Collapsed Variational Bounds for Bayesian Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent interest in learning variational Bayesian Neural Networks (BNNs) focused on constructing new variational posteriors and improving the optimization of variational parameters. Variational methods also allow hyperparameters of BNNs to be learned by optimization of the Evidence Lower Bound (ELBO). This approach has received less attention, even though the performance of variational BNNs is known to be very sensitive to the model hyperparameter setting used. Current practice often fixes the hyperparameters to standard values or tunes a small number of them using heuristics or cross-validation. In this paper, we employ hierarchical BNN models and use collapsed variational bounds, commonly applied to other hierarchical Bayesian models, to derive tighter ELBOs for BNNs. The experiments show two interesting findings. First, the new bounds significantly improve performance of Gaussian mean-field VI applied to BNNs demonstrating that mean-field approximations can work well even in deep models. Second, we find that the ELBO is a good optimization target for learning prior hyperparameters of deep networks.

1 Introduction

Variational methods provide computationally cheap approximations to the posterior over weights in a Bayesian Neural Network as compared to alternatives like MCMC. Research on Variational Inference (VI) for BNNs has seen multiple advances [25, 56, 20, 4, 33] enabling optimization of the Evidence Lower Bound (ELBO) for large neural networks and data sets [50, 54, 11, 12, 57]. However, the under-confidence of mean-field VI (MF-VI) in output space has caused poor adoption of VI in applications, especially in sequential decision making [5, 53, 59]. Arguably, one of the main driving forces improving the performance of VI for BNNs has been deriving better estimators of the gradient of the ELBO [25, 56, 20, 4, 33]. However, improved optimization schemes have been considered mostly for variational parameters, not for the prior hyperparameters of BNNs, which are usually fixed to zero mean isotropic Gaussians [48, 60, 64, 11, 16, 4]. There are conflicting views on whether the hyperparameters of Gaussian priors in BNNs can be learned reliably by maximizing the ELBO (approximate marginal likelihood maximization), with some work advocating this approach [62, 45, 43, 26] and others warning against it [4]. By employing collapsed VI bounds [31, 36, 40, 55] to posteriors over hyperparameters in BNNs, we derive novel objective functions (new ELBOs) for performing VI in BNNs. Applying VI to BNNs using our derived ELBOs leads to two findings: (i) MF-VI performs well in deep models where the hyperparameters of Gaussian priors over weights have been learned, but severely under-fits data in output space if the prior over weights is fixed and, (ii) the derived tighter ELBOs are good optimization targets to learn the hyperparameters of Gaussian priors, even in large networks.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

2 Preliminaries

We consider performing VI in a supervised learning scenario given N data points $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$. We denote a neural network as $f_{\boldsymbol{W}}$, the observation model as $p(\boldsymbol{y}|f_{\boldsymbol{W}}(\boldsymbol{x}))$, where $\boldsymbol{W}$ denotes the vectorized weights and biases of the network, and the likelihood of the parameters as $p(\mathcal{D}|\boldsymbol{W}) = \prod_{i=1}^N p(\boldsymbol{y}_i|f_{\boldsymbol{W}}(\boldsymbol{x}_i))$. We focus on Gaussian mean-field priors (with non-zero means) over weights $p(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) = \mathcal{N}(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$. Similarly, we restrict our attention to mean-field Gaussian approximate posteriors $q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) = \mathcal{N}(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ meaning $w \sim \mathcal{N}(\mu_q, \sigma_q^2)$ for any coordinate w in $\boldsymbol{W}$ (both $\boldsymbol{\mu}_q$ and $\boldsymbol{\sigma}_q^2$ are also vectors in this notation). VI finds an approximate posterior $q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ by minimizing the KL divergence $D_{KL}(q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)||p(\boldsymbol{W}|\mathcal{D}, \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2))$ w.r.t. the variational parameters $\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2$. This is done by optimizing the Evidence Lower Bound (ELBO) [24, 3]:

$$\mathcal{L}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2; \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) = \sum_{i=1}^N \mathbb{E}_{q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)} \log p\big(\boldsymbol{y}_i|f_{\boldsymbol{W}}(\boldsymbol{x}_i)\big) - D_{KL}\big(q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)||p(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)\big). \quad (1)$$

We will denote the first term on RHS as $\mathcal{L}_{data}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$. As $D_{KL}(q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)||p(\boldsymbol{W}|\mathcal{D}, \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)) \geq 0$, ELBO lower bounds log marginal likelihood $\log p(\mathcal{D}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) \geq \mathcal{L}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2; \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ Replacing D_{KL} in the Eq. (1) with the expression for the Gaussian KL divergence results in:

$$\mathcal{L}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2; \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) = \mathcal{L}_{data}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) - \frac{1}{2}\mathbf{1}^T\left[\frac{\boldsymbol{\sigma}_q^2 + \boldsymbol{\mu}_q^2}{\boldsymbol{\sigma}_p^2} - \log\boldsymbol{\sigma}_q^2 + \log\boldsymbol{\sigma}_p^2 - \mathbf{1}\right], \quad (2)$$

where we abuse the notation by defining element-wise operations on vectors and $\mathbf{1}$ denotes the vector of all ones. See [4, 33] for low variance updates for mean-field VI optimizing $\mathcal{L}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2; \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ w.r.t. $\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2$ by reparametrizing $\boldsymbol{W}$. Bayesian predictions are approximated by integrating against the learned variational posterior $q^*(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ as $p(\boldsymbol{y}^*|\boldsymbol{x}^*) \approx \mathbb{E}_{q^*(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)} p(\boldsymbol{y}^*|f_{\boldsymbol{W}}(\boldsymbol{x}^*))$. For BNNs, prior means are most often set to zero, $\boldsymbol{\mu}_p = \mathbf{0}$, and prior variances are set to a constant $\boldsymbol{\sigma}_p = \gamma\mathbf{1}$ (often scaled by the size of a previous layer [48, 9]). A systematic method of finding $\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2$ would optimize the marginal likelihood $\log p(\mathcal{D}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ [43] (II-type MLE). Differentiating $\log p(\mathcal{D}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ gives $\nabla_{\boldsymbol{\mu}_p} \log p(\mathcal{D}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) = \mathbb{E}_{p(\boldsymbol{W}|\mathcal{D}, \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)} \nabla_{\boldsymbol{\mu}_p} \log p(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$, which requiring sampling from the posterior $p(\boldsymbol{W}|\mathcal{D}, \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ every update and cannot be done for large models. Since $\log p(\mathcal{D}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) \geq \mathcal{L}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2; \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$, the ELBO can be used as a proxy to learn $\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2$ [62, 43], and for very simple models maximums of marginal likelihood are also maximums of ELBO [44].

Over-pruning in BNNs. Applying mean-field VI to BNN prunes most of the posterior weights, i.e. approximate posteriors converge to prior [4, 58, 19]. While this is beneficial for compressing the model, pruned weights contribute noise to predictions and make modeling data difficult. In one hidden layer network optimization of the ELBO attempts to stop the noise from pruned weights by setting posterior weights in the output layer as close to 0 as possible [58]. The noise from pruned weights is one of the reasons for increasing under-confidence in output space as the size of the network increases [58, 12, 54, 57, 19, 38]. In Figure 1 we highlight this problem for 1 hidden layer MF-VI BNN with 50 hidden units, ReLU activations, zero-mean Gaussian prior over weights and heteroscedastic Gaussian likelihood learned with 200 random *boston* regression data points. In Figure 1 (middle) we plot test NLL as hidden layer's units are removed in ascending order of average KL penalties of incoming weights (only last 4 units are modeling data). Removing pruned weights improves test NLL, meaning that pruned weights contribute only noise to predictions. This is schematically explained in Figure 1 (a), which shows pruned posterior weights injecting noise. Figure 1 (right) shows distributions of means and log variances of posterior incoming to the hidden units, showing that most of the weights have been pruned (zero mean, log variance -2) and contribute only to predictive noise.

