Bayesian Inference in Augmented Bow Tie Networks

Jimmy T.H. Smith
Stanford University
jsmith14@stanford.edu

Dieterich Lawson
Stanford University
jdlawson@stanford.edu

Scott W. Linderman
Stanford University
swl1@stanford.edu

Abstract

We develop a deep generative model that generalizes feed-forward, rectified linear neural networks with stochastic activations. We call these models *bow tie networks* because of the shape of their activation distributions. We leverage the Pólya-gamma augmentation scheme to render the model conditionally conjugate and derive a block Gibbs sampling algorithm to approximate the posterior distribution over activations and model parameters. The resulting algorithm is massively parallelizable. We show a proof-of-concept of this model and Bayesian inference algorithm on a variety of standard regression benchmarks.

1 Introduction

Consider a deep generative model for nonlinear regression. Let $\boldsymbol{x}_n \in \mathbb{R}^{D_x}$ denote the inputs, $\boldsymbol{y}_n \in \mathbb{R}^{D_y}$ the outputs, and $\boldsymbol{a}_n = \{\boldsymbol{a}_{n,l}\}_{l=1}^L$ with $\boldsymbol{a}_{n,l} \in \mathbb{R}^{D_l}$ the latent activations at each of L intermediate layers. We model the joint distribution as,

$$p(\boldsymbol{y}_n, \boldsymbol{a}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = \left[\prod_{l=1}^{L} \mathcal{N}(\boldsymbol{a}_{n,l} \mid \boldsymbol{\mu}_{n,l}, \boldsymbol{\Sigma}_l)\right] \mathcal{N}(\boldsymbol{y}_n \mid \boldsymbol{u}_{n,L+1}, \boldsymbol{\Sigma}_{L+1}) \tag{1}$$

$$\boldsymbol{\mu}_{n,l} \triangleq f(\boldsymbol{u}_{n,l}) \qquad \boldsymbol{u}_{n,1} \triangleq \boldsymbol{W}_1 \boldsymbol{x}_n + \boldsymbol{b}_1, \qquad \boldsymbol{u}_{n,l} \triangleq \boldsymbol{W}_l \boldsymbol{a}_{n,l-1} + \boldsymbol{b}_l \text{ for } l > 1, \tag{2}$$

where f is a nonlinear function applied element-wise. The parameters consist of the weights, biases, and variances, $\boldsymbol{\theta} = \{\boldsymbol{W}_l, \boldsymbol{b}_l, \boldsymbol{\Sigma}_l\}_{l=1}^{L+1}$. Though we have modeled the activations as random variables, we can recover standard, feed-forward neural networks when $\boldsymbol{\Sigma}_l \to \boldsymbol{0}$ for $l = 1, \ldots, L$.

Our goal is to infer the posterior distribution over parameters and activations given a set of observed inputs and outputs,

$$p(\boldsymbol{\theta}, \{\boldsymbol{a}_n\}_{n=1}^N \mid \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N) \propto p(\boldsymbol{\theta}) \prod_{n=1}^{N} p(\boldsymbol{y}_n, \boldsymbol{a}_{n,1}, \ldots, \boldsymbol{a}_{n,L} \mid \boldsymbol{x}_n, \boldsymbol{\theta}), \tag{3}$$

under a prior $p(\boldsymbol{\theta})$.

MCMC methods like Hamiltonian Monte Carlo and variational approaches like Bayes by backprop [Blundell et al., 2015] could be used for posterior inference since they only require that the log probability be differentiable. Instead, we introduce additional structure to the model which allows for alternative methods that may yield more efficient inference. Specifically, we propose a deep generative model that relaxes the model above so that it is amenable to Pólya-gamma augmentation [Polson et al., 2013]. This renders the model conditionally linear and Gaussian. Given this, with tractable conditional distributions over large subsets of variables, methods such as Gibbs sampling can update many variables at once and potentially converge more quickly than local, gradient-based algorithms.

We call the model a *bow tie network* due to the shape of the conditional distribution of its activation function. After introducing the model, we derive a Gibbs sampling algorithm that leverages the simple conditionals to perform Bayesian inference. We demonstrate a prototype of the model and

Bayesian Deep Learning workshop, NeurIPS 2021.

algorithm on various regression tasks and compare it to alternative algorithms for Bayesian deep learning.

