Learning Deep Sigmoid Belief Networks with Data Augmentation

Zhe Gan **Ricardo Henao** **David Carlson** **Lawrence Carin**
Department of Electrical and Computer Engineering, Duke University, Durham NC 27708, USA

Abstract

Deep directed generative models are developed. The multi-layered model is designed by stacking sigmoid belief networks, with sparsity-encouraging priors placed on the model parameters. Learning and inference of layer-wise model parameters are implemented in a Bayesian setting. By exploring the idea of data augmentation and introducing auxiliary Pólya-Gamma variables, simple and efficient Gibbs sampling and mean-field variational Bayes (VB) inference are implemented. To address large-scale datasets, an online version of VB is also developed. Experimental results are presented for three publicly available datasets: MNIST, Caltech 101 Silhouettes and OCR letters.

1 Introduction

The Deep Belief Network (DBN) (Hinton et al., 2006) and Deep Boltzmann Machine (DBM) (Salakhutdinov and Hinton, 2009) are two popular deep probabilistic generative models that provide state-of-the-art results in many problems. These models contain many layers of hidden variables, and utilize an *undirected* graphical model called the Restricted Boltzmann Machine (RBM) (Hinton, 2002) as the building block. A nice property of the RBM is that gradient estimates on the model parameters are relatively quick to calculate, and stochastic gradient descent provides relatively efficient inference. However, evaluating the probability of a data point under an RBM is nontrivial due to the computationally intractable partition function, which has to be estimated, for example using an annealed importance sampling algorithm (Salakhutdinov and Murray, 2008).

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

A *directed* graphical model that is closely related to these models is the Sigmoid Belief Network (SBN) (Neal, 1992). The SBN has a fully generative process and data are readily generated from the model using ancestral sampling. However, it has been noted that training a deep directed generative model is difficult, due to the "explaining away" effect. Hinton et al. (2006) tackle this problem by introducing the idea of "complementary priors" and show that the RBM provides a good initialization to the DBN, which has the same generative model as the SBN for all layers except the two top hidden layers. In the work presented here we directly deal with training and inference in SBNs (without RBM initialization), using recently developed methods in the Bayesian statistics literature.

Previous work on SBNs utilizes the ideas of Gibbs sampling (Neal, 1992) and mean field approximations (Saul et al., 1996). Recent work focuses on extending the wake-sleep algorithm (Hinton et al., 1995) to training fast variational approximations for the SBN (Mnih and Gregor, 2014). However, almost all previous work assumes no prior on the model parameters which connect different layers. An exception is the work of Kingma and Welling (2013), but this is mentioned as an extension of their primary work. Previous Gibbs sampling and variational inference procedures are implemented only on the hidden variables, while gradient ascent is employed to learn good model parameter values. The typical regularization on the model parameters is early stopping and/or L^2 regularization. In an SBN, the model parameters are not straightforwardly locally conjugate, and therefore fully Bayesian inference has been difficult.

The work presented here provides a method for placing priors on the model parameters, and presents a simple Gibbs sampling algorithm, by extending recent work on data augmentation for Bayesian logistic regression (Polson et al., 2013). More specifically, a set of Pólya-Gamma variables are used for each observation, to reformulate the logistic likelihood as a scale mixture, where each mixture component is conditionally normal with respect to the model parameters. Efficient mean-field variational learning and inference are

also developed, to optimize a data-augmented variational lower bound; this approach can be scaled up to large datasets. Utilizing these methods, sparsity-encouraging priors are placed on the model parameters and the posterior distribution of model parameters is estimated (not simply a point estimate). Based on extensive experiments, we provide a detailed analysis of the performance of the proposed method.

2 Model formulation

2.1 Sigmoid Belief Networks

Deep directed generative models are considered for binary data, based on the Sigmoid Belief Network (SBN) (Neal, 1992) (using methods like those discussed in Salakhutdinov et al. (2013), the model may be readily extended to real-valued data). Assume we have N binary visible vectors, the nth of which is denoted $\boldsymbol{v}_n \in \{0,1\}^J$. An SBN is a Bayesian network that models each $\boldsymbol{v}_n$ in terms of binary hidden variables $\boldsymbol{h}_n \in \{0,1\}^K$ and weights $\mathbf{W} \in \mathbb{R}^{J\times K}$ as

$$p(v_{jn}=1|\boldsymbol{w}_j,\boldsymbol{h}_n,c_j)=\sigma(\boldsymbol{w}_j^\top \boldsymbol{h}_n+c_j)\,, \qquad (1)$$

$$p(h_{kn}=1|b_k)=\sigma(b_k)\,, \qquad (2)$$

where $\sigma(\cdot)$ is the logistic function defined as $\sigma(x)=1/(1+\exp(-x))$, $\boldsymbol{v}_n=[v_{1n},\ldots,v_{Jn}]^\top$, $\boldsymbol{h}_n=[h_{1n},\ldots,h_{Kn}]^\top$, $\mathbf{W}=[\boldsymbol{w}_1,\ldots,\boldsymbol{w}_J]^\top$, $\boldsymbol{c}=[c_1,\ldots,c_J]^\top$ and $\boldsymbol{b}=[b_1,\ldots,b_K]^\top$ are bias terms. The "local" latent vector $\boldsymbol{h}_n$ is observation-dependent (a function of n), while the "global" parameters $\mathbf{W}$ are used to characterize the mapping from $\boldsymbol{h}_n$ to $\boldsymbol{v}_n$ for all n.

