August 21, 2017

RT0980

Mathematics   32 pages

Research Report

Boltzmann machines for time-series

Takayuki Osogami

IBM Research - Tokyo
IBM Japan, Ltd.
19-21 Nihonbashi Hakozaki-cho, Chuo-ku
Tokyo 103-8510, Japan

Boltzmann machines for time-series

Takayuki Osogami

IBM Research - Tokyo

osogami@jp.ibm.com

Abstract

We review Boltzmann machines extended for time-series. These models often have recurrent structure, and back propagration through time (BPTT) is used to learn their parameters. The per-step computational complexity of BPTT in online learning, however, grows linearly with respect to the length of preceding time-series (*i.e.*, learning rule is not local in time), which limits the applicability of BPTT in online learning. We then review dynamic Boltzmann machines (DyBMs), whose learning rule is local in time. DyBM's learning rule relates to spike-timing dependent plasticity (STDP), which has been postulated and experimentally confirmed for biological neural networks.

1 Introduction

The Boltzmann machine is a stochastic model for representing probability distributions over binary patterns [28]. In this paper, we review Boltzmann machines that have been studied as stochastic (generative) models of time-series. Such Boltzmann machines define probability distributions over time-series of binary patterns. They can also be modified to deal with time-series of real-valued patterns, similar to Boltzmann machines modified for real-valued patterns (*e.g.*, Gaussian Boltzmann machines; see Section 6.3 from [28]). We will follow the probabilistic representations of [28] for intuitive interpretations in terms of probabilities.

In Section 3, we start with a Conditional Restricted Boltzmann Machine (CRBM) [40], which is a conditional Boltzmann machine (Section 4 from [28]) that gives conditional probability of the next pattern given a fixed number of preceding patterns. A limitation of a CRBM is that it can take into account only the dependency within a fixed horizon, and as we increase the length of this horizon, the complexity of learning grows accordingly.

To overcome this limitation of CRBMs, researchers have proposed Boltzmann machines having recurrent structures, which we review in Section 4. These include spiking Boltzmann machines [12], temporal restricted Boltzmann machines (TRBMs) [37], recurrent temporal restricted Boltzmann machines (RTRBMs) [38], and extensions of those models. A standard approach to learning those models having recurrent structures is back propagation through time (BPTT).

However, BPTT is undesirable when we learn time-series in an online manner, where we update the parameters of a model every time a new pattern arrives. Such online learning is needed when we want to quickly adapt to a changing environment or when we do not have sufficient memory to store the time-series. Unfortunately, the per-step computational complexity of BPTT in online learning grows linearly with respect to the length of preceding time-series. This computational complexity limits the applicability of BPTT to online learning.

In Section 5, we review the dynamic Boltzmann machine (DyBM) [32, 31] and its extensions. The DyBM's per-step computational complexity in online learning is independent of the length of preceding time-series. We discuss how the learning rule of the DyBM relates to spike-timing dependent plasticity (STDP), which has been postulated and experimentally confirmed for biological neural networks.

This survey paper is based on a personal note prepared for the third of the four parts of a tutorial given at the 26th International Joint Conference on Artificial Intelligence (IJCAI-17) held in Melbourne, Australia on August 21, 2017. See a tutorial webpage[1] for information about the tutorial. A survey corresponding to the first part of the tutorial (Boltzmann machines and energy-based models) can be found in [28]. We follow the definitions and notations used in [28].

2 Learning energy-based models for time-series

Consider a possibly multi-dimensional time-series:

$$\mathbf{x} \equiv (\mathbf{x}^{[t]})_{t=0}^{T}, \tag{1}$$

where $\mathbf{x}^{[t]}$ denotes the binary pattern (vector) at time t. We will use $\mathbf{x}^{[s,t]}$ to denote the time-series of the patterns from time s to t.

A goal of learning time-series is to maximize the log-likelihood of a given time-series $\mathbf{x}$ (or a collection of multiple time-series) with respect to the distribution $\mathbb{P}_\theta(\cdot)$ defined by a model under consideration, where we use θ to denote the set of the parameters of the model:

$$f(\theta) \equiv \log \mathbb{P}_\theta(\mathbf{x}) = \sum_{t=0}^{T} \log \mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{x}^{[0,t-1]}), \tag{2}$$

where $\mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{x}^{[0,t-1]})$ denotes the conditional probability that the pattern at time t is $\mathbf{x}^{[t]}$ given that the patterns up to time $t-1$ is $\mathbf{x}^{[0,t-1]}$. Here, $\mathbb{P}_\theta(\mathbf{x}^{[0]} \,|\, \mathbf{x}^{[0,-1]})$ denotes the probability that the pattern at time 0 is $\mathbf{x}^{[0]}$, where $\mathbf{x}^{[0,-1]}$ should be interpreted as an empty history.

We study models where the probability is represented with energy $E_\theta(\cdot)$ as follows:

$$\mathbb{P}_\theta(\mathbf{x}) = \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}, \tilde{\mathbf{h}}), \tag{3}$$

where

$$\mathbb{P}_\theta(\mathbf{x}, \mathbf{h}) = \frac{\exp\Big(-E_\theta(\mathbf{x}, \mathbf{h})\Big)}{\displaystyle\sum_{\tilde{\mathbf{x}}} \sum_{\tilde{\mathbf{h}}} \exp\Big(-E_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{h}})\Big)}, \tag{4}$$

the summation with respect to $\tilde{\mathbf{x}}$ is over all of the possible binary time-series of length T, and the summation with respect to $\tilde{\mathbf{h}}$ is over all of the possible hidden values.

The gradient of $f(\theta)$ can then be represented as follows (see (73)):

$$\nabla f(\theta) = -\mathbb{E}_{\text{target}}\left[\mathbb{E}_\theta\left[\nabla E_\theta(\boldsymbol{X}, \boldsymbol{H}) \,|\, \boldsymbol{X}\right]\right] + \mathbb{E}_\theta\left[\nabla E_\theta(\boldsymbol{X}, \boldsymbol{H})\right], \tag{5}$$

where $\boldsymbol{X}$ represents the random time-series, $\boldsymbol{H}$ represents the random hidden values, $\mathbb{E}_\theta$ denotes the expectation with respect to the model distribution $\mathbb{P}_\theta$, and $\mathbb{E}_{\text{target}}$ denotes the expectation with respect to the target distribution, which in our case is the empirical distribution of time-series. When a single time-series $\mathbf{x}$ is given as the target, (5) is reduced to

$$\nabla f(\theta) = -\mathbb{E}_\theta\left[\nabla E_\theta(\mathbf{x}, \boldsymbol{H})\right] + \mathbb{E}_\theta\left[\nabla E_\theta(\boldsymbol{X}, \boldsymbol{H})\right], \tag{6}$$

In other words, $f(\theta)$ can be maximized by maximizing the sum of

$$f_t(\theta) \equiv \log \mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{x}^{[0,t-1]}), \tag{7}$$

[1] https://researcher.watson.ibm.com/researcher/view_group.php?id=7834

where

$$\mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{x}^{[0,t-1]}) = \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}^{[t]}, \mathbf{h} \,|\, \mathbf{x}^{[0,t-1]}) \tag{8}$$

$$\mathbb{P}_\theta(\mathbf{x}^{[t]}, \mathbf{h} \,|\, \mathbf{x}^{[0,t-1]}) = \frac{\exp\Big(- E_\theta(\mathbf{x}^{[t]}, \mathbf{h} \mid \mathbf{x}^{[0,t-1]})\Big)}{\displaystyle\sum_{\tilde{\mathbf{x}}^{[t]}} \sum_{\tilde{\mathbf{h}}} \exp\Big(- E_\theta(\tilde{\mathbf{x}}^{[t]}, \tilde{\mathbf{h}} \mid \mathbf{x}^{[0,t-1]})\Big)}, \tag{9}$$

and $E_\theta(\mathbf{x}^{[t]}, \mathbf{h} \mid \mathbf{x}^{[0,t-1]})$ is the conditional energy of $(\mathbf{x}^{[t]}, \mathbf{h})$ given $\mathbf{x}^{[0,t-1]}$.

The gradient of $f_t(\theta)$ is given analogously to $\nabla f(\theta)$:

$$\nabla f_t(\theta) = -\mathbb{E}_{\text{target}} \left[\mathbb{E}_\theta \left[\nabla E_\theta(\boldsymbol{X}^{[t]}, \boldsymbol{H} \,|\, \boldsymbol{X}^{[t]}, \mathbf{x}^{[0,t-1]}) \right] \right] + \mathbb{E}_\theta \left[\nabla E_\theta(\boldsymbol{X}^{[t]}, \boldsymbol{H} \mid \mathbf{x}^{[0,t-1]}) \right]. \tag{10}$$

When the target is a single time-series $\mathbf{x}$, we have

$$\nabla f_t(\theta) = -\mathbb{E}_\theta \left[\nabla E_\theta(\mathbf{x}^{[t]}, \boldsymbol{H} \,|\, \mathbf{x}^{[0,t-1]}) \right] + \mathbb{E}_\theta \left[\nabla E_\theta(\boldsymbol{X}^{[t]}, \boldsymbol{H} \mid \mathbf{x}^{[0,t-1]}) \right]. \tag{11}$$

3 Non-recurrent Boltzmann machines for time-series

By (2), any model that can represent the conditional probability $\mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{x}^{[0,t-1]})$ can be used for time-series. In this section, we start with a Boltzmann machine that can be used to model a D-th order Markov model for an arbitrarily determined D. In D-th order Markov models, the conditional probability can be represented as

$$\mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{x}^{[0,t-1]}) = \mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{x}^{[t-D,t-1]}). \tag{12}$$

3.1 Conditional restricted Boltzmann machines

Figure 1a shows a particularly structured Boltzmann machine called Conditional Restricted Boltzmann Machine (CRBM) [40]. A CRBM represents the conditional probability on the right-hand side of (12). In the figure, we set $D = 2$.

The CRBM consists of $D + 1$ layers of visible units and a layer of hidden units. The units within each layer have no connections, but units between different layers may be connected to each other. Each visible layer corresponds to a pattern at a time $s \in [t - D, t]$.

The CRBM is a conditional Boltzmann machine shown in Figure 2c from [28] but with a particular structure to represent time-series. The visible layers corresponding to $\mathbf{x}^{[t-D,t-1]}$ are the input, and the visible layer corresponding to $\mathbf{x}^{[t]}$ is the output. The parameters θ of the CRBM are independent of t.

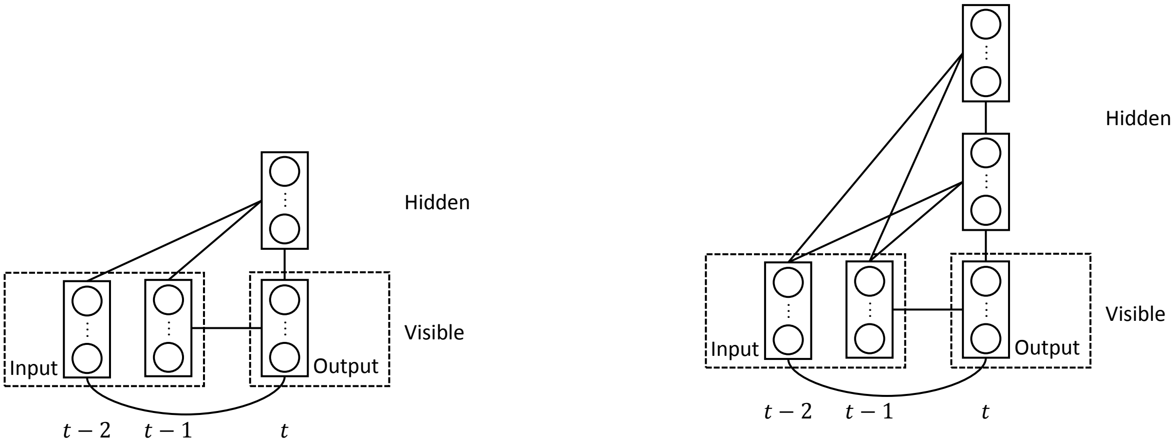
More formally, the energy of a CRBM is given by

$$E_\theta\big(\mathbf{x}^{[t]}, \mathbf{h} \mid \mathbf{x}^{[t-D,t-1]}\big) = -(\mathbf{b}^{\mathrm{V}})^\top \mathbf{x}^{[t]} - (\mathbf{b}^{\mathrm{H}})^\top \mathbf{h} - \mathbf{h}\, \mathbf{W}^{\mathrm{HV}}\, \mathbf{x}^{[t]} - \sum_{d=1}^{D} (\mathbf{x}^{[t-d]})^\top\, \mathbf{W}^{[d]}\, \mathbf{x}^{[t]}, \tag{13}$$

where $\mathbf{x}^{[t]}$ is output, $\mathbf{x}^{[t-D,t-1]}$ is input, and $\mathbf{h}$ is hidden.

We can then represent the conditional probability as follows (see (14) from [28]):

$$\mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{x}^{[t-D,t-1]}) = \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}^{[t]}, \tilde{\mathbf{h}} \,|\, \mathbf{x}^{[t-D,t-1]}), \tag{14}$$



(a) Single hidden layer

(b) Multiple hidden layers

Figure 1: Conditional restricted Boltzmann machines.

where

$$\mathbb{P}_\theta(\mathbf{x}^{[t]}, \tilde{\mathbf{h}} \,|\, \mathbf{x}^{[t-D,t-1]}) = \frac{\exp\Big(-E_\theta\big(\mathbf{x}^{[t]}, \tilde{\mathbf{h}} \mid \mathbf{x}^{[t-D,t-1]}\big)\Big)}{\displaystyle\sum_{\tilde{\mathbf{x}}^{[t]}} \Big(-E_\theta\big(\tilde{\mathbf{x}}^{[t]}, \tilde{\mathbf{h}} \mid \mathbf{x}^{[t-D,t-1]}\big)\Big)}, \tag{15}$$

and the summation with respect to $\tilde{\mathbf{h}}$ is over all of the possible binary hidden patterns, and the summation with respect to $\tilde{\mathbf{x}}^{[t]}$ is defined analogously.

One can then learn the parameters $\theta = (\mathbf{b}^{\mathrm{V}}, \mathbf{h}^{\mathrm{h}}, \mathbf{W}^{\mathrm{HV}}, \mathbf{W}^{[1]}, \ldots, \mathbf{W}^{D})$ of the model by following a gradient-based method in Section 4 from [28].

3.2 Extensions of conditional restricted Boltzmann machines

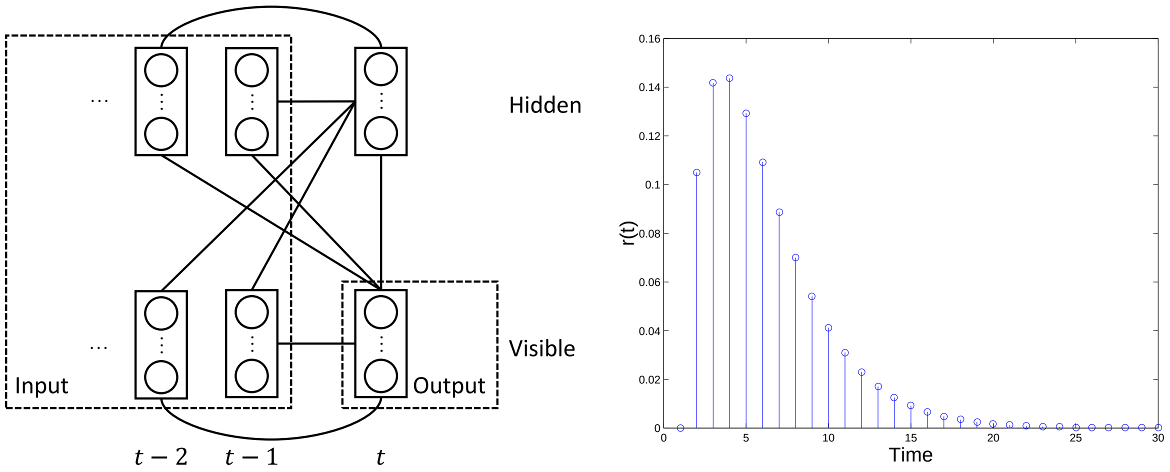
The CRBM has been extended in various ways. Taylor et al. study a CRBM with multiple layers of hidden units [40] (see Figure 1b). Memisevic and Hinton study a CRBM extended with three-way interactions (*i.e.*, a higher order Boltzmann machine), which they refer to as a gated CRBM [24]. Specifically, the energy of the gated CRBM involves

$$-\sum_{i,j,k} w_{i,j,k}\, x_i\, y_j\, h_k, \tag{16}$$

where $\mathbf{x}$ denotes input values, $\mathbf{y}$ denotes output values, and $\mathbf{h}$ denotes hidden values. A drawback of the gated CRBM is its increased number of parameters due to the three-way interactions. Taylor and Hinton study a factored CRBM, where the three-way interaction is represented with a reduced number of parameters as follows [39]:

$$-\sum_{f} \sum_{i,j,k} w^{\mathbf{v}}_{i,f}\, w^{\mathbf{y}}_{j,f}\, w^{\mathbf{h}}_{k,f}\, x_i\, y_j\, h_k, \tag{17}$$

where the summation with respect to f is over a set of factors under consideration.



