $p(\boldsymbol{v}_t|\boldsymbol{h}_t, \boldsymbol{v}_{t-1}) = \prod_{m=1}^M y_{mt}^{v_{mt}}$, where

$$y_{mt} = \frac{\exp(\boldsymbol{w}_{2m}^\top \boldsymbol{h}_t + \boldsymbol{w}_{4m}^\top \boldsymbol{v}_{t-1} + c_m)}{\sum_{m'=1}^M \exp(\boldsymbol{w}_{2m'}^\top \boldsymbol{h}_t + \boldsymbol{w}_{4m'}^\top \boldsymbol{v}_{t-1} + c_{m'})} . \tag{14}$$

TSBN Variants: Boosting Performance

- High Order: $\boldsymbol{h}_t, \boldsymbol{v}_t$ depends on $\boldsymbol{h}_{t-n:t-1}, \boldsymbol{v}_{t-n:t-1}$.
- 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

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

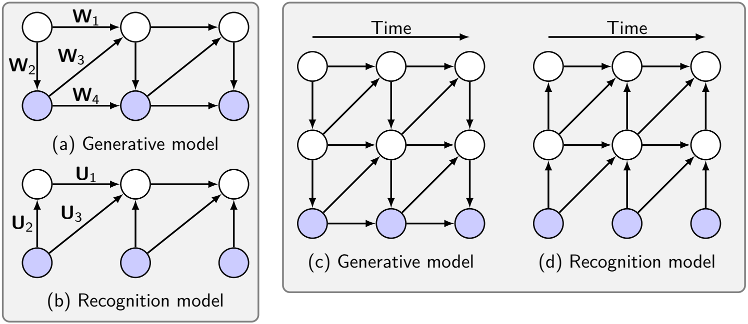
$$\mathcal{L}(\mathbf{V}, \boldsymbol{\theta}, \phi) = \mathbb{E}_{q_\phi(\mathbf{H}|\mathbf{V})}[\log p_{\boldsymbol{\theta}}(\mathbf{V}, \mathbf{H}) - \log q_\phi(\mathbf{H}|\mathbf{V})] \,. \tag{15}$$

- How to design $q_\phi(\mathbf{H}|\mathbf{V})$? No mean-field VB!
- Recogniton 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

Recognition model is also defined as a TSBN.

$$q_\phi(\mathbf{H}|\mathbf{V}) = q(\boldsymbol{h}_1|\boldsymbol{v}_1) \cdot \prod_{t=2}^{T} q(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{v}_t, \boldsymbol{v}_{t-1}) \,, \tag{16}$$



(a) Generative model
(b) Recognition model
(c) Generative model
(d) Recognition model

Parameter Learning

- Gradients:

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

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

- Variance Reduction:
  1. centering the learning signal
  2. variance normalization

Table of Contents

Experiments

- Datasets:

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

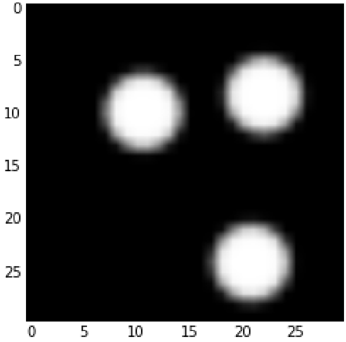
- Notations:
  1. *HMSBN*: TSBN model with $\mathbf{W}_3 = \mathbf{0}$ and $\mathbf{W}_4 = \mathbf{0}$;
  2. *DTSBN-S*: Deep TSBN with stochastic hidden layer;
  3. *DTSBN-D*: Deep TSBN with deterministic hidden layer.
- Experimental Setup:
  - Random initialization
  - RMSprop optimization
  - Monte Carlo sampling using only a single sample
  - Weight decay regularization

Experiments

- Prediction of $\boldsymbol{v}_t$ given $\boldsymbol{v}_{1:t-1}$
  1. first obtain a sample from $q_\phi(\boldsymbol{h}_{1:t-1}|\boldsymbol{v}_{1:t-1})$;
  2. calculate the conditional posterior $p_\theta(\boldsymbol{h}_t|\boldsymbol{h}_{1:t-1}, \boldsymbol{v}_{1:t-1})$ of the current hidden state ;
  3. make a prediction for $\boldsymbol{v}_t$ using $p_\theta(\boldsymbol{v}_t|\boldsymbol{h}_{1:t}, \boldsymbol{v}_{1:t-1})$.
- Generation: ancestral sampling.

