Structured Variational Learning of Bayesian Neural Networks with Horseshoe Priors

Soumya Ghosh [1 2] **Jiayu Yao** [3] **Finale Doshi-Velez** [3]

Abstract

Bayesian Neural Networks (BNNs) have recently received increasing attention for their ability to provide well-calibrated posterior uncertainties. However, model selection—even choosing the number of nodes—remains an open question. Recent work has proposed the use of a horseshoe prior over node pre-activations of a Bayesian neural network, which effectively turns off nodes that do not help explain the data. In this work, we propose several modeling and inference advances that consistently improve the compactness of the model learned while maintaining predictive performance, especially in smaller-sample settings including reinforcement learning.

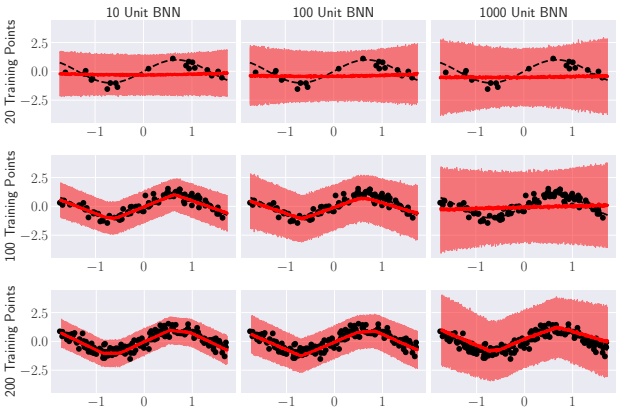

Figure 1. Predictive distributions from a single layer BNN with $\mathcal{N}(0, 1)$ priors over weights, containing 10, 100, and 1000 units, trained on noisy samples (in black) from a smooth 1 dimensional function shown in black. With fixed data increasing BNN capacity leads to over-inflated uncertainty.

1. Introduction

Bayesian Neural Networks (BNNs) are increasingly the de-facto approach for modeling stochastic functions. By treating the weights in a neural network as random variables, and performing posterior inference on these weights, BNNs can avoid overfitting in the regime of small data, provide well-calibrated posterior uncertainty estimates, and model a large class of stochastic functions with heteroskedastic and multi-modal noise. These properties have resulted in BNNs being adopted in applications ranging from active learning (Hernández-Lobato & Adams, 2015; Gal et al., 2016a) and reinforcement learning (Blundell et al., 2015; Depeweg et al., 2017).

While there have been many recent advances in training BNNs (Hernández-Lobato & Adams, 2015; Blundell et al., 2015; Rezende et al., 2014; Louizos & Welling, 2016; Hernandez-Lobato et al., 2016), model-selection in BNNs has received relatively less attention. Unfortunately, the consequences for a poor choice of architecture are severe: too few nodes, and the BNN will not be flexible enough to model the function of interest; too many nodes, and the BNN predictions will have large variance. We note that these Bayesian model selection concerns are subtlely different from overfitting and underfitting concerns that arise from maximum likelihood training: here, more expressive models (e.g. those with more nodes) require more data to concentrate the posterior. When there is insufficent data, the posterior uncertainty over the BNN weights will remain large, resulting in large variances in the BNN's predictions. We illustrate this issue in Figure 1, where we see a BNN trained with too many parameters has higher variance around its predictions than one with fewer. Thus, the core concern of Bayesian model selection is to identify a model class expressive enough that it can explain the observed data set, but not so expressive that it can explain everything (Rasmussen & Ghahramani, 2001; Murray & Ghahramani, 2005).

Model selection in BNNs is challenging because the number of nodes in a layer is a discrete quantity. Recently, (Ghosh & Doshi-Velez, 2017; Louizos et al., 2017) independently proposed performing model selection in Bayesian neural networks by placing Horseshoe pri-

[1]IBM research, Cambridge, MA, USA [2]MIT-IBM Watson AI Lab [3]Harvard University, Cambridge, MA, USA. Correspondence to: Soumya Ghosh <ghoshso@us.ibm.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

ors (Carvalho et al., 2009) over the weights incident to each node in the network. This prior can be interpreted as a continuous relaxation of a spike-and-slab approach that would assign a discrete on-off variable to each node, allowing for computationally-efficient optimization via variational inference.

In this work, we expand upon this idea with several innovations and careful experiments. Via a combination of using regularized horseshoe priors for the node-specific weights and variational approximations that retain critical posterior structure, we both improve upon the statistical properties of the earlier works and provide improved generalization, especially for smaller data sets and in sample-limited settings such as reinforcement learning. We also present a new thresholding rule for pruning away nodes. Unlike previous work our rule does not require computing a point summary of the inferred posteriors. We compare the various model and inference combinations on a diverse set of regression and reinforcement learning tasks. We find that the proposed innovations consistently improve upon the compactness of the models learned without sacrificing predictive performance.

2. Bayesian Neural Networks

A Bayesian neural network endows the parameters $\mathcal{W}$ of a neural network with distributions $\mathcal{W} \sim p(\mathcal{W})$. When combined with inference algorithms that infer posterior distributions over weights, they are able to capture posterior as well as predictive uncertainties. For the following, consider a fully connected deep neural network with $L-1$ hidden layers, parameterized by a set of weight matrices $\mathcal{W} = \{W_l\}_1^L$, where W_l is of size $\mathbb{R}^{K_{l-1}+1 \times K_l}$, and K_l is the number of units in layer l. The network maps an input $x \in \mathbb{R}^D$ to a response $f(\mathcal{W}, x)$ by recursively applying the transformation $h(W_l^T[z_l^T, 1]^T)$, where $z_l \in \mathbb{R}^{K_l \times 1}$ is the input into layer l, the initial input z_0 is x, and h is a point-wise non-linearity such as the rectified-linear function, $h(a) = \max(0, a)$.

Given N observation response pairs $\mathcal{D} = \{x_n, y_n\}_{n=1}^N$ and $p(\mathcal{W})$, we are interested in the posterior distribution $p(\mathcal{W} \mid \mathcal{D}) \propto \prod_{n=1}^N p(y_n \mid f(\mathcal{W}, x_n))p(\mathcal{W})$, and in using it for predicting responses to unseen data x_*, $p(y_* \mid x_*) = \int p(y_* \mid f(\mathcal{W}, x_*))p(\mathcal{W} \mid \mathcal{D})d\mathcal{W}$. The prior $p(\mathcal{W})$ allows one to encode problem-specific beliefs as well as general properties about weights.