(a) Structure of a spiking Boltzmann machine

(b) $r(\cdot)$ used in [12]

Figure 2: A spiking Boltzmann machine studied in [12]. In (b), we use the figure in the version available at http://www.cs.toronto.edu/∼fritz/absps/nips00-ab.pdf.

4 Boltzmann machines for time-series with recurrent structures

4.1 Spiking Boltzmann machines

A spiking Boltzmann machine studied in [12] can be shown to be essentially equivalent to the Boltzmann machine illustrated in Figure 2. This Boltzmann machine consists of input units, output units, and hidden units. The input units represent historical values of visible units and hidden units. Although hidden units are random and cannot be simply given as input, Hinton and Brown make the approximation of using sampled values $\boldsymbol{H}^{(-\infty,t-1]}(\omega)$ as the input hidden units [12].

Specifically, given the visible values $\mathbf{x}^{[<t]} \equiv \mathbf{x}^{(-\infty,t-1]}$ and sampled hidden values $\boldsymbol{H}^{[<t]}(\omega)$ up to time $t-1$, the energy with the visible values $\mathbf{x}^{[t]}$ and hidden values $\mathbf{h}^{[t]}$ at time t can be represented as follows:

$$\begin{aligned} E_\theta(\mathbf{x}^{[t]}, \mathbf{h}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)) = &-(\mathbf{b}^{\mathrm{V}})^\top \mathbf{x}^{[t]} - (\mathbf{b}^{\mathrm{H}})^\top \mathbf{h}^{[t]} - (\mathbf{h}^{[t]})^\top r(\tau)\, \mathbf{W}^{\mathrm{HV}}\, \mathbf{x}^{[t]} \\ &- \sum_{\tau=1}^{\infty} (\boldsymbol{H}^{[t-\tau]}(\omega))^\top r(\tau)\, \mathbf{W}^{\mathrm{HH}}\, \mathbf{h}^{[t]} - \sum_{\tau=1}^{\infty} (\mathbf{x}^{[t-\tau]})^\top r(\tau)\, \mathbf{W}^{\mathrm{VH}}\, \mathbf{h}^{[t]} \\ &- \sum_{\tau=1}^{\infty} (\boldsymbol{H}^{[t-\tau]}(\omega))^\top r(\tau)\, \mathbf{W}^{\mathrm{HV}}\, \mathbf{x}^{[t]} - \sum_{\tau=1}^{\infty} (\mathbf{x}^{[t-\tau]})^\top r(\tau)\, \mathbf{W}^{\mathrm{VV}}\, \mathbf{x}^{[t]}, \end{aligned} \tag{18}$$

where $r(\cdot)$ is an arbitrarily chosen function and is not the target of learning. Namely, the Boltzmann machine has an infinite number of units but can be characterized by a finite number of parameters $\theta \equiv (\mathbf{b}^{\mathrm{V}}, \mathbf{b}^{\mathrm{H}}, \mathbf{W}^{\mathrm{VV}}, \mathbf{W}^{\mathrm{VH}}, \mathbf{W}^{\mathrm{HV}}, \mathbf{W}^{\mathrm{HH}})$. Figure 2(b) shows the specific $r(\cdot)$ used in [12][2].

Notice that the Boltzmann machine in Figure 2 can be seen as a restricted Boltzmann machine (RBM) whose bias and weight can depend on $\mathbf{x}^{[<t]}$ and $\boldsymbol{H}^{[<t]}(\omega)$, because (18) can be represented as

$$E_\theta(\mathbf{x}^{[t]}, \mathbf{h}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{H}^{[<t]}(\omega)) = -\mathbf{b}^{\mathrm{H}}(t,\omega)\, \mathbf{h}^{[t]} - \mathbf{b}^{\mathrm{V}}(t,\omega)\, \mathbf{x}^{[t]} - \mathbf{h}^{[t]}\, \mathbf{W}\, \mathbf{x}^{[t]}, \tag{19}$$

[2] Although it is not clear from the descriptions in [12], the labels in the horizontal axis should probably be shifted by one, so that $r(0) = 0$, $r(1) \approx 0.1$, and so on.

where $\mathbf{b}^{\mathrm{H}}(t,\omega)$ is the time-varying bias for hidden units, $\mathbf{b}^{\mathrm{V}}(t,\omega)$ is the time-varying bias for visible units, and $\mathbf{W}$ is the weight between visible units and hidden units:

$$\mathbf{b}^{\mathrm{H}}(t,\omega) \equiv \mathbf{b}^{\mathrm{H}} + \sum_{\tau=1}^{\infty} (\boldsymbol{H}^{[t-\tau]}(\omega))^{\top}\, r(\tau)\, \mathbf{W}^{\mathrm{HH}} + \sum_{\tau=1}^{\infty} (\mathbf{x}^{[t-\tau]})^{\top}\, r(\tau)\, \mathbf{W}^{\mathrm{VH}} \tag{20}$$

$$\mathbf{b}^{\mathrm{V}}(t,\omega) \equiv \mathbf{b}^{\mathrm{V}} + \sum_{\tau=1}^{\infty} (\boldsymbol{H}^{[t-\tau]}(\omega))^{\top}\, r(\tau)\, \mathbf{W}^{\mathrm{HV}} + \sum_{\tau=1}^{\infty} (\mathbf{x}^{[t-\tau]})^{\top}\, r(\tau)\, \mathbf{W}^{\mathrm{VV}} \tag{21}$$

$$\mathbf{W} \equiv r(0)\, \mathbf{W}^{\mathrm{HV}}. \tag{22}$$

We can then represent the conditional probability as follows:

$$\mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{x}^{[<t]}, \boldsymbol{H}^{(-\infty,t-1}(\omega)) = \sum_{\tilde{\mathbf{h}}^{[t]}} \mathbb{P}_\theta\big(\mathbf{x}^{[t]}, \tilde{\mathbf{h}}^{[t]} \,|\, \mathbf{x}^{[<t]}, \boldsymbol{H}^{(-\infty,t-1]}(\omega)\big) \tag{23}$$

where

$$\mathbb{P}_\theta(\mathbf{x}^{[t]}, \tilde{\mathbf{h}}^{[t]} \,|\, \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)) = \frac{\exp\Big(- E_\theta\big(\mathbf{x}^{[t]}, \tilde{\mathbf{h}}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{(-\infty,t-1]}(\omega)\big)\Big)}{\displaystyle\sum_{\tilde{\mathbf{x}}^{[t]}} \exp\Big(- E_\theta\big(\tilde{\mathbf{x}}^{[t]}, \tilde{\mathbf{h}}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)\big)\Big)}. \tag{24}$$

We now discuss the choice of $r(0) = 0$, which appears to be the case in Figure 2(b). In this case, the energy is reduced to

$$E_\theta(\mathbf{x}^{[t]}, \mathbf{h}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)) = -(\mathbf{b}^{\mathrm{H}}(t,\omega))^{\top}\, \mathbf{h}^{[t]} - (\mathbf{b}^{\mathrm{V}}(t,\omega))^{\top}\, \mathbf{x}^{[t]}. \tag{25}$$

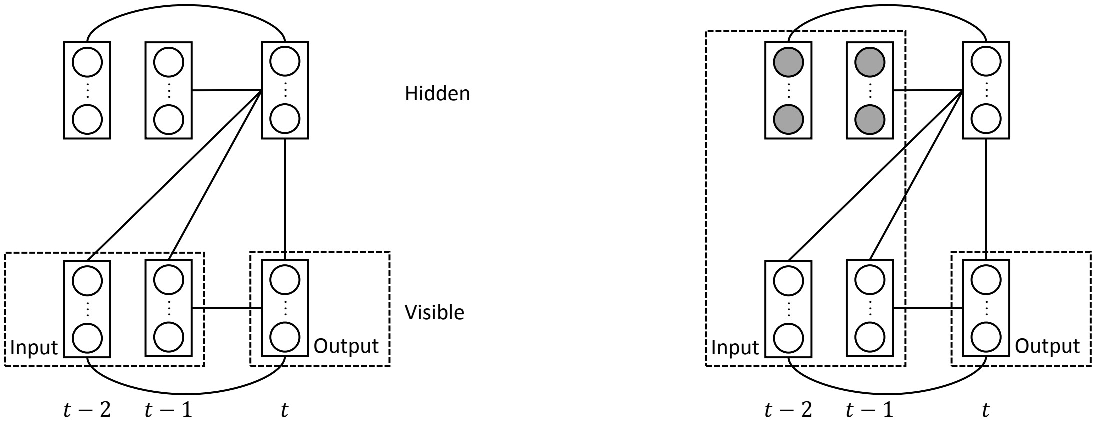
Because there are no connections between visible units and hidden units at time t, the hidden values at t do not affect the distribution of the visible values at t. The only role of the hidden units is that the sampled hidden values are used to update the time-varying bias, $\mathbf{b}^{\mathrm{V}}(s,\omega)$ and $\mathbf{b}^{\mathrm{H}}(s,\omega)$ for $s > t$. A problem is that there is no mechanism that allows us to learn appropriate values of $\mathbf{W}^{\mathrm{VH}}$ and $\mathbf{W}^{\mathrm{HH}}$ until we observe succeeding visible values. Namely, the hidden values $\mathbf{h}^{[t]}$ are sampled with the dependency on $\mathbf{W}^{\mathrm{VH}}$ and $\mathbf{W}^{\mathrm{HH}}$, but whether the sampled hidden values are good or not can only be known when those hidden values are used as input. This helps us to learn appropriate values of $\mathbf{W}^{\mathrm{HV}}$, but not $\mathbf{W}^{\mathrm{VH}}$ or $\mathbf{W}^{\mathrm{HH}}$. See [30] for further discussion.

4.2 Temporal restricted Boltzmann machines

Sutskever and Hinton study a model related to a CRBM, which they refer to as a temporal restricted Boltzmann machine (TRBM) [37]. While a CRBM defines the conditional distribution of the (visible and hidden) values at time t given only the visible values from time $t-D$ to $t-1$, a TRBM defines the corresponding conditional probability given both the visible values and the hidden values from time $t-D$ to $t-1$. See Figure 3a. Similar to the CRBM, the parameters θ of the TRBM do not depend on time t.

Unlike the CRBM, the TRBM is not a conditional RBM. This is because the TRBM with a single parameter is used for every t, and the distribution of hidden values is shared among those TRBM at varying t. In particular, hidden values of the TRBM can depend on the future visible values.

Because this dependency makes learning and inference hard, it is ignored in [37]. Namely, the values at each time t is conditionally independent of the values after time t given the values at and before time t. In particular, the distribution of the hidden values at time t is completely determined by the visible values up to time t. The distribution of the hidden values before time t can thus be considered as input when we use the TRBM to define the conditional distribution of the values at time t (see Figure 3b). For each sampled values of hidden units, TRBM in 3b is a CRBM.



(a) TRBM

(b) TRBM with approximations in [37]

Figure 3: Temporal restricted Boltzmann machines. In (b), the gray circles indicate that expected values are used for the input hidden units.

Furthermore, in [37], the expected values (see Section 5.4 from [28]) are used for the hidden values. Then the input hidden units in Figure 3b takes real values in $[0,1]$ that are completely determined by the visible values before time t.

More formally, with the approximations in [37], the TRBM with parameter θ defines the probability distribution over the time-series of visible and hidden values as follows:

$$\mathbb{P}_\theta(\mathbf{x}) = \prod_{t=0}^{T} \sum_{\tilde{\mathbf{h}}^{[t]}} \mathbb{P}_\theta(\mathbf{x}^{[t]}, \tilde{\mathbf{h}}^{[t]} \,|\, \mathbf{x}^{[t-D,t-1]}, \mathbf{r}^{[t-D,t-1]}), \tag{26}$$

where

$$\mathbb{P}_\theta(\mathbf{x}^{[t]}, \tilde{\mathbf{h}}^{[t]} \,|\, \mathbf{x}^{[t-D,t-1]}, \mathbf{r}^{[t-D,t-1]}) = \frac{\exp\Big(- E_\theta(\mathbf{x}^{[t]}, \tilde{\mathbf{h}}^{[t]} \,|\, \mathbf{x}^{[t-D,t-1]}, \mathbf{r}^{[t-D,t-1]})\Big)}{\displaystyle\sum_{\tilde{\mathbf{x}}^{[t]}} \exp\Big(- E_\theta(\tilde{\mathbf{x}}^{[t]}, \tilde{\mathbf{h}}^{[t]} \,|\, \mathbf{x}^{[t-D,t-1]}, \mathbf{r}^{[t-D,t-1]})\Big)} \tag{27}$$

is the conditional distribution defined by the Boltzmann machine shown in Figure 3b, where $\mathbf{r}^{[t-D,t-1]}$ are expected hidden values. Specifically, (27) is used to compute

$$\mathbf{r}^{[t]} = \mathbb{E}_\theta[\boldsymbol{H}^{[t]} \mid \mathbf{x}^{[0,t]}], \tag{28}$$

which is subsequently used with (27) for $t \leftarrow t+1$, where the expectation in (28) is with respect to

$$\mathbb{P}_\theta(\mathbf{h}^{[t]} \mid \mathbf{x}^{[0,t-1]}) = \frac{\mathbb{P}_\theta(\mathbf{x}^{[t]}, \mathbf{h}^{[t]} \mid \mathbf{x}^{[t-D,t-1]}, \mathbf{r}^{[t-D,t-1]})}{\displaystyle\sum_{\tilde{\mathbf{x}}^{[t]}} \mathbb{P}_\theta(\tilde{\mathbf{x}}^{[t]}, \mathbf{h}^{[t]} \mid \mathbf{x}^{[t-D,t-1]}, \mathbf{r}^{[t-D,t-1]})}. \tag{29}$$

Notice that $\mathbf{r}^{[t]}$ can be computed from $\mathbf{x}^{[0,t]}$ in a deterministic manner with dependency on θ. However, this dependency on θ is ignored in learning TRBMs.

4.3 Recurrent temporal restricted Boltzmann machines

To overcome the intractability of the TRBM without approximations, Sutskever et al. study a refined model of TRBM, which they refer to as a recurrent temporal restricted Boltzmann machine (RTRBM)

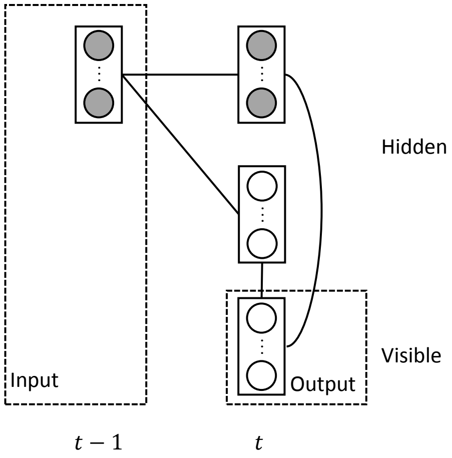


Figure 4: A recurrent temporal restricted Boltzmann machine.

[38]. The RTRBM simplifies the TRBM by removing connections between visible layers and connections between hidden layers that are separated by more than one lag. This means that the (visible and hidden) values at time t are conditionally independent of the the visible values before time t and the hidden values before time $t-1$ given the hidden values at time $t-1$. Similar to the approximation made for the TRBM in Figure 3b, the RTRBM uses the expected values for the hidden values at time $t-1$ but defines the conditional distribution of the (visible and hidden) values at time t over their binary values. See Figure 4.

