Capsule Generative Models*

Yifeng Li[1][0000−0002−4873−6928] and Xiaodan Zhu[2][0000−0003−3856−3696]

[1] Digital Technologies Research Centre, National Research Council Canada, Ottawa ON K1A 0R6, Canada yifeng.li@nrc-cnrc.gc.ca
[2] Department of Electrical and Computer Engineering, Queen's University, Kingston, ON K7L 3N6, Canada xiaodan.zhu@queensu.ca

Abstract. Neuroscience studies inspire that structures are needed in the hidden space of deep learning models. In this paper, we propose a capsule restricted Boltzmann machine and a capsule Helmholtz machine by replacing individual hidden variables with encapsulated groups of hidden variables. Our preliminary experiments show that capsule activities in both models can be dynamically determined in context, and these activity spectra exhibit between-class patterns and within-class variations. This paper offers a novel approach to visualizing and understanding the hidden states.

Keywords: Capsule · Restricted Boltzmann machine · Helmholtz machine · Deep generative model

1 Introduction

The development of deep neural networks, based on distributed representations [10], has achieved large successes in a wide range of fields such as computer vision [15,14], speech recognition [12], and natural language processing (e.g., language models [1]), among many others. The limitation on supporting the manipulation of complex semantics and relations [5,7], however, triggers the exploration for revised distributed representations and alternative models. Recently, the concept of *deterministic* capsule networks [20,8] has been proposed to address the limitations of typical *discriminative* models, i.e., convolutional neural networks. In this paper, we further introduce and adapt the concept of capsules to *generative* models. Generative models have evolved from restricted Boltzmann machine (RBM) based models, such as Helmholtz machines (HMs) [4] and deep belief nets [11], to variational auto-encoders (VAEs) [13] and generative adversarial networks (GANs) [6] with prosperity in image and video generation.

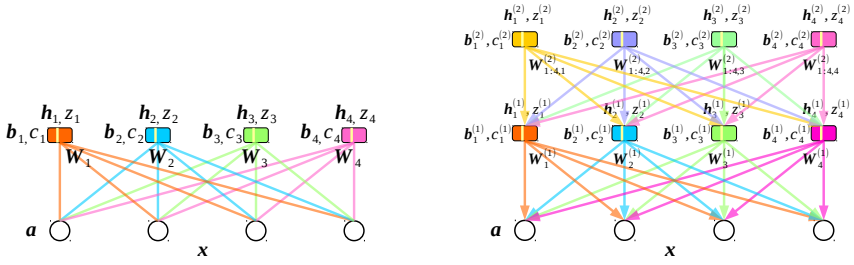
In this paper, we propose a capsule RBM model that has *generative* capsules in its hidden layer, and a capsule HM where a capsule's status depends on active capsules above it. We also contribute a new wake-sleep algorithm for estimating the model parameters in capsule HM. We shall first present these models and their learning algorithms, and show promising benefits of using capsules in deep generative models (DGMs). The paper is then concluded with improvements and observations made with the proposed models.

*Supported by National Research Council Canada.

2 Method

2.1 Capsule RBM

Based on theories of exponential family RBMs (exp-RBMs) [23,17,16] and discriminative capsule nets [20,8], we propose capsule RBM (cap-RBM) replacing hidden variables with stochastic capsules. We use $\boldsymbol{x}$ to represent the visible vector. The k-th capsule in the hidden layer includes $\boldsymbol{h}_k$ which hosts multiple hidden random variables following any distribution from the exponential family, and z_k a binary random variable indicating whether this capsule is active. An example of such network is displayed in Fig. 1a.



(a) Capsule RBM. (b) Capsule HM.

Fig. 1: Examples of capsule RBM and capsule HM. The meanings of the variables and parameters in capsule RBM are explained as follows. $\boldsymbol{a}$: bias parameters on the visible units. $\boldsymbol{b}_k$: bias parameters on $\boldsymbol{h}_k$. c_k: bias parameter on z_k. $\boldsymbol{W}_k$: interaction matrix between the k-th capsule ($\boldsymbol{h}_k$ and z_k) and $\boldsymbol{x}$. $\boldsymbol{\Omega}$: interaction matrix between $\boldsymbol{x}$ and $\boldsymbol{z}$.

Model Definition We assume visible vector $\boldsymbol{x}$ has M units, the hidden layer has K capsules (each contains J random variables and one switch variable). For simplicity, we write $\boldsymbol{h} = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_K\}$. Following the steps in defining exp-RBMs, first of all, the base distributions of capsule RBM are defined in natural form as

$$\begin{cases} p(\boldsymbol{x}) = \prod_{m=1}^{M} \exp\left(\boldsymbol{a}_m^{\mathrm{T}} \boldsymbol{s}_m + \log f(x_m) - A(\boldsymbol{a}_m)\right) & (1) \\ p(\boldsymbol{h}) = \prod_{k=1}^{K} \prod_{j=1}^{J} \exp\left(\boldsymbol{b}_{k,j}^{\mathrm{T}} \boldsymbol{t}_{k,j} + \log g(h_{k,j}) - B(\boldsymbol{b}_{k,j})\right) & (2) \\ p(\boldsymbol{z}) = \prod_{k=1}^{K} \exp\left(c_k z_k - C_k(z_k)\right), & (3) \end{cases}$$

where $p(\boldsymbol{x})$ and $p(\boldsymbol{h})$ are from the exponential class, and $p(\boldsymbol{z})$ is Bernoulli distributed. $\boldsymbol{a}_m = [a_m^{(1)}, \cdots, a_m^{(R)}]^{\mathrm{T}}$ and $\boldsymbol{b}_{k,j} = [b_{k,j}^{(1)}, \cdots, b_{k,j}^{(U)}]^{\mathrm{T}}$ are respectively

the natural parameters of x_m and $h_{k,j}$, $\boldsymbol{s}_m$ and $\boldsymbol{t}_{k,j}$ are respectively their sufficient statistics, $A(\boldsymbol{a}_m)$ and $B(\boldsymbol{b}_{k,j})$ are corresponding log-partition functions, and $f(x_m)$ and $g(h_{k,j})$ are base measures. For example, if $x_m \sim \mathcal{N}(\mu, \lambda^{-1})$, then $\boldsymbol{a}_m = [a_m^{(1)}, a_m^{(2)}]^{\mathrm{T}} = [\mu\lambda, -\frac{\lambda}{2}]^{\mathrm{T}}$, $\boldsymbol{s}_m = [x, x^2]^{\mathrm{T}}$, $A(\boldsymbol{a}_m) = \frac{1}{2}\log\frac{\pi}{-a_m^{(2)}} - \frac{a_m^{(1)}}{4a_m^{(2)}}$, and $f(x_m) = 1$. If $h_{k,j} \sim \mathcal{BE}(p)$, i.e. Bernoulli, $b_{k,j} = \log\frac{p}{1-p}$, $t_{k,j} = x$, $B(b_{k,j}) = \log\big(1 + \exp(b_{k,j})\big)$, and $g(h_{k,j}) = 1$. Similar notations apply to $p(\boldsymbol{z})$. See [17] for a list of exponential family distributions formulated in natural parameters.

