Parsimonious Bayesian Deep Networks

Mingyuan Zhou

The University of Texas at Austin, Austin, TX, USA

Introduction

We propose parsimonious Bayesian deep networks (PBDNs) to infer capacity-regularized network architectures from the data:

- Use Bayesian nonparametrics (gamma process) to determine the size of a hidden layer.
- Use a forward model selection strategy to determine the depth of the network.
- Capacity regularization is built into the greedy-layer-wise construction and training of the deep network, requiring neither cross-validation nor fine-tuning.
- Inference via Gibbs sampling or MAP-SGD, low computational complexity for out-of-sample prediction.
- Interpretable data subtypes near the decision boundaries.

Infinite support hyperplane machine (iSHM)

Hierarchical model

$$y_i \,|\, G, \boldsymbol{x}_i \sim \text{Bernoulli}\big(1 - e^{-\sum_{k=1}^{\infty} r_k \ln(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k})}\big)$$

$G = \sum_{k=1}^{\infty} r_k \delta_{\boldsymbol{\beta}_k}$ represents a draw from a gamma process

Noisy-Or interpretation:

$$P(y_i = 1 \,|\, \{r_k, \boldsymbol{\beta}_k\}_k, \boldsymbol{x}_i) = 1 - \prod_{k=1}^{\infty}(1 - p_{ik})$$

$$p_{ik} = 1 - e^{-r_k \ln(1+e^{\boldsymbol{\beta}_k'\boldsymbol{x}_i})}$$

Noisy-Or hierarchical representation:

$$y_i = \bigvee_{k=1}^{\infty} b_{ik}, \quad b_{ik} \sim \text{Bernoulli}(p_{ik})$$

$$p_{ik} = 1 - e^{-\theta_{ik}}, \quad \theta_{ik} \sim \text{Gamma}(r_k, e^{\boldsymbol{\beta}_k'\boldsymbol{x}_i})$$

Alternative hierarchical representation:

$$y_i = \delta(m_i \geq 1), \quad m_i = \sum_{k=1}^{\infty} m_{ik},$$

$$m_{ik} \sim \text{Pois}(\theta_{ik}), \quad \theta_{ik} \sim \text{Gamma}(r_k, e^{\boldsymbol{\beta}_k'\boldsymbol{x}_i})$$

Hierarchical model:

$$r_k \sim \text{Gamma}(\gamma_0/K, 1/c_0), \ \gamma_0 \sim \text{Gamma}(a_0, 1/b_0), \ c_0 \sim \text{Gamma}(e_0, 1/f_0)$$

$$\boldsymbol{\beta}_k \sim \prod_{v=0}^{V} \int \mathcal{N}(0, \alpha_{vk}^{-1}) \text{Gamma}(\alpha_{vk}; a_\beta, 1/b_{\beta k}) d\alpha_{vk}, \ \ b_{\beta k} \sim \text{Gamma}(e_0, 1/f_0)$$

NeurIPS | 2018

Model properties

Bias towards data labeled as one:

$$\text{NLL}(\boldsymbol{x}_i) = \lambda_i - \ln(e^{\lambda_i} - 1) \text{ if } y_i = 1 \text{ and } \text{NLL}(\boldsymbol{x}_i) = \lambda_i \text{ if } y_i = 0$$

$$\lambda_i = \sum_{k=1}^{\infty} r_k \ln(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k})$$

Convex polytope bounded decision boundary:

If $\quad P(y_i = 1 \,|\, \{r_k, \boldsymbol{\beta}_k\}_k, \boldsymbol{x}_i) \leq p_0$

Then $\quad \boldsymbol{x}_i'\boldsymbol{\beta}_k \leq \ln\big[(1-p_0)^{-\frac{1}{r_k}} - 1\big], \ \ k \in \{1, 2, \ldots\}$

