Fast Predictive Uncertainty for Classification with Bayesian Deep Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

In Bayesian Deep Learning, distributions over the output of classification neural networks are approximated by first constructing a Gaussian distribution over the weights, then sampling from it to receive a distribution over the categorical output distribution. This is costly. We reconsider old work to construct a Dirichlet approximation of this output distribution, which yields an analytic map between Gaussian distributions in logit space and Dirichlet distributions (the conjugate prior to the categorical) in the output space. We argue that the resulting Dirichlet distribution has theoretical and practical advantages, in particular more efficient computation of the uncertainty estimate, scaling to large datasets and networks like ImageNet and DenseNet. We demonstrate the use of this Dirichlet approximation by using it to construct a lightweight uncertainty-aware output ranking for the ImageNet setup.

1 Introduction

Quantifying the uncertainty of neural networks' (NNs) predictions is important in safety-critical applications such as medical-diagnosis [1] and self-driving vehicles [2; 3]. Architectures for classification tasks produce a probability distribution as their output, constructed by applying the softmax to the point-estimate output of the penultimate layer. However, it has been shown that this distribution is overconfident [4; 5] and thus cannot be used for predictive uncertainty quantification.

Approximate Bayesian methods provide quantified uncertainty over the network's parameters and thus the outputs in a tractable fashion. The commonly used Gaussian approximate posterior [6; 7; 8; 9] approximately induces a Gaussian distribution over the logits of a NN [10]. However, the associated predictive distribution, which is the expectation of the softmax function w.r.t. the Gaussian does not have an analytic form. It is thus generally approximated by Monte Carlo (MC) integration requiring multiple samples. Predictions in Bayesian neural networks (BNNs) are thus generally expensive operations.

In this paper, we re-introduce an old but largely overlooked idea originally proposed by David JC MacKay [11] in a different setting (arguably the inverse of the Deep Learning setting). Dirichlet distributions are generally defined on the simplex. But when its variable is defined on the inverse softmax's domain, its shape effectively approximates a Gaussian. The inverse of this approximation, which will be called the *Laplace Bridge* here [12], analytically maps a Gaussian distribution onto a Dirichlet distribution. Given a Gaussian distribution over the logits of a NN, one can thus efficiently obtain an approximate Dirichlet distribution over the softmax outputs (Figure 1). Our contributions in this paper are: We re-visit MacKay's derivation with particular attention to a symmetry constraint that becomes necessary in our "inverted" use of the argument from the Gaussian to the Dirichlet family. We then validate the quality of this approximation both by theoretical and empirical arguments and demonstrate significant speed-up over MC-integration. Finally, we show a use-case, leveraging the

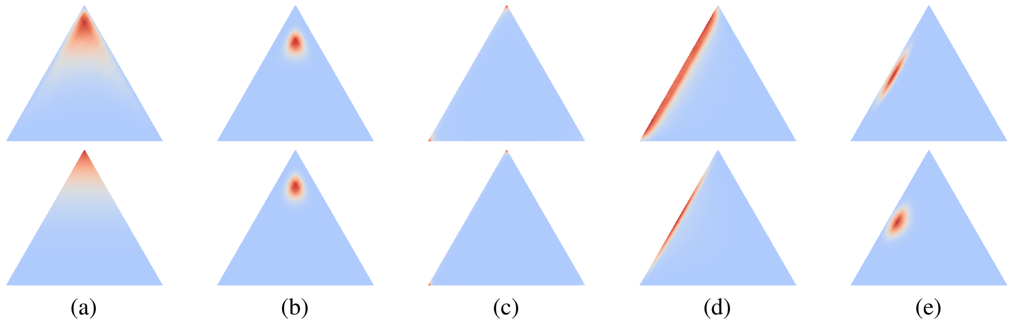

Figure 1: Densities on the simplex of the true distribution (top row, computed by MC integration) and "Laplace Bridge" approximation constructed in this paper (bottom row). For column (a) and (b), two different Gaussians were constructed, such that the resulting MAP estimate is the same, but the uncertainty differs. For (c), (d) and (e) the same mean with decreasing uncertainty was used. We find that in all cases the Laplace Bridge is a good approximation and captures the desired properties.

analytic properties of Dirichlet distributions to improve the popular top-k metric through uncertainties. We want to emphasize that the Laplace Bridge can be applied to any Gaussian over the outputs independent of the way it was generated; it is not restricted to a Laplace approximation of the network.

Section 2 provides the mathematical derivation. Section 3 and 3.1 discuss the Laplace Bridge in the context of neural networks and with a deeper analysis of different ways to do posterior inference. We compare it to the recent approximations of the predictive distributions of NNs in Section 4. Experiments are presented in Section 5.

