EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Mathematisch-
Naturwissenschaftliche
Fakultät

Methoden des Maschinellen Lernens

Masterarbeit

Fast Predictive Uncertainty for Classification with Bayesian Deep Networks

Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik
Methoden des Maschinellen Lernens
Marius Hobbhahn, `marius.hobbhahn@student.uni-tuebingen.de`, 2019/20

Bearbeitungszeitraum:	vom 20.02.20 bis 20.06.20

Betreuer/Gutachter:	Prof. Dr. Philipp Hennig, Universität Tübingen
Zweitgutachter:	Prof. Dr. Zeynep Akata, Universität Tübingen

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich diese schriftliche Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe.

Marius Hobbhahn (Matrikelnummer 4003731), June 9, 2020

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.

Zusammenfassung

Im Bayesian Deep Learning werden Verteilungen über die Ausgaben von neuronalen Netzen dadurch erzeugt, dass zunächst eine Gaussverteilung über die Gewichte konstruiert wird und aus dieser dann samples gezogen werden, welche, nach Anwendung der softmax Funktion, eine kategorische Verteilung darstellen. Das ist aufwendig. Wir nutzen altes, aber nützliches, Wissen um eine Dirichlet Approximation der Ausgabeverteilung zu konstruieren. Diese Brücke stellt eine analytische Abbildung zwischen einer Normalverteilung im logit-Raum und einer Dirichletverteilung (dem conjugate prior der Kategorischen Verteilung) im Ausgaberaum dar. Wir argumentieren, dass die resultierende Dirichletverteilung theoretische und praktische Vorteile hat. Die Berechnung der Unsicherheit bezüglich der Ausgabeverteilung ist beispielsweise wesentlich effizienter und die Methode lässt sich einfach auf große Datensätze und Netzwerke, wie ImageNet und DenseNet, hochskalieren. Wir demonstrieren diesen Fakt mit einem Unsicherheits-bewussten Ausgaberanking für ImageNet.

Contents

1 Introduction

Quantifying the uncertainty of neural networks' (NNs) predictions is important in safety-critical applications such as medical-diagnosis Begoli et al. (2019) and self-driving vehicles McAllister et al. (2017); Michelmore et al. (2018). 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 (Nguyen et al., 2015; Hein et al., 2019) 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 (MacKay, 1992a; Graves, 2011; Blundell et al., 2015; Ritter et al., 2018) approximately induces a Gaussian distribution over the logits of a NN (Mackay, 1995). 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 thesis, we re-introduce an old but largely overlooked idea originally proposed by David JC MacKay (1998) 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 (Hennig et al., 2012), 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 1.1). Our contributions in this thesis 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

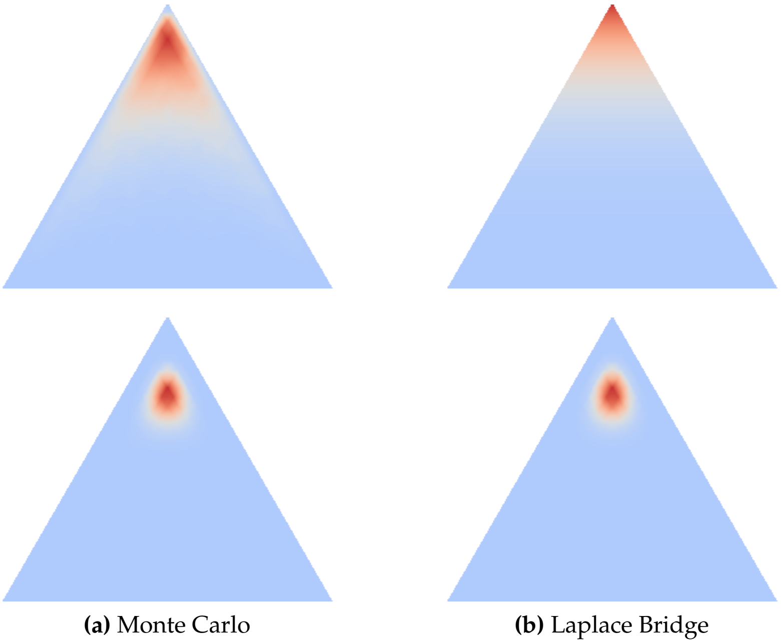


Figure 1.1: Densities on the simplex of the true distribution (left column, computed by exhaustive sampling by mapping a Gaussian random variable through the softmax transformation) and "Laplace Bridge" approximation constructed in this thesis (right column). For the top and bottom rows, two different Gaussians were used, such that the resulting mode is the same, but the uncertainty differs.

significant speed-up over MC-integration. Finally, we show a use-case, leveraging the analytic properties of Dirichlet distributions to improve the popular top-k metric through uncertainties.

To sharpen the intuition about the modifications to classical BNNs proposed by this thesis consider Figure 1.2.

We think that the Laplace Bridge is a valuable method to estimate predictive uncertainty because it is easy to add to already existing architectures and it is very fast compared to sampling schemes. When combined with a Laplace approximation of the weights, the Laplace Bridge can use pre-trained models and is, therefore, a simple extension to existing architectures. The cost of computing the Laplace Bridge are lower than drawing one (!) sample from a Gaussian distribution over the logits and the result is a fully parameterized Dirichlet distribution over the output space. This implies that the computational cost during application is reduced to a minimum. Having fast predictive uncertainty is important because it means viability for safety-critical applications, e.g.

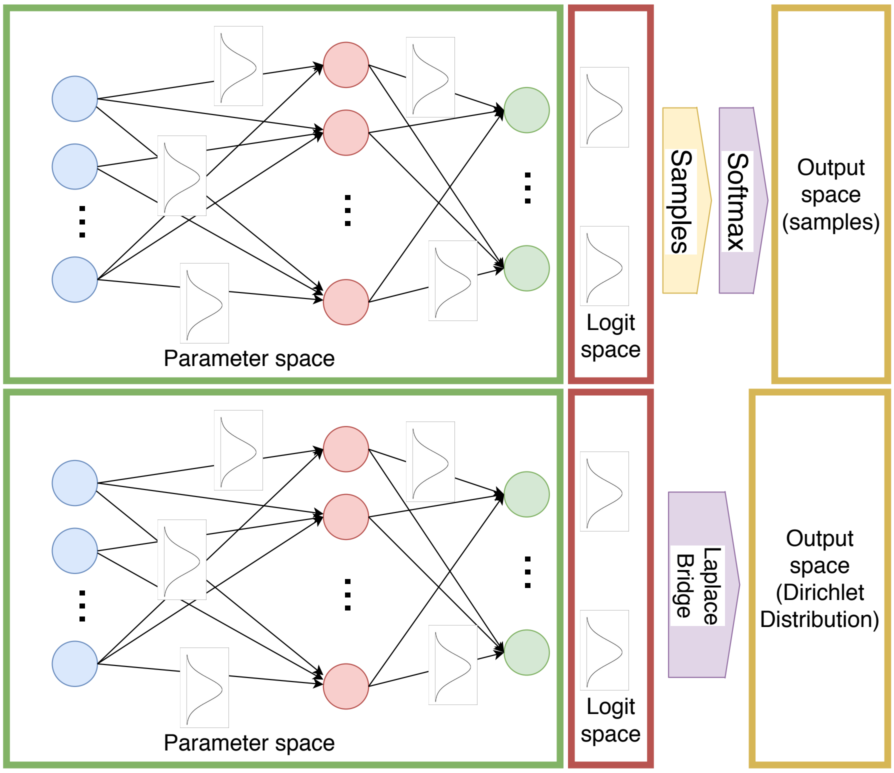


Figure 1.2: Visualization of the modification of BNNs through the Laplace Bridge. The *Parameter space* part is representative for any method that yields a Gaussian predictive posterior in the logit space. **Top:** The softmax function is applied to samples drawn from the posterior predictive Gaussian. The result is a distribution represented by samples. **Bottom:** The Laplace Bridge is applied to the posterior predictive Gaussian. The result is a parameterized Dirichlet distribution over the outputs.

self-driving cars where a difference of milliseconds can increase safety and be used in rapid succession for multiple hundred frames per second.

Chapter 2 provides the mathematical derivation. Chapter 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 Chapter 4. Empirical experiments are presented in Chapter 5.