2 Bow Tie Networks

Take the generative model from eq. 1 and assume that $f(u) = \max\{0, u\}$ is the rectified linear (ReLU) function. Nonlinear activation functions such as this make posterior inference hard—if f were the identity, we could place conjugate priors on the parameters and derive closed form Gibbs updates for the parameters and activations (more details below). Observe, however, that we could equivalently write the rectified linear function as $f(u) = zu$ where $z = \mathbb{I}[u > 0]$ is a binary activation that determines whether the node is *on* ($z = 1$) or *off* ($z = 0$). Intuitively, if we knew the binary activations, then f would be conditionally linear, which would make inference much simpler.

Motivated by this observation, we propose a stochastic relaxation of the rectified linear network. In the notation of eq. 1, let

$$\boldsymbol{\mu}_{n,l} \triangleq \boldsymbol{z}_{n,l} \odot \boldsymbol{u}_{n,l} \tag{4}$$

$$\boldsymbol{z}_{n,l} \overset{\text{ind}}{\sim} \text{Bern}(\sigma(\boldsymbol{u}_{n,l}/\tau)), \tag{5}$$

where $\odot$ denotes the elementwise product, σ is the logistic function, $\boldsymbol{w}_{l,d} \in \mathbb{R}^{D_{l-1}}$ is the d-th row of $\boldsymbol{W}_l$, and $\tau \geq 0$ is a temperature parameter. Under this model, the rectified linear units are turned on or off stochastically depending on their input.

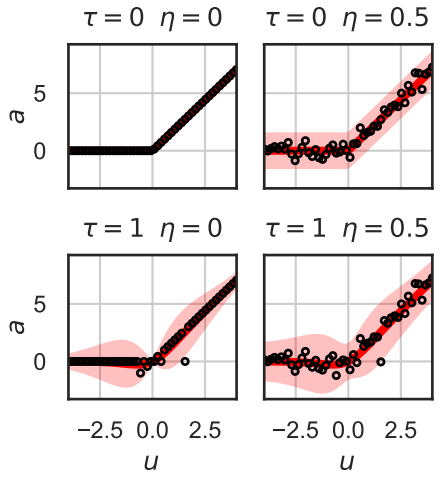

Figure 1: Conditional distribution of the activation a given the input u (marginalizing over the binary activations) for various settings of the temperature τ and the noise η. Red: the conditional mean and variance; black dots: samples from the distribution. When $\tau = \eta = 0$, we recover the standard rectified linear function; for nonzero values, the conditional distribution looks like a bow tie.

Figure 1 shows the conditional distribution of the activation $a_{n,l,d}$ given the input $u_{n,l,d}$ and marginalizing over the binary activations $z_{n,l,d}$, for a few values of the temperature τ and the variance $\eta_{l,d}^2 = [\boldsymbol{\Sigma}_l]_{dd}$. For intermediate values of the temperature and noise, the conditional distribution looks like a bow tie, hence the name.

3 Bayesian inference via Pólya-gamma augmentation

The bow tie network introduces binary activations to the model so that,

$$p(\boldsymbol{y}_n, \boldsymbol{a}_n, \boldsymbol{z}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = \left[\prod_{l=1}^{L} \text{Bern}(\boldsymbol{z}_{n,l} \mid \sigma(\boldsymbol{u}_{n,l}/\tau))\, \mathcal{N}(\boldsymbol{a}_{n,l} \mid \boldsymbol{z}_{n,l} \odot \boldsymbol{u}_{n,l}, \boldsymbol{\Sigma}_l)\right] \mathcal{N}(\boldsymbol{y}_n \mid \boldsymbol{u}_{n,L+1}, \boldsymbol{\Sigma}_{L+1}) \tag{6}$$

The Bernoulli terms still make posterior inference challenging. However, this formulation is amenable to Pólya-gamma (PG) augmentation [Polson et al., 2013]—an auxiliary variable method that equates the Bernoulli pmf with a scale-mean mixture of Gaussians. Let $\boldsymbol{\gamma}_n = \{\boldsymbol{\gamma}_{n,l}\}_{l=1}^L$ with $\boldsymbol{\gamma}_{n,l} \in \mathbb{R}_+^{D_l}$ denote the collection of PG auxiliary variables. The joint distribution on the extended space is,