More formally, let $\mathbf{r}^{[t-1]}$ denote the expected values of the hidden units at time $t-1$:

$$\mathbf{r}^{[t-1]} \equiv \mathbb{E}_\theta\big[\mathbf{H}^{[t-1]} \,|\, \mathbf{x}^{[0,t-1]}\big], \tag{30}$$

where $\mathbf{H}^{[t-1]}$ is the random vector representing the hidden values at time t, and $\mathbb{E}_\theta[\cdot \,|\, \mathbf{x}^{[0,t-1]}]$ represents the conditional expectation given the visible values up to time $t-1$. The probability distribution of the values at time t is then given by

$$\mathbb{P}_\theta(\mathbf{x}^{[t]}, \mathbf{h}^{[t]} \,|\, \mathbf{r}^{[t-1]}) = \frac{\exp\Big(-E_\theta(\mathbf{x}^{[t]}, \mathbf{h}^{[t]} \mid \mathbf{r}^{[t-1]})\Big)}{\sum_{\tilde{\mathbf{x}},\tilde{\mathbf{h}}} \exp\Big(-E_\theta(\tilde{\mathbf{x}}^{[t]}, \tilde{\mathbf{h}}^{[t]} \mid \mathbf{r}^{[t-1]})\Big)}, \tag{31}$$

where the conditional energy is given by

$$E_\theta(\mathbf{x}^{[t]}, \mathbf{h}^{[t]} \mid \mathbf{r}^{[t-1]}) \equiv -(\mathbf{b}^{\mathrm{V}})^\top \mathbf{x}^{[t]} - (\mathbf{b}^{\mathrm{H}})^\top \mathbf{h}^{[t]} - (\mathbf{r}^{[t-1]})^\top \mathbf{U}\,\mathbf{h}^{[t]} - (\mathbf{x}^{[t]})^\top \mathbf{W}\,\mathbf{h}^{[t]}. \tag{32}$$

4.3.1 Inference

The marginal conditional distribution of visible values at time $t-1$ can then be represented as follows:

$$\mathbb{P}_\theta(\mathbf{x}^{[t]} \,|\, \mathbf{r}^{[t-1]}) = \frac{\exp\Big(-F_\theta(\mathbf{x}^{[t]} \mid \mathbf{r}^{[t-1]})\Big)}{\sum_{\tilde{\mathbf{x}}^{[t]}} \exp\Big(-F_\theta(\tilde{\mathbf{x}}^{[t]} \mid \mathbf{r}^{[t-1]})\Big)}, \tag{33}$$

where the conditional free-energy is given by

$$F_\theta(\mathbf{x}^{[t]} \mid \mathbf{r}^{[t-1]}) \equiv -\log \sum_{\tilde{\mathbf{h}}^{[\mathbf{t}]}} \exp\Big(-E_\theta(\mathbf{x}^{[t]}, \tilde{\mathbf{h}}^{[t]} \,|\, \mathbf{r}^{[t-1]})\Big). \tag{34}$$

Once the visible values at time t is given, the conditional probability distribution of the hidden values at time t can be represented as follows:

$$\mathbb{P}_\theta(\mathbf{h}^{[t]} \,|\, \mathbf{r}^{[t-1]}, \mathbf{x}^{[t]}) = \frac{\exp\Big(- E_\theta(\mathbf{h}^{[t]} \mid \mathbf{r}^{[t-1]}, \mathbf{x}^{[t]})\Big)}{\sum_{\tilde{\mathbf{h}}^{[t]}} \exp\Big(- E_\theta(\tilde{\mathbf{h}}^{[t]} \mid \mathbf{r}^{[t-1]}, \mathbf{x}^{[t]})\Big)}, \tag{35}$$

where the $(\mathbf{b}^{\mathrm{V}})^\top \mathbf{x}^{[t]}$ term is canceled out between the numerator and the denominator, and the conditional energy is given by

$$E_\theta(\mathbf{h}^{[t]} \mid \mathbf{r}^{[t-1]}, \mathbf{x}^{[t]}) \equiv -\Big(\mathbf{b}^{\mathrm{H}} + \mathbf{U}^\top \mathbf{r}^{[t-1]} + \mathbf{W}^\top \mathbf{x}^{[t]}\Big)^\top \mathbf{h}^{[t]}. \tag{36}$$

By Corollary 1 from [28], the hidden values at time t are conditionally independent of each other given $\mathbf{r}^{[t-1]}$ and $\mathbf{x}^{[t]}$:

$$\mathbb{P}_\theta(\mathbf{h}^{[t]} \,|\, \mathbf{r}^{[t-1]}, \mathbf{x}^{[t]}) = \prod_i \mathbb{P}_\theta(h_i^{[t]} \,|\, \mathbf{r}^{[t-1]}, \mathbf{x}^{[t]}), \tag{37}$$

where

$$\mathbb{P}_\theta(h_i^{[t]} \,|\, \mathbf{r}^{[t-1]}, \mathbf{x}^{[t]}) = \frac{\exp\big(- b_i^{[t]}\, h_i^{[t]}\big)}{1 + \exp\big(- b_i^{[t]}\big)}, \tag{38}$$

where $b_i^{[t]}$ is the i-th element of

$$\mathbf{b}^{[t]} \equiv \mathbf{b}^{\mathrm{H}} + \mathbf{U}^\top \mathbf{r}^{[t-1]} + \mathbf{W}^\top \mathbf{x}^{[t]} \tag{39}$$

for $t \geq 1$, and

$$\mathbf{b}^{[0]} \equiv \mathbf{b}^{\mathrm{init}} + \mathbf{W}^\top \mathbf{x}^{[0]}. \tag{40}$$

where we now follow [38] and allow the hidden units at time 0 to have own bias $\mathbf{b}^{\mathrm{init}}$ that can differ from $\mathbf{b}^{\mathrm{H}}$. The expected values are thus given by

$$\mathbf{r}^{[t]} = \frac{1}{1 + \exp\big(\mathbf{b}^{[t]}\big)}, \tag{41}$$

where the operations are defined elementwise.

4.3.2 Learning

The parameters of an RTRBM can be trained through back propagation through time, analogous to recurrent neural networks, but with contrastive divergence. To understand how we can train RTRBMs, Figure 5 shows an RTRBM unfolded through time. Recall that the expected values of hidden units are deterministically updated from $\mathbf{r}^{[t-1]}$ to $\mathbf{r}^{[t]}$ according to (39)-(41). Hence, $\mathbf{r}^{[t]}$ can be understood as hidden values of a recurrent neural network (RNN) [34]. An RTRBM can then be seen as an RNN but gives an RBM as an output instead of real values, which would be given as an output from the standard RNN. We will derive the learning rule of the RTRBM, closely following [25] but using our notations.

When an RTRBM is unfolded through time, its energy can be represented as follows:

$$E_\theta(\mathbf{x}, \mathbf{h}) = -\sum_{t=0}^{T} (\mathbf{b}^{\mathrm{V}})^\top \mathbf{x}^{[t]} - (\mathbf{b}^{\mathrm{init}})^\top \mathbf{h}^{[0]} - \sum_{t=1}^{T} (\mathbf{b}^{\mathrm{H}})^\top \mathbf{h}^{[t]} - \sum_{t=0}^{T} (\mathbf{x}^{[t]})^\top \mathbf{W}\, \mathbf{h}^{[t]} - \sum_{t=1}^{T} (\mathbf{r}^{[t-1]})^\top \mathbf{U}\, \mathbf{h}^{[t]}. \tag{42}$$

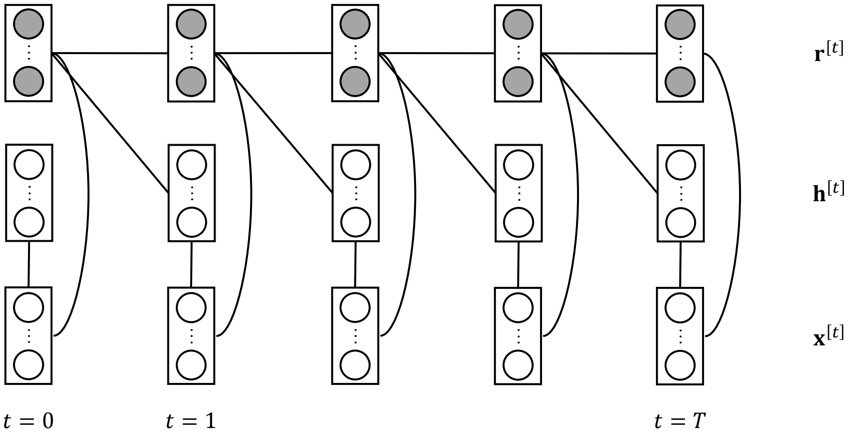


Figure 5: A recurrent temporal restricted Boltzmann machine unfolded through time, where $T = 4$.

By (5), we can maximize the log-likelihood of a given time-series $\mathbf{x}$ with a gradient-based approach. What we need in (5) is the gradient of the energy with respect to the parameter. A caveat is that the energy in (42) depends on $\mathbf{r}^{[\cdot]}$, which in turn depends on θ in a recursive manner. Also, expectation with respect to $\mathbb{P}_theta$ in (5) needs to be computed with approximation such as contrastive divergence (see Section 5.2).

We first study the last term of (42). Let

$$Q_s \equiv \sum_{t=s}^{T} (\mathbf{r}^{[t-1]})^\top \mathbf{U}\, \mathbf{h}^{[t]} \tag{43}$$

$$= (\mathbf{r}^{[s-1]})^\top \mathbf{U}\, \mathbf{h}^{[s]} + Q_{s+1}, \tag{44}$$

for $s \in [1, T]$, where $Q_{T+1} \equiv 0$, so that $Q \equiv Q_1$ is the last term of (42). Taking the partial derivative with respect to $r_i^{[s-1]}$, we obtain

$$\frac{\partial Q_s}{\partial r_i^{[s-1]}} = \frac{\partial}{\partial r_i^{[s-1]}} (\mathbf{r}^{[s-1]})^\top \mathbf{U}\, \mathbf{h}^{[s]} + \sum_j \frac{\partial r_j^{[s]}}{\partial r_i^{[s-1]}} \frac{\partial Q_{s+1}}{\partial r_j^{[s]}} \tag{45}$$

$$= \mathbf{U}_{i,:}\, \mathbf{h}^{[s]} + \sum_j r_j^{[s]} (1 - r_j^{[s]})\, u_{i,j}\, \frac{\partial Q_{s+1}}{\partial r_j^{[s]}}, \tag{46}$$

where $\mathbf{U}_{i,:}$ denotes the i-th row of $\mathbf{U}$, $u_{i,j}$ denotes the (j,i)-th element of $\mathbf{U}$, and the last equality follows from (39)-(41). In vector-matrix notations, we can write

$$\nabla_{\mathbf{r}^{[s-1]}} Q_s = \mathbf{U} \left(\mathbf{h}^{[s]} + \mathbf{r}^{[s]} \cdot (1 - \mathbf{r}^{[s]}) \cdot \nabla_{\mathbf{r}^{[s]}} Q_{s+1} \right), \tag{47}$$

where $\cdot$ denotes elementwise multiplication. Because Q_s is not a function of $\mathbf{r}^{[0]}, \ldots, \mathbf{r}^{[s-2]}$, we have

$$\nabla_{\mathbf{r}^{[s-1]}} Q = \nabla_{\mathbf{r}^{[s-1]}} Q_s. \tag{48}$$

Therefore, the partial derivative of Q with respect to $\mathbf{r}^{[s-1]}$ is given recursively as follows:

$$\nabla_{\mathbf{r}^{[s-1]}} Q = \mathbf{U} \left(\mathbf{h}^{[s]} + \mathbf{r}^{[s]} \cdot (1 - \mathbf{r}^{[s]}) \cdot \nabla_{\mathbf{r}^{[s]}} Q \right) \tag{49}$$

for $s = 1, \ldots, T$ and

$$\nabla_{\mathbf{r}^{[T]}} Q = \mathbf{0}. \tag{50}$$

We now take the derivative of Q with respect to the parameters in θ, starting with $\mathbf{U}$:

$$\frac{\mathrm{d}Q}{\mathrm{d}u_{i,j}} = \sum_{t=0}^{T} \sum_k \frac{\partial r_k^{[t]}}{\partial u_{i,j}} \frac{\partial Q}{\partial r_k^{[t]}} + \frac{\partial Q}{\partial u_{i,j}} \tag{51}$$

$$= \sum_{t=1}^{T} r_j^{[t]} \, (1 - r_j^{[t]}) \, r_i^{[t-1]} \, \frac{\partial Q}{\partial r_j^{[t]}} + \sum_{t=1}^{T} r_i^{[t-1]} \, h_j^{[t]}, \tag{52}$$

where the last equality follows from (39)-(41). In vector-matrix notations, we can write

$$\nabla_{\mathbf{U}} Q = \sum_{t=1}^{T} \mathbf{r}^{[t-1]} \left(\mathbf{r}^{[t]} \cdot (1 - \mathbf{r}^{[t]}) \cdot \nabla_{\mathbf{r}^{[t]}} Q + \mathbf{h}^{[t]} \right)^{\top}, \tag{53}$$

where $\nabla_{\mathbf{r}^{[t]}} Q$ is given by (49)-(50).

The gradient of Q with respect to other parameters can be derived as follows:

$$\nabla_{\mathbf{W}} Q = \sum_{t=0}^{T} \mathbf{x}^{[t]} \left(\mathbf{r}^{[t]} \cdot (1 - \mathbf{r}^{[t]}) \cdot \nabla_{\mathbf{r}^{[t]}} Q \right)^{\top} \tag{54}$$

$$\nabla_{\mathbf{b}^{\mathrm{H}}} Q = \sum_{t=1}^{T} \mathbf{r}^{[t]} \cdot (1 - \mathbf{r}^{[t]}) \cdot \nabla_{\mathbf{r}^{[t]}} Q \tag{55}$$

$$\nabla_{\mathbf{b}^{\mathrm{init}}} Q = \mathbf{r}^{[0]} \cdot (1 - \mathbf{r}^{[0]}) \cdot \nabla_{\mathbf{r}^{[0]}} Q \tag{56}$$

$$\nabla_{\mathbf{b}^{\mathrm{V}}} Q = \mathbf{0} \tag{57}$$

The gradients of Q can be used to show the following gradients of the energy:

$$\nabla_{\mathbf{U}} E_\theta(\mathbf{x}, \mathbf{h}) = -\sum_{t=1}^{T} \mathbf{r}^{[t-1]} \left(\mathbf{r}^{[t]} \cdot (1 - \mathbf{r}^{[t]}) \cdot \nabla_{\mathbf{r}^{[t]}} Q + \mathbf{h}^{[t]} \right)^{\top} \tag{58}$$

$$\nabla_{\mathbf{W}} E_\theta(\mathbf{x}, \mathbf{h}) = -\sum_{t=0}^{T} \mathbf{x}^{[t]} \, (\mathbf{h}^{[t]})^{\top} - \sum_{t=0}^{T} \mathbf{x}^{[t]} \left(\mathbf{r}^{[t]} \cdot (1 - \mathbf{r}^{[t]}) \cdot \nabla_{\mathbf{r}^{[t]}} Q \right)^{\top} \tag{59}$$

$$\nabla_{\mathbf{b}^{\mathrm{H}}} E_\theta(\mathbf{x}, \mathbf{h}) = -\sum_{t=1}^{T} \mathbf{h}^{[t]} - \sum_{t=1}^{T} \mathbf{r}^{[t]} \cdot (1 - \mathbf{r}^{[t]}) \cdot \nabla_{\mathbf{r}^{[t]}} Q \tag{60}$$

$$\nabla_{\mathbf{b}^{\mathrm{init}}} E_\theta(\mathbf{x}, \mathbf{h}) = -\mathbf{h}^{[0]} - \mathbf{r}^{[0]} \cdot (1 - \mathbf{r}^{[0]}) \cdot \nabla_{\mathbf{r}^{[0]}} Q \tag{61}$$

$$\nabla_{\mathbf{b}^{\mathrm{V}}} E_\theta(\mathbf{x}, \mathbf{h}) = -\sum_{t=0}^{T} \mathbf{x}^{[t]}, \tag{62}$$

where $\nabla_{\mathbf{r}^{[t]}} Q$ is given by (49)-(50). The gradient of the log-likelihood of the given time-series $\mathbf{x}$ now follows from (72) in [28].

4.3.3 Extensions

The RTRBM has been extended in various ways. Mittleman et al. study a structured RTRBM, where units are partitioned into blocks, and only the connections between particular blocks are allowed [25]. Lyu et al. replaces the RNN of RTRBM with the one with Long Short-Term Memory (LSTM) [22]. Schrauwen and Buesing replaces the RNN of RTRBM with an echo state network [36].

An RNN-RBM slightly generalizes RTRBM by relaxing the constraint of the RTRBM that $\mathbf{r}^{[t]}$ must be the expected value of $\mathbf{h}^{[t]}$ [7]. Namely, an RNN-RBM is an RNN but gives an RBM as an output, where the RNN and RBM do not share parameters, while an RTRBM shares parameters between an RNN and an RBM.

5 Dynamic Boltzmann machines

BPTT is not desirable for online learning, where we update θ every time a new pattern $\mathbf{x}^{[t]}$ is observed. The per-step computational complexity of BPTT in online learning grows linearly with the length of the preceding time-series. Such online learning, however, is needed for example when we cannot store all observed patterns in memory or when we want to adapt to changing environment.

The dynamic Boltzmann machine (DyBM) is proposed as a time-series model that allows efficient online learning [31, 32]. The per-step computational complexity of the learning rule of a DyBM is independent of the length of the preceding time-series. In Section 5.1, we start by reviewing the DyBM introduced in [31, 32] with relation of its learning rule to spike-timing dependent plasticity (STDP).