3. Bayesian Neural Networks with Regularized Horseshoe Priors

Let $w_{kl} \in \mathbb{R}^{K_{l-1}+1 \times 1}$ denote the set of all weights incident into unit k of hidden layer l. Ghosh & Doshi-Velez (2017); Louizos et al. (2017) introduce a prior such that each unit's weight vector w_{kl} is conditionally independent and follow a group Horseshoe prior (Carvalho et al., 2009),

$$w_{kl} \mid \tau_{kl}, \upsilon_l \sim \mathcal{N}(0, (\tau_{kl}^2 \upsilon_l^2)\mathbb{I}),$$
$$\tau_{kl} \sim C^+(0, b_0), \quad \upsilon_l \sim C^+(0, b_g). \tag{1}$$

Here, $\mathbb{I}$ is an identity matrix, $a \sim C^+(0, b)$ is the Half-Cauchy distribution with density $p(a|b) = 2/\pi b(1 + (a^2/b^2))$ for $a > 0$, τ_{kl} is a unit specific scale parameter, while the scale parameter υ_l is shared across the layer. This horseshoe prior exhibits Cauchy-like flat, heavy tails while maintaining an infinitely tall spike at zero. As a result, it allows sufficiently large unit weight vectors w_{kl} to escape un-shrunk—by having a large scale parameter—while providing severe shrinkage to small weights. By forcing all weights incident on a unit to share scale parameters, we are able to induce sparsity at the unit level, turning off units that are unnecessary for explaining the data well. Intuitively, the shared layer wide scale υ_l pulls all units in layer l to zero, while the heavy tailed unit specific τ_{kl} scales allow some of the units to escape the shrinkage.

Regularized Horseshoe Priors While the horseshoe prior has some good properties, when the amount of training data is limited, units with essentially no shrinkage can produce large weights can adversely affect generalization performance of HS-BNNs, with minor perturbations of the data leading to vastly different predictions. To deal with this issue, here we consider the regularized horseshoe prior (Piironen & Vehtari, 2017). Under this prior w_{kl} is drawn from,

$$w_{kl} \mid \tau_{kl}, \upsilon_l, c \sim \mathcal{N}(0, (\tilde{\tau}_{kl}^2 \upsilon_l^2)\mathbb{I}), \tilde{\tau}_{kl}^2 = \frac{c^2 \tau_{kl}^2}{c^2 + \tau_{kl}^2 \upsilon_l^2}. \tag{2}$$

Note that for the weight node vectors that are strongly shrunk to zero, we will have tiny $\tau_{kl}^2 \upsilon_l^2$. When, $\tau_{kl}^2 \upsilon_l^2 \ll c^2$, $\tilde{\tau}_{kl}^2 \to \tau_{kl}^2 \upsilon_l^2$, recovering the original horseshoe prior. On the other hand, for the un-shrunk weights $\tau_{kl}^2 \upsilon_l^2$ will be large, and when $\tau_{kl}^2 \upsilon_l^2 \gg c^2$, $\tilde{\tau}_{kl}^2 \to c^2$. Thus, these weights under the regularized Horseshoe prior follow $w_{kl} \sim \mathcal{N}(0, c^2\mathbb{I})$ and c acts as a weight decay hyperparameter. We place a Inv-Gamma(c_a, c_b) prior on c^2. In the experimental section, we find that the regularized HS-BNN does indeed improve generalization over HS-BNN. Below, we describe two essential parametrization considerations essential for using the regularized horseshoe in practice.

Half-Cauchy re-parameterization for variational learning. Instead of directly parameterizing the Half-Cauchy random variables in Equations 1 and 2, we use a convenient auxiliary variable parameterization (Wand et al., 2011) of the distribution, $a \sim C^+(0, b) \iff a^2 \mid \lambda \sim$ Inv-Gamma$(\frac{1}{2}, \frac{1}{\lambda})$; $\lambda \sim$ Inv-Gamma$(\frac{1}{2}, \frac{1}{b^2})$, where $v \sim$ Inv-Gamma(a, b) is the Inverse Gamma distribution with

density $p(v) \propto v^{-a-1}\exp\{-b/v\}$ for $v > 0$. This avoids the challenges posed by the direct approximation during variational learning — standard exponential family variational approximations struggle to capture the thick Cauchy tails, while a Cauchy approximating family leads to high variance gradients.

Since the number of output units is fixed by the problem at hand, a sparsity inducing prior is not appropriate for the output layer. Instead, we place independent Gaussian priors, $w_{kL} \sim \mathcal{N}(0, \kappa^2\mathbb{I})$ with vague hyper-priors $\kappa \sim C^+(0, b_\kappa = 5)$ on the output layer weights. The joint distribution of the regularized Horseshoe Bayesian neural network is then given by,

$$p(\mathcal{D}, \theta) = p(c \mid c_a, c_b) r(\kappa, \rho_\kappa \mid b_\kappa) \prod_{k=1}^{K_L} \mathcal{N}(w_{kL} \mid 0, \kappa\mathbb{I}) \prod_{l=1}^{L} r(\upsilon_l, \vartheta_l \mid b_g) \prod_{k=1}^{K_l} r(\tau_{kl}, \lambda_{kl} \mid b_0) \mathcal{N}(w_{kl} \mid 0, (\tilde{\tau}_{kl}^2 \upsilon_l^2)\mathbb{I}) \quad (3)$$
$$\prod_{n=1}^{N} p(y_n \mid f(\mathcal{W}, x_n)),$$

where $p(y_n|f(\mathcal{W}, x_n))$ is the likelihood function and $r(a, \lambda|b) = \text{Inv-Gamma}(a^2|\frac{1}{2}, \frac{1}{\lambda})\text{Inv-Gamma}(\lambda|\frac{1}{2}, \frac{1}{b^2})$, with $\theta = \{\mathcal{W}, \mathcal{T}, \kappa, \rho_\kappa, c\}$, $\mathcal{T} = \{\{\tau_{kl}\}_{k=1,l=1}^{K,L}, \{\upsilon_l\}_{l=1}^{L}, \{\lambda_{kl}\}_{k=1,l=1}^{K,L}, \{\vartheta_l\}_{l=1}^{L}\}$.

Non-Centered Parameterization The regularized horseshoe (and the horseshoe) prior both exhibit strong correlations between the weights w_{kl} and the scales $\tau_{kl}\upsilon_l$. While their favorable sparsity inducing properties stem from this coupling, it also gives rise to coupled posteriors that exhibit pathological funnel shaped geometries (Betancourt & Girolami, 2015; Ingraham & Marks, 2016) that are difficult to reliably sample or approximate.