The SBN is closely related to the RBM, which is a Markov random field with the same bipartite structure as the SBN. Specifically, the energy function of an RBM is defined as

$$-E(\boldsymbol{v}_n,\boldsymbol{h}_n)=\boldsymbol{v}_n^\top\boldsymbol{c}+\boldsymbol{v}_n^\top\mathbf{W}\boldsymbol{h}_n+\boldsymbol{h}_n^\top\boldsymbol{b}\,, \qquad (3)$$

and the probability of an observation $\boldsymbol{v}_n$ is

$$p(\boldsymbol{v}_n)=\frac{1}{Z}\sum_{\boldsymbol{h}_n}\exp(-E(\boldsymbol{v}_n,\boldsymbol{h}_n))\,, \qquad (4)$$

where Z is a computationally intractable partition function that guarantees $p(\boldsymbol{v}_n)$ is a valid probability distribution. In contrast, the energy function of an SBN may be written as

$$-E(\boldsymbol{v}_n,\boldsymbol{h}_n)=\boldsymbol{v}_n^\top\boldsymbol{c}+\boldsymbol{v}_n^\top\mathbf{W}\boldsymbol{h}_n+\boldsymbol{h}_n^\top\boldsymbol{b} -\sum_j\log(1+\exp(\boldsymbol{w}_j^\top\boldsymbol{h}_n+c_j))\,. \qquad (5)$$

The additional term in (5), when compared to (3), makes the energy function no longer a linear function of weights $\mathbf{W}$, but a simple partition function is obtained. Therefore, the full likelihood under an SBN is trivial to calculate. Furthermore, SBNs explicitly exhibit the generative process to obtain data, in which the hidden layer provides a directed "explanation" for patterns generated in the visible layer.

2.2 Autoregressive Structure

Instead of assuming that the visible variables in an SBN are conditionally independent given the hidden units, a more flexible model can be built by using an autoregressive structure. The autoregressive sigmoid belief network (ARSBN) (Gregor et al., 2014) is an SBN with within-layer dependency captured by a fully connected directed acyclic graph, where each unit x_j can be predicted by its parent units $\boldsymbol{x}_{<j}$, defined as $\{x_1,\ldots,x_{j-1}\}$. To be specific,

$$p(v_{jn}=1|\boldsymbol{h}_n,\boldsymbol{v}_{<j,n})=\sigma(\boldsymbol{w}_j^\top\boldsymbol{h}_n+\boldsymbol{s}_{j,<j}^\top\boldsymbol{v}_{<j,n}+c_j)\,,$$

$$p(h_{kn}=1|\boldsymbol{h}_{<k,n})=\sigma(\boldsymbol{u}_{k,<k}^\top\boldsymbol{h}_{<k,n}+b_k)\,, \qquad (6)$$

where $\mathbf{S}=[\boldsymbol{s}_1,\ldots,\boldsymbol{s}_J]^\top$ and $\mathbf{U}=[\boldsymbol{u}_1,\ldots,\boldsymbol{u}_K]^\top$ are a lower triangular matrix that contains the autoregressive weights within layers, while $\mathbf{W}$ is utilized to capture the dependencies between different layers. The graphical model is provided in Supplemental Section A. If no hidden layer exists, we obtain the fully visible sigmoid belief network (Frey, 1998), in which accurate probabilities of test data points can be calculated.

In the work presented here, only stochastic autoregressive layers are considered, while Gregor et al. (2014) further explore the utilization of deterministic hidden layers. Furthermore, instead of using the simple linear autoregressive structure, one can increase the representational power of the model by using more-complicated autoregressive models, such as the work by Larochelle and Murray (2011), where each conditional $p(v_{jn}|\boldsymbol{v}_{<j,n})$ is modeled by a neural network.

2.3 Deep Sigmoid Belief Networks

Similar to the way in which deep belief networks and deep Boltzmann machines build hierarchies, one can stack additional hidden layers to obtain a fully directed deep sigmoid belief network (DSBN). Consider a deep model with L layers of hidden variables. To generate a sample, we begin at the top, layer L. For each layer below, activation $\boldsymbol{h}^{(l)}$ is formed by a sigmoid transformation of the layer above $\boldsymbol{h}^{(l+1)}$ weighted by $\mathbf{W}^{(l+1)}$. We repeat this process until the observation is reached. Therefore, the complete generative model can be written as

$$p(\boldsymbol{v}_n,\boldsymbol{h}_n)=p(\boldsymbol{v}_n|\boldsymbol{h}_n^{(1)})p(\boldsymbol{h}_n^{(L)})\prod_{l=1}^{L-1}p(\boldsymbol{h}_n^{(l)}|\boldsymbol{h}_n^{(l+1)})\,. \qquad (7)$$

Let $\{h_{1n}^{(l)}, h_{2n}^{(l)}, \ldots, h_{K_l n}^{(l)}\}$ represent the set of hidden units for observation n in layer l. For the top layer, the prior probability can be written as $p(h_{kn}^{(L)} = 1) = \sigma(c_k^{(L+1)})$, where $c_k^{(L+1)} \in \mathbb{R}$. Defining $\boldsymbol{v}_n = \boldsymbol{h}_n^{(0)}$, conditioned on the hidden units $\boldsymbol{h}_n^{(l)}$, the hidden units at layer $l-1$ are drawn from

$$p(h_{kn}^{(l-1)}|\boldsymbol{h}_n^{(l)}) = \sigma((\boldsymbol{w}_k^{(l)})^\top \boldsymbol{h}_n^{(l)} + c_k^{(l)}), \qquad (8)$$

where $\mathbf{W}^{(l)} = [\boldsymbol{w}_1^{(l)}, \ldots, \boldsymbol{w}_{K_{l-1}}^{(l)}]^\top$ connects layers l and $l-1$ and $\boldsymbol{c}^{(l)} = [c_1^{(l)}, \ldots, c_{K_{l-1}}^{(l)}]^\top$ is the bias term.

