Projected BNNs: Avoiding weight-space pathologies by learning latent representations of neural network weights

Melanie F. Pradier[(1)], Weiwei Pan[(1)], Jiayu Yao[(1)],
Soumya Ghosh[(2)], Finale Doshi-Velez[(1)]

(1) Harvard University
(2) IBM Research

Abstract

While modern neural networks are making remarkable gains in terms of predictive accuracy, characterizing uncertainty over the parameters of these models (in a Bayesian setting) is challenging because of the high-dimensionality of the network parameter space and the correlations between these parameters. In this paper, we introduce a novel framework for variational inference for Bayesian neural networks that (1) encodes complex distributions in high-dimensional parameter space with representations in a low-dimensional latent space and (2) performs inference efficiently on the low-dimensional representations. Across a large array of synthetic and real-world datasets, we show that our method improves uncertainty characterization and model generalization when compared with methods that work directly in the parameter space.

1 Introduction

Deep learning provides a flexible framework for function approximation and, as a result, deep models have become a standard approach in many domains including machine vision, natural language processing, speech recognition, bioinformatics, and game-playing [LeCun et al., 2015]. However, deep models tend to overfit when the number of training examples is small; furthermore, in practice, the primary focus in deep learning is often on computing point estimates of model parameters, and thus these models do not provide uncertainties for their predictions – making them unsuitable for applications in critical domains such as personalized medicine. Bayesian neural networks (BNN) promise to address these issues by modeling the uncertainty in the network weights, and correspondingly, the uncertainty in output predictions[MacKay, 1992b, Neal, 2012].

Unfortunately, characterizing uncertainty over parameters of modern neural networks in a Bayesian setting is challenging due to the high-dimensionality of the weight space and complex patterns of dependencies among the weights. In these cases, Markov-chain Monte Carlo (MCMC) techniques for performing inference often fail to mix across the weight space, and standard variational approaches not only struggle to escape local optima, but also fail to capture dependencies between the weights. A recent body of work has attempted to improve the quality of inference for Bayesian neural networks (BNNs) via improved approximate inference methods [Graves, 2011, Blundell et al., 2015, Hernández-Lobato et al., 2016], or by improving the flexibility of the variational approximation for variational inference [Gershman et al., 2012, Ranganath et al., 2016, Louizos and Welling, 2017].

In this work, we introduce a novel approach in which we remove potential redundancies in neural network parameters by learning a non-linear projection of the weights onto a low-dimensional latent space. Our approach takes advantage of the following insight: learning (standard network) *parameters* is easier in the high-dimensional space, but characterizing (Bayesian) *uncertainty* is easier in the

low-dimensional space. Low-dimensional spaces are generally easier to explore, especially if we have fewer correlations between dimensions, and can be better captured by standard variational approximations (e.g. mean field). At the same time, the non-linear transformation between latent space and weight space allows us to encode flexible approximating distributions for posteriors over weights.

Our main contribution is a model that encodes the uncertainty in the weights of a neural network via a low dimensional latent space as well as a framework for performing high-quality inference on this model. We demonstrate, on synthetic datasets, the ability of our model to capture complex posterior distributions over weights by encoding them as distributions in latent space. We show that, as a result, our model is able to more accurately capture the uncertainty in the posterior predictive distribution. Finally, we demonstrate that, across a wide range of real-world data sets, our approach produces accurate predictions on held-out data with highly compressed latent representations of the weights.

2 Related Work

Classic work on Bayesian neural networks can be traced back to [Buntine and Weigend, 1991, MacKay, 1992a, Neal, 1993]. Neal [1993] introduced Hamiltonian Monte Carlo (HMC) to explore the Bayesian neural network posterior. MacKay [1992a] andBuntine and Weigend [1991] instead relied on Laplace approximation to characterize the network posterior. While HMC remains the "gold standard" for posterior inference in BNNs, it does not scale well to modern architecture or large datasets. Similarly, vanilla application of the Laplace approximation has difficulty scaling to modern architectures with millions of parameters.

Many recent works in variational inference have attempted to move beyond fully-factorized approximations of Bayesian neural network posteriors by modeling structured correlations amongst BNN weights. While nearly all of these approaches perform inference directly on the weight space [Sun et al., 2017, Louizos and Welling, 2016, Gal and Ghahramani, 2016], we perform inference in a latent space of lower dimensionality. Most comparable to our approach are works that build flexible approximating families of distributions using auxiliary random variables, either by mixtures of simple distributions [Agakov and Barber, 2004, Maaløe et al., 2016, Ranganath et al., 2016, Salimans et al., 2013] or by non-linear transformations of simple distributions [Rezende and Mohamed, 2015, Kingma et al., 2016, Louizos and Welling, 2017].

In particular, Louizos and Welling [2017] assume Normal-distributed variational distributions for the weights with *layer-wise* multiplicative noise that are *linearly* projected onto a latent space via normalizing flows. However, they still assume that the posterior factorizes layer wise. In contrast, our approach learns *non-linear* projections of the *entire* network weights onto a latent space. Furthermore, our generative model differs from the one in Louizos and Welling [2017] and allows us to optimize a tighter bound on the log evidence.

In terms of modeling, our work is close in spirit to [Karaletsos et al., 2018], where the authors represent nodes in a neural network by latent variables via a *deterministic linear* projection, drawing network weights conditioned on those representations. In contrast to their approach, we learn a *distribution* over *non-linear* projections, and find a latent representation for the *weights* directly rather than projecting the *nodes*.

A number of recent works use the idea of hypernetworks, neural networks that output parameters of other networks, to parametrize the variational distribution in a flexible manner [Krueger et al., 2017, Pawlowski et al., 2017].These works perform inference directly over weights, requiring one to use invertible latent projections or otherwise approximate implicit weight densities. Our approach of performing inference in the latent space avoids these challenges. Furthermore, while uncertainty over the latent projection is not considered in Krueger et al. [2017], Pawlowski et al. [2017], we incorporate this uncertainty explicitly in both our generative and variational models.