Inference

Gibbs sampling with closed-form update equations

Maximum a posteriori estimation via stochastic gradient descent

$$f(\{\boldsymbol{\beta}_k, \ln r_k\}_1^K, \{y_i, \boldsymbol{x}_i\}_{i_1}^{i_M}) = \sum_{k=1}^{K}\big(-\tfrac{\gamma_0}{K}\ln r_k + c_0 e^{\ln r_k}\big) + (a_\beta + 1/2)\sum_{v=0}^{V}\sum_{k=0}^{K} [\ln(1 + \beta_{vk}^2/(2b_{\beta k}))] + \tfrac{N}{M}\sum_{i=i_1}^{i_M}\big[-y_i \ln\big(1 - e^{-\lambda_i}\big) + (1 - y_i)\lambda_i\big]$$

Prune hyperplane k if $\sum_i b_{ik} = 0$

Train a pair of iSHMs under two opposite labeling settings

Network-depth learning via forward model selection

Greedy layer-wise construction and training

Add and train iSHM pairs one at a time:

$$\tilde{\boldsymbol{x}}_i^{(t+1)} = \big[\ln(1 + e^{(\boldsymbol{x}_i^{(t)})'\boldsymbol{\beta}_1^{(t\to t+1)}}), \ldots, \ln(1 + e^{(\boldsymbol{x}_i^{(t)})'\boldsymbol{\beta}_{K_{t+1}}^{(t\to t+1)}})\big]'$$

$$\boldsymbol{x}_i^{(t+1)} = [1, (\tilde{\boldsymbol{x}}_i^{(t)})', (\tilde{\boldsymbol{x}}_i^{(t+1)})']' \in \mathbb{R}^{K_t + K_{t+1} + 1}$$

Model selection criteria

$$\text{AIC}(T) = \sum_{t=1}^{T}[2(K_t + 1)K_{t+1}] + 2K_{T+1} - 2\sum_i \Big[\ln P(y_i \,|\, \boldsymbol{x}_i^{(T)}) + \ln P(y_i^* \,|\, \boldsymbol{x}_i^{(T)})\Big]$$

$$\text{AIC}_\epsilon(T) = \sum_{t=1}^{T} 2\left(\||\mathbf{B}_t| > \epsilon\beta_{t\,\max}\|_0 + \||\mathbf{B}_t^*| > \epsilon\beta_{t\,\max}^*\|_0\right) + 2K_{T+1} - 2\sum_i \Big[\ln P(y_i \,|\, \boldsymbol{x}_i^{(T)}) + \ln P(y_i^* \,|\, \boldsymbol{x}_i^{(T)})\Big]$$

Algorithm 2: Greedy layer-wise training for PBDN.

Denote $\boldsymbol{x}_i^{(1)} = \boldsymbol{x}_i$ and $\text{AIC}(T) = \infty$.
for Layer $t = 1 : \infty$ **do**
  Train an iSHM to predict y_i given $\boldsymbol{x}_i^{(t)}$;
  Train an iSHM to predict $y_i^* = 1 - y_i$ given $\boldsymbol{x}_i^{(t)}$;
  Compute $P(y_i \,|\, \boldsymbol{x}_i^{(t)})$, $P(y_i^* \,|\, \boldsymbol{x}_i^{(t)})$, and $\text{AIC}(t)$;
  if $\text{AIC}(t) < \text{AIC}(t-1)$ **then**
    Combine two iSHMs to produce $\boldsymbol{x}_i^{(t+1)}$;
  else
    Use the first $(t-1)$ iSHM pairs to compute the conditional class probability $P(y_i \,|\, \boldsymbol{x}_i)$;
  end if
end for

	LR	SVM	RVM	AMM	CPM	DNN (8-4)	DNN (32-16)	DNN (128-64)	PBDN1	PBDN2	PBDN4	AIC Gibbs	AIC$_\epsilon$ Gibbs	AIC SGD	AIC$_\epsilon$ SGD
Mean of SVM normalized errors	2.237	1.000	1.110	1.234	1.227	1.260	1.087	1.031	1.219	1.009	0.998	1.006	0.996	1.073	1.029
Mean of SVM normalized K	0.006	1.000	0.113	0.069	0.046	0.073	0.635	8.050	0.042	0.060	0.160	0.057	0.064	0.128	0.088

Table 3: The inferred depth of PBDN that increases its depth until a model selection criterion starts to rise.