2.4 Bayesian sparsity shrinkage prior

The learned features are often expected to be sparse. In imagery, for example, features learned at the bottom layer tend to be localized, oriented edge filters which are similar to the Gabor functions known to model V1 cell receptive fields (Lee et al., 2008).

Under the Bayesian framework, sparsity-encouraging priors can be specified in a principled way. Some canonical examples are the spike-and-slab prior, the Student's-t prior, the double exponential prior and the horseshoe prior (see Polson and Scott (2012) for a discussion of these priors). The three parameter beta normal (TPBN) prior (Armagan et al., 2011), a typical global-local shrinkage prior, has demonstrated better (mixing) performance than the aforementioned priors, and thus is employed in this paper. The TPBN shrinkage prior can be expressed as scale mixtures of normals. If $W_{jk} \sim \text{TPBN}(a, b, \phi)$, where $j = 1, \ldots, J, k = 1, \ldots, K$, (leaving off the dependence on the layer l, for notational convenience) then

$$\begin{aligned} W_{jk} &\sim N(0, \zeta_{jk}), \qquad (9)\\ \zeta_{jk} &\sim \text{Gamma}(a, \xi_{jk}), \quad \xi_{jk} \sim \text{Gamma}(b, \phi_k),\\ \phi_k &\sim \text{Gamma}(1/2, \omega), \quad \omega \sim \text{Gamma}(1/2, 1). \end{aligned}$$

When $a = b = \frac{1}{2}$, the TPBN recovers the horseshoe prior. For fixed values of a and b, decreasing ϕ encourages more support for stronger shrinkage. In high-dimensional settings, ϕ can be fixed at a reasonable value to reflect an appropriate expected sparsity rather than inferring it from data.

Finally, to build up the fully generative model, commonly used isotropic normal prior are imposed on the bias term $\boldsymbol{b}$ and $\boldsymbol{c}$, i.e. $\boldsymbol{b} \sim N(0, \nu_b \mathbf{I}_K), \boldsymbol{c} \sim N(0, \nu_c \mathbf{I}_J)$.

Note that when performing model learning, we truncate the number of hidden units at each layer at K, which may be viewed as an upper bound within the model on the number of units at each layer. With the aforementioned shrinkage on $\mathbf{W}$, the model has the capacity to infer the subset of units (possibly less than K) actually needed to represent the data.

3 Learning and inference

In this section, Gibbs sampling and mean field variational inference are derived for the sigmoid belief networks, based on data augmentation. From the perspective of learning, we desire distributions on the model parameters $\{\mathbf{W}^{(l)}\}$ and $\{\boldsymbol{c}^{(l)}\}$, and distributions on the data-dependent $\{\boldsymbol{h}_n^{(l)}\}$ are desired in the context of inference. The extension to ARSBN is straightforward and provided in Supplemental Section C. We again omit the layer index l in the discussion below.

3.1 Gibbs sampling

Define $\mathbf{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N]$ and $\mathbf{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N]$. From recent work for the Pólya-Gamma data augmentation strategy (Polson et al., 2013), that is, if $\gamma \sim \text{PG}(b, 0)$, $b > 0$, then

$$\frac{(e^\psi)^a}{(1+e^\psi)^b} = 2^{-b} e^{\kappa\psi} \int_0^\infty e^{-\gamma\psi^2/2} p(\gamma) d\gamma, \qquad (10)$$

where $\kappa = a - b/2$. Properties of the Pólya-Gamma variables are summarized in Supplemental Section B. Therefore, the data-augmented joint posterior of the SBN model can be expressed as

$$\begin{aligned} &p(\mathbf{W}, \mathbf{H}, \boldsymbol{b}, \boldsymbol{c}, \gamma^{(0)}, \gamma^{(1)}|\mathbf{V}) \qquad (11)\\ &\propto \exp\left\{\sum_{j,n}(v_{jn} - \frac{1}{2})(\boldsymbol{w}_j^\top \boldsymbol{h}_n + c_j) - \frac{1}{2}\gamma_{jn}^{(0)}(\boldsymbol{w}_j^\top \boldsymbol{h}_n + c_j)^2\right\}\\ &\cdot \exp\left\{\sum_{k,n}(h_{kn} - \frac{1}{2})b_k - \frac{1}{2}\gamma_k^{(1)} b_k^2\right\} \cdot p_0(\gamma^{(0)}, \gamma^{(1)}, \mathbf{W}, \boldsymbol{b}, \boldsymbol{c}), \end{aligned}$$

where $\gamma^{(0)} \in \mathbb{R}^{J\times N}$ and $\gamma^{(1)} \in \mathbb{R}^K$ are augmented random variables drawn from the Pólya-Gamma distribution. The term $p_0(\gamma^{(0)}, \gamma^{(1)}, \mathbf{W}, \boldsymbol{b}, \boldsymbol{c})$ contains the prior information of the random variables within. Let $p(\cdot|-)$ represent the conditional distribution given other parameters fixed, then the conditional distributions used in the Gibbs sampling are as follows.

For $\gamma^{(0)}, \gamma^{(1)}$: The conditional distribution of $\gamma^{(0)}$ is

$$\begin{aligned} p(\gamma_{jn}^{(0)}|-) &\propto \exp\left(-\frac{1}{2}\gamma_{jn}^{(0)}(\boldsymbol{w}_j^\top \boldsymbol{h}_n + c_j)^2\right) \cdot \text{PG}(\gamma_{jn}^{(0)}|1, 0)\\ &= \text{PG}(1, \boldsymbol{w}_j^\top \boldsymbol{h}_n + c_j), \qquad (12) \end{aligned}$$

where $\text{PG}(\cdot, \cdot)$ represents the Pólya-Gamma distribution. Similarly, we can obtain $p(\gamma_k^{(1)}|-) = \text{PG}(1, b_k)$.

