Generalized Boltzmann Machine with Deep Neural Structure

Yingru Liu
Stony Brook University

Dongliang Xie
Beijing Univ. of Post and Telecoms.

Xin Wang
Stony Brook University

Abstract

Restricted Boltzmann Machine (RBM) is an essential component in many machine learning applications. As a probabilistic graphical model, RBM posits a shallow structure, which makes it less capable of modeling real-world applications. In this paper, to bridge the gap between RBM and artificial neural network, we propose an energy-based probabilistic model that is more flexible on modeling continuous data. By introducing the pairwise inverse autoregressive flow into RBM, we propose two generalized continuous RBMs which contain deep neural network structure to more flexibly track the practical data distribution while still keeping the inference tractable. In addition, we extend the generalized RBM structures into sequential setting to better model the stochastic process of time series. Performance improvements on probabilistic modeling and representation learning are demonstrated by the experiments on diverse datasets.

1 Introduction

As a special instance of undirected graphical model, Restricted Boltzmann Machine (RBM) [Hinton, 2012, Carlson et al., 2015, Ping and Ihler, 2017] is an essential component for many machine learning applications, such as representation learning [Bengio et al., 2013] and probabilistic modeling [Salakhutdinov and Murray, 2008a]. RBM has many appealing advantages to be a building block of deep learning schemes, including well-defined inference distribution, universal approximation property [Le Roux and Bengio, 2008] and concise structure. Many RBM-inspired models [Dahl et al., 2010, Courville et al., 2014, Guo et al., 2018] are proposed to track the probability distributions of various kinds of data. In this paper, we focus on RBMs for data with continuous values.

As one major limitation of RBM, it needs a shallow structure to make the unambiguous inference. It contains a visible layer and a hidden layer whose units are random variables. Specific energy function on Boltzmann distribution is given to define the dependency between visible and hidden layers. The parameters for the conditional distributions of the visible layer are linear or quadratic functions of the conditional random variables. The restriction to the use of a specific energy function prevents the application of RBM to modeling practical data with more sophisticated distribution.

In this paper, we propose two generalized variants of continuous RBMs which have significantly improved modeling capability, and we call them pairwise inverse autoregressive flow RBM (pIAF-RBM). Compared with traditional RBM-based models, pIAF-RBM consists of an undirected subgraph and a series of transformations. The transformation consists of two unidirectional structures, each is an independent neural network. This allows the mapping of a specific distribution to a more general one, which relaxes the distribution assumption often made by traditional RBM. Therefore, the distribution of pIAF-RBM is more general and can better model real-world data. The contributions of this paper are three-fold:

* We introduce deep neural structures to relax the distribution assumption of RBM based on Inverse Autoregressive Flow (IAF). The most challenging problem is that a neural network is difficult to be inverted and the lack of the tractable inverse function of IAF would damage the undirected structure of RBM. To solve this problem, we define another neural network to estimate the inverse function. The resulted IAF is bidirectional and we call this dual structure as pair-wise IAF (pIAF).

* Based on pIAF, we propose two generalized variants of continuous RBMs, each consisting of hierarchical

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

neural network structures to modulate the probabilistic models given by the energy functions. We call them pIAF-GRBM and pIAF-ssRBM.

* We further extend pIAF-GRBM and pIAF-ssRBM to work in the sequential setting by incorporating a recurrent neural network (RNN) to model the stochastic process of time series. The resulted pIAF-RNN-RBMs are then employed in stochastic modeling and unsupervised feature learning, which are still challenging problems in artificial intelligence field.

The rest of this paper is organized as follows. We revisit the mathematical background of RBMs in Section 2 and then propose our pIAF-RBMs in Section 3. In section 4, we propose pIAF-RNN-RBMs for time series. We present our experimental results in Section 5 and conclude the work in Section 6.

2 Background of Restricted Boltzmann Machine

RBM is a group of undirected graphical models containing bipartite graph structure to indicate dependencies between data observations and unobserved factors. It posits a joint distribution of the visible layer V and unobserved states X as $P_\varphi(V,X) = \exp(-E_\varphi(V,X))/Z_\varphi$, where φ denotes parameters and Z_φ is called partition function. The marginal distribution of observation is further defined as $P_\varphi(V) = \exp(-\mathcal{F}_\varphi(V))/Z_\varphi$, where $\mathcal{F}_\varphi(V)$ is the free-energy function. The visible variable V can be defined as either discrete or continuous random vectors. In our work, we concentrate on the continuous case. In this section, we revisit two types of RBMs for continuous data. One is Gaussian RBM (GRBM), the other is Spike-and-Slab RBM (ssRBM).

2.1 Gaussian RBM

Gaussian RBM (GRBM) is a specific kind of RBM for modeling the data with continuous values. It consists of a continuous visible layer V and a binary hidden layer $X = [H]$. The energy function of GRBM is defined as:

$$E_\varphi(V,H) = \frac{(V-b_v)^T(V-b_v)}{2\beta^2} - b_h^T H - H^T W \frac{V}{\beta^2},$$

where $\varphi = \{W, b_v, b_h, \beta\}$ are parameters. The division between the vector $V - b_v$ and the variance parameter β is element-wise. By applying Bayesian rules, the conditional distributions between V and H are given as $P_\varphi(V|H) = \mathcal{N}(W^T H + b_v, \beta^2 I)$ and $P_\varphi(H|V) = \mathcal{S}(WV + b_h)$, where $\mathcal{N}(\mu, \beta^2 I)$ denotes Gaussian distribution and $\mathcal{S}(\cdot)$ denotes the sigmoid function. It has been shown that GRBM is less efficient in learning the variance parameter β for the data vector. Hence, β is set as a constant and noise-free reconstruction is encouraged [Hinton, 2012].