$$p(\boldsymbol{y}_n, \boldsymbol{a}_n, \boldsymbol{z}_n, \boldsymbol{\gamma}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) \propto \left[\prod_{l=1}^{L} \text{PG}(\boldsymbol{\gamma}_{n,l} \mid 0, 1) e^{\tau^{-1}(\boldsymbol{z}_{n,l} - \frac{1}{2})^\top \boldsymbol{u}_{n,l} - \frac{1}{2\tau^2} \boldsymbol{u}_{n,l}^\top \boldsymbol{G}_{n,l} \boldsymbol{u}_{n,l}} \mathcal{N}(\boldsymbol{a}_{n,l} \mid \boldsymbol{z}_{n,l} \odot \boldsymbol{u}_{n,l}, \boldsymbol{\Sigma}_l)\right] \times \mathcal{N}(\boldsymbol{y}_n \mid \boldsymbol{u}_{n,L+1}, \boldsymbol{\Sigma}_{L+1}), \tag{7}$$

where $\text{PG}(\boldsymbol{\gamma}_{n,l} \mid 1, 0)$ is the probability density of a bank of D_l independent, standard PG random variables and $\boldsymbol{G}_{n,l} = \text{diag}(\boldsymbol{\gamma}_{n,l})$.

	Test RMSE			Test Log-likelihood		
Dataset	BBB	fBNN	Bow Tie	BBB	fBNN	Bow Tie
Boston	3.171 ± 0.149	2.378 ± 0.104	3.324 ± 0.167	-2.602 ± 0.031	-2.301 ± 0.038	-2.506 ± 0.035
Concrete	5.678 ± 0.087	4.935 ± 0.180	5.241 ± 0.206	-3.149 ± 0.018	-3.096 ± 0.016	-3.045 ± 0.041
Energy	0.565 ± 0.018	0.412 ± 0.017	1.096 ± 0.086	-1.500 ± 0.006	-0.684 ± 0.020	-1.334 ± 0.060
Wine	0.643 ± 0.012	0.673 ± 0.014	0.651 ± 0.020	-0.977 ± 0.017	-1.040 ± 0.013	-0.992 ± 0.059
Yacht	1.174 ± 0.086	0.607 ± 0.068	0.896 ± 0.083	-2.408 ± 0.007	-1.033 ± 0.033	-1.139 ± 0.093

Table 1: Performance of bow tie networks, Bayes by backprop (BBB) Blundell et al. [2015], and functional variational Bayesian neural networks (fBNN) Sun et al. [2019] on UCI regression datasets Dua and Graff [2017]. Bow tie networks are generally competitive with both of these methods. BBB and fBNN results are from Sun et al. [2019].

The log probability under the augmented model is quadratic in $\boldsymbol{u}_{n,l}$, which is in turn a linear function of the weights, biases, and preceding layer's activations. Thus, the augmented model is conjugate with a Gaussian prior on the weights and an inverse Wishart prior on the covariance matrices (or more generally a matrix-normal inverse Wishart prior). Moreover, the conditional distribution of the activations $\boldsymbol{a}_n = (\boldsymbol{a}_{n,1}, \ldots, \boldsymbol{a}_{n,L})$ is a linear Gaussian chain. That means we can Gibbs sample the activations via an efficient forward filtering backward sampling algorithm in order $O(NLD^3)$ time (where $D = \max_l D_l$). When the covariances are constrained to be diagonal, the binary activations follow simple, independent Bernoulli conditionals. Likewise, the PG variables follow a tilted PG conditional distribution, which is also easy to sample from. Finally, note that the latent variables are conditionally independent across data points, so the Gibbs sampler can be easily parallelized across this dimension. Complete details are in the appendix.

4 Experiments

We evaluated bow tie networks on a standard set of regression tasks from the UCI dataset repository Dua and Graff [2017]. Following Sun et al. [2019], we selected only datasets with fewer than 2000 data points so that full-batch learning was tractable. The bow tie network architecture used a single hidden layer of 50 units, consistent with the experiment in Sun et al. [2019]. The activation temperature τ was fixed at 0.1 and 1,000 samples were collected after a burn-in of 26,000 steps. See table 1 for the experimental results. Bow tie networks were generally competitive with both Bayes by backprop (BBB) and functional variational Bayesian neural networks (fBNNs) on the regression tasks.

5 Conclusions

We presented bow tie networks, a novel model for Bayesian neural networks that allows for fast and accurate inference via Pólya-gamma augmentation. In the future we will improve the performance of bow tie networks by exploring alternative sampling strategies such as annealing the temperature over training. We also hope to scale bow tie networks to larger datasets and explore connections with inference in the function space.

References

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France, 07–09 Jul 2015. PMLR.

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using Pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504): 1339–1349, 2013.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*, 2019.

A Complete conditional distributions

In this section we provide the complete conditional distributions necessary for the Gibbs sampling algorithm described in Section 3.