To draw samples from the Pólya-Gamma distribution, two strategies are utilized: (i) using rejection sampling to draw samples from the closely related exponentially tilted Jacobi distribution (Polson et al., 2013); (ii) using a truncated sum of random variables from the Gamma distribution and then match the first moment

to keep the samples unbiased (Zhou et al., 2012). Typically, a truncation level of 20 works well in practice.

For H: The sequential update of the local conditional distribution of $\mathbf{H}$ is $p(h_{kn}|-) = \text{Ber}(\sigma(d_{kn}))$, where

$$d_{kn} = b_k + \boldsymbol{w}_k^\top \boldsymbol{v}_n - \frac{1}{2}\sum_{j=1}^{J}\left(w_{jk} + \gamma_{jn}^{(0)}(2\psi_{jn}^{\setminus k} w_{jk} + w_{jk}^2)\right),$$

where $\psi_{jn}^{\setminus k} = \boldsymbol{w}_j^\top \boldsymbol{h}_n - w_{jk}h_{kn} + c_j$. Note that $\boldsymbol{w}_k$ and $\boldsymbol{w}_j$ represent the kth column and the transpose of the jth row of $\mathbf{W}$, respectively. The difference between an SBN and an RBM can be seen more clearly from the sequential update. Specifically, in an RBM, the update of h_{kn} only contains the first two terms, which implies the update of the h_{kn} are independent of each other. In the SBN, the existence of the third term demonstrates clearly the posterior dependencies between hidden units. Although the rows of $\mathbf{H}$ are correlated, the columns of $\mathbf{H}$ are independent, therefore the sampling of $\mathbf{H}$ is still efficient.

For W: The prior is a TPBN shrinkage prior with $p_0(\boldsymbol{w}_j) = N(0, \text{diag}(\boldsymbol{\zeta}_j))$, then we have $p(\boldsymbol{w}_j|-) = N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, where

$$\boldsymbol{\Sigma}_j = \left[\sum_{n=1}^{N}\gamma_{jn}^{(0)}\boldsymbol{h}_n\boldsymbol{h}_n^\top + \text{diag}(\boldsymbol{\zeta}_j^{-1})\right]^{-1}, \tag{13}$$

$$\boldsymbol{\mu}_j = \boldsymbol{\Sigma}_j\left[\sum_{n=1}^{N}(v_{jn} - \frac{1}{2} - c_j\gamma_{jn}^{(0)})\boldsymbol{h}_n\right]. \tag{14}$$

The update of the bias term $\boldsymbol{b}$ and $\boldsymbol{c}$ are similar to the above equation.

For TPBN shrinkage: One advantage of this hierarchical shrinkage prior is the full local conjugacy that allows the Gibbs sampling easily implemented. Specifically, the following posterior conditional distribution can be achieved: (1) $\zeta_{jk}|- \sim \mathcal{GIG}(0, 2\xi_{jk}, W_{jk}^2)$; (2) $\xi_{jk}|- \sim \text{Gamma}(1, \zeta_{jk}+\phi_k)$; (3) $\phi_k|- \sim \text{Gamma}(\frac{1}{2}J + \frac{1}{2}, \omega + \sum_{j=1}^{J}\xi_{jk})$; (4) $\omega|- \sim \text{Gamma}(\frac{1}{2}K + \frac{1}{2}, 1 + \sum_{k=1}^{K}\phi_k)$, where $\mathcal{GIG}$ denotes the generalized inverse Gaussian distribution.

3.2 Mean field variational Bayes

Using the VB inference with the traditional mean field assumption, we approximate the posterior distribution with $Q = \prod_{j,k} q_{w_{jk}}(w_{jk}) \prod_{j,n} q_{h_{jn}}(h_{jn}) q_{\gamma_{jn}^{(0)}}(\gamma_{jn}^{(0)})$; for notational simplicity the terms concerning $\boldsymbol{b}, \boldsymbol{c}, \gamma^{(1)}$ and the parameters of the TPBN shrinkage prior are omitted. The variational lower bound can be obtained as

$$\mathcal{L} = \langle \log p(\mathbf{V}|\mathbf{W}, \mathbf{H}, \boldsymbol{c})\rangle + \langle \log p(\mathbf{W})\rangle + \langle \log p(\mathbf{H}|\boldsymbol{b})\rangle - \langle \log q(\mathbf{W})\rangle - \langle \log q(\mathbf{H})\rangle, \tag{15}$$

where $\langle\cdot\rangle$ represents the expectation w.r.t. the variational approximate posterior.

Note that $\langle \log p(\mathbf{V}|-)\rangle = \sum_{j,n}\langle \log p(v_{jn}|-)\rangle$, and each term inside the summation can be further lower bounded by using the augmented Pólya-Gamma variables. Specifically, defining $\psi_{jn} = \boldsymbol{w}_j^\top \boldsymbol{h}_n + c_j$, we can obtain

$$\langle \log p(v_{jn}|-)\rangle \geq -\log 2 + (v_{jn} - 1/2)\langle\psi_{jn}\rangle - \frac{1}{2}\langle\gamma_{jn}^{(0)}\rangle\langle\psi_{jn}^2\rangle + \langle \log p_0(\gamma_{jn}^{(0)})\rangle - \langle \log q(\gamma_{jn}^{(0)})\rangle, \tag{16}$$

by using (10) and Jensen's inequality. Therefore, the new lower bound $\mathcal{L}'$ can be achieved by substituting (16) into (15). Note that this is a looser lower bound compared with the original lower bound $\mathcal{L}$, due to the data augmentation. However, closed-form coordinate ascent update equations can be obtained, shown below.

