Well-Calibrated Bayesian Neural Networks

On the empirical assessment of calibration and construction of well-calibrated Neural Networks

Jonathan Heek

Supervisor: T. Adel

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy

Christ's College

August 2018

I would like to dedicate this thesis to my parents and my loving partner.

Declaration

I, Jonathan Heek of Christ's College, being a candidate for the MPhil in Machine Learning, Speech and Language Technology, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose. This document contains 13,104 words excluding abstract, captions, and bibliography.

Jonathan Heek
August 2018

Acknowledgements

I would like to acknowledge my supervisor Tameem Adel for introducing me to the topic of calibration. I would also like to thank Yingzhen Li for explaining to me in great detail the framework of α-divergences.

Abstract

A well-calibrated model guarantees that predicted probabilities closely match observed frequencies. This report uses the calibration framework to assess the quality of uncertainty estimates in Bayesian Neural Networks. Exact Bayesian Inference is intractable for these models. Therefore, Variational Inference and MCMC methods are considered as approximate inference methods. We find that these methods only yield well-calibrated models when the hyper-parameters are explicitly optimised for a calibration test rather than test accuracy. Furthermore, this report introduces two novel algorithms for performing Dropout Variational Inference with a well-defined Evidence Lower Bound and a new stochastic acceptance test with adjustable bias/efficiency tradeoff. We further observe that using Bayesian compression methods during training improves calibration as well as test accuracy.

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

$\mathscr{B}$	Bernoulli distribution
$\mathbb{E}[X]$	Expectation of the random variable X
$\mathscr{G}(\alpha, \beta)$	Gamma distribution with shape $\alpha > 0$ and rate $\beta > 0$
$\mathscr{N}(\mu, \Sigma)$	Normal distribution with mean μ and covariance Σ
$P(.)$	Probability mass function
$p(.)$	Probability density function
$\mathrm{Var}[X]$	Variance of the random variable X

Greek Symbols

$\delta(.)$	Dirac delta function

Other Symbols

$x \odot y$	The element-wise product of two matrices

Acronyms / Abbreviations

BBB	Bayes By Backprop
CLT	Central Limit Theorem
SOTA	State Of The Art
ELBO	Evidence Lower BOund
HMC	Hamiltonian Monte Carlo

MCMC	Markov Chain Monte Carlo
SGD	Stochastic Gradient Descent/Ascent
SGHMC	Stochastic Gradient Hamilton Monte Carlo
SGLD	Stochastic Gradient Langevin Dynamics
VI	Variational Inference

Chapter 1

Calibration in Bayesian statistics

1.1 Introduction

The recent success of Deep Learning approaches has significantly increased the ability of Machine Learning practitioners to perform difficult tasks using complex models which encode little prior information about the task at hand. Research in Neural Networks has predominantly focused on competing on a human level in tasks that were previously thought to be impossible for a machine. For example, by beating the world champion in the game of Go [61], transferring the style of an image [16], or learning to drive a car [32].

By now it is evident that Neural Networks perform on-par or even outperform humans in many tasks. Still many real-life applications require more than a model which predicts the most likely outcome. This becomes clear in critical domains like self driving cars or medical diagnostics where poor predictions can result in significant losses. These applications also require robust and reliable predictions with strong guarantees about the model's uncertainty [31].

The importance of using probabilities to express the uncertainty of a prediction has been widely recognised in both classical statistics [10, 49] and in modern Machine Learning [14, 20, 22, 41]. Contemporary research nonetheless lacks a widely recognised benchmark for the quality of predictive uncertainty.

Instead the quality of the uncertainty estimates is instead determined based on theoretical grounds [23, 33, 42]; the model's ability to generalise to test data or out-of-distribution examples [7, 15, 19, 39]; and assigning low probability to adversarial examples [41]. Although all these justifications are important, none of these approaches provide direct evidence for the quality of uncertainty estimates.

This report formalises the quality of uncertainty estimates by using the framework of calibration [10]. The predictions of a model are said to be well-calibrated if the expectation over any random variable derived from these predictions matches the observed long-term average. Sec. 1.3 will formalise this definition.

Calibration offers a powerful method for testing the quality of uncertainty estimates because well-calibrated predictions can be interpreted as objective probabilities. In the Bayesian framework a prediction must be considered subjective in the sense that predictions depend on prior assumptions about the behaviour of a random process. Objective probability follows the Frequentist interpretation of probability which states the probability of a random event corresponds to its long term frequency. Sec. 1.3.1 will demonstrate how subjective (Bayesian) and objective (Frequentist) probabilities can be related through the calibration theorem which states that any subjective probability model must consider itself to be well-calibrated.

Calibration has strong connections with other important topics in Machine Learning. Recent work on Adversarial Examples shows that for neural classification models there is a large subset of the input space in which the model makes false predictions with high confidence [18]. The fact that these mistakes are made with high confidence can be seen as a type of miscalibration. Another related subfield is AI Fairness which aims to reduce biases against subpopulations in the data. Calibration can serve as a definition of fairness [37] and is also a condition for a popular fair classification algorithm [54]. Furthermore, a model with good uncertainty estimates can be used to automatically determine a good tradeoff between exploration and exploitation in Reinforcement Learning by means of Thompson sampling [7]. Current VI and sampling methods are unable to outperform simple linear approaches in Thompson Sampling based RL [57]. This suggests the uncertainty estimates are not of sufficient quality.

Previous work has already shown that modern Neural Networks are poorly calibrated [20]. However, these results are limited to Neural Networks where the model parameters are inferred using MAP estimate. This report aims to give a more detailed overview focussing on Bayesian methods. Bayesian Inference on Neural Network models can be performed using MCMC methods [1] and Variational Inference methods [25]. These approaches promise better uncertainty estimates by approximating the posterior distribution of the model. Nonetheless, the effects the approximations have on the calibration properties of the resulting posterior remain largely unspecified.

In the first place, this report motivates the use of calibration as a measure for the quality of uncertainty estimates. Secondly, Chap. 2 and Chap. 4 respectively give a theoretical and empirical analysis of the properties of commonly used VI and sampling methods. We find that optimal generalisation does not always coincide with good uncertainty estimates and significant hyper-parameter tuning is needed to make the tested approximate inference methods well-calibrated. Additionally, we find significant differences in the ability of these methods to scale from simple toy problems to more complex tasks like MNIST and FashionMNIST.

This report describes a number of novel algorithms designed to obtain well-calibrated Bayesian Neural Networks: Sec. 2.5 describes a way to combine Gaussian Dropout with Bayes By Backprop to obtain a proper VI method with some degree of weight correlation; Sec. 2.6.2 describes variational distribution that learns the scale of weights with a separate variational variable which induces sparsity and improves performance; and finally Sec. 2.7.2 introduces a algorithm for stochastic acceptance of MCMC proposals with an adaptive bias/variance tradeoff. Furthermore, we observe an interesting connection between compression and calibration. Sec. 2.6.3 describes how Variational Inference can be used to prune weights and features from a network, and how this can improve both accuracy and calibration.

1.2 Prediction based decisions

The quality of a model's predictions is often assessed by analysing the outcome of some utility function over a test set. The utility function $U(x, a)$ defines the utility of action/decision a given that the outcome of the random variable $X = x$. The optimal decision $\hat{a}$ is found by maximising the expected utility which is known as Bayes decision rule

$$\hat{a} = \arg\max_{a} \mathbb{E}_{p(x)} \left[U(x, a) \right]. \tag{1.1}$$

Thus, given a distribution $p(x)$ the optimal decision w.r.t. some utility function $U(x, a)$ can be computed.

Consider for example a classification problem where the utility of a prediction over classes $\omega_k \in \{\omega_1, \ldots, \omega_K\}$ is $U(\omega, \hat{\omega}) = 1$ if $\omega = \hat{\omega}$ and $U(\omega, \hat{\omega}) = 0$ otherwise. This function is known as the zero-one utility. Using Bayes decision rule, the expected utility is maximised for $\hat{\omega} = \omega^*$ the class with the largest probability. In this case the average utility for the unobserved test dataset coincides with the classification rate.

The value of a model's predictions depends on the utility function that is appropriate for the application. Using a utility based benchmark gives only partial information about the general quality of the model predictions. For example, using the zero-one utility function, or equivalently the classification rate, limits the analysis to the predictive power of the model, because the expected utility only depends on the certainty in the maximum likelihood class.