Dataset	banana	breast cancer	titanic	waveform	german	image	ijcnn1	a9a
AIC-Gibbs	2.30 ± 0.48	1.00 ± 0.00	1.00 ± 0.00	1.90 ± 0.74	1.30 ± 0.67	2.40 ± 0.52	2.00 ± 0.00	1.00 ± 0.00
AIC$_{\epsilon=0.01}$-Gibbs	2.30 ± 0.48	1.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.67	1.60 ± 0.84	2.60 ± 0.52	3.40 ± 0.55	1.00 ± 0.00
AIC-SGD	3.20 ± 0.78	1.90 ± 0.99	1.00 ± 0.00	2.40 ± 0.52	2.80 ± 0.63	2.90 ± 0.74	3.20 ± 0.45	3.20 ± 0.45
AIC$_{\epsilon=0.01}$-SGD	2.80 ± 0.63	1.00 ± 0.00	1.00 ± 0.00	1.50 ± 0.53	1.00 ± 0.00	2.00 ± 0.00	3.00 ± 0.00	1.00 ± 0.00

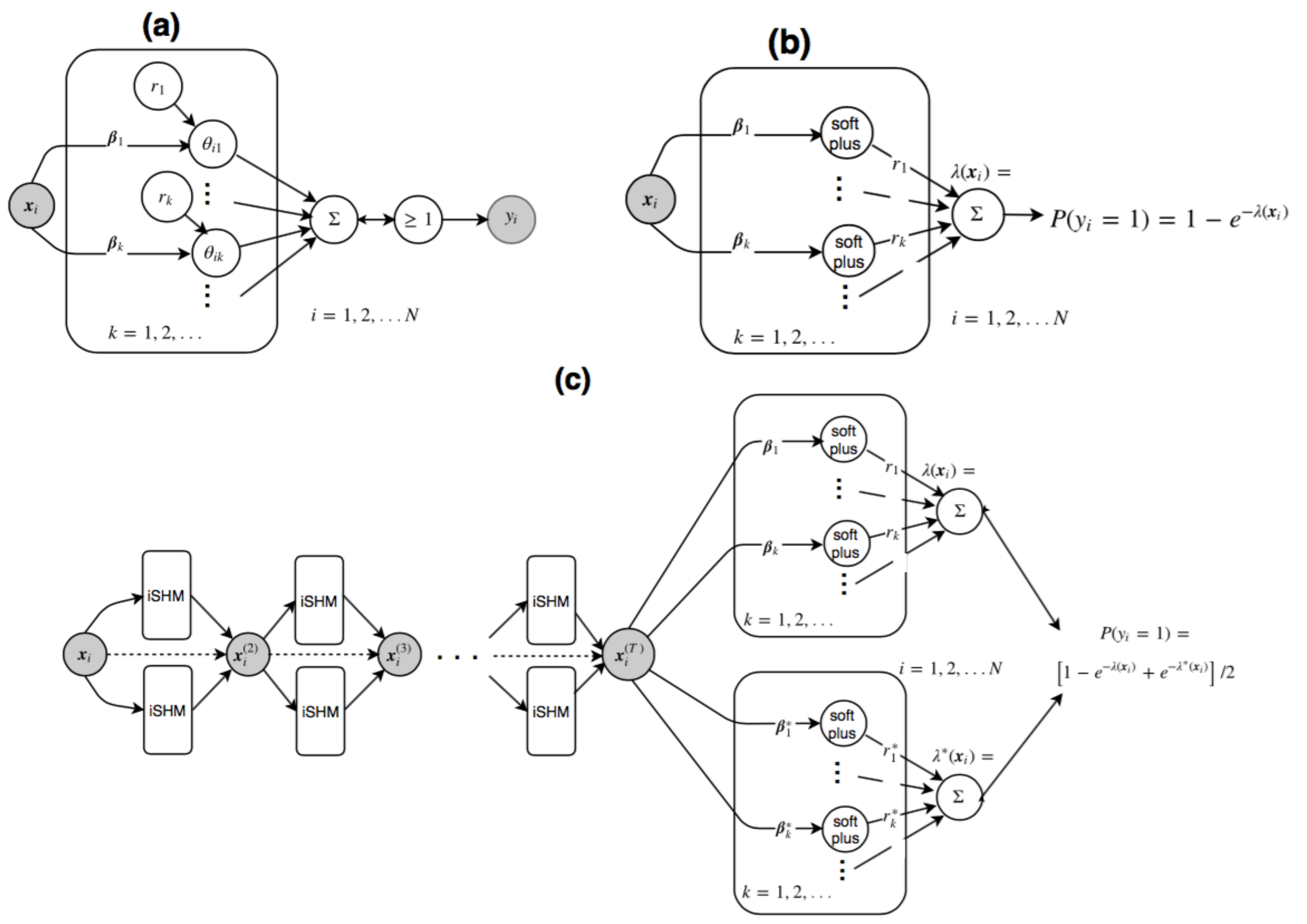


Figure 1: Visualization of PBDN, each layer of which is a pair of iSHMs trained on a "two spirals" dataset under two opposite labeling settings.