2 The Laplace Bridge

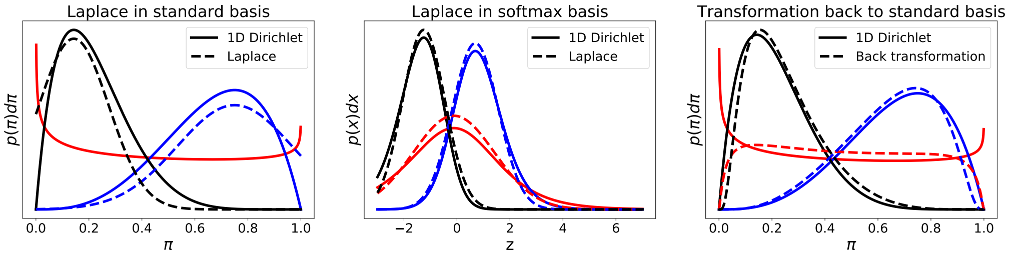

Figure 2: (Adapted from Hennig et al. [12]). Visualization of the Laplace Bridge for the Beta distribution (1D special case of the Dirichlet). **Left:** "Generic" Laplace approximations of standard Beta distributions by Gaussians. Note that the Beta Distribution (red curve) does not even have a valid approximation because the Hessian is not positive semi-definite. **Middle:** Laplace approximation to the same distributions after basis transformation through the softmax (4). The transformation makes the distributions "more Gaussian" (i.e. uni-modal, bell-shaped, with support on the real line) compared to the standard basis, thus making the Laplace approximation more accurate. **Right:** The same Beta distributions, with the back-transformation of the Laplace approximations from the middle figure to the simplex, yielding a much improved approximate distribution. In particular, in contrast to the left-most image, the dashed lines now actually are probability distributions (they integrate to 1 on the simplex).

Laplace approximations[1] are a popular and light-weight method to approximate a general probability distribution $q(\mathbf{x})$ with a Gaussian $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. It sets $\boldsymbol{\mu}$ to a mode of q, and $\boldsymbol{\Sigma} = -(\nabla^2 \log q(\mathbf{x})|_{\boldsymbol{\mu}})^{-1}$, the inverse Hessian of $\log q$ at that mode. This scheme can work well if the true distribution is unimodal and defined on the real vector space.

[1] For clarity: Laplace approximations are *also* one out of several possible ways to construct a Gaussian approximation to the weight posterior of a neural network, by constructing a second-order Taylor approximation of the empirical risk at the trained weights. This is *not* the way they are used in this section. The Laplace Bridge is agnostic to how the input Gaussian distribution is constructed. It could, e.g., also be constructed as a variational approximation, or the moments of Monte Carlo samples. See also Section 3.1.

The Dirichlet distribution, which has the density function

$$\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) := \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}\,, \tag{1}$$

is defined on the probability simplex and can be multimodal in the sense that the maxima of the distribution lie at the boundary of the simplex when $\alpha_k < 1$, for all $k = 1, \dots, K$. Both issues preclude a Laplace approximation, at least in the naïve form described above. However, MacKay [11] noted that both can be fixed, elegantly, by a change of variable. Details of the following argument can be found in the supplements. Consider the K-dimensional variable $\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$ defined as the softmax of $\mathbf{z} \in \mathbb{R}^K$:

$$\pi_k(\mathbf{z}) := \frac{\exp(z_k)}{\sum_{l=1}^{K} \exp(z_l)}\,, \tag{2}$$

for all $k = 1, \dots, K$. We will call $\mathbf{z}$ the logit of $\boldsymbol{\pi}$. When expressed as a function of $\mathbf{z}$, the density of the Dirichlet in $\boldsymbol{\pi}$ has to be multiplied by the Jacobian determinant

$$\det \frac{\partial \boldsymbol{\pi}}{\partial \mathbf{z}} = \prod_k \pi_k(z), \tag{3}$$

thus removing the -1 terms in the exponent:

$$\mathrm{Dir}_{\mathbf{z}}(\boldsymbol{\pi}(\mathbf{z})|\boldsymbol{\alpha}) := \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k(\mathbf{z})^{\alpha_k}\,, \tag{4}$$

This density of $\mathbf{z}$ (!), the Dirichlet distribution in the *softmax basis*, can now be accurately approximated by a Gaussian through a Laplace approximation, yielding an analytic map from the parameter space $\boldsymbol{\alpha} \in \mathbb{R}_+^K$ to the parameter space of the Gaussian ($\boldsymbol{\mu} \in \mathbb{R}^K$ and symmetric positive definite $\boldsymbol{\Sigma} \in \mathbb{R}^{K \times K}$), given by

$$\mu_k = \log \alpha_k - \frac{1}{K} \sum_{l=1}^{K} \log \alpha_l \tag{5}$$

$$\Sigma_{k\ell} = \delta_{k\ell} \frac{1}{\alpha_k} - \frac{1}{K} \left[\frac{1}{\alpha_k} + \frac{1}{\alpha_\ell} - \frac{1}{K} \sum_{u=1}^{K} \frac{1}{\alpha_u} \right]. \tag{6}$$

The corresponding derivations require care because the Gaussian parameter space is evidently larger than that of the Dirichlet and not fully identified by the transformation. A pseudo-inverse of this map was provided by Hennig et al. [12]. It maps the Gaussian parameters to those of the Dirichlet as

$$\alpha_k = \frac{1}{\Sigma_{kk}} \left(1 - \frac{2}{K} + \frac{e^{\mu_k}}{K^2} \sum_{l=1}^{K} e^{-\mu_l} \right) \tag{7}$$

(Note that this equation ignores off-diagonal elements of $\boldsymbol{\Sigma}$, more discussion below). Together, Eqs. 5, 6 and 7 will here be used for Bayesian Deep Learning, and jointly called the *Laplace Bridge*. Note that, even though the Laplace Bridge implies a reduction of the expressiveness of the distribution, we show in Section 3 that this map is still sufficiently accurate.