Consider for example a character recognition model. When such a model is used to index the textual context of a large corpus of images a zero-one utility function might be justified.

However, when the same character recognition model is used to read a bank check or a doctor's prescription one would certainly want to avoid a case where a model takes action without the appropriate degree of confidence in its predictions.

An appropriate utility for a task quantifies both the reward of correct predictions and the risk associated with misprediction. Maximising an arbitrary utility function does not only depend on a model's predictive power (low entropy) but also its ability to correctly estimate the uncertainty in a prediction (well-calibrated). Once we have verified that a model's predictions are well-calibrated and have low entropy, we can be confident that the model will be useful and reliable for arbitrary decision making tasks depending on the predicted variables.

1.3 Measuring calibration

This section formally defines a calibration measure based on [10], generalised to the case of random variables instead of random events. Consider the set of all random variables $\mathscr{B}_t$ observed up to time $t \in [1, \infty)$. Such that $\mathscr{B}_t \subseteq \mathscr{B}_{t+1}$ and a subjective probability distribution over all random variables $P(\mathscr{B}_\infty)$.

An expectation $\tilde{X}_t$ is formed at $t-1$ about a random variable $X_t \in \mathscr{B}_t$ observed at time t based on all previously observed information $\tilde{X}_t = E_{P(X_t|\mathscr{B}_{t-1})}(X_t)$. Consider a selection of time instances where $\xi_t = 1$ if the time instance t is selected and $\xi_t = 0$ otherwise. Let

$$\nu_t = \sum_{i=1}^{t} \xi_i, \quad p_t = \frac{1}{\nu_t} \sum_{i=1}^{t} \xi_i X_i, \quad \pi_t = \frac{1}{\nu_t} \sum_{i=1}^{t} \xi_i \tilde{X}_i. \tag{1.2}$$

The subjective distribution over the random variables $\{X_t | \xi_t = 1\}$ is said be calibrated if

$$\lim_{t\to\infty} p_t - \pi_t = \lim_{t\to\infty} \frac{1}{\nu_t} \sum_{i=1}^{t} \xi_i \left(X_i - \tilde{X}_i \right) = 0, \ \text{ if } \lim_{t\to\infty} \nu_t = \infty. \tag{1.3}$$

These random variables are measurable at time t if $\xi_t \in \mathscr{B}_t$. Because ξ is a constructed variable it must be sampled as $\xi_t \sim P(\xi_t|\mathscr{B}_{t-1})$ to be measurable at time t. This implies

$$\xi_i = P(\xi_t|\mathscr{B}_{t-1}) = f_i(\mathscr{B}_{t-1}). \tag{1.4}$$

The constraint on ξ ensures that the calibration test cannot depend on the outcome of the random variable. Otherwise it would be trivial to construct a failing calibration test for any model. For example, consider a Bernoulli model with $p = 0.5$ for the outcome of a fair coin toss. Let $\xi_t = 1$ for each case of heads $X_t = 1$. Clearly, the calibration condition (1.3) would not hold even for a fair coin, despite the fact the model describes the random process perfectly. Thus, only calibration measures with $\xi_t \sim P(\xi_t|\mathscr{B}_{t-1})$ and $v_t \to \infty$ are admissible.

The constraint $v_t \to \infty$ ensures that the selected sample is infinite large such that any random fluctuations in the observations are averaged out (due to finite variance). In practise, we cannot fulfil this constraint and we must resort to a hypothesis test to determine if the equality can be rejected with a certain degree of confidence.

1.3.1 The calibration theorem

Let $\beta_t = 1/v_t$ if $v_t > 0$ and $\beta_t = 0$ otherwise. Furthermore, let $Z_t = \beta_t\xi_t(X_t - \tilde{X}_t)$. Recall that $\tilde{X}_t = \mathbb{E}_{P(X_t|\mathscr{B}_{t-1})}(X_t)$ and therefore $E(Z_t|\mathscr{B}_{t-1})) = 0$. The variance of Z_t reduces to the second moment

$$\mathbb{E}\left[Z_t^2\right] = \mathbb{E}\left[(\beta_t\xi_t)^2\mathrm{Var}(X_i|\mathscr{B}_{t-1})\right] \leq C\mathbb{E}\left[(\beta_t\xi_t)^2\right], \tag{1.5}$$

where the $\mathrm{Var}(X_i|\mathscr{B}_{t-1})$ is assumed to have a finite upper bound C. Let $U_t = \sum_{i=1}^t Z_i$ which is a martingale because $\mathbb{E}(Z_t|\mathscr{B}_{t-1})) = 0$. Note that $\mathbb{E}[X_tX_{t+i}] = 0$ for $i \geq 1$. Therefore,

$$\mathbb{E}\left[U_t^2\right] = \sum_{i=1}^t \mathbb{E}(Z_i^2) \leq C\sum_{i=1}^t \mathbb{E}\left[(\beta_i\xi_i)^2\right]. \tag{1.6}$$

The non-zero elements of the sequence $(\beta_i\xi_i)^2$ are $\{1/1, 1/2^2, 1/3^2, \dots\}$. Consequently,

$$E\left[U_t^2\right] \leq C\sum_{n=1}^\infty \frac{1}{n^2} = C\frac{\pi^2}{6} \tag{1.7}$$

By the martingale convergence theorem [13, VII.8 Theorem 1], U_t converges almost surely. Kronecker's lemma implies $\beta_t \sum_{i=1}^t \xi_t(X_t - \tilde{X}_t) \to 0$ for every point where U_t converges [10]. Hence, the calibration condition (1.3) holds almost surely under the distribution $P(\mathscr{B}_\infty)$ if $\mathrm{Var}(X_i|\mathscr{B}_{t-1})$ has a finite upper bound and $v_t \to \infty$. The condition on the variance of X_i can be loosened to

$$\mathbb{E}\left[U_t^2\right] = \sum_{i=1}^t \mathbb{E}\left[(\beta_t\xi_t)^2\mathrm{var}(X_i|\mathscr{B}_{t-1})\right] < \infty. \tag{1.8}$$

The calibration theorem implies that a predictive distribution with finite variance over an arbitrary random will be perfectly calibrated for every admissible calibration test under the expectation of the model. When an admissible calibration test fails, it must be concluded that the model does not accurately reflect the true random process that generated the data.

Although simple in essence, the calibration theorem requires careful interpretation. The fact that a model expects itself to be well-calibrated provides no guarantees for the quality of uncertainty estimates. It also does not imply that probabilistic models are inherently calibrated. In fact, a deterministic model is just a special case of a probabilistic model for which all predictions have zero entropy. Even such a model is subjectively well-calibrated although it provides no useful information about uncertainty (unless it never makes an error).

The strength of the calibration theorem lies in the implication that no calibration is impossible to pass. Utility based benchmarks rarely have a known upper bound. For example, if it is assumed that there is no ambiguity in digit classification then any error rate greater than 0% on MNIST must be considered suboptimal. However, if there is some ambiguity in the labels then even the true model will not attain a perfect classification rate. Once classifiers get close to the point where ambiguity dominates the error rate, the risk of overfitting the test set increases. Recent work suggest that Neural Networks trained on popular public datasets are prone to overfitting the test set [55]. Calibration tests could help alleviate this issue which is otherwise inevitable for popular benchmarks. In the first place, by providing an additional metric for the quality of the model. And secondly, because as opposed to utility based benchmarks, calibration metrics always have a well-defined optimum.

The calibration theorem further implies that identifiable models are well-calibrated [10]. For identifiable models the posterior converges towards a delta at the true model parameters and the remaining uncertainty is due to the randomness of the data generating process. It is thus sufficient for a well-calibrated model to assign some prior probability to the true model parameters. For example, a Bernoulli model for a (possibly unfair) coin toss is well-calibrated as long as some prior probability is assigned to the true probability of tossing heads.

Chapter 2

Bayesian Neural Networks

This chapter gives an overview of the methods used to perform (approximate) Bayesian Inference on the parameters of a Neural Network. Sec. 2.1 shows how a Bayesian interpretation of stochastic optimisation and regularisation methods can partially explain how Neural Networks are able to generalise despite the abundance of poor local optima by doing approximate Bayesian Inference and sampling. Sec. 2.2 discusses Dropout Variational Inference methods which formalise the Bayesian interpretation of dropout.

