Collapsed Variational Bounds for Bayesian Neural Networks

Marcin B. Tomczak, Siddharth Swaroop, Andrew Y. K. Foong, Richard E. Turner
University of Cambridge
Cambridge, UK
{mbt27,ss2163,ykf21,ret26}@cam.ac.uk

Abstract

Recent interest in learning large variational Bayesian Neural Networks (BNNs) has been partly hampered by poor predictive performance caused by underfitting, and their performance is known to be very sensitive to the prior over weights. Current practice often fixes the prior parameters to standard values or tunes them using heuristics or cross-validation. In this paper, we treat prior parameters in a distributional way by extending the model and collapsing the variational bound with respect to their posteriors. This leads to novel and tighter Evidence Lower Bounds (ELBOs) for performing variational inference (VI) in BNNs. Our experiments show that the new bounds significantly improve the performance of Gaussian mean-field VI applied to BNNs on a variety of data sets, demonstrating that mean-field VI works well even in deep models. We also find that the tighter ELBOs can be good optimization targets for learning the hyperparameters of hierarchical priors.

1 Introduction

There has been a lot of recent interest in developing methods for Bayesian Neural Networks (BNNs), and variational methods provide computationally cheap approximations to the posterior over weights when compared to alternatives like MCMC. Research on Variational Inference (VI) for BNNs has seen multiple advances enabling optimization of the Evidence Lower Bound (ELBO) \cite{29,65,24,7,38}, allowing for scaling to large neural networks and datasets \cite{57,63,15,16,66}. 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 \cite{9,60,69}. Although a lot of work has improved performance by deriving better estimators of the gradient of the ELBO \cite{29,65,24,7,38}, this work has focused on the variational parameters of distributions over weights, with the prior over weights usually fixed to zero-mean isotropic Gaussians \cite{55,70,15,20,7,40}.

In this paper, we also apply inference to the prior parameters of Gaussian BNNs (e.g. means and variances of a Gaussian prior over weights). We do this by using collapsed VI bounds \cite{36,43,47,64}, which analytically solve the inference over prior parameters and allow us to derive tighter ELBOs for performing VI in BNNs. Many previous works argue that point estimates for parameters of Gaussian prior in BNNs can be learned reliably by maximizing the ELBO \cite{72,52,50,31}, although there have also been arguments against such an approach \cite{7}. Experiments using our derived ELBOs lead to two findings: (i) MF-VI performs well in deep models where the prior parameters of Gaussian distributions 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 hierarchical Gaussian priors, even in large networks. We provide the code implementing the introduced algorithms at \url{https://github.com/marctom/collapsed_bnns}.

2 Collapsed Variational Bounds for Bayesian Neural Networks

We consider performing VI in a supervised learning scenario given \( N \) data points \( D = \{(x_i, y_i)\}_{i=1}^{N} \). We denote a neural network as \( f_W \), the observation model as \( p(y|f_W(x)) \), where \( W \) denotes the vectorized weights and biases of the network, and the likelihood of the weights \( W \) as \( p(D|W) = \prod_{i=1}^{N} p(y_i|f_W(x_i)) \). We focus on Gaussian mean-field priors (with non-zero means) over vectorized network weights \( p(W|\mu_p, \sigma_p^2) = \mathcal{N}(W|\mu_p, \sigma_p^2) \) meaning \( w \sim \mathcal{N}(\mu_p, \sigma_p^2) \) for any coordinate \( w \) in \( W \) (both \( \mu_p \) and \( \sigma_p^2 \) are also vectors in this notation). Similarly, we restrict our attention to mean-field Gaussian approximate posteriors \( q(W|\mu_q, \sigma_q^2) = \mathcal{N}(W|\mu_q, \sigma_q^2) \). VI finds an approximate posterior \( q(W|\mu_q, \sigma_q^2) \) by minimizing the KL divergence \( D_{KL}(q(W|\mu_q, \sigma_q^2)||p(W|D, \mu_p, \sigma_p^2)) \) w.r.t. the variational parameters \( \mu_q, \sigma_q^2 \). This is done by optimizing the Evidence Lower Bound (ELBO) \([28, 4]:\)

\[
\mathcal{L}(\mu_q, \sigma_q^2; \mu_p, \sigma_p^2) = \sum_{i=1}^{N} \mathbb{E}_{q(W|\mu_q, \sigma_q^2)} \log p(y_i|f_W(x_i)) - D_{KL}(q(W|\mu_q, \sigma_q^2)||p(W|\mu_p, \sigma_p^2)).
\]

We denote the first term on RHS as \( \mathcal{L}_{data}(\mu_q, \sigma_q^2) \). As \( D_{KL}(q(W|\mu_q, \sigma_q^2)||p(W|D, \mu_p, \sigma_p^2)) \geq 0 \), the ELBO lower bounds \( \log p(D|\mu_p, \sigma_p^2) \). Previous work \([4, 59]\) introduces low variance updates for mean-field VI by reparametrizing weights \( W \). Bayesian predictions are approximated by integrating the learned variational posterior \( q^*(W|\mu_q, \sigma_q^2) \) as \( p(y*|x^*, D) \approx \mathbb{E}_{q^*(W|\mu_q, \sigma_q^2)} p(y^*|f_W(x*)) \).