2 The Laplace Bridge

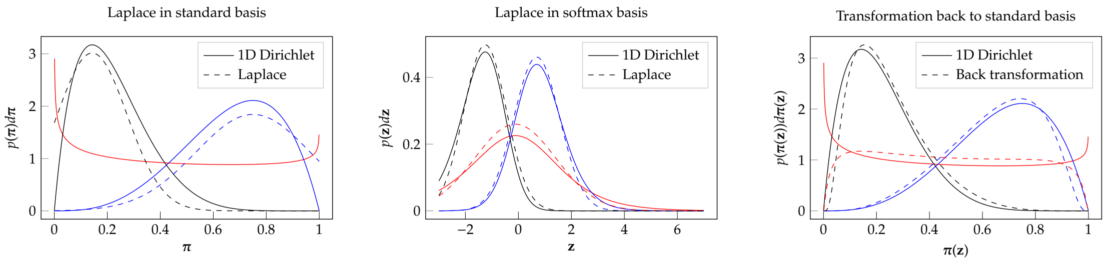


Figure 2.1: (Adapted from Hennig et al. (2012)). Visualization of the Laplace Bridge for the Beta distribution (special 1D case of the Dirichlet). **Left:** “Generic” Laplace approximations of standard Beta distributions by Gaussians. Note that the Beta Distribution (red curve) does not even have a valid approximation because the Hessian is not positive semi-definite. **Middle:** Laplace approximation to the same distributions after basis transformation through the softmax (7.4). 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.

The Dirichlet distribution, which has the density function

$$\operatorname{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) := \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}, \tag{2.1}$$

[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.

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 (1998) noted that both can be fixed, elegantly, by a change of variable. Details of the following argument can be found in the appendix. Consider the K-dimensional variable $\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$ defined as the softmax of $\mathbf{z} \in \mathbb{R}^K$:

$$\pi_k(\mathbf{z}) := \frac{\exp(z_k)}{\sum_{l=1}^{K} \exp(z_l)}, \tag{2.2}$$

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

$$\det \frac{\partial \boldsymbol{\pi}}{\partial \mathbf{z}} = \prod_k \pi_k(z), \tag{2.3}$$

thus removing the -1 terms in the exponent:

$$\mathrm{Dir}_{\mathbf{z}}(\boldsymbol{\pi}(\mathbf{z})|\boldsymbol{\alpha}) := \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k(\mathbf{z})^{\alpha_k}, \tag{2.4}$$

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

$$\mu_k = \log \alpha_k - \frac{1}{K} \sum_{l=1}^{K} \log \alpha_l \tag{2.5}$$

$$\Sigma_{k\ell} = \delta_{k\ell} \frac{1}{\alpha_k} - \frac{1}{K} \left[\frac{1}{\alpha_k} + \frac{1}{\alpha_\ell} - \frac{1}{K} \sum_{u=1}^{K} \frac{1}{\alpha_u} \right]. \tag{2.6}$$

A visualization of the Laplace Bridge for the one-dimensional special case can be found in figure 2.1. 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. (2012).

It maps the Gaussian parameters to those of the Dirichlet as

$$\alpha_k = \frac{1}{\Sigma_{kk}} \left(1 - \frac{2}{K} + \frac{e^{\mu_k}}{K^2} \sum_{l=1}^{K} e^{-\mu_l} \right), \tag{2.7}$$

(Note that this equation ignores off-diagonal elements of $\mathbf{\Sigma}$, more discussion below). Together, Eqs. 2.5, 2.6 and 2.7 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 Chapter 3 that this map is still sufficiently accurate.

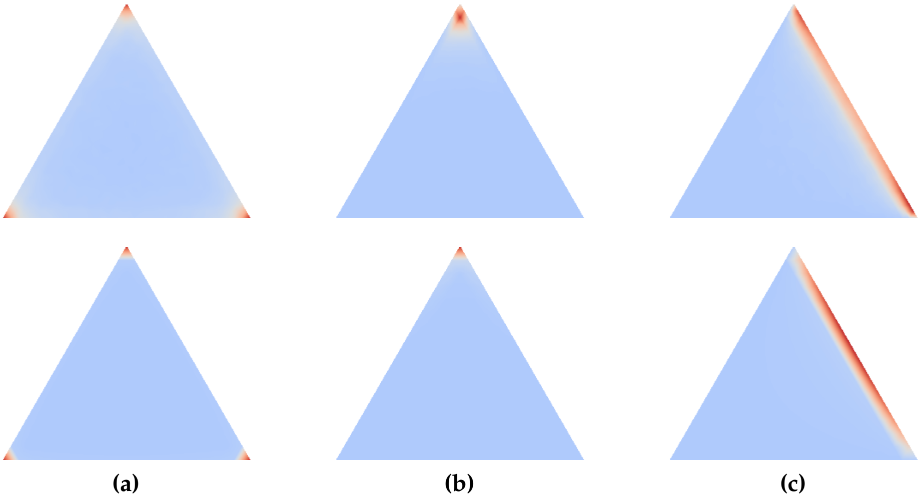


Figure 2.2: As in Figure 1.1, more densities of the true distribution (top) arising from mapping a Gaussian random variable through the softmax, and the corresponding Dirichlet pdf produced by the Laplace Bridge (bottom). The Dirichlet approximation, with its reduced parameter-space, captures most of the features of the ground-truth distribution.

Figures 1.1, 2.2 and 2.3 show the quality of the resulting approximation. We consider multiple different $\boldsymbol{\mu}, \mathbf{\Sigma}$ in three dimensions, i.e. simulating a classification task with three classes. We sample from the Gaussian and apply the softmax transform to all samples and compare the resulting histogram on the simplex to the probability density function of the corresponding Dirichlet. Figure 1.1 emphasizes that a point estimate is insufficient. Since the mean for the Dirichlet is the normalized $\boldsymbol{\alpha}$ parameter vector, the parameters that generate Figure 1.1 ($\boldsymbol{\alpha}_1 = [2,2,6]^\top$ and $\boldsymbol{\alpha}_2 = [11,11,51]^\top$) yield the same point estimate even though their distributions are clearly different. The figures show that the Laplace Bridge is a sufficiently good approximation and that it maps a change of uncertainty as

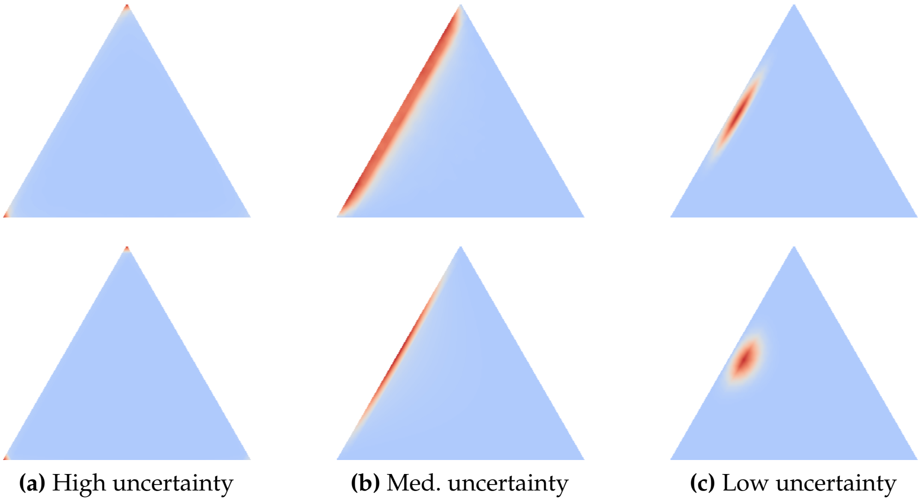


Figure 2.3: The densities (via histograms) of the true predictive distribution (top) arising from a Gaussian random variable and the corresponding densities approximated via the Laplace Bridge (bottom).

expected.

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

$$q(\mathbf{z}|\mathbf{x}) \approx \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\boldsymbol{\theta}}^\top \mathbf{x}, \mathbf{J}(\mathbf{x})^\top \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \mathbf{J}(\mathbf{x}))\,, \tag{3.1}$$

where $\mathbf{J}(\mathbf{x})$ is the $P \times K$ Jacobian matrix representing the derivative $\frac{\partial \mathbf{z}}{\partial \boldsymbol{\theta}}$ (Mackay, 1995). 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. (2.7), 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. For output classes of "comparably high" probability (as defined below), the variance $\mathrm{Var}(\pi_k|\boldsymbol{\alpha})$ under the Laplace Bridge increases with the variance of the underlying Gaussian. In this sense, the Laplace Bridge approximates the uncertainty information encoded in the output of a BNN.

Proposition 1 (proof in supplements). *Let* $\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$ *be obtained via the Laplace Bridge from a Gaussian distribution* $\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *over* $\mathbb{R}^K$. *Then, for each* $k = 1, \dots, K$, *letting* $\alpha_{\neq k} := \sum_{l \neq k} \alpha_l$, *if*

$$\alpha_k > \frac{1}{4}\left(\sqrt{9\alpha_{\neq k}^2 + 10\alpha_{\neq k} + 1} - \alpha_{\neq k} - 1\right),$$

then the variance $\mathrm{Var}(\pi_k|\boldsymbol{\alpha})$ *of the k-th component of* $\boldsymbol{\pi}$ *is increasing in* $\boldsymbol{\Sigma}_{kk}$.

