Learning Deep Sigmoid Belief Networks with Data Augmentation

Zhe Gan, Ricardo Henao, David Carlson and Lawrence Carin

Duke University, Durham NC 27708, USA

INTRODUCTION

Objective: Designing simple and efficient Bayesian inference algorithms for deep learning models.

Main idea:

- Deep directed generative models are developed by stacking sigmoid belief networks (SBN).
- Sparsity-encouraging priors are placed on model parameters.
- Learning and inference of layer-wise model parameters are implemented in a fully Bayesian setting, by exploring the idea of data augmentation.

MODEL FORMULATION

Sigmoid Belief Network: Assume we observe $\boldsymbol{v} \in \{0,1\}^J$, modeled using hidden variable $\boldsymbol{h} \in \{0,1\}^K$ and weights $\mathbf{W} \in \mathbb{R}^{J \times K}$ as

$$p(v_j = 1 | \boldsymbol{w}_j, \boldsymbol{h}, c_j) = \sigma(\boldsymbol{w}_j^\top \boldsymbol{h} + c_j)\,,$$
$$p(h_k = 1 | b_k) = \sigma(b_k)\,,$$

where $\sigma(\cdot)$ is the logistic function, $\boldsymbol{v} = [v_1 \ \ldots \ v_J]^\top$, $\mathbf{W} = [\boldsymbol{w}_1 \ \ldots \ \boldsymbol{w}_J]^\top$, and $\boldsymbol{c} = [c_1 \ \ldots \ c_J]^\top$ and $\boldsymbol{b} = [b_1 \ \ldots \ b_K]^\top$ are bias terms.

Relationship with RBM: The energy function of an SBN is:

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

(In contrast) The energy function of an RBM is:

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

Autoregressive Structure:

$$p(v_j = 1 | \boldsymbol{h}, \boldsymbol{v}_{<j}) = \sigma(\boldsymbol{w}_j^\top \boldsymbol{h} + \boldsymbol{s}_{j,<j}^\top \boldsymbol{v}_{<j} + c_j)\,,$$
$$p(h_k = 1 | \boldsymbol{h}_{<k}) = \sigma(\boldsymbol{u}_{k,<k}^\top \boldsymbol{h}_{<k} + b_k)\,.$$

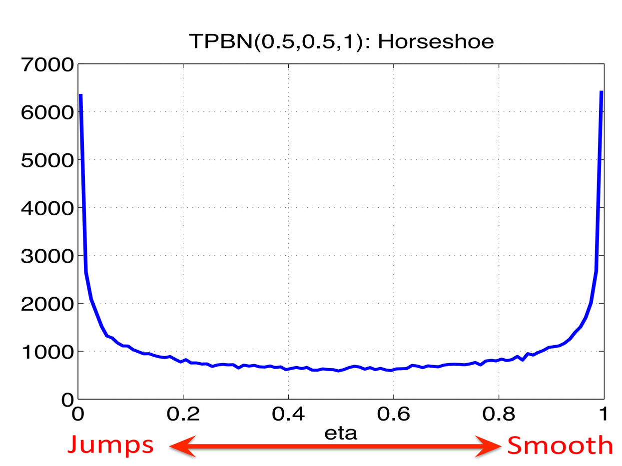
where $\mathbf{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_J]^\top$ and $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K]^\top$ are a lower triangular matrix that contains the autoregressive weights within layers.

Deep Sigmoid Belief Networks:

$$p(\boldsymbol{v}, \boldsymbol{h}) = p(\boldsymbol{v} | \boldsymbol{h}^{(1)}) p(\boldsymbol{h}^{(L)}) \prod_{\ell=1}^{L-1} p(\boldsymbol{h}^{(\ell)} | \boldsymbol{h}^{(\ell+1)})\,,$$
$$p(h_k^{(\ell-1)} | \boldsymbol{h}^{(\ell)}) = \sigma((\boldsymbol{w}_k^{(\ell)})^\top \boldsymbol{h}_n^{(\ell)} + c_k^{(\ell)})\,.$$

Bayesian Sparsity Shrinkage: Three Parameter Beta Normal ($\mathcal{TPBN}$) prior on $\mathbf{W}^{(\ell)}$

$$W_{jk}^{(\ell)} \sim \mathcal{N}(0, \zeta_{jk})\,,$$
$$\zeta_{jk} \sim \text{Gamma}(a, \xi_{jk})\,,$$
$$\xi_{jk} \sim \text{Gamma}(b, \phi_k)\,,$$
$$\phi_k \sim \text{Gamma}(1/2, \omega)\,,$$
$$\omega \sim \text{Gamma}(1/2, 1)\,.$$