For $\gamma^{(0)}, \gamma^{(1)}$: optimizing $\mathcal{L}'$ over $q(\gamma_{jn}^{(0)})$, we have

$$q(\gamma_{jn}^{(0)}) \propto \exp\left(-\frac{1}{2}\gamma_{jn}^{(0)}\langle\psi_{jn}^2\rangle\right)\cdot \text{PG}(\gamma_{jn}^{(0)}|1, 0) = \text{PG}\left(1, \sqrt{\langle\psi_{jn}^2\rangle}\right). \tag{17}$$

Similarly, we can obtain $p(\gamma_k^{(1)}|-) = \text{PG}(1, \sqrt{\langle b_k^2\rangle})$. In the update of other variational parameters, only the expectation of $\gamma_{jn}^{(0)}$ is needed, which can be calculated by $\langle\gamma_{jn}^{(0)}\rangle = \frac{1}{2\sqrt{\langle\psi_{jn}^2\rangle}}\tanh(\frac{\sqrt{\langle\psi_{jn}^2\rangle}}{2})$. The variational distribution for other parameters are in the exponential family, hence the update equations can be derived from the Gibbs sampling, which are straightforward and provided in Supplemental Section D.

In order to calculate the variational lower bound, the augmented Pólya-Gamma variables are integrated out, and the expectation of the logistic likelihood under the variational distribution is estimated by Monte Carlo integration algorithm. In the experiments 10 samples are used and were found sufficient in all cases considered.

The computational complexity of the above inference is $\mathcal{O}(NK^2)$, where N is the total number of training data points. Every iteration of VB requires a full pass through the dataset, which can be slow when applied to large datasets. Therefore, an online version of VB inference is developed, building upon the recent online implementation of latent Dirichlet allocation (Hoffman et al., 2013). In online VB, stochastic optimization is applied to the variational objective. The key observation is that the coordinate ascent updates in VB precisely correspond to the natural gradient of the variational objective. To implement online VB, we subsample the data, compute the gradient estimate based

on the subsamples and follow the gradient with a decreasing step size.

3.3 Learning deep networks using SBNs

Once one layer of the deep network is trained, the model parameters in that layer are frozen (at the mean of the inferred posterior) and we can utilize the inferred hidden units as the input "data" for the training of the next higher layer. This greedy layer-wise pre-training algorithm has been shown to be effective for DBN (Hinton et al., 2006) and DBM (Salakhutdinov and Hinton, 2009) models, and is guaranteed to improve the data likelihood under certain conditions. In the training of a deep SBN, the same strategy is employed. After finishing pre-training (sequentially for all the layers), we then "un-freeze" all the model parameters, and implement global training (refinement), in which parameters in the higher layer now can also affect the update of parameters in the lower layer.

Discriminative fine-tuning (Salakhutdinov and Hinton, 2009) is implemented in the training of DBM by using label information. In the work presented here for the SBN, we utilize label information (when available) in a multi-task learning setting (like in Bengio et al. (2013)), where the top-layer hidden units are generated by multiple sets of bias terms, one for each label, while all the other model parameters and hidden units below the top layer are shared. This multi-task learning is only performed when generating samples from the model.

4 Related work

The SBN was proposed by Neal (1992), and in the original paper a Gibbs sampler was proposed to do inference. A natural extension to a variational approximation algorithm was proposed by Saul et al. (1996), using the mean field assumption. A Gaussian-field (Barber and Sollich, 1999) approach was also used for inference, by making Gaussain approximations to the unit input. However, due to the fact that the model is not locally conjugate, all the methods mentioned above are only used for inference of distributions on the hidden variables $\mathbf{H}$, and typically model parameters $\mathbf{W}$ are learned by gradient descent.

Another route to do inference on SBNs are based on the idea of Helmholtz machines (Dayan et al., 1995), which are multi-layer belief networks with recognition models, or inference networks. These recognition models are used to approximate the true posterior. The wake-sleep algorithm (Hinton et al., 1995) was first proposed to do inference on such recognition models. Recent work focuses on training the recognition models by maximizing a variational lower bound on the marginal log likelihood (Mnih and Gregor, 2014; Gregor et al., 2014; Kingma and Welling, 2013; Rezende et al., 2014).

In the work reported here, we focus on providing a fully Bayesian treatment on the "global" model parameters and the "local" data-dependent hidden variables. An advantage of this approach is the ability to impose shrinkage-based (near) sparsity on the model parameters. This sparsity helps regularize the model, and also aids in interpreting the learned model. The idea of Pólya-Gamma data augmentation was first proposed to do inference on Bayesian logistic regression (Polson et al., 2013), and later extended to the inference of negative binomial regression (Zhou et al., 2012), logistic-normal topic models (Chen et al., 2013), and discriminative relational topic models (Chen et al., 2014). The work reported here serves as another application of this data augmentation strategy, and a first implementation of analysis of a deep-learning model in a fully Bayesian framework.

5 Experiments

We present experimental results on three publicly available binary datasets: MNIST, Caltech 101 Silhouettes, and OCR letters. To assess the performance of SBNs trained using the proposed method, we show the samples generated from the model and report the average log probability that the model assigns to a test datum.

5.1 Experiment setup

For all the experiments below, we consider a one-hidden-layer SBN with $K = 200$ hidden units, and a two-hidden-layer SBN with each layer containing $K = 200$ hidden units. The autoregressive version of the model is denoted ARSBN. The fully visible sigmoid belief network without any hidden units is denoted FVSBN.