3 Collapsed Variational Bounds for Bayesian Neural Networks

A BNN defined by a prior over weights $p(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ and priors over hyperparameters $p(\boldsymbol{\mu}_p)$, $p(\boldsymbol{\sigma}_p^2)$ represents a hierarchical Bayesian model. In this section we focus on answering the question: are more advanced inference schemes developed for hierarchical models applicable to learn hyperparameters of BNNs? We demonstrate the answer is affirmative, and that applying collapsed variational bounds [31, 36, 40, 22] to BNNs enables a systematic way of deriving novel, tighter lower bounds to evidence

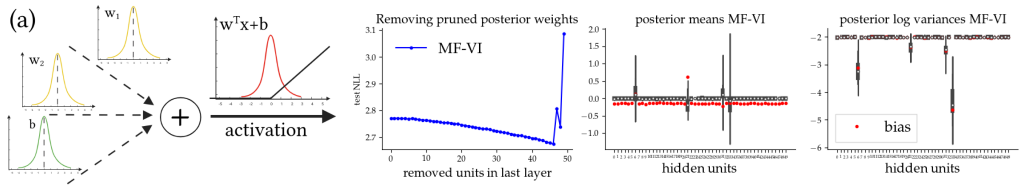

Figure 1: Pruned weights in MF-VI contribute noise to predictions (a). Removing pruned weights can *improve* predictions (middle). Most of the weights are pruned resulting in excessive noise (right).

useful for learning variational posteriors over weights. The idea behind a collapsed bound is a simple: Suppose we maximize $f(x, y)$ and we can derive $x^*(y) = \text{argmax}_x f(x, y)$. Substituting x^* into f results in $f(x^*(y), y)$ which only depends on y, is easier to optimize, and has the property that $f(x^*(y), y) \geq f(x, y)\ \forall_y$. Collapsed variational bounds apply this reasoning to the ELBO with independent posteriors taking the roles of x and y. We defer derivations to Appendix C.

Inferring network hyperparameters. We consider inference over both network weights $\boldsymbol{W}$ and prior hyperparameters $\boldsymbol{\mu}_p$ and $\boldsymbol{\sigma}_p^2$ employing the factorization $q(\boldsymbol{W}, \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) = q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ where for clarity we omit the parametrization of $q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$. The variational principle lower bounds the log marginal likelihood $\log p(\mathcal{D})$ as:

$$\log p(\mathcal{D}) \geq \mathbb{E}_{q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)} \log \frac{p(\mathcal{D}, \boldsymbol{W}, \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)}{q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)} = \mathcal{L}^q\big(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2, q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)\big). \tag{3}$$

Note that learning $q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ does not complicate computing the BNN's predictive distribution $p(\boldsymbol{y}|\boldsymbol{x})$. The bound in Eq. (3) reduces to $\mathcal{L}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ in Eq. (1) when $q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) = q(\boldsymbol{\mu}_p)q(\boldsymbol{\sigma}_p^2)$, and $q(\boldsymbol{\mu}_p)$, $q(\boldsymbol{\sigma}_p^2)$ are delta functions. If we are able to find a distribution $q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ maximizing $\mathcal{L}^q$, we can substitute q^* into Eq. (3) to derive a tighter (collapsed) bound on the marginal likelihood (as opposed to performing coordinate ascent):

$$\log p(\mathcal{D}) \geq \mathbb{E}_{q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)} \log \frac{p(\mathcal{D}, \boldsymbol{W}, \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)}{q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)} = \mathcal{L}^*(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) \geq \mathcal{L}^q\big(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2, q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)\big), \tag{4}$$

where $\mathcal{L}^*$ depends only on the variational parameters $\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2$. The tighter bound in Eq. (4) has been also referred to as a KL corrected bound [31] and marginal VI bound [36, 40]. Collapsed variational bounds are desirable as they have been shown to make learning significantly more efficient [55].

Collapsing the variational bound for BNNs. We now provide a method to derive a collapsed variational bound by maximizing $\mathcal{L}^q(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2, q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2))$ w.r.t. the posterior over network hyperparameters $q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ in BNNs. We use the property that the BNN likelihood term $\log p(\boldsymbol{y}|\boldsymbol{x}, f_{\boldsymbol{W}}(\boldsymbol{x}))$ does not depend on the prior hyperparameters $\boldsymbol{\mu}_p$ and $\boldsymbol{\sigma}_p^2$, i.e. $p(\mathcal{D}|\boldsymbol{W}, \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) = p(\mathcal{D}|\boldsymbol{W})$. Therefore seeking $q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ is significantly simplified. We define,

$$\Phi\big(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2, q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)\big) = \mathbb{E}_{q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)}\Big[\mathbb{E}_{q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)} \log p(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)\Big] - D_{KL}\big(q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)||p(\boldsymbol{\mu}_p)p(\boldsymbol{\sigma}_p^2)\big), \tag{5}$$

which allows us to decompose the bound $\mathcal{L}^q$ in Eq. (3) into:

$$\mathcal{L}^q\big(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2, q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)\big) = \mathcal{L}_{data}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) + \Phi\big(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2, q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)\big) + \mathcal{H}[q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)], \tag{6}$$

where $\mathcal{H}[q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)]$ denotes the differential entropy and only Φ depends on the posterior $q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$. Hence, finding $q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ can be done by maximizing Eq. (5). Note that the objective in Eq. (5) is analogous to Eq. (1), hence optimizing Φ takes the form of a nested VI problem: inferring $q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ where the "data" distribution is replaced by the current variational posterior over weights $q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$. This implies the solutions $q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ are straightforward to derive in closed-form for priors/approximate posteriors from the exponential family. Using these insights, we can write the maximizer $q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) = \text{argmax}_{q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)} \Phi$ as,

$$\log q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2) \propto \log p(\boldsymbol{\mu}_p) + \log p(\boldsymbol{\sigma}_p^2) + \mathbb{E}_{q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)} \log p(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2). \tag{7}$$

Substituting the optimal variational posterior $q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ into Eq. (5) gives:

$$\Phi^*(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) = \mathop{\mathbb{E}}_{q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)}\Big[\mathop{\mathbb{E}}_{q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)} \log p(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)\Big] - D_{KL}\big(q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)||p(\boldsymbol{\mu}_p)p(\boldsymbol{\sigma}_p^2)\big), \tag{8}$$

where we have that $\mathcal{L}^* = \mathcal{L}_{data} + \Phi^* + \mathcal{H}[q(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)]$. Note that the approximate hyperparameters $q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ can be analytically integrated out in many cases, i.e. the first expectation on the RHS in Eq. (8) can be solved in closed-form (we provide examples later in the text). It is important to remember that optimizing $\mathcal{L}^*$ learns both the variational parameters $\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2$ and the posterior over hyperparameters $q^*(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$, with the latter being implicit as we express it as a function of $\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2$ given by Eq. (7). Eventually the optimization target to learn the variational posterior $q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ is given by $\mathcal{L}^*(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) = \mathcal{L}_{data}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) + f(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$. Eliminating $\boldsymbol{\mu}_p$ and $\boldsymbol{\sigma}_p^2$ from $\mathcal{L}^*$ reduces both memory requirement and time per update by half compared to gradient learning of $\boldsymbol{\mu}_p$ and $\boldsymbol{\sigma}_p^2$ (which does not perform well as we show in Section 5). To derive a collapsed bound $\mathcal{L}^*$, enabling learning of the variational posterior for a BNN, we need to complete four steps. Step I: decide upon priors $p(\boldsymbol{\mu}_p), p(\boldsymbol{\sigma}_p^2), p(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ and approximate posteriors $q(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2), q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$. Step II: calculate q^* using Eq. (7). Step III: integrate out q^* in the definition of Φ^* given by Eq. (8) (first expectation on RHS). Step IV: obtain the bound as $\mathcal{L}^* = \mathcal{L}_{data} + \Phi^* + \mathcal{H}[q(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)]$.