Intuitively, this result describes the condition that needs to be fulfilled such that the variance of the resulting Dirichlet scales with the variance of the k-th component of the Gaussian. It can be seen as a proxy for a high quality approximation. An empirical evaluation testing the frequency of the condition being fulfilled can be found in the appendix.

Further 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,

$$\mathbb{E}\boldsymbol{\pi} = \left(\frac{\alpha_1}{\sum_{l=1}^K \alpha_l}, \dots, \frac{\alpha_K}{\sum_{l=1}^K \alpha_l} \right)^\top , \tag{3.2}$$

which can be seen in the second image of Figure 2.1. Further, Dirichlets have Dirichlet marginals: If $p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$, then

$$p([\pi_1, \pi_2, \dots, \pi_j, \sum_{k>j} \pi_k]^\top) = \text{Dir}(\alpha_1, \alpha_2, \dots, \alpha_j, \sum_{k>j} \alpha_k) . \tag{3.3}$$

An additional benefit of the Laplace Bridge for BNNs is that it is more flexible than a MC-integral. If we let $p(\boldsymbol{\pi})$ be the distribution over $\boldsymbol{\pi} := \text{softmax}(\mathbf{z}) := [e^{z_1} / \sum_l e^{z_l}, \dots, e^{z_K} / \sum_l e^{z_l}]^\top$, 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 $\text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$ 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 (3.3), the Laplace Bridge provides a convenient yet cheap way of answering this question.

3.1 Posterior inference

In principle, the Gaussian over the weights required by the Laplace Bridge for BNNs (see Equation 3.1) can be constructed by any Gaussian approximate Bayesian methods such as variational Bayes (Graves, 2011; Blundell et al., 2015) and Laplace approximations for neural networks (MacKay, 1992a; Ritter et al., 2018). 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

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{(\mathbf{x},t)\in\mathcal{D}} p(y = t|\boldsymbol{\theta}), \tag{3.4}$$

be the posterior over the parameters $\boldsymbol{\theta}$ 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

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

For various applications in Deep Learning, the approximation in (3.1) 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 $O(PK)$ 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. (2015); Wilson et al. (2016), etc. and has been shown empirically to be effective in uncertainty quantification tasks (Brosse et al., 2020). In this case, given the approximate last-layer posterior

$$p(\mathbf{W}^L|\mathcal{D}) \approx \mathcal{N}(\text{vec}(\mathbf{W}^L)|\text{vec}(\mathbf{W}^L_{\text{MAP}}), \mathbf{H}^{-1}_{\mathbf{W}^L}), \tag{3.5}$$

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

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{W}^L_{\text{MAP}}\boldsymbol{\phi}(\mathbf{x}), (\boldsymbol{\phi}(\mathbf{x})^\top \otimes \mathbf{I})\mathbf{H}^{-1}_{\mathbf{W}^L}(\boldsymbol{\phi}(\mathbf{x}) \otimes \mathbf{I})), \tag{3.6}$$

where ⊗ denotes the Kronecker product.

An even more efficient last-layer approximation can be obtained using a Kronecker-factored matrix normal distribution (Louizos, Welling, 2016; Sun et al., 2017; Ritter et al., 2018). That is, we assume the posterior distribution to be

$$p(\mathbf{W}^L|\mathcal{D}) \approx \mathcal{MN}(\mathbf{W}^L|\mathbf{W}^L_{\text{MAP}}, \mathbf{U}, \mathbf{V}) \,, \tag{3.7}$$

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 $\mathbf{H}^{-1}_{\mathbf{W}^L}$ (Martens, Grosse, 2015). In this case, for any $\mathbf{x} \in \mathbb{R}^N$, one can easily show that the distribution over logits is given by

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{W}^L_{\text{MAP}}\boldsymbol{\phi}(\mathbf{x}), (\boldsymbol{\phi}(\mathbf{x})^\top \mathbf{V} \boldsymbol{\phi}(\mathbf{x}))\mathbf{U}) \,, \tag{3.8}$$

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

$$p(\mathbf{W}^L|\mathcal{D}) \approx \mathcal{N}(\text{vec}(\mathbf{W}^L)|\text{vec}(\mathbf{W}^L_{\text{MAP}}), \text{diag}(\boldsymbol{\sigma}^2)) \,, \tag{3.9}$$

where $\boldsymbol{\sigma}^2$ is obtained via the diagonal of the Hessian of the log-posterior w.r.t. vec($\mathbf{W}^L$) at 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 (Spiegelhalter, Lauritzen, 1990; MacKay, 1992b). However, while there exist some bounds (Titsias, 2016) and approximations of the expected log-sum-exponent function (Ahmed, Xing, 2007; Braun, McAuliffe, 2010), 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 of interest in this domain, too, as the approximation of this integral can be analytically computed via (3.2).

Approaches like Wu et al. (2018) and Haussmann et al. (2019) show the effectiveness of, and a general desire to provide sampling free methods for Bayesian Deep Learning. The Laplace Bridge furthers this trend since it is also sampling-free.

Recently, it has been proposed to model the distribution of softmax outputs of a network directly. Similar to the Laplace Bridge, Malinin, Gales (2018, 2019); Sensoy et al. (2018) 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 (Graves, 2011; Blundell et al., 2015; Louizos, Welling, 2016; Sun et al., 2017, etc) without any additional training procedure. Further differences between the Laplace Bridge and Malinin, Gales (2018, 2019); Sensoy et al. (2018) include a) they require retraining of the network while ours can use pre-trained weights. The Laplace Bridge is, therefore, easier to apply to already-existing architectures and b) Malinin, Gales (2018, 2019) require OOD samples during training while our method bases its uncertainty estimate solely on the information already included in the weights.

5 Experiments

We conduct five experiments. In Section 5.1, we analyze the approximation quality of the Laplace Bridge applied to a BNN on the MNIST LeCun, Cortes (2010) 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. In Section 5.4 we visualize some properties of the Laplace Bridge and compare it to sampling-based methods. Finally, in Section 5.5, we present analysis on ImageNet Russakovsky et al. (2014) to demonstrate the scalability of the Laplace Bridge and the advantage of having a full Dirichlet distribution over softmax outputs.

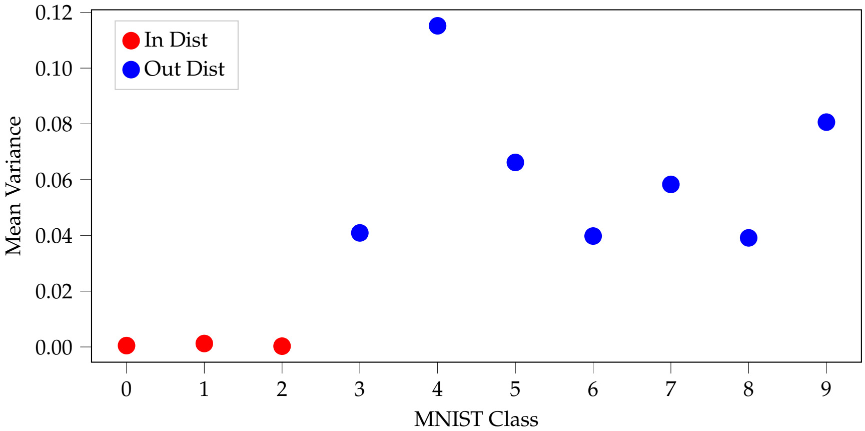
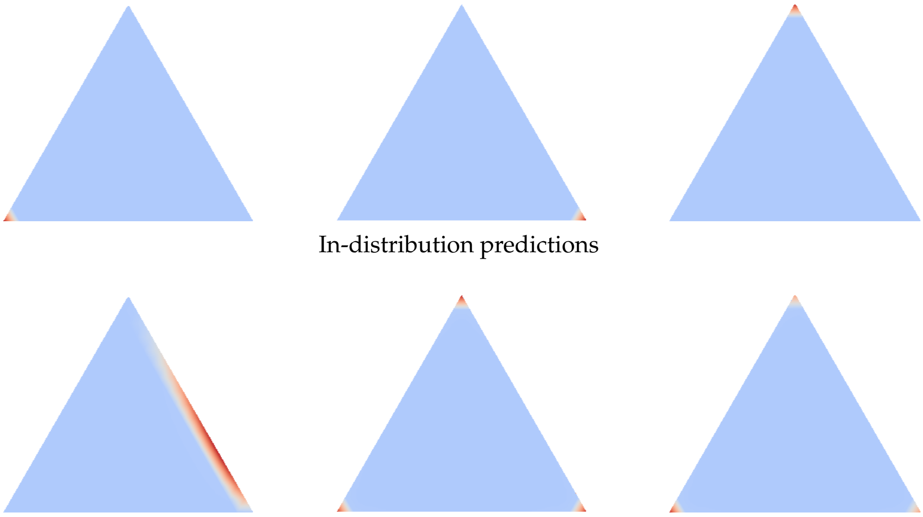


Figure 5.1: Average variance of the Dirichlet distributions of each MNIST class. The in-distribution uncertainty (variance) is nearly nil, while out-of-distribution variance is high.