For BNNs, prior means are most often set to zero \( \mu_p = 0 \), and prior variances are set to a constant \( \sigma_p^2 = \gamma I \) (often scaled by the size of a previous layer \([55, 13]\)). We consider a more general setup where the weights of a BNN are defined by a hierarchical model \( p(W|\mu_p, \sigma_p^2)p(\mu_p)p(\sigma_p^2) \) and aim to perform inference over weights \( W \) and \( \mu_p, \sigma_p^2 \). For clarity, we will refer to \( \mu_p, \sigma_p^2 \) as prior parameters, leaving the term hyperparameters for any other free variables.

2.1 Deriving collapsed variational bounds

In this section, we develop an efficient optimization scheme for variational BNNs with weights defined in a hierarchical fashion, where the inference over prior parameters is done analytically and prior parameters are subsequently marginalized out. This defines new learning objectives for learning variational BNNs, which we show by providing specific examples later in the text. More specifically, we apply collapsed variational bounds \([36, 43, 47, 26]\) to BNNs, and present a general, systematic way of deriving novel, tighter lower bounds on the model evidence which are useful for learning variational posteriors over weights. The idea behind a collapsed bound is simple: Suppose we maximize \( f(x, y) \) and we can derive the optimal \( x^*(y) = \arg\max_x f(x, y) \). Substituting \( x^* \) into \( f \) results in \( f(x^*(y), y) \), which only depends on \( y \), is easier to optimize, and has the property that \( \forall_y f(x^*(y), y) \geq f(x, y) \). Collapsed variational bounds apply this reasoning to the ELBO with factorized variational posteriors taking the roles of \( x \) and \( y \).

Inferring prior parameters in BNN. We consider inference over both network weights \( W \) and prior parameters \( \mu_p, \sigma_p^2 \), which we treat as latent variables and subsequently discuss collapsing the inference w.r.t. prior parameters \( \mu_p \) and \( \sigma_p^2 \). We use the factorization \( q(W, \mu_p, \sigma_p^2; \mu_q, \sigma_q^2) = q(W|\mu_q, \sigma_q^2)q(\mu_p, \sigma_p^2) \), where we omit the variational parameters of \( q(\mu_p, \sigma_p^2) \) for concise notation. Learning \( q(\mu_p, \sigma_p^2) \) under this assumption does not complicate computing the BNN’s predictive distribution \( p(y|x, D) \), i.e. \( \mu_q, \sigma_q^2 \) do not need to be sampled at the prediction time, since, \( p(y|x, D) = \mathbb{E}_{q(W, \mu_p, \sigma_p^2|D)} p(y|f_W(x)) \approx \mathbb{E}_{q(W|\mu_q, \sigma_q^2)} p(y|f_W(x)) = \mathbb{E}_{q(W|\mu_q, \sigma_q^2)} p(y|f_W(x)) \).

To apply inference, we invoke the variational principle which lower bounds the log marginal likelihood \( \log p(D) \) as:

\[
\log p(D) \geq \mathbb{E}_{q(W|\mu_q, \sigma_q^2)q(\mu_p, \sigma_p^2)} \log \frac{p(D, W, \mu_p, \sigma_p^2)}{q(W|\mu_q, \sigma_q^2)q(\mu_p, \sigma_p^2)} = \mathcal{L}^q(\mu_q, \sigma_q^2, q(\mu_p, \sigma_p^2)).
\]

The bound in Eq. \([8]\) reduces to \( \mathcal{L}(\mu_q, \sigma_q^2) \) in Eq. \([1]\) when \( q(\mu_p, \sigma_p^2) = q(\mu_p)q(\sigma_p^2) \) and \( q(\mu_p) \), \( q(\sigma_p^2) \) are delta functions. If we are able to analytically derive the distribution \( q^*(\mu_p, \sigma_p^2) \) maximizing
\[ \mathcal{L}^*, \text{ we can substitute the resulting distribution into Eq. (3) to derive a tighter (collapsed) bound on the marginal likelihood, saving the need to perform coordinate ascent or gradient updates to learn the prior parameters:} \]

\[
\log p(D) \geq \mathbb{E}_{q(W | \mu_q, \sigma_q^2)}[\log p(D, W, \mu_p, \sigma_p^2)] - \mathbb{E}_{q(W | \mu_q, \sigma_q^2)}[\log q(W)] - \mathbb{E}_{q(W | \mu_q, \sigma_q^2)}[KL(q(W) || p(W))].
\]

where \( L^*(\mu_q, \sigma_q^2) \geq L^0(\mu_q, \sigma_q^2, q(\mu_p, \sigma_p^2)) \) and \( L^* \) depends only on the variational parameters \( \mu_q, \sigma_q^2 \). Collapsed variational bounds are desirable as they have been shown to make learning significantly more efficient \cite{blei2017}. The tighter bound in Eq. (4) has been also referred to as a KL corrected bound \cite{kingma2013} and marginal VI bound \cite{blei2017}.