Adopting non-centered parameterizations (Ingraham & Marks, 2016), helps alleviate the issue. Consider a reformulation of Equation 2,

$$\beta_{kl} \sim \mathcal{N}(0, \mathbb{I}), \quad w_{kl} = \tilde{\tau}_{kl}\upsilon_l\beta_{kl}, \quad (4)$$

where the distribution on the scales are left unchanged. Since the scales and weights are sampled from independent prior distributions and are *marginally* uncorrelated, such a parameterization is referred to as non-centered. The likelihood is now responsible for introducing the coupling between the two, when conditioning on observed data. Non-centered parameterizations are known to lead to simpler posterior geometries (Betancourt & Girolami, 2015). Empirically (Ghosh & Doshi-Velez, 2017) have shown that adopting a non-centered parameterization significantly improves the quality of the posterior approximation for BNNs with Horseshoe priors. Thus, we also adopt non-centered parameterizations for the regularized Horseshoe BNNs.

4. Structured Variational Learning of Regularized Horseshoe BNNs

We approximate the intractable posterior $p(\theta \mid \mathcal{D})$ with a computationally convenient family. We exploit recently proposed stochastic extensions to scale to both large architectures and datasets, and use black-box variants to deal with non-conjugacy. We begin by selecting a tractable family of distributions $q(\theta \mid \phi)$, with free variational parameters ϕ. Learning involves optimizing ϕ such that the Kullback-Liebler divergence between the approximation and the true posterior, $\text{KL}(q(\theta \mid \phi)||p(\theta \mid \mathcal{D}))$ is minimized. This is equivalent to maximizing the lower bound to the marginal likelihood (or evidence) $p(\mathcal{D})$, $p(\mathcal{D}) \geq \mathcal{L}(\phi) = \mathbb{E}_{q_\phi}[\ln p(\mathcal{D}, \theta)] + \mathbb{H}[q(\theta \mid \phi)]$. The choice of the approximating family governs the quality of inference.

4.1. Variational Approximation Choices

The more flexible the approximating family the better it approximates the true posterior. Below, we first describe a straight-forward fully-factored approximation and then a more sophisticated structured approximation that we demonstrate has better statistical properties.

Fully Factorized Approximations The simplest possibility is to use a fully factorized variational family,

$$q(\theta \mid \phi) = \prod_{a \in \{c, \kappa, \rho_\kappa\}} q(a \mid \phi_a) \prod_{i,j,l} q(\beta_{ij,l} \mid \phi_{\beta_{ij,l}}) \prod_{k,l} q(\tau_{kl} \mid \phi_{\tau_{kl}}) q(\lambda_{kl} \mid \phi_{\lambda_{kl}}) \prod_{l} q(\upsilon_l \mid \phi_{\upsilon_l}) q(\vartheta_l \mid \phi_{\vartheta_l}). \quad (5)$$

Restricting the variational distribution for the non-centered weight $\beta_{ij,l}$ between units i in layer $l-1$ and j in layer l, $q(\beta_{ij,l} \mid \phi_{\beta_{ijl}})$ to the Gaussian family $\mathcal{N}(\beta_{ij,l} \mid \mu_{ij,l}, \sigma_{ij,l}^2)$, and the non-negative scale parameters τ_{kl}^2 and υ_l^2 and the variance of the output layer weights to the log-Normal family, $q(\ln \tau_{kl}^2 \mid \phi_{\tau_{kl}}) = \mathcal{N}(\mu_{\tau_{kl}}, \sigma_{\tau_{kl}}^2)$, $q(\ln \upsilon_l^2 \mid \phi_{\upsilon_l}) = \mathcal{N}(\mu_{\upsilon_l}, \sigma_{\upsilon_l}^2)$, and $q(\ln \kappa^2 \mid \phi_\kappa) = \mathcal{N}(\mu_\kappa, \sigma_\kappa^2)$, allows for the development of straightforward inference algorithms (Ghosh & Doshi-Velez, 2017; Louizos et al., 2017). It is not necessary to impose distributional constraints on the variational approximations of the auxiliary variables ϑ_l, λ_{kl}, or ρ_κ. Conditioned on the other variables the optimal variational family for these latent variables follow inverse Gamma distributions. We refer to this approximation as the *factorized* approximation.

Parameter-tied factorized approximation. The conditional variational distribution on w_{kl} implied by Equations 5 and 7 is $q(w_{kl} \mid \tau_{kl}, \upsilon_l) = \mathcal{N}(w_{kl} \mid \tau_{kl}\upsilon_l\mu_{kl}, (\tau_{kl}\upsilon_l)^2\Psi)$, where Ψ is a diagonal matrix with elements populated by $\sigma_{ij,l}^2$ and μ_{kl} consists of the corresponding variational means $\mu_{ij,l}$. The distributions of weights incident into a unit are thus coupled through $\tau_{kl}\upsilon_l$ while all weights

in a layer are coupled through the layer wise scale υ_l. This view suggests that using a simpler approximating family $q(\beta_{ij,l} \mid \phi_{\beta_{ijl}}) = \mathcal{N}(\beta_{ij,l} \mid \mu_{ij,l}, 1)$ results in an isotropic Gaussian approximation $q(w_{kl} \mid \tau_{kl}, \upsilon_l) = \mathcal{N}(w_{kl} \mid \tau_{kl}\upsilon_l\mu_{kl}, (\tau_{kl}\upsilon_l)^2\mathbb{I})$. Crucially, the scale parameters $\tau_{kl}\upsilon_l$ still allow for pruning of units when the scales approach zero. Moreover, by tying the variances of the non-centered weights together this approximation effectively halves the number of variational parameters and speeds up training (Ghosh & Doshi-Velez, 2017). We call this the *tied-factorized* approximation.

Structured Variational Approximations Although computationally convenient, the factorized approximations fail to capture posterior correlations among the network weights, and more pertinently, between weights and scales.

We take a step towards a more structured variational approximation by using a layer-wise matrix variate Gaussian variational distribution for the non-centered weights and retaining the form of all the other factors from Equation 5. Let $\beta_l \in \mathbb{R}^{K_{l-1}+1 \times K_l}$ denote the set of weights betweens layers $l-1$ and l, then under this variational approximation we have $q(\beta_l \mid \phi_{\beta_l}) = \mathcal{MN}(\beta_l \mid M_{\beta_l}, U_{\beta_l}, V_{\beta_l})$, where $M_{\beta_l} \in \mathbb{R}^{K_{l-1}+1 \times K_l}$ is the mean, $V_{\beta_l} \in \mathbb{R}^{K_l \times K_l}$ and $U_{\beta_l} \in \mathbb{R}^{K_{l-1}+1 \times K_{l-1}+1}$ capture the covariances among the columns and rows of β_l, thereby modeling dependencies among the variational approximation to the weights in a layer. Louizos & Welling (2016) demonstrated that even when U_{β_l} and V_{β_l} are restricted to be diagonal, the matrix Gaussian approximation can lead to significant improvements over fully factorized approximations for vanilla BNNs. We call this the *semi-structured*[1] approximation.