5.1 Uncertainty estimates on MNIST

We empirically investigate the approximation quality of the Laplace Bridge in a "real-world" BNN on the MNIST dataset. A convolutional network with 2 convolutional and 2 fully-connected layers is trained on the first three digits of MNIST (the digits 0, 1, and 2). Adam optimizer with learning rate 1e-3 and weight decay 5e-4 is used. The batch size is 128. To obtain the posterior over the weights of this network, we perform a full (all-layer) Laplace approximation using BackPACK (Dangel et al., 2019) 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 5.1. We show for each $k = 1, \dots, K$, the average variance $\frac{1}{D_k} \sum_{i=1}^{D_k} \mathrm{Var}(\pi_k(f_\theta(\mathbf{x}_i)))$ 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: The mean variance of the first three classes is close to zero, while that of the other classes is higher. Therefore, these variances are informative for detecting OOD data. Samples of the in- and out-of-distribution sets reflect this difference in uncertainty, as shown in Figure 5.2. While these results could also be obtained via sampling, the Laplace Bridge provides a computationally lightweight alternative for estimating predictive uncertainty.



In-distribution predictions

Out-of-distribution predictions

Figure 5.2: Top: In-distribution pdfs. All probability mass is concentrated in the corner of the respective correct class. **Bottom:** Out-of-distribution pdfs. The probability mass is distributed more equally since the networks' uncertainty about is higher.

		Diag Sampling			**Laplace Bridge (mean)**		
Train	**Test**	**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	0.949 ± 0.001	-	6.6	**0.969** ± 0.002	-	**0.017**
CIFAR-10	CIFAR-100	**0.724** ± 0.002	**0.884** ± 0.004	6.6	0.774 ± 0.003	0.858 ± 0.004	**0.016**
CIFAR-10	SVHN	**0.659** ± 0.028	**0.931** ± 0.007	17.0	0.704 ± 0.036	0.923 ± 0.008	**0.041**
SVHN	SVHN	0.986 ± 0.000	-	17.1	**0.991** ± 0.000	-	**0.040**
SVHN	CIFAR-10	0.537 ± 0.012	0.995 ± 0.000	6.61	**0.392** ± 0.016	**0.996** ± 0.000	**0.169**
SVHN	CIFAR-100	0.543 ± 0.009	0.994 ± 0.000	6.61	**0.400** ± 0.013	**0.996** ± 0.000	**0.016**
CIFAR-100	CIFAR-100	**0.527**s ± 0.004	-	6.68	0.263 ± 0.003	-	**0.017**
CIFAR-100	CIFAR-10	0.276 ± 0.004	**0.707** ± 0.004	6.67	**0.068** ± 0.003	0.703 ± 0.003	**0.018**
CIFAR-100	SVHN	0.348 ± 0.014	0.647 ± 0.011	17.2	**0.074** ± 0.012	**0.661** ± 0.013	**0.040**

Table 5.1: OOD detection results. Optimally, the MMC for OOD data is low and the AUROC is high. 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. Time is measured in seconds. Five runs with different seeds per experiment were conducted. 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.

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 (Hendrycks, Gimpel, 2016). 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 employed. The OOD datasets for this case are FMNIST Xiao et al. (2017), notMNIST Bulatov (2011), and KMNIST Clanuwat et al. (2018). (ii) For larger datasets, i.e. CIFAR-10 Krizhevsky (2009), SVHN Netzer et al. (2011), and CIFAR-100 Krizhevsky (2009), we use a ResNet-18 network (He et al., 2016). Since this network is large, (3.1) 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. Experiments comparing the Laplace Bridge to a KFAC approximation of the last layer and sampling from all weights of the network can be found in the appendix.

The results are presented in Table 5.1. The Laplace Bridge is competitive to the baseline in terms of the MMC and AUROC metrics. In the case of MNIST and SVHN the Bridge is better than the MC-integral w.r.t. the AUROC metric. Moreover, the Laplace Bridge is also better than the sampling baseline in terms of the MMC metric in the SVHN and CIFAR-100 datasets. 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[1]. 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 Monte Carlo 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 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 5.3 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.

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

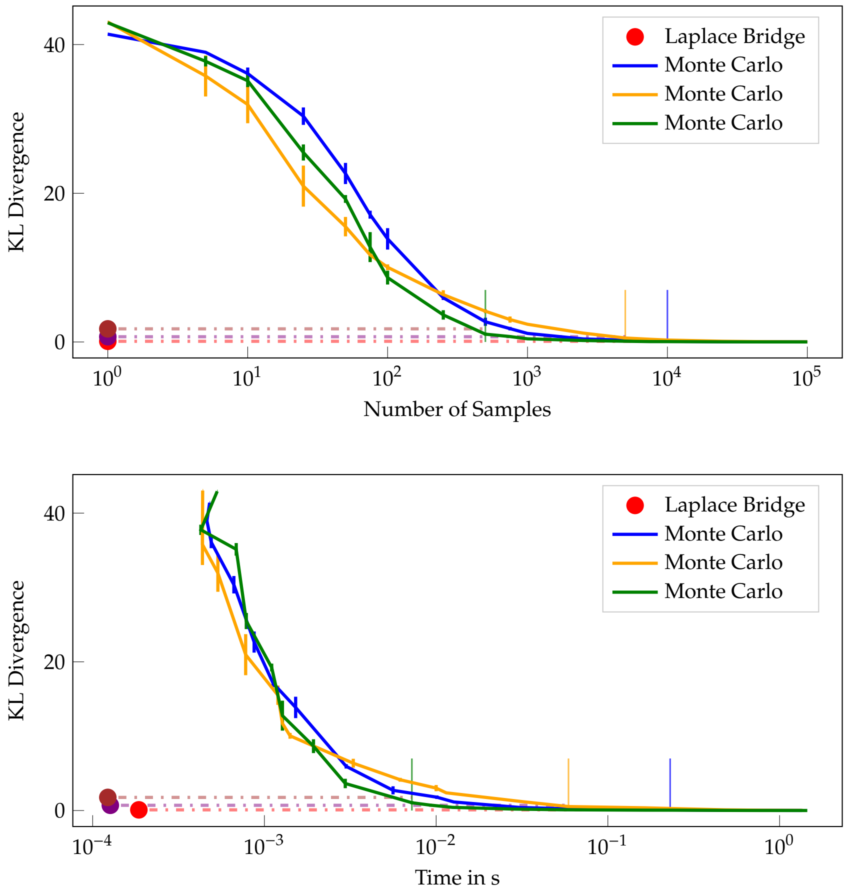


Figure 5.3: KL-divergence plotted against the number of samples (top) and wall-clock time (bottom). Monte Carlo density estimation becomes as good as the Laplace Bridge after around 750 to 10000 samples and takes at least 100 times longer. The three lines represent three different samples.

5.4 Toy dataset

To understand the properties of the Laplace Bridge we visualize its predictions on a toy dataset. The dataset is generated by drawing from four different 2D Gaussians and the task is for a neural network to classify them. The network is a simple four-layer network with ReLU activations and 100 units per layer. A visualization is created by using different methods for calculating predictive uncertainty for all points on a two-dimensional grid. There are four methods to predict uncertainty that are independent of the Laplace Bridge: the MAP estimate, a diagonal approximation of the Hessian, a Kronecker-factorized approximation of the Hessian and the exact Hessian. Their respective predictive entropy can be found on the left column of Figure 5.4. This is compared to the MAP prediction of the Laplace Bridge, its predictive entropy, the variance of the MAP estimate of the Dirichlet and a MAP estimate that is weighted by it's respective variance. These can be found in the right column of Figure 5.4. The estimates in the left column get increasingly better since we include more information about the uncertainty. We conclude that the entropy and the variance of the Dirichlet are only marginally better than the original MAP estimate. Reweighing the estimate by the variance improves it slightly. However, the Laplace Bridge is not able to produce a similarly good estimate as a Kronecker-factorized or exact Hessian.