Graphical Model

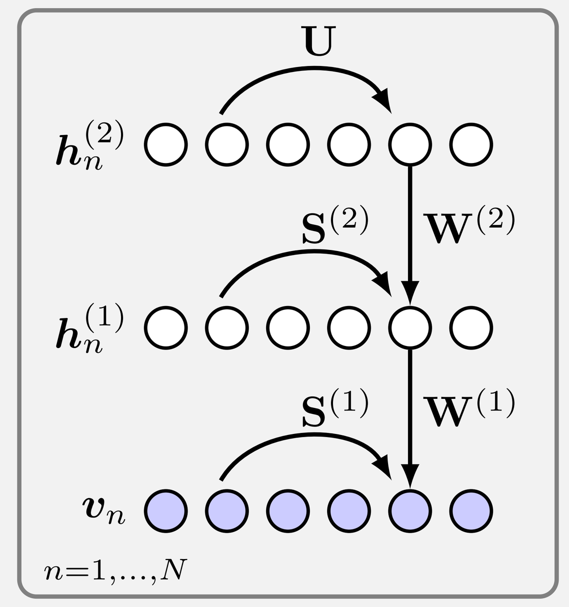


Figure: Graphical model for the deep SBN with autoregressive structure.

LEARNING & INFERENCE

Main idea: If $\gamma \sim \text{PG}(b, 0)$ for $b > 0$, then (PG denotes Pólya-Gamma distribution)

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

where $\kappa = a - b/2$ and $\gamma | \psi \sim \text{PG}(b, \psi)$.

Gibbs Sampling: We can write the likelihood function for $\mathbf{W}$ (omitting $\mathbf{c}$) and $\boldsymbol{h}$ as

$$L(\mathbf{W}, \boldsymbol{h}) \propto \exp\left\{\sum_j^J \left(v_j - \frac{1}{2}\right) \boldsymbol{w}_j^\top \boldsymbol{h} - \frac{1}{2} \gamma_j (\boldsymbol{h}^\top \boldsymbol{w}_j \boldsymbol{w}_j^\top \boldsymbol{h})\right\}$$
$$\propto \exp\left\{\left(\mathbf{v} - \frac{1}{2}\right)^\top \mathbf{W} \boldsymbol{h} - \frac{1}{2} \boldsymbol{h}^\top \mathbf{W}^\top \mathbf{\Gamma} \mathbf{W} \boldsymbol{h}\right\}\,,$$

where $\mathbf{\Gamma} = \text{diag}(\gamma_1, \ldots, \gamma_J)$. Hence, $L(\mathbf{W}, \boldsymbol{h})$ is conjugate to Gaussian prior $p(w_{jk})$; $\boldsymbol{h} | \boldsymbol{v} \sim \text{Bernoulli}(\cdot)$; $\gamma_j | \boldsymbol{w}_j, \boldsymbol{h} \sim \text{PG}(1, \boldsymbol{w}_j^\top \boldsymbol{h})$.

Mean-field VB: Define $\psi_j = \boldsymbol{w}_j^\top \boldsymbol{h}$, then

$$q(\gamma_j) \propto \exp\left(-\frac{1}{2} \gamma_j \langle \psi_j^2 \rangle\right) \cdot \text{PG}(\gamma_j | 1, 0) = \text{PG}\left(1, \sqrt{\langle \psi_j^2 \rangle}\right)\,.$$