Learn prior means, fix prior variances. To exemplify the derived method, we first consider a setting where the prior variances are constant so that $p(\boldsymbol{W}|\boldsymbol{\mu}_p) = \mathcal{N}(\boldsymbol{W}|\boldsymbol{\mu}_p, \gamma\mathbf{1})$ with a Gaussian prior $p(\boldsymbol{\mu}_p) = \mathcal{N}(\boldsymbol{\mu}_p|\mathbf{0}, \alpha\mathbf{1})$. We also employ a mean-field Gaussian posterior $q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ and a Gaussian posterior $q(\boldsymbol{\mu}_p)$ (step I). Substituting into Eq. (7) for every coordinate μ_p we have:

$$q^*(\mu_p) \propto -\frac{(\mu_q - \mu_p)^T(\mu_q - \mu_p)}{2\gamma} - \frac{\mu_p^T \mu_p}{2\alpha}. \tag{9}$$

The above dependency is equivalent to Bayesian inference over μ_p given Gaussian likelihood/prior and observation μ_q, so $q^*(\boldsymbol{\mu}_p) = \mathcal{N}(\boldsymbol{\mu}_p|\frac{\alpha}{\alpha+\gamma}\boldsymbol{\mu}_q, \frac{\alpha+\gamma}{\alpha\gamma}\mathbf{1})$ (step II). Next we substitute $q^*(\boldsymbol{\mu}_p)$ into Φ^* in Eq. (8) and integrate the first expectation on the RHS (step III):

$$\Phi^* \triangleq -\mathop{\mathbb{E}}_{\mathcal{N}(\boldsymbol{\mu}_p|\frac{\alpha}{\alpha+\gamma}\boldsymbol{\mu}_q, \frac{\alpha+\gamma}{\alpha\gamma}\mathbf{1})}\left[\frac{\mathbf{1}^T\boldsymbol{\sigma}_q^2 + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^T(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)}{2\gamma}\right] - \frac{\alpha\boldsymbol{\mu}_q^T\boldsymbol{\mu}_q}{2(\alpha+\gamma)^2} = -\frac{\mathbf{1}^T\boldsymbol{\sigma}_q^2}{2\gamma} - \frac{\boldsymbol{\mu}_q^T\boldsymbol{\mu}_q}{2(\alpha+\gamma)}. \tag{10}$$

Substituting Eq. (10) into $\mathcal{L}^* = \mathcal{L}_{data} + \Phi^* + \mathcal{H}[q(\boldsymbol{W}|\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)]$ results in a new bound (step IV):

$$\mathcal{L}_m^*(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) \triangleq \mathcal{L}_{data}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) - \frac{1}{2\gamma}\left[\mathbf{1}^T\boldsymbol{\sigma}_q^2 + \alpha_{reg}\boldsymbol{\mu}_q^T\boldsymbol{\mu}_q\right] + \frac{1}{2}\left[\mathbf{1}^T\log\boldsymbol{\sigma}_q^2 - \frac{\alpha+\gamma}{\alpha^2\gamma} - \log\frac{\alpha^2\gamma}{\alpha+\gamma}\right], \tag{11}$$

where we defined the hyperparameter $\alpha_{reg} = \gamma/(\gamma+\alpha) \in (0,1)$ and for fixed γ there is 1-to-1 relation between α and α_{reg}. When $\alpha \to 0$ (so $\alpha_{reg} \to 1$) we recover the standard expression for the ELBO given by Eq. (1) with $q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ and prior $\mathcal{N}(\boldsymbol{W}|\mathbf{0}, \gamma\mathbf{1})$. This corresponds to a dogmatic prior $p(\boldsymbol{\mu}_p) = \mathcal{N}(\boldsymbol{\mu}_p|\mathbf{0}, \alpha\mathbf{1})$ converging to a delta spike at 0, and no inference of $\boldsymbol{\mu}_p$. When we allow for inference, the regularization of the (approximate) posterior mean $\boldsymbol{\mu}_q$ decreases. When $\alpha \to \infty$ (uninformative prior) the modeling can be prone to over-fitting due to insufficient regularization in the model specification. Note that this is reflected by the term $-\frac{\alpha+\gamma}{\alpha^2\gamma} - \log\frac{\alpha^2\gamma}{\alpha+\gamma}$ which can be interpreted as Occam's Razor factor [52] penalizing both small and large α as it diverges to negative infinity when $\alpha \to 0$ or $\alpha \to +\infty$. Stated differently, the derived tighter bound $\mathcal{L}_m^*$ inherits the desirable Occam's Razor property of the marginal likelihood $\log p(\mathcal{D}|\alpha)$ penalizing low and high values of α. This enables finding well-performing values of α by evaluating $\mathcal{L}_m^*$ (without having a validation set), as we show in Section 5. We also show in Section 5 that $\alpha_{reg} \in (0.025, 0.1)$ significantly improves upon the predictions of MF-VI with fixed Gaussian priors.

Learn both prior means and variances. We now discuss the scheme learning hyperparameters of both $p(\boldsymbol{W}|\boldsymbol{\mu}_p) = \mathcal{N}(\boldsymbol{W}|\boldsymbol{\mu}_p, \gamma\mathbf{1})$ and $p(\boldsymbol{W}|\boldsymbol{\sigma}_p^2) = \mathcal{N}(\boldsymbol{W}|\mathbf{0}, 1/\boldsymbol{\tau}_p)$. We again consider a mean-field Gaussian posterior $q(\boldsymbol{W}|\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ and employ the following priors: $p(\boldsymbol{\mu}_p|t\boldsymbol{\tau}) = \mathcal{N}(\boldsymbol{\mu}_p|\mathbf{0}, \frac{1}{t\boldsymbol{\tau}})$, $p(\boldsymbol{\tau}) = \mathcal{G}(\boldsymbol{\tau}|\boldsymbol{\alpha}, \boldsymbol{\beta})$. The optimal posteriors q^* are given by $q^*(\boldsymbol{\mu}_p) = \mathcal{N}(\boldsymbol{\mu}_p|\frac{1}{1+t}\boldsymbol{\mu}_q, \frac{1}{(t+1)\boldsymbol{\tau}_p})$ and $q^*(\boldsymbol{\sigma}_p^2) = \mathcal{G}(\boldsymbol{\sigma}_p^2|(\alpha+\frac{1}{2})\mathbf{1}, \beta + \frac{t}{2(t+1)}\boldsymbol{\mu}_q^2 + \frac{1}{2}\boldsymbol{\sigma}_q^2)$. The bound $\mathcal{L}^*(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ becomes,

$$\mathcal{L}_{mv}^*(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) \triangleq \mathcal{L}_{data}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2) - (\alpha + \frac{1}{2})\mathbf{1}^T\log\left[\beta\mathbf{1} + \frac{\delta}{2}\boldsymbol{\mu}_q^2 + \frac{1}{2}\boldsymbol{\sigma}_q^2\right] + \frac{1}{2}\mathbf{1}^T\log\boldsymbol{\sigma}_q^2, \tag{12}$$

where $\delta = t/(1+t)$. This recovers the case of setting prior mean $\boldsymbol{\mu}_p = \mathbf{0}$ and learning only prior variances $\boldsymbol{\sigma}_p^2$ (see Appendix B) when the prior precision $t\boldsymbol{\tau}_p$ over $\boldsymbol{\mu}_p$ goes to ∞, i.e. $t \to \infty$.

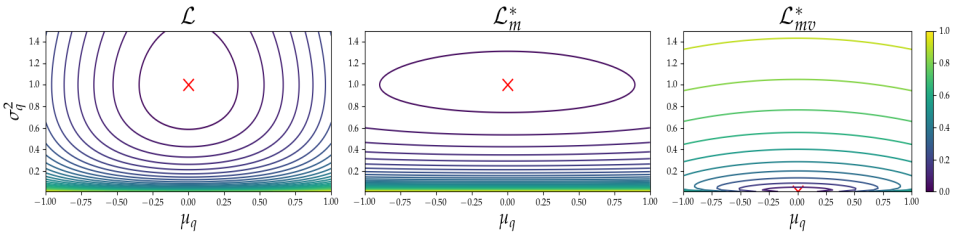

Figure 2: Comparison of regularizing terms in $\mathcal{L}$, $\mathcal{L}_m^*$ and $\mathcal{L}_{mv}^*$ scaled to unity. In the absence of gradients from $\mathcal{L}_{data}$ (approximate) posteriors over weights converge to blue region (red cross-mark). Optimizing $\mathcal{L}$ prunes weights to prior Gaussians which still influence predictions causing under-fitting (see Section 5) and is opposite to optimizing $\mathcal{L}_{mv}^*$ which prunes weights to spiked Gaussians.