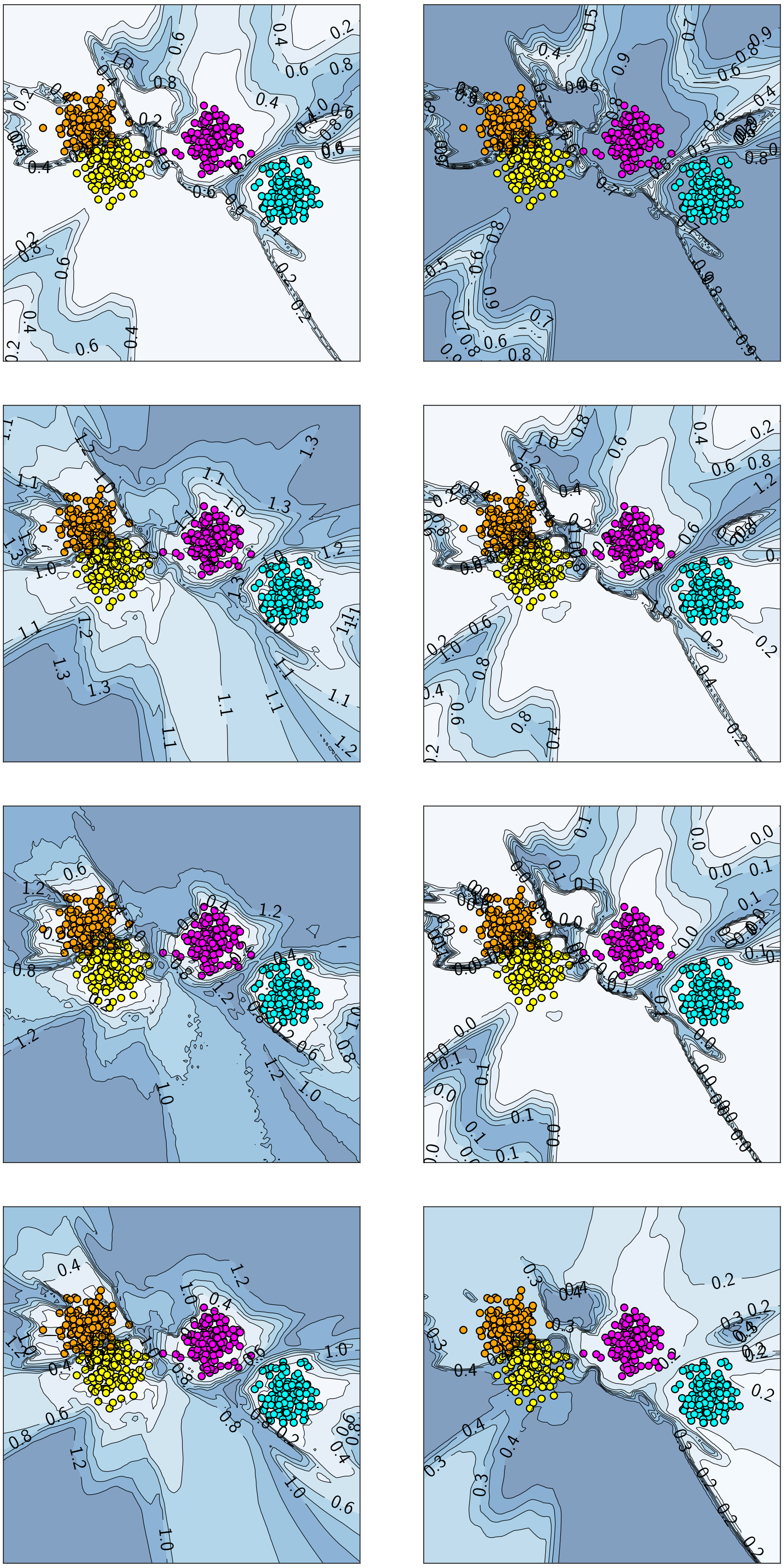


Figure 5.4: Left: Entropy of the MAP estimate, a diagonal approximation of the Hessian, a Kronecker-factorized approximation, and the exact Hessian. **Right:** MAP prediction of the Dirichlet coming from the Laplace Bridge, its predictive entropy, the variance of the Dirichlet, and a MAP estimate weighed by its variance. We find that the Laplace Bridge entropy and variance are only marginally better than the MAP estimate but the reweighed version improves it.

5.5 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 5.5.

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 pathologies. Consider two examples: i) Assume the network has to classify a hypothetical image of "raptor" and it is confident that the label is either a "hawk" or an "eagle". Then all probability mass should be distributed between those two classes. The three other classes within the top-5 are not needed to inform the decision. ii) Assume the network has to classify an image of which it is confident that it is a "fish" but it is uncertain between ten different possible fish species. Which five of the ten fish classes is within the top-5 is nearly arbitrary and so is the thereby following classification.

Leveraging the probabilistic output provided by the Laplace Bridge, we propose a simple decision rule that can handle both examples and is more fine-grained due to its awareness of uncertainty. One may call such a rule *uncertainty-aware top-k*; it is shown in Algorithm 1. 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 $\text{Beta}(\alpha_i, \sum_{j \neq i} \alpha_j)$, 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 Nickerson (2000) or its Bayesian alternatives Masson (2011). 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 5.5 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[2] 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. A more detailed analysis is shown in Figure 5.6. Most of the predictions given by the uncertainty-aware metric still yielded a top-1 prediction.

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

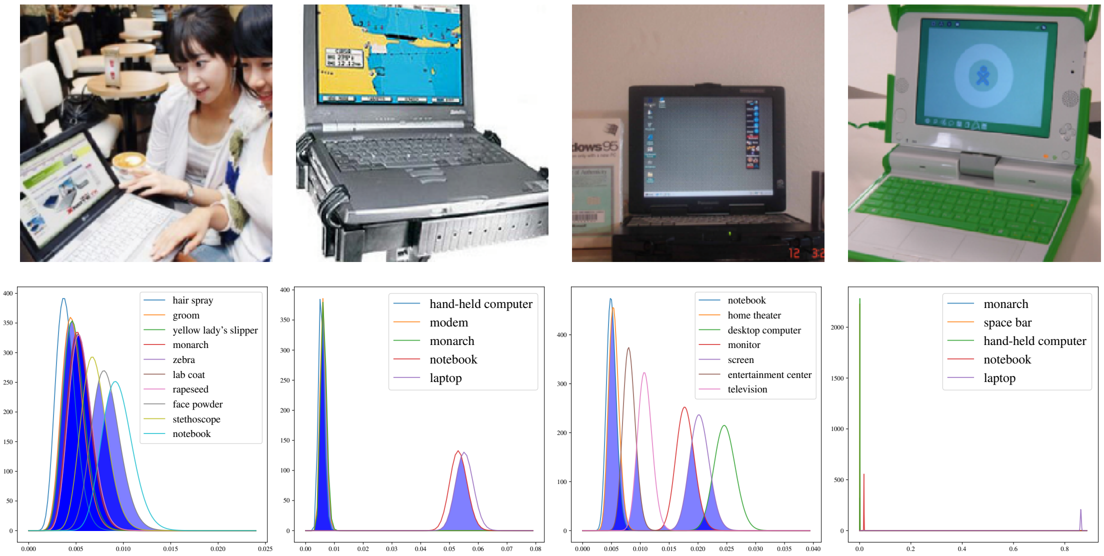


Figure 5.5: Upper row: images from the "laptop" class of ImageNet. **Bottom row:** Beta marginal distributions of the top-k predictions for the respective image. In the first column, the overlap between the marginal of all classes is large, signifying high uncertainty, i.e. the prediction is "I do not 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.

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.

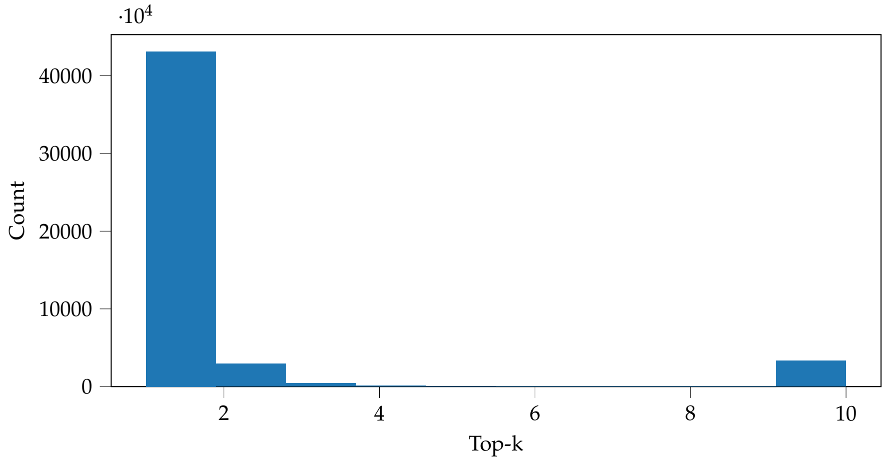


Figure 5.6: A histogram of ImageNet predictions' length using the proposed uncertainty-aware top-k. Most test images are a top-1 prediction, indicating high confidence. There are some top-2, top-3, and top-10 predictions, showing an increasing uncertainty.