The horseshoe prior exhibits strong correlations between weights and their scales, which encourages strong posterior coupling between β_{kl} and τ_{kl}. For effective shrinkage towards zero, it is important that the variational approximations are able to capture this strong dependence. To do so, let $B_l = \begin{bmatrix} \beta_l \\ \nu_l^T \end{bmatrix}$, $\nu_l = [\nu_{1l}, \ldots, \nu_{K_l l}]^T$, and $\nu_{kl} = \ln\tau_{kl}$. Now using the variational approximation $q(B_l \mid \phi_{B_l}) = \mathcal{MN}(B_l \mid M_l, U_l, V_l)$, allows us to retain the coupling between weights incident into a unit and the corresponding unit specific scales, with appropriate parameterizations of U_l. In particular, we note that a diagonal U_l fails to capture the necessary correlations, and defeats the purpose of using a matrix Gaussian variational family to model the posterior of B_l. To retain computational efficiency while capturing dependencies among the rows of B_l we enforce a low-rank structure, $U_l = \Psi_l + h_l h_l^T$, where $\Psi_l \in \mathbb{R}^{K_{l-1}+2 \times K_{l-1}+2}$ is a diagonal matrix and $h_l \in \mathbb{R}^{K_{l-1}+2 \times 1}$ is a column vector. We retain a diagonal structure for $V_l \in \mathbb{R}^{K_l \times K_l}$. We call this approximation the *structured* approximation. In the experimental section, we find that this structured approximation, indeed leads to stronger shrinkage towards zero in the recovered solutions. When combined with a pruning rule, it significantly compresses networks with excess capacity. Table 1 summarizes the variational approximations introduced in this section.

[1] it captures correlations among weights but not between weights and scales

Table 1. Variational Approximation Families.

APPROXIMATION	DESCRIPTION
FACTORIZED	$q(\nu_l \mid \phi_{\nu_l})q(\beta_l \mid \phi_{\beta_l}) = \prod_{i,j,l} \mathcal{N}(\beta_{kl} \mid \mu_{ij,l}, \sigma^2_{ij,l}) \prod_{k,l} q(\nu_{kl} \mid \phi_{\nu_{kl}})$
FACTORIZED (TIED)	$q(\nu_l \mid \phi_{\nu_l})q(\beta_l \mid \phi_{\beta_l}) = \prod_{i,j,l} \mathcal{N}(\beta_{kl} \mid \mu_{ij,l}, 1) \prod_{k,l} q(\nu_{kl} \mid \phi_{\nu_{kl}})$
SEMI-STRUCTURED	$q(\nu_l \mid \phi_{\nu_l})q(\beta_l \mid \phi_{\beta_l}) = \mathcal{MN}(\beta_l \mid M_{\beta_l}, U_{\beta_l}, V_{\beta_l}) \prod_{k,l} q(\nu_{kl} \mid \phi_{\nu_{kl}})$
STRUCTURED	$q(\beta_l, \nu_l \mid \phi_{B_l}) = \mathcal{MN}(B_l \mid M_l, U_l, V_l)$

4.2. Black Box Variational Inference

Irrespective of the variational family choice, the resulting evidence lower bound (ELBO),

$$\begin{aligned} \mathcal{L}(\phi) = \sum_n \mathbb{E}[\ln p(y_n \mid f(\beta, \mathcal{T}, \kappa, x_n))] + \\ \mathbb{E}[\ln p(\mathcal{T}, \beta, \kappa, \rho_\kappa \mid b_0, b_g, b_\kappa)] + \mathbb{H}[q(\theta \mid \phi)], \end{aligned} \tag{6}$$

is challenging to evaluate. Here we have used β to denote the set of all non-centered weights in the network. The non-linearities introduced by the neural network and the potential lack of conjugacy between the neural network parameterized likelihoods and the Horseshoe priors render the first expectation in Equation 6 intractable.

Recent progress in black box variational inference (Kingma & Welling, 2014; Rezende et al., 2014; Ranganath et al., 2014; Titsias & Lázaro-gredilla, 2014) subverts this difficulty. These techniques compute noisy unbiased estimates of the gradient $\nabla_\phi \hat{\mathcal{L}}(\phi)$, by approximating the offending expectations with unbiased Monte-Carlo estimates and relying on either score function estimators (Williams, 1992; Ranganath et al., 2014) or reparameterization gradients (Kingma & Welling, 2014; Rezende et al., 2014; Titsias & Lázaro-gredilla, 2014) to differentiate through the sampling process. With the unbiased gradients in hand, stochastic gradient ascent can be used to optimize the ELBO. In practice, reparameterization gradients exhibit significantly lower variances than their score function counterparts and are typically favored for differentiable models. The reparameterization gradients rely on the existence of a parameterization that separates the source of randomness from the parameters with respect to which the gradients are sought. For our Gaussian variational approximations, the well known non-centered parameterization, $\zeta \sim \mathcal{N}(\mu, \sigma^2) \Leftrightarrow \epsilon \sim \mathcal{N}(0, 1), \zeta = \mu + \sigma\epsilon$, allows us

to compute Monte-Carlo gradients,

$$\nabla_{\mu,\sigma}\mathbb{E}_{q_w}[g(w)] \Leftrightarrow \nabla_{\mu,\sigma}\mathbb{E}_{\mathcal{N}(\epsilon|0,1)}[g(\mu+\sigma\epsilon)] \approx \frac{1}{S}\sum_s \nabla_{\mu,\sigma} g(\mu+\sigma\epsilon^{(s)}), \tag{7}$$

for any differentiable function g and $\epsilon^{(s)} \sim \mathcal{N}(0,1)$. Furthermore, all practical implementations of variational Bayesian neural networks use a further re-parameterization to lower variance of the gradient estimator. They sample from the implied variational distribution over a layer's pre-activations instead of directly sampling the much higher dimensional weights (Kingma et al., 2015).