Figure 1 shows the quality of the resulting approximation. We consider multiple different $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ in three dimensions. We exhaustively sample from the Gaussian and apply the softmax. The resulting histogram is compared to the PDF of the corresponding Dirichlet. The first part of the figure emphasizes that a point estimate is insufficient. Since the mean for the Dirichlet is the normalized $\boldsymbol{\alpha}$ parameter vector, the parameters ($\boldsymbol{\alpha}_1 = [2, 2, 6]^\top$ and $\boldsymbol{\alpha}_2 = [11, 11, 51]^\top$) yield the same point estimate even though their distributions are clearly different. The second part shows how the Laplace Bridge maps w.r.t decreasing uncertainty.

3 The Laplace Bridge for BNNs

Let $f_{\boldsymbol{\theta}} : \mathbb{R}^N \to \mathbb{R}^K$ be an L-layer neural network parametrized by $\boldsymbol{\theta} \in \mathbb{R}^P$, with a Gaussian approximate posterior $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$. For any input $\mathbf{x} \in \mathbb{R}^N$, one way to obtain an approximate Gaussian distribution on the pre-softmax output (logit vector) $f_{\boldsymbol{\theta}}(\mathbf{x}) =: \mathbf{z}$ is as

$$q(\mathbf{z}|\mathbf{x}) \approx \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\boldsymbol{\theta}}^\top \mathbf{x}, \mathbf{J}(\mathbf{x})^\top \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \mathbf{J}(\mathbf{x}))\,, \tag{8}$$

where $\mathbf{J}(\mathbf{x})$ is the $P \times K$ Jacobian matrix representing the derivative $\frac{\partial \mathbf{z}}{\partial \boldsymbol{\theta}}$ [10]. Approximating the density of the softmax of this Gaussian random variable as a Dirichlet, using the Laplace Bridge, *analytically* approximates the predictive distribution in a single step, as opposed to many samples. From Eq. (7), this requires $\mathcal{O}(K)$ computations to construct the K parameters α_k of the Dirichlet. In contrast, MC-integration has computational costs of $\mathcal{O}(MJ)$, where M is the number of samples and J is the cost of sampling from $q(\mathbf{z}|\mathbf{x})$ (typically J is of order K^2 after an initial $\mathcal{O}(K^3)$ operation for a matrix decomposition of the covariance). The Monte Carlo approximation has the usual sampling error of $\mathcal{O}(1/\sqrt{M})$, while the Laplace Bridge has a fixed but small error (empirical comparison in Section 5.3).

We now discuss several qualitative properties of the Laplace Bridge relevant for the uncertainty quantification use case in Deep Learning. Some benefits of this approximation arise from the convenient analytical properties of the Dirichlet exponential family. For example, a point estimate of the posterior predictive distribution is directly given by the Dirichlet's mean,

$$\mathbb{E}\boldsymbol{\pi} = \left(\frac{\alpha_1}{\sum_{l=1}^{K} \alpha_l}, \dots, \frac{\alpha_K}{\sum_{l=1}^{K} \alpha_l} \right)^\top , \tag{9}$$

Further, Dirichlets have Dirichlet marginals: If $p(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$, then

$$p([\pi_1, \pi_2, \dots, \pi_j, \sum_{k>j} \pi_k]^\top) = \mathrm{Dir}(\alpha_1, \alpha_2, \dots, \alpha_j, \sum_{k>j} \alpha_k) . \tag{10}$$

An additional benefit of the Laplace Bridge for BNNs is that it is more flexible than an MC-integral. If we let $p(\boldsymbol{\pi})$ be the distribution over $\boldsymbol{\pi} := \mathrm{softmax}(\mathbf{z}) := [e^{z_1}/\sum_l e^{z_l}, \dots, e^{z_K}/\sum_l e^{z_l}]^\top$, then the MC-integral can be seen as a "point-estimate" of this distribution since it approximates $\mathbb{E}\boldsymbol{\pi}$. In contrast, the Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$ approximates the distribution $p(\boldsymbol{\pi})$. Thus, the Laplace Bridge enables tasks that can be done only with a distribution but not a point estimate. For instance, one could ask "what is the distribution of the first L classes?" when one is dealing with K-class ($L < K$) classification. Since the marginal distribution can be computed analytically (10), the Laplace Bridge provides a convenient yet cheap way of answering this question. A theoretical statement on the behavior of Laplace Bridge w.r.t its variance can be found in the supplements.

3.1 Posterior inference

In principle, the Gaussian over the weights required by the Laplace Bridge for BNNs (see Equation 8) can be constructed by any Gaussian approximate Bayesian methods such as variational Bayes [7; 8] and Laplace approximations for neural networks [6; 9]. We will focus on the Laplace approximation, which uses the same principle as the Laplace Bridge. However, in the Laplace approximation for neural networks, the posterior distribution over the weights of a network is the one that is approximated as a Gaussian, instead of a Dirichlet distribution over the outputs as in the Laplace Bridge.