Second, the joint distribution has a similar form as in regular RBM: $p(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{z}) = \frac{1}{Z}\exp\big(-E(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{z})\big)$, where Z is the partition function, and the energy function $E(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{z})$ is defined as

$$E(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{z}) = -\sum_{m=1}^{M}\big(\boldsymbol{a}_m^{\mathrm{T}}\boldsymbol{s}_m + \log f(x_m)\big) - \sum_{k=1}^{K}\sum_{j=1}^{J}\big(\boldsymbol{b}_{k,j}^{\mathrm{T}}\boldsymbol{t}_{k,j} + \log g(h_{k,j})\big) - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{z} - \sum_{k=1}^{K} z_k(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{h}_k) - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Omega}\boldsymbol{z}, \tag{4}$$

where the first three terms are bias terms and the last two terms define the interaction between observations and capsules.

Third, we can obtain the conditionals in decomposable forms as given by

$$p(\boldsymbol{x}|\boldsymbol{h}, \boldsymbol{z}) = \prod_{m=1}^{M} p\big(x_m|\eta(\hat{\boldsymbol{a}}_m)\big), \tag{5}$$

$$p(\boldsymbol{h}|\boldsymbol{x}, \boldsymbol{z}) = \prod_{k=1}^{K}\prod_{j=1}^{J} p\big(h_{k,j}|\eta(\hat{\boldsymbol{b}}_{k,j})\big), \tag{6}$$

$$p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{h}) = \prod_{k=1}^{K} \mathcal{BE}\big(z_k|\eta(\hat{c}_k)\big), \tag{7}$$

where function $\eta(\cdot)$ maps the natural posterior parameters to the standard forms, and the posterior natural parameters are computed as

$$\hat{a}_m^{(r)} = a_m^{(r)} + \delta(r = 1)\Big(\sum_{k=1}^{K} z_k(\boldsymbol{W}_k)_{m,:}\boldsymbol{h}_k + \boldsymbol{\Omega}_{m,:}\boldsymbol{z}\Big), \tag{8}$$

$$\hat{b}_{k,j}^{(u)} = b_{k,j}^{(u)} + \delta(u = 1)\Big(z_k(\boldsymbol{W}_k^{\mathrm{T}})_{j,:}\boldsymbol{x}\Big), \tag{9}$$

$$\hat{c}_k = c_k + \boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{h}_k + (\boldsymbol{\Omega}^{\mathrm{T}})_{k,:}\boldsymbol{x}, \tag{10}$$

where, $r \in \{1, \cdots, R\}$, $u \in \{1, \cdots, U\}$, without loss of generality we assume the first statistics for x_m and $h_{k,j}$ are respectively x_m and $h_{k,j}$, and $\delta(\cdot)$ is a Kronecker delta function. Transformation tables for exponential family distributions between natural and standard forms can be found in [17].

Variable z_k determines whether the k-th capsule can interact with the visible variables; that is, only when $z_k = 1$, the k-th capsule has an impact on $\boldsymbol{x}$. Thus,

the model dynamically decides on which capsules are active depending on the context. The conditional distribution of $\boldsymbol{x}$ is determined by its interactions with the active capsules and the switch variable. The conditional distribution of $\boldsymbol{h}_k$ depends on its interaction with $\boldsymbol{x}$. In turn, the value of the switch variable $\boldsymbol{z}$ depends on all the other variables. Specifically, this model has the following metrics: (1) An observation $\boldsymbol{x}$ will only activate a subset of capsules, some of which can be ubiquitous while some may be specific. (2) This leads to a block-wise structured representation of an observation in the hidden space. (3) Through examining the activity of dozens of capsules, indicated by $\boldsymbol{z}$, it offers a better interpretation compared to disentangling the meaning of hundreds or thousands of individual hidden variables in other generative models.

Model Learning The model parameters are denoted by $\boldsymbol{\theta} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{W}, \boldsymbol{\Omega}\}$, where we simply write $\boldsymbol{b} = \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_K\}$ and $\boldsymbol{W} = \{\boldsymbol{W}_1, \cdots, \boldsymbol{W}_K\}$. Similar to exp-RBM, the gradients w.r.t. these parameters can be computed in the form:

$$\Delta_\theta = \frac{1}{N}\sum_{n=1}^{N}\Big(\mathrm{E}_{p(\boldsymbol{h},\boldsymbol{z}|\boldsymbol{x}_n)}\big[\frac{\partial E(\boldsymbol{x}_n,\boldsymbol{h},\boldsymbol{z})}{\partial\boldsymbol{\theta}}\big] - \mathrm{E}_{p(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z})}\big[\frac{\partial E(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z})}{\partial\boldsymbol{\theta}}\big]\Big). \tag{11}$$

Derivatives of the energy function w.r.t. these parameters are computed as below:

$$\begin{aligned}
&\frac{\partial E(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z})}{\partial a_m^{(r)}} = -s_m^{(r)}, \qquad \frac{\partial E(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z})}{\partial b_{k,j}^{(u)}} = -t_{k,j}^{(u)}, \quad \frac{\partial E(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z})}{\partial \boldsymbol{c}} = -\boldsymbol{z},\\
&\frac{\partial E(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z})}{\partial \boldsymbol{W}_k} = -z_k\boldsymbol{x}\boldsymbol{h}_k^{\mathrm{T}}, \qquad \frac{\partial E(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z})}{\partial \boldsymbol{\Omega}} = -\boldsymbol{x}\boldsymbol{z}^{\mathrm{T}}.
\end{aligned} \tag{12}$$

In Eq. 11, the first part is a data-dependent term that can be estimated by mean-field variational approximation using the conditional distributions over $\boldsymbol{h}$ and $\boldsymbol{z}$ while fixing $\boldsymbol{x}_n$. The second part is a data-independent term requiring Monte Carlo approximation where sampling can be performed by Gibbs sampling using the conditional distributions of $\boldsymbol{x}$, $\boldsymbol{h}$, and $\boldsymbol{z}$, respectively.