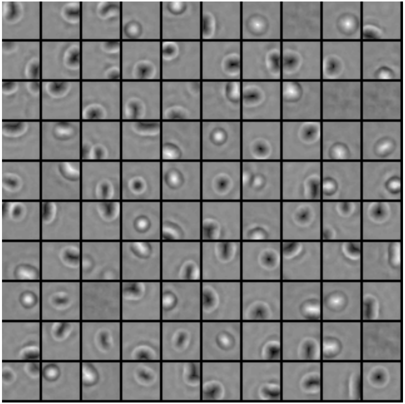
Bouncing balls dataset

Each video is of length 100 and of resolution 30×30.



Bouncing balls dataset: Dictionaries

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.
3. Using multiple layers improve performance.

Model	Dim	Order	Pred. Err.	Neg. Log. Like.
DTSBN-s	100-100	2	**2.79** ± 0.39	**69.29** ± 1.52
DTSBN-d	100-100	2	2.99 ± 0.42	70.47 ± 1.52
TSBN	100	4	3.07 ± 0.40	70.41 ± 1.55
TSBN	100	1	9.48 ± 0.38	77.71 ± 0.83
RTRBM$^\diamond$	3750	1	3.88 ± 0.33	–
SRTRBM$^\diamond$	3750	1	3.31 ± 0.33	–

Motion capture dataset: Prediction

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

MODEL	WALKING	RUNNING
DTSBN-S	**4.40** $\pm$ 0.28	**2.56** $\pm$ 0.40
DTSBN-D	4.62 $\pm$ 0.01	2.84 $\pm$ 0.01
TSBN	5.12 $\pm$ 0.50	4.85 $\pm$ 1.26
HMSBN	10.77 $\pm$ 1.15	7.39 $\pm$ 0.47
SS-SRTRBM$^{\diamond}$	8.13 $\pm$ 0.06	5.88 $\pm$ 0.05
G-RTRBM$^{\diamond}$	14.41 $\pm$ 0.38	10.91 $\pm$ 0.27

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.

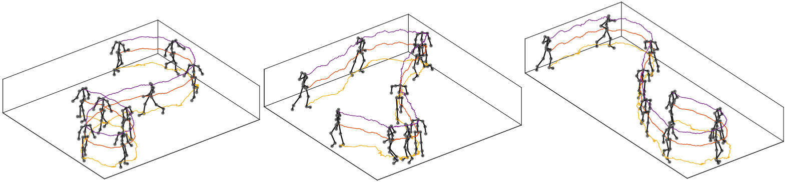


Figure: Generated motion trajectories. (Left) Walking. (Middle) Running-running-walking. (Right) Running-walking.

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].

Model	Piano.	Nott.	Muse.	JSB.
TSBN	-7.98	-3.67	-6.81	-7.48
RNN-NADE◇	**-7.05**	**-2.31**	**-5.60**	**-5.56**
RTRBM◇	-7.36	-2.62	-6.35	-6.35
RNN◇	-8.37	-4.46	-8.13	-8.71

Polyphonic music dataset: Generation

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

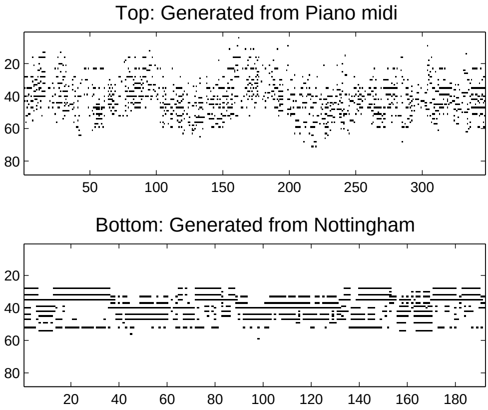


Figure: Generated samples..

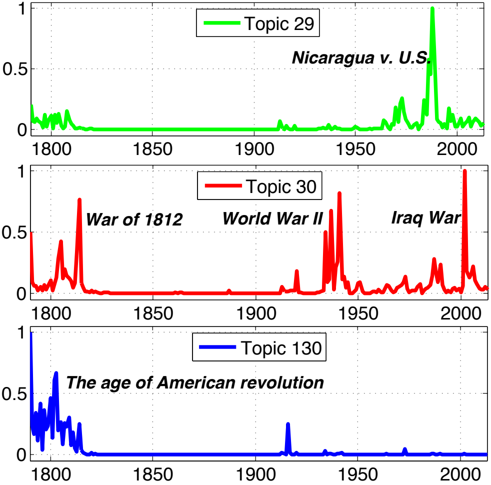
State of the Union dataset: Prediction

- 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.