Sec. 2.3 introduces the Bayes By Backprop algorithm which learns a mean-field Gaussian approximation to the posterior. Sec. 2.4 derives an alternative Dropout VI method using multiplicative Gaussian noise which allows for the amount of regularisation to be optimised. However, Gaussian dropout lacks has various technical issues. Sec. 2.5 proposes a new, well-defined Gaussian VI method which uses both multiplicative and additive Gaussian noise and can be interpreted as a generalisation of both Gaussian dropout and BBB. Sec. 2.6 introduces a novel VI method based on sparse priors and argues how compression can result in well-calibrated model.

Sec. 2.7 discusses posterior sampling using MCMC methods and Sec. 2.8 shows how samples from the posterior can be used to obtain an approximation to the posterior predictive.

2.1 Bayes by accident

Traditionally, SGD has been considered as an optimisation algorithm which converges to a local optimum under mild conditions and step size annealing [58]. However, guaranteed convergence does not explain how neural networks typically generalise well beyond the training set despite the abundance of poor local optima. First of all the optimiser might get stuck in one of the many saddle points or local optima that are far from the global optimum [9, 17]. Secondly, a sufficiently complex classification model can learn an arbitrary labelling for a finite dataset. Empirical evidence shows modern neural networks generalise well despite being powerful enough to learn arbitrary labelings [68].

We draw inspiration from both optimisation and Bayesian literature in order to make Bayesian Neural Networks well-calibrated. Research in Neural Net optimisation is primarily concerned with convergence speed and generalisation. From a Bayesian perspective we are primarily concerned with finding a good approximation of Bayesian inference. This section highlights some cases where these two approaches overlap.

2.1.1 Stochastic optimisation

Recent studies attempting to explain the tendency of SGD to converge towards a good local optimum have shown that neural networks do not simply memorise the training labels [40]. Furthermore, it has been observed that the loss surface around the optimum found using SGD is typically flat [26, 33]. The flatness of a local optimum $\hat{\theta}$ is defined as

$$F(\hat{\theta}) = \int_{\theta \in \Theta} \exp\left(\log p(X, \hat{\theta}) + (\theta - \hat{\theta})^T H (\theta - \hat{\theta})\right) d\theta = p(X, \hat{\theta})\,(2\pi)^{M/2}\sqrt{|H|}, \qquad (2.1)$$

where $H = \nabla\nabla^T \log p(X, \hat{\theta})$. Note that a prior $p(\theta)$ is essential to avoid ambiguity because any optimum can be made arbitrary sharp by rescaling or reparameterisation of θ [12]. The local flatness as defined in (2.1) is the approximate model evidence $p(X)$ for the posterior found when using a Laplace approximation at $\hat{\theta}$.

From the Bayesian perspective a flat optimum corresponds to a larger model evidence. Such a model should therefore be preferred over a model with a sharper optimum or equivalently less (local) evidence. A flat optimum is more robust in the sense that a small perturbation of the model parameters only has a small effect on the log-likelihood.

Empirical evidence shows that SGD prefers flat optima in particular when the batch size is small because of additional noise in the stochastic gradient [33]. When the step size ϵ is constant, SGD behaves as a Markov Chain that approximately samples the posterior around a local mode [45]. In the idealised case, the stochastic gradient noise is approximately Gaussian due to the CLT and the covariance is proportional to the Fisher Information $\mathcal{I}$, assuming that the posterior is approximately gaussian $P(\theta|X) \approx \mathcal{N}(\hat{\theta}, \mathcal{I}^{-1}/N)$[1] due to the Bernstein-von Mises theorem [1, 59]

[1] For simplicity it is assumed that $P(\theta|X) \propto P(X|\theta)$ or equivalently $P(\theta) \propto 1$.

$$\begin{aligned}\theta_t &= \theta_{t-1} + \epsilon\frac{N}{n}\nabla \log P(\tilde{X}_t, \theta_{t-1}) \\ &\approx \theta_{t-1} + \epsilon\nabla \log P(X, \theta_{t-1}) + \mathcal{N}\left(0, \epsilon^2\frac{N^2}{n}\mathcal{I}\right),\end{aligned} \tag{2.2}$$

where X_t is a mini-batch of size n and N is the size of the complete training set X. Note that $\nabla \log P(X, \theta_{t-1}) \approx N\mathcal{I}(\hat{\theta} - \theta_t)$. Using $(N\mathcal{I})^{-1}$ to precondition the optimiser yields

$$\begin{aligned}\theta_t &= \theta_{t-1} + \epsilon\frac{1}{n}\mathcal{I}^{-1}\nabla \log P(\tilde{X}_t, \theta_{t-1}) \\ &\approx \mathcal{N}\left(\theta_t; \hat{\theta}, \mathcal{I}^{-1}\right),\end{aligned} \tag{2.3}$$

where $\epsilon = n$ such that the sequence θ_t are independent samples from the approximate posterior. For an elaborate discussion on the stationary distribution of SGD with a constant step size see [45]. Sec. 2.7 will discuss how SGD can be extended such that it samples from the posterior in the general case, which allows for a Bayesian treatment of the model parameters.

In practice, SGD is used to obtain a point estimate of the model parameters despite its Bayesian interpretation. A small batch size and fixed learning rate means that a point estimate is not necessarily approximating a posterior mode and is instead likely to be a sample from a region of high probability or flat mode. However, SOTA neural network models typically still overfit the training data and are consequently strongly miscalibrated without additional regularisation [63].

2.1.2 Variational Inference

Traditional regularisation techniques like L2 regularisation correspond to defining a non-uniform prior over the model parameters. Therefore, these techniques are consist with the Bayesian treatment of uncertainty. In contrast, stochastic regularisation techniques like dropout [63], multiplicative/additive Gaussian noise [35, 7], and batch normalisation [28] cannot be encapsulated in the prior $p(\theta)$.

Stochastic regularisation injects noise into the output of hidden layers or the model parameters which is referred to as the local and global noise parameterisation, respectively. For example, dropout can be implemented by drawing Bernoulli distributed random numbers $\varepsilon_i \sim \mathcal{B}(\varepsilon_i; p)$ and multiplying with either the outputs of hidden units $z_i' = \varepsilon_i z_i$ (local parameterisation) or the row of weights that generated the output $W_i' = \varepsilon_i W_i$ (global parameterisation). The two methods are equivalent when the batch size is 1. However, for larger batch sizes the global method uses the same noise vector ε for each batch item. The local parameterisation therefore has lower variance [35].

ELBO

Consider approximating an intractable posterior $p(\theta|X)$ with the distribution $q_\phi(\theta)$. Furthermore, assume $q_\phi(\theta)$ can be reparameterised as $q_\phi(\theta) = p(\theta|\varepsilon, \phi)p(\varepsilon)$, where $p(\theta|\varepsilon, \phi) = \delta(\theta - g(\varepsilon, \phi))$. The objective $\mathscr{L}(\phi)$ is to minimise the KL-divergence between $p(\theta|X)$ and $q_\phi(\theta)$:

$$\mathscr{L}(\phi) = \mathrm{KL}[q_\phi(\theta)||p(\theta|x)] = \mathbb{E}_{q_\phi(\theta)}\left[\log \frac{q_\phi(\theta)}{p(X, \theta)}\right] + C. \tag{2.4}$$

Stochastic optimisation requires an unbiased estimator for $\nabla\mathscr{L}(\phi)$. However, the expectation depends of ϕ, hence the derivative operator cannot be moved into the expectation naively [7]. Using the reparameterisation in terms of ε yields

$$\mathscr{L}(\phi) = \mathbb{E}_{p(\varepsilon)}\left[\log \frac{q_\phi(\theta = g(\varepsilon, \phi))}{p(X, \theta = g(\varepsilon, \phi))}\right]. \tag{2.5}$$

In this form the derivative can be moved inside the expectation and a mini-batch can be used because the expectation is linear in the data. Thus, the unbiased estimator for the gradient is

$$\nabla\mathscr{L}(\phi) = \frac{1}{n}\sum_{i=1}^{n} \nabla_\phi \left[\log q_\phi(\theta = g(\varepsilon_i, \phi)) - N \log p(X_i, \theta = g(\varepsilon_i, \phi))\right], \tag{2.6}$$

where X_i is a randomly sampled training example and $\varepsilon_i \sim p(\varepsilon)$. If the KL divergence w.r.t. the prior can be computed analytically an estimator with lower variance is

$$\begin{aligned}\nabla\mathscr{L}(\phi) = \frac{1}{n}\sum_{i=1}^{n} \nabla_\phi &\left[\log q_\phi(\theta = g(\varepsilon_i, \phi)) - N \log p(X_i|\theta = g(\varepsilon_i, \phi))\right] \\ &+ \nabla\mathrm{KL}[q_\phi(\theta)||p(\theta)].\end{aligned} \tag{2.7}$$

Variational Inference for Neural Networks using global reparameterisation was first proposed using a factorised gaussian distribution [7]. A similar algorithm without reparameterisation was introduced in [19] by using the characteristic function of the Gaussian distribution to derive unbiased estimators. Reparameterisation based estimators of the gradient are found to have significantly lower variance in practise [14].