Setting $\delta < 1$ in Eq. (12) weakens the regularization of posterior mean $\boldsymbol{\mu}_q$, allowing it to vary more as described previously. The regularizer in Eq. (12) is a decreasing function of μ_q, i.e. without gradients from $\mathcal{L}_{data}$, the posterior mean μ_q of a weight w converges to 0. When $\alpha = \frac{1}{2}$ and $\mu_q = 0$ then $\log(\beta + \frac{1}{2}\sigma_q^2) \geq \frac{1}{2}\log\sigma_q^2 + \frac{1}{2}\log 2\beta$, so when $\mu_q = 0$ and in the absence of gradients from $\mathcal{L}_{data}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$ the posterior variance σ_q^2 converges to 2β. Hence posterior weights which do not experience gradients from $\mathcal{L}_{data}$ converge to $\mathcal{N}(w|0, 2\beta)$ and do not introduce excessive noise to predictions (as opposed to Figure 1). In Section 5 we show $\mathcal{L}_{mv}^*$ also outperforms standard MF-VI.

Comparison of regularizing terms. We compare the regularization terms on $\boldsymbol{\mu}_q$ and $\boldsymbol{\sigma}_q^2$ in the standard mean-field ELBO in Eq. (1) and introduced ELBOs in Eq. (11) and Eq. (12) in Figure 2, where we use prior $p(w) = \mathcal{N}(w|0, 1)$. In all three cases for non-zero posterior means $\boldsymbol{\mu}_q$ regularization attempts to induce non-zero variances $\boldsymbol{\sigma}_q^2$, which are set by striking the balance between the gradients from the data term $\mathcal{L}_{data}$ and regularization of $\boldsymbol{\mu}_q$ and $\boldsymbol{\sigma}_q^2$. In the absence of (or weak) gradients from $\mathcal{L}_{data}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2)$, Gaussian posteriors optimized with Eq. (1) converge to their prior $\mathcal{N}(w|0, 1)$ [4, 58], shown in Figure 2 (left) causing under-fitting. Down-weighting $\boldsymbol{\mu}_q^T\boldsymbol{\mu}_q$ in Eq. (11) weakens the regularization of $\boldsymbol{\mu}_q$ allowing posterior means to vary more, and this can be interpreted as scaling μ_p axis in Figure 2 by $\sqrt{\alpha}$. As we show in Section 5, this enables posterior means μ_q to saturate activations and reduce the noise from pruned weights. Optimizing Eq. (12) is roughly opposite to Eq. (1) and causes weights without data term gradients to be pruned to $\mathcal{N}(0, 2\beta)$ (Figure 2 right, where $\beta = 0.01$). Weights pruned in this way do influence the predictions for small β and this fixes the excessive injection of noise by pruned weights, as we show in Section 5.

4 Related work

Performance of BNNs is known to be sensitive to prior hyperparameters [47, 61, 26], but optimizing hyperparameters in BNNs has not been widely adopted, although there are exceptions [43, 62]. The Laplace approximation has been used for hyperparameters selection in BNNs [42, 26], but it does not use the ELBO. Modeling weights of BNNs in hierarchical fashion has been explored in many works [19, 18, 39, 28, 29, 51, 2, 8]. The closest to our approach is learning a hierarchical horseshoe prior over BNNs weights [19]. However, [19] argues more expressive priors are needed for BNNs, as opposed to showing that Gaussian mean-field priors can provide good predictions. In addition an approach presented in [19] is computationally costly, does not consider collapsed bounds to derive tighter ELBOs and does not learn prior means. There is an active discussion in the community as to whether the true BNN posterior provides satisfactory predictions with work supporting the Bayesian approach [47, 64, 27, 1] and arguing against [60]. Similarly, there are conflicting views on the performance of MF-VI in BNNs with fixed Gaussian priors as some authors claim it can work well [13] and others arguing against [54, 15, 14]. A number of problems of mean-field VI BNNs have been exposed in shallow networks including over-pruning [58] and poor uncertainty in output space [15, 14, 19]. To resolve under-fitting of mean-field BNNs, previous work has re-weighted terms arising from the KL divergence e.g. [38].

5 Experiments

In this section we explore the predictive performance of the introduced variational bounds and benchmark them together with relevant algorithms. We find that using the bounds in Eq. (11), and

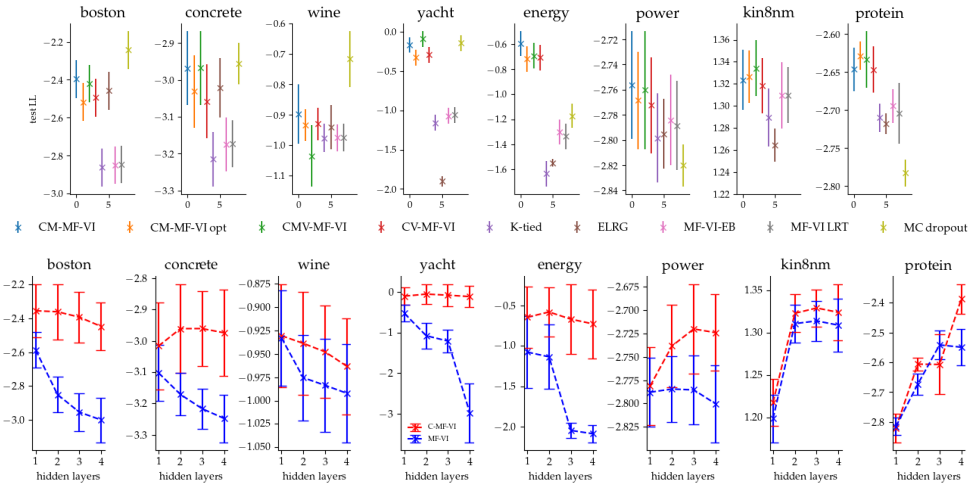

Figure 3: Comparison of collapsed bounds with other algorithms on UCI data sets (top). CM-MF-VI and MF-VI for deeper networks (bottom) where CM-MF-VI avoids severe degradation of predictions.

Eq. (12) significantly improves the predictive performance of standard MF-VI BNNs (fixed Gaussian priors). We also find that learning the prior means, i.e. bound Eq. (11) is more effective and gives sufficiently good predictive performance whereas learning prior variances mostly does not lead to significant gains in the quality of predictions. We explain the reasons for this later in the text and Appendix F. We defer the details of the experimental setup to Appendix D and report additional experimental results in Appendix E. We consider comparing the following algorithms: MF-VI LRT (zero mean mean-field Gaussian priors with fixed variances [48] which we call standard MF-VI) using local reparametrization [33], MF-VI FV learning mean-field Gaussian posterior with fixed variances [58], MF-VI BD mean-field Gaussian VI down-weighting KL penalty by 0.5, MF-VI EB learning prior means with ELBO gradient updates [44, 62], BBB Bayes by Backprop using the ADAM optimizer (as opposed to SGD in [4]), MF-VI K-tied normal distribution [54], ELRG/SLANG learning low rank posteriors proposed in [57, 46]. We denote the algorithm optimizing $\mathcal{L}_m$ as CM-MF-VI, the bound $\mathcal{L}_{mv}$ when $\delta \to 1$ (prior mean $\boldsymbol{\mu}_p$ fixed to $\mathbf{0}$) as CV-MF-VI, the bound $\mathcal{L}_{mv}$ as CMV-MF-VI and CM-MF-VI OPT, which additionally optimizes α_{reg} using the bound in Eq. (11).