2.2 Spike-and-Slab RBM

To better utilize the correlation information among each element of data vector, Spike-and-Slab RBM (ssRBM) is proposed in [Courville et al., 2014]. The hidden layer is composed of a continuous random vector S and a binary random vector H. The energy function of $\{V, S, H\}$ is given by:

$$\begin{aligned} E_\varphi(V,H,S) =& \frac{1}{2}V^T\Big(\sum_{H_i \in H} \Phi_i H_i + \Lambda\Big)V - b_h^T H_n \\ &+ \frac{1}{2}S^T \operatorname{diag}(\alpha) S - V^T W(S \odot H) \\ &+ \alpha^T \operatorname{diag}(\mu^2) H - S^T \operatorname{diag}(\alpha \odot \mu) H, \end{aligned}$$

and the corresponding conditional distributions are derived as:

$$\begin{aligned} &P_\varphi(V|H,S) = \mathcal{N}\Big(C_1 W(S \odot H_n), C_1\Big), \\ &P_\varphi(S|V,H) = \mathcal{N}\Big(\frac{(W^T V)^2}{2\alpha} \odot H + \mu \odot H, \operatorname{diag}(\alpha)^{-1}\Big), \\ &P_\varphi(V|H) = \mathcal{N}\Big(C_0 W(\mu \odot H), C_0\Big), \\ &P_\varphi(H|V) = \mathcal{S}\Big(\frac{(W^T V)^2}{2\alpha} + W^T V \odot \mu - \frac{V^T\{\Phi_i\}V}{2} + b_h\Big), \end{aligned}$$

where $\odot$ denotes element-wise multiplication, $V^T\{\Phi_i\}V = \{V^T\Phi_1 V, V^T\Phi_2 V, \dots\}$ and $\{C_0, C_1\}$ are corresponding covariances.

As shown above, the energy functions of GRBM and ssRBM are designed to obtain closed-form conditional distributions. The parameters in the conditional distributions of V are linear or quadratic functions of the conditional random variables. These closed-form distributions allow efficient sampling from RBM while they also limit its model capability. In this paper, we incorporate the expression power of neural networks into RBM through a special kind of normalizing flows [Rezende and Mohamed, 2015, Kingma et al., 2016]. The conditional distributions of our model are more flexible than classic RBM while the efficient sampling process is still retained.

3 Pairwise Inverse Autoregressive Flow RBMs

Our pairwise Inverse Autoregressive Flow RBMs (pIAF-RBMs) exploit the nonlinear expression capability of autoregressive neural networks to increase the flexibility of model distributions. The structures of pIAF-GRBM and pIAF-ssRBM are depicted in figure 1 (a). The undirected links among $\{H, \widetilde{V}\}$ and $\{H, S, \widetilde{V}\}$ represent their probabilistic dependencies and define the

RBM subgraphs of our models. Between the input V and the undirected subgraphs, a series of density transformations with neural network structures are applied to adjust the probability density of V.

The design of the neural network structure relies on the principle of Inverse Autoregressive Flow (IAF) [Kingma et al., 2016]. As the original IAF is not defined to have the closed-form inverse function which is needed in our models, we first propose the novel pair-wise IAF (pIAF) that contains the other redeeming autoregressive neural network to approximate the inverse of IAF. After that, we propose GRBM and ssRBM with the pIAF enhancement.

3.1 Pairwise Inverse Autoregressive Flow (pIAF)

Given a random vector $\widetilde{V}$ and an invertible mapping $V = F_v(\widetilde{V})$, the relation between the probability densities $P(\widetilde{V})$ and $P(V)$ are defined as $P(V) = P(\widetilde{V}) \cdot |\det \partial F_v^{-1}/\partial V|$. A normalizing flow [Rezende and Mohamed, 2015] describes the transformation of a probability density by defining a sequence of invertible mappings. Many kinds of normalizing flows have been studied, including Hamiltonian flow, planar flow, radial flow and inverse autoregressive flow [Kingma et al., 2016]. In our work, we employ the inverse autoregressive flow, as it embeds the power of hierarchical autoregressive neural network and it scales well to high-dimensional spaces.