2.2 Capsule HM

Based on the concept of capsule RBM, there exist multiple ways to construct a directed DGM. In this paper, we investigate the most basic idea that the distribution of capsule in layer l depends on its interactions with all active capsules in layer $l+1$. We name this DGM as capsule Helmholtz machine (capsule HM or cap-HM). An example of capsule HM with two hidden capsule layers is illustrated in Fig. 1b. Other sophisticated forms of capsule DGMs are further discussed at the end of this paper.

Model Definition For a capsule HM with L hidden layers, the joint distribution of visible variables and hidden variables can be decomposed as follows:

$$p(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{h}^{(1)},\boldsymbol{z}^{(1)})\Big(\prod_{l=1}^{L-1} p(\boldsymbol{h}^{(l)},\boldsymbol{z}^{(l)}|\boldsymbol{h}^{(l+1)},\boldsymbol{z}^{(l+1)})\Big)p(\boldsymbol{h}^{(L)},\boldsymbol{z}^{(L)}), \tag{13}$$

where, at the left-hand side, we let $\boldsymbol{h} = \{\boldsymbol{h}^{(1)}, \cdots, \boldsymbol{h}^{(L)}\}$, $\boldsymbol{z} = \{\boldsymbol{z}^{(1)}, \cdots, \boldsymbol{z}^{(L)}\}$ for narrative convenience. The pair $\boldsymbol{h}^{(l)}$ and $\boldsymbol{z}^{(l)}$ host all capsules in layer l, where the k-th capsule can be written as $C_k^{(l)} = \{\boldsymbol{h}_k^{(l)}, z_k^{(l)}\}$. There are three components at the right-hand side of Eq. 13. They are respectively discussed below.

First, the conditional distribution of visible variable is similar to that in capsule RBM:

$$p(\boldsymbol{x}|\boldsymbol{h}^{(1)}, \boldsymbol{z}^{(1)}) = \prod_{m=1}^{M} p(x_m|\eta(\hat{\boldsymbol{a}}_m)), \tag{14}$$

where $\hat{\boldsymbol{a}}^{(r)} = \boldsymbol{a}^{(r)} + \delta(r=1)\big(\sum_{k_1=1}^{K_1} z_{k_1}^{(1)} \boldsymbol{W}_{k_1}^{(1)} \boldsymbol{h}_{k_1}^{(1)} + \boldsymbol{\Omega}\boldsymbol{z}\big)$.

Second, different from the traditional Helmholtz machine and directed DGMs, capsule HM is a mixture of directed and undirected components. The interaction of capsules in two adjacent layers are directed, but the conditional distribution of capsules within an intermediate layer is in fact an undirected sub-model which is defined using a special energy function, as formulated below:

$$p(\boldsymbol{h}^{(l)}, \boldsymbol{z}^{(l)}|\boldsymbol{h}^{(l+1)}, \boldsymbol{z}^{(l+1)}) = \frac{1}{Z_l} \exp\big(-E(\boldsymbol{h}^{(l)}, \boldsymbol{z}^{(l)}|\boldsymbol{h}^{(l+1)}, \boldsymbol{z}^{(l+1)})\big), \tag{15}$$

where the energy is computed as

$$E(\boldsymbol{h}^{(l)}, \boldsymbol{z}^{(l)}|\boldsymbol{h}^{(l+1)}, \boldsymbol{z}^{(l+1)}) = -\sum_{k_l=1}^{K_l} \big(\sum_{u=1}^{U} \boldsymbol{b}_{k_l}^{(l,u)\mathrm{T}} \boldsymbol{t}_{k_l}^{(l,u)} + \sum_{j=1}^{J} \log g(h_{k_l,j}^{(l)})\big) - \boldsymbol{c}^{(l)\mathrm{T}} \boldsymbol{z}^{(l)}$$
$$- \sum_{k_l=1}^{K_l} \sum_{k_{l+1}=1}^{K_{l+1}} z_{k_l}^{(l)} z_{k_{l+1}}^{(l+1)} \boldsymbol{h}_{k_l}^{(l)\mathrm{T}} \boldsymbol{W}_{k_l,k_{l+1}}^{(l+1)} \boldsymbol{h}_{l+1}^{(l+1)} - \boldsymbol{z}^{(l)\mathrm{T}} \boldsymbol{\Omega}^{(l+1)} \boldsymbol{z}^{(l+1)}, \tag{16}$$

where $\{\boldsymbol{b}_{k_l}^{(l,u)}, \boldsymbol{c}^{(l)}, \boldsymbol{W}_{k_l,k_{l+1}}^{(l+1)}, \boldsymbol{\Omega}^{(l+1)}\}$ are the parameters in this sub-model, K_l is the number of capsules in layer l, and U is the number of sufficient statistics. In virtue of the exponential family formulation, we can obtain the conditional distributions of individual variables in Eq. 15 for approximation and sampling:

$$p\big(\boldsymbol{h}^{(l)}|\boldsymbol{z}^{(l)}, \boldsymbol{h}^{(l+1)}, \boldsymbol{z}^{(l+1)}\big) = \sum_{k_l=1}^{K_l} \sum_{j=1}^{J} p\big(h_{k_l,j}^{(l)}|\eta(\hat{\boldsymbol{b}}_{k_l,j}^{(l)})\big), \tag{17}$$

$$p\big(\boldsymbol{z}^{(l)}|\boldsymbol{h}^{(l)}, \boldsymbol{h}^{(l+1)}, \boldsymbol{z}^{(l+1)}\big) = \sum_{k_l=1}^{K_l} \mathcal{BE}\big(z_{k_l}|\eta(\hat{c}_{k_l})\big), \tag{18}$$