The SBN model is trained using both Gibbs sampling and mean field VB, as well as the proposed online VB method. The learning and inference discussed above is almost free of parameter tuning; the hyperparameters settings are given in Section 2. Similar reasonable settings on the hyperparameters yield essentially identical results. The hidden units are initialized randomly and the model parameters are initialized using an isotropic normal with standard deviation 0.1. The maximum number of iterations for VB inference is set to 40, which is large enough to observe convergence. Gibbs sampling used 40 burn-in samples and 100 posterior collection samples; while this number of samples

Figure 1: Performance on the MNIST dataset. (Left) Training data. (Middle) Averaged synthesized samples. (Right) Learned features at the bottom layer.

is clearly too small to yield sufficient mixing and an accurate representation of the posteriors, it yields effective approximations to parameter means, which are used when presenting results. For online VB, the mini-batch size is set to 5000 with a fixed learning rate of 0.1. Local parameters were updated using 4 iterations per mini-batch, and results are shown over 20 epochs.

The properties of the deep model were explored by examining $\mathbb{E}_{p(\boldsymbol{v}|\boldsymbol{h}^{(2)})}[\boldsymbol{v}]$. Given the second hidden layer, the mean was estimated by using Monte Carlo integration. Given $\boldsymbol{h}^{(2)}$, we sample $\boldsymbol{h}^{(1)} \sim p(\boldsymbol{h}^{(1)}|\boldsymbol{h}^{(2)})$ and $\boldsymbol{v} \sim p(\boldsymbol{v}|\boldsymbol{h}^{(1)})$, repeat this procedure 1000 times to obtain the final *averaged* synthesized samples.

The test data log probabilities under VB inference are estimated using the variational lower bound. Evaluating the log probability using the Gibbs output is difficult. For simplicity, the harmonic mean estimator is utilized. As the estimator is biased (Wallach et al., 2009), we refer to the estimate as an upper bound.

ARSBN requires that the observation variables are put in some fixed order. In the experiments, the ordering was simply determined by randomly shuffling the observation vectors, and no optimization of the ordering was tried. Repeated trials with different random orderings gave empirically similar results.

5.2 Binarized MNIST dataset

We conducted the first experiment on the MNIST digit dataset which contains $60,000$ training and $10,000$ test images of ten handwritten digits (0 to 9), with 28×28 pixels. The binarized version of the dataset is used according to (Murray and Salakhutdinov, 2009). Analysis was performed on $10,000$ randomly selected training images for Gibbs and VB inference. We also examine the online VB on the whole training set.

Table 1: Log probability of test data on MNIST dataset. "Dim" represents the number of hidden units in each layer, starting with the bottom one. ($*$) taken from Salakhutdinov and Murray (2008), ($\diamond$) taken from Mnih and Gregor (2014), ($\triangleright$) taken from Salakhutdinov and Hinton (2009). SBN.multi denotes SBN trained in the multi-task learning setting.

Model	Dim	Test log-prob.
SBN (online VB)	25	-138.34
RBM* (CD3)	25	-143.20
SBN (online VB)	200	-118.12
SBN (VB)	200	-116.96
SBN.multi (VB)	200	-113.02
SBN.multi (VB)	$200-200$	-110.74
FVSBN (VB)	$-$	-100.76
ARSBN (VB)	200	-102.11
ARSBN (VB)	$200-200$	-101.19
SBN (Gibbs)	200	-94.30
SBN$^\diamond$ (NVIL)	200	-113.1
SBN$^\diamond$ (NVIL)	$200-200$	-99.8
DBN*	$500-2000$	-86.22
DBM$^\triangleright$	$500-1000$	-84.62

The results for MNIST, along with baselines from the literature are shown in Table 1. We report the log probability estimates from our implementation of Gibbs sampling, VB and online VB using both the SBN and ARSBN model.

First, we examine the performance in a low-dimensional model, with $K = 25$, and the results are shown in Table 1. All VB methods give similar results, so only the result from the online method is shown for brevity. VB SBN shows improved performance over an RBM in this size model (Salakhutdinov and Murray, 2008).

Next, we explore an SBN with $K = 200$ hidden units. Our methods achieve similar performance to the Neural Varitional Inference and Learning (NVIL) algorithm (Mnih and Gregor, 2014), which is the current

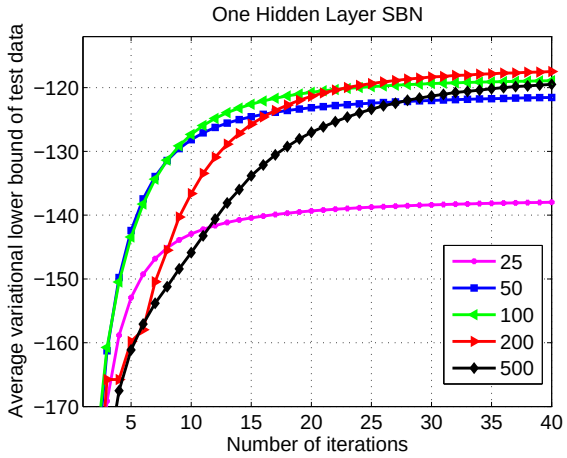

Figure 2: The impact of the number of hidden units on the average variational lower bound of test data under the one-hidden-layer SBN

state of the art for training SBNs.

Using a second hidden layer, also with size 200, gives performance improvements in all algorithms. In VB there is an improvement of 3 nats for the SBN model when a second layer is learned. Furthermore, the VB ARSBN method gives a test log probability of -101.19. The current state of the art on this size deep sigmoid belief network is the NVIL algorithm with -99.8, which is quantitatively similar to our results. The online VB implementation gives lower bounds comparable to the batch VB, and will scale better to larger data sizes.