In [Kingma et al., 2016], the inverse autoregressive flow is defined as a chain of invertible mapping steps which are designed by

$$\begin{aligned}&[m_t, \sigma_t] = \text{autoregressiveNN}(V_{t-1}; \theta_t)\\ &V_t = \sigma_t \odot V_{t-1} + (1-\sigma_t) \odot m_t,\ t = 1, 2, \ldots, \mathcal{T}.\end{aligned}$$

[Forward Flow]

where $[m_t, \sigma_t]$ is the output of an autoregressive neural network with multiple computational hidden layers and $\mathcal{T}$ is the total number of steps. θ_t denotes the model parameters. The probability density of the ending of transformation chain is given by:

$$\log P(V_T) = \log P(V_0) - \sum_{t=1}^{\mathcal{T}} \sum_{d=1}^{D} \log \sigma_{t,d},$$

where D is the dimension of the random vector V_0. Although inverse autoregressive flow is shown to be invertible with a simple expression of the determinant of differentials $\partial F_v^{-1}/\partial V$, the form of inverse function is not tractable. In the design of our models, the inverse function is essential to allow information transits through the undirected graphical model. Therefore, we propose a pair-wise inverse autoregressive flow (pIAF) that uses an auxiliary autoregressive neural network to approximate the inverse function of IAF, which are defined as:

$$\begin{aligned}&[\widehat{m}_t, \widehat{\sigma}_t] = \text{autoregressiveNN}_2(V_t; \phi_t),\\ &V_{t-1} = \frac{V_t - (1-\widehat{\sigma}_t) \odot \widehat{m}_t}{\widehat{\sigma}_t},\ t = \mathcal{T}, \mathcal{T}-1, \ldots, 1.\end{aligned}$$

[Backward Flow]

where ϕ_t denotes model parameters of the backward flow.

In this paper, the autoregressive NNs are implemented by MADE [Germain et al., 2015]. In the training process, one of these two autoregressive neural networks is optimized by maximum likelihood together with the undirected subgraph. The other is trained to recover the input of the first neural network by optimizing the cost function given by

$$\mathcal{L}_{\text{aux}}(\theta \text{ or } \phi) = ||\log \sigma_t - \log \widehat{\sigma}_t||_1 + ||m_t - \widehat{m}_t||_1, \tag{1}$$

where $||\cdot||_1$ is the L-1 loss. One step of pIAF is depicted in Figure 1 (b).

3.2 GRBM with pIAF (pIAF-GRBM)

With the definition of pIAF, we propose pIAF-GRBM to fully release the power of GRBM with a hierarchical neural network structure and a flexible distribution following the practical data. Consider the energy function of GRBM whose input is the output of normalizing flow of observation V:

$$\begin{aligned}E_\varphi(V, H) =& \frac{1}{2}(F_\phi(V) - b_v)^T(F_\phi(V) - b_v) - b_h^T H\\ &- H^T W F_\phi(V),\end{aligned}$$

where we set the variance parameter β to 1. $\widetilde{V} = F_\phi(V)$ denotes any kind of normalizing flow of V and $\phi = \{W, b_v, b_h\}$ denotes parameters of the undirected subgraph in pIAF-GRBM. Substituting $\widetilde{V}$ into the energy function, $E_\varphi(\widetilde{V}, H)$ is identical with the energy function of GRBM and we have the conditional distributions $P(\widetilde{V}|H) = \mathcal{N}(W^T H + b_v, \mathbb{I})$ and $P(H|\widetilde{V}) = \mathcal{S}(W\widetilde{V} + b_h)$. According to the properties of normalizing flows, we can further obtain the tractable conditional distributions between the visible layer V and binary latent state H as

$$\begin{aligned}&P_\varphi(V|H) = \mathcal{N}(W^T H + b_v, \mathbb{I}) \cdot \Big|\det \frac{\partial F_\phi}{\partial V}\Big|,\\ &P_\varphi(H|V) = \mathcal{S}(W F_\phi(V) + b_h),\end{aligned}$$

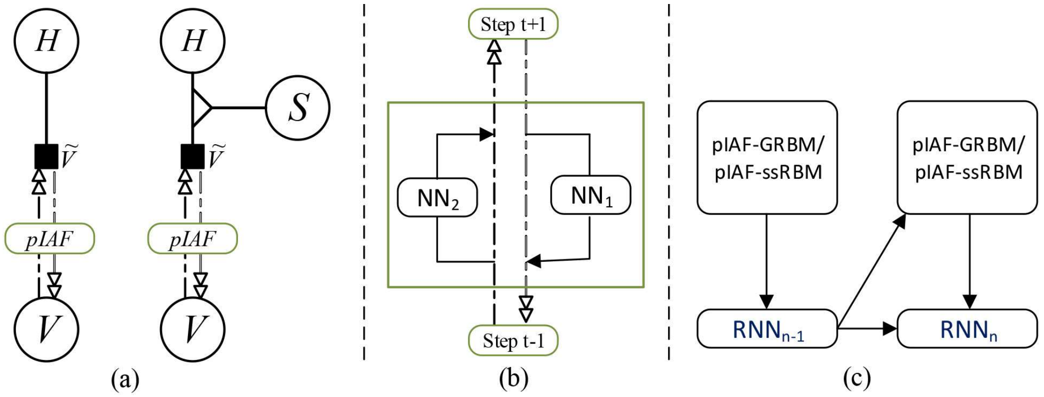

Figure 1: (a) Model Topologies of pIAF-GRBM and pIAF-ssRBM; (b) One step of pIAF; (c) Model topology of pIAF-RNN-RBM.