Another interesting line of work relies on (non-Bayesian) neural network ensembles to estimate predictive uncertainty [Lakshminarayanan et al., 2017, Pearce et al., 2018]. The idea is simple: each network in the ensemble will learn similar values close to the training data, and different ones in regions of the space far from the training data. Whereas Lakshminarayanan et al. [2017] rely on multiple random restarts and adversarial training, Pearce et al. [2018] introduce noise in the regularization term of each network, which directly relates to the randomized MAP sampling literature [Lu and Van Roy, 2017]. All these work have the disadvantage of being heuristic. In [Pearce et al., 2018], the authors derive a parallelism between their ensemble sampling approach and Bayesian behavior, but this only holds for infinite single-layer neural networks. In contrast, our approach relies on NN ensembles to initialize the variational inference scheme, making it more principled.

Finally, the dimensionality reduction aspect of our model is comparable to neural network compression. There are a number of non-Bayesian methods for compressing standard neural networks, all of which, to our knowledge, rely on linear dimensionality reduction techniques on the space of weights and or network nodes [Denil et al., 2013, Sainath et al., 2013, Xue et al., 2013, Nakkiran et al., 2015]. In contrast, our work is focused on non-linear dimensionality reduction in a Bayesian setting with the aim to improve uncertainty quantification.

3 Background and Notation

Let $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots (\mathbf{x}_N, \mathbf{y}_N)\}$ be a dataset of N i.i.d observed input-output pairs. We model this data by $\mathbf{y} = f_{\boldsymbol{w}}(\mathbf{x}) + \epsilon$, where ϵ is a noise variable and $\boldsymbol{w}$ refers to the weights of a neural network.[1]. In the Bayesian setting, we assume some prior over the weights $\boldsymbol{w} \sim p(\boldsymbol{w})$. One common choice is to posit i.i.d normal priors over each network weight $w_i \sim \mathcal{N}\left(0, \sigma_w^2\right)$.

Our objective is to infer the posterior distribution over functions $p(f_{\boldsymbol{w}}|\mathcal{D})$, which is equivalent to inferring the posterior distribution over the weights $p(\boldsymbol{w}|\mathcal{D})$. Given the posterior distribution, we model predictions for new observations and their associated uncertainties through the posterior predictive distribution:

$$p(\mathbf{y}^\star|\mathbf{x}^\star, \mathcal{D}) = \int p(\mathbf{y}^\star|\mathbf{x}^\star, \boldsymbol{w}) p(\boldsymbol{w}|\mathcal{D}) d\boldsymbol{w}. \tag{1}$$

The posterior $p(\boldsymbol{w}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})$ generally does not have an analytic form due to the non-linearity of $f_{\boldsymbol{w}}$; thus, one must resort to approximate inference techniques. Variational inference, for example, attempts to find a distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{w})$ that closely approximates the true posterior $p(\boldsymbol{w}|\mathcal{D})$. The measure of proximity is typically the KL-divergence:

$$\begin{aligned} & D_{\mathrm{KL}}\big(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})\big) \\ &= \int q_{\boldsymbol{\lambda}}(\boldsymbol{w})[\log q_{\boldsymbol{\lambda}}(\boldsymbol{w}) - \log p(\boldsymbol{w}|\mathcal{D})] d\boldsymbol{w} \\ &= -\mathcal{H}(q) - \mathbb{E}_q[\log p(\mathcal{D}, \boldsymbol{w})] + \log p(\mathcal{D}) \\ &= -\mathcal{L}(\boldsymbol{\lambda}) + \log p(\mathbf{y}|\mathbf{x}). \end{aligned} \tag{2}$$

where $\mathcal{L}(\boldsymbol{\lambda}) = \mathcal{H}(q) + \mathbb{E}_q[\log p(\mathcal{D}, \boldsymbol{w})]$ is the evidence lower bound (ELBO) on the marginal likelihood $\log p(\mathbf{y}|\mathbf{x})$. Approximate inference in this case can be cast as the problem of optimizing $\mathcal{L}(\boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$, since minimizing the KL-divergence between $q_{\boldsymbol{\lambda}}(\boldsymbol{w})$ and $p(\boldsymbol{w}|\mathcal{D})$ is equivalent to maximixing the lower bound $\mathcal{L}(\boldsymbol{\lambda})$, as shown in Eq. (2).

For BNNs, the posterior distribution $p(\boldsymbol{w}|\mathcal{D})$ often contains strong correlations, such that the optimization of $\boldsymbol{\lambda}$ is prone to get stuck in local optima. Moreover, simple variational families $q_{\boldsymbol{\lambda}}(\boldsymbol{w})$, such as mean field approximations, fail to capture those correlations.

[1] In this paper, $\boldsymbol{w}$ refers to both, network weights and biases, since each layer can be augmented with an extra input dimension containing 1's to account for the biases.

4 Latent Projection BNN

4.1 Generative Model

In our approach, which we call *Projected Bayesian Neural Network (Proj-BNN)*, we posit that the neural network weights $\boldsymbol{w}$ are generated from a latent space or *manifold* of much smaller dimensionality. That is, we assume the following generative model:

$$\boldsymbol{z} \sim p(\boldsymbol{z}), \quad \boldsymbol{\phi} \sim p(\boldsymbol{\phi}), \quad \boldsymbol{w} = g_{\boldsymbol{\phi}}(\boldsymbol{z}), \quad \mathbf{y} \sim \mathcal{N}(f_{\boldsymbol{w}}(\mathbf{x}), \sigma_y^2) \tag{3}$$

where $\boldsymbol{w}$ lies in $\mathbb{R}^{D_w}$, the latent representation $\boldsymbol{z}$ lie in a lower dimensional space $\mathbb{R}^{D_z}$, and $\boldsymbol{\phi}$ parametrizes the arbitrary projection function $g_{\boldsymbol{\phi}} : \mathbb{R}^{D_z} \to \mathbb{R}^{D_w}$. Given this generative model, our objective is to compute the joint posterior distribution $p(\boldsymbol{z}, \boldsymbol{\phi}|\mathbf{y}, \mathbf{x})$ over both the latent representation $\boldsymbol{z}$ and the latent projection parameters $\boldsymbol{\phi}$.