where J is the number of random variables within a capsule, $\mathcal{BE}(\cdot)$ is Bernoulli probability function, and the posterior natural parameters are computed as

$$\hat{\boldsymbol{b}}_{k_l}^{(u)} = \boldsymbol{b}_{k_l}^{(u)} + \delta(u=1)\big(\sum_{k_{l+1}=1}^{K_{l+1}} z_{k_l}^{(l)} z_{k_{l+1}}^{(l+1)} \boldsymbol{W}_{k_l,k_{l+1}}^{(l+1)} \boldsymbol{h}_{l+1}^{(l+1)}\big), \tag{19}$$

$$\hat{c}_{k_l}^{(l)} = c_{k_l}^{(l)} + \sum_{k_{l+1}=1}^{K_{l+1}} z_{k_{l+1}}^{(l+1)} \boldsymbol{h}_{k_l}^{(l)\mathrm{T}} \boldsymbol{W}_{k_l,k_{l+1}}^{(l+1)} \boldsymbol{h}_{k_{l+1}}^{(l+1)} + \boldsymbol{\Omega}_{k,:}^{(l+1)} \boldsymbol{z}^{(l+1)}. \tag{20}$$

From Eq. 19, we can see that the k_{l+1}-th capsule at layer $l+1$ (denoted by $C_{k_{l+1}}^{(l+1)}$) influences the k_l-th capsule at layer l (denoted by $C_{k_l}^{(l)}$), only when both capsules are active. Eq. 20 implies that only active capsules at the upper layer can potentially activate capsule $C_{k_l}^{(l)}$.

Finally, the distribution of the top layer can be assumed to be completely factorizable, as formulated below:

$$\begin{aligned} p\big(\boldsymbol{h}^{(L)}, \boldsymbol{z}^{(L)}\big) &= p\big(\boldsymbol{h}^{(L)}\big) p\big(\boldsymbol{h}^{(Z)}\big) \\ &= \sum_{k_L=1}^{K_L} \sum_{j=1}^{J} p\big(h_{k_L,j}^{(L)} | \eta(\boldsymbol{b}_{k_L,j}^{(L)})\big) \sum_{k_L=1}^{K_L} p\big(z_{k_L} | \eta(c_{k_L}^{(L)})\big). \end{aligned} \tag{21}$$

Using sophisticated structure on the top layer is possible, like the case of DBN where RBM is topped on HM. We will pursue this direction in future development.

Given a sample to this capsule HM, a subset of capsules at each layer could be activated. This dynamic context-dependent pattern reflected by $\boldsymbol{z}$ thus offers a novel way to visualize and interpret the hidden representations in DGM. We will prove this concept in the experimental section.

Model Learning In the same spirit of HM, a corresponding recognition component is introduced to capsule HM to approximate inference $p_{\boldsymbol{\theta}}(\boldsymbol{h}, \boldsymbol{z}|\boldsymbol{x})$ using $q_{\boldsymbol{\phi}}(\boldsymbol{h}, \boldsymbol{z}|\boldsymbol{x})$. The model thus has generative parameters $\boldsymbol{\theta} = \{\boldsymbol{a}^{(r)}, \boldsymbol{b}_{k_l}^{(l,u)}, \boldsymbol{c}^{(l)}, \boldsymbol{W}_{k_{l-1},k_l}^{(l)}, \boldsymbol{\Omega}^{(l)}\}$ and recognition parameters $\boldsymbol{\phi} = \{\boldsymbol{b}_{k_l}^{(R,l,u)}, \boldsymbol{c}^{(R,l)}, \boldsymbol{W}_{k_{l-1},k_l}^{(R,l)}, \boldsymbol{\Omega}^{(R,l)}\}$. We propose a wake-sleep algorithm which is different from HM's wake-sleep algorithm in two aspects: (1) $q(\boldsymbol{h}, \boldsymbol{z}|\boldsymbol{x})$ in wake phase and $p(\boldsymbol{h}^{(l-1)}, \boldsymbol{z}^{(l-1)}|\boldsymbol{h}^{(l)}, \boldsymbol{z}^{(l)})$ in sleep phase are respectively mean-field variational approximations, because each intermediate layer is an energy-based model (note that $p(\boldsymbol{x}|\boldsymbol{h}^{(1)}, \boldsymbol{z}^{(1)})$ does not use variational approximation, but an exponential family distribution); (2) we applied a k-step (persistent) contrastive divergence sampling method [22] (we used $k=1$ in our experiments), such that the states of the top layer $\{\boldsymbol{h}^{(L)}, \boldsymbol{z}^{(L)}\}$ obtained from the wake phase are recycled as the start of the sleep phase. This also implies that Gibbs sampling can be used when generating samples through alternating wake-sleep phases after model learning. Prior work on Helmholtz machine [16] and variational autoencoder [25] has already shown that ancestral sampling often generates unsensible samples. The generative parameters are estimated by maximizing the evidence lower bound (ELBO) which is defined as

$$J(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathrm{E}_{q_\phi(\boldsymbol{h},\boldsymbol{z}|\boldsymbol{x})}\big[p_\theta(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{z})\big] + H\big(q_\phi(\boldsymbol{h}, \boldsymbol{z}|\boldsymbol{x})\big), \tag{22}$$

where $H(\cdot)$ is the entropy of a distribution. Thus, the gradient w.r.t. the generative parameters can be generally written as

$$\Delta_{\boldsymbol{\theta}} = \frac{\partial J(\boldsymbol{x})}{\partial \boldsymbol{\theta}} = \mathrm{E}_{q(\boldsymbol{h},\boldsymbol{z}|\boldsymbol{x})}\Big[\frac{\partial \log p(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{z})}{\partial \boldsymbol{\theta}}\Big]. \tag{23}$$

Since the model is described in the language of exponential family, we can make use of an important property of exponential family distribution (first-order

derive of log-partition function w.r.t. natural parameter is the expectation of sufficient statistics) to obtain gradients of the interaction matrices from the first hidden layer to visible layer:

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z})}{\partial \boldsymbol{W}^{(1)}_{k_1}} &= \frac{\partial \log p(\boldsymbol{x}|\boldsymbol{h}^{(1)},\boldsymbol{z}^{(1)})}{\partial \hat{\boldsymbol{a}}^{(1)}}\frac{\partial \hat{\boldsymbol{a}}^{(1)}}{\partial \boldsymbol{W}^{(1)}_{k_1}} \\ &= \Big(\frac{\partial \hat{\boldsymbol{a}}^{(1)\mathrm{T}}\boldsymbol{x}}{\partial \hat{\boldsymbol{a}}^{(1)}} - \frac{\partial A(\hat{\boldsymbol{a}}^{(1)})}{\partial \hat{\boldsymbol{a}}^{(1)}}\Big)\frac{\partial \hat{\boldsymbol{a}}^{(1)}}{\partial \boldsymbol{W}^{(1)}_{k_1}} = (\boldsymbol{x} - \langle \boldsymbol{x}\rangle)(z_{k_1}\boldsymbol{h}^{(1)}_{k_1})^{\mathrm{T}}, \end{aligned} \tag{24}$$

where $\langle \boldsymbol{x}\rangle$ denotes expectation under the generative distribution.