Variational distribution on pre-activations The "local" re-parametrization is straightforward for all the approximations except the structured approximation. For that, observe that $q(B_l \mid \phi_{B_l})$ factorizes as $q(\beta_l \mid \nu_l, \phi_{\beta_l})q(\nu_l \mid \phi_{\nu_l})$. Moreover, conditioned on $\nu_l \sim q(\nu_l \mid \phi_{\nu_l})$, β_l follows another matrix Gaussian distribution. The conditional variational distribution is $q(\beta_l \mid \nu_l, \phi_{\beta_l}) = \mathcal{MN}(M_{\beta_l|\nu_l}, U_{\beta_l|\nu_l}, V)$. It then follows that $b = \beta_l^T a$ for an input $a \in \mathbb{R}^{K_{l-1}+1\times 1}$ into layer l, is distributed as,

$$b \mid a, \nu_l, \phi_{\beta_l} \sim \mathcal{N}(b \mid \mu_b, \Sigma_b), \tag{8}$$

with $\mu_b = M^T_{\beta_l|\nu_l} a$, and $\Sigma_b = (a^T U_{\beta_l|\nu_l} a)V$. Since, $a^T U_{\beta_l|\nu_l} a$ is scalar and V is diagonal, Σ is diagonal as well. For regularized HS-BNN, recall that the pre-activation of node k in layer l, is $u_{kl} = \tilde{\tau}_{kl}\upsilon_l b$, and the corresponding variational posterior is,

$$q(u_{kl} \mid \mu_{u_{kl}}, \sigma^2_{u_{kl}}) = \mathcal{N}(u_{kl} \mid \mu_{u_{kl}}, \sigma^2_{u_{kl}}), \quad \mu_{u_{kl}} = \tilde{\tau}^{(s)}_{kl}\upsilon^{(s)}_l \mu_{bk}; \quad \sigma^2_{u_{kl}} = \tilde{\tau}^{(s)^2}_{kl}\upsilon^{(s)^2}_l \Sigma_{bk,k}, \tag{9}$$

where $\tau^{(s)}_{kl}$, $\upsilon^{(s)}_l, c^{(s)}$ are samples from the corresponding log-Normal posteriors and $\tilde{\tau}^{(s)}_{kl}$ is constructed as $c^{(s)^2}\tau^{(s)^2}_{kl}/(c^{(s)^2} + \tau^{(s)^2}_{kl}\upsilon^{(s)^2}_l)$.

Algorithm We now have a simple prescription for optimizing Equation 6. Recursively sampling the variational posterior of Equation 9 for each layer of the network, allows us to forward propagate information through the network. Using the reparameterizations (Equation 7), allows us to differentiate through the sampling process. We compute the necessary gradients through reverse mode automatic differentiation tools (Maclaurin et al., 2015). With the gradients in hand, we optimize $\mathcal{L}(\phi)$ with respect to the variational weights ϕ_B, per-unit scales $\phi_{\tau_{kl}}$, per-layer scales ϕ_{υ_l}, and the variational scale for the output layer weights, ϕ_κ using Adam (Kingma & Ba, 2014). Conditioned on these, the optimal variational posteriors of the auxiliary variables ϑ_l, λ_{kl}, and ρ_κ follow Inverse Gamma distributions. Fixed point updates that maximize $\mathcal{L}(\phi)$ with respect to $\phi_{\vartheta_l}, \phi_{\lambda_{kl}}, \phi_{\rho_\kappa}$, holding the other variational parameters fixed are available. It can be shown that, $q(\lambda_{kl} \mid \phi_{\lambda_{kl}}) = \text{Inv-Gamma}(\lambda_{kl} \mid 1, \mathbb{E}[\frac{1}{\tau_{kl}}] + \frac{1}{b_0^2})$. The distributions of the other auxiliary variables are analogous. By alternating between gradient and fixed point updates to maximize the ELBO in a coordinate ascent fashion we learn all variational parameters jointly (see Algorithm 1 of the supplement). Further details are available in the supplement.

Computational Considerations The primary computational bottleneck for the structured approximation arises in computing the pre-activations in equation 8. While computing Σ_b in the factorized approximation involves a single inner product, in the structured case it requires the computation of the quadratic form $a^T U_{M_{\beta_l|\nu_l}} a$ and a point wise multiplication with the elements of V_l. Owing to the diagonal plus rank-one structure of $U_{M_{\beta_l|\nu_l}}$, we only need two inner products, followed by a scalar squaring and addition to compute the quadratic form and K_l scalar multiplications for the point-wise multiplication with V_l. Thus the structured approximation is only marginally more expensive. Further, it uses only $K_l + 2 \times (K_{l-1} + 1)$ weight variance parameters per layer, instead of $K_l \times (K_{l-1} + 1)$ parameters used by the factorized approximation. Not having to compute gradients and update these additional parameters further mitigates the performance difference.

4.3. Pruning Rule

The group Horseshoe and its regularized variant provide strong shrinkage towards zero for small w_{kl}. However, since we infer a posterior distribution the shrunk weights, although tiny, are never actually zero. A user-defined thresholding rule is required to prune away the shrunk weights. In the past, Louizos et al. (2017) have first summarized the inferred posterior distributions using a point estimate and then used the point summary to define a thresholding rule. Here, we propose an alternate thresholding rule that does not rely on a point summary of the posterior. We prune away a unit, if $p(\tau_{kl}\upsilon_l < \delta) > p_0$, where δ and p_0 are user defined parameters, with $\tau_{kl} \sim q(\tau_{kl} \mid \phi_{\tau_{kl}})$ and $\upsilon_l \sim q(\upsilon_l \mid \phi_{\upsilon_l})$. To see why this rule is sensible, recall that for units which experience strong shrinkage the regularized Horseshoe tends to the Horseshoe. Under the Horseshoe prior, $\tau_{kl}\upsilon_l$ governs the (non-negative) scale of the weight node vector w_{kl}. Therefore, under our thresholding rule, we prune away nodes whose posterior scales, place probability greater than p_0 below a sufficiently small threshold δ. In our experiments, we set $p_0 = 0.9$ and δ to either $1e-3$ or $1e-5$. Further, under our variational approximations, $\tau_{kl}\upsilon_l$ follows a log-Normal distribution. Evaluating the thresholding rule is as simple as evaluating the log-Normal cumulative distribution.

5. Related Work

Bayesian neural networks have a long history. Early work can be traced back to (Buntine & Weigend, 1991; MacKay, 1992; Neal, 1993). These early approaches do not scale well to modern architectures or the large datasets required to learn them. Recent advances in stochastic MCMC methods (Li et al., 2016; Welling & Teh, 2011) and stochastic variational methods (Blundell et al., 2015; Rezende et al., 2014), black-box variational and alpha-divergence minimization (Hernandez-Lobato et al., 2016; Ranganath et al., 2014), and probabilistic backpropagation (Hernández-Lobato & Adams, 2015) have reinvigorated interest in BNNs by allowing scalable inference.