**Collapsing the variational bound for BNNs.** We now provide a method to derive a collapsed variational bound by maximizing \( L^0(\mu_q, \sigma_q^2, q(\mu_p, \sigma_p^2)) \) w.r.t. the posterior over weights prior parameters \( q(\mu_p, \sigma_p^2) \) in BNNs. We use the property that the BNN likelihood term \( \log p(y | x, f_W(x)) \) does not depend on the prior parameters \( \mu_p, \sigma_p^2, \) i.e. \( p(D | W, \mu_p, \sigma_p^2) = p(D | W) \). The bound \( L^0 \) from Eq. (4) decomposes into three terms:

\[
L^0(\mu_q, \sigma_q^2, q(\mu_p, \sigma_p^2)) = L_{\text{data}}(\mu_q, \sigma_q^2) + \Phi(\mu_q, \sigma_q^2, q(\mu_p, \sigma_p^2)) + H[q(W | \mu_q, \sigma_q^2)],
\]

where \( H[q(W | \mu_q, \sigma_q^2)] \) denotes the differential entropy, and we have defined

\[
\Phi(\mu_q, \sigma_q^2, q(\mu_p, \sigma_p^2)) = \mathbb{E}_{q(\mu_p, \sigma_p^2)} \left[ \mathbb{E}_{q(W | \mu_q, \sigma_q^2)} \log p(W | \mu_p, \sigma_p^2) \right] - D_{KL}(q(\mu_p, \sigma_p^2) || p(\mu_p)p(\sigma_p^2)).
\]

Only \( \Phi \) depends on the posterior \( q(\mu_p, \sigma_p^2) \), and therefore finding \( q^*(\mu_p, \sigma_p^2) \) can be done by maximizing \( \Phi(\mu_q, \sigma_q^2, q(\mu_p, \sigma_p^2)) \).

We can significantly simplify this by noting that the objective in Eq. (6) is analogous to Eq. (1), hence optimizing \( \Phi \) takes the form of a nested VI problem: we want to infer \( q(\mu_p, \sigma_p^2) \) where the "data" distribution is replaced by the current variational posterior over weights \( q(W | \mu_q, \sigma_q^2) \). This implies the solutions \( q^*(\mu_p, \sigma_p^2) \) are straightforward to derive in closed-form priors/approximate posteriors from the exponential family \cite{blei2017}. Specifically, we can write the maximizer \( q^*(\mu_p, \sigma_p^2) \) as:

\[
\log q^*(\mu_p, \sigma_p^2) \propto \log p(\mu_p) + \log p(\sigma_p^2) + \mathbb{E}_{q(W | \mu_q, \sigma_q^2)} \log p(W | \mu_p, \sigma_p^2).
\]

Substituting the optimal variational posterior \( q^*(\mu_p, \sigma_p^2) \) into Eq. (6) gives us \( \Phi^* \), and we can substitute this into Eq. (5) to give our final collapsed bound \( L^* = L_{\text{data}} + \Phi^* + H[q(W | \mu_q, \sigma_q^2)] \). Furthermore, when calculating \( \Phi^* \), the inferred prior parameters \( q^*(\mu_p, \sigma_p^2) \) can be analytically integrated out in many cases, i.e. the outer expectation on the RHS in Eq. (6) can be solved in closed-form (we provide examples in Section 2.2), leading to a concise optimization objective.

The final optimization target to learn the variational posterior \( q(W | \mu_q, \sigma_q^2) \) is given by \( L^*(\mu_q, \sigma_q^2) \). Compared to gradient learning of \( \mu_p \) and \( \sigma_p^2 \), eliminating \( \mu_p \) and \( \sigma_p^2 \) from \( L^* \) performs better (as we show in Section 4), and halves both the memory requirement and time taken per update. It is important to remember that optimizing \( L^* \) learns both the variational parameters \( \mu_q, \sigma_q^2 \) and the posterior over prior parameters \( q^*(\mu_p, \sigma_p^2) \), but the latter is implicit as we express it as a function of \( \mu_q, \sigma_q^2 \) given by Eq. (7).

In summary, there are four steps to derive a collapsed bound \( L^* \), which we directly use to learn the variational posterior of a BNN. These steps are given in Algorithm 1. We next give two concrete examples of collapsed bounds.

**2.2 Examples of tighter ELBOs**

**Learn prior means, fix prior variances.** As an easier example, we first consider fixing the prior variance of weights, and learning prior means. We follow the four steps of Algorithm 1 to derive our collapsed bound. Detailed derivations are in Appendix B.

**Step 1:** This step defines a model and the family of variational approximations. We choose \( p(W | \mu_k) = \mathcal{N}(W | \mu_k, \gamma 1) \) with a Gaussian prior \( p(\mu_k | \alpha) = \mathcal{N}(\mu_k | 0, \alpha 1) \), where \( \alpha \) and \( \gamma \) are
Algorithm 1 Deriving a collapsed bound $\mathcal{L}^*$ for BNN in four steps.

**Step I:** Choose priors $p(\mu_p), p(\sigma^2_p), p(W|\mu_p, \sigma^2_p)$, and approximate posteriors $q(\mu_p, \sigma^2_p), q(W|\mu_q, \sigma^2_q)$ over network weights.

**Step II:** Calculate optimal prior parameters $q^*(\mu_p, \sigma^2_p)$ using Eq. 7.

**Step III:** Form $\Phi^*$ and solve $E_{q^*}(\mu_p, \sigma^2_p) \left[ E_q(W|\mu_q, \sigma^2_q) \log p(W|\mu_p, \sigma^2_p) \right]$ in Eq. 6.

**Step IV:** The collapsed bound is $\mathcal{L}^* = \mathcal{L}_{data} + \Phi^* + \mathcal{H}[q(W|\mu_q, \sigma^2_q)]$.

hyperparameters. We also employ a mean-field Gaussian posterior $q(W|\mu_q, \sigma^2_q)$ and a Gaussian posterior $q(\mu_p)$.