Let $\widetilde{\boldsymbol{a}}_{n,l} = (\boldsymbol{a}_{n,l}, 1)$ and $\widetilde{\boldsymbol{w}}_{l,d} \triangleq (\boldsymbol{w}_{l,d}, b_{l,d})$ where $\boldsymbol{w}_{l,d} \in \mathbb{R}^{D_{l-1}}$ is the d-th row of the weights $\boldsymbol{W}_l$. Assume, for simplicity, that the noise covariances are diagonal $\boldsymbol{\Sigma}_l = \mathrm{diag}(\eta_{l,1}^2, \ldots, \eta_{l,D_l}^2)$. Then assume a factored prior,

$$p(\boldsymbol{\theta}) = \prod_{l=1}^{L} \prod_{d=1}^{D_l} \mathcal{N}(\widetilde{\boldsymbol{w}}_{l,d} \mid \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\, \mathrm{Ga}(\eta_{l,d}^{-2} \mid \alpha_0, \beta_0). \tag{8}$$

Each of the updates below can be performed in parallel. For example, activations can be sampled in parallel for all data points, the weights can be sampled in parallel for indices all layers and output dimensions, and the binary activations and auxiliary variables can be sampled in parallel for all data points, layers, and dimensions. Leveraging this parallelism could lead to much faster implementations on massively parallel architectures like GPUs.

Conditional distribution of the activations. The most interesting conditional distribution is that of the activations. Given the parameters, binary activations, and auxiliary variables, the conditional distribution of the activations is proportional to,

$$p(\boldsymbol{a}_n \mid \boldsymbol{y}_n, \boldsymbol{z}_n, \boldsymbol{\gamma}_n, \boldsymbol{x}_n, \boldsymbol{\theta}) \propto \exp\left\{ -\frac{1}{2} \sum_{l=1}^{L} \boldsymbol{a}_{n,l}^\top \boldsymbol{J}_{n,l} \boldsymbol{a}_{n,l} - \sum_{l=2}^{L} \boldsymbol{a}_{n,l}^\top \boldsymbol{L}_{n,l} \boldsymbol{a}_{n,l-1} + \sum_{l=1}^{L} \boldsymbol{h}_{n,l}^\top \boldsymbol{a}_{n,l} \right\}. \tag{9}$$

This is a linear Gaussian chain, just like a Gaussian linear dynamical system (LDS). To simplify the expressions below, let $\boldsymbol{Z}_{n,l} = \mathrm{diag}(\boldsymbol{z}_{n,l})$. Then the coefficients are,

$$\boldsymbol{J}_{n,l} = \boldsymbol{\Sigma}_l^{-1} + \boldsymbol{W}_{l+1}^\top (\tau^{-2} \boldsymbol{G}_{n,l+1} + \boldsymbol{Z}_{n,l+1} \boldsymbol{\Sigma}_{l+1}^{-1} \boldsymbol{Z}_{n,l+1}) \boldsymbol{W}_{l+1} \qquad \text{for } l < L \tag{10}$$

$$\boldsymbol{J}_{n,L} = \boldsymbol{\Sigma}_L^{-1} + \boldsymbol{W}_{L+1}^\top \boldsymbol{\Sigma}_{L+1} \boldsymbol{W}_{L+1} \tag{11}$$

$$\boldsymbol{L}_{n,l} = -\boldsymbol{Z}_{n,l} \boldsymbol{\Sigma}_l^{-1} \boldsymbol{W}_l \tag{12}$$

$$\boldsymbol{h}_{n,1} = \boldsymbol{Z}_1 \boldsymbol{\Sigma}_1^{-1} \boldsymbol{b}_1 + \boldsymbol{W}_2^\top (\tfrac{1}{\tau}(\boldsymbol{z}_{n,2} - \tfrac{1}{2}) - \tfrac{1}{\tau^2} \boldsymbol{G}_{n,2} \boldsymbol{b}_2 - \boldsymbol{Z}_{n,2} \boldsymbol{\Sigma}_2^{-1} \boldsymbol{b}_2) + \boldsymbol{Z}_{n,1} \boldsymbol{\Sigma}_1^{-1} \boldsymbol{W}_1 \boldsymbol{x}_n \tag{13}$$