EXPERIMENTS

I. MNIST: Generated Samples:

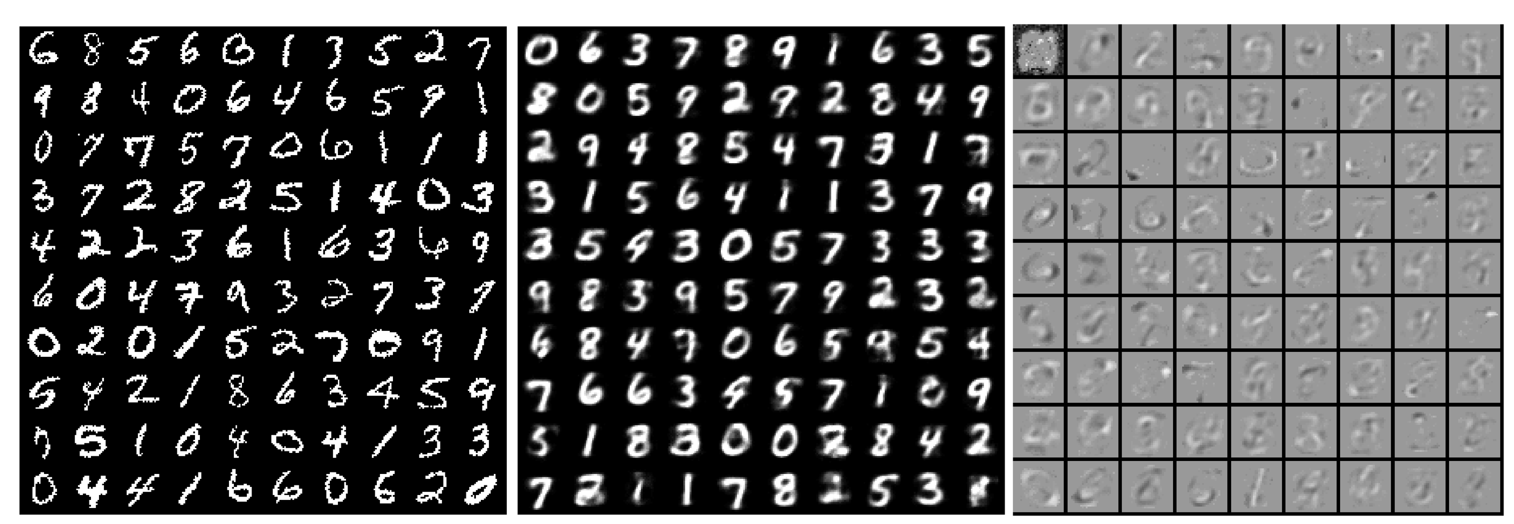


Figure: Performance on the MNIST dataset. (Left) Training data. (Middle) Averaged synthesized samples. (Right) Learned features at the bottom layer.

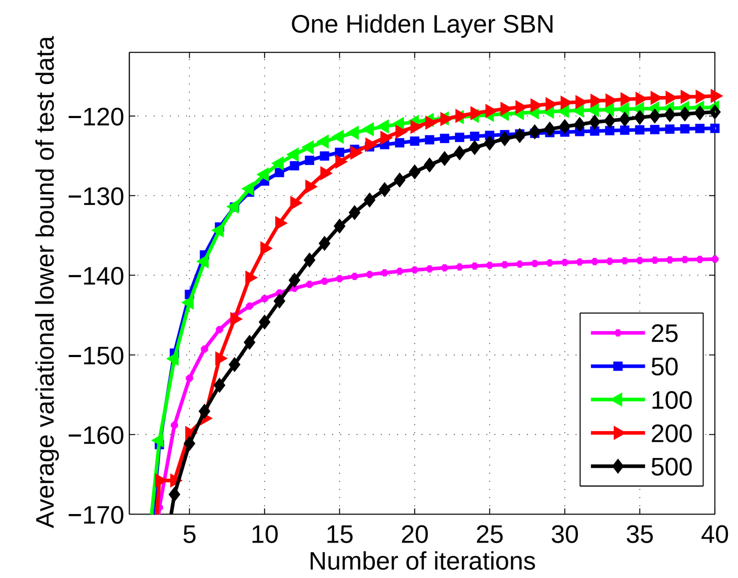


Figure: The impact of the number of hidden units on the lower bound.

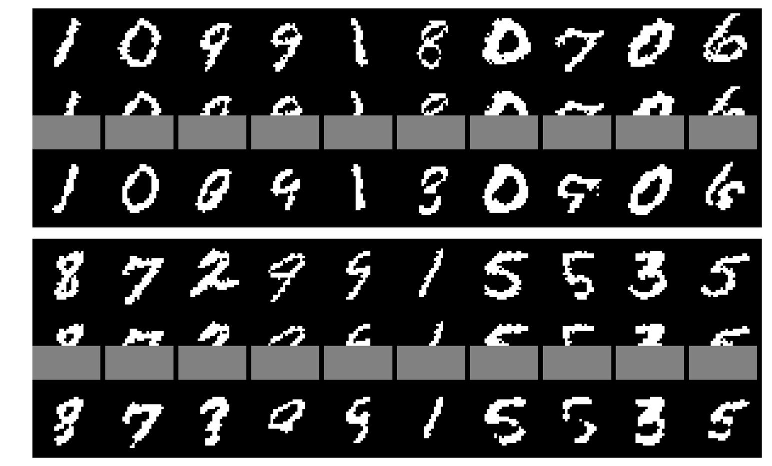


Figure: Missing data prediction. For each subfigure, (Top) Original data. (Middle) Hollowed region. (Bottom) Reconstructed data.