**Step II:** This step analytically finds the optimal variational distribution $q^*(\mu_p)$. Substituting our distributions into Eq. 7 for every coordinate $\mu_p$ gives us

$$\log q^*(\mu_p) \propto -\frac{(\mu_q - \mu_p)^2}{2\gamma} - \frac{\mu_p^2}{2\alpha}. \tag{8}$$

In this case the variational posterior $q^*(\mu_p)$ matches the true posterior, as the above dependency is equivalent Bayesian inference over $\mu_p$ given Gaussian likelihood/prior and observation $\mu_q$, so $q^*(\mu_p) = N(\mu_p | \frac{\mu_q}{\alpha+\gamma}, \frac{\alpha}{\alpha+\gamma})$.

**Step III:** This step forms $\Phi^*$ by calculating the divergences between optimal posteriors $q^*$ and priors and simplifies the derived optimization objective by marginalizing out approximate prior parameters. The divergence between two Gaussians $D_{KL}(q^*(\mu_p)||p(\mu_p))$ is straightforward to compute. Next we solve the integral $\mathcal{I} = E_{q^*}(\mu_p) \left[ E_q(W|\mu_q, \sigma^2_q) \log p(W|\mu_p, \sigma^2_p) \right]$:

$$\mathcal{I} \triangleq \mathbb{E}_{\mathcal{N}(\mu_p, \frac{\gamma}{\alpha+\gamma} \mu_q, \frac{\alpha}{\alpha+\gamma})} \left[ \frac{1}{2} \sigma^2_q + \frac{1}{2} (\mu_q - \mu_p)^T (\mu_q - \mu_p) \right] \triangleright \frac{1}{2} \frac{1}{\gamma} \sigma^2_q + \frac{1}{2} \frac{1}{\gamma} (\mu_q - \mu_p)^T (\mu_q - \mu_p) \tag{9}$$

Step IV: This step forms a collapsed bound that will be used to learn variational posteriors. Substituting the above into $\mathcal{L}^* = \mathcal{L}_{data} + \Phi^* + \mathcal{H}[q(W|\mu_q, \sigma^2_q)]$ results in our new bound:

$$\mathcal{L}_{\alpha reg}^*(\mu_q, \sigma^2_q) \triangleq \mathcal{L}_{data}(\mu_q, \sigma^2_q) - \frac{1}{2\gamma} \left[ \frac{1}{2} \sigma^2_q + \alpha_{\text{reg}} (\mu_q - \mu_p)^T (\mu_q - \mu_p) \right] + \frac{1}{2} \log \sigma^2_q + \frac{D}{2} \log \alpha_{\text{reg}},$$

where we have defined the hyperparameter $\alpha_{\text{reg}} = \gamma/(\gamma + \alpha) \in (0, 1)$, and for fixed $\gamma$ there is a 1-to-1 relation between $\alpha$ and $\alpha_{\text{reg}}$. When $\alpha \to 0$ (so $\alpha_{\text{reg}} \to 1$) we recover the standard expression for the ELBO given by Eq. 10 with $q(W|\mu_q, \sigma^2_q)$ and prior $\mathcal{N}(W|0, \gamma I)$. This corresponds to a dogmatic prior $p(\mu_p) = \mathcal{N}(\mu_p|0, \alpha_1)$ converging to a delta spike at 0, and no inference of $\mu_p$. When we allow for inference, the regularization of the (approximate) posterior mean $\mu_q$ decreases. When $\alpha_{\text{reg}} \to 0$ (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 $\frac{D}{2} \log \alpha_{\text{reg}}$, which can be interpreted as an Occam’s Razor penalty [59]; this term penalizes small $\alpha_{\text{reg}}$ as it diverges to negative infinity when $\alpha_{\text{reg}} \to 0$. Since $\alpha_{\text{reg}} \in (0, 1)$, this enables finding well-performing values of $\alpha_{\text{reg}}$ by evaluating $\mathcal{L}_{\alpha reg}^*$ (without having a validation set) [50], and we show this in our experiments in Section 4.

We find that $\alpha_{\text{reg}} \in (0.01, 0.1)$ significantly improves upon the predictions of MF-VI with fixed Gaussian priors. For existing implementations learning BNNs with Gaussian variational posterior/prior we suggest the default setting $\alpha_{\text{reg}} = 0.05$ in front of the mean regularization in the expression for KL divergence to instantaneously improve the predictions (as opposed to down-weighting the whole KL divergence term). Larger $\alpha_{\text{reg}}$ increases the strength of the regularization of the model.

**Learn both prior means and variances.** We now discuss a scheme to learn hyperparameters of both prior means and variances. We follow the same four steps from Algorithm 1. We again defer the detailed derivations to Appendix 13.

**Step I:** $p(W|\mu_p, \tau_p) = \mathcal{N}(W|\mu_p, \frac{1}{\tau_p})$ and $p(\mu_p|t, \tau) = \mathcal{N}(\mu_p|0, \frac{1}{t\tau^2})$, $p(\tau) = \mathcal{G}(\tau|\alpha, \beta)$. We again consider posteriors $q(W|\mu_q, \sigma^2_q) = \mathcal{N}(W|\mu_q, \sigma^2_q)$, $q(\mu_p|\tau_p) = \mathcal{N}(\mu_p)$ and $q(\tau_p) = \mathcal{G}(\tau_p)$. No further analysis is required.
Step II: The optimal posteriors $q^*$ are given by $q^*(\mu_p) = \mathcal{N}(\mu_p | \frac{1}{1+t} \mu_q, \frac{1}{(1+t)^2} \tau_p)$ and $q^*(\tau_p) = G(\tau_p | (\alpha + \frac{1}{2} t) 1, \beta + \frac{t}{2(1+t)} \mu_q^2 + \frac{1}{2} \sigma_q^2)$.