Model	Dim	MP	PP
HMSBN	25	**0.327** $\pm$ 0.002	0.353 $\pm$ 0.070
DHMSBN-s	25-25	0.299 $\pm$ 0.001	**0.378** $\pm$ 0.006
GP-DPFA $^\diamond$	100	0.223 $\pm$ 0.001	0.189 $\pm$ 0.003
DRFM$^\diamond$	25	0.217 $\pm$ 0.003	0.177 $\pm$ 0.010

State of the Union dataset: Dynamic Topic Modeling

The learned trajectory exhibits different temporal patterns across the topics.



State of the Union dataset: Dynamic Topic Modeling

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

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

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

References I

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.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio.
Learning phrase representations using rnn encoder-decoder for statistical machine translation.
In *arXiv:1406.1078*, 2014.

Z. Gan, C. Chen, R. Henao, D. Carlson, and L. Carin.
Scalable deep poisson factor analysis for topic modeling.
In *ICML*, 2015.

Z. Gan, R. Henao, D. Carlson, and L. Carin.
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.
In *Neural computation*, 1997.

R. Kalman.
Mathematical description of linear dynamical systems.
In *J. the Society for Industrial & Applied Mathematics, Series A: Control*, 1963.

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.
In *Artificial intelligence*, 1992.

L. Rabiner and B. Juang.
An introduction to hidden markov models.
In *ASSP Magazine, IEEE*, 1986.

I. Sutskever and G. Hinton.
Learning multilevel distributed representations for high-dimensional sequences.
In *AISTATS*, 2007.

References III

I. Sutskever, G. Hinton, and G. Taylor.
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.

G. Taylor, G. Hinton, and S. Roweis.
Modeling human motion using binary latent variables.
In *NIPS*, 2006.

Backup I

Contrastive Divergence:

$$\frac{\partial P(\boldsymbol{v})}{\partial \boldsymbol{\theta}} = \mathbb{E}_{p(\boldsymbol{v},\boldsymbol{h})}\left[\frac{\partial}{\partial \theta} E(\boldsymbol{v},\boldsymbol{h})\right] - \mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v})}\left[\frac{\partial}{\partial \boldsymbol{\theta}} E(\boldsymbol{v},\boldsymbol{h})\right] \tag{17}$$

Backup II

The contribution of $\boldsymbol{v}$ to the log-likelihood can be lower-bounded as follows

$$\begin{aligned} \log p_\theta &= \log \sum_{\boldsymbol{h}} p_\theta(\boldsymbol{v}, \boldsymbol{h}) & (18) \\ &\geq \sum_{\boldsymbol{h}} q_\phi(\boldsymbol{h}|\boldsymbol{v}) \log \frac{p_\theta(\boldsymbol{v}, \boldsymbol{h})}{q_\phi(\boldsymbol{h}|\boldsymbol{v})} & (19) \\ &= \mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{v})}[\log p_\theta(\boldsymbol{v}, \boldsymbol{h}) - \log q_\phi(\boldsymbol{h}|\boldsymbol{v})] & (20) \\ &= \mathcal{L}(\boldsymbol{v}, \boldsymbol{\theta}, \phi) & (21) \end{aligned}$$

We can also rewrite the bound as

$$\mathcal{L}(\boldsymbol{v}, \boldsymbol{\theta}, \phi) = \log p_\theta(\boldsymbol{v}) - KL(q_\phi(\boldsymbol{h}|\boldsymbol{v}), p_\theta(\boldsymbol{h}|\boldsymbol{v})) \qquad (22)$$

Backup III

Differentiating the variational lower bound w.r.t. to the recognition model parameters gives