Minimising the KL of the variational distribution w.r.t. the posterior is equivalent to maximising the Evidence Lower Bound (ELBO) [36]

$$\log p(X) \geq \mathbb{E}_{q_\phi(\theta)}[\log p(X|\theta)] - \mathrm{KL}[q_\phi(\theta)||p(\theta)]. \tag{2.8}$$

Based on the asymmetry of the KL divergences it has been argued that the reverse KL that is minimised in VI only covers a local mode rather than approximating the entire posterior [23]. Because the approximation of the posterior concentrates around a mode it is likely to underestimate uncertainty [65]. However, the experiments in Chap. 4 will demonstrate that for Bayesian Neural Networks the ELBO objective actually tends to significant underfit the data.

Note that the summation in (2.7) is the same as the gradient of the maximum likelihood objective if $q_\phi(\theta)$ is a constant in terms of ϕ. Furthermore, the prior can be chosen in such a way that $\nabla_\phi \mathrm{KL}[q_\phi(\theta)||p(\theta)]$ coincides with the gradient of a traditional regularisation term like L2. Therefore, stochastic maximum likelihood optimisation with L2 normalisation and dropout can be interpreted as a form of Variational Inference [15]. Dropout Variational Inference does not require additional parameters like Gaussian Variational Inference and is able to encode some of the covariance in the weights of a single feature[2].

2.2 Dropout Variational Inference

Consider the form of (2.7) for Dropout Variational Inference [14]. The approximate posterior $q_\phi(\theta)$ is defined as

$$\varepsilon_i \sim \mathscr{B}(\varepsilon_i; 1 - p_{drop,i}), \;\; p(\theta_i|\varepsilon, \phi) = \delta(\theta_i - \varepsilon_i \phi_i), \tag{2.9}$$

where θ_i is the set of weights such that the output of a hidden unit activation before dropout $z_i = x_i^T \theta_i$. For a Bayesian Neural Network with a prior $p(\theta) = \prod p(\theta_i)$, the gradient of the KL term[3] in (2.7) reduces to

[2] A feature either means a convolutional filter for CNNs or a hidden unit in a fully connected layer.

[3] The KL divergence is ill-defined because the variational distribution consists of point masses and the prior is a continuous distribution. In [14] it is shown that the gradient can be computed by defining the Bernoulli distribution as $p(\epsilon) = p\mathcal{N}(0, \sigma^2) + (1 - p)\mathcal{N}(1, \sigma^2)$ with $\sigma^2 \to 0$. However, this implicitly assumes that we may take the gradient of a diverging limit.

$$\begin{aligned}\nabla \mathrm{KL}[q_\phi(\theta)||p(\theta)] &= -\sum_i \nabla \mathbb{E}_{q_\phi(\theta_i)}\left[\log p(\theta)\right] \\ &= -\sum_i \nabla \mathbb{E}_{p(\epsilon_i)}[\log p(\epsilon_i \phi_i)] \\ &= -(1 - p_{drop,i}) \sum_i \nabla \log p(\phi_i).\end{aligned} \tag{2.10}$$

From (2.10) it is trivial to find the the prior of a typical regularisation term and vice versa. For example, a L2 weight penalty $\alpha \left\|\theta\right\|_2^2 /2$ corresponds to a Gaussian prior

$$p(\theta_i) = \mathcal{N}\left(\theta_i; 0, \frac{1}{\alpha}\frac{1}{1 - p_{drop,i}}\right). \tag{2.11}$$

Thus, optimising a Neural Network with dropout and some regularisation of the weights can be interpreted as performing Variational Inference with a prior defined by (2.10).

2.2.1 alpha-divergence

The exclusive KL-divergence $\mathrm{KL}[q||p]$ used in the derivation of Variational Inference favours matching a small region around a posterior mode (zero-forcing) [23, 41]. In contrast, the inclusive KL-divergence $\mathrm{KL}[p||q]$ requires $q(\theta) > 0$ when $p(\theta) > 0$ otherwise $\mathrm{KL}[p||q] = \infty$ (zero-avoiding). As a result q must cover the entire posterior and is penalised for how well it fits the posterior globally.

Ideally, a controllable trade-off should be made between precisely covering a local, high probability region of the posterior and approximately covering the posterior as a whole. Therefore, consider instead the Amari's α-divergence $\mathrm{D}_\alpha[p||q]$ [2] which is defined as

$$D_\alpha[p||q] = -\frac{1}{\alpha(1-\alpha)}\left(1 - \int p(\theta)^\alpha q(\theta)^{1-\alpha} d\theta\right), \quad \alpha > 0. \tag{2.12}$$

Amari's α-divergence has the exclusive and inclusive KL divergence as limiting cases:

$$\begin{aligned}\lim_{\alpha\to 0} D_\alpha[p||q] &= \mathrm{KL}[q||p], \\ \lim_{\alpha\to 1} D_\alpha[p||q] &= \mathrm{KL}[p||q].\end{aligned} \tag{2.13}$$

For Bayesian Neural Networks the integral in (2.12) has no closed form solution and is approximated using a Monte Carlo estimate. When sampling from $q(\theta)$, the MC estimator for (2.12) computes

$$\tilde{D}_\alpha[p||q] = -\frac{1}{\alpha(1-\alpha)}\left(1 - \mathbb{E}_{q(\theta)}\left[\left(\frac{p(\theta)}{q(\theta)}\right)^\alpha\right]\right) \tag{2.14}$$

The reformulation in (2.14) is exact for $0 < \alpha < 1$. However, for the limit $\alpha \to 1$, the approximation yields

$$\lim_{\alpha\to 1} \tilde{D}_\alpha[p||q] = \mathrm{KL}[p^*||q], \quad p^*(\theta) = \begin{cases} p(\theta) & q(\theta) > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{2.15}$$

Thus, the MC alpha-divergence generalises the inclusive KL if $p(\theta)$ and $q(\theta)$ are defined on the same support[4]. For Dropout Variational Inference this condition clearly does not hold in general because the support of $q(\theta)$ exists of a finite number of points whereas $p(\theta)$ is typically dense.

Minimising the alpha-divergence $D_\alpha[p(\theta|X)||q_\phi(\theta)]$ directly is difficult because it requires an unbiased estimator of $p(X|\theta)^\alpha$ which is only possible when using the full dataset X or with product estimators like the Poisson estimator.

Alternatively, the alpha-divergence can be minimised locally which results in the loss being linear in the likelihood terms. The local divergence quantifies the average impact that a likelihood term $p(X_i|\theta)$ has on the posterior. The local approximation originates from the Expectation Propagation algorithm where all posterior factor $\{p(X_i|\theta)\}_i$ and $p(\theta)$ are approximated by a separate function $q^i_\phi(\theta)$.

For Neural Networks this is unpractical because it is impossible to store a distribution over the parameters for every example in the training set [23]. Therefore, a single function $q^i_\phi(\theta) = q_\phi(\theta)^{1/N}$ is used for all factors instead. The objective reduces to

$$\mathscr{L}_\alpha(\phi) = -\frac{1}{\alpha}\sum_{i=1}^N \log \mathbb{E}_{q_\phi(\theta)}\left[\left(\frac{p(X_i|\theta)p(\theta)^{1/N}}{q_\phi(\theta)^{1/N}}\right)^\alpha\right]. \tag{2.16}$$

The loss function in (2.16) is known as the BB-α energy [23]. For dropout VI we cannot immediately optimise (2.16) because the infinite entropy caused by the Dirac deltas in the variational distribution $q_\phi(\theta)$ cannot be isolated from the finite density terms.