and the log-likelihood function as

$$\log P_\varphi(V) = -\frac{(F_\phi(V)-b_v)^T(F_\phi(V)-b_v)}{2} + \sum \text{softplus}\big(WF_\phi(V)+b_h\big) - \log Z_\varphi + \log\big|\det\frac{\partial F_\phi}{\partial V}\big|, \qquad (2)$$

where $\sum$ softplus$(\cdot)$ means the summation of elements of the vector output from the softplus function. F_ϕ modulates the density of V by the determinant of differentials, when keeping the conditional distributions tractable as traditional GRBM. This modulation relaxes the distribution assumption of V and hence increases the model flexibility. Recall that normalizing flow is a computational mapping instead of a sequential sampling, $F_\phi(V)$ is computationally feasible in current hardware architecture. In our implementation, we consider defining F_ϕ by the backward flow of pIAF proposed in subsection 3.1, and the corresponding model is called pIAF-GRBM. Therefore, the logarithm of determinant in the log-likelihood of pIAF-GRBM has a simple expression as

$$\log\big|\det\frac{\partial F_\phi}{\partial V}\big| = -\sum_{t=1}^{\mathcal{T}}\sum_{d=1}^{D}\log\widehat{\sigma}_{t,d},$$

where $\mathcal{T}$ is the number of steps in the flow and D is the dimension of input vector.

3.3 ssRBM with pIAF (pIAF-ssRBM)

Similar to pIAF-GRBM, pIAF can be applied to enhance the model flexibility of ssRBM. As ssRBM is efficient in capturing the correlation among elements of the observation vector, it is supposed that pIAF-ssRBM will be a more powerful model to disentangle the explanatory factors of complicated data.

The energy function of pIAF-ssRBM is defined as

$$E_\varphi(V,H,S) = \frac{1}{2}F_\phi(V)^T\Big(\sum_{H_i\in H}\Phi_i H_i + \Lambda\Big)F_\phi(V) - b_h^T H_n + \frac{1}{2}S^T\text{diag}(\alpha)S - F_\phi(V)^T W(S\odot H) + \alpha^T\text{diag}(\mu^2)H - S^T\text{diag}(\alpha\odot\mu)H,$$

where $F_\phi(V)$ is the backward flow of pIAF. We can further obtain the log-likelihood function of pIAF-ssRBM as

$$\log P_\varphi(V) = -\frac{1}{2}F_\phi(V)^T\Lambda F_\phi(V) + \frac{1}{2}\sum_{\alpha_i\in\alpha}\log 2\pi\alpha_i^{-1} - \log Z_\varphi - \sum_{t=1}^{T}\sum_{d=1}^{D}\log\widehat{\sigma}_{t,d} + \sum\text{softplus}\Big(\frac{(W^TF_\phi(V))^2}{2\alpha} + W^TF_\phi(V)\odot\mu - \frac{F_\phi(V)^T\{\Phi_i\}F_\phi(V)}{2} + b_h\Big), \qquad (3)$$

where $(\cdot)^2$ is element-wise. We further have the tractable conditional distributions of pIAF-ssRBM given as

$$P_\varphi(V|H,S) = \mathcal{N}\Big(C_1W(F_s(S)\odot H_n), C_1\Big)\cdot\prod_{t=1}^{T}\prod_{d=1}^{D}\sigma_{t,d}^{(v)},$$

$$P_\varphi(V|H) = \mathcal{N}\Big(C_0W(\mu\odot H), C_0\Big)\cdot\prod_{t=1}^{T}\prod_{d=1}^{D}\sigma_{t,d}^{(v)},$$

$$P_\varphi(H|V) = \mathcal{S}\Big(\frac{(W^TF_v(V))^2}{2\alpha} + W^TF_v(V)\odot\mu - \frac{F_v(V)^T\{\Phi_i\}F_v(V)}{2} + b_h\Big),$$

$$P_\varphi(S|V,H) = \prod_{t=1}^{T}\prod_{d=1}^{D_s}\sigma_{t,d}^{(s)}\cdot\mathcal{N}\Big(\frac{(W^TF_v(V))^2}{2\alpha}\odot H + \mu\odot H, \text{diag}(\alpha)^{-1}\Big),$$

With all the tractable distributions, the sampling process of our model is simple. Gibbs sampling of the undirected subgraph can be applied directly to obtain $\widetilde{V}$ and V is further generated by $F_{\phi}^{-1}(\widetilde{V}) \approx F_{\theta}(\widetilde{V})$.

3.4 Training of pIAF-RBMs

The training of our pIAF-RBM models consists of two steps. In the first step, we update the parameters of RBM and the forward path of pIAF by optimizing the log-likelihood functions given in Eq. (2) or Eq. (3). In the second step, we update the parameters of the backward path of pIAF by optimizing the auxiliary loss $\mathcal{L}_{\text{aux}}(\phi)$. As finding the gradient of $\log Z_{\phi}$ can be intractable and requires approximation, we exploit Contrastive Divergence (CD) [Hinton, 2012] and Persistent Contrastive Divergence (PCD) [Tieleman, 2008] to estimate $\nabla \log Z_{\phi}$ with Gibbs Sampling. The whole training process is given in Algorithm 1.