	400 units		800 units	
algorithm	test NLL ↓	test ER ↓	test NLL ↓	test ER ↓
CM-MF-VI	$\mathbf{0.047 \pm 0.006}$	$\mathbf{1.34 \pm 0.24}\%$	$\mathbf{0.048 \pm 0.005}$	$\mathbf{1.42 \pm 0.26}\%$
CV-MF-VI	0.068 ± 0.007	$2.13 \pm 0.34\%$	0.066 ± 0.003	$2.09 \pm 0.10\%$
CMV-MF-VI	0.049 ± 0.005	$1.45 \pm 0.25\%$	0.052 ± 0.004	$1.60 \pm 0.01\%$
CM-MF-VI OPT	$\mathbf{0.047 \pm 0.001}$	$1.42 \pm 0.09\%$	$\mathbf{0.047 \pm 0.006}$	$1.49 \pm 0.17\%$
MF-VI LRT [33]	0.094 ± 0.002	$2.42 \pm 0.28\%$	0.099 ± 0.001	$2.58 \pm 0.07\%$
MF-VI BD	0.092 ± 0.002	$2.43 \pm 0.06\%$	0.099 ± 0.004	$2.56 \pm 0.22\%$
MF-VI FV	0.052 ± 0.001	$1.67 \pm 0.13\%$	0.053 ± 0.004	$1.58 \pm 0.26\%$
BBB ADAM [4]	0.095 ± 0.008	$2.48 \pm 0.45\%$	0.097 ± 0.007	$2.49 \pm 0.33\%$
ELRG-VI $K = 5$ [57]	0.053 ± 0.006	$1.54 \pm 0.18\%$	0.058 ± 0.005	$1.68 \pm 0.17\%$
K-TIED $K = 10$ [54]	0.105 ± 0.004	$2.67 \pm 0.16\%$	0.108 ± 0.004	$2.61 \pm 0.17\%$
MF-VI SGD [4]	–	1.82%	–	1.99%
SLANG K=32 [46]	–	1.72%	–	–

Table 1: Test NLL and error rate for vectorized MNIST classification with two hidden layer BNN. CM-MF-VI outperforms other algorithms, collapsed bounds improve upon standard MF-VI.

UCI regression. We first consider 20 train-test splits for 8 UCI regression data sets [10] where we learn 2 hidden layer BNNs (results for 1 hidden layer in Appendix E) with 50 units and ReLU activations as in [23, 17], but we use a heteroscedastic observation model $p(\boldsymbol{y}|f^1_{\boldsymbol{W}}(\boldsymbol{x}), f^2_{\boldsymbol{W}}(\boldsymbol{x})) = \mathcal{N}(\boldsymbol{y}|f^1_{\boldsymbol{W}}(\boldsymbol{x}), \exp(f^2_{\boldsymbol{W}}(\boldsymbol{x})))$, where $f^1_{\boldsymbol{W}}(\boldsymbol{x}), f^2_{\boldsymbol{W}}(\boldsymbol{x})$ are two heads of the network. We optimize the objectives for 200K steps with the ADAM optimizer [32] with default settings. We report a comparison of test log-likelihood in Figure 3 (top). We observe that: the collapsed bounds CM-MF-VI, CV-MF-VI, CMV-MF-VI outperform MF-VI on all data (only CMV-MF-VI worse on wine), (ii)

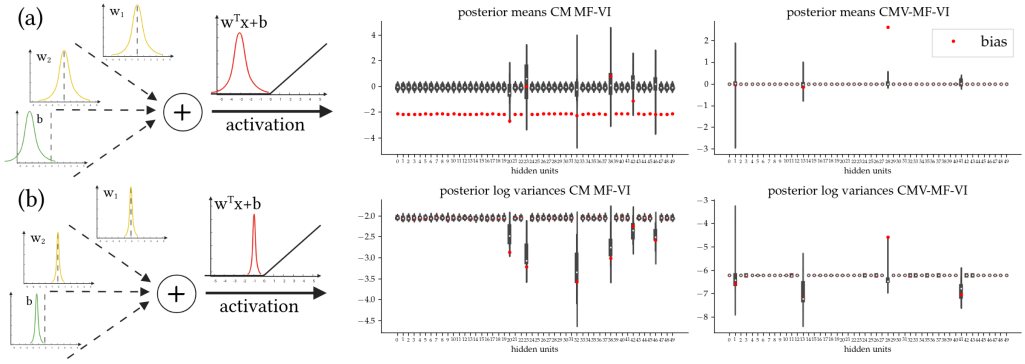

Figure 4: Learning prior hyperparameters in BNNs fixes over-pruning. CM-MF-VI: units where incoming posterior weights converge to prior have posterior biases set to low value, which saturates the activation. CMV-MF-VI: pruned units converge to spiked Gaussians.

bounds learning prior means CM-MF-VI, CMV-MF-VI outperform or match the bound learning only prior variances CV-MF-VI on all data sets, (iii) CM-MF-VI outperforms or matches MC-dropout on all data sets but boston and wine (the performance of dropout is unstable unless p is tuned per data set, e.g. it is poor on kin8nm and not shown). In addition, CM-MF-VI OPT matches the performance of CM-MF-VI, showing ELBO can be used to learn hyperparameter α_{reg}. Next we report a comparison between CM-MF-VI and standard MF-VI as the depth of the network increases in Figure 3 (bottom). We see standard MF-VI starts to rapidly under-perform on most data sets (for larger data sets standard MF-VI performs well as it converges to MAP). Learning prior means/variances by optimizing bounds in Eq. (11) or Eq. (12) allows to mitigate rapid degradation of predictions.

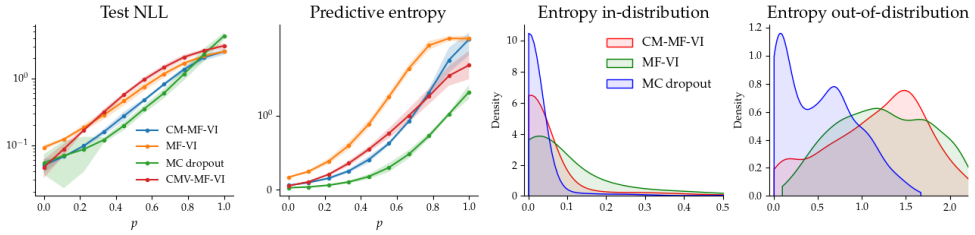

Figure 5: Comparison of standard MF-VI and collapsed bounds (CM-MF-VI, CV-MF-VI) for in-distribution and out-of-distribution data. Left: replacing $p\%$ of MNIST image with random bits, right learning BNN on MNIST and testing on fashionMNIST.

Fixing over-pruning in BNNs. We now explain why learning the hyperparameters of Gaussian prior in the BNN mitigates problems arising from over-pruning. In Figure 4 we plot the distribution of learned mean-field Gaussian posterior means and log variances (middle and right) for units feeding into the last hidden layer in a 1 hidden layer heteroscedastic BNN with 50 hidden units and ReLU activations learned on 200 data points from *boston* regression data set. We see that CM-MF-VI sets means of posterior biases to lower values ($\approx -2.$) to saturate the activations for units where incoming weights converge to the prior. Saturated activations stop the pruned weights from contributing variance into the predictions and result in better modeling of data, as we schematically show in Figure 4 (a). For CMV-MF-VI posterior weights corresponding to inactive units converge to delta spikes with very slightly negative means, in line with observations made in Figure 2, hence they do not influence predictions, shown in Figure 4 (b). Overall, learning the means and variances of Gaussian priors mitigates the influence of network units that have been pruned and which otherwise contribute only to the predictive variance and cause under-fitting.