An approximation of (2.16) proposed in [41] is derived by introducing a reparameterisation of the variational distribution

$$q_\phi(\theta) = \frac{1}{Z}\tilde{q}_\phi(\theta)\left(\frac{\tilde{q}_\phi(\theta)}{p(\theta)}\right)^{\frac{\alpha}{N-\alpha}}, \tag{2.17}$$

[4] More precisely, the MC alpha-divergence generalises the inclusive KL when $p(\theta) = 0$ whenever $q(\theta) = 0$. Hence, the measure dp must be absolutely continuous w.r.t. dq.

where Z is the normalisation constant of $q_\phi(\theta)$. Substituting (2.17) into the the BB-α energy (2.16) yields

$$\begin{aligned}\mathscr{L}_\alpha(\phi) &= -\frac{1}{\alpha}\sum_{i=1}^{N}\log\int q_\phi(\theta)^{1-\frac{\alpha}{N}}p(X_i|\theta)^\alpha p(\theta)^{\frac{\alpha}{N}}d\theta \\ &= \frac{N-\alpha}{\alpha}\log Z - \frac{1}{\alpha}\sum_{i=1}^{N}\log\int \tilde{q}_\phi(\theta)p(X_i|\theta)^\alpha d\theta \\ &= R_\beta[\tilde{q}(\theta)||p(\theta)] - \frac{1}{\alpha}\sum_{i=1}^{N}\mathbb{E}_{\tilde{q}_\phi(\theta)}[p(X_i|\theta)^\alpha],\ \ \beta = \frac{N}{N-\alpha},\end{aligned} \tag{2.18}$$

where $R_\beta[.]$ is the Rényi divergence [56] which is defined as

$$R_\alpha[p||q] = \frac{1}{\alpha-1}\int p(\theta)^\alpha q(\theta)^{1-\alpha}d\theta. \tag{2.19}$$

Just like the Amari's α-divergence, the inclusive KL-divergence is a limiting case of the Rényi divergence

$$\lim_{\alpha\to 1} R_\alpha[p||q] = KL[p||q]. \tag{2.20}$$

Because $\frac{N}{N-\alpha} \to 1$ and $q_\phi(\theta) \to \tilde{q}_\phi(\theta)$ as $N \to \infty$, [41] proposes to approximate (2.18) by

$$\mathscr{L}_\alpha(\phi) \approx \mathrm{KL}[q_\phi(\theta)||p(\theta)] - \frac{1}{\alpha}\sum_{i=1}^{N}\log\mathbb{E}_{q_\phi(\theta)}[p(X_i|\theta)^\alpha]. \tag{2.21}$$

The approximation of the BB-α loss makes it possible to extend dropout VI to (local) α-divergences. The loss in (2.21) differs from the VI objective (2.5) only in the likelihood term. For $\alpha \to 0$, the BB-α loss (2.21) reduces to the VI loss (2.5). The loss in (2.21) can be minimised by using SGD with either the local or global reparameterisation trick. Using a MC estimate for the expectation does however lead to a biased estimator of the gradient. Empirical results show that the bias is small compared to the variance [23].

For $0 < \alpha < 1$ the solution found by BB-α can be interpreted as a trade-off between a zero-forcing and zero-avoiding posterior approximation. Based on empirical observation of the test accuracy for various α-divergences, it has been suggested that $\alpha = 0.5$ works well in practise [23].

2.2.2 Shortcomings

Dropout variational inference is found to be a simple and effective technique to reduce overfitting and improve test accuracy [15, 41, 46]. However, It has also been demonstrated that Dropout VI can lead to pathological uncertainty estimates [52]. Dropout VI is unable to model posterior concentration which makes its use problematic for applications where calibration is critical[5].

The lack of posterior concentration is best understood from a simple counterexample. Consider the VI optimisation objective for a dataset X' which consists of N copies of the original dataset X and a (nearly) uniform prior. The new objective becomes

$$\mathscr{L}(\phi) = \mathbb{E}_{q_\phi(\theta)}[\underbrace{\log q_\phi(\theta)}_{\text{entropy}} - \underbrace{N \log p(X|\theta)}_{\text{likelihood}}]. \tag{2.22}$$

The true posterior $p(\theta|X) \propto p(X|\theta)^N$ and will thus concentrate around the mode(s) of the posterior as $N \to \infty$. The entropy term for Dropout VI is constant and therefore $\hat{\phi} = \arg\max \mathscr{L}(\phi)$ does not depend on N. Hence, the posterior approximation will always try to find the maximum likelihood model that can be represented by the noisy model irrespectively of how concentrated the true posterior is. The only regularisation comes from the prior and the variational distribution used in Dropout VI which cannot represent a MAP approximation $q(\theta) = \delta(\theta - \hat{\theta})$ because some probability mass is always assigned to $\theta_i = 0$.

2.3 Bayes By Backprop

An alternative approach to Dropout VI is Bayes By Backprop [7] which aims to learn a fully factorised Gaussian approximation of the posterior. Previous attempts to learn a Gaussian approximation did not make use of reparameterisation trick [19] and were therefore not competitive with Dropout VI due to large variance in the gradient estimate.

Note that a Gaussian distribution $\theta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ can be reparameterised as

$$\theta = g_\phi(\epsilon) = \epsilon \odot \sigma + \mu, \quad \epsilon \sim \mathcal{N}(0, I). \tag{2.23}$$

Using the ELBO objective (2.8) yields

$$\log p(X) \geq \mathscr{L}(\phi) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \left[\log p(X|\theta) + \log \frac{p(\theta)}{q_\phi(\theta)} \right]. \tag{2.24}$$

[5] More generally, the argument that follows holds for any family of distributions $q_\phi(\theta)$ where the entropy $\mathbb{E}[-\log q_\phi(\theta)]$ is invariant w.r.t. the variational parameters ϕ.

Various MC estimators can be used to obtain a stochastic gradient for the objective. The simplest and most flexible approach is to use the global reparameterisation trick (2.23). Alternatively, the prior can be restricted to a family of distribution for which the KL term of the ELBO (2.24) can be computed analytically. The variance of the estimate can be reduced further by using the local reparameterisation of the Gaussian noise

$$z'_j = \theta_j^T z = \mu_j^T z + \sqrt{\sum_i z_i^2 \sigma_{ji}^2}\, \epsilon_j, \quad \epsilon_j \sim \mathcal{N}(0, 1), \tag{2.25}$$

where z' are the new activations before applying non-linearities and z is the input of the layer.

The main benefits of BBB is that the variational distribution has full support resulting in a finite, well-defined ELBO and the adjustable noise which allows for the posterior approximation to concentrate when the amount of data increases. A drawback of BBB is that no correlation between weights is captured in the approximate posterior.

2.3.1 Pathological optima

The flexible entropy of BBB does not immediately result in well-calibrated Bayesian Neural Nets. The optima found by BBB strongly depend on the initialisation of the variances and the choice of prior. The effect remains largely undocumented but seems to be caused by the KL term which is typically at least an order of magnitude larger than the likelihood term in (2.24).

Optimisation of the ELBO objective tends to result in a posterior approximation with too much variance leading to significantly underconfident predictions. This phenomenon has been attributed to the bias in BBB and VI methods in general to learn a variational distribution that overfits the prior and ignores the data, causing many weights to have zero mean and variance close to the prior variance [64]. The experiments in Chap. 4 will show that this effect has significant negative consequences for calibration.

2.4 Variational Gaussian Dropout

An alternative Gaussian variational distribution is multiplicative Gaussian noise which can be interpreted as a variation of dropout. In Gaussian dropout the weights of each feature are multiplied with Gaussian noise centred at zero

$$\theta_i = \epsilon_i \mu_i \sim \mathcal{N}(\mu, \sigma_i^2 \mu_i \mu_i^T), \quad \epsilon_i \sim \mathcal{N}(1, \sigma_i^2). \tag{2.26}$$

Note that the distribution over the weights is degenerate because the covariance matrix is not full rank. The derivation of Gaussian VI that follows is based on [35]. It is important to note that the derivation is by no means rigorous. It contains numerous technical issues [27] and should therefore be treated as a intuitive motivation rather than a proof of correctness.