It is apparently challenging to compute gradients of the parameters in the intermediate hidden layers ($1 < l < L$). Considering that the corresponding conditionals are undirected energy-based sub-models, we can apply the findings in general Boltzmann machines to derive the gradients:

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{x},\boldsymbol{h},\boldsymbol{z})}{\partial \boldsymbol{W}^{(l+1)}_{k_l,k_{l+1}}} &= \frac{\partial \log p(\boldsymbol{h}^{(l)},\boldsymbol{z}^{(l)}|\boldsymbol{h}^{(l+1)},\boldsymbol{z}^{(l+1)})}{\partial \boldsymbol{W}^{(l+1)}_{k_l,k_{l+1}}} \\ &= \frac{\partial E(\boldsymbol{h}^{(l)},\boldsymbol{z}^{(l)})}{\partial \boldsymbol{W}^{(l+1)}_{k_l,k_{l+1}}} - \mathrm{E}_{p(\boldsymbol{h}^{(l)},\boldsymbol{z}^{(l)}|\boldsymbol{h}^{(l+1)},\boldsymbol{z}^{(l+1)})}\Big[\frac{\partial E(\boldsymbol{h}^{(l)},\boldsymbol{z}^{(l)})}{\partial \boldsymbol{W}^{(l+1)}_{k_l,k_{l+1}}}\Big] \\ &= (z_{k_l}\boldsymbol{h}^{(l)}_{k_l})(z_{k_{l+1}}\boldsymbol{h}^{(l+1)}_{k_{l+1}})^{\mathrm{T}} - \langle (z_{k_l}\boldsymbol{h}^{(l)}_{k_l})(z_{k_{l+1}}\boldsymbol{h}^{(l+1)}_{k_{l+1}})^{\mathrm{T}}\rangle \\ &= \big(z_{k_l}\boldsymbol{h}^{(l)}_{k_l} - \langle z_{k_l}\boldsymbol{h}^{(l)}_{k_l}\rangle\big)(z_{k_{l+1}}\boldsymbol{h}^{(l+1)}_{k_{l+1}})^{\mathrm{T}}, \end{aligned} \tag{25}$$

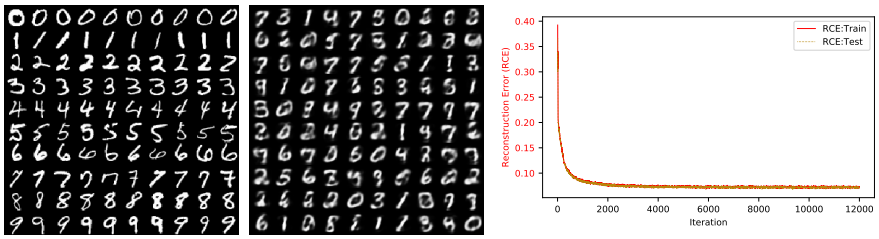
which surprisingly has the same form as in Eq. 24. The gradients of the recognition parameters can be derived in the same way. Thus, we omit them.

3 Experiments

3.1 Capsule RBM

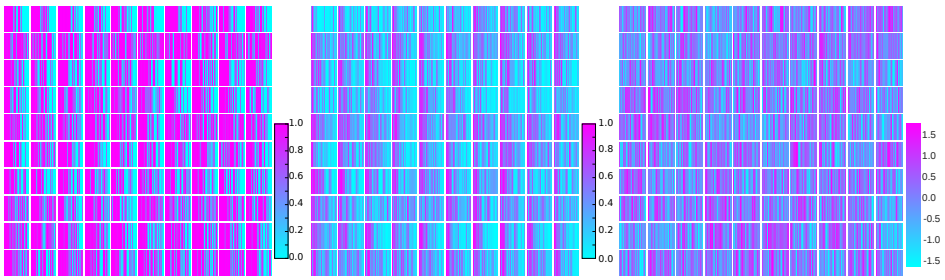
MNIST Data We here investigate the performance of capsule RBM on the MNIST data (`http://yann.lecun.com/exdb/mnist`, see Fig. 2a for some examples from the data). We let both base distributions of $\boldsymbol{x}$ and $\boldsymbol{h}_k$ be Bernoulli, and set K=40, J=16, the initial learning rate to be 0.02 (gradually decreased), batch size 100, and the number of epochs 20. We tracked the reconstruction error of training and test samples, which reduced quickly along learning (see Fig. 2c). Fig. 2b shows 100 images generated by Gibbs sampling with the learned model.

We obtained the capsule activities (values of $\boldsymbol{z}$) of the actual images as in Fig. 2a using mean-field approximation, and displayed them as spectra in Fig. 3a. Interestingly, class-wise patterns (e.g. patterns in classes 0 and 1) and within-class variations can be observed from the spectra. Furthermore, digits sharing similar parts tend to have partially similar patterns (e.g. digits "1", "4", "7", and "9" all have vertical strokes). By contrast, if a Bernoulli-Bernoulli RBM (without capsules) with 640 hidden variables was used, its hidden states couldn't very



(a) Actual. (b) Generated. (c) Reconstruction errors.

Fig. 2: (a): Actual MNIST images. (b): Generated MNIST images by capsule RBM. (c): Reconstruction errors of training and testing MNIST images during learning capsule RBM.