Algorithm 1 Uncertainty-aware top-k

Input: A Dirichlet parameter $\boldsymbol{\alpha} \in \mathbb{R}^K$ obtained by applying the Laplace Bridge to the Gaussian over the logit of an input, a percentile threshold T e.g. 0.05, a function class_of that returns the underlying class of a sorted index.

```
$\tilde{\boldsymbol{\alpha}}$ = sort_descending($\boldsymbol{\alpha}$)          // start with the highest confidence
$\alpha_0 = \sum_i \alpha_i$
$C$ = {class_of(1)}          // initialize top-k, must include at least one class
for $i = 2, \dots, K$ do
    $F_{i-1} = \text{Beta}(\tilde{\alpha}_{i-1}, \alpha_0 - \tilde{\alpha}_{i-1})$          // the previous marginal CDF
    $F_i = \text{Beta}(\tilde{\alpha}_i, \alpha_0 - \tilde{\alpha}_i)$          // the current marginal CDF
    $l_{i-1} = F_{i-1}^{-1}(T/2)$          // left $\frac{T}{2}$ percentile of the previous marginal
    $r_i = F_i^{-1}(1 - T/2)$          // right $\frac{T}{2}$ percentile of the current marginal
    if $r_i > l_{i-1}$ then
        $C = C \cup \{\text{class\_of}(i)\}$          // overlap detected, add the current class
    else
        break          // No more overlap, end the algorithm
    end if
end for
```

Output: C *// return the resulting top-k prediction*

6 Discussion

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.

7 Appendix

Appendix A: Background and Proofs

Change of Variable for pdf

Let $\mathbf{x}$ be an n-dimensional continuous random variable with joint density function $p_{\mathbf{x}}$. If $\mathbf{y} = G(\mathbf{x})$, where G is a differentiable function, then $\mathbf{y}$ has density $p_{\mathbf{y}}$:

$$g(\mathbf{y}) = f\Big(G^{-1}(\mathbf{y})\Big)\left|\det\left[\left.\frac{dG^{-1}(\mathbf{z})}{d\mathbf{z}}\right|_{\mathbf{z}=\mathbf{y}}\right]\right| \tag{7.1}$$

where the differential is the Jacobian of the inverse of G evaluated at $\mathbf{y}$. This procedure, also known as 'change of basis', is at the core of the Laplace bridge since it is used to transform the Dirichlet into the softmax basis.

Proof for Proposition

Proof. Considering that α_k is a decreasing function of $\mathbf{\Sigma}_{kk}$ by definition (7.29), it is sufficient to show that under the hypothesis, the derivative of $\frac{\partial}{\partial\alpha_k}\mathrm{Var}(\pi_k|\boldsymbol{\alpha})$ is negative.

By definition, the variance $\mathrm{Var}(\pi_k|\boldsymbol{\alpha})$ is

$$\mathrm{Var}(\pi_k|\boldsymbol{\alpha}) = \frac{\frac{\alpha_k}{\alpha_k+\alpha_{\neq k}} - \frac{\alpha_k^2}{(\alpha_k+\alpha_{\neq k})^2}}{\alpha_k+\alpha_{\neq k}+1}.$$

The derivative is therefore

$$\frac{\partial}{\partial\alpha_k}\mathrm{Var}(\pi_k|\boldsymbol{\alpha}) = \\ \frac{\alpha_{\neq k}(\alpha_{\neq k}^2 - \alpha_{\neq k}\alpha_k + \alpha_{\neq k} - \alpha_k(2\alpha_k+1))}{(\alpha_k+\alpha_{\neq k})^3(\alpha_k+\alpha_{\neq k}+1)^2}.$$

Solving $\frac{\partial}{\partial \alpha_k}\mathrm{Var}(\pi_k|\boldsymbol{\alpha}) < 0$ for α_k yields

$$\alpha_k > \frac{1}{4}\left(\sqrt{9\alpha_{\neq k}^2 + 10\alpha_{\neq k} + 1} - \alpha_{\neq k} - 1\right).$$

Therefore, under this hypothesis, $\mathrm{Var}(\pi_k|\boldsymbol{\alpha})$ is a decreasing function of α_k. □

Experimental Evaluation of the Proposition

To test how often the condition is fulfilled we count its frequency. The fact that the condition is fulfilled implies a good approximation. The fact that the condition is not fulfilled does not automatically imply a bad approximation.

		frequency
MNIST	MNIST	-
MNIST	FMNIST	-
MNIST	notMNIST	-
MNIST	KMNIST	-
CIFAR-10	CIFAR-10	0.998
CIFAR-10	CIFAR-100	0.925
CIFAR-10	SVHN	0.832
SVHN	SVHN	0.999
SVHN	CIFAR-100	0.668
SVHN	CIFAR-10	0.653
CIFAR-100	CIFAR-100	0.662
CIFAR-100	CIFAR-10	0.214
CIFAR-100	SVHN	0.166

Table 7.1

Appendix B: Laplace Approximation of the Dirichlet

Assume we have a Dirichlet in the standard basis with parameter vector $\boldsymbol{\alpha}$ and probability density function:

$$\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) := \frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\pi_k^{\alpha_k-1}, \tag{7.2}$$

We aim to transform the basis of this distribution via the softmax transform to be in the new base π:

$$\pi_k(\mathbf{z}) := \frac{\exp(z_k)}{\sum_{l=1}^{K}\exp(z_l)}, \tag{7.3}$$

Usually, to transform the basis we would need the inverse transformation $H^{-1}(\mathbf{z})$ as described in the main paper. However, the softmax does not have an analytic inverse. Therefore David JC MacKay uses the following trick. Assume we know that the distribution in the transformed basis is:

$$\mathrm{Dir}_{\mathbf{z}}(\boldsymbol{\pi}(\mathbf{z})|\boldsymbol{\alpha}) := \frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\pi_k(\mathbf{z})^{\alpha_k}, \tag{7.4}$$

then we can show that the original distribution is the result of the basis transform by the softmax.

The Dirichlet in the softmax basis: We show that the density over $\boldsymbol{\pi}$ shown in Equation 7.4 transforms into the Dirichlet over $\mathbf{z}$. First, we consider the special case where $\boldsymbol{\pi}$ is confined to a $I-1$ dimensional subspace satisfying $\sum_i \boldsymbol{\pi}_i = c$. In this subspace we can represent $\boldsymbol{\pi}$ by an $I-1$ dimensional vector $\mathbf{a}$ such that

$$\pi_i = a_i \quad i,...,I-1 \tag{7.5}$$

$$\pi_I = c - \sum_i^{I-1} a_i \tag{7.6}$$

and similarly we can represent $\mathbf{z}$ by an $I-1$ dimensional vector ϱ:

$$z_i = \varrho_i \quad i,...,I-1 \tag{7.7}$$

$$z_I = 1 - \sum_i^{I-1} \varrho_i \tag{7.8}$$

then we can find the density over ϱ (which is proportional to the required density over z) from the density over $\boldsymbol{\pi}$ (which is proportional to the given density over $\boldsymbol{\pi}$) by finding the determinant of the $(I-1)\times(I-1)$ Jacobian $\mathbf{J}$ given by

$$
\begin{aligned}
J_{ik} = \frac{\partial \xi_i}{\partial a_i} &= \sum_{j}^{I} \frac{\partial x_i}{\partial \pi_j} \frac{\partial \pi_j}{\partial a_k} \\
&= \delta_{ik}\mathbf{x}_i - \mathbf{x}_i\mathbf{x}_k + \mathbf{x}_i\mathbf{x}_I = \mathbf{x}_i(\delta_{ik} - (\mathbf{x}_k - \mathbf{x}_I)) \qquad (7.9)
\end{aligned}
$$

We define two additional $I-1$ dimensional helper vectors $\mathbf{x}_k^+ := \mathbf{x}_k - \mathbf{x}_I$ and $n_k := 1$, and use $\det(I - xy^T) = 1 - x \cdot y$ from linear algebra. It follows that

$$
\begin{aligned}
\det J &= \prod_{i=1}^{I-1} \mathbf{x}_i \times \det[I - n\mathbf{x}^{+^T}] \\
&= \prod_{i=1}^{I-1} \mathbf{x}_i \times (1 - n \cdot \mathbf{x}^+) \qquad (7.10) \\
&= \prod_{i=1}^{I-1} \mathbf{x}_i \times \left(1 - \sum_k \mathbf{x}_k^+\right) = I \prod_{i=1}^{I} \mathbf{x}_i
\end{aligned}
$$

Therefore, using Equation 7.4 we find that

$$
P(\mathbf{z}) = \frac{P(\boldsymbol{\pi})}{|\det \mathbf{J}|} \propto \prod_{i=1}^{I} \mathbf{z}_i^{\alpha_i - 1} \qquad (7.11)
$$

This result is true for any constant c since it can be put into the normalizing constant. Thereby we make sure that the integral of the distribution is 1 and we have a valid probability distribution.

Appendix C: Inverting the Laplace Approximation of the Dirichlet

Note that the following section is a copy of Hennig (2010) derivation. We don't claim any new contribution but merely want to give an overview of the content.