In Section 5.2, we study the relaxation of some of the constraints that the DyBM has required in [31, 32] in a way that it becomes more suitable for inference and learning [27]. The primary purpose of these constraints in [31, 32] was to mimic a particular form of STDP. The relaxed DyBM generalizes the original DyBM and allows us to interpret it as a form of logistic regression for time-series data.

In Section 5.3, we review DyBMs dealing with real-valued time-series [27, 8]. These DyBMs are analogous to how Gaussian Boltzmann machines [23, 43, 13] deal with real-valued patterns as opposed to Boltzmann machines [2, 14] for binary values. The Gaussian DyBM can be related to a vector autoregressive (VAR) model [21]. Specifically, we show that a special case of the Gaussian DyBM is a VAR model having additional variables that capture long term dependency of time-series. These additional variables correspond to DyBM's eligibility traces, which represent how recently and frequently spikes arrived from a neuron to another. We also review an extension of the Gaussian DyBM to deal with time-series patterns in continuous space [17].

5.1 Dynamic Boltzmann machines for binary-valued time-series

5.1.1 Finite dynamic Boltzmann machines[3]

The DyBM in [31, 32] is defined as a limit of a sequence of Boltzmann machines (DyBM-T) consisting of T layers as T tends to infinity (see Figure 6). Formally, the DyBM-T is defined as the CRBM (see Section 3.1) having T layers of N visible units ($T-1$ layers of input units and one layer of output units) and no hidden units, so that its conditional energy is defined as

$$E_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[t-T+1,t-1]}) = -\mathbf{b}^\top \mathbf{x}^{[t]} - \sum_{\delta=1}^{T-1} (\mathbf{x}^{[t-\delta]})^\top \, \mathbf{W}^{[\delta]} \, \mathbf{x}^{[t]}, \tag{63}$$

where the weight of the DyBM-T ($\mathbf{W}^{[1]}, \ldots, \mathbf{W}^{[T-1]}$) assumes a particular parametric form with a finite number of parameters that are independent of T, which we discuss in the following.

The parametric form of the weight in the DyBM-T is motivated by observations from biological neural networks [1] but leads to particularly simple, exact, and efficient learning rule. In biological neural networks, STDP has been postulated and supported experimentally. In particular, the weight from a pre-synaptic neuron to a post-synaptic neuron is strengthened, if the post-synaptic neuron fires (generates a spike) shortly *after* the pre-synaptic neuron fires (*i.e.*, long term potentiation or LTP). This weight is weakened, if the post-synaptic neuron fires shortly *before* the pre-synaptic neuron fires (*i.e.*, long term depression or LTD). These dependency on the timing of spikes is missing in the Hebbian rule for the Boltzmann machine (see (36) from [28]).

To have a learning rule with the characteristics of STDP with LTP and LTD, the DyBM-T assumes the weight of the form illustrated in Figure 7. For $\delta > 0$, we define the weight, $w_{i,j}^{[\delta]}$, as the sum of two weights, $\hat{w}_{i,j}^{[\delta]}$ and $w_{j,i}^{[-\delta]}$:

$$w_{i,j}^{[\delta]} = \hat{w}_{i,j}^{[\delta]} + \hat{w}_{j,i}^{[-\delta]}, \tag{64}$$

[3] This section closely follows [31].

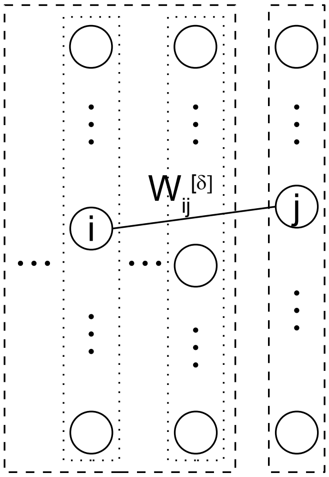


Figure 6: A dynamic Boltzmann machine unfolded through time (Figure 1(c) from [31]).

where

$$\hat{w}_{i,j}^{[\delta]} = \begin{cases} 0 & \text{if } \delta = 0 \\ u_{i,j}\, \lambda^{\delta-d} & \text{if } \delta \geq d \\ -v_{i,j}\, \mu^{-\delta} & \text{otherwise.} \end{cases} \tag{65}$$

for $\lambda, \mu \in [0, 1)$. For simplicity, we assume a single decay rate λ for $\delta \geq d$ and a single decay rate μ for $\delta < d$, as opposed to multiple ones in [31, 32]. For simplicity, we assume that the conduction delay d is uniform for all connections, as opposed to variable conduction delay in [32]. See also [9, 29] for ways to tune the values of the conduction delay.

In Figure 7, the value of $\hat{w}_{i,j}^{[\delta]}$ is high when $\delta = d$, the conduction delay from i-th (pre-synaptic) unit to the j-th (post-synaptic) unit. Namely, the post-synaptic neuron is likely to fire (*i.e.*, $x_j^{[0]} = 1$) immediately after the spike from the pre-synaptic unit arrives with the delay of d (*i.e.*, $x_i^{[-d]} = 1$). This likelihood is controlled by the LTP weight $u_{i,j}$. The value of $\hat{w}_{i,j}^{[\delta]}$ gradually decreases, as δ increases from d. That is, the effect of the stimulus of the spike arrived from the i-th unit diminishes with time [1].

The value of $\hat{w}_{i,j}^{[d-1]}$ is low, suggesting that the post-synaptic unit is unlikely to fire (*i.e.*, $x_j^{[0]} = 1$) immediately *before* the spike from the i-th (pre-synaptic) unit arrives. This unlikelihood is controlled by the LTD weight $v_{i,j}$. As δ decreases from $d-1$, the magnitude of $\hat{w}_{i,j}^{[\delta]}$ gradually decreases [1]. Here, δ can get smaller than 0, and $\hat{w}_{i,j}^{[\delta]}$ with $\delta < 0$ represents the weight between the spike of the pre-synaptic neuron that is generated after the spike of the post-synaptic neuron.

5.1.2 Dynamic Boltzmann machine as a limit of a sequence of finite dynamic Boltzmann machines[4]

The DyBM is defined as a limit of the sequence of DyBM-T as $T \to \infty$. Because each DyBM-T is a CRBM, we can also define the limit of the sequence of the conditional probability defined by DyBM-T, and this limit is considered as the conditional probability defined by the DyBM. Likewise, the conditional energy of the DyBM is defined as the limit of the sequence of the conditional energy of DyBM-T.

[4] This section closely follows [27].

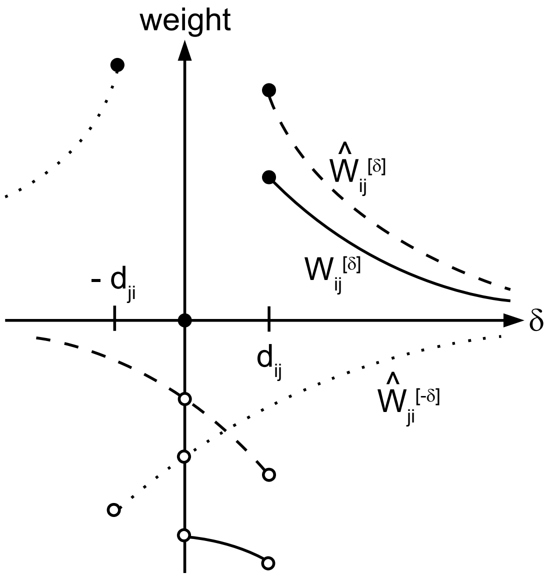


Figure 7: The figure illustrates Equation (64) with particular forms of Equation (65) (Figure 2 from [31]). The horizontal axis represents δ, and the vertical axis represents the value of $w_{i,j}^{[\delta]}$ (solid curves), $\hat{w}_{i,j}^{[\delta]}$ (dashed curves), or $\hat{w}_{j,i}^{[-\delta]}$ (dotted curves). Notice that $w_{i,j}^{[\delta]}$ is defined for $\delta > 0$ and is discontinuous at $\delta = d$. On the other hand, $\hat{w}_{i,j}^{[\delta]}$ and $\hat{w}_{j,i}^{[-\delta]}$ are defined for $-\infty < \delta < \infty$ and discontinuous at $\delta = d_{i,j}$ and $\delta = -d_{j,i}$, respectively, where recall that we assume $d_{i,j} = d_{j,i} = d$ in this paper.

Specifically, as $T \to \infty$, the conditional energy of DyBM-T in (63) converges to

$$E_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}) = -\mathbf{b}^\top \mathbf{x}^{[t]} - \sum_{d=1}^{\infty} (\mathbf{x}^{[t-d]})^\top \, \mathbf{W}^{[d]} \, \mathbf{x}^{[t]}, \tag{66}$$

where the convergence is due to the parametric form (65). This conditional energy in turn defines the conditional distribution via (9), where we now have no hidden units. Although the conditional energy (66) of the DyBM involves an infinite sum, it can be evaluated with a finite sum because of the parametric form (65).

In fact, the DyBM can be understood as an artificial model of a spiking neural network where all computation for inference and learning is performed locally at each synapse using only the information available around the synapse. Specifically, a (pre-synaptic) neuron is connected to a (post-synaptic) neuron via a first-in-first-out (FIFO) queue and a synapse (see Figure 8). At each discrete time t, a neuron i either fires ($x_i^{[t]} = 1$) or not ($x_i^{[t]} = 0$). The spike travels along the FIFO queue and reaches the synapse after conduction delay, d. In other words, the FIFO queue has the length of $d - 1$ and stores, at time t, the spikes that have been generated by the pre-synaptic neuron from time $t - d + 1$ to time $t - 1$.

Each synapse in a DyBM stores a quantity called a synaptic eligibility trace. The value of the synaptic eligibility increases when a spike arrives at the synapse from the FIFO queue; otherwise, it is decreased by a constant factor. Specifically, at time t, the value of the synaptic eligibility trace, $\alpha_i^{[t]}$, that is stored at the synapse from a pre-synaptic neuron i is updated as follows:

$$\alpha_i^{[t]} = \lambda \, (\alpha_i^{[t-1]} + x_i^{[t-d+1]}), \tag{67}$$

where λ is a decay rate and satisfies $0 \le \lambda < 1$. Figure 9 shows an example of how the value of the synaptic eligibility trace changes depending on the spikes arrived at the synapse. Observe that

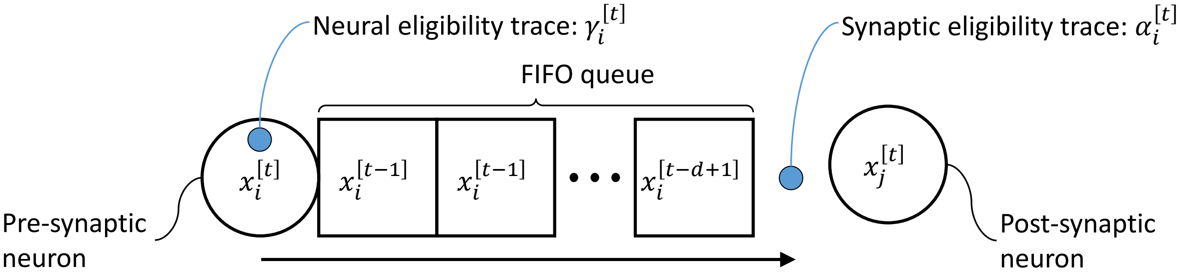


Figure 8: A connection from a (pre-synaptic) neuron i to a (post-synaptic) neuron j in a DyBM.

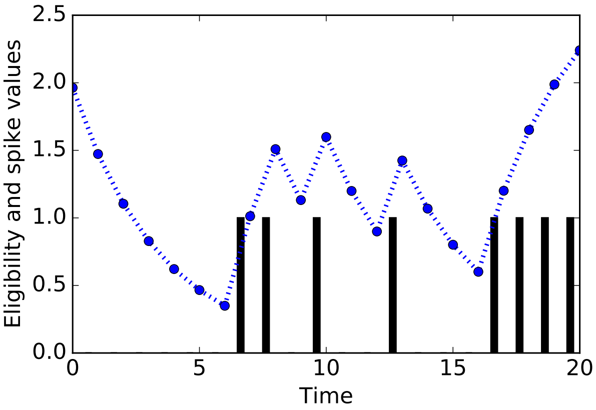


Figure 9: The value of a synaptic or neural eligibility trace as a function of time. For a synaptic eligibility trace at a synapse, the bars represent the spikes arrived from a FIFO queue at that synapse. For a neural eligibility trace at a neuron, the bars represent the spikes generated by that neuron.

$\alpha_i^{[t]}$ represents how recently and frequently spikes arrived from a pre-synaptic neuron i and can be represented non-recursively as follows:

$$\alpha_i^{[t-1]} = \sum_{s=-\infty}^{t-d} \lambda^{t-s-d} \, x_i^{[s]}. \tag{68}$$

Each neuron in a DyBM stores a quantity called a neural eligibility trace[5]. The value of the neural eligibility increases when the neuron fires; otherwise, it is decreased by a constant factor. Specifically, at time t, the value of the neural eligibility trace, $\gamma_i^{[t]}$, at a neuron i is updated as follows:

$$\gamma_i^{[t]} = \mu \, (\gamma_i^{[t-1]} + x_i^{[t]}), \tag{69}$$

where μ is a decay rate and satisfies $0 \le \mu < 1$. Observe that $\gamma_i^{[t]}$ represents how recently and frequently the neuron i has fired and can be represented non-recursively as follows:

$$\gamma_i^{[t-1]} = \sum_{s=-\infty}^{t-1} \mu^{t-s} \, x_j^{[s]} \tag{70}$$

A neuron in a DyBM fires according to the probability distribution that depends on the energy of the DyBM. A neuron is more likely to fire when the energy becomes lower if it fires than otherwise.

[5] We assume a single neural eligibility trace, as opposed to multiple ones in [32], at each neuron.

Let $E_{\theta,j}\big(x_j^{[t]}|\mathbf{x}^{[<t]}\big)$ be the energy associated with a neuron j at time t, which can depend on whether j fires at time t (*i.e.*, $x_j^{[t]}$) as well as the preceding spiking activities of the neurons in the DyBM (*i.e.*, $\mathbf{x}^{[<t]}$). The firing probability of a neuron j is then given by

$$\mathbb{P}_{\theta,j}(x_j^{[t]}|\mathbf{x}^{[<t]}) = \frac{\exp\big(-E_{\theta,j}(x_j^{[t]}|\mathbf{x}^{[<t]})\big)}{\displaystyle\sum_{\tilde{x}\in\{0,1\}} \exp\big(-E_{\theta,j}(\tilde{x}|\mathbf{x}^{[<t]})\big)} \tag{71}$$

for $x_j^{[t]} \in \{0,1\}$. Specifically, $E_{\theta,j}\big(x_j^{[t]}|\mathbf{x}^{[<t]}\big)$ can be represented as follows:

$$E_{\theta,j}\big(x_j^{[t]}|\mathbf{x}^{[<t]}\big) = -b_j\, x_j^{[t]} + E_{\theta,j}^{\mathrm{LTP}}\big(x_j^{[t]}|\mathbf{x}^{[<t]}\big) + E_{\theta,j}^{\mathrm{LTD}}\big(x_j^{[t]}|\mathbf{x}^{[<t]}\big), \tag{72}$$

where b_j is the bias parameter of a neuron j and represents how likely j spikes (j is more likely to fire if b_j has a large positive value), and we define

$$E_{\theta,j}^{\mathrm{LTP}}\big(x_j^{[t]}|\mathbf{x}^{[<t]}\big) \equiv -\sum_{i=1}^{N} u_{i,j}\,\alpha_i^{[t-1]}\,x_j^{[t]} \tag{73}$$

$$E_{\theta,j}^{\mathrm{LTD}}\big(x_j^{[t]}|\mathbf{x}^{[<t]}\big) \equiv \sum_{i=1}^{N} v_{i,j}\,\beta_i^{[t-1]}\,x_j^{[t]} + \sum_{k=1}^{N} v_{j,k}\,\gamma_k^{[t-1]}\,x_j^{[t]}, \tag{74}$$

where $\beta_i^{[t-1]}$ represents how soon and frequently spikes will arrive at the synapse from the FIFO queues from i to j:

$$\beta_i^{[t-1]} \equiv \sum_{s=t-d+1}^{t-1} \mu^{s-t}\,x_i^{[s]}. \tag{75}$$

Although $\beta_i^{[t-1]}$ can also be represented in a recursive manner, recursively computed $\beta_i^{[t-1]}$ is prone to numerical instability.

In (73), the summation with respect to i is over all of the pre-synaptic neurons that are connected to j. Here, $u_{i,j}$ is the weight parameter from i to j and represents the strength of Long Term Potentiation (LTP). This weight parameter is thus referred to as LTP weight. A neuron j is more likely to fire ($x_j^{[t]} = 1$) when $\alpha_i^{[t-1]}$ is large for a pre-synaptic neuron i connected to j (spikes have recently arrived at j from i) and the corresponding $u_{i,j}$ is positive and large (LTP from i to j is strong).