Steps III and IV: The bound $\mathcal{L}^*(\mu_q, \sigma_q^2)$ becomes,

$$\mathcal{L}_{mv}^*(\mu_q, \sigma_q^2) = \mathcal{L}_{data}(\mu_q, \sigma_q^2) - \left(\alpha + \frac{1}{2} T^T \log \left[ \beta I + \frac{t}{2} \mu_q^2 + \frac{1}{2} \sigma_q^2 \right] + \frac{1}{2} T^T \log \sigma_q^2 \right),$$

where $\delta = t/(1+t)$. Eq. (10) recovers the case of setting prior mean $\mu_p = 0$ and learning only prior variances $\sigma_p^2$ when the prior precision $t \tau_p$ over $\mu_p$ goes to $\infty$, i.e. $t \rightarrow \infty$. Setting $\delta < 1$ in Eq. (10) weakens the regularization of posterior mean $\mu_p$, allowing it to vary more. The regularizer in Eq. (7) is a decreasing function of $\mu_q$, i.e. without gradients from $\mathcal{L}_{data}$, the posterior mean $\mu_q$ of a weight $w$ converges to 0. In Section 4 we show that $\mathcal{L}_{mv}^*$ outperforms standard MF-VI.

Comparison of regularizing terms. We compare the regularization terms on $\mu_q$ and $\sigma_q^2$ in the standard mean-field ELBO in Eq. (1) and introduced ELBOs in Eq. (9) and Eq. (10) in Figure 1 where we use prior the $p(w) = \mathcal{N}(w | 0, 1)$. We consider what happens in the absence of gradients from the data term $\mathcal{L}_{data}$ to better understand the bounds. In Figure 1(left) we see that Gaussian posteriors (in the absence of data) optimized with Eq. (1) converge to their prior $\mathcal{N}(w | 0, 1)$ (68). In Figure 1(middle) we are down-weighting $\mu_q$ in Eq. (9), hence weakening the regularization of $\mu_q$. This allows posterior means to vary more, and can be interpreted as scaling the $\mu_q$ axis in Figure 1 by $\sqrt{\alpha}$. As we show in Section 4 this enables posterior means $\mu_q$ to saturate activations and reduce the noise from pruned weights. Optimizing Eq. (10) is roughly opposite to Eq. (1) and causes weights to be pruned to $\mathcal{N}(0, 2\beta)$ (Figure 1(right), where $\beta = 0.01$). Weights pruned in this way do influence the predictions for small $\beta$ and this fixes the excessive injection of noise by pruned weights, as we show in Section 4.

3 Related work

Performance of BNNs is known to be sensitive to prior hyperparameters [34, 71, 31], but optimizing hyperparameters in BNNs has not been widely adopted, although there are exceptions [50, 72]. Some authors heuristically propose using a Gaussian centered on the MAP estimate as a prior for BNNs [38, 29]. The Laplace approximation has been used for hyperparameter selection in BNNs [49, 51, 61], but it does not use the ELBO. Hierarchical models in BNNs have been explored in many works [23, 24, 46, 33, 54, 58, 3, 2]. The closest to our approach is learning a hierarchical horseshoe prior over BNNs weights [23]. However, their approach 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 [54, 74, 32, 11] and arguing against [70]. 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 [17] and others arguing against [69, 19, 18]. A number of problems of mean-field VI BNNs have been exposed in shallow networks including over-pruning [69] and poor uncertainty in output space [19, 18, 24]. To resolve under-fitting of mean-field BNNs, previous work has re-weighted terms arising from the KL divergence e.g. [45].
4 Experiments

In this section we explore the predictive performance of the introduced variational bounds and benchmark them together with other algorithms. We find that using the bounds in Eq. (9) and Eq. (10) significantly improves the predictive performance of standard MF-VI BNNs (fixed Gaussian priors). We also find that learning the prior means according to Eq. (9) is more effective and gives sufficiently good predictive performance, whereas learning prior variances (Eq. (10)) mostly does not lead to significant gains in the quality of predictions. We defer the details of the experimental setup to Appendix C and report additional experimental results in Appendix D.

The algorithms we propose in this paper optimize various tighter ELBOs: CM-MF-VI optimizes $\mathcal{L}_m$ from Eq. (9); CM-MF-VI OPT additionally optimizes $\alpha_{\text{reg}}$ using the tighter ELBO (Eq. (9)); CV-MF-VI optimizes for just prior variances, using the bound $\mathcal{L}_{\alpha_{\text{opt}}}$ from Eq. (10) when $\delta \to 1$ (prior mean $\mu_p$ fixed to 0); and CMV-MF-VI optimizes both prior means and variances from Eq. (10). We compare to the following baselines: MF-VI LRT learns zero-mean mean-field Gaussian priors with fixed variances [55] (which we call standard MF-VI) using local reparametrization [38]; MF-VI FV learns a mean-field Gaussian posterior with fixed variances [68]; MF-VI BD is mean-field Gaussian VI while down-weighting KL penalty by 0.5; MF-VI EB learns prior means with ELBO gradient updates [51, 72]; BBB is Bayes by Backprop using the ADAM optimizer (as opposed to SGD in [7]); MF-VI $k$-tied normal distribution [63]; ELRG/SLANG learn low rank posteriors [66, 53].