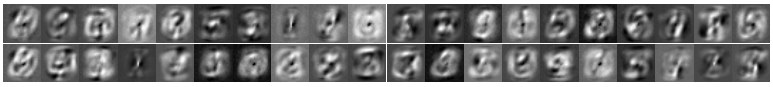
(a) Capsule RBM. (b) RBM. (c) VAE.

Fig. 3: Comparison of hidden representations of MNIST images from Fig. 2a. (a): Capsule activities in capsule RBM. (b): Hidden states in a Bernoulli-Bernoulli RBM without capsules. (c): Hidden states inferred by encoder in VAE.

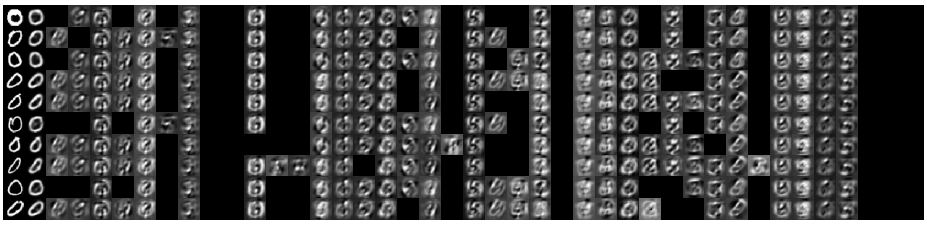
clearly exhibit such patterns to the interpretable level (see Fig. 3b). We also learned a VAE with two convolutional layers in its encoder, 40 Gaussian latent variables, and two convolutional layer in its decoder. We then obtained the values of latent variables using the encoder fed by the images from Fig. 2a, and display them in Fig. 3c. The VAE latent space does not show any consistent patterns.

To investigate what information the k-th capsule can represent, we obtained the values of this capsule's projective field by sampling $\boldsymbol{x}$ using Markov chain over $p(\boldsymbol{x}, \boldsymbol{h}|\boldsymbol{z}_k)$ where $\boldsymbol{z}_k$ is a one-hot vector of length K with the k-th element being one. Using this method, we visualize all capsules' projective fields in Fig. 4a, and find that these projective fields do show some distinct signatures. Furthermore, for an actual input image, we can infer the values of each capsule and then plot the reconstructed image and the projective fields of active capsules (see Fig. 4b and 4c for example). From such plots, we can easily see what kind of capsules are shared among classes or unique to a specific class.

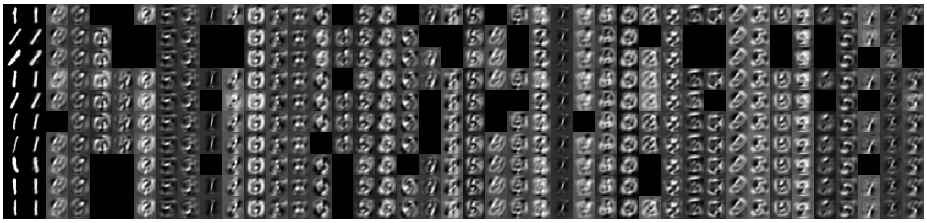
Fashion-MNIST Data We also explored capsule RBM on the Fashion-MNIST data [24] (see Fig. 5a for some test images). We let both base distributions of $\boldsymbol{x}$



(a) Projective field of each capsule.



(b) Digit 0s.

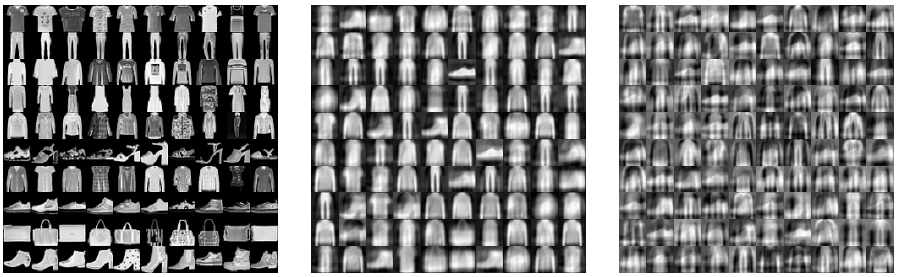


(c) Digit 1s.

Fig. 4: Visualizing the projective field of each capsule on MNIST data. (a): Visualization of each capsule's projective field. (b): Digit 0s approximated by a linear combination of a subset of capsules' projective fields. (c): Digit 1s approximated by a linear combination of a subset of capsules' projective fields. In (b) and (c), the first columns are actual images; the second columns are corresponding reconstructed images; the rest columns are corresponding to projective fields of individual capsules.

and $\boldsymbol{h}_k$ be Gaussian, and set K=20, J=16, and the initial learning rate to be 0.005. Fig. 5b gives some images generated from the capsule RBM model after training. In contrast, Fig. 5c shows some images generated from a Gaussian-Gaussian RBM (without capsules) with 320 hidden units. We can see that this RBM suffers from severe problem of overlapping objects when generating new samples using Gibbs sampling, while our capsule RBM does not suffer from such symptom. We also tried smaller numbers of hidden variables in the Gaussian-Gaussian RBM, but this problem did not disappear. (Interestingly, an RBM, whose hidden states are instead discrete, usually does not have such issue.)

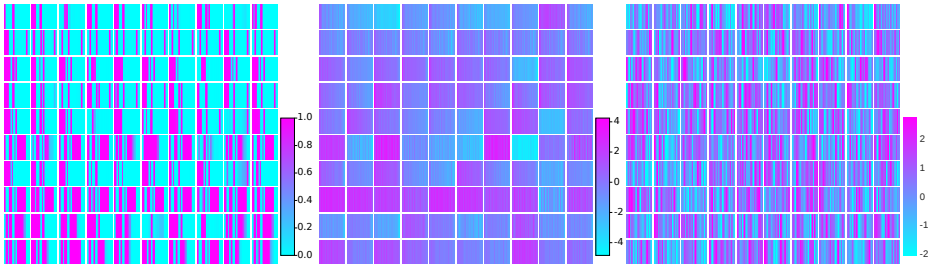
Fig. 6a displays the capsule activities of our capsule RBM corresponding to actual test images in Fig. 5a. Again, the capsule activity spectra exhibit class-specific patterns (e.g. T-shirt/top (row 1) versus Trouser (row 2)). Close classes tend to share similar spectral patterns (e.g. Sandal (row 6) and Sneaker (row 8)). Furthermore, within each class, variations can be observed among capsule activity spectra of samples. When we ran the Gaussian-Gaussian RBM without capsules, the hidden representations in continuous values corresponding to images



(a) Actual images. (b) Generated by cap-RBM. (c) Generated by RBM.