In (74), the summation with respect to i is over all of the pre-synaptic neurons that are connected to j, and the summation with respect to k is over all of the post-synaptic neurons which j is connected to. Here, $v_{i,j}$ represents the strength of Long Term Depression from i to j and referred to as LTD weight. The neuron j is less likely to fire when β_i is large for a pre-synaptic neuron i connected to j (spikes will soon and frequently reach j from i) and the corresponding $v_{i,j}$ is positive and large (LTD from i to j is strong). The second term in (74) represents that a pre-synaptic neuron j is less likely to fire if a post-synaptic neuron has recently and frequently fired (γ_k is large), and the strength of this LTD is given by $v_{j,k}$. Notice that the timing of a spike is measured with respect to when the spike reaches synapse, where the spike from a pre-synaptic neuron has the delay d, and the spike from a post-synaptic neuron reaches immediately.

The learning rule of the DyBM has been derived in a way that it maximizes the log likelihood of given time-series with respect to the probability distribution given by (71) [32]. Specifically, at time t, the DyBM updates its parameters according to

$$b_j \leftarrow b_j + \eta\,\big(x_j^{[t]} - \mathbb{E}_{\theta,j}[X_j^{[t]} \mid \mathbf{x}^{[<t]}]\big) \tag{76}$$

$$u_{i,j} \leftarrow u_{i,j} + \eta\,\alpha_i^{[t-1]}\,\big(x_j^{[t]} - \mathbb{E}_{\theta,j}[X_j^{[t]} \mid \mathbf{x}^{[<t]}]\big) \tag{77}$$

$$v_{i,j} \leftarrow v_{i,j} + \eta\,\beta_i^{[t-1]}\,\big(\mathbb{E}_{\theta,j}[X_j^{[t]} \mid \mathbf{x}^{[<t]}] - x_j^{[t]}\big) + \eta\,\gamma_j^{[t-1]}\,\big(\langle X_i^{[t]}\rangle - x_i^{[t]}\big) \tag{78}$$

for each of neurons i and j, where η is a learning rate, $x_j^{[t]}$ is the training data given to j at time t, and $\mathbb{E}_{\theta,j}[X_j^{[t]} \mid \mathbf{x}^{[<t]}]$ denotes the expected value of $x_j^{[t]}$ (*i.e.*, firing probability of a neuron j at time t) according to the probability distribution given by (71). By following stochastic gradient methods [6, 18, 10, 41, 33], the learning rate η may be adjusted over time t.

5.1.3 Relation to spike-timing dependent plasticity[6]

In spike-timing dependent plasticity (STDP), the amount of the change in the weight between two neurons that fired together depends on the precise timings when the two neurons fired. STDP supplements the Hebbian rule [11] and has been experimentally confirmed in biological neural networks [5].

In (77), $u_{i,j}$ is increased (LTP gets stronger) when $x_j^{[t]} = 1$ is given to j. Then j becomes more likely to fire when spikes from i have recently and frequently arrived at j (*i.e.*, $\alpha_i^{[\cdot]}$ is large). This amount of the change in $u_{i,j}$ depends on $\alpha_i^{[t-1]}$, exhibiting a key property of STDP. In particular, $u_{i,j}$ is increased by a large amount if spikes from i have recently and frequently arrived at j.

According to the second term on the right-hand side of (78), $v_{i,j}$ is increased (LTD gets stronger) when $x_j^{[t]} = 0$ is given to a post-synaptic neuron j. Then j becomes less likely to fire when spikes from i are expected to reach j soon (*i.e.*, $\beta_i^{[\cdot]}$ is large). This amount of the change in $v_{i,j}$ is large if there are spikes in the FIFO queue from i to j and they are close to j. According to the last term of (78), $v_{i,j}$ is increased when $x_i^{[t]} = 0$ is given to the pre-synaptic i, and this amount of the change in $v_{i,j}$ is proportional to γ_j (*i.e.*, how frequently and recently the post-synaptic j has fired). This learning rule of (78) thus exhibits some of the key properties of LTD with STDP.

In (76), b_j is increased when $x_j^{[t]} = 1$ is given to j, so that j becomes more likely to fire (in accordance with the training data), but the amount of the change in b_j is small if j is already likely to fire ($\mathbb{E}_{\theta,j}[X_j^{[t]} \mid \mathbf{x}^{[<t]}] \approx 1$). This dependency on $\mathbb{E}_{\theta,j}[X_j^{[t]} \mid \mathbf{x}^{[<t]}]$ can be considered as a form of homeostatic plasticity [42, 20].

Related work There has been a significant amount of the prior work towards understanding STDP from the perspectives of machine learning [26, 4, 35]. For example, Nessler et al. show that STDP can be understood as approximating the expectation maximization (EM) algorithm [26]. Nessler et al. study a particularly structured (winner-take-all) network and its learning rule for maximizing the log likelihood of given static patterns. On the other hand, the DyBM does not assume particular structures in the network, and the learning rule having the properties of STDP applies for any synapse in the network. Also, the learning rule of the DyBM maximizes the log likelihood of given time-series, and its learning rule does not involve approximations beyond what is assumed in stochastic gradient methods [6].

5.2 Giving flexibility to the DyBM[7]

It has been shown in [32] that the DyBM in Section 5.1 has the capability of associative memory and anomaly detection for sequential patterns, but the applications of the DyBM have been limited to simple tasks with relatively low dimensional time-series. In [27], we relax some of the constraints of this DyBM in a way that it gives more flexibility that is useful for learning and inference.

Specifically, observe that the first term on the right-hand side of (74) can be rewritten with the

[6] This section closely follows [27].

[7] This section closely follows [27].

definition of $\beta_i^{[t-1]}$ in (75) as follows:

$$\sum_{i=1}^{N} v_{i,j}\, \beta_i^{[t-1]}\, x_j^{[t]} = \sum_{i=1}^{N} \sum_{s=t-d+1}^{t-1} v_{i,j}\, \mu^{s-t}\, x_i^{[s]}\, x_j^{[t]} \tag{79}$$

$$= \sum_{i=1}^{N} \sum_{\delta=1}^{d-1} v_{i,j}^{[\delta]}\, x_i^{[t-\delta]}\, x_j^{[t]}, \tag{80}$$

where we let $v_{i,j}^{[\delta]} \equiv v_{i,j}\, \mu^{-\delta}$. Here, $v_{i,j}^{[\delta]}$ represents how unlikely j fires at time t if i fired at time $t-\delta$. The parametric form of $v_{i,j}^{[\delta]} \equiv v_{i,j}\, \mu^{-\delta}$ assumes that this LTD weight decays geometrically as the interval, δ, between the two spikes increases.

In the following, we relax this constraint on $v_{i,j}^{[\delta]}$ for $\delta = 1, \ldots, d-1$ and assumes that these LTD weights can take independent values. Then the energy of the DyBM with N neurons can be represented conveniently with matrix and vector operations:

$$E_\theta(\mathbf{x}^{[t]} | \mathbf{x}^{[<t]}) \equiv \sum_{j=1}^{N} E_{\theta,j}(x_j^{[t]} | \mathbf{x}^{[<t]}) \tag{81}$$

$$- \mathbf{b}^\top \mathbf{x}^{[t]} - (\boldsymbol{\alpha}_\lambda^{[t-1]})^\top \mathbf{U}\, \mathbf{x}^{[t]} + \sum_{\delta=1}^{d-1} (\mathbf{x}^{[t-\delta]})^\top \mathbf{V}^{[\delta]}\, \mathbf{x}^{[t]} + (\mathbf{x}^{[t]})^\top \mathbf{V}\, \boldsymbol{\gamma}_\mu^{[t-1]}, \tag{82}$$

where $\mathbf{b} \equiv (b_j)_{j=1,\ldots,N}$ is a vector, $\mathbf{U} \equiv (u_{i,j})_{(i,j)\in\{1,\ldots,N\}^2}$ is a matrix, and other boldface letters are defined analogously (a vector is lowercase and a matrix is uppercase). For eligibility traces ($\boldsymbol{\alpha}_\lambda^{[t-1]}$ and $\boldsymbol{\gamma}_\mu^{[t-1]}$), we append the subscript to explicitly represent the dependency on the decay rate (λ and μ). The functional form of the energy completely determines the dynamics of a DyBM, and relaxing its constraints allows the DyBM to represent a wider class of dynamical systems.

Notice that the last term of (82) can be divided into two terms:

$$(\mathbf{x}^{[t]})^\top \mathbf{V}\, \boldsymbol{\gamma}_\mu^{[t-1]} = (\boldsymbol{\gamma}_\mu^{[t-1]})^\top \mathbf{V}\, \mathbf{x}^{[t]} \tag{83}$$

$$= (\boldsymbol{\alpha}_\mu^{[t-1]})^\top \mathbf{V}\, \mathbf{x}^{[t]} + \sum_{\delta=1}^{d-1} (\mathbf{x}^{[t-\delta]})^\top \hat{\mathbf{V}}^{[\delta]}\, \mathbf{x}^{[t]}, \tag{84}$$

where $\boldsymbol{\alpha}_\mu^{[t-1]}$ is the same as the vector of synaptic eligibility traces but with the decay rate μ, and $\hat{\mathbf{V}}^{[\delta]} \equiv \mu^{-\delta}\, \mathbf{V}$. Comparing (84) and (82), we find that, without loss of generality, the energy of the DyBM can be represented with the following form:

$$E_\theta(\mathbf{x}^{[t]} | \mathbf{x}^{[<t]}) = -\left(\mathbf{b}^\top + \sum_{\delta=1}^{d-1} (\mathbf{x}^{[t-\delta]})^\top \mathbf{W}^{[\delta]} + \sum_{\ell=1}^{L} (\boldsymbol{\alpha}_{\lambda_\ell}^{[t-1]})^\top \mathbf{U}^{[\ell]} \right) \mathbf{x}^{[t]}, \tag{85}$$

where we define $\mathbf{W}^{[\delta]} = -\mathbf{V}^{[\delta]} - \hat{\mathbf{V}}^{[\delta]}$. The energy in (85) reduces to the original energy in (72) when $\mathbf{W}^{[\delta]} = -\mu^{-\delta}\, \mathbf{V} - \mu^{\delta}\, \mathbf{V}^\top$, $\mathbf{U}^{[1]} = \mathbf{U}$, $\mathbf{U}^{[2]} = -\mu^{d}\, \mathbf{V}^\top$, $\lambda_1 = \lambda$, $\lambda_2 = \mu$, and $L = 2$. With $L > 2$, one can also incorporate multiple synaptic or neural eligibility traces with varying decay rates in [32].

Equivalently, we can represent the energy using neural eligibility traces, $\boldsymbol{\gamma}_{\mu_\ell}$, instead of synaptic eligibility traces, $\boldsymbol{\alpha}_{\lambda_\ell}$, as follows:

$$E_\theta(\mathbf{x}^{[t]} | \mathbf{x}^{[<t]}) = -\left(\mathbf{b}^\top + \sum_{\delta=1}^{d-1} (\mathbf{x}^{[t-\delta]})^\top \mathbf{W}^{[\delta]} + \sum_{\ell=1}^{L} (\boldsymbol{\gamma}_{\mu_\ell}^{[t-1]})^\top \mathbf{V}_\ell \right) \mathbf{x}^{[t]}. \tag{86}$$

5.2.1 Learning rule in vector-matrix notations

The learning rule corresponding to the representation with (85) is as follows:

$$\mathbf{b} \leftarrow \mathbf{b} + \eta\,(\mathbf{x}^{[t]} - \mathbb{E}_\theta[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}]) \tag{87}$$

$$\mathbf{W}^{[\delta]} \leftarrow \mathbf{W}^{[\delta]} + \eta\,\mathbf{x}^{[t-\delta]}\,(\mathbf{x}^{[t]} - \mathbb{E}_\theta[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}])^\top \tag{88}$$

$$\mathbf{U}^{[\ell]} \leftarrow \mathbf{U}^{[\ell]} + \eta\,\boldsymbol{\alpha}_{\lambda_\ell}^{[t-1]}\,(\mathbf{x}^{[t]} - \mathbb{E}_\theta[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}])^\top \tag{89}$$

for each δ and each ℓ, where $\mathbb{E}_\theta[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}]$ is the conditional expectation with respect to

$$\mathbb{P}_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}) = \frac{\exp(-E_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}))}{\sum_{\tilde{\mathbf{x}}^{[t]}} \exp(-E_\theta(\tilde{\mathbf{x}}^{[t]} \mid \mathbf{x}^{[<t]}))}. \tag{90}$$

Specifically,

$$\mathbb{E}_\theta[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}] = \frac{\exp(\mathbf{m}^{[t]})}{1 + \exp(\mathbf{m}^{[t]})} \tag{91}$$

with

$$\boldsymbol{m}^{[t]} \equiv \mathbf{b}^\top + \sum_{\delta=1}^{d-1} (\mathbf{x}^{[t-\delta]})^\top \mathbf{W}^{[\delta]} + \sum_{\ell=1}^{L} (\boldsymbol{\alpha}_{\lambda_\ell}^{[t-1]})^\top \mathbf{U}^{[\ell]}, \tag{92}$$

where exponentiation and division of vectors are elementwise.

The form of (91) implies that the DyBM is a kind of a logit model, where the feature vector, $(\mathbf{x}^{[t-d+1]}, \ldots, \mathbf{x}^{[t-1]}, \boldsymbol{\alpha}_\lambda^{[t-1]}, \boldsymbol{\alpha}_\mu^{[t-1]})$, depends on the prior values, $\mathbf{x}^{[<t]}$, of the time-series. By applying the learning rules given in (76)-(78) to given time-series, we can learn the parameters of the DyBM or equivalently the parameters of the logit model (*i.e.*, $\mathbf{b}$, $\mathbf{W}^{[\delta]}$ for $\delta = 1, \ldots, d-1$, and $\mathbf{U}^{[\ell]}$ for $\ell = 1, \ldots, L$) in (91).

5.3 Dynamic Boltzmann machines for real-valued time-series

5.3.1 Gaussian dynamic Boltzmann machines[8]

In this section, we show how a DyBM can deal with real-valued time-series in the form of a Gaussian DyBM [27, 8]. A Gaussian DyBM assumes that $x_j^{[t]}$ follows a Gaussian distribution for each j:

$$p_\theta^{(j)}(x_j^{[t]} | \mathbf{x}^{[<t]}) = \frac{1}{\sqrt{2\,\pi\,\sigma_j^2}} \exp\Big(-\frac{\big(x_j^{[t]} - m_j^{[t]}\big)^2}{2\,\sigma_j^2}\Big), \tag{93}$$

where $m_j^{[t]}$ is given by (92), and σ_j^2 is a variance parameter. This Gaussian distribution is in contrast to the Bernoulli distribution of the DyBM given by (71).

The conditional energy of the Gaussian DyBM can be represented as follows:

$$\begin{aligned} E_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}) &= \sum_{j=1}^{N} \frac{(x_j^{[t]} - m_j^{[t]})^2}{2\,\sigma_j^2} & (94) \\ &= \sum_{j=1}^{N} \frac{(x_j^{[t]} - b_j)^2}{2\,\sigma_j^2} - \sum_{\delta=1}^{d-1}\sum_{i=1}^{N}\sum_{j=1}^{N} x_i^{[t-\delta]}\,w_{i,j}^{[\delta]}\,x_j^{[t]} - \sum_{\ell=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{i,\lambda_\ell}^{[t-1]}\,u_{i,j}^{[\ell]}\,x_j^{[t]} + C, & (95) \end{aligned}$$

[8] This section closely follows [27].

where C is the term that does not depend on $\mathbf{x}^{[t]}$. Because C is canceled out between the numerator and the denominator in (9), we omit it from the conditional energy. By letting $\mathbf{W}_\sigma^{[\delta]}$ be the matrix whose $(i,j))$ element is $w_{i,j}^{[\delta]}/\sigma_j^2$ and $\mathbf{U}_\sigma^{[\ell]}$ be the matrix whose $(i,j))$ element is $u_{i,j}^{[\ell]}/\sigma_j^2$, the conditional energy of the Gaussian DyBM can be represented as follows:

$$E_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}) = \sum_{j=1}^{N} \frac{(x_j^{[t]} - b_j)^2}{2\,\sigma_j^2} - \sum_{\delta=1}^{d-1} (\mathbf{x}^{[t-\delta]})^\top \, \mathbf{W}_\sigma^{[\delta]} \, \mathbf{x}^{[t]} - \sum_{\ell=1}^{L} (\alpha_{\lambda_\ell}^{[t-1]})^\top \, \mathbf{U}_\sigma^{[\ell]} \, \mathbf{x}^{[t]}. \tag{96}$$

The conditional energy of the Gaussian DyBM may be compared against the energy of the Gaussian Bernoulli restricted Boltzmann machine (see [19] or (181) from [28]).