The TPBN prior infers the number of units needed to represent the data. The impact on the number of hidden units on the test set performance is shown in Figure 2. The models learned using 100, 200 and 500 hidden units achieve nearly identical test set performance, showing that our methods are not overfitting the data as the number of units increase. All models with $K > 100$ typically utilize 81 features. Thus, the TPBN prior gives "tuning-free" selection on the hidden layer size K. The learned features are shown in Figure 1. These features are sparse and consistent with results from sparse features learning algorithms (Lee et al., 2008).

The generated samples for MNIST are presented in Figure 1. The synthesized digits appear visually good and match the true data well.

We further demonstrate the ability of the model to predict missing data. For each test image, the lower half of the digit is removed and considered as missing data. Reconstructions are shown in Figure 3, and the model produces good completions. Because the labels of the images are uncertain when they are partially

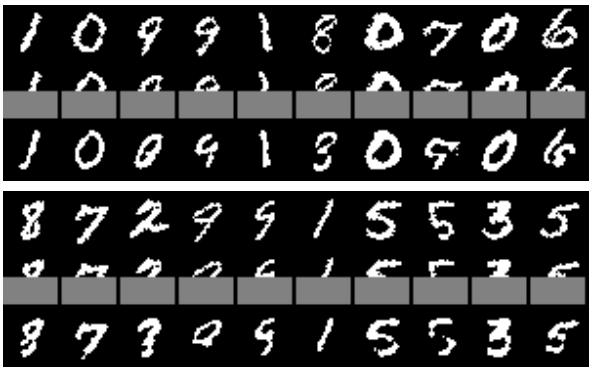

Figure 3: Missing data prediction. For each subfigure, (Top) Original data. (Middle) Hollowed region. (Bottom) Reconstructed data.

observed, the model can generate different digits than the true digit (see the transition from 9 to 0, 7 to 9 etc.).

5.3 Caltech 101 Silhouettes dataset

The second experiment is based on the Caltech 101 Silhouettes dataset (Marlin et al., 2010), which contains 6364 training images and 2307 test images. Estimated log probabilities are reported in Table 2.

Table 2: Log probability of test data on Caltech 101 Silhouettes dataset. "Dim" represents the number of hidden units in each layer, starting with the bottom one. $(*)$ taken from Cho et al. (2013).

Model	Dim	Test log-prob.
SBN (VB)	200	-136.84
SBN (VB)	$200-200$	-125.60
FVSBN (VB)	–	-96.40
ARSBN (VB)	200	-96.78
ARSBN (VB)	$200-200$	-97.57
RBM*	500	-114.75
RBM*	4000	-107.78

In this dataset, adding the second hidden layer to the VB SBN greatly improves the lower bound. Figure 5 demonstrates the effect of the deep model on learning. The first hidden layer improves the lower bound quickly, but saturates. When the second hidden layer is added, the model once again improves the lower bound on the test set. Global training (refinement) further enhances the performance. The two-layer model does a better job capturing the rich structure in the 101 total categories. For the simple dataset (MNIST with 10 categories, OCR letters with 26 categories, discussed below), this large gap is not observed.

Remarkably, our implementation of FVSBN beats the state-of-the-art results on this dataset (Cho et al., 2013) by 10 nats. Figure 4 shows samples drawn from

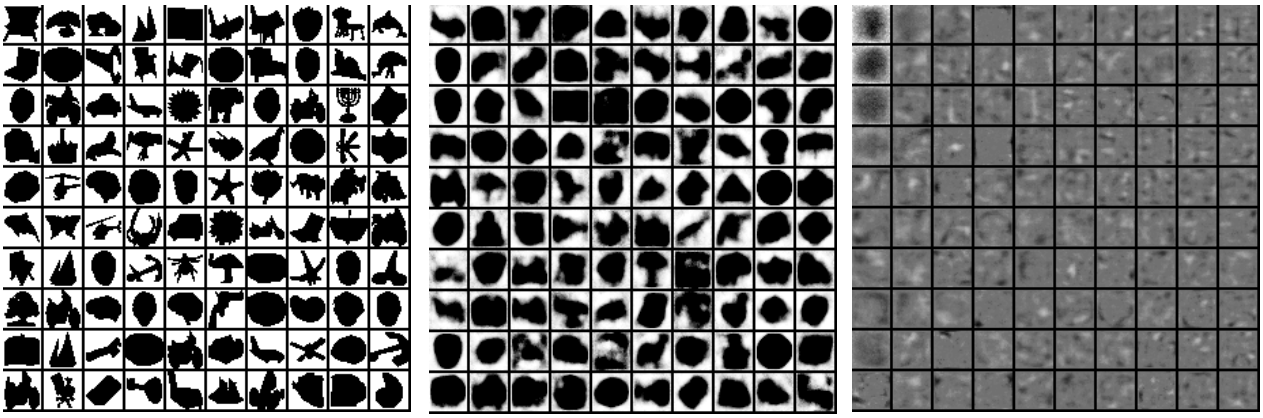

Figure 4: Performance on the Caltech 101 Silhouettes dataset. (Left) Training data. (Middle) Averaged synthesized samples. (Right) Learned features at the bottom layer.