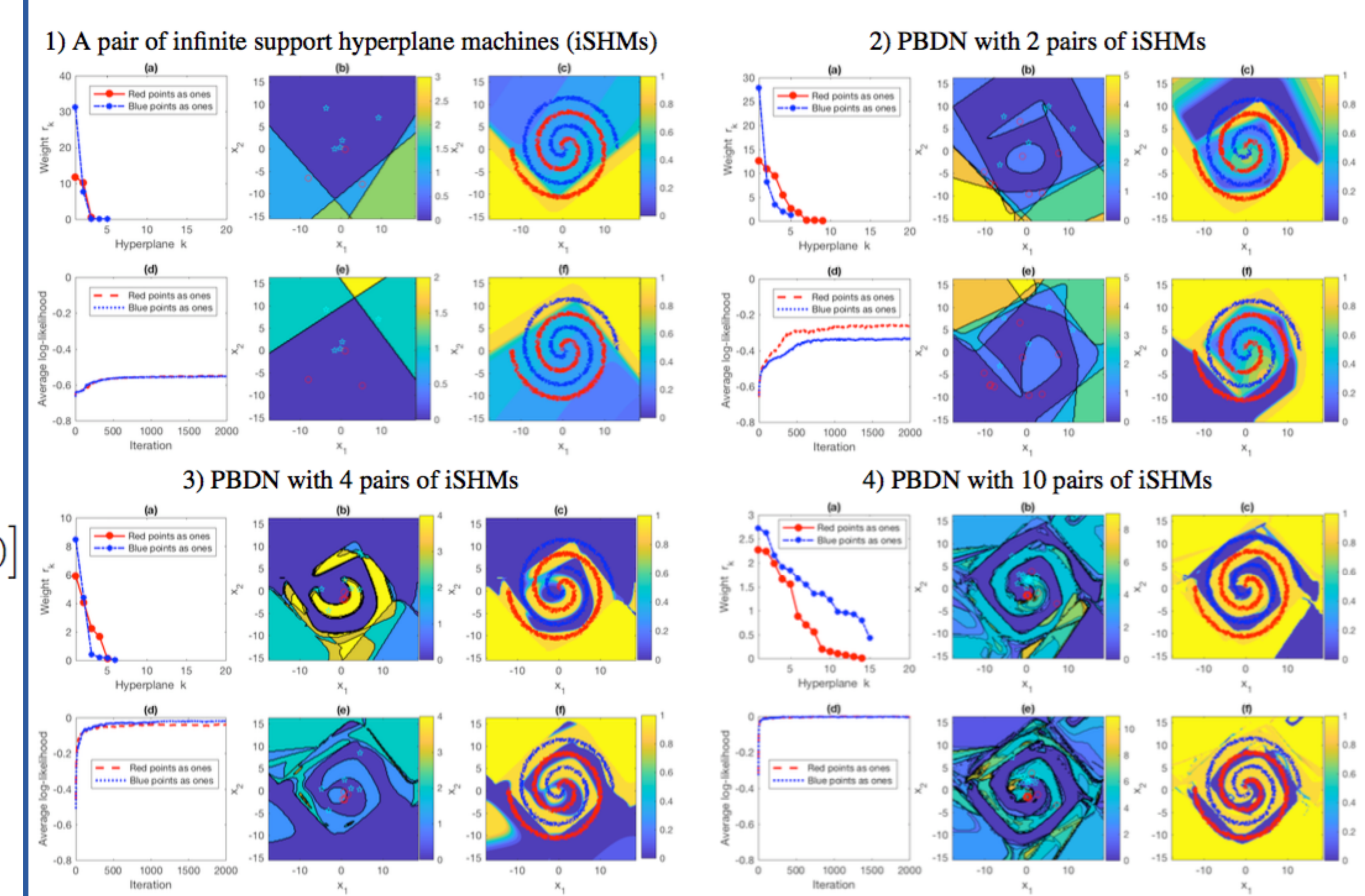
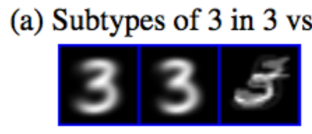
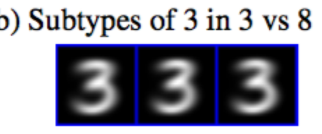
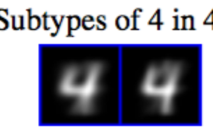
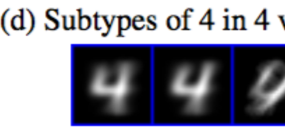
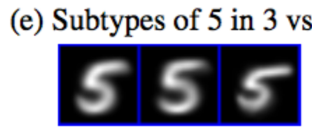
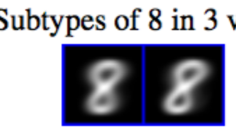
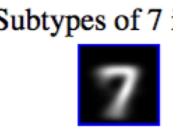
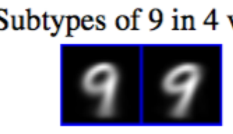


Table 1: Visualization of the subtypes inferred by PBDN in a random trial and comparison of classification error rates over five random trials between PBDN and a two-hidden-layer DNN (128-64) on four different MNIST binary classification tasks.

	(a) Subtypes of 3 in 3 vs 5 	(b) Subtypes of 3 in 3 vs 8 	(c) Subtypes of 4 in 4 vs 7 	(d) Subtypes of 4 in 4 vs 9 
	(e) Subtypes of 5 in 3 vs 5 	(f) Subtypes of 8 in 3 vs 8 	(g) Subtypes of 7 in 4 vs 7 	(h) Subtypes of 9 in 4 vs 9 
PBDN	$2.53\% \pm 0.22\%$	$2.66\% \pm 0.27\%$	$1.37\% \pm 0.18\%$	$2.95\% \pm 0.47\%$
DNN	$2.78\% \pm 0.36\%$	$2.93\% \pm 0.40\%$	$1.21\% \pm 0.12\%$	$2.98\% \pm 0.17\%$