Figure 3: (a.) Pruned weights in standard MF-VI contribute noise to predictions as noise leaks through the activation function. (b.) Removing pruned weights from BNN can improve performance, as the removed weights contribute only to the predictive variance and cause under-fitting.
VI, CV-MF-VI, CMV-MF-VI outperform MF-VI on all data (only CMV-MF-VI is worse on wine), show posterior log variances instead of posterior means. Most weights have been pruned. In the top row plots (c.) and (e.) the posterior means, where red is the means of the bias parameters, variances of pruned units are all very small.

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

<table>
<thead>
<tr>
<th>algorithm</th>
<th>test NLL ↓</th>
<th>test ER ↓</th>
<th>test NLL ↓</th>
<th>test ER ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM-MF-VI</td>
<td>0.047 ± 0.006</td>
<td>1.34 ± 0.24%</td>
<td>0.048 ± 0.005</td>
<td>1.42 ± 0.26%</td>
</tr>
<tr>
<td>CV-MF-VI</td>
<td>0.068 ± 0.007</td>
<td>2.13 ± 0.34%</td>
<td>0.066 ± 0.003</td>
<td>2.09 ± 0.10%</td>
</tr>
<tr>
<td>CMV-MF-VI</td>
<td>0.049 ± 0.005</td>
<td>1.45 ± 0.25%</td>
<td>0.052 ± 0.004</td>
<td>1.60 ± 0.01%</td>
</tr>
<tr>
<td>CM-MF-VI (4000 batch)</td>
<td>0.041 ± 0.002</td>
<td>1.31 ± 0.04%</td>
<td>0.042 ± 0.002</td>
<td>1.20 ± 0.06%</td>
</tr>
<tr>
<td>MF-VI LRT</td>
<td>0.094 ± 0.002</td>
<td>2.42 ± 0.28%</td>
<td>0.099 ± 0.001</td>
<td>2.58 ± 0.07%</td>
</tr>
<tr>
<td>MF-VI BD</td>
<td>0.092 ± 0.002</td>
<td>2.43 ± 0.06%</td>
<td>0.099 ± 0.004</td>
<td>2.56 ± 0.22%</td>
</tr>
<tr>
<td>MF-VI FV</td>
<td>0.052 ± 0.001</td>
<td>1.67 ± 0.13%</td>
<td>0.053 ± 0.004</td>
<td>1.58 ± 0.26%</td>
</tr>
<tr>
<td>BBB ADAM</td>
<td>0.095 ± 0.008</td>
<td>2.48 ± 0.45%</td>
<td>0.097 ± 0.007</td>
<td>2.49 ± 0.33%</td>
</tr>
<tr>
<td>ELRG-VI K = 5</td>
<td>0.053 ± 0.006</td>
<td>1.54 ± 0.18%</td>
<td>0.058 ± 0.005</td>
<td>1.68 ± 0.17%</td>
</tr>
<tr>
<td>K-TIED K = 10</td>
<td>0.105 ± 0.004</td>
<td>2.67 ± 0.16%</td>
<td>0.108 ± 0.004</td>
<td>2.61 ± 0.17%</td>
</tr>
<tr>
<td>MF-VI SGD</td>
<td>—</td>
<td>1.82%</td>
<td>—</td>
<td>1.99%</td>
</tr>
<tr>
<td>SLANG K=32</td>
<td>—</td>
<td>1.72%</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Figure 4: Learning prior parameters in BNN fixes under-fitting caused by over-pruning. In (c.),(d.), (e.) and (f.) we show the distribution of statistics of variational posterior over weights in the first layer. The top row plots (c.) and (e.) the posterior means, where red is the means of the bias parameters, and black is the means of the weight parameters into a hidden unit. The bottom row plots (d.) and (f.) show posterior log variances instead of posterior means. Most weights have been pruned. In CM-MF-VI, shown in (a.), units where posterior weights converge to prior have posterior biases set to low value, which saturates the activation stopping them from influencing the predictions. We see this in (c.), where the pruned units all have biases with negative posterior means. This saturation is made possible by weaker mean regularization in Eq. (9) governed by $\alpha_{reg}$. In CMV-MF-VI, shown in (b.), pruned units converge to spiked Gaussians. We see this in the (f.), where the posterior log variances of pruned units are all very small.

UCI regression. We first consider 20 train-test splits for 8 UCI regression data sets. We learn 2 hidden layer BNNs (results for 1 hidden layer in Appendix D) with 50 units and ReLU activations, but we use a heteroscedastic observation model $p(y|f^{1}_W(x), f^{2}_W(x)) = \mathcal{N}(y|f^{1}_W(x), \exp(f^{2}_W(x)))$, where $f^{1}_W(x)$, $f^{2}_W(x)$ are two heads of the network. We optimize the objectives for 200K steps with the ADAM optimizer with default settings. We report a comparison of test log-likelihood in Figure 2 (top). We observe that (i) the collapsed bounds CM-MF-VI, CV-MF-VI, CMV-MF-VI outperform MF-VI on all data (only CM-MF-VI is worse on wine), (ii) 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 except for 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 that the ELBO can be used to learn hyperparameter $\alpha_{reg}$. Next we report a comparison between CM-MF-VI and standard MF-VI as the depth of the network increases in Figure 2 (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. (9) or Eq. (10) allows to mitigate rapid degradation of
predictions. We now explain why standard MF-VI performs badly, finding this is due to over-pruning in BNNs. We investigate this effect and find that learning prior means and variances mitigates the effect of pruned units on the predictive distribution.