MNIST classification with MLP. Classifying vectorized MNIST images using two hidden layer network with ReLU activation is a standard benchmark for BNNs [4]. We demonstrate that collapsed bounds give large improvements in test NLL compared to using standard MF-VI and provides the best predictions across tested algorithms. We report test NLLs and test error rates (ER) averaged over 5 random seeds in Table 1. For 400 hidden units, CM-MF-VI/CMV-MF-VI achieve test NLL of 0.047, compared to 0.094 for standard MF-VI. CM-MF-VI OPT again matches the performance of CM-MF-VI, showing α_{reg} can be learned with Eq. (11).

test NLL ↓ / ER ↓	MNIST	K-MNIST	F-MNIST	SVHN	CIFAR10
CMV-MF-VI	$\mathbf{0.021 \pm 0.001}$	0.152 ± 0.006	$\mathbf{0.253 \pm 0.006}$	$\mathbf{0.313 \pm 0.006}$	0.807 ± 0.005
CM-MF-VI	$\mathbf{0.021 \pm 0.001}$	$\mathbf{0.141 \pm 0.006}$	$\mathbf{0.254 \pm 0.005}$	$\mathbf{0.315 \pm 0.004}$	0.809 ± 0.009
CV-MF-VI	0.038 ± 0.003	0.239 ± 0.005	0.317 ± 0.006	0.321 ± 0.006	0.821 ± 0.005
CM-MF-VI OPT	0.024 ± 0.001	0.158 ± 0.004	0.258 ± 0.003	$\mathbf{0.314 \pm 0.004}$	$\mathbf{0.789 \pm 0.005}$
MF-VI	0.061 ± 0.001	0.319 ± 0.006	0.371 ± 0.003	0.340 ± 0.001	0.848 ± 0.009
MAP	0.048 ± 0.007	0.402 ± 0.027	0.336 ± 0.001	0.775 ± 0.012	1.134 ± 0.065
MC dropout	0.027 ± 0.001	0.222 ± 0.011	0.326 ± 0.007	0.400 ± 0.009	1.018 ± 0.017
MF-VI EB	0.060 ± 0.001	0.319 ± 0.003	0.372 ± 0.003	0.340 ± 0.005	0.843 ± 0.010
CMV-MF-VI	$0.73 \pm 0.02\%$	$\mathbf{4.00 \pm 0.06}\%$	$\mathbf{8.90 \pm 0.22}\%$	$8.49 \pm 0.20\%$	$27.76 \pm 0.37\%$
CM-MF-VI	$\mathbf{0.67 \pm 0.04}\%$	$\mathbf{3.75 \pm 0.30}\%$	$\mathbf{9.05 \pm 0.31}\%$	$\mathbf{8.21 \pm 0.20}\%$	$27.63 \pm 0.48\%$
CV-MF-VI	$1.24 \pm 0.12\%$	$6.85 \pm 0.10\%$	$11.47 \pm 0.13\%$	$8.96 \pm 0.20\%$	$28.39 \pm 0.14\%$
CM-MF-VI OPT	$0.72 \pm 0.05\%$	$4.23 \pm 0.06\%$	$9.33 \pm 0.07\%$	$8.47 \pm 0.03\%$	$\mathbf{26.80 \pm 0.41}\%$
MF-VI	$1.51 \pm 0.05\%$	$9.17 \pm 0.25\%$	$12.96 \pm 0.28\%$	$9.20 \pm 0.15\%$	$28.71 \pm 0.16\%$
MAP	$1.16 \pm 0.11\%$	$7.92 \pm 0.36\%$	$11.94 \pm 0.22\%$	$12.78 \pm 0.12\%$	$34.63 \pm 1.48\%$
MC dropout	$0.81 \pm 0.05\%$	$6.07 \pm 0.49\%$	$12.01 \pm 0.17\%$	$9.78 \pm 0.22\%$	$33.86 \pm 0.42\%$
MF-VI EB	$1.53 \pm 0.07\%$	$9.18 \pm 0.19\%$	$12.96 \pm 0.08\%$	$9.31 \pm 0.28\%$	$28.56 \pm 0.45\%$

Table 2: Image classification with LeNet CNN. Collapsed bounds visibly outperform standard MF-VI.

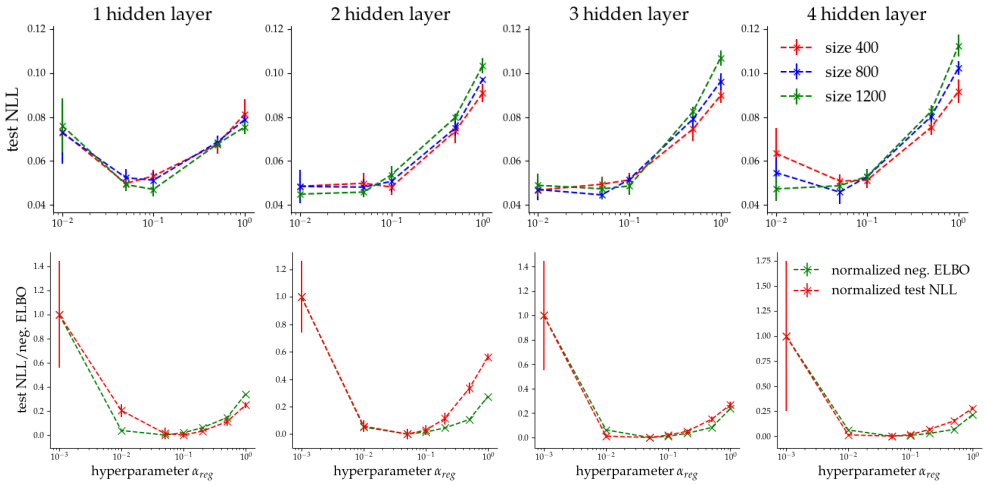

Figure 6: Top: performance of CM-MF-VI for different settings of hyperparameter α and different network architectures ($\alpha = 1$ is standard MF-VI). Setting $\alpha = 0.075$ to perform well across many network architectures/data sets. Bottom: ELBO $\mathcal{L}_m$ correlates with test NLL.

Perturbed MNIST images. We now investigate if in domain improvements of CM-MF-VI/CMV-MF-VI come at the cost of OOD performance. We find this is not the case: OOD performance is as good as before i.e. adjusting the prior hyperparameters on observed data does not degrade out-of-distribution (OOD) uncertainty. To show that, we learn a 2 hidden layer BNN with ReLU activations on vectorized MNIST images as previously, but test it on fashionMNIST data set [63] and randomly perturbed MNIST images where every pixel is flipped with probability p. We report test NLL, predictive entropy for perturbed MNIST experiment in Figure 5 (left). We see that CM-MF-VI gives lower test NLL than MF-VI for any p, but it also maintains growing entropy when $p \to 1$ i.e. when we transition to OOD data. In Figure 5 (right) we plot histograms of predictive entropy for MNIST in-distribution data and OOD data fashionMNIST. CM-MF-VI significantly improves upon MF-VI's under-confidence without excessive reduction in uncertainty in the OOD setting.

Limitations of the algorithm: adjusting hyperparameters. The bounds in Eq. (11), Eq. (12) have hyperparameters corresponding to the choice of hyper-prior. These additional free-parameters are the main limitation of the derived algorithms — nevertheless, CM-MF-VI has one hyperparameter α_{reg} which allows us to control the strength of regularization of the model. We analyze the test predictive performance for different settings of α_{reg} for CM-MF-VI in Figure 6 (top) averaged over 5 random seeds. We found the setting $\alpha_{reg} = 0.075$ works well across many different problems and is robust to the choice of size/depth of the network. Increasing α_{reg} to 0.2 provides strong regularization and works well for data sets very prone to over-fitting. For large models (e.g. ResNets)