Given a dataset $\mathcal{D} := \{(\mathbf{x}_i, t_i)\}_{i=1}^{D}$ and a prior $p(\boldsymbol{\theta})$, let

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta}) p(\mathcal{D}|\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{(\mathbf{x},t)\in\mathcal{D}} p(y = t|\boldsymbol{\theta}, \mathbf{x}) , \tag{11}$$

be the posterior over the parameter $\boldsymbol{\theta}$ of an L-layer network $f_{\boldsymbol{\theta}}$. Then we can get an approximation of the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ by fitting a Gaussian $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$ where

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{\theta}} &= \boldsymbol{\theta}_{\text{MAP}} , \\ \boldsymbol{\Sigma}_{\boldsymbol{\theta}} &= (-\nabla^2|_{\boldsymbol{\theta}_{\text{MAP}}} \log p(\boldsymbol{\theta}|\mathcal{D}))^{-1} =: \mathbf{H}_{\boldsymbol{\theta}}^{-1} . \end{aligned}$$

That is, we fit a Gaussian centered at the mode $\boldsymbol{\theta}_{\text{MAP}}$ of $p(\boldsymbol{\theta}|\mathcal{D})$ with the covariance determined by the curvature at that point. We assume that the prior $p(\boldsymbol{\theta})$ is a zero-mean isotropic Gaussian $\mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \sigma^2\mathbf{I})$ and the likelihood function is the Categorical density

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{(\mathbf{x},t)\in\mathcal{D}} \mathrm{Cat}(y = t|\mathrm{softmax}(f_{\boldsymbol{\theta}}(\mathbf{x}))) .$$

For various applications in Deep Learning, the approximation in (8) is often computationally too expensive. Indeed, for each input $\mathbf{x} \in \mathbb{R}^N$, one has to do K backward passes to compute the Jacobian

$\mathbf{J}(\mathbf{x})$. Moreover, it requires an $\mathcal{O}(PK)$ storage which is also expensive since P is often in the order of millions. A cheaper alternative is to fix all but the last layer of $f_{\boldsymbol{\theta}}$ and only apply the Laplace approximation on $\mathbf{W}_L$, the last layer's weight matrix. This scheme has been used successfully by Snoek et al. [13]; Wilson et al. [14], etc. and has been shown empirically to be effective in uncertainty quantification tasks [15]. In this case, given the approximate last-layer posterior

$$p(\mathbf{W}^L|\mathcal{D}) \approx \mathcal{N}(\text{vec}(\mathbf{W}^L)|\text{vec}(\mathbf{W}^L_{\text{MAP}}), \mathbf{H}^{-1}_{\mathbf{W}^L}) \,, \tag{12}$$

one can efficiently compute the distribution over the logits. That is, let $\boldsymbol{\phi} : \mathbb{R}^N \to \mathbb{R}^Q$ be the first $L-1$ layers of $f_{\boldsymbol{\theta}}$, seen as a feature map. Then, for each $\mathbf{x} \in \mathbb{R}^N$, the induced distribution over the logit $\mathbf{W}^L \boldsymbol{\phi}(\mathbf{x}) =: \mathbf{z}$ is given by

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{W}^L_{\text{MAP}}\boldsymbol{\phi}(\mathbf{x}), (\boldsymbol{\phi}(\mathbf{x})^\top \otimes \mathbf{I})\mathbf{H}^{-1}_{\mathbf{W}^L}(\boldsymbol{\phi}(\mathbf{x}) \otimes \mathbf{I})) \,, \tag{13}$$

where $\otimes$ denotes the Kronecker product.

An even more efficient last-layer approximation can be obtained using a Kronecker-factored matrix normal distribution [16; 17; 9]. That is, we assume the posterior distribution to be

$$p(\mathbf{W}^L|\mathcal{D}) \approx \mathcal{MN}(\mathbf{W}^L|\mathbf{W}^L_{\text{MAP}}, \mathbf{U}, \mathbf{V}) \,, \tag{14}$$

where $\mathbf{U} \in \mathbb{R}^{K \times K}$ and $\mathbf{V} \in \mathbb{R}^{Q \times Q}$ are the Kronecker factorization of the inverse Hessian matrix $\mathbf{H}^{-1}_{\mathbf{W}^L}$ [18]. In this case, for any $\mathbf{x} \in \mathbb{R}^N$, one can easily show that the distribution over logits is given by

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{W}^L_{\text{MAP}}\boldsymbol{\phi}(\mathbf{x}), (\boldsymbol{\phi}(\mathbf{x})^\top \mathbf{V} \boldsymbol{\phi}(\mathbf{x}))\mathbf{U}) \,, \tag{15}$$

which is easy to implement and computationally cheap. Finally, and even more efficient, is a last-layer approximation scheme with a diagonal Gaussian approximate posterior, i.e. the so-called mean-field approximation. In this case, we assume the posterior distribution to be

$$p(\mathbf{W}^L|\mathcal{D}) \approx \mathcal{N}(\text{vec}(\mathbf{W}^L)|\text{vec}(\mathbf{W}^L_{\text{MAP}}), \text{diag}(\boldsymbol{\sigma}^2)) \,, \tag{16}$$

where $\boldsymbol{\sigma}^2$ is obtained via the diagonal of the Hessian of the log-posterior w.r.t. $\text{vec}(\mathbf{W}^L)$ at $\text{vec}(\mathbf{W}^L_{\text{MAP}})$.