We now derive a learning rule for the Gaussian DyBM in a way that it maximizes the log-likelihood of given time-series $\mathbf{x}$:

$$\sum_t \log p_\theta(\mathbf{x}^{[t]} | \mathbf{x}^{[<t]}) = \sum_t \sum_{i=1}^{N} \log p_i(x_i^{[t]} | \mathbf{x}^{[-\infty,t-1]}), \tag{97}$$

where the summation over t is over all of the time steps of $\mathbf{x}$, and the conditional independence between $x_i^{[t]}$ and $x_j^{[t]}$ for $i \neq j$ given $\mathbf{x}^{[<t]}$ is the fundamental property of the DyBM as shown in [32].

The approach of stochastic gradient is to update the parameters of the Gaussian DyBM at each step, t, according to the gradient of the conditional probability density of $\mathbf{x}^{[t]}$:

$$\nabla \log p_\theta(\mathbf{x}^{[t]} | \mathbf{x}^{[<t]}) = - \sum_{i=1}^{N} \Big(\frac{1}{2} \nabla \log \sigma_i^2 + \nabla \frac{(x_i^{[t]} - m_i^{[t]})^2}{2\,\sigma_i^2} \Big), \tag{98}$$

where the equality follow from (93). From (98) and (92), we can derive the derivative with respect to each parameter as follows:

$$\frac{\partial}{\partial b_j} \log p_\theta(\mathbf{x}^{[t]} | \mathbf{x}^{[-\infty,t-1]}) = \frac{x_j^{[t]} - \mu_j^{[t]}}{\sigma_j^2} \, x_j^{[t]} \tag{99}$$

$$\frac{\partial}{\partial u_{i,j}} \log p_\theta(\mathbf{x}^{[t]} | \mathbf{x}^{[-\infty,t-1]}) = \frac{x_j^{[t]} - \mu_j^{[t]}}{\sigma_j^2} \, \alpha_{i,j}^{[t-1]} \tag{100}$$

$$\frac{\partial}{\partial w_{i,j}^{[\delta]}} \log p_\theta(\mathbf{x}^{[t]} | \mathbf{x}^{[-\infty,t-1]}) = \frac{x_j^{[t]} - \mu_j^{[t]}}{\sigma_j^2} \, x_i^{[t-\delta]} \tag{101}$$

$$\frac{\partial}{\partial \sigma_j} \log p_\theta(\mathbf{x}^{[t]} | \mathbf{x}^{[-\infty,t-1]}) = -\frac{1}{\sigma_j} + \frac{\big(x_j^{[t]} - \mu_j^{[t]}\big)^2}{\sigma_j^3}, \tag{102}$$

where $\delta \in \{1, \ldots, d-1\}$, $\ell \in \{1, \ldots, \}$, and $i, j \in \{1, \ldots, N\}$.

These parameters are thus updated with learning rate η as follows:

$$\mathbf{b} \leftarrow \mathbf{b} + \eta \, \frac{\mathbf{x}^{[t]} - \mathbf{m}^{[t]}}{\boldsymbol{\sigma}^2} \tag{103}$$

$$\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} + \eta \, \frac{\big(\mathbf{x}^{[t]} - \mathbf{m}^{[t]}\big)^2 - \boldsymbol{\sigma}^2}{\boldsymbol{\sigma}^3} \tag{104}$$

$$\mathbf{W}^{[\delta]} \leftarrow \mathbf{W}^{[\delta]} + \eta \, \mathbf{x}^{[t-\delta]} \left(\frac{\mathbf{x}^{[t]} - \mathbf{m}^{[t]}}{\boldsymbol{\sigma}^2} \right)^\top \tag{105}$$

$$\mathbf{U}^{[\ell]} \leftarrow \mathbf{U}^{[\ell]} + \eta \, \boldsymbol{\alpha}_{\lambda_\ell}^{[t-1]} \left(\frac{\mathbf{x}^{[t]} - \mathbf{m}^{[t]}}{\boldsymbol{\sigma}^2} \right)^\top \tag{106}$$

where division and exponentiation of vectors are elementwise, and $\mathbf{m}^{[t]}$ is given by (92).

The maximum likelihood estimator of $\mathbf{x}^{[t]}$ by the Gaussian DyBM is given by $\boldsymbol{m}^{[t]}$ in (92). The Gaussian DyBM can thus be understood as a modification to the standard vector autoregressive (VAR) model. Specifically, the last term in the right-hand side of (92) involves eligibility traces, which can be understood as features of historical values, $\mathbf{x}^{(-\infty,t-d]}$, and are added as new variables to the VAR model. Because the value of the eligibility traces can depend on the infinite past, the Gaussian DyBM can take into account the history beyond the lag $d-1$.

5.3.2 Natural gradients[9]

In this section, we study a learning rule based on natural gradient for the Gaussian DyBM. Consider a stochastic model that gives the probability density of a pattern $\mathbf{x}$ as $p_\theta(\mathbf{x})$. With natural gradients [3], the parameters, θ, of the stochastic model are updated as follows:

$$\theta_{t+1} = \theta_t - \eta\, G^{-1}(\theta_t)\, \nabla \log p_\theta(\mathbf{x}) \tag{107}$$

at each step t, where η is the learning rate, and $G(\theta)$ denotes the Fisher information matrix:

$$G(\theta) \equiv \int p_\theta(\mathbf{x}) \left(\nabla \log p_\theta(\mathbf{x})\, \nabla \log p_\theta(\mathbf{x})^\top\right) d\mathbf{x}. \tag{108}$$

Due to the conditional independence in (97), it suffices to derive a natural gradient for each Gaussian unit. Here, we consider the parametrization with mean m and variance $v \equiv \sigma^2$. The probability density function of a Gaussian distribution is represented with this parametrization as follows:

$$p(x; m, v) = \frac{1}{\sqrt{2\pi\, v}} \exp\left(-\frac{(x-m)^2}{2v}\right). \tag{109}$$

The log likelihood of x is then given by

$$\log p(x; m, v) = -\frac{(x-m)^2}{2v} - \frac{1}{2}\log v - \frac{1}{2}\log 2\pi. \tag{110}$$

Hence, the gradient and the inverse Fisher information matrix in (107) are given as follows:

$$\nabla \log p_\theta(\mathbf{x}) = \left(\begin{array}{c} \frac{x-m}{v} \\ \frac{(x-m)^2}{2v^2} - \frac{1}{2v} \end{array}\right) \tag{111}$$

$$G^{-1}(\theta) = \left(\begin{array}{cc} \frac{1}{v} & 0 \\ 0 & \frac{1}{2v^2} \end{array}\right)^{-1} = \left(\begin{array}{cc} v & 0 \\ 0 & 2v^2 \end{array}\right), \tag{112}$$

The parameters $\theta_t \equiv (m_t, v_t)$ are then updated as follows:

$$m_{t+1} = m_t + \eta\, (x - m_t) \tag{113}$$

$$v_{t+1} = v_t + \eta\, \left((x-m_t)^2 - v_t\right). \tag{114}$$

In the context of a Gaussian DyBM, the mean is given by (92), where $m_j^{[t]}$ is linear with respect to b_j, $w_{i,j}$, and $u_{i,j}^{[\ell]}$. Also, the variance is given by σ_j^2. Hence, the natural gradient gives the learning rules for these parameters as follows:

$$\mathbf{b} \leftarrow \mathbf{b} + \eta\, (\mathbf{x}^{[t]} - \mathbf{m}^{[t]}) \tag{115}$$

$$\boldsymbol{\sigma}^2 \leftarrow \boldsymbol{\sigma}^2 + \eta\, \left(\left(\mathbf{x}^{[t]} - \mathbf{m}^{[t]}\right)^2 - \boldsymbol{\sigma}^2\right) \tag{116}$$

$$\mathbf{W}^{[\delta]} \leftarrow \mathbf{W}^{[\delta]} + \eta\, \mathbf{x}^{[t-\delta]}\, (\mathbf{x}^{[t]} - \mathbf{m}^{[t]})^\top \tag{117}$$

$$\mathbf{U}^{[\ell]} \leftarrow \mathbf{U}^{[\ell]} + \eta\, \boldsymbol{\alpha}_{\lambda_\ell}^{[t-1]}\, (\mathbf{x}^{[t]} - \mathbf{m}^{[t]})^\top, \tag{118}$$

where the exponentiation of a vector is elementwise. We can compare (115)-(118) against what the standard gradient gives in (103)-(106).

[9] This section closely follows [27].

5.3.3 Using nonlinear features in Gaussian DyBMs

The Gaussian DyBM is a linear model and has limited capability in modeling complex time-series. A way to take into account non-linear features of time-series with a Gaussian DyBM is to apply non-linear mapping to input time-series and feed the resulting non-linear features as additional input to the Gaussian DyBM. An example of such non-linear mapping is an echo state network (ESN) [16].

An ESN maps an input sequence, $\mathbf{x}$, into $\boldsymbol{\psi}$ recursively as follows:

$$\boldsymbol{\psi}^{[t]} = (1-\rho)\,\boldsymbol{\psi}^{[t-1]} + \rho\,\tanh\left(\mathbf{W}_{\mathrm{rec}}\,\boldsymbol{\psi}^{[t-1]} + \mathbf{W}_{\mathrm{in}}\,\mathbf{x}^{[t]}\right), \tag{119}$$

where $\mathbf{W}_{\mathrm{rec}}$ and $\mathbf{W}_{\mathrm{in}}$ are randomly chosen and fixed parameters[10], and ρ is a leak parameter satisfying $0 < \rho < 1$. In (119), tanh is a hyperbolic tangent function but may be replaced with other nonlinear functions such as a sigmoid function.

An eligibility trace may be considered as a linear counterpart of the nonlinear features created by an ESN. Because these features are generated by mappings with fixed parameters and just given as input to a Gaussian DyBM, the learning rules for the Gaussian DyBM stay unchanged. The nonlinear DyBM in [8] uses an ESN in a slightly different manner.

5.4 Functional dynamic Boltzmann machines

We now review a functional DyBM, which models time-series of functions (patterns over a continuous space $\mathcal{Z}$) [17]. Recall that a Gaussian DyBM defines the conditional distribution of the next real-valued vector given the preceding sequence of real-valued vectors. A functional DyBM defines the conditional distribution of the next function (*i.e.*, $g^{[t]}$) given the preceding sequence of partial observations of preceding functions. At each time s, a set of points $Z^{[s]} \equiv (z_i^{[s]})_{i=1,\ldots,N_s}$ is observed, where N_s is the number of points that are observed at s.

The functional DyBM assumes that the conditional distribution of $g^{[t]}(\cdot)$ is given by a Gaussian process, whose mean $\mu^{[t]}(\cdot)$ varies over time depending on preceding functions as follows:

$$\mu^{[t]}(z) = b(z) + \sum_{\delta=1}^{d-1} \int_{\mathcal{Z}} w^{[\delta]}(z,z')\, g^{[t-\delta]}(z')\,\mathrm{d}z' + \sum_{\ell=1}^{L} \int_{\mathcal{Z}} u_\ell(z,z')\,\alpha_\ell^{[t-1]}(z')\,\mathrm{d}z' \tag{120}$$

for $x \in \mathcal{Z}$, where $b(\cdot)$ is a functional bias, $w^{[\delta]}(\cdot,\cdot)$ and $u_\ell(\cdot,\cdot)$ are functional weight for each δ and for each ℓ, and

$$\alpha_\ell^{[t-1]}(\cdot) = \sum_{s=-\infty}^{t-d} \lambda_\ell^{t-s-d}\, g^{[s]}(\cdot) \tag{121}$$

is a functional eligibility trace for each ℓ. The covariance $k_{\sigma^2}(\cdot,\cdot)$ of the Gaussian process consists of two components such that

$$k_{\sigma^2}(z,z') = k(z,z') + \sigma^2\,\delta(z,z'), \tag{122}$$

where $k(\cdot,\cdot)$ is a arbitrary kernel, $\delta(\cdot,\cdot)$ is a delta function, and σ is a hyperparameter.

For tractability, Kajino proposes particular parametrization for the functional bias and functional weight [17]. Let $P = (p_1,\ldots,p_M)$ be a set of arbitrarily selected M points in $\mathbf{Z}$

$$b(z) = k_{\sigma^2}(z,P)\,\mathbf{b} \tag{123}$$

$$w^{[\delta]}(z,z') = k_{\sigma^2}(z,P)\,\mathbf{W}^{[\delta]}\,k_{\sigma^2}(P,z') \tag{124}$$

$$u_\ell(z,z') = k_{\sigma^2}(z,P)\,\mathbf{U}^{[\ell]}\,k_{\sigma^2}(P,z') \tag{125}$$

[10] The spectral radius of $\mathbf{W}_{\mathrm{rec}}$ is set smaller than 1.

for each δ and each ℓ, where

$$k_{\sigma^2}(z, P) \equiv (k_{\sigma^2}(z, p_i))_{i=1,\ldots,M} \tag{126}$$

is a row vector, and

$$k_{\sigma^2}(P, z') \equiv (k_{\sigma^2}(p_i, z'))_{i=1,\ldots,M} \tag{127}$$

is a column vector.

Because $g^{[t]}$ is in the reproducing kernel Hilbert space with kernel k_{σ^2}, substituting (123)-(125) into (120) gives the following expression:

$$\mu_\theta^{[t]}(z) = k_{\sigma^2}(z, P)\bigg(\mathbf{b} + \sum_{\delta=1}^{d-1} \mathbf{W}^{[\delta]}\, g^{[t-\delta]}(P) + \sum_{\ell=1}^{L} \mathbf{U}^{[\ell]}\, \alpha_\ell^{[t-1]}(P)\bigg), \tag{128}$$

where $g^{[t-\delta]}(P)$ is a column vector with i-th element being $g^{[t-\delta]}(p_i)$, and the eligibility-trace vector $\alpha_\ell^{[t-1]}(P)$ can be recursively updated as follows:

$$\alpha_\ell^{[t]}(P) = \lambda_\ell \left(\alpha_\ell^{[t-1]}(P) + g^{[t-d+1]}(P)\right). \tag{129}$$

Here, we use $\theta \equiv (\mathbf{b}, \mathbf{W}^{[1]}, \ldots, \mathbf{W}^{[d-1]}, \mathbf{U}^{[1]}, \ldots, \mathbf{U}^{[L]})$ to collectively denote the parameters.

While $g^{[s]}(p_i)$ for $i \in [1, M]$ is not observed, Kajino uses a maximum a posteriori (MAP) estimator $\hat{g}^{[s]}(p_i)$ in [17]:

$$\hat{g}^{[s]}(p_i) = \mu_\theta^{[s]}(p_i) + k(p_i, Z^{[t]})\, k_{\sigma^2}(Z^{[t]}, Z^{[t]})^{-1} \left(g^{[t]}(Z^{[t]}) - \mu_\theta^{[t]}(Z^{[t]}))\right), \tag{130}$$

where $k_{\sigma^2}(Z^{[t]}, Z^{[t]})$ is an $N_s \times N_s$ matrix with (i, j)-th element being $k_{\sigma^2}(z_i^{[t]}, z_j^{[t]})$, and $\mu_\theta^{[t]}(Z^{[t]})$ is a column vector defined analogously to $g^{[t]}(Z^{[t]})$.

The objective of learning a functional DyBM is to maximize the log likelihood of observed values. The conditional probability density of the functional values of locations $Z^{[t]}$ at time t is given by

$$p_\theta(g^{[t]}(Z^{[t]}) \mid g^{[<t]}) \sim \exp\Big(-\frac{1}{2}\big(g^{[t]}(Z^{[t]}) - \mu_\theta^{[t]}(Z^{[t]})\big)^\top k_{\sigma^2}(Z^{[t]}, Z^{[t]})^{-1} \left(g^{[t]}(Z^{[t]}) - \mu_\theta^{[t]}(Z^{[t]})\right)\Big). \tag{131}$$

The objective is thus to maximize

$$f(\theta) \equiv \sum_t f_t(\theta), \tag{132}$$

where

$$f_t(\theta) \equiv \log p_\theta(g^{[t]}(Z^{[t]}) \mid g^{[<t]}) \tag{133}$$

$$= -\frac{1}{2}\big(g^{[t]}(Z^{[t]}) - \mu_\theta^{[t]}(Z^{[t]})\big)^\top k_{\sigma^2}(Z^{[t]}, Z^{[t]})^{-1} \left(g^{[t]}(Z^{[t]}) - \mu_\theta^{[t]}(Z^{[t]})\right) + C, \tag{134}$$

where C is the term independent of θ.