II. Caltech 101 Silhouettes: Generated Samples:

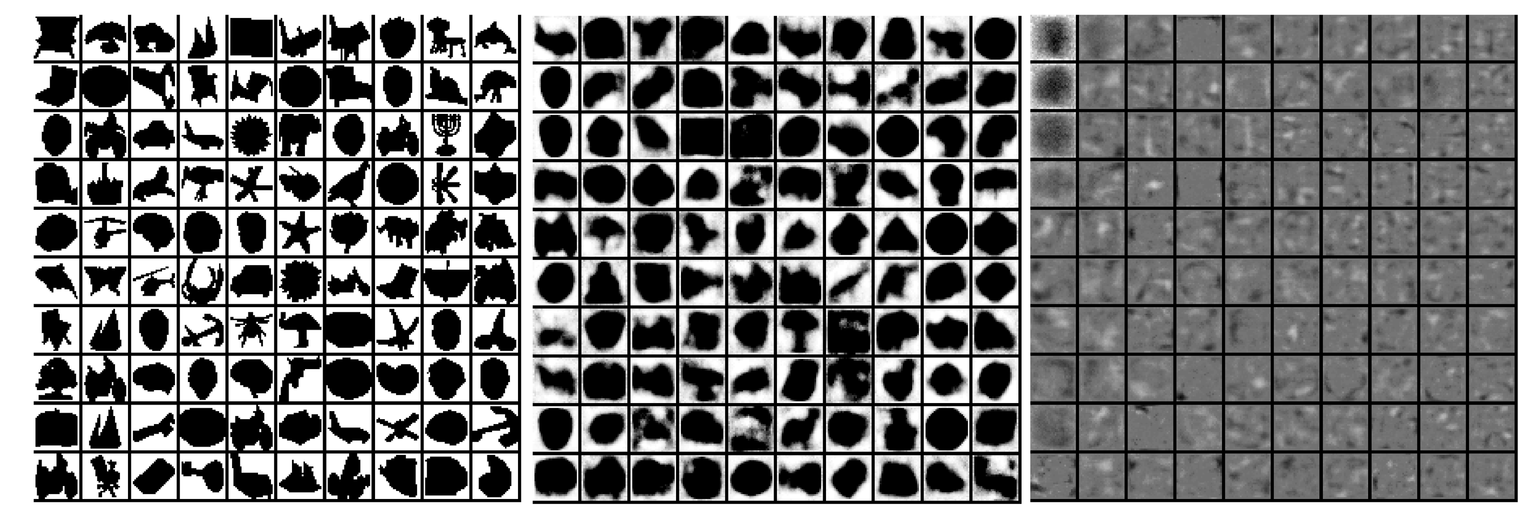
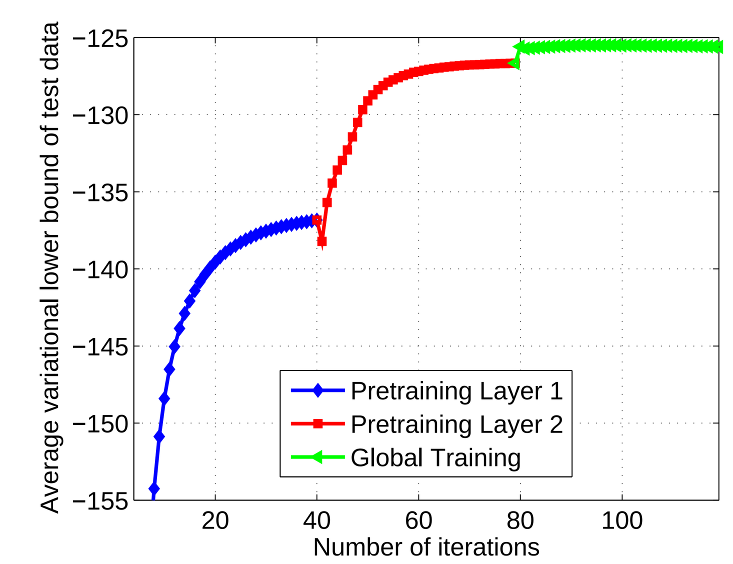


Figure: Performance on the Caltech 101 Silhouettes dataset. (Left) Training data. (Middle) Averaged synthesized samples. (Right) Learned features at the bottom layer.

Variational Lower Bound:

Table: Log probability of test data.

Model	Dim	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



III. OCR letters: Variational Lower Bound:

Table: OCR letters.

Model	Dim	Log-prob.
SBN (online)	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

Table: MNIST.

Model	Dim	Log-prob.
SBN (VB)	200	-116.96
FVSBN (VB)	$-$	-100.76
ARSBN (VB)	200	-102.11
ARSBN (VB)	$200-200$	-101.19
SBN° (NVIL)	200	-113.1
SBN° (NVIL)	$200-200$	-99.8
DBN*	$500-2000$	-86.22
DBM▹	$500-1000$	-84.62

ACKNOWLEDGEMENTS

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