Fig. 5: Performance of capsule RBM on Fashion-MNIST.

from Fig. 5a are totally not visually interpretable as shown in Fig. 6c. We trained a VAE with two convolutional layers and 20 latent variables, and find that its latent space corresponding to the same set of actual images does not exhibit consistent patterns. As in the experiments on MNIST data, we also draw the individual capsules' projective fields in Fig. 7 from which we straightforwardly observe shared and specific patterns.

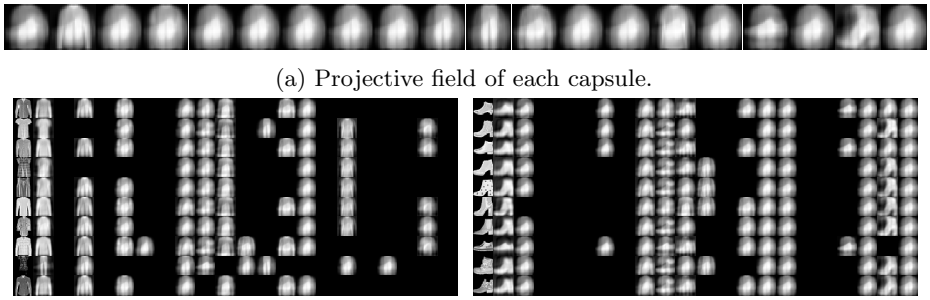


(a) Capsule RBM. (b) Gaussian RBM. (c) VAE.

Fig. 6: Comparison of hidden representations of Fashion-MNIST images from Fig. 5a. (a): Capsule activities in capsule RBM. (c): Hidden states in a Gaussian-Gaussian RBM without capsules. (c): Hidden states inferred by encoder in VAE.

3.2 Capsule HM

We experimented with a two-hidden-layer capsule HM with 40 capsules per layer on the MNIST data. Similarly we also investigated a two-hidden-layer capsule HM with 20 capsules per layer on the Fashion-MNIST data. The distributions, capsule size, and other hyperparameter settings are the same as in the capsule RBMs for the MNIST and Fashion-MNIST data, respectively. Layer-wise pretraining using capsule RBM is essential before running our wake-sleep fine-tuning algorithm. In each iteration of the wake-sleep algorithm, 20 steps are allowed for mean-filed approximation of capsule values. Fig. 8 displays the learning curves (quantified by

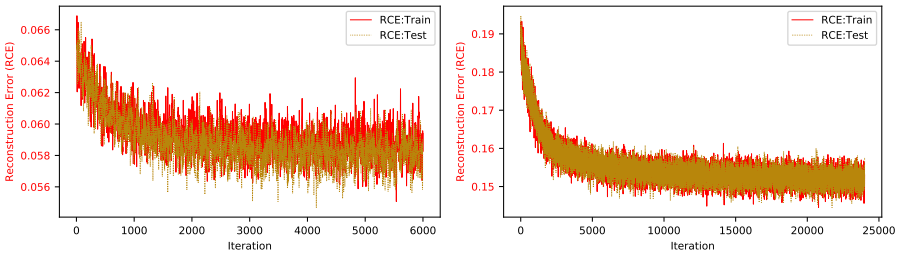


(a) Projective field of each capsule.

(b) Shirts. (c) Ankle boots.

Fig. 7: Visualizing the projective field of each capsule on Fashion-MNIST data. (a): Visualization of each capsule's projective field. b: Shirts approximated by a linear combination of a subset of capsules' projective fields. c: Ankle boots approximated by a linear combination of a subset of capsules' projective fields. In (b) and (c), the first columns are actual images; the second columns are corresponding reconstructed images; the rest columns are corresponding to projective fields of individual capsules.

reconstruction errors) during fine-tuning. The pretraining at each layer converges quickly. Importantly, the fine-tuning algorithm is then able to gradually converge further without sign of overfitting.

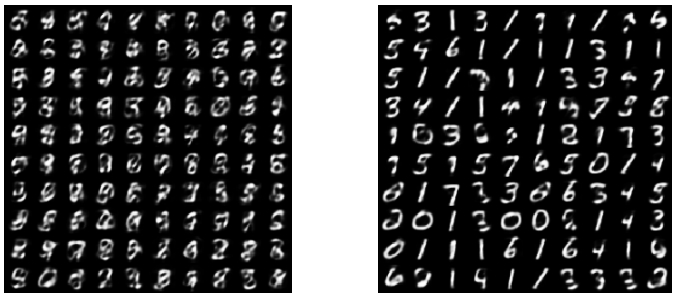


(a) Fine-tuning on MNIST. (b) Fine-tuning on Fashion-MNIST.

Fig. 8: Reconstruction errors in learning processes of Capsule HM on MNIST and Fashion-MNIST data.

After the models were trained, we compared two sampling methods, including ancestral sampling using only the sleep phase and Gibbs sampling through long wake-sleep loops. From Fig. 9a and 9b, we can see that ancestral sampling did not generate images of good quality, while Gibbs sampling could return much better images. This observation is consistent with research on VAE [25].

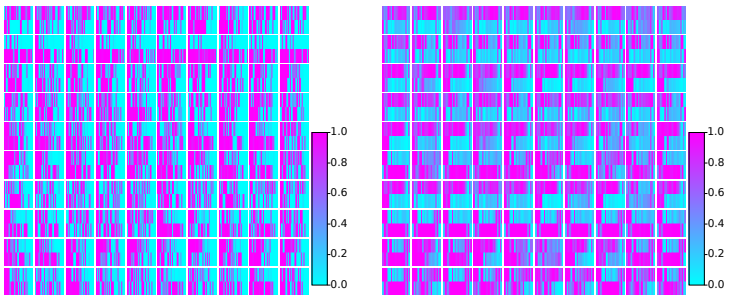
Fig. 10a visualizes the capsule activity spectra inferred using the wake-phase of our algorithm on real MNIST images from Fig. 2a. The two-layer spectra



(a) Ancestral sampling. (b) Gibbs sampling.