Work on learning structure in BNNs has received less attention. (Blundell et al., 2015) introduce a mixture-of-Gaussians prior on the weights, with one mixture tightly concentrated around zero, thus approximating a spike and slab prior over weights. Others (Kingma et al., 2015; Gal & Ghahramani, 2016) have noticed connections between Dropout (Srivastava et al., 2014) and approximate variational inference. In particular, (Molchanov et al., 2017) show that the interpretation of Gaussian dropout as performing variational inference in a network with log uniform priors over weights leads to sparsity in weights. The goal of turning off edges is very different than the approach considered here, which performs model selection over the appropriate number of nodes. More closely related to us, are the recent works of (Ghosh & Doshi-Velez, 2017) and (Louizos et al., 2017). The authors consider group Horseshoe priors for unit pruning. We improve upon these works by using regularized Horseshoe priors that improve generalization, structured variational approximations that provide more accurate inferences, and by proposing a new thresholding rule to prune away units with small scales. Yet others (Neklyudov et al., 2017) have proposed pruning units via truncated log-normal priors over unit scales. However, they do not place priors over network weights and are unable to infer posterior uncertainty over weights, which may lead to poorer predictive uncertainties. In related but orthogonal research (Adams et al., 2010; Song et al., 2017) focused on the problem of structure learning in deep belief networks.

There is also a body of work on learning structure in non-Bayesian neural networks. Early work (LeCun et al., 1990; Hassibi et al., 1993) pruned networks by analyzing second-order derivatives of the objectives. More recently, (Wen et al., 2016) describe applications of structured sparsity not only for optimizing filters and layers but also computation time. Closer to our work in spirit, (Ochiai et al., 2016), (Scardapane et al., 2017; Alvarez & Salzmann, 2016) and (Murray & Chiang, 2015) who use group sparsity to prune groups of weights—e.g. weights incident to a node. However, these approaches don't model the uncertainty in weights and provide uniform shrinkage to all parameters. Our approach similarly provides group shrinkage while retaining weight uncertainties.

6. Experiments

In this section, we present experiments that evaluate various aspects of the proposed regularized Horseshoe Bayesian neural network (reg-HS) and the structured variational approximation. In all experiments, we use a learning rate of 0.005, the global horseshoe scale $b_g = 10^{-5}$, a batch size of 128, $c_a = 2$, and $c_b = 6$. For the structured approximation, we also found that constraining Ψ, V, and h to unit-norms resulted in better predictive performance. Additional experimental details are in the supplement.

Regularized Horseshoe Priors provide consistent benefits, especially on smaller data sets. We begin by comparing reg-HS against BNNs using the standard Horseshoe (HS) prior on a collection of diverse datasets from the UCI repository. We follow the protocol of (Hernández-Lobato & Adams, 2015) to compare the two models. To provide a controlled comparison, and to tease apart the effects of model versus inference enhancements we employ factorized variational approximations for either model. In figure 2, the UCI datasets are sorted from left to right, with the smallest on the left. We find that the regularized Horseshoe leads to consistent improvements in predictive performance. As expected, the gains are more prominent for the smaller datasets for which the regularization afforded by the regularized Horseshoe is crucial for avoiding overfitting. In the remainder, all reported experimental results use the reg-HS prior.

Structured variational approximations provide greater shrinkage. Next, we evaluate the effect of utilizing structured variational approximations. In preliminary experiments, we found that of the approximations described in Section 4.1, the structured approximation outperformed the semi-structured variant while the factorized approximation provided better predictive performance than the tied approximation. In this section we only report results comparing models employing these two variational families.

Toy Data First, we explore the effects of structured and factorized variational approximations on predictive uncertainties. Following (Ghosh & Doshi-Velez, 2017) we consider a noisy regression problem: $y = sin(x) + \epsilon, \epsilon \sim \mathcal{N}(0, 0.1)$, and explore the relationship between predictive uncertainty and model capacity. We compare a single layer 1000 unit BNN using a standard normal prior against BNNs with the regularized horseshoe prior utilizing factorized and structured variational approximations. Figures 1 and 3 show that while a BNN severely over-estimates the predictive uncertainty, models using the reg-HS priors by pruning away excess capacity, significantly improve the estimated uncer-