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

$$= \nabla_\phi \sum_{\boldsymbol{h}} q_\phi(\boldsymbol{h}|\boldsymbol{v}) \log p_\theta(\boldsymbol{v}, \boldsymbol{h}) - \nabla_\phi \sum_{\boldsymbol{h}} q_\phi(\boldsymbol{h}|\boldsymbol{v}) \log q_\phi(\boldsymbol{h}|\boldsymbol{v})$$

$$= \sum_{\boldsymbol{h}} \log p_\theta(\boldsymbol{v}, \boldsymbol{h}) \nabla_\phi q_\phi(\boldsymbol{h}|\boldsymbol{v}) - \sum_{\boldsymbol{h}} (\log q_\phi(\boldsymbol{h}|\boldsymbol{v}) + 1) \nabla_\phi q_\phi(\boldsymbol{h}|\boldsymbol{v})$$

$$= \sum_{\boldsymbol{h}} (\log p_\theta(\boldsymbol{v}, \boldsymbol{h}) - \log q_\phi(\boldsymbol{h}|\boldsymbol{v})) \nabla_\phi q_\phi(\boldsymbol{h}|\boldsymbol{v}) \tag{24}$$

where we have used the fact that $\sum_{\boldsymbol{h}} \nabla_\phi q_\phi(\boldsymbol{h}|\boldsymbol{v}) = \nabla_\phi \sum_{\boldsymbol{h}} q_\phi(\boldsymbol{h}|\boldsymbol{v}) = \nabla_\phi 1 = 0$. Using the identity $\nabla_\phi q_\phi(\boldsymbol{h}|\boldsymbol{v}) = q_\phi(\boldsymbol{h}|\boldsymbol{v}) \nabla_\phi \log q_\phi(\boldsymbol{h}|\boldsymbol{v})$, then gives

$$\nabla_\phi \mathcal{L}(\boldsymbol{v}) = \mathbb{E}_q[(\log p_\theta(\boldsymbol{v}, \boldsymbol{h}) - \log q_\phi(\boldsymbol{h}|\boldsymbol{v})) \times \nabla_\phi \log q_\phi(\boldsymbol{h}|\boldsymbol{v})] \tag{25}$$

Backup IV

$$
\begin{aligned}
\mathbb{E}_q[\nabla_\phi \log q_\phi(\boldsymbol{h}|\boldsymbol{v})] &= \mathbb{E}_q\left[\frac{\nabla_\phi q_\phi(\boldsymbol{h}|\boldsymbol{v})}{q_\phi(\boldsymbol{h}|\boldsymbol{v})}\right] && (26)\\
&= \sum_{\boldsymbol{h}} \nabla_\phi q_\phi(\boldsymbol{h}|\boldsymbol{v}) && (27)\\
&= \nabla_\phi \sum_{\boldsymbol{h}} q_\phi(\boldsymbol{h}|\boldsymbol{v}) && (28)\\
&= \nabla_\phi 1 && (29)\\
&= 0 && (30)
\end{aligned}
$$

Backup V

Algorithm 1 Compute gradient estimates.

$\Delta\boldsymbol{\theta} \leftarrow 0, \Delta\boldsymbol{\phi} \leftarrow 0, \Delta\boldsymbol{\lambda} \leftarrow 0, \; \mathcal{L} \leftarrow 0$
for $t \leftarrow 1$ **to** T **do**
  $\boldsymbol{h}_t \sim q_{\boldsymbol{\phi}}(\boldsymbol{h}_t|\boldsymbol{v}_t)$
  $l_t \leftarrow \log p_{\boldsymbol{\theta}}(\boldsymbol{v}_t, \boldsymbol{h}_t) - \log q_{\boldsymbol{\phi}}(\boldsymbol{h}_t|\boldsymbol{v}_t)$
  $\mathcal{L} \leftarrow \mathcal{L} + l_t$
  $l_t \leftarrow l_t - C_{\boldsymbol{\lambda}}(\boldsymbol{v}_t)$
end for
$c_b \leftarrow \text{mean}(l_1, \ldots, l_T)$
$v_b \leftarrow \text{variance}(l_1, \ldots, l_T)$
$c \leftarrow \alpha c + (1-\alpha)c_b$
$v \leftarrow \alpha v + (1-\alpha)v_b$
for $t \leftarrow 1$ **to** T **do**
  $l_t \leftarrow \frac{l_t - c}{\max(1, \sqrt{v})}$
  $\Delta\boldsymbol{\theta} \leftarrow \Delta\boldsymbol{\theta} + \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{v}_t, \boldsymbol{h}_t)$
  $\Delta\boldsymbol{\phi} \leftarrow \Delta\boldsymbol{\phi} + l_t \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{h}_t|\boldsymbol{v}_t)$
  $\Delta\boldsymbol{\lambda} \leftarrow \Delta\boldsymbol{\lambda} + l_t \nabla_{\boldsymbol{\lambda}} C_{\boldsymbol{\lambda}}(\boldsymbol{v}_t)$
end for