$$\boldsymbol{h}_{n,l} = \boldsymbol{Z}_l \boldsymbol{\Sigma}_l^{-1} \boldsymbol{b}_l + \boldsymbol{W}_{l+1}^\top (\tfrac{1}{\tau}(\boldsymbol{z}_{n,l+1} - \tfrac{1}{2}) - \tfrac{1}{\tau^2} \boldsymbol{G}_{n,l+1} \boldsymbol{b}_{l+1} - \boldsymbol{Z}_{n,l+1} \boldsymbol{\Sigma}_{l+1}^{-1} \boldsymbol{b}_{l+1}) \tag{14}$$

$$\boldsymbol{h}_{n,L} = \boldsymbol{Z}_L \boldsymbol{\Sigma}_L^{-1} \boldsymbol{b}_L + \boldsymbol{W}_{L+1}^\top \boldsymbol{\Sigma}_{L+1}^{-1} (\boldsymbol{y}_n - \boldsymbol{b}_{l+1}) \tag{15}$$

We can draw exact samples from this conditional distribution in $O(LD^3)$ time where $D = \max_l D_l$ using the forward filtering backward sampling algorithm. Note that we have presented the conditional distribution in "information form" rather than in terms of mean parameters, as linear Gaussian dynamical systems.

Conditional distribution of the weights and biases. Under this prior, the conditional distribution of the weights is,

$$p(\widetilde{\boldsymbol{w}}_{l,d} \mid \boldsymbol{a}_{n,l}, \boldsymbol{z}_{n,l}, \boldsymbol{\gamma}_{n,l}, \boldsymbol{a}_{n,l-1}) = \mathcal{N}(\widetilde{\boldsymbol{w}}_{l,d} \mid \boldsymbol{J}_{l,d}^{-1} \boldsymbol{h}_{l,d}, \boldsymbol{J}_{l,d}^{-1}) \tag{16}$$

$$\text{where } \boldsymbol{J}_{l,d} = \boldsymbol{\Sigma}_0^{-1} + \sum_{n=1}^{N} (\tau^{-2} \gamma_{n,l,d} + \eta_{l,d}^{-2} z_{n,l,d})\, \widetilde{\boldsymbol{a}}_{n,l} \widetilde{\boldsymbol{a}}_{n,l}^\top \tag{17}$$

$$\boldsymbol{h}_{l,d} = \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 + \sum_{n=1}^{N} (\tau^{-1}(z_{n,l,d} - \tfrac{1}{2}) + \eta_{l,d}^{-2} z_{n,l,d}\, a_{n,l,d})\, \widetilde{\boldsymbol{a}}_{n,l} \tag{18}$$

Conditional distribution of the variances. The conditional distribution of the precisions (inverse variances) is,

$$p(\eta_{l,d}^{-2} \mid \boldsymbol{a}_{n,l}, \boldsymbol{z}_{n,l}, \boldsymbol{\gamma}_{n,l}, \boldsymbol{a}_{n,l-1}, \boldsymbol{w}_{l,d}, b_{l,d}) = \mathrm{Ga}(\eta_{l,d}^{-2} \mid \alpha_{l,d}, \beta_{l,d}) \tag{19}$$

$$\text{where } \alpha_{l,d} = \alpha_0 + \frac{N}{2} \tag{20}$$

$$\beta_{l,d} = \beta_0 + \frac{1}{2} \sum_{n=1}^{N} (a_{n,l,d} - z_{n,l,d} u_{n,l,d})^2 \tag{21}$$

Conditional distribution of the binary activations. The conditional distribution of the binary activations, marginalizing over the PG auxiliary variables, is,

$$p(z_{n,l,d} \mid \boldsymbol{a}_{n,l}, \boldsymbol{a}_{n,l-1}, \boldsymbol{\theta}) = \mathrm{Bern}(\sigma(\nu_{n,l,d})) \tag{22}$$

$$\text{where } \nu_{n,l,d} = \frac{u_{n,l,d}}{\tau} + \frac{a_{n,l,d} u_{n,l,d}}{\eta_{l,d}^2} - \frac{u_{n,l,d}^2}{2\eta_{l,d}^2} \tag{23}$$

Conditional distribution of the PG auxiliary variables. Since we Gibbs sample the binary activations from their conditional distribution under the non-augmented model (eq. 6), we must immediately Gibbs sample the PG auxiliary variables after updating the binary activations. Their conditional distribution is,

$$p(\gamma_{n,l,d} \mid \widetilde{\boldsymbol{a}}_{n,l-1}, \boldsymbol{\theta}) = \mathrm{PG}(\gamma_{n,l,d} \mid 1, \tau^{-1} u_{n,l,d}) \tag{24}$$