The gradient of $f_t(\theta)$ is given by

$$\nabla f_t(\theta) = \nabla \mu_\theta^{[t]}(Z^{[t]})^\top\, k_{\sigma^2}(Z^{[t]}, Z^{[t]})^{-1} \left(g^{[t]}(Z^{[t]}) - \mu_\theta^{[t]}(Z^{[t]})\right), \tag{135}$$

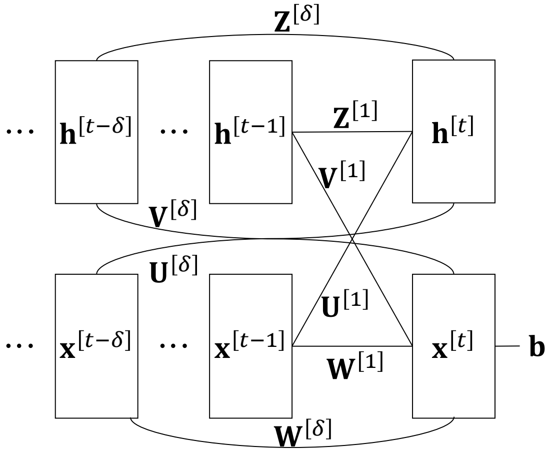


Figure 10: A dynamic Boltzmann machine with hidden units (modified Figure 1 from [30]).

where (128) gives

$$\frac{\partial}{\partial b_i}\mu_\theta^{[t]}(Z^{[t]})^\top = k_{\sigma^2}(p_i, Z^{[t]}) \tag{136}$$

$$\frac{\partial}{\partial w_{i,j}^{[\delta]}}\mu_\theta^{[t]}(Z^{[t]})^\top = k_{\sigma^2}(p_i, Z^{[t]})\, g^{[t-\delta]}(p_j) \tag{137}$$

$$\frac{\partial}{\partial u_{i,j}^{[\ell]}}\mu_\theta^{[t]}(Z^{[t]})^\top = k_{\sigma^2}(p_i, Z^{[t]})\, \alpha_\ell^{[t-1]}(p_j) \tag{138}$$

for each i, j, δ, ℓ.

The gradient implies the following learning rule with stochastic gradient:

$$\mathbf{b} \leftarrow \mathbf{b} + \eta\, k_{\sigma^2}(P, Z^{[t]})\, k_{\sigma^2}(Z^{[t]}, Z^{[t]})^{-1}\left(g^{[t]}(Z^{[t]}) - \mu_\theta^{[t]}(Z^{[t]})\right) \tag{139}$$

$$\mathbf{W}^{[\delta]} \leftarrow \mathbf{W}^{[\delta]} + \eta\, k_{\sigma^2}(P, Z^{[t]})\, k_{\sigma^2}(Z^{[t]}, Z^{[t]})^{-1}\left(g^{[t]}(Z^{[t]}) - \mu_\theta^{[t]}(Z^{[t]})\right) g^{[t-\delta]}(P)^\top \tag{140}$$

$$\mathbf{U}^{[\ell]} \leftarrow \mathbf{U}^{[\ell]} + \eta\, k_{\sigma^2}(P, Z^{[t]})\, k_{\sigma^2}(Z^{[t]}, Z^{[t]})^{-1}\left(g^{[t]}(Z^{[t]}) - \mu_\theta^{[t]}(Z^{[t]})\right) \alpha_\ell^{[t-1]}(P)^\top, \tag{141}$$

where η is a learning rate, and $g^{[t-\delta]}(P)$ is estimated with the MAP estimator $\hat{g}^{[t-\delta]}(P)$ in (130).

5.5 Dynamic Boltzmann machines with hidden units[11]

In this section, we study a DyBM with hidden units (see Figure 10). Each layer of this DyBM corresponds to a time $t-\delta$ for $0 \le \delta < \infty$ and has two parts: visible and hidden. The visible part $\mathbf{x}^{[t-\delta]}$ at the δ-th layer represents the values of the time-series at time $t-\delta$. The hidden part $\mathbf{h}^{[t-\delta]}$ represents the values of hidden units at time $t-\delta$. Here, units within each layer do not have connections to each other. We let $\mathbf{x}^{[<t]} \equiv (\mathbf{x}^{[s]})_{s<t}$ and define $\mathbf{h}^{[<t]}$ analogously.

The Boltzmann machine in Figure 10 has bias parameter $\mathbf{b}$ and weight parameter ($\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, $\mathbf{Z}$). Let $\theta \equiv (\mathbf{V}, \mathbf{W}, \mathbf{b})$ be the parameters connected to visible units $\mathbf{x}^{[t]}$ (from the units in the past, $\mathbf{x}^{[s]}$ or $\mathbf{h}^{[s]}$ for $s < t$) and $\phi \equiv (\mathbf{U}, \mathbf{Z})$. The conditional energy of this Boltzmann machine is given as follows:

$$E_{\theta,\phi}(\mathbf{x}^{[t]}, \mathbf{h}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) = E_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) + E_\phi(\mathbf{h}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}), \tag{142}$$

[11] This section closely follows [30].

where we define

$$E_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) = -\mathbf{b}^\top \mathbf{x}^{[t]} - \sum_{\delta=1}^{\infty} (\mathbf{x}^{[t-\delta]})^\top \mathbf{W}^{[\delta]}\, \mathbf{x}^{[t]} - \sum_{\delta=1}^{\infty} (\mathbf{h}^{[t-\delta]})^\top \mathbf{V}^{[\delta]}\, \mathbf{x}^{[t]} \tag{143}$$

$$E_\theta(\mathbf{h}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) = -\mathbf{b}^\top \mathbf{h}^{[t]} - \sum_{\delta=1}^{\infty} (\mathbf{x}^{[t-\delta]})^\top \mathbf{U}^{[\delta]}\, \mathbf{h}^{[t]} - \sum_{\delta=1}^{\infty} (\mathbf{h}^{[t-\delta]})^\top \mathbf{Z}^{[\delta]}\, \mathbf{h}^{[t]}. \tag{144}$$

and assume the following parametric form for $\delta \geq d$:

$$\mathbf{W}^{[\delta]} = \lambda^{\delta-d}\, \mathbf{W}^{[d]} \tag{145}$$

$$\mathbf{V}^{[\delta]} = \lambda^{\delta-d}\, \mathbf{V}^{[d]} \tag{146}$$

$$\mathbf{Z}^{[\delta]} = \lambda^{\delta-d}\, \mathbf{Z}^{[d]} \tag{147}$$

$$\mathbf{U}^{[\delta]} = \lambda^{\delta-d}\, \mathbf{U}^{[d]}, \tag{148}$$

where λ is a decay rate satisfying $0 \leq \lambda < 1$. Then the conditional energy can be represented as follows:

$$\begin{aligned} & E_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) \\ & = -\mathbf{b}^\top \mathbf{x}^{[t]} - \sum_{\delta=1}^{d-1} (\mathbf{x}^{[t-\delta]})^\top \mathbf{W}^{[\delta]}\, \mathbf{x}^{[t]} - \sum_{\delta=1}^{d-1} (\mathbf{h}^{[t-\delta]})^\top \mathbf{V}^{[\delta]}\, \mathbf{x}^{[t]} - (\boldsymbol{\alpha}^{[t-1]})^\top \mathbf{W}^{[d]}\, \mathbf{x}^{[t]} - (\boldsymbol{\beta}^{[t-1]})^\top \mathbf{V}^{[d]}\, \mathbf{x}^{[t]}, \end{aligned} \tag{149}$$

$$\begin{aligned} & E_\phi(\mathbf{h}^{[s]} \mid \mathbf{x}^{[<s]}, \mathbf{h}^{[<s]}) \\ & = -\sum_{\delta=1}^{d-1} (\mathbf{x}^{[s-\delta]})^\top \mathbf{U}^{[\delta]}\, \mathbf{h}^{[s]} - \sum_{\delta=1}^{d-1} (\mathbf{h}^{[s-\delta]})^\top \mathbf{Z}^{[\delta]}\, \mathbf{h}^{[s]} - (\boldsymbol{\alpha}^{[s-1]})^\top\, \mathbf{U}^{[d]} \mathbf{h}^{[s]} - (\boldsymbol{\beta}^{[s-1]})^\top\, \mathbf{Z}^{[d]} \mathbf{h}^{[s]}. \end{aligned} \tag{150}$$

where $\boldsymbol{\alpha}^{[t-1]}$ corresponds to the eligibility trace in the DyBM in (68), and we define an eligibility trace $\boldsymbol{\beta}^{[t-1]}$ for the hidden part analogously:

$$\boldsymbol{\alpha}^{[t-1]} \equiv \sum_{\delta=d}^{\infty} \lambda^{\delta-d}\, \mathbf{x}^{[t-\delta]} \tag{151}$$

$$\boldsymbol{\beta}^{[t-1]} \equiv \sum_{\delta=d}^{\infty} \lambda^{\delta-d}\, \mathbf{h}^{[t-\delta]}. \tag{152}$$

The energy in (149)-(150) gives the conditional probability distribution over $\mathbf{x}^{[t]}$ and $\mathbf{h}^{[t]}$ given $\mathbf{x}^{[<t]}$ and $\mathbf{h}^{[<t]}$. Specifically, we have

$$\mathbb{P}_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) = \frac{\exp(-E_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}))}{\sum_{\tilde{\mathbf{x}}^{[t]}} \exp(-E_\theta(\tilde{\mathbf{x}}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}))} \tag{153}$$

$$\mathbb{P}_\phi(\mathbf{h}^{[s]} \mid \mathbf{x}^{[<s]}, \mathbf{h}^{[<s]}) = \frac{\exp(-E_\phi(\mathbf{h}^{[s]} \mid \mathbf{x}^{[<s]}, \mathbf{h}^{[<s]}))}{\sum_{\tilde{\mathbf{h}}^{[t]}} \exp(-E_\phi(\tilde{\mathbf{h}}^{[s]} \mid \mathbf{x}^{[<s]}, \mathbf{h}^{[<s]}))} \tag{154}$$

for any binary vectors $\mathbf{x}^{[t]}$ and $\mathbf{h}^{[t]}$.

5.5.1 Learning a dynamic Boltzmann machine with hidden units

The DyBM with hidden units gives the probability of a time-series, $\mathbf{x} \equiv (\mathbf{x}^{[t]})_{t=\ell,\ldots,u}$, by

$$\mathbb{P}_{\theta,\phi}(\mathbf{x}) = \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\phi(\tilde{\mathbf{h}} \mid \mathbf{x}) \prod_{t=\ell}^{u} \mathbb{P}_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \tilde{\mathbf{h}}^{[<t]}) \tag{155}$$

where $\sum_{\tilde{\mathbf{h}}}$ denotes the summation over all of the possible values of hidden units from time $t = \ell$ to $t = u$, and

$$\mathbb{P}_\phi(\tilde{\mathbf{h}} \mid \mathbf{x}) \equiv \prod_{s=\ell}^{u} \mathbb{P}_\phi(\tilde{\mathbf{h}}^{[s]} \mid \mathbf{x}^{[<s]}, \tilde{\mathbf{h}}^{[<s]}), \tag{156}$$

where we arbitrarily define $\mathbf{x}^{[s]} = \mathbf{0}$ and $\tilde{\mathbf{h}}^{[s]} = \mathbf{0}$ for $s < \ell$.

We seek to maximize the log likelihood of a given $\mathbf{x}$ by maximizing a lower bound given by Jensen's inequality:

$$\begin{aligned}
\log \mathbb{P}_{\theta,\phi}(\mathbf{x}) &= \log \Big(\sum_{\tilde{\mathbf{h}}} \mathbb{P}_\phi(\tilde{\mathbf{h}} \mid \mathbf{x}) \prod_{t=\ell}^{u} \mathbb{P}_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \tilde{\mathbf{h}}^{[<t]}) \Big) && (157) \\
&\ge \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\phi(\tilde{\mathbf{h}} \mid \mathbf{x}) \log \Big(\prod_{t=\ell}^{u} \mathbb{P}_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \tilde{\mathbf{h}}^{[<t]}) \Big) && (158) \\
&= \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\phi(\tilde{\mathbf{h}} \mid \mathbf{x}) \sum_{t=\ell}^{u} \log \mathbb{P}_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \tilde{\mathbf{h}}^{[<t]}) && (159) \\
&= \sum_{t=\ell}^{u} \sum_{\tilde{\mathbf{h}}^{[<t]}} \mathbb{P}_\phi(\tilde{\mathbf{h}}^{[<t]} \mid \mathbf{x}^{[<t-1]}) \log \mathbb{P}_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \tilde{\mathbf{h}}^{[<t]}) \\
&\equiv L_{\theta,\phi}(\mathbf{x}), && (160)
\end{aligned}$$

where the summation with respect to $\tilde{\mathbf{h}}^{[<t]}$ is over all of the possible values of $\tilde{\mathbf{h}}^{[s]}$ for $s \le t-1$, and

$$\mathbb{P}_\phi(\tilde{\mathbf{h}}^{[<t]} \mid \mathbf{x}^{[<t-1]}) \equiv \prod_{s=\ell}^{t-1} \mathbb{P}_\phi(\tilde{\mathbf{h}}^{[s]} \mid \mathbf{x}^{[<s]}, \tilde{\mathbf{h}}^{[<s]}). \tag{161}$$

Learning weight to visible units The gradient of the lower bound with respect to θ is then given by

$$\nabla_\theta L_{\theta,\phi}(\mathbf{x}) = \sum_{t=\ell}^{u} \sum_{\tilde{\mathbf{h}}^{[<t]}} \mathbb{P}_\phi(\tilde{\mathbf{h}}^{[<t]} \mid \mathbf{x}^{[<t-1]}) \nabla_\theta \log \mathbb{P}_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \tilde{\mathbf{h}}^{[<t]}). \tag{162}$$

The right-hand side of (162) is a summation of expected gradients, which suggests a method of stochastic gradient. Namely, at each step t, we sample $\boldsymbol{H}^{[t-1]}(\omega)$ according to $\mathbb{P}_\phi(\mathbf{h}^{[t-1]} \mid \mathbf{x}^{[<t-1]}, \mathbf{h}^{[<t-1]})$ and update θ on the basis of

$$\nabla_\theta \log \mathbb{P}_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)). \tag{163}$$

This learning rule is equivalent to the one for the model where all of the units are visible, except that the values for the hidden units are given by sampled values. Therefore, the learning rule for θ

follows directly from Section 5.1.2:

$$\mathbf{b} \leftarrow \mathbf{b} + \eta \left(\mathbf{x}^{[t]} - \mathbb{E}_\theta\big[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)\big] \right) \tag{164}$$

$$\mathbf{W}^{[d]} \leftarrow \mathbf{W}^{[d]} + \eta\, \boldsymbol{\alpha}^{[t-1]} \left(\mathbf{x}^{[t]} - \mathbb{E}_\theta\big[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)\big] \right)^\top \tag{165}$$

$$\mathbf{V}^{[d]} \leftarrow \mathbf{V}^{[d]} + \eta\, \boldsymbol{\beta}^{[t-1]}(\omega) \left(\mathbf{x}^{[t]} - \mathbb{E}_\theta\big[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)\big] \right)^\top \tag{166}$$

$$\mathbf{W}^{[\delta]} \leftarrow \mathbf{W}^{[\delta]} + \eta\, \mathbf{x}^{[t-\delta]} \left(\mathbf{x}^{[t]} - \mathbb{E}_\theta\big[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)\big] \right)^\top \tag{167}$$

$$\mathbf{V}^{[\delta]} \leftarrow \mathbf{V}^{[\delta]} + \eta\, \boldsymbol{H}^{[t-\delta]}(\omega) \left(\mathbf{x}^{[t]} - \mathbb{E}_\theta\big[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)\big] \right)^\top \tag{168}$$

for $1 \leq \delta < d$, where $\mathbb{E}_\theta[\boldsymbol{X}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)]$ denotes the conditional expectation with respect to $\mathbb{P}_\theta(\cdot \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega))$, and we make explicit that $\boldsymbol{\beta}^{[s-1]}$ is computed with sampled hidden values:

$$\boldsymbol{\beta}^{[s-1]}(\omega) = \sum_{\delta=d}^{\infty} \lambda^{\delta-d}\, \boldsymbol{H}^{[s-\delta]}(\omega). \tag{169}$$