4 Related Work

In Bayesian neural networks, analytic approximations of posterior predictive distributions have attracted a great deal of research. In the binary classification case, for example, the probit approximation has been proposed already in the 1990s [19; 20]. However, while there exist some bounds [21] and approximations of the expected log-sum-exponent function [22; 23], in the multi-class case, obtaining a good analytic approximation of the expected softmax function under a Gaussian measure is still considered an open problem. The Laplace Bridge is a close approximation of this integral and can be analytically computed via (9). The Laplace Bridge furthers the trend of sampling-free solutions within Bayesian Deep Learning (e.g. [24] and [25]). Recently, it has been proposed to model the distribution of softmax outputs of a network directly. Similar to the Laplace Bridge, Malinin and Gales [26, 27]; Sensoy et al. [28] proposed to use the Dirichlet distribution to model the posterior predictive for non-Bayesian networks. They further proposed novel training techniques in order to directly learn the Dirichlet. In contrast, the Laplace Bridge tackles the problem of approximating the distribution over the softmax outputs of the ubiquitous Gaussian-approximated Bayesian networks [7; 8; 16; 17, etc] without any additional training procedure. This allows the Laplace Bridge to be used with pre-trained networks.

5 Experiments

We conduct four experiments. In Section 5.1, we analyze the approximation quality of the Laplace Bridge applied to a BNN on the MNIST [29] dataset. Then, we compare the Laplace Bridge to the MC-integral in terms of the out-of-distribution (OOD) detection performance in Section 5.2. Their computational costs are compared in Section 5.3. Finally, in Section 5.4, we present analysis on ImageNet [30] to demonstrate the scalability of the Laplace Bridge and the advantage of having a full Dirichlet distribution over softmax outputs. We chose to use a Laplace approximation of the network because it is the fastest way to get a Gaussian over the outputs and therefore in the spirit of our method. Other methods, such as Variational Inference or Ensembles could also be applied though.

5.1 Uncertainty estimates on MNIST

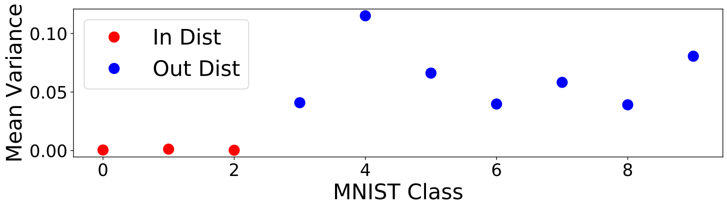

Figure 3: Average variance of the Dirichlet distributions of each MNIST class. The in-distribution uncertainty (variance) is nearly nil, while out-of-distribution variance is higher.

We empirically investigate the approximation quality of the Laplace Bridge in a "real-world" BNN on the MNIST dataset. A CNN with 2 convolutional and 2 fully-connected layers is trained on the first three digits of MNIST (the digits 0, 1, and 2). To obtain the posterior over the weights of this network, we perform a full (all-layer) Laplace approximation using BackPACK [31] to get the diagonal Hessian. The network is then evaluated on the full test set of MNIST (containing all ten classes). We present the results in Figure 3. We show for each $k = 1, \ldots, K$, the average variance $\frac{1}{D_k} \sum_{i=1}^{D_k} \mathrm{Var}(\pi_k(f_{\boldsymbol{\theta}}(\mathbf{x}_i)))$ of the resulting Dirichlet distribution over the softmax outputs, where D_k is the number of test points predicted with label k. The results show that the variance of the Dirichlet distribution obtained via the Laplace Bridge is useful for uncertainty quantification: OOD data can be easily detected since the mean variance of the first three classes is nearly zero while that of the others is higher.

Table 1: OOD detection results. While there is arguable no clear winner when it comes to discriminating in- and out-distribution data w.r.t. both metrics, the Laplace Bridge is around 400 times faster on average. 1000 samples were drawn from the Gaussian over the outputs. The (F-, K-, not-)MNIST experiments were done with a Laplace approximation of the entire network while the others only used the last layer.

		Diag Sampling			KFAC Sampling			Dirichlet mode		
Train	**Test**	**MMC**	**AUROC**	**Time**	**MMC**	**AUROC**	**Time**	**MMC**	**AUROC**	**Time**
MNIST	MNIST	0.932 ± 0.007	-	6.6	-	-	-	**0.987** ± 0.001	-	**0.016**
MNIST	FMNIST	0.407 ± 0.010	0.989 ± 0.002	6.6	-	-	-	**0.377** ± 0.019	**0.994** ± 0.002	**0.016**
MNIST	notMNIST	**0.535** ± 0.018	0.958 ± 0.006	12.3	-	-	-	0.630 ± 0.018	**0.962** ± 0.007	**0.029**
MNIST	KMNIST	**0.500** ± 0.014	0.974 ± 0.005	6.6	-	-	-	0.630 ± 0.018	**0.975** ± 0.004	**0.016**
CIFAR-10	CIFAR-10	0.948	-	13.6	0.857 ± 0.003	-	13.4	**0.966**	-	**0.031**
CIFAR-10	CIFAR-100	0.708	**0.889**	13.6	**0.562** ±0.003	0.880 ± 0.012	13.5	0.742	0.866	**0.027**
CIFAR-10	SVHN	0.643	0.933	35.2	**0.484** ±0.004	**0.939** ±0.001	35.2	0.647	0.934	**0.070**
SVHN	SVHN	0.986	-	34.5	0.947 ± 0.002	-	34.6	**0.993**	-	**0.073**
SVHN	CIFAR-100	0.595	0.984	13.3	**0.460** ±0.004	0.986 ± 0.001	13.4	0.526	0.985	**0.027**
SVHN	CIFAR-10	0.593	0.984	13.3	**0.458** ±0.004	0.986 ± 0.001	13.3	0.520	**0.987**	**0.028**
CIFAR-100	CIFAR-100	**0.762**	-	24.5	0.404	-	24.6	0.590	-	**0.030**
CIFAR-100	CIFAR-10	0.467	0.788	24.4	0.213	0.788	24.6	**0.206**	**0.791**	**0.027**
CIFAR-100	SVHN	0.461	0.795	63.4	0.180 ± 0.001	**0.838** ±0.001	63.8	**0.170**	0.815	**0.069**