For a given $\mathbf{z}$, all $\mathbf{z}'$ satisfying $\mathbf{z}' = \mathbf{z} + c\mathbf{1}$ share the same value $\sigma(\mathbf{z}')$ for any $c \in \mathbb{R}$ with $\mathbf{1} = [1, 1, ..., 1]^\top$. Since the Laplace Bridge is a map between a Gaussian and a Dirichlet distribution we must ensure that the Dirichlet is in fact a distribution, i.e. that multiple values don't map to the same result. To solve this ambiguity we introduce a soft constrain $r = \exp[-\frac{\tau}{2}(\mathbf{1}^\top \mathbf{z})^2]$ which can be interpreted as a soft projection of the subspaces forming parallel lines to $\mathbf{1}$ onto their intersection with the hyperplane defined by $\mathbf{1}^\top \mathbf{z} = 0$.

When $r \to \infty$, where the constraint becomes a Dirac distribution, we can again use a reformulation of the parameter space, using $K-1$ parameters $\mathbf{a}$ defined through

$$z_k = \begin{cases} a_k & \text{if } k = 1, 2, ..., K-1 \\ -\sum_{k=1}^{K-1} a_k & \text{if } k = K \end{cases} \tag{7.12}$$

Through the figures of the 1D Dirichlet approximation in the main thesis, we have already established that the mode of the Dirichlet lies at the mean of the Gaussian distribution and therefore $\boldsymbol{\pi}(\mathbf{y}) = \frac{\alpha}{\sum_i \alpha_i}$. Additionally, the elements of $\mathbf{y}$ must sum to zero. These two constraints combined yield only one possible solution for $\boldsymbol{\mu}$.

$$\mu_k = \log \alpha_k - \frac{1}{K} \sum_{l=1}^{K} \log \alpha_l \tag{7.13}$$

Calculating the covariance matrix $\boldsymbol{\Sigma}$ is more complicated but layed out in the following. The logarithm of the Dirichlet is, up to additive constants

$$\log p_y(y|\alpha) = \sum_k \alpha_k \pi_k - \frac{\tau}{2} \mathbf{1}^\top \mathbf{z} \tag{7.14}$$

Using π_k as the softmax of $\mathbf{y}$ as shown in Equation 7.3 we can find the elements of the Hessian $\mathbf{L}$

$$L_{kl} = \hat{\alpha}(\delta_{kl} \hat{\pi}_k - \hat{\pi}_k \hat{\pi}_l) + \tau(\mathbf{1}\mathbf{1}^\top)_{kl} \tag{7.15}$$

where $\hat{\boldsymbol{\alpha}} := \sum_k \alpha_k$ and $\hat{\boldsymbol{\pi}} = \frac{\alpha_k}{\hat{\alpha}}$ for the value of $\boldsymbol{\pi}$ at the mode. The term $(\mathbf{1}\mathbf{1}^\top)_{kl}$ is a convoluted way of writing a one that makes the following math easier to parse.

To analytically invert $\mathbf{L}$ we introduce a rectangular matrix $\mathbf{X} \in \mathbb{R}^{K \times 2}$ with elements

$$X_{ku} = \hat{\pi}_k \delta_{1u} + \mathbf{1}_k \delta_{2u} = \begin{pmatrix} \hat{\pi}_1 & 1 \\ \vdots & \vdots \\ \hat{\pi}_K & 1 \end{pmatrix} \tag{7.16}$$

and the square matrices $\mathbf{A} \in \mathbb{R}^{K \times K}$ and $\mathbf{B} \in \mathbb{R}^{2 \times 2}$ with

$$\mathbf{A} = \mathrm{diag}(\alpha) \qquad \text{and} \qquad \mathbf{B} = \begin{pmatrix} \hat{\alpha} & 0 \\ 0 & \tau \end{pmatrix} \tag{7.17}$$

which allows us to write

$$\mathbf{L} = \mathbf{A} + \mathbf{X}\mathbf{B}\mathbf{X}^\top \tag{7.18}$$

Both **A** and **B** are diagonal with strictly positive diagonal elements and thus invertible. Therefore we can use the *matrix inversion lemma*, which states

$$(\mathbf{A} + \mathbf{X}\mathbf{B}\mathbf{X}^\top)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{X}(\mathbf{B}^{-1} + \mathbf{X}^\top\mathbf{A}^{-1}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{A}^{-1} \tag{7.19}$$

The 2×2 expression in brackets is known as the *Schur complement* and we can compute it with

$$\begin{aligned}
(\mathbf{B}^{-1} + \mathbf{X}^\top\mathbf{A}^{-1}\mathbf{X})_{ij} &= \mathbf{B}^{-1}_{ij} + \left(\frac{\alpha_k}{\hat{\alpha}}\delta_{i1} + n_k\delta_{i2}\right)\frac{1}{\alpha_k}\delta_{kl}\left(\frac{\alpha_l}{\hat{\alpha}}\delta_{j1} + n_l\delta_{j2}\right) && (7.20)\\
&= \mathbf{B}^{-1}_{ij} + \frac{1}{\hat{\alpha}}\delta_{i1}\delta_{j1} + \frac{D}{\hat{\alpha}}(\delta_{i1}\delta_{j2} + \delta_{i2}\delta_{j1}) + \delta_{i2}\delta_{j2}\sum_k \frac{1}{\alpha_k} && (7.21)\\
\mathbf{B}^{-1} + \mathbf{X}^\top\mathbf{A}^{-1}\mathbf{X} &= \begin{pmatrix} 0 & K/\hat{\alpha} \\ K/\hat{\alpha} & \tau^{-1} + \sum_k \alpha_k^{-1} \end{pmatrix} && (7.22)
\end{aligned}$$

The inverse of a 2×2 matrix is

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{7.23}$$

so we get the inverse of the Schur compliment (which exists for α, τ with $\alpha_k > 0$ and $\tau > 0$)

$$(\mathbf{B}^{-1} + \mathbf{X}^\top\mathbf{A}^{-1}\mathbf{X})^{-1} = \begin{pmatrix} -\frac{\hat{\alpha}^2}{K}\left(\frac{1}{\tau} + \sum_k \frac{1}{\alpha_k}\right) & \frac{\hat{\alpha}}{K} \\ \frac{\hat{\alpha}}{K} & 0 \end{pmatrix} \tag{7.24}$$

we can now project this back to $\mathbb{R}^{K \times K}$ and get

$$L_{kl}^{-1} = \delta_{kl}\frac{1}{\alpha_k} - \frac{1}{K}\left[\frac{1}{\alpha_k} + \frac{1}{\alpha_l} - \frac{1}{K}\left(\frac{1}{\tau} + \sum_u^K \frac{1}{\alpha_u}\right)\right] \tag{7.25}$$

because the inverse is defined for all positive values of τ, we can now safely take the limit of $\tau \to \infty$, which hardens the constraint on the subspace $\mathbf{1}^\top \mathbf{z}$.

We are mostly interested in the diagonal elements since we desire a sparse encoding for computational reasons and we otherwise needed to map a $K \times K$ covariance matrix to a $K \times 1$ Dirichlet parameter vector which would be a very overdetermined mapping. Note that K is a scalar, not a matrix. The diagonal elements of $\boldsymbol{\Sigma} = \mathbf{L}^{-1}$ can be calculated as

$$\Sigma_{kk} = \frac{1}{\alpha_k}\left(1 - \frac{2}{K}\right) + \frac{1}{K^2}\sum_l^k \frac{1}{\alpha_l}. \tag{7.26}$$

To invert this mapping we transform Equation 7.13 to

$$\alpha_k = e^{\mu_k}\prod_l^K \alpha_l^{1/K} \tag{7.27}$$

by applying the logarithm and re-ordering some parts. Inserting this into Equation 7.26 and re-arranging yields

$$\prod_l^K \alpha_l^{1/K} = \frac{1}{\boldsymbol{\Sigma}_{kk}}\left[e^{-\mu}\left(1 - \frac{2}{K}\right) + \frac{1}{K^2}\sum_u^K e^{-\mu_u}\right] \tag{7.28}$$

which can be re-inserted into Equation 7.27 to give

$$\alpha_k = \frac{1}{\Sigma_k k}\left(1 - \frac{2}{K} + \frac{e^{-\mu_k}}{K^2}\sum_l^K e^{-\mu_k}\right) \tag{7.29}$$

which is the final mapping. With Equations 7.13 and 7.26 we are able to map from Dirichlet to Gaussian and with Equation 7.29 we are able to map the inverse direction.

Appendix D: Experiments Details

The exact experimental setups, i.e. network architectures, learning rates, random seeds, etc. can be found in the accompanying GitHub repository[1]. This section is mostly used to justify some of the decisions we made during the process in more detail and highlight some miscellaneous interesting things.

Uncertainty estimates on MNIST

Most of the experimental setup is already explained in the main paper. The exact details can be found in the accompanying code. Every experiment has been conducted with 5 different seeds.