The variational entropy term is given by[6]

$$-\mathbb{E}[\log q_\phi(\theta)] = \sum_i -\mathbb{E}_{\epsilon_i}[\log p(\epsilon_i)] + \sum_j \log|\mu_{ij}| \tag{2.27}$$

The prior for Gaussian Dropout VI is implicit. That is $\mathrm{KL}[q(\theta)||p(\theta)]$ is constant in μ. Consider the prior $\log p(\theta_{ij}) = -\log|\theta_{ij}| + \log(c/2)$ for $-1/2c < \log|\theta_{ij}| < 1/2c$. The KL term in the ELBO (2.8) reduces to[7]

$$-\mathrm{KL}[q(\theta)||p(\theta)] = \sum_i -\mathbb{E}_{\epsilon_i}[\log|\epsilon_i|] + \log\sigma_i + C \tag{2.28}$$

The term $\mathbb{E}_{\epsilon_i}[\log|\epsilon_i|]$ is intractable but can be well-approximated by a third order polynomial [8] w.r.t. σ^2 [35]

$$-\mathrm{KL}[q(\theta)||p(\theta)] \approx \sum_i \log\sigma_i + c_1\sigma_i^2 + c_2\sigma_i^4 + c_3\sigma_i^6, \tag{2.29}$$

where the constants are given by

$$c_1 = 1.16145124, \quad c_2 = -1.50204118, \quad c_3 = 0.58629921. \tag{2.30}$$

The variance of the multiplicative noise σ_i fulfils a similar function as the dropout probability in Bernoulli dropout. Additionally, a derivative of the ELBO w.r.t. σ_i can be computed and it is therefore possible to let σ_i be a variational parameter as opposed to a hyperparameter.

[6] The density of $q_\phi(\theta)$ is taken w.r.t. the Lebesgue measure $d\mu$, the line in the direction of the weight vector going through the origin. Defining the density w.r.t. $d\theta$, as appropriate, would cause the density to be ∞ on the line defined by μ. Consequently, the entropy term would diverge to $-\infty$ similar to what was previously observed in Bernoulli Dropout VI where it was denoted with Dirac delta functions.

[7] The ELBO is ill-defined due to the diverging entropy term which causes the bound to be uninformative $\log p(X) \geq -\infty$. The diverging term is constant in the parameters and is therefore ignored such that we can still derive an optimisation procedure and a finite objective.

[8] In [35] it is argued that $-\mathbb{E}_{\epsilon_i}[\log|\epsilon_i|]$ is strictly positive. However, more recent work has shown that this is not the case [47]. Therefore, this approximation becomes increasingly poor as $\sigma \to \infty$. A better approximation is proposed in [47] which unfortunately could not be incorporated into the experiments because the flaw in [35] was discovered in the final stage of the project.

The dropout rate p_i and multiplicative Gaussian noise σ_i can be related by matching moments. Consider the dropout operation to be a mask $\epsilon_i \sim \mathscr{B}(1-p_i)$ and the activations after dropout $z_i' = z_i \odot \epsilon_i/(1-p)$. Computing moments of Bernoulli dropout yields

$$\begin{aligned} \mathbb{E}[z_i'] &= z_i, \\ \mathrm{Var}[z_i'] &= z_i^2 \frac{p_i}{1-p_i}. \end{aligned} \tag{2.31}$$

Thus, a network with Bernoulli dropout masks $\mathscr{B}(1-p_i)$ has approximately the same amount of stochastic regularisation as a Gaussian dropout network with $\sigma_i^2 = p_i/(1-p_i)$ [63].

In [35], a constraint on the injected noise $\sigma^2 < 1$ was used to avoid the variational distribution from moving to full sparsity. Follow up research showed that the constraint was necessary due to errors in the approximation of the KL term (2.29) [47].

2.5 Dropout-BBB

The BBB algorithm demonstrates how Variational Inference can provide a sound approximate Bayesian Inference technique which can be used in combination with Stochastic Gradient Ascent. Unfortunately, BBB often does not work as well as dropout VI methods [46]. In contrast, Gaussian and Bernoulli dropout offer good generalisation performance in practise but suffer from a degenerate ELBO due to a mismatching support with the posterior and prior distributions. One possible bottleneck of BBB is the lack of weight correlation in BBB because it uses a fully-factorised distribution to keep time and memory complexity linear in the number of weights.

Thus, we aim to induce a full rank Gaussian distribution over the weights that has some correlations. Furthermore, the variational distribution should allow for a linear parameterisation in the number of weights, and linear time sampling and pdf evaluation. Most Gaussian VI methods fail on one of these criteria. For example, BBB has no correlations, Gaussian Dropout is not full rank, and more recent approaches like [44] do not allow exact linear pdf evaluation.

Recall that the Gaussian dropout VI defines a degenerate Gaussian distribution over the weights $p(\theta_i) = \mathcal{N}(\mu, \sigma_i^2 \mu_i \mu_i^T)$. The covariance matrix can be made full-rank by adding a diagonal matrix $\mathcal{N}(\mu, \Sigma = \alpha^2 \mu\mu^T + A)$ with $A_{ii} = \sigma_i^2$ and $A_{ij} = 0$ for $i \neq j$. Thus, the Variational distribution over the weights consists of a mean which is multiplied with Gaussian noise $\epsilon \sim \mathcal{N}(1, \alpha^2)$ and independent Gaussian noise $\mathcal{N}(0, A)$. The Variational distributions of BBB and Gaussian dropout are special cases of this distribution. Not however that the ELBO is finite for $\alpha^2 = 0$ but diverges for $|A| \to 0$.

Sampling from this distribution is possible with both the global and local reparameterisation trick

$$\theta_j = (1 + \alpha_j \epsilon_j^{(m)})\mu_j + \epsilon_j^{(a)} \odot \sigma_j, \ \ \epsilon_j^{(m)} \sim \mathcal{N}(0,1), \ \ \epsilon_j^{(a)} \sim \mathcal{N}(0,I), \tag{2.32}$$

$$z'_j = \theta_j^T z = \mu_j^T z + \sqrt{(\alpha_j \mu_j^T z)^2 + \sum_i \sigma_{ji}^2 z_i^2}\,\epsilon_j, \ \ \epsilon_j \sim \mathcal{N}(0,1). \tag{2.33}$$

Evaluating the ELBO requires the log-likelihood of the weights

$$\log q_\phi(\theta) = -\frac{D}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(\theta - \mu)^T \Sigma^{-1}(\theta - \mu), \tag{2.34}$$

The inverse of the covariance Σ^{-1} can be expressed analytically by the Sherman–Morrison formula

$$\Sigma^{-1} = A^{-1} - \frac{\tilde{\mu}\tilde{\mu}^T}{1 + \alpha\tilde{\mu}^T\mu}, \ \ \tilde{\mu} = \alpha\frac{\mu}{\sigma^2}. \tag{2.35}$$

Applying the matrix determinant lemma yields the log determinant of the covariance

$$\begin{aligned} \log|\Sigma| &= \log|A| + \log\left(1 + \alpha^2\mu^T A^{-1}\mu\right) \\ &= \sum_{i=1}^{D}\log\sigma_i^2 + \log\gamma, \ \ \gamma = 1 + \alpha^2\sum_{i=1}^{D}\frac{\mu_i^2}{\sigma_i^2} \end{aligned} \tag{2.36}$$

Substituting (2.35) into (2.34) yields

$$\log q_\phi(\theta) = -\frac{1}{2}\left(D\log 2\pi + \log|\Sigma| - \frac{\left([\theta - \mu]^T\tilde{\mu}_i\right)^2}{\gamma} + \sum_{i=1}^{D}\frac{(\theta_i - \mu_i)^2}{\sigma_i^2}\right). \tag{2.37}$$

Thus, the log likelihood reduces to an expression that can be evaluated in linear time w.r.t. the number of parameters.

A disadvantage of this approach is that the multiplicative noise does not affect the additive noise (2.32). Sec. 2.6.3 defines an alternative approach which multiplies the activations after additive noise by introducing additional model parameters.

2.6 Sparse Variational Inference

Variational Inference was originally used for improving test accuracy and uncertainty estimates. However, it can be equally useful for compressing a network such that it can be stored using less memory or evaluated with less floating point operations [43, 47]. Sec. 2.6.1 introduces the idea of compression priors which are meant to induce sparsity in the network parameters. Sec. 2.6.2 and Sec. 2.6.3 introduces two VI methods which are designed to infer a sparse distribution over the model parameters.

2.6.1 Compression priors

The scale invariant prior (2.29) is no longer necessary to cancel out the vanishing entropy for weights close to zero because in this case the additive noise will dominate. Still a scale invariant prior is appealing because a Gaussian prior $p(\theta) = \mathcal{N}(0, \sigma_p^2)$ either causes the weights to shrink for small σ_p^2 or promotes large variances in the posterior approximation due to the KL term in the ELBO (2.8).