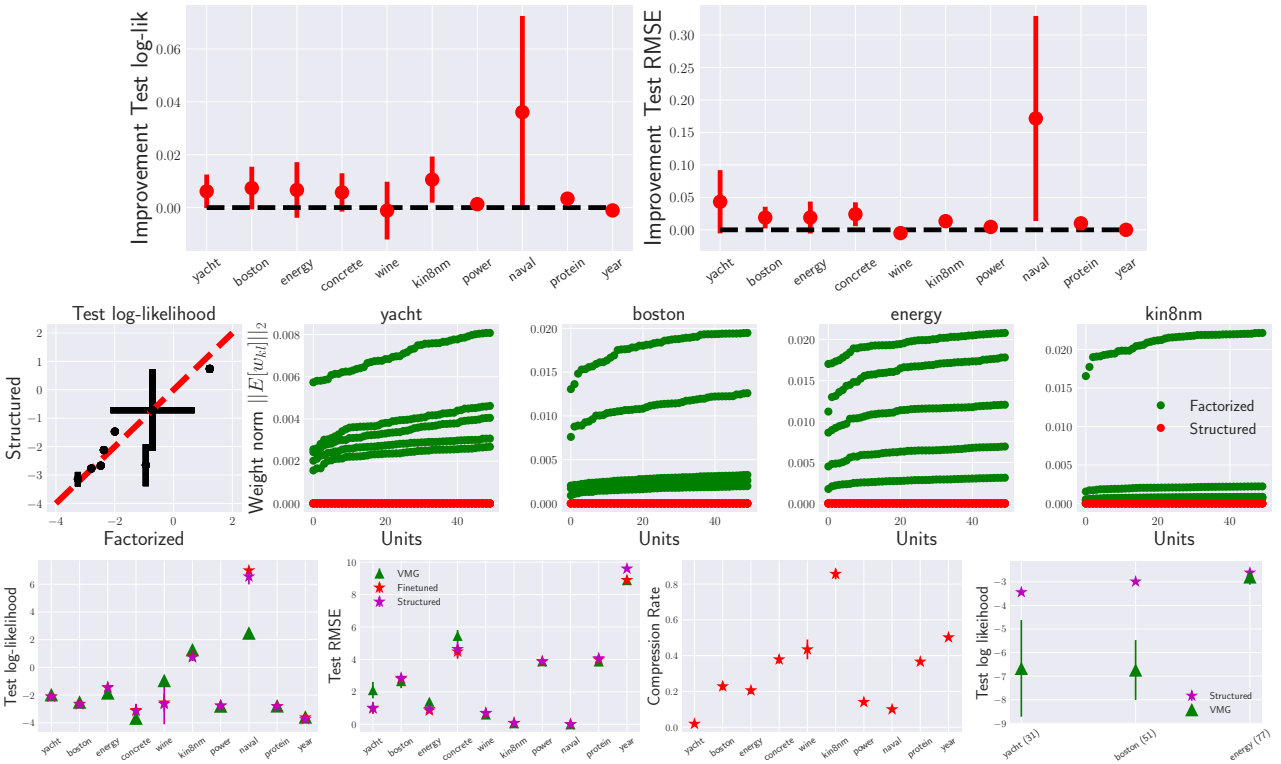

Figure 2. Top: Regularized Horseshoe results in consistent improvements over the vanilla horseshoe prior. The datasets are sorted according to their size with 'yacht' being the smallest and 'year' being the largest. Relative improvement is defined as $(x - y)/max(|x|, |y|)$. *Middle*: Structured variational approximations result in similar predictive performance but consistently recover solutions that exhibit stronger shrinkage. The left most figure plots the predictive log likelihoods achieved by the two approximations, each point corresponds to a UCI dataset. We also plot the fifty units with the smallest $||E[w_{kl}]||_2$, on a number of datasets. Each point in the plot displays the inferred $||E[w_{kl}]||_2$ for a unit in the network. We plot recovered expected weight norms from all five random trials for both the factorized and structured approximation. The structured approximation (in red) consistently provides stronger shrinkage. The factorized approximation both produces weaker shrinkage and the degree of shrinkage exhibits higher variance with random trials. *Bottom*: The structured approximation is competitive with VMG while using much smaller networks. Fine tuning occasionally leads to small improvements. Compression rates are defined as the fraction of un-pruned units. The rightmost plot compares VMG and reg-HS BNN in small data regimes on the three smallest UCI datasets. In parenthesis we indicate the number of training instances. The shrinkage afforded by reg-HS leads to improved performance over VMG which employs priors that lack shrinkage towards zero.

tainty. Furthermore, we observe that the structured approximation best alleviates the under-fitting issues.

Controlled comparisons on UCI benchmarks We return to the UCI benchmark to carefully vet the different variational approximations. We deviate from prior work, by using networks with significantly more capacity than previously considered for this benchmark. In particular, we use single layer networks with an order of magnitude more hidden units (500) than considered in previous work (50). This additional capacity is more than that needed to explain the UCI benchmark datasets well. With this experimental setup, we are able to evaluate how well the proposed methods perform at pruning away extra modeling capacity. For all but the 'year' dataset, we report results from five trials each trained on a random $90/10$ split of the data. For the large year dataset, we ran a single trial (details in the supplement). Figure 2 shows consistently stronger shrinkage.

Comparison against Factorized approximations. The factorized and structured variational approximations have similar predictive performance. However, the structured approximation consistently recovers solutions that exhibit much stronger shrinkage towards zero. Figure 2 demonstrates this effect on several UCI datasets, with more in the supplement. We have plotted 50 units with the smallest $||w_{kl}||_2$ weight norms recovered by the factorized and structured approximations, from five random trials. Both approximations provide shrinkage towards zero, but the structured approximation has significantly stronger shrinkage. Further, the degree of shrinkage from the factorized approximation varies significantly between random initializations. In contrast, the structured approximation *consistently* provides strong shrinkage. We compare the shrinkages using $||\mathbb{E}[w_{kl}]||_2$ instead of applying the pruning rule from section 4.3 and comparing the resulting compression rates. This is because although the scales $\tau_{kl}\upsilon_l$ inferred

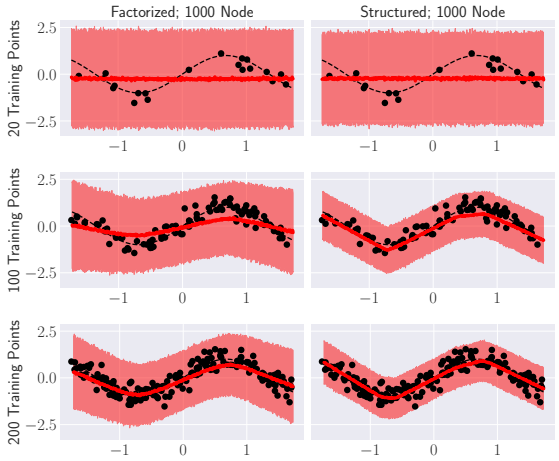

Figure 3. Regularized Horseshoe BNNs prune away excess capacity and are more resistant to underfitting. Variational approximations aware of model structure improve fits.

by the factorized approximation provide a clear separation between signal and noise, they do not exhibit shrinkage toward zero. However, $w_{kl} = \tau_{kl} \upsilon_l \beta_{kl}$ does exhibit shrinkage and provides a fair comparison.

Comparison against competing methods. We compare the reg-HS model with structured variational approximation against the variational matrix Gaussian (VMG) approach of (Louizos & Welling, 2016), which has previously been shown to outperform other variational approaches to learning BNNs. We used the pruning rule with $\delta = 10^{-3}$ for all but the 'year' dataset, for which we set $\delta = 10^{-5}$. Figure 2 demonstrates that structured reg-HS is competitive with VMG in terms of predictive performance. We either perform similarly or better than VMG on the majority of the datasets. More interestingly, structured reg-HS achieves competitive performance while pruning away excess capacity and achieving significant compression. We also fine-tuned the pruned model by updating the weight means while holding others fixed. However, this didn't significantly affect predictive performance. Finally, we evaluate how reg-HS compares against VMG in the low data regime. For the three smallest UCI datasets we use ten percent of the data for training. In such limited data regimes (Figure 2) the shrinkage afforded by reg-HS leads to clear improvements in predictive performance over VMG.

HS-BNNs improve reinforcement learning performance. So far, we have focused on using BNNs simply for prediction. One application area in which having good predictive uncertainty estimates is crucial is in model-based reinforcement learning scenarios (e.g. (Depeweg et al., 2017; Gal et al., 2016b; Killian et al., 2017)): here, it is essential not only to have an estimate of what state an agent may be in after taking a particular action, but also an accurate sense of all the states the agent may end up in. In the following, we apply our regularized HS-BNN with structured approximatins to two domains: the 2D map of Killian et al. (2017) and acrobot Sutton & Barto (1998). In each domain, we collected training samples by training a DDQN online (updated every episode) and an epsilon-greedy policy that started at 1 and decayed to 0.15. This procedure ensured that we had a wide variety of samples that were still biased in coverage toward the optimal policy. To simulate resource constrained scenarios, we limited ourselves to 346 samples for the 2D map and 822 training samples for acrobot. We considered two architectures, a single hidden layer network with 500 units, and a two layer network with 100 units per layer as the transition function for each domain. Then we simulated from each BNN to learn a DDQN policy (2 layers of width 256, 512; learning rate $5e-4$) and tested this policy on the original simulator.

As in our prediction results, training a moderately-sized BNN with so few data results in severe underfitting, which in turn, adversely affects the quality of the policy that is learned. We see in table 2 that the better fitting of the structured reg-HS-BNN results in higher task performance, across domains and model architectures.