4.2 Inference

Following the same structure in the generative model, we propose a variational distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi}) = q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}) q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi})$ such that:

$$\boldsymbol{z} \sim q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}), \ \boldsymbol{\phi} \sim q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi}), \ \boldsymbol{w} = g_{\boldsymbol{\phi}}(\boldsymbol{z}). \tag{4}$$

In particular, we posit a mean-field posterior approximation for each independent term $q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}) \doteq \mathcal{N}(\tilde{\boldsymbol{\mu}}_z, \tilde{\boldsymbol{\sigma}}_z \mathbf{I})$ and $q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi}) \doteq \mathcal{N}(\tilde{\boldsymbol{\mu}}_\phi, \tilde{\boldsymbol{\sigma}}_\phi \mathbf{I})$, where $\tilde{\boldsymbol{\mu}}_z, \tilde{\boldsymbol{\mu}}_\phi$ and $\tilde{\boldsymbol{\sigma}}_\phi, \tilde{\boldsymbol{\sigma}}_\phi$ refer to the mean and standard deviation vectors of each Normal variational distribution respectively, $\boldsymbol{\lambda}_z = \{\tilde{\boldsymbol{\mu}}_z, \tilde{\boldsymbol{\sigma}}_z\}$, and $\boldsymbol{\lambda}_\phi = \{\tilde{\boldsymbol{\mu}}_\phi, \tilde{\boldsymbol{\sigma}}_\phi\}$. Note that, although we adopt a fully factorized posterior approximation for $\boldsymbol{z}$ and $\boldsymbol{\phi}$, the induced posterior approximation $q_{\boldsymbol{\lambda}}(\boldsymbol{w})$ over the network weights $\boldsymbol{w}$ can capture complex dependencies because of the non-linear transformation $g_{\boldsymbol{\phi}}$.

Given such variational distribution, a straightforward inference algorithm is to use black-box variational inference (BBVI) [Ranganath et al., 2014] with the local reparametrization trick [Kingma et al., 2015] to minimize the joint evidence lower bound (ELBO) $\mathcal{L}(\boldsymbol{\lambda})$ given by:

$$\mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q \Big[\log p\big(\mathbf{y}|\mathbf{x}, g_{\boldsymbol{\phi}}(\boldsymbol{z})\big) \Big] - D_{\mathrm{KL}}\big(q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}) || p(\boldsymbol{z})\big) - D_{\mathrm{KL}}\big(q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi}) || p(\boldsymbol{\phi})\big). \tag{5}$$

However, jointly optimizing the projection parameters $\boldsymbol{\phi}$ and latent representation $\boldsymbol{z}$ is a hard optimization problem; direct optimization of the ELBO in Eq. (5) is prone to local minima and leads to poor performance solutions. To alleviate this issue, we first come up with an intelligent initialization (no uncertainty) for both $\boldsymbol{\phi}$ and $\boldsymbol{z}$, and then perform principled variational inference using BBVI. In the following, we describe the complete inference framework in three stages.

Characterize the space of plausible weights. In the first step, we seek to gather a diverse set of weight parameters (without considering uncertainty), which will be used to learn a smart initialization of $\boldsymbol{\phi}$ and $\boldsymbol{z}$ afterwards. As explained in Section 1, learning parameters in the high-dimensional $\boldsymbol{w}$-space is easy, the difficult part is to get accurate uncertainty estimations. More precisely, we collect multiple, high-quality candidate weight solutions $\{\mathbf{w_c}^{(r)}\}_{r=1}^{R}$ by training an ensemble of R neural networks over multiple restarts. Indeed, NN ensembles have been shown to provide diverse sets of predictive functions while remaining accurate. Lakshminarayanan et al. [2017], Pearce et al. [2018] Initializing from multiple restarts allow us to recover a variety of weights which will lead to similar function values where there is training data, but different outputs otherwise. Although there exist methods in the literature to force diversity in the solutions obtained by ensembles of neural networks, for simplicity, we opt for the simplest approach using multiple restarts at different initialization points, only keeping the best restarts in terms of log-likelihood in the validation set.[2]

[2]For instance, our pipeline implementation filters out the top-10% of the worst restarts.

Learn a point-estimate for the projection function. In order to find an intelligent initialization in the BBVI algorithm for the projection function $g_{\boldsymbol{\phi}}$ and latent representation $\boldsymbol{z}$, we perform dimensionality reduction on the previously collected weights $\{\mathbf{w_c}^{(r)}\}_{r=1}^{R}$. We opt for an autoencoder to account for non-linear complex transformations, but other alternatives can also be applied. Let $f_{\boldsymbol{\theta}} : D_w \to D_z$ and $g_{\boldsymbol{\phi}} : D_z \to D_w$ denote the encoder and decoder, respectively, of the autoencoder $h_{\boldsymbol{\theta},\boldsymbol{\phi}}$. Our aim is to find a point estimate for the parameters of the projection.

While we want to find latent projections that minimize the reconstruction error of the weights, at the same time, we also need to explicitly encourage for projections that will map into weights that yield high log likelihood values for our original training data $\mathcal{D}$. We find that, in practice, the explicit constraint on the predictive accuracy of reconstructed weights is required since weights that are "similar" in Euclidean norm may yield models of very different predictive qualities.[3] This results in the following loss to minimize:

$$\{\boldsymbol{\theta}^\star, \boldsymbol{\phi}^\star\} = \underset{\boldsymbol{\theta},\boldsymbol{\phi}}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \min_{\boldsymbol{\theta},\boldsymbol{\phi}} \Bigg\{ \frac{1}{R}\sum_{r=1}^{R} \Big(\mathbf{w_c}^{(r)} - g_{\boldsymbol{\phi}}\Big(f_{\boldsymbol{\theta}}\Big(\mathbf{w_c}^{(r)}\Big)\Big) + \gamma^{(r)}\Big)^2 + \beta\, \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\frac{1}{R}\sum_{r=1}^{R} \log p\big(y|x, g_{\boldsymbol{\phi}}\Big(f_{\boldsymbol{\theta}}\Big(\mathbf{w_c}^{(r)}\Big)\Big)\right] \Bigg\}, \tag{6}$$