Algorithm 1 Training Process of pIAF-RBMs

```
Input: dataset $\{v\}$, steps $K$ of Gibbs Sampling
if use PCD then
    Initialize a random sample $v_0$
end if
while training do
    Select a data $v$ from $\{v\}$.
    if use PCD then
        Sample $v_0$ by Gibbs Sampling starting at $v$
    else if use CD then
        Sample $v_0$ by Gibbs Sampling from previous $v_0$
    end if
    Approximate $\nabla \log P_{\varphi}(V)$ by $v$ and $v_0$
    Update $\varphi \leftarrow \text{Optimizer}(\nabla \log P_{\varphi}(V))$
    Update $\{\phi_t\} \leftarrow \text{Optimizer}(\nabla \log P_{\varphi}(V))$
    Compute $\nabla \mathcal{L}_{\text{aux}}(\theta_t)$ for $\{\theta_t\}$
    Update $\{\theta_t\} \leftarrow \text{Optimizer}(\nabla \mathcal{L}_{\text{aux}}(\theta_t))$
end while
```

4 Temporal RBMs with pIAF for Time Series

RBM is originally an important tool in learning representative feature for data without temporal dependency. As pIAF-GRBM and pIAF-ssRBM are more generalized energy-based models than traditional RBMs, we intend to extend pIAF-GRBM and pIAF-ssRBM into sequential setting for stochastic modeling as well as representation learning on time series.

The topology of our proposed sequential model called pIAF-RNN-RBM is depicted in Figure 1 (c). Consider a sequence of N data vectors denoted as $V_{1:N}$, we first introduce a recurrent neural network (RNN) to compute the temporal transition. For n-th data vector, the transition is computed as

$$[d_n, \text{state}_n] = \text{RNN}(V_n, \text{state}_{n-1}),$$

where d_n denotes the output of the recurrent layer at the n_{th} time slot and state_n denotes the corresponding state of RNN. Therefore, the log-likelihood of the observed sequence can be decomposed as

$$\begin{aligned} \log P(V_{1:N}) &= \sum_{n=1}^{N} \log P(V_n | V_{1:n-1}) \\ &= \sum_{n=1}^{N} \log P(V_n | d_{n-1}), \end{aligned}$$

where d_0 is set as zero. After that, we incorporate pIAF-GRBM and pIAF-ssRBM into the framework to estimate $P(V_n|d_{n-1})$. Specific interface should be defined for pIAF-GRBM and pIAF-ssRBM so that the temporal transition d_{n-1} is able to modulate parameters to capture the evolution of data distribution $P(V_n|d_{n-1})$. In our design, the interface is defined in the bias terms of GRBM/ssRBM subgraph. The revision is given in Table 1. $\{b_h^{(n)}, b_v^{(n)}\}$ denote the time-variant bias parameters of GRBM and ssRBM.

Table 1: Time-variant bias for each part of pIAF-RNN-RBM.

GRBM	ssRBM
$b_h^{(n)} = U_1 d_{n-1} + b_h$	$b_h^{(n)} = U_1 d_{n-1} + b_h$
$b_v^{(n)} = U_2 d_{n-1} + b_v$	$-$

Beyond the probabilistic modeling, we are also interested in unsupervised representation learning for time series. While RBM-inspired methods have been widely used in images to extract appealing features, representation learning on time series is a problem that is important but challenging. when pIAF-RNN-GRBM and pIAF-RNN-ssRBM is supposed to have better probabilistic modeling than its counterpart RNN-RBM [Boulanger-Lewandowski et al., 2012], the hidden activations $\{H_{1:N}\}$ and $\{H_{1:N}, S_{1:N}\}$ can be regarded as a feature that embeds the information among data sequence. Therefore, our models can be applied for finding a low-dimensional embedding that preserves the properties of original data, when the dimension of latent states is set smaller than the dimension of input.

5 Experiments

In this section, we evaluate our methods on diverse datasets to demonstrate their performance and empirically analyze the characteristics of our models. Two major applications of RBMs are considered. One is probabilistic modeling; the other is representation learning. In the experiments, we concentrate on the model structure comparison between RBMs and pIAF-RBMs. The code will be available on Github.

5.1 Probabilistic Modeling

We first evaluate the model capability to estimate the probability density of high-dimensional data. Four image datasets are considered in this experiments, including MNIST [Lecun et al., 1998], SVHN [Netzer et al., 2011],CIFAR-10 [Krizhevsky, 2009] and CIFAR-100 [Krizhevsky, 2009]. To better understand the proposed models with full-connected neural structure, the color images are transfered into grayscale and MADE [Germain et al., 2015] is used to implement pIAF. For color images, one can easily define an convolutional structure in pIAF by replacing MADE with PixelCNN [van den Oord et al., 2016].

Table 2: Test log-likelihood (bits/dim) on image datasets

	MNIST	SVHN
CD-1		
GRBM	-1.4585 ± 0.0345	-1.4825 ± 0.0175
pIAF-GRBM	0.4131 ± 0.2738	-0.4265 ± 0.0228
ssRBM	0.6661 ± 0.0226	0.9449 ± 0.0912
pIAF-ssRBM	1.7740 ± 0.0215	1.4990 ± 0.0627
PCD-1		
GRBM	-2.0140 ± 0.0610	-1.6288 ± 0.0248
pIAF-GRBM	1.8517 ± 0.0259	0.2586 ± 0.0318
ssRBM	0.0833 ± 0.0482	-0.9512 ± 0.6582
pIAF-ssRBM	2.0542 ± 0.1065	1.0873 ± 0.0036
	CIFAR-10	CIFAR-100
CD-1		
GRBM	-1.7911 ± 0.0062	-1.6790 ± 0.0047
pIAF-GRBM	-0.3927 ± 0.0200	-0.4891 ± 0.0319
ssRBM	0.6793 ± 0.0852	0.5934 ± 0.0817
pIAF-ssRBM	1.8750 ± 0.0133	1.8658 ± 0.0503
PCD-1		
GRBM	-1.8164 ± 0.0149	-1.6704 ± 0.0113
pIAF-GRBM	-0.0563 ± 0.3493	-0.3030 ± 0.0439
ssRBM	-0.4441 ± 0.0498	-1.9285 ± 1.9108
pIAF-ssRBM	0.9887 ± 0.0600	1.1162 ± 0.0419