Table 2. Model-based reinforcement learning. The under-fitting of the standard BNN results in lower task performance, whereas the HS-BNN is more robust to this underfitting.

	2D Map	
	Test RMSE	Avg. Reward
BNN x-500-y	0.187	975.386
BNN x-100-100-y	0.089	966.716
Structured x-500-y	**0.058**	**995.416**
Structured x-100-100-y	0.061	992.893
	Acrobot	
BNN x-500-y	0.924	-156.573
BNN x-100-100-y	0.710	-23.419
Structured x-500-y	**0.558**	-108.443
Structured x-100-100-y	0.656	**-17.530**

7. Discussion and Conclusion

We demonstrated that the regularized horseshoe prior, combined with a structured variational distribution, is a computationally efficient tool for model selection in Bayesian neural networks. By retaining crucial posterior dependencies, the structured approximation provided, to our knowledge, state of the art shrinkage for BNNs while being competitive in predictive performance to existing approaches. We found, model re-parameterizations — decomposition of the Half-Cauchy priors into inverse gamma distributions and non-centered representations essential for avoiding poor local optima. There remain several interesting follow-on directions, including, modeling enhancements that use layer, node, or even weight specific weight decay c, or layer specific global shrinkage parameter b_g to provide different levels of shrinkage to different parts of the BNN.

References

Adams, R. P., Wallach, H. M., and Ghahramani, Z. Learning the structure of deep sparse graphical models. In *AISTATS*, 2010.

Alvarez, J. M. and Salzmann, M. Learning the number of neurons in deep networks. In *NIPS*, pp. 2270–2278, 2016.

Betancourt, M. and Girolami, M. Hamiltonian monte carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, 79:30, 2015.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. In *Proceedings of the 32nd ICML (ICML-15)*, pp. 1613–1622, 2015.

Buntine, W. L. and Weigend, A. S. Bayesian backpropagation. *Complex systems*, 5(6):603–643, 1991.

Carvalho, C. M., Polson, N. G., and Scott, J. G. Handling sparsity via the horseshoe. In *AISTATS*, 2009.

Depeweg, S., Hernández-Lobato, J. M., Doshi-Velez, F., and Udluft, S. Learning and policy search in stochastic dynamical systems with bayesian neural networks. *ICLR*, 2017.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

Gal, Y., Islam, R., and Ghahramani, Z. Deep Bayesian active learning with image data. In *Bayesian Deep Learning workshop, NIPS*, 2016a.

Gal, Y., McAllister, R., and Rasmussen, C. E. Improving pilco with bayesian neural network dynamics models. In *Data-Efficient Machine Learning workshop, ICML*, 2016b.

Ghosh, S. and Doshi-Velez, F. Model selection in bayesian neural networks via horseshoe priors. *NIPS Workshop on Bayesian Deep Learning*, 2017.

Hassibi, B., Stork, D. G., and Wolff, G. J. Optimal brain surgeon and general network pruning. In *Neural Networks, 1993., IEEE Intl. Conf. on*, pp. 293–299. IEEE, 1993.

Hernandez-Lobato, J., Li, Y., Rowland, M., Bui, T., Hernández-Lobato, D., and Turner, R. Black-box alpha divergence minimization. In *ICML*, pp. 1511–1520, 2016.

Hernández-Lobato, J. M. and Adams, R. P. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *ICML*, 2015.

Ingraham, J. B. and Marks, D. S. Bayesian sparsity for intractable distributions. *arXiv:1602.03807*, 2016.

Killian, T. W., Daulton, S., Doshi-Velez, F., and Konidaris, G. Robust and efficient transfer learning with hidden parameter markov decision processes. In *NIPS*, pp. 6251–6262, 2017.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Stochastic gradient VB and the variational auto-encoder. In *ICLR*, 2014.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *NIPS*, 2015.

LeCun, Y., Denker, J. S., and Solla, S. A. Optimal brain damage. In *NIPS*, pp. 598–605, 1990.

Li, C., Chen, C., Carlson, D. E., and Carin, L. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *AAAI*, 2016.

Louizos, C. and Welling, M. Structured and efficient variational deep learning with matrix Gaussian posteriors. In *ICML*, pp. 1708–1716, 2016.

Louizos, C., Ullrich, K., and Welling, M. Bayesian compression for deep learning. *NIPS*, 2017.

MacKay, D. J. A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Maclaurin, D., Duvenaud, D., and Adams, R. P. Autograd: Effortless gradients in numpy. In *ICML AutoML Workshop*, 2015.

Molchanov, D., Ashukha, A., and Vetrov, D. Variational dropout sparsifies deep neural networks. *arXiv:1701.05369*, 2017.

Murray, I. and Ghahramani, Z. A note on the evidence and bayesian occam's razor. Technical report, Gatsby Unit, 2005.

Murray, K. and Chiang, D. Auto-sizing neural networks: With applications to n-gram language models. *arXiv:1508.05051*, 2015.

Neal, R. M. Bayesian learning via stochastic dynamics. In *NIPS*, 1993.

Neklyudov, K., Molchanov, D., Ashukha, A., and Vetrov, D. P. Structured bayesian pruning via log-normal multiplicative noise. In *NIPS*, pp. 6778–6787, 2017.

Ochiai, T., Matsuda, S., Watanabe, H., and Katagiri, S. Automatic node selection for deep neural networks using group lasso regularization. *arXiv:1611.05527*, 2016.

Piironen, J. and Vehtari, A. On the hyperprior choice for the global shrinkage parameter in the horseshoe prior. *AISTATS*, 2017.

Ranganath, R., Gerrish, S., and Blei, D. M. Black box variational inference. In *AISTATS*, pp. 814–822, 2014.

Rasmussen, C. E. and Ghahramani, Z. Occam's razor. In *NIPS*, pp. 294–300, 2001.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 31st ICML*, pp. 1278–1286, 2014.

Scardapane, S., Comminiello, D., Hussain, A., and Uncini, A. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

Song, Z., Muraoka, Y., Fujimaki, R., and Carin, L. Scalable model selection for belief networks. In *NIPS*, pp. 4612–4622, 2017.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1): 1929–1958, 2014.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Titsias, M. and Lázaro-gredilla, M. Doubly stochastic variational Bayes for non-conjugate inference. In *ICML*, pp. 1971–1979, 2014.

Wand, M. P., Ormerod, J. T., Padoan, S. A., Fuhrwirth, R., et al. Mean field variational Bayes for elaborate distributions. *Bayesian Analysis*, 6(4):847–900, 2011.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th ICML (ICML-11)*, pp. 681–688, 2011.

Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. Learning structured sparsity in deep neural networks. In *NIPS*, pp. 2074–2082, 2016.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.