where $\mathbf{w_c}^{(r)}$ denotes the set of weights gathered from the r-th network in the ensemble of R neural networks, and $\gamma^{(r)} \sim \mathcal{N}(0, 1)$ is an additional input noise term that makes training more robust. The first term in Eq. (6) corresponds to the average mean square error of each $\mathbf{w_c}^{(r)}$ and its reconstructed version $\hat{\mathbf{w}}_\mathbf{c}^{(r)} = g_{\boldsymbol{\phi}}\big(f_{\boldsymbol{\theta}}\big(\mathbf{w_c}^{(r)}\big)\big)$, while the second term accounts for the reconstruction error (in terms of log likelihood) of the original output data $\mathbf{y}$ given $\mathbf{x}$ and $\hat{\mathbf{w}}_\mathbf{c}^{(r)}$. We call this approach a *prediction-constrained* autoencoder. Prediction-constrained models have previously been introduced in the context of mixture and topic models [Hughes et al., 2017, 2018].

Learn the approximate posterior $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})$. Given the point-estimate projection parameters $\boldsymbol{\phi}^\star$ from the previous stage, we can now initialize the mean variational parameters for $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})$, and perform principled posterior approximation using black-box variational inference. In particular, we optimize the variational parameters $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_z, \boldsymbol{\lambda}_\phi\}$ to minimize the KL-divergence $D_{\mathrm{KL}}\big(q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})||p(\boldsymbol{z}, \boldsymbol{\phi}|\mathbf{y}, \mathbf{x})\big)$. For simplicity, we assume a mean-field structure $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi}) = q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}) q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi})$ for the latent representation $\boldsymbol{z}$ and projection parameters $\boldsymbol{\phi}$. To facilitate the optimization task, we first optimize the variational distribution in latent space $q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})$ (assuming $\boldsymbol{\phi}$ fixed) via black-box variational inference (BBVI) [Ranganath et al., 2014] with the local reparametrization trick [Kingma et al., 2015], after which we jointly fine-tune the uncertainty in both the latent space and the projection parameter. In other words, we proceed to optimize two different evidence lower bounds (ELBO). We first assume an approximate mean-field variational distribution $q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})$ in the latent space. The following lower-bound $\mathcal{L}(\boldsymbol{\lambda}_z)$ on the marginal log-likelihood can be derived:

$$\log p(\mathbf{y}|\mathbf{x}) \geq \int q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}) \log \frac{p\big(\mathbf{y}|\mathbf{x}, g_{\boldsymbol{\phi}}(\boldsymbol{z})\big)p(\boldsymbol{z})}{q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})} d\boldsymbol{z} \tag{7}$$

$$\mathcal{L}(\boldsymbol{\lambda}_z) = \mathbb{E}_q\Big[\log p\big(\mathbf{y}|\mathbf{x}, g_{\boldsymbol{\phi}}(\boldsymbol{z})\big)\Big] - D_{\mathrm{KL}}\big(q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})||p(\boldsymbol{z})\big). \tag{8}$$

The expectation in Eq. (8) and gradient can be estimated using simple Monte Carlo integration along with the reparametrization trick [Kingma and Welling, 2013, Rezende et al., 2014]:

$$\mathbb{E}_q\Big[\log p\big(\mathbf{y}|\mathbf{x}, g_{\boldsymbol{\phi}}(\boldsymbol{z})\big)\Big] \approx \frac{1}{S}\sum_{s=1}^{S} \log p\big(\mathbf{y}|\mathbf{x}, g_{\boldsymbol{\phi}}(\boldsymbol{z}^{(s)})\big)), \tag{9}$$

[3] If we only optimize for minimum weight reconstruction, we find projections whose reconstructed weights have lower quality in terms of test log likelihood $p(\mathbf{y}|\mathbf{x}, \boldsymbol{w_c})$.

where $\boldsymbol{z}^{(s)} = \boldsymbol{\epsilon}\, \tilde{\boldsymbol{\sigma}} + \tilde{\boldsymbol{\mu}}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathrm{I})$ is the auxiliary noise variable for the reparametrization trick, and $\boldsymbol{\lambda}_z = \{\tilde{\boldsymbol{\mu}}_z, \tilde{\boldsymbol{\sigma}}_z\}$ are the variational parameters of the distribution $q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})$. The KL-divergence term in Eq. (8) can be computed in closed form, as both distributions $p(\boldsymbol{z})$ and $q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})$ are normal.

Finally, we proceed to optimize all the variational parameters $\boldsymbol{\lambda}$ jointly by minimizing the augmented ELBO $\mathcal{L}(\boldsymbol{\lambda})$ from Eq. (5). The variational parameters $\boldsymbol{\lambda}_z$ are initialized to the optimized values of the previous ELBO optimization in Eq. (8), whereas the variational parameters $\tilde{\boldsymbol{\mu}}_\phi$ are initialized to the point-estimate $\boldsymbol{\phi}^\star$ computed in the 2nd stage, and $\tilde{\boldsymbol{\sigma}}_\phi$ can be initialized randomly.[4]. Algorithm 1 summarizes the three-step framework for tractable inference.

Algorithm 1 Inference for Proj-BNN

Input: observations $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$

1. Gather multiple sets of weights $\{{\mathbf{w_c}}^{(r)}\}_{r=1}^R$ by training an ensemble of R neural networks over random restarts.
2. Train a prediction-constrained autoencoder $h_{\boldsymbol{\theta},\boldsymbol{\phi}}$ using $\{{\mathbf{w_c}}^{(r)}\}_{r=1}^R$ as input data to minimize the loss function described in (6).
3. Perform BBVI in latent space to learn an approximate posterior distribution over latent representations $\boldsymbol{z}$ and projection parameters $\boldsymbol{\phi}$.
   - Optimize ELBO $\mathcal{L}(\boldsymbol{\lambda}_z)$ in Eq. (8) to obtain $q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})$ closest to $p(\boldsymbol{z}|\mathbf{y}, \mathbf{x}, \boldsymbol{\phi})$ in terms of KL-divergence.
   - Optimize ELBO $\mathcal{L}(\boldsymbol{\lambda})$ in Eq. (5) to obtain $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})$ closest to $p(\boldsymbol{z}, \boldsymbol{\phi}|\mathbf{y}, \mathbf{x})$ in terms of KL-divergence.