Note that the scale invariant prior can be interpreted as an improper mixture of Gaussians

$$p(\theta) \propto \int\limits_{\sigma=0}^{\infty} \frac{1}{\sigma}\mathcal{N}(\theta; 0, \sigma^2) d\sigma \propto \frac{1}{|\theta|}. \tag{2.38}$$

In this form (2.38) the scale invariant prior can be related to a family of sparsity inducing priors [43]

$$p(\theta) = \int\limits_{\sigma=0}^{\infty} p(\sigma)\mathcal{N}(\theta; 0, \sigma^2) d\sigma. \tag{2.39}$$

Various well known compression priors are special cases of (2.39). For example $p(\sigma) = 0.5\delta(\sigma - \sigma_1) + 0.5\delta(\sigma - \sigma_2)$ with $\sigma_1 \ll \sigma_2$ corresponds to a spike-and-slab prior which was found to be an effective prior in the original work on the BBB algorithm [7]. If we let $p(\sigma^2) = \text{Exponential}(1)$ then $p(\theta) = \text{Laplace}(0, 1/\sqrt{2})$.

Although conceptually simple, the spike-and-slab prior can be seen as a continuous relaxation of an L_0 norm which becomes less smooth as $\sigma_1 \to 0$. This makes training with a spike-and-slab prior difficult due to strong non-convexity [48]. Therefore, we propose to generalise the spike-and-slab by discretisation of the scale-invariant prior (2.38)

$$p(\sigma) = \frac{1}{\sum_{i=1}^{n} \theta_i^{-T}} \sum_{i=1}^{n} \left(\frac{1}{\sigma_i}\right)^T \delta(\sigma - \sigma_i), \quad \sigma_1 < \sigma_2 < \cdots < \sigma_n \tag{2.40}$$

The regularisation term in (2.8) will be minimised by learning a distribution where many weights have mean 0 and variance close to σ_1. These weights will have no significant contribution to the output of the network and could therefore be pruned away. Pruning/compression is not the goal of this work. However, we find that picking a prior naively leads to a regularisation term that is at least an order of magnitude larger than the likelihood term. In this setup the ELBO can often be maximised trivially by minimising the KL term which results in a network that does not learn anything about the data.

Many weights in a well-trained neural network can be pruned away without affecting the accuracy [11]. Thus, a smaller network will result in a smaller KL term such that maximisation of the ELBO is more likely to be dominated by finding a good fit for the data.

2.6.2 Gauss-Gamma Variational Inference

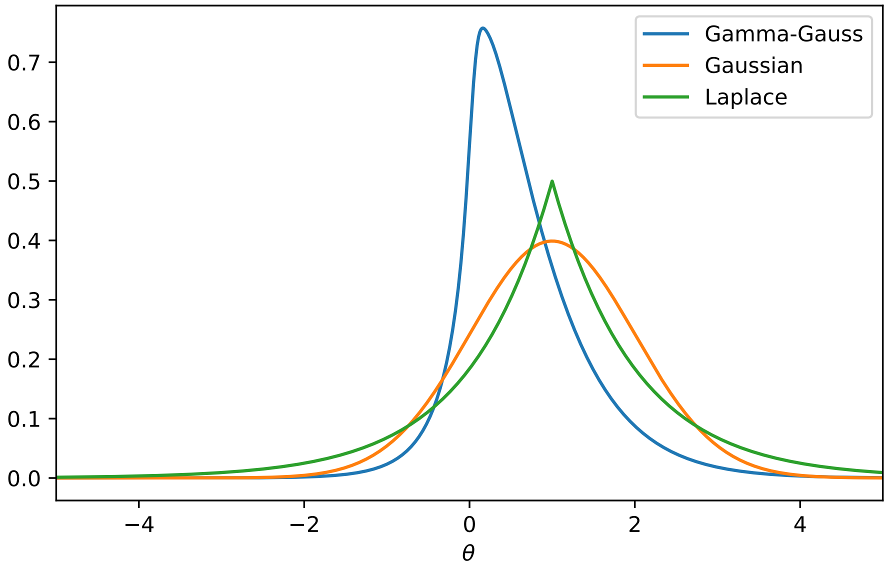


Fig. 2.1 A non-central Gauss-Gamma distribution ($\mu = 1, \sigma^2 = 1, \alpha = 1$, and $\beta = 1/2$) together with a Laplace and Gaussian distribution.

The decomposition in (2.39) can be taken a step further by considering the scale σ^2 and the Gaussian noise as separate model parameters

$$\theta = x\sqrt{z}, \quad p(x, z) = \mathcal{N}(x; \mu_p, \sigma_p^2)\mathcal{G}(z; \alpha_p, \beta_p), \tag{2.41}$$

where θ is no longer a model parameter and the variance z is constraint to follow a Gamma distribution. This parameterisation allows a rich family of priors. We will consider only central priors $\mu_p = 0$ which makes the variance of Gaussian distribution redundant $\sigma_p^2 = 1$. For example, a Laplace prior $p(\theta) = \text{Laplace}(0, b)$ can be realised with $\alpha_p = 1$ and $\beta_p = 2b^2$. The scale invariant is an improper limiting case with $\alpha_p, \beta_p \to 0$.

A variational approximation can be defined as

$$q_\phi(x, z) = \mathcal{N}(x; \mu, \sigma^2)\mathcal{G}(z; \alpha, \beta) \tag{2.42}$$

The Gauss-Gamma variational distribution (2.42) has an interesting sparsity property as can be seen in Fig. 2.1 which shows a non-central Gamma Gaussian distribution together with a Gaussian an Laplace distribution. The distribution concentrates most of its weight around zero even for non-central parameterisations. Thus, this variational distribution favours sparse networks. A Gaussian or Laplace variational distribution is unable to capture a distribution with non-zero mean with most of its mass around zero.

The reparameterisation of θ further allows an easy extension to group sparsity

$$\theta_i = x_i\sqrt{Zz_i}, \quad p(x_i, z_i) = \mathcal{N}(x; 0, 1)\mathcal{G}(z; \alpha_p, \beta_p), \quad p(Z) = \mathcal{G}(Z; A_p, B_p), \tag{2.43}$$

where θ_{ii} are the weights for a single feature or filter. The variational distribution is the same as (2.42) with an additional Gamma distribution over Z.

Gauss-Gamma VI has a more flexible variational distribution compared to Dropout-BBB and allows for the KL term to be expressed in closed form for any (sparse) prior of the form (2.43). Unlike Dropout-BBB the multiplicative noise is treated as a separate model parameter. The over-parameterisation of Gauss-Gamma VI might result in worse approximation despite the increased flexibility.

Gauss-Gamma Variational Inference is related to group Bayesian compression [43] which uses a Normal prior on the weights x and a scale invariant or horseshoe prior on Z with a Normal and Log-Normal variational distribution on x and Z, respectively.

2.6.3 Bayesian Compression

Recent work has shown that VI can be used to achieve competitive results on model compression [47, 43]. Variational Inference methods in general and BBB especially suffer from a bias towards fitting the prior (Sec. 2.3.1) which will lead to under-fitting of the training data and poorly calibrated, underconfident models.

Prior work has already shown that the uncertainty estimates in VI methods can be used to prune weights or entire features the network without significant loss in training accuracy [7]. In [43] BBB is extended to work with group sparsity by multiplying the activation of each feature y_i with a factor z_i with a scale invariant prior $p(z_i) \propto 1/z_i$ and Gaussian variational distribution. This VI method is similar to Gaussian Dropout VI. The differences are that the multiplicative noise is considered a model parameter and the original network weights are captured by a factorised Gaussian like in BBB. The VI distribution thus has full support but is over-parameterised and uses the improper scale-invariant prior for the dropout mask z_i. This extension allows both weights and features to be pruned. Removing features is especially useful because it makes evaluation of the model faster.

For BBB with a Gaussian prior, the variational distribution of most weights tend to fit the prior perfectly. These weights do not contribute to fitting the data and instead only insert noise into the activations of the network. We hypothesise that pruning can be used to improve the calibration of networks by removing these weights and features that do not contribute to fitting the data from the network. Additionally, the pruned network can be stored in less memory and requires less computational resources to evaluate.