The experiment settings are given as follows: the size of stochastic hidden layer is 100 for MNIST and 200 for the rest. For pIAF-GRBM and pIAF-ssRBM, one step of pIAF is constructed by two MADEs with three Relu layers. The size of Relu layer is 512 for MNIST and 1024 for the others. The batch size is set as 125. Two training methods are considered to approximate the intractable gradient. One is Contrastive Divergence (CD); the other is Persistent Contrastive Divergence (PCD). The step of Gibbs sampling required by CD and PCD is set as 1. All the models are trained until convergence. Each case is run 5 times and we report the mean and standard deviation of the test log-likelihood (LL) in Table 2. Anneal Importance Sampling (AIS) [Salakhutdinov and Murray, 2008b] is applied to estimate the partition function when computing LL.

Log-likelihood: According to Table 2, under the same training method, RBMs with pIAF structures always significantly outperform RBMs without pIAF, which demonstrates the effectiveness of applying pIAF to more accurately track the practical data distribution. The standard deviation of LL is smaller in most models trained by CD-1. This is because the Gibbs sampling chain of CD-1 starts at the observed data and therefore leads to a less-variant gradient estimation. pIAF-GRBM and pIAF-ssRBM trained by PCD-1 have better average performance than these trained by CD-1. However, the variances are very large in some cases. This is due to the slow mixture rate of the persistent sampling chain used in PCD-1 that is unable to chase the updates of RBMs.

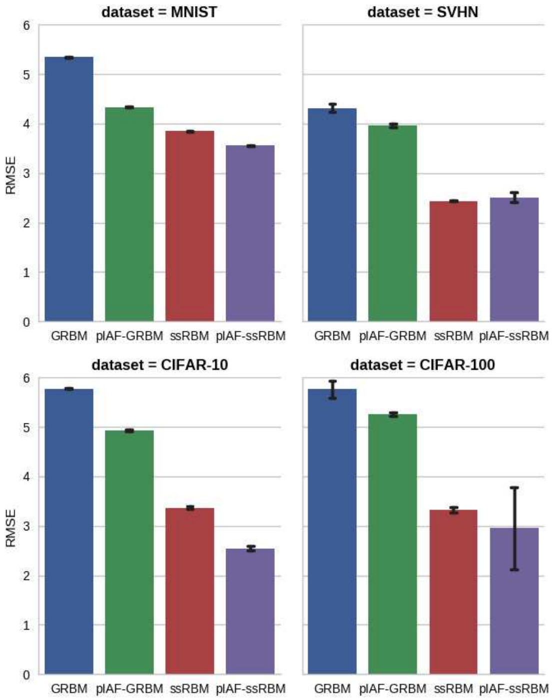

Figure 2: Reconstruction error given by RMSE on four image datasets

Reconstruction Error: In our design of pIAF, we construct a forward flow by incorporating an auxiliary neural network to inverse the complicated equation used to represent the backward flow. The neural networks in backward flow are trained with RBM parameters by maximizing the likelihood, and the neural networks in forward flow are trained by minimizing the auxiliary cost $\mathcal{L}_{\text{aux}}$. Although this could lead to an additional deviation when reconstructing the input from the hidden layer, the auxiliary neural networks are easy to train thanks to the efficient dual structure of pIAF. Therefore, the reconstruction errors of pIAF-RBMs are smaller than RBMs without pIAF, which is demonstrated in Figure 2.

5.2 Representation Learning

After showing the performance improvement on probabilistic modeling, we further evaluate our models on representation learning. We use Anuran Calls dataset from UCI Repository [Dheeru and Karra Taniskidou, 2017], where the input data is 22-dimensional vector and the task is to compress the information of input into 5-dimensional feature vector. For GRBM and pIAF-GRBM, the feature is given by $P(H|V)$. For ssRBM and pIAF-ssRBM, the feature is given by the mean of $(H \odot S)$. The feature vector is further sent to a linear SVM for classification. When the dimension of the feature vector is smaller than raw data, the representations generated by GRBM, ssRBM and pIAF-GRBM suffer from the loss of information and the classification performance gets worse. RBMs with pIAF structure have better performance than those without pIAF. Classification based on pIAF-ssRBM is even more accurate than using raw data.