5.2 OOD detection

We compare the performance of the Laplace Bridge to the MC-integral on a standard OOD detection benchmark suite, to test whether the Laplace Bridge gives similar results to the MC sampling method and compare their computational overhead. Following prior literature, we use the standard mean-maximum-confidence (MMC) and area under the ROC-curve (AUROC) metrics [32]. For an in-distribution dataset, a higher MMC value is desirable while for the OOD dataset we want a lower MMC value (optimally, $1/K$ in K-class classification problems). For the AUROC metric, the higher the better, since it represents how good a method is for distinguishing in- and out-of-distribution datasets.

The test scenarios are as follows: (i) The same convolutional network as in Section 5.1 is trained on the MNIST dataset. To approximate the posterior over the parameter of this network, a full (all-layer) Laplace approximation with the exact Hessian is used. The OOD datasets for this case are FMNIST [33], notMNIST [34], and KMNIST [35]. (ii) For larger datasets, i.e. CIFAR-10 [36], SVHN [37], and CIFAR-100 [36], we use a ResNet-18 network [38]. Since this network is large, (8) in conjunction with a full Laplace approximation is too costly. We, therefore, use a last-layer Laplace approximation to obtain the approximate diagonal Gaussian posterior. The OOD datasets for CIFAR-10, SVHN, and CIFAR-100 are SVHN and CIFAR100; CIFAR-10 and CIFAR-100; and

SVHN and CIFAR-10, respectively. In all scenarios, the networks are well-trained with 99% accuracy on MNIST, 95.4% on CIFAR-10, 76.6% on CIFAR-100, and 100% on SVHN. For the sampling baseline, we use 1000 posterior samples to compute the predictive distribution. We use the mean of the Dirichlet to obtain a comparable approximation to the MC-integral. Further comparisons with a KFAC approximation [9] of the last layer and ensemble networks can be found in the supplements.

The results are presented in Table 1. The Laplace Bridge is competitive to the baseline w.r.t. MMC and AUROC. The Laplace Bridge is even able to win around half of the comparisons with KFAC sampling, a state of the art feasible approximation of the Hessian. The reason why the KFAC approximation is not shown for MNIST is that the Hessian for the full (all-layer) Laplace approximation does not fit into memory. The key observation, however, is that the Bridge is on average around 400 times faster than the sampling baseline while returning at least competitive, if not even improved fidelity.

5.3 Time comparison

We compare the computational cost of the density-estimated p_{sample} distribution via sampling and the Dirichlet distribution obtained from the Laplace Bridge p_{LB} for approximating the true distribution p_{true} over softmax-Gaussian samples[2]. Different amounts of samples are drawn from the Gaussian, the softmax is applied and the KL divergence between the histogram of the samples with the true distribution is computed. We use KL-divergences $D_{\text{KL}}(p_{\text{true}}||p_{\text{sample}})$ and $D_{\text{KL}}(p_{\text{true}}||p_{\text{LB}})$, respectively, to measure similarity between the approximations and ground truth while the number of samples for p_{sample} is increased on a logarithmic scale. The true distribution p_{true} is constructed via MC with 100k samples. The experiment is conducted for three different Gaussian distributions over $\mathbb{R}^3$. Since the softmax applied to a Gaussian does not have a closed-form analytic solution, the algebraic calculation of the approximation error is not possible and an empirical evaluation via sampling is the best option. The fact that there is no analytic solution is part of the justification for using the Laplace Bridge in the first place.

Figure 4 suggests that the number of samples required such that the distribution p_{sample} is approximating the true distribution p_{true} as good as the Dirichlet distribution obtained via the Laplace Bridge is large, i.e. somewhere between 500 and 10000. This translates to a wall-clock time advantage of at least a factor of 100 before sampling becomes competitive in quality with the Laplace Bridge.

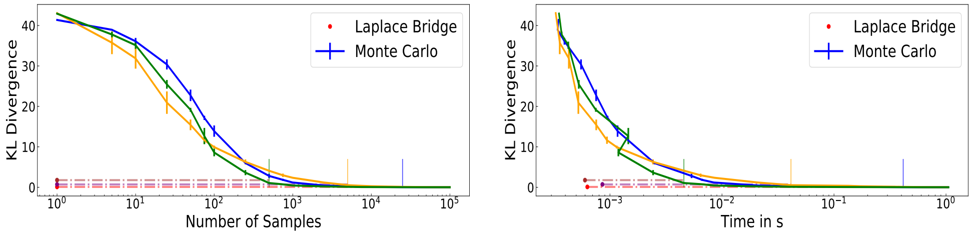

Figure 4: KL-divergence plotted against the number of samples (left) and wall-clock time (right). Monte Carlo density estimation becomes as good as the Laplace Bridge after around 750 to 10000 samples and takes at least 100 times longer. The three lines represent three different samples.