model	test NLL↓ / ER ↓	CMV-MF-VI	CM-MF-VI	CV-MF-VI	MF-VI	MC dropout	MAP
RESNET18	STL10	$\mathbf{1.04 \pm 0.00}$	1.10 ± 0.02	1.57 ± 0.01	1.61 ± 0.01	1.17 ± 0.05	1.64 ± 0.03
	SVHN	0.15 ± 0.00	$\mathbf{0.14 \pm 0.00}$	0.20 ± 0.01	0.22 ± 0.00	0.18 ± 0.00	0.35 ± 0.01
	CIFAR100	$\mathbf{1.43 \pm 0.01}$	1.53 ± 0.00	2.00 ± 0.01	2.23 ± 0.03	1.75 ± 0.00	4.25 ± 0.05
	CIFAR10	0.41 ± 0.00	$\mathbf{0.39 \pm 0.00}$	0.59 ± 0.00	0.68 ± 0.02	0.49 ± 0.00	0.93 ± 0.02
SHUFFLENET	STL10	$\mathbf{0.96 \pm 0.01}$	0.99 ± 0.02	1.22 ± 0.05	1.70 ± 0.08	1.28 ± 0.01	1.78 ± 0.08
	SVHN	$\mathbf{0.27 \pm 0.01}$	$\mathbf{0.26 \pm 0.01}$	0.31 ± 0.01	0.31 ± 0.01	1.11 ± 0.01	0.32 ± 0.00
	CIFAR100	2.03 ± 0.01	$\mathbf{1.99 \pm 0.02}$	2.21 ± 0.06	2.28 ± 0.02	3.08 ± 0.01	4.48 ± 0.03
	CIFAR10	$\mathbf{0.65 \pm 0.00}$	$\mathbf{0.65 \pm 0.00}$	0.71 ± 0.01	0.72 ± 0.01	1.24 ± 0.00	1.10 ± 0.02
ALEXNET	STL10	1.50 ± 0.07	1.48 ± 0.04	1.80 ± 0.07	1.86 ± 0.03	$\mathbf{1.07 \pm 0.06}$	1.80 ± 0.32
	SVHN	0.30 ± 0.00	$\mathbf{0.28 \pm 0.00}$	0.42 ± 0.01	0.51 ± 0.01	0.38 ± 0.01	0.72 ± 0.06
	CIFAR100	2.24 ± 0.02	$\mathbf{2.16 \pm 0.06}$	2.62 ± 0.02	2.95 ± 0.04	$\mathbf{2.19 \pm 0.01}$	7.03 ± 0.14
	CIFAR10	0.72 ± 0.01	$\mathbf{0.69 \pm 0.01}$	0.99 ± 0.01	1.19 ± 0.01	0.74 ± 0.01	1.79 ± 0.06
RESNET18	STL10	$37.69 \pm 0.25\%$	$39.75 \pm 0.53\%$	$64.88 \pm 0.54\%$	$66.58 \pm 0.63\%$	$29.98 \pm 1.17\%$	$\mathbf{29.30 \pm 0.56}\%$
	SVHN	$\mathbf{3.76 \pm 0.02}\%$	$\mathbf{3.75 \pm 0.02}\%$	$5.26 \pm 0.21\%$	$5.73 \pm 0.09\%$	$4.11 \pm 0.15\%$	$4.94 \pm 0.07\%$
	CIFAR100	$\mathbf{39.41 \pm 0.39}\%$	$40.39 \pm 0.37\%$	$53.78 \pm 0.54\%$	$59.46 \pm 0.72\%$	$45.51 \pm 0.36\%$	$47.92 \pm 0.34\%$
	CIFAR10	$13.75 \pm 0.06\%$	$\mathbf{13.34 \pm 0.24}\%$	$20.22 \pm 0.30\%$	$22.92 \pm 1.14\%$	$16.36 \pm 0.28\%$	$15.31 \pm 0.35\%$
SHUFFLENET	STL10	$\mathbf{34.48 \pm 0.18}\%$	$34.79 \pm 0.78\%$	$46.43 \pm 2.83\%$	$73.74 \pm 5.67\%$	$47.72 \pm 1.12\%$	$41.41 \pm 1.13\%$
	SVHN	$7.85 \pm 0.32\%$	$\mathbf{7.31 \pm 0.27}\%$	$8.90 \pm 0.15\%$	$8.87 \pm 0.21\%$	$26.19 \pm 0.69\%$	$8.28 \pm 0.19\%$
	CIFAR100	$54.04 \pm 0.07\%$	$\mathbf{52.66 \pm 0.36}\%$	$58.95 \pm 1.08\%$	$60.84 \pm 0.54\%$	$73.96 \pm 0.19\%$	$63.52 \pm 0.22\%$
	CIFAR10	$\mathbf{22.71 \pm 0.61}\%$	$\mathbf{22.68 \pm 0.14}\%$	$24.66 \pm 0.53\%$	$25.01 \pm 0.47\%$	$42.54 \pm 0.08\%$	$28.19 \pm 0.59\%$
ALEXNET	STL10	$56.59 \pm 4.82\%$	$55.97 \pm 2.99\%$	$74.38 \pm 5.70\%$	$78.28 \pm 1.32\%$	$37.33 \pm 0.99\%$	$\mathbf{35.74 \pm 0.67}\%$
	SVHN	$7.92 \pm 0.20\%$	$\mathbf{7.41 \pm 0.13}\%$	$11.61 \pm 0.13\%$	$13.17 \pm 0.20\%$	$8.87 \pm 0.19\%$	$10.35 \pm 0.71\%$
	CIFAR100	$58.14 \pm 0.83\%$	$\mathbf{54.91 \pm 1.07}\%$	$66.82 \pm 0.64\%$	$73.17 \pm 1.39\%$	$\mathbf{55.39 \pm 0.23}\%$	$60.14 \pm 0.19\%$
	CIFAR10	$24.35 \pm 0.30\%$	$\mathbf{23.40 \pm 0.20}\%$	$34.73 \pm 0.34\%$	$41.63 \pm 0.31\%$	$24.74 \pm 0.43\%$	$26.04 \pm 0.66\%$

Table 3: Test NLL and error rates (ER) for introduced collapsed bounds on large scale CNN experiments. Collapsed bounds provide visibly better predictions than MF-VI, which under-fits.

setting $\alpha_{reg} \leq 0.025$ can give even better predictive performance. We report all hyperparameters used in experiments in Appendix D. In Figure 6 (bottom) we show normalized values of ELBO and test NLL for different values α_{reg}. This has an important practical implication: approximately tuning α_{reg} can be guided by ELBO. Figure 6 (bottom) should be compared with Fig 6 in [30] where the authors show the same property using approximation of $\log p(\mathcal{D})$, as opposed to us using ELBO.

Image classification LeNet. We now consider image classification with the LeNet architecture [37] on 6 data sets: MNIST, fashionMNIST, K-MNIST [6], CIFAR10, CIFAR100 [34] and SVHN [49]. We optimize the objectives for 800 epochs (except MAP 50 epochs and MC dropout 100 epochs) using batch size 512 and ADAM optimizer with default parameters. We report test NLL and test error rates (ER) averaged over 3 random seeds and standard deviation error bars in Table 2. We again observe the introduced bounds CM-MF-VI, CV-MF-VI and CMV-MF-VI outperform standard MF-VI in both test NLL and test error rate. CM-MF-VI performs slightly better than CMV-MF-VI, but the differences are not statistically significant. Both CM-MF-VI and CMV-MF-VI outperform MAP, CV-MF-VI and MC dropout in test NLL and test ER.

Image classification large CNNs. We follow by experimenting with larger CNNs: ResNet18 [21], ShuffleNet[41] and AlexNet[35]. We use 4 data sets: CIFAR10, CIFAR100, STL10 [7] and SVHN. We again compare CMV-MF-VI, CV-MF-VI and CM-MF-VI to MF-VI, MC dropout and MAP. We optimize the objectives for 800 epochs (MAP and MC dropout early stopped at 200) with the default ADAM optimizer and the same data augmentation as in [50] and average results over 3 random seeds. We gather the results in Table 3. Experiments with large CNNs confirm our findings from previous experiments: (i) CM-MF-VI/CMV-MF-VI always outperform standard MF-VI by a visible margin and result in good predictive performance (e.g. outperforming SOTA VOGN [50]), (ii) learning prior means CM-MF-VI/CMV-MF-VI outperforms learning prior variances (CV-MF-VI).