Learning weight to hidden units Now we take the gradient of $L_{\theta,\phi}(\mathbf{x})$ with respect to ϕ:

$$\nabla_\phi L_{\theta,\phi}(\mathbf{x}) = \sum_{t=\ell}^{u} \sum_{\tilde{\mathbf{h}}^{[<t]}} \nabla_\phi p_\phi(\tilde{\mathbf{h}}^{[<t]} \mid \mathbf{x}^{[<t-1]})\, \log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \tilde{\mathbf{h}}^{[<t]}), \tag{170}$$

where

$$\begin{aligned}
\nabla_\phi p_\phi(\tilde{\mathbf{h}}^{[<t]} \mid \mathbf{x}^{[<t-1]}) &= \nabla_\phi \prod_{s=\ell}^{t-1} p_\phi(\tilde{\mathbf{h}}^{[s]} \mid \mathbf{x}^{[<s]}, \tilde{\mathbf{h}}^{[<s]}) && (171)\\
&= \sum_{s=\ell}^{t-1} \nabla_\phi \log p_\phi(\tilde{\mathbf{h}}^{[s]} \mid \mathbf{x}^{[<s]}, \tilde{\mathbf{h}}^{[<s]}) \prod_{s'=\ell}^{t-1} p_\phi(\tilde{\mathbf{h}}^{[s']} \mid \mathbf{x}^{[<s']}, \tilde{\mathbf{h}}^{[<s']}) \\
&= p_\phi(\tilde{\mathbf{h}}^{[<t]} \mid \mathbf{x}^{[<t-1]}) \sum_{s=\ell}^{t-1} \nabla_\phi \log p_\phi(\tilde{\mathbf{h}}^{[s]} \mid \mathbf{x}^{[<s]}, \tilde{\mathbf{h}}^{[<s]}). && (172)
\end{aligned}$$

Plugging (172) into the right-hand side of (170), we obtain

$$\nabla_\phi L_{\theta,\phi}(\mathbf{x}) = \sum_{t=\ell}^{u} \sum_{\tilde{\mathbf{h}}^{[<t]}} p_\phi(\tilde{\mathbf{h}}^{[<t]} \mid \mathbf{x}^{[<t-1]})\, \log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \tilde{\mathbf{h}}^{[<t]}) \sum_{s=\ell}^{t-1} \nabla_\phi \log p_\phi(\tilde{\mathbf{h}}^{[s]} \mid \mathbf{x}^{[<s]}, \tilde{\mathbf{h}}^{[<s]}). \tag{173}$$

Similar to (162), the expression of (173) suggests a method of stochastic gradient: at each time t, we sample $\boldsymbol{H}^{[t-1]}(\omega)$ according to $p_\phi(\mathbf{h}^{[t-1]} \mid \mathbf{x}^{[<t-1]})$ and update ϕ on the basis of the following stochastic gradient:

$$\log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega))\, G_{t-1}, \tag{174}$$

where

$$G_{t-1} \equiv \sum_{s=\ell}^{t-1} \nabla_\phi \log p_\phi(\boldsymbol{H}^{[s]}(\omega) \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)). \tag{175}$$

The learning rules for $\mathbf{U}$ and $\mathbf{Z}$ are derived from (174)-(175) as follows:

$$\mathbf{U}^{[d]} \leftarrow \mathbf{U}^{[d]} + \eta \, \log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) \sum_{s=\ell}^{t-1} \boldsymbol{\alpha}^{[s-1]} \Big(\boldsymbol{H}^{[s]}(\omega) - \mathbb{E}_\phi\big[\mathbf{H}^{[\mathbf{s}]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)\big] \Big)^\top \tag{176}$$

$$\mathbf{Z}^{[d]} \leftarrow \mathbf{Z}^{[d]} + \eta \, \log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) \sum_{s=\ell}^{t-1} \boldsymbol{\beta}^{[s-1]}(\omega) \Big(\boldsymbol{H}^{[s]}(\omega) - \mathbb{E}_\phi\big[\boldsymbol{H}^{[s]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)\big] \Big)^\top \tag{177}$$

$$\mathbf{U}^{[\delta]} \leftarrow \mathbf{U}^{[\delta]} + \eta \, \log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) \sum_{s=\ell}^{t-1} \mathbf{x}^{[s-\delta]} \Big(\boldsymbol{H}^{[s]}(\omega) - \mathbb{E}_\phi\big[\mathbf{H}^{[\mathbf{s}]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)\big] \Big)^\top \tag{178}$$

$$\mathbf{Z}^{[\delta]} \leftarrow \mathbf{Z}^{[\delta]} + \eta \, \log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \mathbf{h}^{[<t]}) \sum_{s=\ell}^{t-1} \boldsymbol{H}^{[s-\delta]}(\omega) \Big(\boldsymbol{H}^{[s]}(\omega) - \mathbb{E}_\phi\big[\boldsymbol{H}^{[s]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)\big] \Big)^\top \tag{179}$$

for $1 \le \delta < d$, where $\mathbb{E}_\phi[\boldsymbol{H}^{[s]}(\omega) \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)]$ denotes the conditional expectation with respect to $\mathbb{P}_\phi(\cdot \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega))$.

Computation of (173) involves mainly two interrelated inefficiencies. First, although (173) can be approximately computed using sampled hidden values $\boldsymbol{H}^{[<t]}(\omega)$ in the same way as (162), the samples cannot be reused after updating ϕ because it was sampled from the distribution with the previous parameter. Second, since each summand of G_{t-1} depends on ϕ, G_{t-1} also has to be recomputed after each update. Thus, the computational complexity of (174) grows linearly with respect to the length of the time-series (*i.e.*, $t - \ell$), in contrast to (163), whose complexity is independent of that length.

Observe in (174) that $\nabla_\phi L_{\theta,\phi}(\mathbf{x})$ consists of the products of $\log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega))$ and $\nabla_\phi \log p_\phi(\boldsymbol{H}^{[s]}(\omega) \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega))$ for $s < t$. Without the dependency on $\log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega))$, the parameter ϕ is updated in a way that $\boldsymbol{H}^{[s]}(\omega)$ is more likely to be generated (*i.e.*, the learning rule would be equivalent to that for visible units). Such an update rule is undesirable, because $\boldsymbol{H}^{[s]}(\omega)$ has been sampled and is not necessarily what we want to sample again. The dependency on $\log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega))$ suggests that ϕ is updated by a large amount if the sampled $\boldsymbol{H}^{[s]}(\omega)$ happens to make the future values, $\mathbf{x}^{[t]}$ for $t > s$, likely. Intuitively, weighting $\nabla_\phi \log p_\phi(\boldsymbol{H}^{[s]}(\omega) \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega))$ by $\log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega))$ for $t > s$ is inevitable, because whether the particular values of hidden units are good for the purpose of predicting future values will only be known after seeing future values.

Approximations One could approximately compute (175) recursively:

$$G_t \leftarrow \gamma \, G_{t-1} + (1-\gamma) \nabla_\phi \log p_\phi(\boldsymbol{H}^{[t]}(\omega) \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)), \tag{180}$$

where $\gamma \in [0, 1)$ is a discount factor. The recursive update rule with $\gamma < 1$ puts exponentially small weight γ^{t-s} on $\nabla_\phi \log p_\phi(\boldsymbol{H}^{[s]}(\omega) \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega))$ computed with an old value of ϕ (*i.e.*, $s \ll t$). This recursively computed G_t is related to the momentum in gradient descent [33].

In (176)-(179), the value of $\mathbb{E}_\phi[\boldsymbol{H}^{[s]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)]$ is computed with the latest values of ϕ. Let $\phi^{[t-1]}$ be the value of ϕ immediately before step t. With the recursive computation of (180), the

learning rules of (176)-(179) are approximated with the following learning rules:

$$\begin{aligned}\mathbf{U}^{[d]} \leftarrow \mathbf{U}^{[d]} + \eta\,(1-\gamma)\,\log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)) \\ \sum_{s=\ell}^{t-1} \gamma^{t-1-s}\,\boldsymbol{\alpha}^{[s-1]} \Big(\boldsymbol{H}^{[s]}(\omega) - \mathbb{E}_{\phi^{[s-1]}}\big[\boldsymbol{H}^{[s]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)\big]\Big)^\top \qquad (181)\end{aligned}$$

$$\begin{aligned}\mathbf{Z}^{[d]} \leftarrow \mathbf{Z}^{[d]} + \eta\,(1-\gamma)\,\log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)) \\ \sum_{s=\ell}^{t-1} \gamma^{t-1-s}\,\boldsymbol{\beta}^{[s-1]} \Big(\boldsymbol{H}^{[s]}(\omega) - \mathbb{E}_{\phi^{[s-1]}}\big[\boldsymbol{H}^{[s]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)\big]\Big)^\top \qquad (182)\end{aligned}$$

$$\begin{aligned}\mathbf{U}^{[\delta]} \leftarrow \mathbf{U}^{[\delta]} + \eta\,(1-\gamma)\,\log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)) \\ \sum_{s=\ell}^{t-1} \gamma^{t-1-s}\,\mathbf{x}^{[s-\delta]} \Big(\boldsymbol{H}^{[s]}(\omega) - \mathbb{E}_{\phi^{[s-1]}}\big[\boldsymbol{H}^{[s]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)\big]\Big)^\top \qquad (183)\end{aligned}$$

$$\begin{aligned}\mathbf{Z}^{[\delta]} \leftarrow \mathbf{Z}^{[\delta]} + \eta\,(1-\gamma)\,\log p_\theta(\mathbf{x}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)) \\ \sum_{s=\ell}^{t-1} \gamma^{t-1-s}\,\boldsymbol{H}^{[s-\delta]}(\omega) \Big(\boldsymbol{H}^{[s]}(\omega) - \mathbb{E}_{\phi^{[s-1]}}\big[\boldsymbol{H}^{[s]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)\big]\Big)^\top \qquad (184)\end{aligned}$$

for $1 \le \delta < d$, where $\boldsymbol{H}^{[s]}(\omega)$ is a sample according to $\mathbb{P}_{\phi^{[s-1]}}(\cdot \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega))$ for each s. In (181)-(181), the quantity such as

$$G'_{t-1} \equiv \sum_{s=\ell}^{t-1} \gamma^{t-1-s}\,\boldsymbol{\alpha}^{[s-1]} \Big(\boldsymbol{H}^{[s]}(\omega) - \mathbb{E}_{\phi^{[s-1]}}\big[\boldsymbol{H}^{[s]} \mid \mathbf{x}^{[<s]}, \boldsymbol{H}^{[<s]}(\omega)\big]\Big)^\top \qquad (185)$$

can be computed recursively as

$$G'_t \leftarrow \gamma\, G'_{t-1} + (1-\gamma)\,\boldsymbol{\alpha}^{[t-1]} \Big(\boldsymbol{H}^{[t]}(\omega) - \mathbb{E}_{\phi^{[s-1]}}\big[\boldsymbol{H}^{[t]} \mid \mathbf{x}^{[<t]}, \boldsymbol{H}^{[<t]}(\omega)\big]\Big)^\top. \qquad (186)$$

In [30], we present an alternative approach of learning the DyBM with hidden units in a bidirectional manner, where we consider a backward DyBM that shares the parameters of the (forward) DyBM. Our key observation is that the parameters that are difficult to learn in the forward DyBM are relatively easy to learn in the backward DyBM. By training both the forward DyBM and the backward DyBM, we can effectively learn the parameters of the forward DyBM.

6 Conclusion

We have reviewed Boltzmann machines for time-series modeling. Such Boltzmann machines can be used for prediction [8, 30, 17, 38, 7, 22] and anomaly detection based on observed time-series. They may be also used to generate time-series such as human motion [40, 39, 38], music [22], and movies.

The use of Boltzmann machines is only one approach to modeling and learning time-series. Popular time-series models include but not limited to recurrent neural networks [34], long short term memory [15], autoregressive models, and hidden Markov models. As we have seen some of the examples, the best time-series model for a particular application might be obtained by appropriately combining some of existing time-series models.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR1304, Japan.

References

[1] L. F. Abbott and S. B. Nelson. Synaptic plasticity: Taming the beast. *Nature Neuroscience*, 3:1178–1183, 2000.

[2] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169, 1985.

[3] S. Amari and H. Nagaoka. *Methods of Information Geometry.* Oxford University Press, 2000.

[4] Y. Bengio, T. Mesnard, A. Fischer, S. Zhang, and Y. Wu. STDP as presynaptic activity times rate of change of postsynaptic activity. arXiv:1509.05936v2, 2016.

[5] G. Bi and M. Poo. Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18:10464–10472, 1998.

[6] L. Bottou. Online learning and stochastic approximations. In D. Saad, editor, *On-Line Learning in Neural Networks*, chapter 2, pages 9–42. Cambridge University Press, 2009.

[7] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1159–1166, 2012.

[8] S. Dasgupta and T. Osogami. Nonlinear dynamic Boltzmann machines for time-series prediction. In *The 31st AAAI Conference on Artificial Intelligence (AAAI-17)*, January 2017.

[9] S. Dasgupta, T. Yoshizumi, and T. Osogami. Regularized dynamic Boltzmann machine with delay pruning for unsupervised learning of temporal sequences. In *Proceedings of the 23rd International Conference on Pattern Recognition*, 2016.

[10] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[11] D. O. Hebb. *The organization of behavior: A neuropsychological approach.* Wiley, 1949.

[12] G. E. Hinton and A. D. Brown. Spiking Boltzmann machines. In *Advances in Neural Information Processing Systems 12*, pages 122–128. November 1999.

[13] G. E. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006.

[14] G. E. Hinton and T. J. Sejnowski. Optimal perceptual inference. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 448–453, June 1983.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[16] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[17] H. Kajino. A functional dynamic Boltzmann machine. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 1987–1993, 2017.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations* (ICLR), arXiv:1412.6980, 2015.

[19] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Computer Science Department, University of Toronto, Toronto, Canada, 2009.

[20] A. Lazar, G. Pipa, and J. Triesch. SORN: A self-organizing recurrent neural network. *Frontiers in Computational Neurosci.*, 3:Article 23, 2009.

[21] H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer-Verlag Berlin Heidelberg, 2005.

[22] Q. Lyu, Z. Wu, and J. Zhu. Polyphonic music modelling with LSTM-RTRBM. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 991–994, October 2015.

[23] T. Marks and J. Movellan. Diffusion networks, products of experts, and factor analysis. In *Proceedings of the Third International Conference on Independent Component Analysis and Blind Source Separation*, 2001.

[24] R. Memisevic and G. E. Hinton. Unsupervised learning of image transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8, 2007.

[25] R. Mittelman, B. Kuipers, S. Savarese, and H. Lee. Structured recurrent temporal restricted Boltzmann machines. In *Proc. 31st Annual International Conference on Machine Learning (ICML 2014)*, pages 1647–1655, June 2014.

[26] B. Nessler, M. Pfeiffer, L. Buesing, and W. Maass. Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Computational Biology*, 9(4):e1003037.

[27] T. Osogami. Learning binary or real-valued time-series via spike-timing dependent plasticity. *CoRR*, abs/1612.04897 (presented at Computing with Spikes NIPS 2016 Workshop, Barcelona, Spain, December 2016), 2016.

[28] T. Osogami. Boltzmann machines and energy-based models. Technical Report RT0979, IBM Research - Tokyo, 2017.

[29] T. Osogami and S. Dasgupta. Learning the values of the hyperparameters of a dynamic Boltzmann machine. *IBM Journal of Research and Development*, 61(4/5):to appear, 2017.

[30] T. Osogami, H. Kajino, and T. Sekiyama. Bidirectional learning for time-series models with hidden units. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, pages 2711–2720, August 2017.

[31] T. Osogami and M. Otsuka. Learning dynamic Boltzmann machines with spike-timing dependent plasticity. Technical Report RT0967, IBM Research, 2015.

[32] T. Osogami and M. Otsuka. Seven neurons memorizing sequences of alphabetical images via spike-timing dependent plasticity. *Scientific Reports*, 5:14149, 2015.

[33] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks: The Official Journal of the International Neural Network Society*, 12(1):145–151, 1999.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations*, chapter 8. MIT Press, 1986.

[35] B. Scellier and Y. Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. arXiv:1602.05179v4, 2016.

[36] B. Schrauwen and L. Buesing. A hierarchy of recurrent networks for speech recognition. In *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[37] I. Sutskever and G. E. Hinton. Learning multilevel distributed representations for high-dimensional sequences. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)*, volume 2, pages 548–555. Journal of Machine Learning Research - Proceedings Track, 2007.

[38] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted Boltzmann machine. In *Advances in Neural Information Processing Systems 21*, pages 1601–1608. December 2008.

[39] G. W. Taylor and G. E. Hinton. Factored conditional restricted Boltzmann machines for modeling motion style. In *Proc. 26th Annual International Conference on Machine Learning (ICML 2009)*, pages 1025–1032, June 2009.

[40] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In P. B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1345–1352. MIT Press, 2007.

[41] T. Tieleman and G. E. Hinton. Lecture 6.5—Rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[42] G. G. Turrigiano and S. B. Nelson. Homeostatic plasticity in the developing nervous system. *Nature Rev. Neurosci.*, 5:97107, 2004.

[43] M. Welling, M. Rosen-Zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In *Advances in Neural Information Processing Systems 17*, pages 1481–1488. 2004.