Table 3: Classfication accuracy on Anuran Dataset

	Families	Genus
Raw Data	$92.36 \pm \text{NA}$	$91.10 \pm \text{NA}$
GRBM	57.96 ± 0.00	54.83 ± 0.00
ssRBM	90.55 ± 4.28	88.13 ± 3.92
pIAF-GRBM	75.05 ± 3.03	72.15 ± 2.43
pIAF-ssRBM	$\mathbf{96.48 \pm 0.10}$	$\mathbf{93.66 \pm 0.08}$
	Species	
Raw Data	$91.66 \pm \text{NA}$	
GRBM	45.75 ± 0.06	
ssRBM	87.02 ± 4.13	
pIAF-GRBM	71.26 ± 3.03	
pIAF-ssRBM	$\mathbf{93.30 \pm 0.44}$	

5.3 Sequential Models for Time series

After evaluating the performance of pIAF-RBMs, we examine the performance on stochastic modeling and representation learning for time series in two datasets: CMU motion capture dataset* and GTZAN music genres dataset †. Each dataset is randomly separated into training set and testing set. The ratio between training and testing sets is 0.8/0.2. The data preprocessing of CMU Motion Capture dataset and GTZAN music genres datasets is given as follows:

* **CMU Motion:** The CMU motion caption dataset consists of over 2500 long records of human motions with at least 1500 frames. In each frame, human activity is represented as a 62-dimensional vector. The long records are segmented by 250 frames and then each dimension of the frames are normalized by global mean and standard deviation.

* **GTZAN:** The GTZAN music genres dataset consists of 1000 songs that evenly belong to 10 different genres. Each audio is transformed into MFCC matrix and scaled between 0 to 1.

The baseline model is RNN-RBM proposed in [Boulanger-Lewandowski et al., 2012]. For Motion Capture, all models contain a 200-dimensional latent state and the RNN component is implemented by a GRU layer with 500 units. pIAF is constructed by 3-layer MADEs. Each layer in MADEs has 250 Relu units. For GTZAN, we are interested in learning low-dimensional embedding of MFCC matrix. The latent state is 5-dimensional. As the account of data is relatively small, RNN component is applied by 100 GRU units and the number of Relu units in MADEs is set to 50.

Table 4: Reconsruction errors of on sequence datasets

	CMU Motion	GTZAN
RNN-GRBM	1.4558	0.1971
RNN-ssRBM	1.8951	0.3990
pIAF-RNN-GRBM	**1.4229**	**0.1822**
pIAF-RNN-ssRBM	1.5815	0.3403

Reconstruction Error: The metric of reconstruction error is given as average RMSE per frame. As shown in Table 4, pIAF-RNN-GRBM and pIAF-RNN-ssRBM outperform RNN-GRBM and RNN-ssRBM in both two datasets. For ssRBM-based models, the improvement is significant. It is also observed that the modeling performance of ssRBM-based models is not as good as GRBM-based models. This is because that ssRBM has only one bias parameter in the spike hidden variable H. When we introduce historical transition into the time-variant bias, only one parameter of ssRBM evolves according to the feedback. Therefore, ssRBM-based sequential models cannot well track the time-variant

*http://mocap.cs.cmu.edu

†http://opihi.cs.uvic.ca/sound/genres.tar.gz

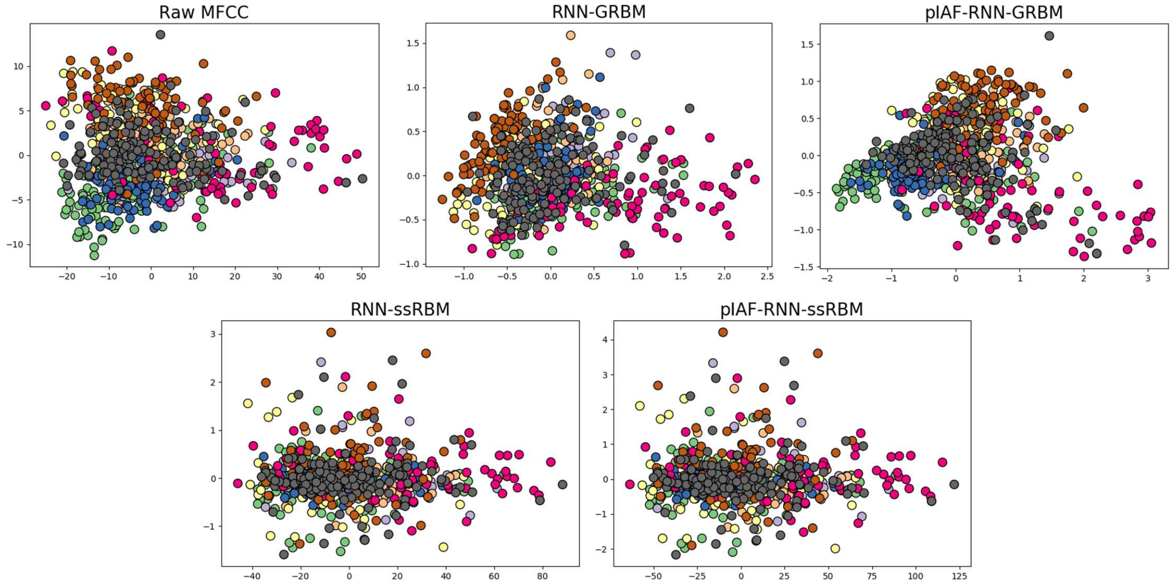

Figure 3: Visualization of the features learnt by various models for GTZAN dataset (Nodes with same color belong to a specific genre.)

distribution of data. The introduction of pIAF provides a larger error reduction.