Result: Approximate posterior $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})$

5 Results

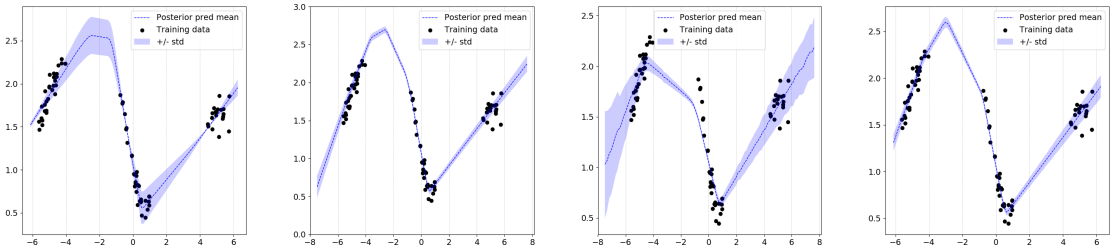
This section contains results on synthetic and real-world datasets to illustrate the performance and potentials of the proposed approach (Proj-BNN). We show that Proj-BNN provides flexible posterior approximations, resulting in better uncertainty estimations, and improved model generalization in terms of held-out test log likelihood across multiple datasets. We will compare Proj-BNN to the following baselines: Bayes by Back Prop (BBB) [Blundell et al., 2015], Multiplicative Normalizing Flow (MNF) [Louizos and Welling, 2017], and Matrix Variate Gaussian Posteriors (MVG) [Louizos and Welling, 2016].

5.1 Synthetic Data

First, we demonstrate on synthetic data that, by performing approximate inference on $\boldsymbol{z}$, we can better capture uncertainty in the posterior predictive. Furthermore, we show that when the space of plausible weights for a regression model is complex, capturing this geometry in approximate inference is difficult; in contrast, performing inference in a simpler latent space allows for better exploration of the solution set in the weight space of the regression model.

Inference in latent space can provide better estimates of posterior predictive uncertainty. In Figure 1, we compare the posterior predictive distributions obtained by Proj-BNN against BBB, MNF, and MVG. The data is generated by sampling points non-uniformly from a

[4]In practice, initializing the standard deviation to small values gave better performance, i.e., $\log \tilde{\boldsymbol{\sigma}}_\phi \sim \mathcal{N}(-9, 0.1)$.



(a) Proj-BNN ($D_z = 2$) (b) BBB (c) MNF (d) MVG

Figure 1: Inferred predictive posterior distribution for a toy data set drawn from a NN with 1-hidden layer, 20 hidden nodes and RBF activation functions. **Proj-BNN is able to learn a plausible predictive mean and better capture predictive uncertainties.**

function represented by a feedforward network (with 1 hidden layer, 20 hidden nodes and RBF activation centered at 0 with length scale 1) whose weights are obtained by applying a fixed linear transform to a fixed latent vector $\boldsymbol{z}$. We observe that our method is able to obtain a mean posterior predictive that fits the data and is furthermore able to capture more uncertainty in the posterior predictive. Notably, benchmark methods tend to underestimate predictive uncertainty, especially in places with few observations, and thus produces over-confident predictions.

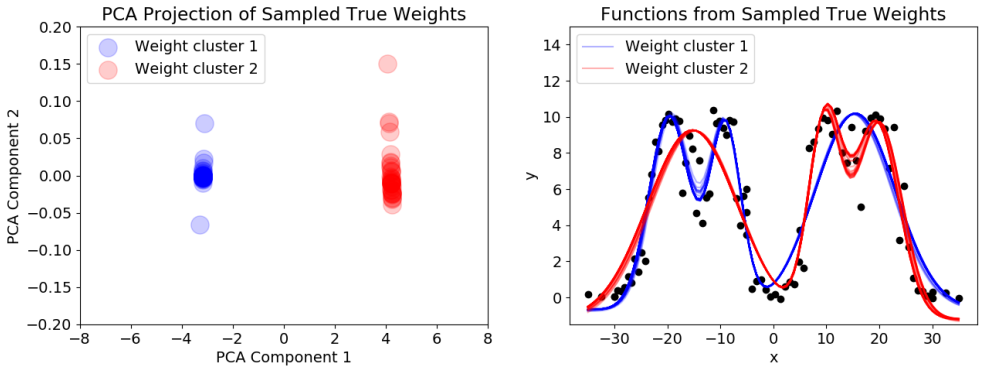
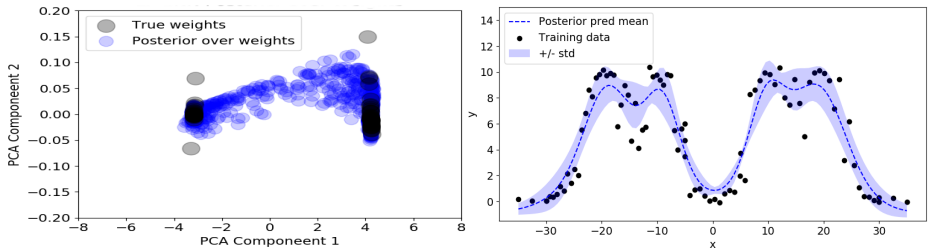


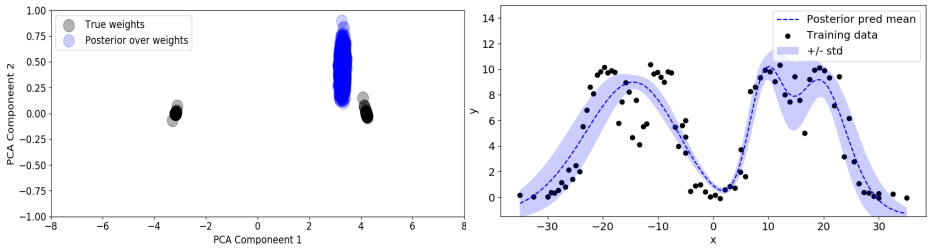
Figure 2: Toy data for regression is generated from a function with four modes. (left) visualizes the weights of the feedforward network, with a single hidden layer with three nodes, obtained by fitting the data multiple times from random weight initializations. The distribution over "good" weights is bimodal. (right) shows examples of functions corresponding to weights sampled from each weight cluster. Each cluster corresponds to fitting a different set of three of the four modes in the data.