5.4 Uncertainty-aware output ranking on ImageNet

Classification tasks on large datasets with many classes, like ImageNet, are not often done in a Bayesian fashion since the posterior inference and sampling are expensive. The Laplace Bridge, in conjunction with the last-layer Bayesian approximations, can be used to alleviate this problem. Furthermore, having a full distribution over the softmax outputs of a BNN gives rise to new possibilities. For example, one could subsume all classes which have sufficiently overlapping marginal distributions into one if they are semantically similar as illustrated in Figure 5.

Another possibility is to improve the standard classification metrics. Large classification tasks like ImageNet are often compared along a top-5 metric, i.e. it is tested whether the correct class is within the five most probable estimates of the network. Although widely accepted, this metric has some

[2] I.e. samples are obtained by first sampling from a Gaussian and transforming it via the softmax function.

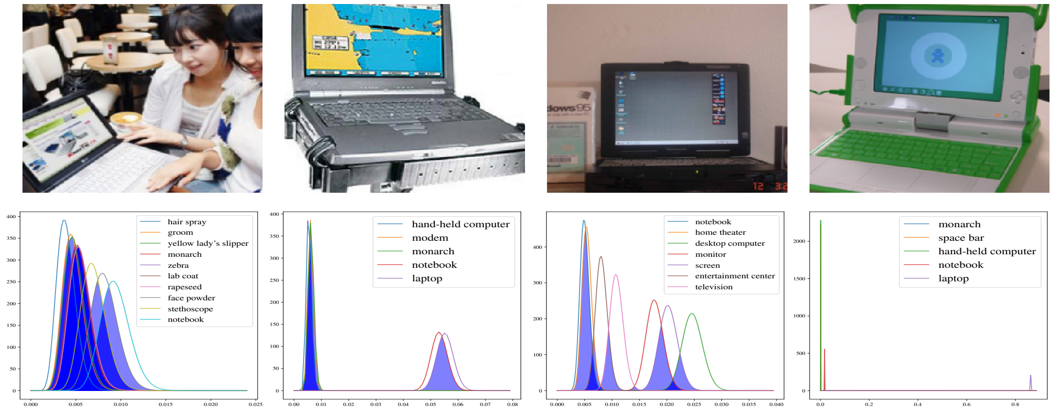

Figure 5: **Upper row:** images from the "laptop" class of ImageNet. **Bottom row:** Beta marginal distributions of the top-k predictions for the respective image. In the first column, the overlap between the marginal of all classes is large, signifying high uncertainty, i.e. the prediction is "I do not know". In the column, "notebook" and "laptop" have confident, yet overlapping marginal densities and we, therefore, have a top-2 prediction: "either a notebook or a laptop". In the third column "desktop computer", "screen" and "monitor" have overlapping marginal densities, yielding a top-3 estimate. The last case shows a top-1 estimate: the network is confident that "laptop" is the only correct label.

pathologies. We can easily construct examples where the top-5 include either too many or too few classes for our purposes which a static rule (always 5) can't handle.

Leveraging the probabilistic output provided by the Laplace Bridge, we propose a simple decision rule that can handle such examples and is more fine-grained due to its awareness of uncertainty. One may call such a rule *uncertainty-aware top-k*; pseudocode for the algorithm is given in the supplements. Instead of taking the top-k as a decision threshold for an arbitrary k we take the uncertainty/confidence of the model to inform the decision. This is more flexible and therefore able to handle situations in which different numbers of classes are plausible outcomes. The Dirichlet distribution obtained from the Laplace Bridge provides this capability. In particular, since the marginal distribution over each component of a Dirichlet distribution is a $\mathrm{Beta}(\alpha_i, \sum_{j \neq i} \alpha_j)$, this can be done analytically and efficiently. The proposed decision rule uses the area of overlap between the marginal distributions of the sorted outcomes. This is similar to hypotheses testing, i.e. t-tests [39] or its Bayesian alternatives [40]. If, for example, two Beta densities overlap more than 5%, we cannot say that they are different distributions with high confidence. All distributions that have sufficient overlap should become the new top-k estimate. Figure 5 shows four examples from the "laptop" class of ImageNet.

We evaluate this decision rule on the test set of ImageNet. The overlap is calculated through the inverse CDF[3] of the respective Beta marginals. The original top-1 accuracy of DenseNet on ImageNet is 0.744. Meanwhile, the uncertainty-aware top-k accuracy is 0.797, where k is on average 1.688. Most of the predictions given by the uncertainty-aware metric still yielded a top-1 prediction (see appendix). This shows that using uncertainty does not imply adding meaningless classes to the prediction. However, there are some non-negligible cases where k equals to 2, 3, or 10. This indicates that whenever there is ambiguity in the class labels, our method is able to detect it, and thus yields a significantly higher accuracy.

6 Conclusion

We have adapted an old but overlooked approximation scheme for new use in Bayesian Deep Learning. Given a Gaussian approximation to the weight-space posterior of a Bayesian neural network and an input, the Laplace Bridge analytically maps the marginal Gaussian prediction on the logits onto a Dirichlet distribution over the softmax vectors. The associated computational cost of $\mathcal{O}(K)$ for K-class prediction compares favorably to that of Monte Carlo sampling. The proposed method both theoretically and empirically preserves predictive uncertainty, offering an attractive, low-cost, high-quality alternative to Monte Carlo sampling. In conjunction with a low-cost, last-layer Bayesian approximation, it can be useful in real-time applications wherever uncertainty is required.