**Over-pruning in BNNs.** Applying mean-field VI to BNN prunes most of the posterior weights, i.e. approximate posteriors converge to prior $\mathcal{N}(0, \mathcal{I})$. While this is beneficial for compressing the model, pruned weights contribute noise to predictions and make modeling data difficult. With 1 hidden layer networks and the regular ELBO, output layer weights are set as close to 0 as possible $\mathcal{N}(0, \mathcal{I})$, thereby reducing the noise from pruned weights in earlier layers. However, some noise still leaks through, as output layer weights are not quite delta-functions. This noise is one of the reasons for increasing under-confidence in output space as the size of the network increases. In Figure 3 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 3 (right) we plot test NLL as pruned 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 lowers test NLL, meaning that pruned weights contribute only noise to predictions. This is schematically explained in Figure 3 (a), which shows pruned posterior weights injecting noise. See Appendix E for further description of over-pruning.

**Fixing under-fitting caused by over-pruning in BNNs.** We now explain why learning the hyperparameters of the Gaussian prior in the BNN mitigates problems arising from over-pruning. In short, learning the means and variances of Gaussian priors mitigates the influence of network units that have been pruned. These units otherwise cause under-fitting by contributing only to predictive variance. 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 output layer. We use 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 4 hence they do not influence predictions, shown in Figure 4 (b).

**MNIST classification with MLP.** Classifying vectorized MNIST images using two hidden layer network with ReLU activation is a standard benchmark for BNNs. 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 (4000 batch) uses batches of 4000 images for an optimization step and achieves the best performance across the tested models, showing the developed algorithms can leverage low variance updates.

**Perturbed MNIST images.** We now investigate if the in-domain improvements of CM-MF-VI/CMV-MF-VI come at the cost of out-of-distribution (OOD) performance. We find this is not the case: OOD performance is as good as before. To show this, we learn a 2 hidden layer BNN with ReLU activations on vectorized MNIST images as previously, but test it on the fashionMNIST data.
which allows us to control the strength of regularization of the model. We analyze the test predictive

<table>
<thead>
<tr>
<th>Table 2: Image classification with LeNet CNN. Collapsed bounds visibly outperform standard MF-VI.</th>
</tr>
</thead>
<tbody>
<tr>
<td>test NLL \downarrow / ER \downarrow</td>
</tr>
<tr>
<td>CM-MF-VI</td>
</tr>
<tr>
<td>CM-MF-VI</td>
</tr>
<tr>
<td>CV-MF-VI</td>
</tr>
<tr>
<td>CM-MF-VI OPT</td>
</tr>
<tr>
<td>MF-VI</td>
</tr>
<tr>
<td>MAP</td>
</tr>
<tr>
<td>MC dropout</td>
</tr>
<tr>
<td>MF-VI EB</td>
</tr>
<tr>
<td>CM-MF-VI</td>
</tr>
<tr>
<td>CM-MF-VI</td>
</tr>
<tr>
<td>CV-MF-VI</td>
</tr>
<tr>
<td>CM-MF-VI OPT</td>
</tr>
<tr>
<td>MF-VI</td>
</tr>
<tr>
<td>MAP</td>
</tr>
<tr>
<td>MC dropout</td>
</tr>
<tr>
<td>MF-VI EB</td>
</tr>
</tbody>
</table>

Figure 6: Top: performance of CM-MF-VI for different settings of hyperparameter \( \alpha \) and different network architectures (\( \alpha = 1 \) is standard MF-VI). Setting \( \alpha = 0.01 \) visibly improves predictions compared to standard MF-VI (fixed Gaussian prior). Bottom: ELBO \( \mathcal{L}_n \) correlates with test NLL.

set \( \mathcal{L}_n \) 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.

**Limitations of the algorithm: adjusting hyperparameters.** The bounds in Eq. (9) and Eq. (10) have hyperparameters, corresponding to the choice of hyper-prior. These additional parameters are the main limitation of the derived algorithms. For example, CM-MF-VI has one hyperparameter \( \alpha_{reg} \) which allows us to control the strength of regularization of the model. We analyze the test predictive performance for different settings of \( \alpha_{reg} \) 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 \( \alpha_{reg} \) to 0.2 provides strong regularization and
We developed a family of algorithms optimizing variational posteriors in BNNs based on collapsed variational bounds. We demonstrated that learning the prior parameters of BNN weights fixes their predictive under-confidence resulting in good empirical performance and robustness to over-fitting. The developed algorithms allowed us to demonstrate that the ELBO can be a suitable optimization target for learning hyperparameters of BNNs. 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 based approximate inference in large network architectures.