Inference in latent space can improve posterior predictive quality by capturing complex geometries of the weight posterior. We argue that the reason for the observed improvement in the quality of posterior predictive obtained by our model is often due to the fact that it is difficult to approximate complex geometry of the solution set in the weight space of sophisticated regression models. Mapping the solution set onto a simpler region in a lower dimensional latent space allows for more efficient approximations. In Figure 2, we visualize samples of plausible weights for a feedforward network (with a single hidden layer, three nodes and RBF activation function) fitted to a data set with four modes. We see that the solution set in the weight space for this model is naturally bimodal,



(a) Proj-BNN posterior over weights

(b) Proj-BNN posterior predictive



(c) BNN posterior over weights (BBB)

(d) BNN posterior predictive (BBB)

Figure 3: (a) shows the variational posterior over weights, $\boldsymbol{w}$, obtained by transforming the variational posterior over $\boldsymbol{z}$. Learning a variational posterior over $\boldsymbol{z}$ captures both modes in the weight space. (c) shows the variational posterior over weights learned by performing inference directly on $\boldsymbol{w}$, using Bayes by Back Prop (BBB). This posterior captures only one mode in the weight space. (b) shows the posterior predictive corresponding to the variational posterior over $\boldsymbol{z}$. The mean of the posterior predictive demonstrates four modes in the data. (d) shows the posterior predictive corresponding to the variational posterior over $\boldsymbol{w}$ using BBB. The mean of the posterior predictive demonstrates only three modes.

where each mode corresponds to functions that fit a particular choice of three of the four modes in the data.

Figure 3 shows that direct variational approximation of the posterior over weights is only able to capture one of the modes in the solution set, while approximating the posterior over $\boldsymbol{z}$'s using our model captures both modes. For the latter, we trained a decoder $g_{\boldsymbol{\phi}}(\cdot)$ that maps an isotropic 2-D Gaussian to weights that we sampled from the solution set. As a result, the posterior predictive mean for our model is able to approximate the four modes in the data, while the predictive mean obtained from directly approximating the posterior over $\boldsymbol{w}$ can approximate only three of the four modes.

5.2 Real Data

We perform nonlinear regression on eight UCI datasets, listed in Figure 4. In all the following experiments, we use a random train-test-validation split of 80-10-10. All datasets are normalized in a preprocessing step to have zero mean and unit standard deviation. For each data set, we first sample $R = 500$ candidate weight solutions $\boldsymbol{w_c}$ by fitting an ensemble of neural networks on the training set. For each optimization subtask listed in Algorithm 1, we perform cross-validation of the step size $\lambda_1 \in \{0.1, 0.01, 0.001, 0.0001\}$ and batch size $B_1 \in \{16, 128, 512\}$. We use Adam [Kingma and Ba, 2014] as the optimizer and the joint unnormalized posterior distribution $p(\boldsymbol{w}|\mathcal{D}_{train})$ as the objective function. Optimization is performed with $L = 50,000$ iterations and early stopping once the log likelihood in validation set stop increasing.