[3] Also known as the quantile function or percent point function

7 Broader Impact

More and more tasks are solved through Deep Learning and Neural Networks. While they often provide state-of-the-art results in terms of their accuracy there are nearly no theoretical bounds on their behavior when confronted with new situations. It is therefore of high importance for a Neural Network to be able to provide well-calibrated uncertainty about its predictions. A network has to be able to say "I don't know" when it receives data that it can't classify sufficiently well or which are far away from the training distribution. Especially in safety-critical tasks such as self-driving vehicles or medical applications uncertainty estimates or even fully parameterized distributions over the output are even more important since the decisions can now be better informed. A self-driving car, for example, can be especially careful when its uncertainty about the class "child" is high.

While the field of Bayesian Deep Learning (BDL) is rapidly improving, many of its applications have one of two problems: either (i) acquiring the uncertainty estimate is computationally expensive since it involves sampling or (ii) Bayesian methods yield good uncertainty estimates but don't yield the same accuracy as conventional methods.

Reducing the computational overhead of BDL is important, especially during test time, because it implies viability for applications where either (i) small differences in time can make large differences in outcome (e.g. breaking earlier to prevent and accident) or (ii) uncertainty estimates are required in rapid succession (e.g. multiple hundred frames per second). Additionally, it also implies less energy usage and, thereby, higher accessibility because of the reduced cost. However, our method mostly saves overhead during test time and not during training. Therefore, we expect the effects on the climate and access to be marginal compared to its other benefits.

While our method, the Laplace Bridge for Neural Networks, does by no means solve the problems of fast and precise uncertainty estimates, it is one step closer. Computing a fully parameterized distribution over the outputs is faster than drawing one (!) sample from the posterior predictive Gaussian and thereby allows for the just described benefits during test time. At the same time, it can be applied to already trained networks such that conventionally effective methods can be used for training without loss of accuracy.

Improving a method and opening up new use cases has problems with dual-use. The Laplace Bridge, while applicable to e.g. self-driving vehicles, could also be used in a drone or other intelligent weapons.

References

[1] E. Begoli, T. Bhattacharya, and D. Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nat Mach Intell*, 1:20–23, 2019.

[2] Rowan McAllister, Yarin Gal, Alex Kendall, Mark van der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In *IJCAI*, 2017.

[3] Rhiannon Michelmore, Marta Kwiatkowska, and Yarin Gal. Evaluating uncertainty quantification in end-to-end autonomous driving control. *CoRR*, abs/1811.06817, 2018.

[4] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

[5] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, May 1992. ISSN 0899-7667.

[7] 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.

[8] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *ArXiv*, 2015.

[9] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018.

[10] 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.

[11] David J.C. MacKay. Choice of basis for laplace approximation. *Machine Learning*, 33(1): 77–86, Oct 1998. ISSN 1573-0565.

[12] P. Hennig, D. Stern, R. Herbrich, and T. Graepel. Kernel topic models. In *Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *JMLR Proceedings*, pages 511–519. JMLR.org, 2012.

[13] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep 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 2171–2180, Lille, France, 07–09 Jul 2015. PMLR.

[14] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain, 09–11 May 2016. PMLR.

[15] Nicolas Brosse, Carlos Riquelme, Alice Martin, Sylvain Gelly, and Éric Moulines. On last-layer algorithms for classification: Decoupling representation from uncertainty estimation. *arXiv preprint arXiv:2001.08049*, 2020.

[16] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *ICML*, 2016.

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

[18] James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *ICML*, 2015.

[19] David J Spiegelhalter and Steffen L Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605, 1990.

[20] David JC MacKay. The evidence framework applied to classification networks. *Neural computation*, 4(5):720–736, 1992.

[21] Michalis Titsias. One-vs-each approximation to softmax for scalable estimation of probabilities. In *NIPS*, 2016.

[22] Amr Ahmed and Eric Xing. On tight approximate inference of the logistic-normal topic admixture model. In *Proceedings of the 11th Tenth International Workshop on Artificial Intelligence and Statistics*, 2007.

[23] Michael Braun and Jon McAuliffe. Variational inference for large-scale models of discrete choice. *Journal of the American Statistical Association*, 105(489):324–335, 2010.

[24] Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E. Turner, José Miguel Hernández-Lobato, and Alexander L. Gaunt. Fixing variational bayes: Deterministic variational inference for bayesian neural networks. *CoRR*, abs/1810.03958, 2018. URL `http://arxiv.org/abs/1810.03958`.

[25] Manuel Haussmann, Sebastian Gerwinn, and Melih Kandemir. Bayesian evidential deep learning with pac regularization, 2019.

[26] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pages 7047–7058, 2018.

[27] Andrey Malinin and Mark Gales. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 14520–14531, 2019.

[28] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems*, pages 3179–3189, 2018.

[29] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[31] Felix Dangel, Frederik Kunstner, and Philipp Hennig. Backpack: Packing more into backprop. *arXiv preprint arXiv:1912.10985*, 2019.

[32] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016.

[33] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[34] Yaroslav Bulatov. notmnist dataset. 2011. URL `http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html`.

[35] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *CoRR*, abs/1812.01718, 2018.

[36] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[37] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[39] Raymond S Nickerson. Null hypothesis significance testing: a review of an old and continuing controversy. *Psychological methods*, 5(2):241, 2000.

[40] Michael E. J. Masson. A tutorial on a practical bayesian alternative to null-hypothesis significance testing. *Behavior Research Methods*, 43(3):679–690, Sep 2011. ISSN 1554-3528.