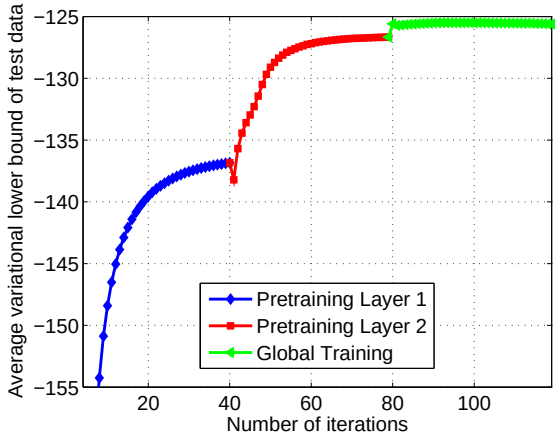

Figure 5: Average variational lower bound obtained from the SBN $200 - 200$ model on the Caltech 101 Silhouettes dataset.

the trained model; different shapes are synthesized and appear visually good.

5.4 OCR letters dataset

The OCR letters dataset contains 16×8 binary pixel images of 26 letters in the English alphabet. The dataset is split into $42,152$ training and $10,000$ test examples. Results are reported in Table 3. The proposed ARSBN with $K = 200$ hidden units achieves a lower bound of -37.97. The state-of-the-art here is a DBM with 2000 hidden units in each layer (Salakhutdinov and Larochelle, 2010). Our model gives results that are only marginally worse using a network with 100 times fewer connections.

Table 3: Log probability of test data on OCR letters dataset. "Dim" represents the number of hidden units in each layer, starting with the bottom one. $(*)$ taken from Salakhutdinov and Larochelle (2010).

Model	Dim	Test log-prob.
SBN (online VB)	200	-48.71
SBN (VB)	200	-48.20
SBN (VB)	$200 - 200$	-47.84
FVSBN (VB)	$-$	-39.71
ARSBN (VB)	200	-37.97
ARSBN (VB)	$200 - 200$	-38.56
SBN (Gibbs)	200	-40.95
DBM*	$2000 - 2000$	-34.24

6 Discussion and future work

A simple and efficient Gibbs sampling algorithm and mean field variational Bayes approximation are developed for learning and inference of model parameters in the sigmoid belief networks. This has been implemented in a novel way by introducing auxiliary Pólya-Gamma variables. Several encouraging experimental results have been presented, enhancing the idea that the deep learning problem can be efficiently tackled in a fully Bayesian framework.

While this work has focused on binary observations, one can model real-valued data by building latent binary hierarchies as employed here, and touching the data at the bottom layer by a real-valued mapping, as has been done in related RBM models (Salakhutdinov et al., 2013). Furthermore, the logistic link function is typically utilized in the deep learning literature. The probit function and the rectified linearity are also considered in the nonlinear Gaussian belief network (Frey and Hinton, 1999). Under the Bayesian framework, by using data augmentation (Polson et al., 2011), the max-margin link could be utilized to model the nonlinearities between layers when training a deep model.

Acknowledgements

This research was supported in part by ARO, DARPA, DOE, NGA and ONR.

References

A. Armagan, M. Clyde, and D. B. Dunson. Generalized beta mixtures of Gaussians. *NIPS*, 2011.

D. Barber and P. Sollich. Gaussian fields for approximate inference in layered sigmoid belief networks. *NIPS*, 1999.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *PAMI*, 2013.

J. Chen, J. Zhu, Z. Wang, X. Zheng, and B. Zhang. Scalable inference for logistic-normal topic models. *NIPS*, 2013.

N. Chen, J. Zhu, F. Xia, and B. Zhang. Discriminative relational topic models. *PAMI*, 2014.

K. Cho, T. Raiko, and A. Ilin. Enhanced gradient for training restricted Boltzmann machines. *Neural computation*, 2013.

P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz machine. *Neural computation*, 1995.

B. J. Frey. *Graphical models for machine learning and digital communication*. MIT press, 1998.

B. J. Frey and G. E. Hinton. Variational learning in nonlinear Gaussian belief networks. *Neural Computation*, 1999.

K. Gregor, A. Mnih, and D. Wierstra. Deep autoregressive networks. *ICML*, 2014.

G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 2006.

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

G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 1995.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *JMLR*, 2013.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv:1312.6114*, 2013.

H. Larochelle and I. Murray. The neural autoregressive distribution estimator. *JMLR*, 2011.

H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area v2. *NIPS*, 2008.

B. M. Marlin, K. Swersky, B. Chen, and N. D. Freitas. Inductive principles for restricted Boltzmann machine learning. *AISTATS*, 2010.

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

I. Murray and R. Salakhutdinov. Evaluating probabilities under high-dimensional latent variable models. *NIPS*, 2009.

R. M. Neal. Connectionist learning of belief networks. *Artificial intelligence*, 1992.

N. G. Polson and J. G. Scott. Local shrinkage rules, Lévy processes and regularized regression. *JRSS*, 2012.

N. G. Polson, S. L. Scott, et al. Data augmentation for support vector machines. *Bayesian Analysis*, 2011.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using Pólya–Gamma latent variables. *JASA*, 2013.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 2014.

R. Salakhutdinov and G. E. Hinton. Deep Boltzmann machines. *AISTATS*, 2009.

R. Salakhutdinov and H. Larochelle. Efficient learning of deep Boltzmann machines. *AISTATS*, 2010.

R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. *ICML*, 2008.

R. Salakhutdinov, J. B. Tenenbaum, and A. Torralba. Learning with hierarchical-deep models. *PAMI*, 2013.

L. K. Saul, T. Jaakkola, and M. I. Jordan. Mean field theory for sigmoid belief networks. *JAIR*, 1996.

H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. *ICML*, 2009.

M. Zhou, L. Li, D. Dunson, and L. Carin. Lognormal and gamma mixed negative binomial regression. *ICML*, 2012.