## 5 Conclusions

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

<table>
<thead>
<tr>
<th>model</th>
<th>test NLL</th>
<th>CMV-MF-VI</th>
<th>CM-MF-VI</th>
<th>CV-MF-VI</th>
<th>MF-VI</th>
<th>MC dropout</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESNET18</td>
<td>STL10</td>
<td>1.04 ±0.00</td>
<td>1.10 ±0.02</td>
<td>1.57 ±0.01</td>
<td>1.61 ±0.01</td>
<td>1.17 ±0.05</td>
<td>1.64 ±0.03</td>
</tr>
<tr>
<td></td>
<td>SVHN</td>
<td>0.15 ±0.00</td>
<td>0.14 ±0.00</td>
<td>0.20 ±0.01</td>
<td>0.22 ±0.00</td>
<td>0.18 ±0.00</td>
<td>0.35 ±0.01</td>
</tr>
<tr>
<td></td>
<td>CIFAR10</td>
<td>1.43 ±0.01</td>
<td>1.53 ±0.00</td>
<td>2.00 ±0.01</td>
<td>2.23 ±0.03</td>
<td>1.75 ±0.00</td>
<td>1.45 ±0.05</td>
</tr>
<tr>
<td></td>
<td>CIFAR10</td>
<td>0.41 ±0.00</td>
<td>0.39 ±0.00</td>
<td>0.59 ±0.00</td>
<td>0.68 ±0.02</td>
<td>0.49 ±0.00</td>
<td>0.95 ±0.02</td>
</tr>
<tr>
<td>SHUFFLENET</td>
<td>STL10</td>
<td>0.96 ±0.01</td>
<td>0.99 ±0.02</td>
<td>1.22 ±0.05</td>
<td>1.70 ±0.08</td>
<td>1.28 ±0.01</td>
<td>1.78 ±0.08</td>
</tr>
<tr>
<td></td>
<td>SVHN</td>
<td>0.27 ±0.01</td>
<td>0.26 ±0.01</td>
<td>0.31 ±0.01</td>
<td>0.31 ±0.01</td>
<td>1.11 ±0.01</td>
<td>0.32 ±0.00</td>
</tr>
<tr>
<td></td>
<td>CIFAR10</td>
<td>2.05 ±0.01</td>
<td>1.99 ±0.02</td>
<td>2.22 ±0.06</td>
<td>2.28 ±0.02</td>
<td>3.06 ±0.01</td>
<td>1.48 ±0.03</td>
</tr>
<tr>
<td></td>
<td>CIFAR10</td>
<td>0.65 ±0.00</td>
<td>0.65 ±0.00</td>
<td>0.71 ±0.01</td>
<td>0.72 ±0.01</td>
<td>1.24 ±0.00</td>
<td>1.10 ±0.02</td>
</tr>
<tr>
<td>ALEXNET</td>
<td>STL10</td>
<td>1.50 ±0.07</td>
<td>1.48 ±0.04</td>
<td>1.80 ±0.07</td>
<td>1.86 ±0.03</td>
<td>1.07 ±0.06</td>
<td>1.80 ±0.32</td>
</tr>
<tr>
<td></td>
<td>SVHN</td>
<td>0.30 ±0.00</td>
<td>0.28 ±0.00</td>
<td>0.42 ±0.01</td>
<td>0.51 ±0.01</td>
<td>0.38 ±0.01</td>
<td>0.72 ±0.06</td>
</tr>
<tr>
<td></td>
<td>CIFAR10</td>
<td>2.23 ±0.02</td>
<td>2.16 ±0.01</td>
<td>2.62 ±0.02</td>
<td>2.95 ±0.04</td>
<td>2.19 ±0.01</td>
<td>7.03 ±10.14</td>
</tr>
<tr>
<td></td>
<td>CIFAR10</td>
<td>0.72 ±0.01</td>
<td>0.69 ±0.01</td>
<td>0.99 ±0.01</td>
<td>1.19 ±0.01</td>
<td>0.74 ±0.01</td>
<td>1.79 ±0.06</td>
</tr>
</tbody>
</table>

*Image classification LeNet.* We now consider image classification with the LeNet architecture [11] on 6 data sets: MNIST, fashionMNIST, K-MNIST [10], CIFAR10, CIFAR100 [11], and SVHN [50]. We optimize the objectives for 800 epochs (except MAP for 50 epochs and MC dropout for 100 epochs as they tend to overfit) 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 [25], ShuffleNet [49] and AlexNet [42]. We use CIFAR10, CIFAR100, STL10 [11] 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 57, and average results over 3 random seeds. We gather the results in Table 7. Experiments with large CNNs confirm our findings from previous experiments: (i) CM-FM-VI/CM-FM-VI always outperform standard MF-VI by a visible margin and result in good predictive performance (e.g. outperforming SOTA VOG [57]), (ii) learning prior means CM-MF-VI/CM-MF-VI outperforms learning just prior variances (CV-MF-VI).
Acknowledgements

Marcin B. Tomczak is funded by a Cambridge Trust scholarship. Siddharth Swaroop is supported by an EPSRC DTP studentship and a Microsoft Research EMEA PhD Award. Andrew Y. K. Foong is supported by a Trinity Hall Research Studentship and the George and Lilian Schiff Foundation. Richard E. Turner is supported by Google, Amazon, ARM, Improbable, EPSRC grants EP/M026957/1 and EP/L000776/1, and the UKRI Centre for Doctoral Training in the Application of Artificial Intelligence to the study of Environmental Risks (AI4ER). The authors thank David Burt for insightful discussions.

References


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