Feature Learning: RBM is an important tool on representation learning. For GRBM-based models, we consider the conditional mean $P(H_n|V_n, d_{n-1})$ as feature. For ssRBM-based models, we extract feature by the conditional mean of $H_n \odot S_n$. We flatten the feature matrix through time axis and then run a PCA with 2 components. The results of GTZAN dataset are visualized in Figure 3. It is shown that the visualization of features extracted by RNN-ssRBM and pIAF-RNN-ssRBM are similar. But the improvement of reconstruction quality is large due to the incorporation of pIAF. The features learnt by pIAF-RNN-GRBM commendably preserves the properties of the input MFCC, while the dimension has been compressed to a quarter. Some nodes with green, blue and orange colors are merged into masses, and locate further away from the black nodes.

Table 5: Music Genre Classification on GTZAN dataset

	SVM	k-NN
RNN-GRBM	29.08	29.08
RNN-ssRBM	25.51	20.41
pIAF-RNN-GRBM	**37.24**	**37.24**
pIAF-RNN-ssRBM	30.10	20.41
	GPC	RF
RNN-GRBM	40.31	35.71
RNN-ssRBM	20.41	24.49
pIAF-RNN-GRBM	**48.47**	**39.80**
pIAF-RNN-ssRBM	20.41	22.45

The quality of extracted features can be further evaluated by applying them in classification. The flatten features extracted from GTZAN dataset are applied as the input for multiple classifiers to predict the music genres. To demonstrate that pIAF-RNN-RBMs are capable of learning more expressive descriptors for time series, we consider only a group of classic and simple classifiers, including Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), Gaussian Process Classifier (GPC) and Random Forest (RF). The classification accuracies are given in Table 5. As the first and second principle components of their extracted features are almost identical in Figure 3, the classification results of RNN-ssRBM and pIAF-RNN-ssRBM are also similar. Regardless of the choices of classifiers, the genre prediction accuracies of pIAF-RNN-GRBM are consistently the best in these four models, which demonstrates that the features extracted by our method is more expressive.

6 Conclusion

In this paper, we propose a generalized family of Restricted Boltzmann Machines with deep neural network structure by defining a pair-wise inverse autoregressive flow. The pIAF-RBM framework we proposed in this paper shows appealing improvement in diverse tasks, including probabilistic modeling and representation learning for both non-sequential and sequential data. In our future work, we intend to expand our models into convolutional structures.

Acknowledgments

The work is supported by the National Science Foundation (NSF) under the grants CNS-1526843 and ECCS-1731238.

References

[Bengio et al., 2013] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1798–1828.

[Boulanger-Lewandowski et al., 2012] Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2012). Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. ArXiv e-prints.

[Carlson et al., 2015] Carlson, D., Cevher, V., and Carin, L. (2015). Stochastic Spectral Descent for Restricted Boltzmann Machines. In AISTATS, pages 111–119.

[Courville et al., 2014] Courville, A., Desjardins, G., Bergstra, J., and Bengio, Y. (2014). The spike-and-slab rbm and extensions to discrete and sparse data distributions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(9):1874–1887.

[Dahl et al., 2010] Dahl, G., aurelio Ranzato, M., rahman Mohamed, A., and Hinton, G. E. (2010). Phone recognition with the mean-covariance restricted boltzmann machine. In NIPS, pages 469–477.

[Dheeru and Karra Taniskidou, 2017] Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository.

[Germain et al., 2015] Germain, M., Gregor, K., Murray, I., and Larochelle, H. (2015). Made: Masked autoencoder for distribution estimation. In ICML, pages 881–889.

[Guo et al., 2018] Guo, H., Kara, K., and Zhang, C. (2018). Layerwise systematic scan: Deep boltzmann machines and beyond. In AISTATS, pages 178–187.

[Hinton, 2012] Hinton, G. E. (2012). A practical guide to training restricted boltzmann machines. In Neural Networks: Tricks of the Trade: Second Edition, pages 599–619.

[Kingma et al., 2016] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In NIPS, pages 4743–4751.

[Krizhevsky, 2009] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

[Le Roux and Bengio, 2008] Le Roux, N. and Bengio, Y. (2008). Representational power of restricted boltzmann machines and deep belief networks. Neural Computation, 20(6):1631–1649.

[Lecun et al., 1998] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324.

[Netzer et al., 2011] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011.

[Ping and Ihler, 2017] Ping, W. and Ihler, A. (2017). Belief Propagation in Conditional RBMs for Structured Prediction. In AISTATS, pages 1141–1149.

[Rezende and Mohamed, 2015] Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. In ICML, pages 1530–1538.

[Salakhutdinov and Murray, 2008a] Salakhutdinov, R. and Murray, I. (2008a). On the quantitative analysis of deep belief networks. In ICML, pages 872–879.

[Salakhutdinov and Murray, 2008b] Salakhutdinov, R. and Murray, I. (2008b). On the quantitative analysis of Deep Belief Networks. In ICML, pages 872–879.

[Tieleman, 2008] Tieleman, T. (2008). Training restricted boltzmann machines using approximations to the likelihood gradient. In ICML, pages 1064–1071.

[van den Oord et al., 2016] van den Oord, A., Kalchbrenner, N., Espeholt, L., kavukcuoglu, k., Vinyals, O., and Graves, A. (2016). Conditional image generation with pixelcnn decoders. In NIPS, pages 4790–4798.