Fig. 9: Generated images by capule HM on MNIST.

exhibit class-specific patterns and intra-class differences. Interestingly, "1" is the simplest digit, but can activate most capsules at the first layer and least capsules at the second layer. Furthermore, similar classes (such as "7" and "9") could exhibit similar spectra. These observations corroborate that the expressions of capsules are indeed context-dependent. We also investigated capsule activities on the Fashion-MNIST data. Similar results were observed (see Fig. 10b).



(a) For MNIST images. (b) For Fashion-MNIST images.

Fig. 10: Capsule activities in capsule HM for MNIST images in Fig. 2a and Fashion-MNIST images in Fig. 5a. Each cell in each panel has two rows corresponding to capsule spectra in the 1st (bottom) and 2nd (up) hidden layers.

4 Conclusions & Discussion

This study explored the use of capsules in shallow and deep generative models. We designed capsule RBMs and capsule HMs by replacing individual hidden units with stochastic capsules and devised new learning algorithms for them. Our empirical results showed that the models are capable of dynamically activating capsules depending on context, and the activity of capsules offers a new way to

interpret representations of observations in the hidden space. Furthermore, we find that Gibbs sampling in capsule RBMs and capsule HMs is able to generate samples of good quality. The source code of our implementation is available at `https://github.com/yifeng-li/cdgm`.

Our current capsule RBM model is accidentally similar to spike-and-slab RBM (ssRBM) [3] which can be viewed as a special case of ours, while the motivations are quite different. The design of ssRBM is motivated by the deficiency of Gaussian-Bernoulli RBM, while our development of capsule RBM is inspired by deterministic capsule nets. In our current design, a collection of active capsules at layer $l+1$ could interact with a capsule at layer l. It would be interesting to enforce a tree structure in the top-down activation path. It can be realized by assigning an activation vector to each capsule to indicate which capsules at the higher layer could activate the current capsule. This vector should follow multinoulli or multinomial distribution, so that similar capsules at layer l could dynamically form clusters for parent capsules at layer $l+1$. This thus models transformation in DGMs, in contrast to discriminative capsules introduced in [20,8]. The benefit of such a modification is beyond interpretation of the hidden representations in a dynamic hierarchical activation tree, because more importantly it also offers to generate better samples due to the modelling of transformation.

Our ideas will be tested on more complex data. Sophisticated design of the input layer may be necessary for spatially or temporally correlated data. In the case of large images, stochastic convolution and deconvolution components shall be used as in transforming auto-encoder [9]. We devised a tied wake-sleep algorithm to learn the model parameters in capsule HM, and obtained promising results. It is possible to develop better learning algorithms for capsule DGMs in inspiration of recent ideas for DGMs, such as weighted wake-sleep algorithm [2], neural variational inference and learning algorithm [18], and variational autoencoder based algorithms [19,13]. Reconstruction error was used in this paper to monitor the learning progress. The variational lower bound of likelihood can be computed using Eq. 22 for comparing various capsule DGMs. However, it may be time-consuming because the log-partition function of the joint distribution of each intermediate layer needs to be estimated by using annealed importance sampling methods for exponential family undirected models [21,17].

References

1. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of Machine Learning Research **2**, 1137–1155 (2003)
2. Bornschein, J., Bengio, Y.: Reweighted wake-sleep. In: International Conference on Learning Representations (2015)
3. Courville, A., Bergstra, J., Bengio, Y.: A spike and slab restricted Boltzmann machine. In: International Conference on Artificial Intelligence and Statistics. pp. 233–241 (2011)
4. Dayan, P., Hinton, G., Neal, R., Zemel, R.: The Helmholtz machine. Neural Computation **7**, 1022–1037 (1995)
5. Fodor, J., Pylyshyn, Z.: Connectionism and cognitive architecture: A critical analysis. Cognition **28**(1-2), 3–71 (1988)

6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
7. Hinton, G.: Aetherial symbols. In: AAAI Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches (2015)
8. Hinton, G., Sabour, S., Frosst, N.: Matrix capsules with EM routing. In: International Conferences on Learning Representations (2018)
9. Hinton, G., Krizhevsky, A., Wang, S.: Transforming auto-encoder. In: International Conference on Artificial Neural Networks. pp. 44–51 (2011)
10. Hinton, G., McClelland, J., Rumelhart, D.: Distributed representations. In: Rumelhart, D., McClelland, J. (eds.) Parallel Distributed Processing: Explorations in The Microstructure of Cognition, pp. 77–109. MIT Press, Cambridge, MA (1986)
11. Hinton, G., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Computation **18**, 1527–1554 (2006)
12. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine **29**(6), 82–97 (2012)
13. Kingma, D., Welling, M.: Auto-encoding variational Bayes. In: International Conference on Learning Representations (2014)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
15. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (1989)
16. Li, Y., Zhu, X.: Exploring Helmholtz machine and deep belief net in the exponential family perspective. In: ICML 2018 Workshop on Theoretical Foundations and Applications of Deep Generative Models (2018)
17. Li, Y., Zhu, X.: Exponential family restricted Boltzmann machines and annealed importance sampling. In: International Joint Conference on Neural Networks. pp. 39–48 (2018)
18. Mnih, A., Gregor, K.: Neural variational inference and learning in belief networks. In: International Conference on Machine Learning. pp. II–1791–II–1799 (2014)
19. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: International Conference on Machine Learning. pp. II–1278–II–1286 (2014)
20. Sabour, S., Frosst, N., Hinton, G.: Dynamic routing between capsules. In: Neural Information Processing Systems. pp. 3856–3866 (2017)
21. Salakhutdinov, R.: Learning and evaluating Boltzmann machines. Tech. rep., Department of Computer Science, University of Toronto, Toronto, Canada (2008)
22. Tieleman, T.: Training restricted Boltzmann machines using approximations to the likelihood gradient. In: International Conference on Machine Learning. pp. 1064–1071 (2008)
23. Welling, M., Rosen-Zvi, M., Hinton, G.: Exponential family harmoniums with an application to information retrieval. In: Advances in Neural Information Processing Systems. pp. 1481–1488 (2005)
24. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms (2017)
25. Zhao, S., Song, J., Ermon, S.: Towards a deeper understanding of variational autoencoding models. In: arXiv:1702.08658 (2017)