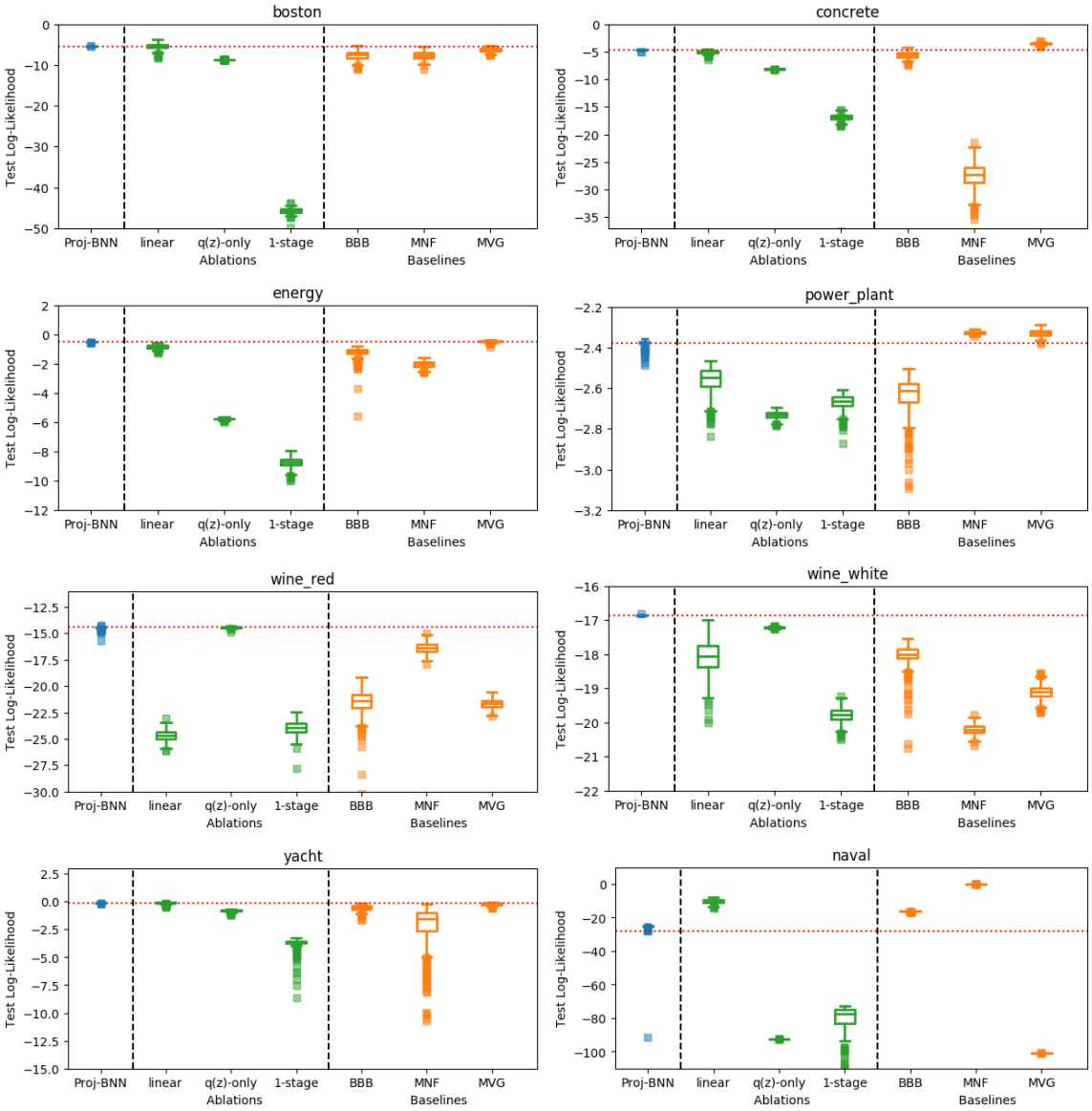


Figure 4: Test log-likelihood for UCI benchmark datasets for best dimensionality of z-space (see Figure 5 for performance across different dimensionality of the latent space D_z). Red dotted horizontal line corresponds to Proj-BNN performance (our approach). Baselines methods are: 1) BBB: mean field (Blundell, et.al 2015); 2) MNF: multiplicative normalizing flow (Louizos et.al, 2017); 3) MVG: multivariate Gaussian prior BNN (Louizos et.al, 2016). Ablations of Proj-BNN are: Proj-BNN with linear projections (linear), Proj-BNN without training the autoencoder, i.e., only stage 3 in Alg. 1 (1-stage), Proj-BNN modeling uncertainty only in $\boldsymbol{z}$ ($q(\boldsymbol{z})$-only). **In all but two cases Proj-BNN performs better or as well as the benchmarks.**

Inference in latent space can improve model generalization. On datasets where ground truth distributions are not available for comparison and the inferred distributions are not easily visualized, we argue that the higher quality posterior and posterior predictive potentially obtained from Proj-BNN can be observed through an improvement in the ability of our model to generalize.

We compare the generalization performance, measured in terms of test likelihood, of Proj-BNN with the three benchmark models, BBB, MNF and MVG. In Figure 4, we see that Proj-BNN performs competitively, if not better than benchmark models, on all but one dataset.

6 Discussion

The geometry of weight posteriors impacts the quality of variational approximations. Experimental results on the synthetic data demonstrates the advantage of Proj-BNN in cases where the geometry of the true posterior over weights is complex, e.g. multimodal or highly non-convex. Here, traditional mean field variational inference will tend to capture only a small part of the true posterior, e.g. a single mode or a small convex region, due to the zero-forcing nature of KL-divergence minimization. Furthermore, in these cases, we find that optimization for variational inference tend to be easily trapped in undesirable local optima. In contrast, provided with a robust non-linear projection onto a low dimensional latent space, we are able to drastically reduce the complexity of the optimization problem, e.g. by reducing the number parameters to be optimized. As a result, the posterior predictive distributions obtained by Proj-BNN often capture more uncertainty, whereas comparable methods that perform inference directly on weights tend to underestimate uncertainty and, thus, can produce over-confident predictions.

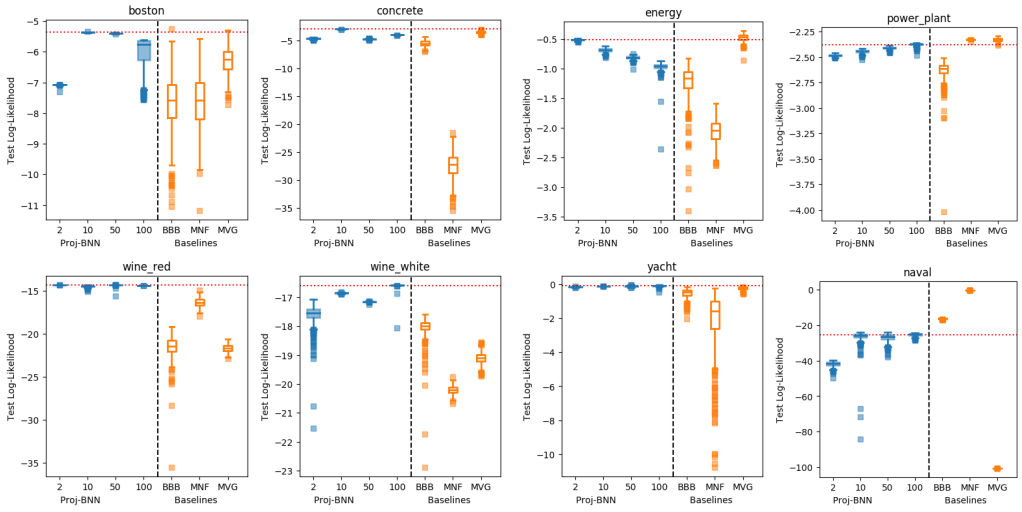


Figure 5: Test loglikelihood for UCI benchmark datasets for varying dimensionality of $\boldsymbol{z}$-space. Red dotted horizontal line corresponds to the best performance of Proj-BNN (our approach). Baselines: a) BBB: mean field (Blundell, et.al 2015); b) MNF: multiplicative normalizing flow (Louizos et.al, 2017); c) MVG: multivariate Gaussian prior BNN (Louizos et.al, 2016).