6 Conclusions

We developed a family of algorithms optimizing variational posteriors in BNNs based on collapsed variational bounds. These algorithms allowed us to demonstrate that ELBO is a suitable optimization target for learning hyperparameters of BNNs. We demonstrated that learning hyperparameters of BNNs fixes their predictive under-confidence resulting in good empirical performance and robustness to over-fitting. We also showed that learning model parameters in BNNs allows us to mitigate the negative consequences of variational posteriors converging to the prior in large networks (over-pruning). Importantly, the introduced algorithms do not incur additional computational cost compared to applying MF-VI to BNNs and can be readily applied to improve the predictive performance of existing implementations. We hope that our approach will enable the practical use of VI in large network architectures.

References

[1] Laurence Aitchison. A statistical theory of cold posteriors in deep neural networks, 2021.

[2] Andrei Atanov, Arsenii Ashukha, Kirill Struminsky, Dmitry Vetrov, and Max Welling. The deep weight prior, 2019.

[3] D. Barber and Christopher Bishop. Ensemble learning in bayesian neural networks. In *Generalization in Neural Networks and Machine Learning*, pages 215–237. Springer Verlag, January 1998.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1613–1622. JMLR.org, 2015.

[5] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4754–4765. Curran Associates, Inc., 2018.

[6] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

[7] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[8] Beau Coker, Melanie Fernandez Pradier, and Finale Doshi-Velez. Porb-nets: Poisson process radial basis function networks. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 1338–1347. PMLR, 03–06 Aug 2020.

[9] Alexander G. de G. Matthews, Mark Rowland, Jiri Hron, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks, 2018.

[10] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[11] Michael W. Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. In *International Conference on Machine Learning (ICML)*, 2020.

[12] Sebastian Farquhar, Michael A. Osborne, and Yarin Gal. Radial bayesian neural networks: Beyond discrete support in large-scale bayesian deep learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1352–1362. PMLR, 26–28 Aug 2020.

[13] Sebastian Farquhar, Lewis Smith, and Yarin Gal. Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4346–4357. Curran Associates, Inc., 2020.

[14] Andrew Foong, David Burt, Yingzhen Li, and Richard Turner. On the expressiveness of approximate inference in bayesian neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15897–15908. Curran Associates, Inc., 2020.

[15] Andrew Y. K. Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E. Turner. 'in-between' uncertainty in bayesian neural networks, 2019.

[16] Vincent Fortuin, Adrià Garriga-Alonso, Florian Wenzel, Gunnar Rätsch, Richard Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited, 2021.

[17] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

[18] Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured variational learning of bayesian neural networks with horseshoe priors, 2018.

[19] Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Model selection in bayesian neural networks via horseshoe priors. *Journal of Machine Learning Research*, 20(182):1–46, 2019.

[20] Alex Graves. Practical variational inference for neural networks. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2348–2356. Curran Associates, Inc., 2011.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[22] James Hensman, Magnus Rattray, and Neil D. Lawrence. Fast variational inference in the conjugate exponential family, 2012.

[23] Jose Miguel Hernandez-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. 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 1861–1869, Lille, France, 07–09 Jul 2015. PMLR.

[24] Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, page 5–13, New York, NY, USA, 1993. Association for Computing Machinery.

[25] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, May 2013.

[26] Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Mohammad Emtiyaz Khan. Scalable marginal likelihood estimation for model selection in deep learning, 2021.

[27] Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like?, 2021.

[28] Theofanis Karaletsos and Thang D Bui. Hierarchical gaussian process priors for bayesian neural network weights. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17141–17152. Curran Associates, Inc., 2020.

[29] Theofanis Karaletsos, Peter Dayan, and Zoubin Ghahramani. Probabilistic meta-representations of neural networks, 2018.

[30] Mohammad Emtiyaz E Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. Approximate inference turns deep networks into gaussian processes. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[31] Nathaniel J. King and Neil D. Lawrence. Fast variational inference for gaussian process models through kl-correction. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 270–281, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[32] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[33] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2575–2583, Cambridge, MA, USA, 2015. MIT Press.

[34] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[36] Miguel Lázaro-Gredilla and Michalis K. Titsias. Variational heteroscedastic gaussian process regression. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 841–848, Madison, WI, USA, 2011. Omnipress.

[37] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, page 319, Berlin, Heidelberg, 1999. Springer-Verlag.

[38] Noel Loo, Siddharth Swaroop, and Richard E. Turner. Generalized variational continual learning, 2020.

[39] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[40] Miguel Lázaro-Gredilla, Steven Van Vaerenbergh, and Neil Lawrence. Overlapping mixtures of gaussian processes for the data association problem, 2011.

[41] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design, 2018.

[42] David J. C. MacKay. The evidence framework applied to classification networks. 4(5):720–736, September 1992.

[43] David J. C. MacKay. Probable networks and plausible predictions — a review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.

[44] David J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Comput.*, 11(5):1035–1068, July 1999.

[45] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. 2016.

[46] Aaron Mishkin, Frederik Kunstner, Didrik Nielsen, Mark Schmidt, and Mohammad Emtiyaz Khan. Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6245–6255. Curran Associates, Inc., 2018.

[47] Radford M. Neal. Bayesian learning via stochastic dynamics. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 475–482. Morgan-Kaufmann, 1993.

[48] Radford M. Neal. *Priors for Infinite Networks*, pages 29–53. Springer New York, New York, NY, 1996.

[49] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. *NIPS*, 01 2011.

[50] Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with bayesian principles. In *Advances in Neural Information Processing Systems 32*, pages 4289–4301. Curran Associates, Inc., 2019.

[51] Hiske Overweg, Anna-Lena Popkes, Ari Ercole, Yingzhen Li, José Miguel Hernández-Lobato, Yordan Zaykov, and Cheng Zhang. Interpretable outcome prediction with sparse bayesian neural networks in intensive care, 2019.

[52] Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 294–300. MIT Press, 2001.

[53] Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling, 2018.

[54] Jakub Swiatkowski, Kevin Roth, Bastiaan S. Veeling, Linh Tran, Joshua V. Dillon, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. The k-tied normal distribution: A compact parameterization of gaussian mean field posteriors in bayesian neural networks, 2020.

[55] Yee Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

[56] Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1971–1979, Bejing, China, 22–24 Jun 2014. PMLR.

[57] Marcin Tomczak, Siddharth Swaroop, and Richard Turner. Efficient low rank gaussian variational inference for neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4610–4622. Curran Associates, Inc., 2020.

[58] Brian Trippe and Richard Turner. Overpruning in variational bayesian neural networks, 2018.

[59] Richard Turner, Pietro Berkes, and Maneesh Sahani. Two problems with variational expectation maximisation for time-series models. *Bayesian Time Series Models*, 01 2008.

[60] Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really?, 2020.

[61] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization, 2020.

[62] Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E. Turner, Jose Miguel Hernandez-Lobato, and Alexander L. Gaunt. Deterministic variational inference for robust bayesian neural networks. In *International Conference on Learning Representations*, 2019.

[63] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[64] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. *International Conference on Learning Representations*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We outline the contributions in introductory sections and in the abstract. We derive tighter ELBOs for BNNs by applying collapsed variational bounds.
   (b) Did you describe the limitations of your work? [Yes] In Section 5 we discuss the limitation of our algorithms - tuning hyperparameters.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work attempts to improve the performance of BNNs. There are several works attempting to do the same [4, 20, 33, 50, 11] where no negative societal impacts have been identified.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] We do not formulate and theorems/lemmas, but we defer derivations of the equations in the main text to Appendix C.
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code will be provide as a part of supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide most important settings in Section 5 and defer full specification of hyperparameters to Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report results with 3 or 5 random seeds as mentioned in the main text and corresponding mean/standard deviation.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We report this in the Appendix D. Our algorithms do not increase the cost/time compared to mean-field Gaussian VI with fixed priors.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite all data sets we have used in Section 5. These are standard academic benchmarks used in most of the works focusing on VI applied to BNNs.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We do not rely on any crowdsourcing nor research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]