OOD detection

Every experiment has been conducted with 5 different seeds. In the tables, the mean and standard deviations are presented. The reason why the sampling procedure for the CIFAR-10 and CIFAR-100 case are similarly fast even though we draw from a 10- vs 100-dimensional Gaussian is because the sampling procedures were parallelized on a GPU. All prior uncertainties over the weights were chosen such that the MMC of the sampling averages was around 5% lower than the MAP estimate. In the following, we show the results including a KFAC approximation of the last layer.

Time comparison

Every experiment has been conducted with 5 different seeds. The presented curves are the averages over these 5 experiments with error bars. The reason why taking one sample is slower than two is because of the way random numbers are generated for the normal distribution. For further information read up on the Box-Mueller Transform.

Uncertainty-aware output ranking on ImageNet

The prior covariances for the Laplace approximation of the Hessian over the weights were chosen such that uncertainty estimate of the Laplace bridge MMC over the outputs was not more than 5% lower than the MAP estimate. The length of the list generated by our uncertainty aware method was chosen such that it contained at least one and maximally ten samples. Originally we wanted to choose the maximal length according

[1] https://github.com/mariushobbhahn/master2020/tree/master/2019-10-Laplace_Bridge

		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**

Table 7.2: Out-of-distribution detection results. A network has been trained on the data set in the **train** column and is tested on the **test** column. Optimally, the MMC for out of distribution data is low and the AUROC is high. There is no clear winner when it comes to discriminating in and OOD w.r.t. both metrics. However, the Laplace Bridge is around 400 times faster on average. Time is measured in seconds. Five runs with different seeds per experiment were conducted. 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.

		Sampling (100)			Dirichlet mode		
Train	Test	MMC	AUROC	Time	MMC	AUROC	Time
MNIST	MNIST	0.981 ± 0.000	-	109.3	**0.987** ± 0.001	-	**0.016**
MNIST	FMNIST	0.482 ± 0.002	0.991 ± 0.000	109.3	**0.377** ± 0.019	**0.994** ± 0.002	**0.016**
MNIST	notMNIST	0.643 ± 0.002	0.960 ± 0.001	44.7	**0.630** ± 0.018	**0.962** ± 0.007	**0.029**
MNIST	KMNIST	**0.617** ± 0.003	**0.976** ± 0.001	109.5	0.630 ± 0.018	0.975 ± 0.004	**0.016**

Table 7.3: Results for sampling from all weights instead of the last layer. Number of samples was 100. Time is measured in seconds.

to the size of the largest category (e.g. fishes or dogs) but the class tree hierarchy of ImageNet does not answer this question meaningfully. We chose ten because there are no reasonable bins larger than ten when looking at a histogram.

Bibliography

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

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

Blundell Charles, Cornebise Julien, Kavukcuoglu Koray, Wierstra Daan. Weight Uncertainty in Neural Networks // ArXiv. 2015.

Braun Michael, McAuliffe Jon. Variational inference for large-scale models of discrete choice // Journal of the American Statistical Association. 2010. 105, 489. 324–335.

Brosse Nicolas, Riquelme Carlos, Martin Alice, Gelly Sylvain, Moulines Éric. On Last-Layer Algorithms for Classification: Decoupling Representation from Uncertainty Estimation // arXiv preprint arXiv:2001.08049. 2020.

notMNIST dataset. // . 2011.

Clanuwat Tarin, Bober-Irizar Mikel, Kitamoto Asanobu, Lamb Alex, Yamamoto Kazuaki, Ha David. Deep Learning for Classical Japanese Literature // CoRR. 2018. abs/1812.01718.

Dangel Felix, Kunstner Frederik, Hennig Philipp. BackPACK: Packing more into backprop // arXiv preprint arXiv:1912.10985. 2019.

Graves Alex. Practical Variational Inference for Neural Networks // Advances in Neural Information Processing Systems 24. 2011. 2348–2356.

Haussmann Manuel, Gerwinn Sebastian, Kandemir Melih. Bayesian Evidential Deep Learning with PAC Regularization. 2019.

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

Hein Matthias, Andriushchenko Maksym, Bitterwolf Julian. Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2019.

Hendrycks Dan, Gimpel Kevin. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks // CoRR. 2016. abs/1610.02136.

Hennig P. Approximate Inference in Graphical Models. XI 2010.

Hennig P., Stern D., Herbrich R., Graepel T. Kernel Topic Models // Fifteenth International Conference on Artificial Intelligence and Statistics. 22. 2012. 511–519. (JMLR Proceedings).

Learning Multiple Layers of Features from Tiny Images. // . 2009.

MNIST handwritten digit database. // . 2010.

Louizos Christos, Welling Max. Structured and efficient variational deep learning with matrix gaussian posteriors // ICML. 2016.

MacKay David J. C. A Practical Bayesian Framework for Backpropagation Networks // Neural Comput. V 1992a. 4, 3. 448–472.

MacKay David JC. The evidence framework applied to classification networks // Neural computation. 1992b. 4, 5. 720–736.

MacKay David J.C. Choice of Basis for Laplace Approximation // Machine Learning. Oct 1998. 33, 1. 77–86.

Mackay David J C. Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks // Network: Computation in Neural Systems. 1995. 6, 3. 469–505.

Malinin Andrey, Gales Mark. Predictive uncertainty estimation via prior networks // Advances in Neural Information Processing Systems. 2018. 7047–7058.

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

Martens James, Grosse Roger. Optimizing neural networks with Kronecker-factored approximate curvature // ICML. 2015.

Masson Michael E. J. A tutorial on a practical Bayesian alternative to null-hypothesis significance testing // Behavior Research Methods. Sep 2011. 43, 3. 679–690.

McAllister Rowan, Gal Yarin, Kendall Alex, Wilk Mark van der, Shah Amar, Cipolla Roberto, Weller Adrian. Concrete Problems for Autonomous Vehicle Safety: Advantages of Bayesian Deep Learning // IJCAI. 2017.

Michelmore Rhiannon, Kwiatkowska Marta, Gal Yarin. Evaluating Uncertainty Quantification in End-to-End Autonomous Driving Control // CoRR. 2018. abs/1811.06817.

Netzer Yuval, Wang Tao, Coates Adam, Bissacco Alessandro, Wu Bo, Ng Andrew Y. Reading

Digits in Natural Images with Unsupervised Feature Learning // NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011. 2011.

Nguyen Anh, Yosinski Jason, Clune Jeff. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images // CVPR. 2015.

Nickerson Raymond S. Null hypothesis significance testing: a review of an old and continuing controversy. // Psychological methods. 2000. 5, 2. 241.

Ritter Hippolyt, Botev Aleksandar, Barber David. A Scalable Laplace Approximation for Neural Networks // International Conference on Learning Representations. 2018.

Russakovsky Olga, Deng Jia, Su Hao, Krause Jonathan, Satheesh Sanjeev, Ma Sean, Huang Zhiheng, Karpathy Andrej, Khosla Aditya, Bernstein Michael S., Berg Alexander C., Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge // CoRR. 2014. abs/1409.0575.

Sensoy Murat, Kaplan Lance, Kandemir Melih. Evidential deep learning to quantify classification uncertainty // Advances in Neural Information Processing Systems. 2018. 3179–3189.

Snoek Jasper, Rippel Oren, Swersky Kevin, Kiros Ryan, Satish Nadathur, Sundaram Narayanan, Patwary Mostofa, Prabhat Mr, Adams Ryan. Scalable Bayesian Optimization Using Deep Neural Networks // Proceedings of the 32nd International Conference on Machine Learning. 37. Lille, France: PMLR, 07–09 Jul 2015. 2171–2180. (Proceedings of Machine Learning Research).

Spiegelhalter David J, Lauritzen Steffen L. Sequential updating of conditional probabilities on directed graphical structures // Networks. 1990. 20, 5. 579–605.

Sun Shengyang, Chen Changyou, Carin Lawrence. Learning structured weight uncertainty in Bayesian neural networks // Artificial Intelligence and Statistics. 2017. 1283–1292.

Titsias Michalis. One-vs-each approximation to softmax for scalable estimation of probabilities // NIPS. 2016.

Wilson Andrew Gordon, Hu Zhiting, Salakhutdinov Ruslan, Xing Eric P. Deep Kernel Learning // Proceedings of the 19th International Conference on Artificial Intelligence and Statistics. 51. Cadiz, Spain: PMLR, 09–11 May 2016. 370–378. (Proceedings of Machine Learning Research).

Wu Anqi, Nowozin Sebastian, Meeds Edward, Turner Richard E., Hernández-Lobato José Miguel, Gaunt Alexander L. Fixing Variational Bayes: Deterministic Variational Inference for Bayesian Neural Networks // CoRR. 2018. abs/1810.03958.

Xiao Han, Rasul Kashif, Vollgraf Roland. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms // CoRR. 2017. abs/1708.07747.