On real datasets, Proj-BNN out-performs or remain competitive, in terms of model generalization, with comparable benchmark methods working with latent representations that are significantly lower in dimension than network weights. This is again evidence that performing inference in lower dimensional latent space can better capture complex posterior distributions in weight space. In cases where Proj-BNN under-performs in comparison to benchmark methods, we conjecture that the shortcoming may be due to insufficient sampling of the weight space during the first stage of ensemble training.

The quality of the non-linear latent projection impacts the quality of variational inference. The quality of the variational approximations obtained by Proj-BNN relies on two conditions

1) the ability to characterize the set of plausible neural network weights given a data set and 2) the ability to learn informative transformations between latent space and weight space. The former requires us to sample intelligently from the weight space. We currently use an ensemble of (non-Bayesian) neural networks to obtain weights samples, while this is a one-time cost per data set, we nonetheless observe that this process can be computationally expensive. We see opportunities here to improve the sample-efficiency as well as to incorporate training objectives that explicitly encourage diversity of the samples we obtain. For condition 2), we note that learning transformations that are able to reconstruct weights from their latent representations is not necessarily helpful for inference, as the reconstructed weights might encode for models that suffer from a drastic decrease in predictive accuracy. We address this problem by adding a predictive constraint while learning these transformations. We note that there is opportunity in this step to incorporate additional constraints that may improve inference in the latent space.

7 Conclusion

In this paper, we have presented a framework, Proj-BNN, for performing approximate inference for Bayesian Neural Networks that can avoid many of the optimization problems of traditional inference methods. In particular, we are able to better capture the geometry of the posterior over weights by learning a probability distribution over non-linear transformations of the weight space onto a simpler latent space and perform mean-field variational inference in the latent space.

Using synthetic data sets, we show that, in cases where the posterior over weights exhibits complex geometry (e.g. is multimodal), variational inference performed on *weights* space will often become trapped in local optima, whereas variational distributions in *latent* space can more easily capture the shape of the true posterior through a non-linear transformation. We compare Proj-BNN with three relevant benchmarks, each of which is an enrichment of the standard variational approach for BNN inference, and show that Proj-BNN is able to better capture posterior predictive uncertainty (without compromising predictive accuracy) on synthetic data. On 8 real datasets, we show that, in terms of test likelihood, Proj-BNN is able to perform competitively if not better than the three benchmark methods, working in latent dimensions that are much smaller than the dimensions of the weight space.

We note that further gains in the quality and efficiency of inference can be attained by improving each phase of our framework. That is, one can implement a number of sample-efficient methods for learning the solution set of the regression model in weight space [Zhang et al., 2015]; one can regularize the latent representation learned by the autoencoder to have desirable geometries [Hosseini-Asl et al., 2016]; one can place complex priors on the latent space, such as nonparametric variational inference [Gershman et al., 2012].

References

Felix V Agakov and David Barber. An auxiliary variational method. In *International Conference on Neural Information Processing*, pages 561–566, 2004.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Wray L Buntine and Andreas S Weigend. Bayesian back-propagation. *Complex systems*, 5(6):603–643, 1991.

Misha Denil, Babak Shakibi, Laurent Dinh, Marc Aurelio Ranzato, and Nando de Freitas. Predicting Parameters in Deep Learning. June 2013. arXiv: 1306.0543.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Samuel Gershman, Matt Hoffman, and David Blei. Nonparametric variational inference. *arXiv preprint arXiv:1206.4665*, 2012.

Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang Bui, and Richard Eric Turner. Black-box α-divergence minimization. 2016.

Ehsan Hosseini-Asl, Jacek M Zurada, and Olfa Nasraoui. Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints. *IEEE transactions on neural networks and learning systems*, 27(12):2486–2498, 2016.

Michael C Hughes, Gabriel Hope, Leah Weiner, Thomas H McCoy, Roy H Perlis, Erik B Sudderth, and Finale Doshi-Velez. Prediction-constrained topic models for antidepressant recommendation. *arXiv preprint arXiv:1712.00499*, 2017.

Michael C Hughes, Gabriel Hope, Leah Weiner, Thomas H McCoy Jr, Roy H Perlis, Erik B Sudderth, and Finale Doshi-Velez. Semi-supervised prediction-constrained topic models. In *AISTATS*, pages 1067–1076, 2018.

Theofanis Karaletsos, Peter Dayan, and Zoubin Ghahramani. Probabilistic Meta-Representations Of Neural Networks. October 2018. arXiv: 1810.00555.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. December 2013. arXiv: 1312.6114.

Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716, 2016.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. *arXiv preprint arXiv:1703.01961*, 2017.

Xiuyuan Lu and Benjamin Van Roy. Ensemble Sampling. *arXiv:1705.07347 [cs, stat]*, May 2017. URL `http://arxiv.org/abs/1705.07347`. arXiv: 1705.07347.

Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

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

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992b.

Preetum Nakkiran, Raziel Alvarez, Rohit Prabhavalkar, and Carolina Parada. Compressing Deep Neural Networks using a Rank-Constrained Topology, 2015.

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

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Nick Pawlowski, Martin Rajchl, and Ben Glocker. Implicit weight uncertainty in neural networks. *arXiv preprint arXiv:1711.01297*, 2017.

Tim Pearce, Mohamed Zaki, Alexandra Brintrup, and Andy Neel. Uncertainty in neural networks: Bayesian ensembling. *arXiv preprint arXiv:1810.05546*, 2018.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. January 2014. arXiv: 1401.4082.

Tara N. Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. Low-rank matrix factorization for Deep Neural Network training with high-dimensional output targets. pages 6655–6659. IEEE, May 2013.

Tim Salimans, David A Knowles, et al. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Shengyang Sun, Changyou Chen, and Lawrence Carin. Learning structured weight uncertainty in bayesian neural networks. In *Artificial Intelligence and Statistics*, pages 1283–1292, 2017.

Jian Xue, Jinyu Li, and Yifan Gong. Restructuring of deep neural network acoustic models with singular value decomposition. In *Interspeech*, pages 2365–2369, 2013.

Xiaohui Zhang, Daniel Povey, and Sanjeev Khudanpur. A diversity-penalizing ensemble training method for deep learning. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.