In contrast to earlier work on compression, we prune the weights and features of the network after each epoch. Chap. 4 will show that this leads to faster training and improved results. We found it useful to quantify the amount of noise in weights and features as ”dropout rates”. This is done by first reparameterisation a gaussian $x \sim \mathcal{N}(\mu, \sigma^2)$ as

$$x = \mu\epsilon, \ \ \epsilon \sim \mathcal{N}\left(1, \alpha^2 = \frac{\sigma^2}{\mu^2}\right). \tag{2.44}$$

Note that $1/\alpha$ is actually the signal to noise ratio of a normally distributed variable. The dropout rate p_{drop} follows by matching moments with a dropout mask (2.31) which yields

$$p_{drop} = \frac{\alpha^2}{1+\alpha^2}. \tag{2.45}$$

The dropout rates for weights and features can be used to determine which parts of the network are redundant. When pruning during training it is important to use a conservative threshold like $p_{drop} > 0.99$. Once the network has converged, the threshold can be lowered until the desired tradeoff between model size and accuracy is obtained.

2.7 Sampling Methods

Sampling methods offer an alternative approach towards inference in Bayesian Neural Networks. MCMC methods require little prior knowledge about the shape of the posterior and the architecture of the model. MCMC methods aim to construct a Markov Chain with transition distribution $p(\theta_{t+1}|\theta_t)$ with the stationary distribution being the posterior $\lim_{t\to\infty} p(\theta_t) = p(\theta|X)$.

To proof a Markov Chain has the posterior as its stationary distribution $P(\theta|X)$, it is sufficient to proof the chain is ergodic and it satisfies detailed balance w.r.t. the posterior [21]

$$p(\theta|X)p(\theta'|\theta) = p(\theta'|X)p(\theta|\theta') \;\forall\; (\theta, \theta'). \tag{2.46}$$

Let $q(\theta'|\theta)$ be any distribution. Then the log posterior ratio is defined as

$$\Delta(\theta', \theta) = \log \frac{p(\theta'|X)q(\theta|\theta')}{p(\theta|X)q(\theta'|\theta)} = -\Delta(\theta, \theta'). \tag{2.47}$$

A Markov Chain is constructed by sampling a proposal state $\theta' \sim q(\theta'|\theta)$ and accepting θ' as the next state with probability $P_A = A(\Delta(\theta', \theta))$ and $\theta' = \theta$ otherwise. The acceptance procedure implicitly yields a transition distribution

$$p(\theta'|\theta) = q(\theta'|\theta)A(\Delta(\theta', \theta)) + \left(1 - \int dq(\theta'|\theta)A(\Delta(\theta', \theta))\right)\delta(\theta' - \theta). \tag{2.48}$$

The detailed balance condition w.r.t. the posterior (2.46) implies

$$\frac{A(\Delta(\theta', \theta))}{A(-\Delta(\theta', \theta))} = \frac{p(\theta'|X)q(\theta|\theta')}{p(\theta|X)q(\theta'|\theta)} = \exp(\Delta(\theta', \theta)). \tag{2.49}$$

Thus, there is a family of functions $A(\Delta)$, defined by the constraint $A(\Delta) = A(-\Delta)\exp(\Delta)$ and $0 \leq A(\Delta) \leq 1$ which can be used to construct a Markov Chain with the posterior as its stationary distribution for an arbitrary proposal distribution $q(\theta'|\theta)$ [60].

The acceptance function only requires the posterior probability ratio $p(\theta'|X)/p(\theta|X) = p(\theta', X)/p(\theta, X)$ which does not depend on the normalisation constant $p(X)$ and is therefore tractable for many probabilistic models.

There exists a unique function $A(\Delta)$ maximising the number of accepted proposals and consequently the sample efficiency

$$A_{\mathrm{MH}}(\Delta) = \min(1, \exp(\Delta)). \tag{2.50}$$

The optimal acceptance function A_{MH} is known as the Metropolis-Hastings acceptance function[21]. The sample efficiency of the Markov Chain also depends on the proposal distribution $q(\theta'|\theta)$. Constructing an effective proposal distribution is non-trivial especially when little is known about the shape of the posterior.

Note that the Markov Chain does not need to be homogeneous. Instead the Markov Chain may consist of an infinite sequence of transition distributions $\{p_t(\theta'|\theta)\}$ where each transition (eventually) leaves the posterior distribution invariant $\lim_{t\to\infty}\int dp(\theta|X)p_t(\theta'|\theta) = p(\theta'|X)$ as long as the the chain is ergodic. Thus, the posterior can be sampled by a mix of proposal distributions and/or transition distributions that keep the posterior invariant by some other means.

2.7.1 Hamiltonian Monte Carlo

The Hamiltonian dynamics can be used to define an effective proposal distribution for a continuous posterior. Let $\pi \sim \mathcal{N}(0, M)$ be an independent auxiliary random variable of the same dimensionality as θ. A sample from $p(\theta|X)$ can be obtained from a sample from the joint distribution $p(\theta, \pi|X) = p(\theta|X)p(\pi)$ by simply discarding the auxiliary variable π.

The proposal $q(\theta', \pi'|\theta, \pi) = \delta(\theta)p(\pi)$ leaves the posterior and θ invariant. Likewise, simulating Hamiltonian dynamics with momentum π and potential $-\log p(\theta|X)$ leaves the posterior invariant

$$\nabla_t \pi = \nabla_\pi \log p(\theta, \pi|X), \quad \nabla_t \theta = -\nabla_\pi \log p(\theta, \pi|X). \tag{2.51}$$

Under mild conditions, alternating between simulating Hamiltonian dynamics for a finite timespan L and resampling the momentum $\pi' \sim p(\pi)$ is ergodic [51]. In practice, the simulation of Hamiltonian dynamics can only be approximated using a time discretisation with time step ϵ. For small ϵ, the MH acceptance probability approaches one $\Delta((\pi', \theta'), (\theta, \pi)) \approx 0$ and the acceptance can be skipped. Reversible integrators like leapfrog are preferred because for such integrators $q(\theta', \pi'|\theta, \pi)/(\theta, \pi|\theta', \pi') = 1$. Thereby eliminating one potential source of error [51].

The Leapfrog integrator for HMC yields

$$\begin{aligned} \pi\,(t + \epsilon/2) &= \pi'(t) + \frac{\epsilon}{2}\nabla_\theta \log P(\theta(t)|X), \\ \theta\,(t) &= \theta(t) + \frac{\epsilon}{2}M^{-1}\pi(t + \epsilon/2), \\ \pi(t + \epsilon) &= \pi'(t) + \frac{\epsilon}{2}\nabla_\theta \log P(\theta(t)|X), \end{aligned} \tag{2.52}$$

where $\pi'(t)$ is the momentum after resampling. The HMC algorithm is a powerful tool whenever the gradient of the posterior $\nabla_\theta \log P(\theta|X)$ can be computed efficiently.

Stochastic HMC

The gradient in (2.52) cannot be replaced naively with a mini-batch estimate without introducing significant bias. Irrespective of the step size ϵ, the entropy of the stationary distribution of HMC increases with L for stochastic gradients [8].

The stochastic gradient noise affects the momentum updates in (2.52). When only a single step is simulated $L = \epsilon$, the Hamiltonian dynamics reduce to Langevin dynamics. The mini-batch noise can then be made arbitrary small using a constraint step size annealing scheme

$$\sum_{t=1}^{\infty} \epsilon_t = \infty, \quad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty. \tag{2.53}$$

The same conditions are required to guarantee the convergence of SGD. The SGLD sampler will eventually sample from the true posterior given (2.53) at the cost of increasingly slow mixing [66].

More generally, Hamiltonian dynamics [8] with friction can be shown to leave the posterior invariant when a stochastic gradient is used (with conditions 2.53). After discretisation, the friction dynamics are equivalent to a simulation of a single step $L = \epsilon$ of the original Hamiltonian Dynamics and partial momentum refreshment

$$\pi'(t) \sim \mathcal{N}\left(\alpha\pi(t), (1-\alpha^2)M\right), \tag{2.54}$$

where α is the friction coefficient. When $\alpha = 0$ the Langevin dynamics sampler is obtained again. The HMC sampler with partial momentum refreshment and stochastic gradient is known as the SGHMC sampler [8].

In practice, letting $\epsilon \to 0$ to obtain unbiased samples is undesirable because the samples become increasingly correlated. Typically, the step size is fixed when the bias has fallen below an acceptable threshold. A simple heuristic is to measure the ratio of injected noise and stochastic gradient noise [66]. However, this threshold is purely a heuristic and does not give any guarantees about the convergence of the stationary distribution of the sampler towards the posterior.

The covariance matrix M can be chosen freely to pre-condition the sampler on the covariance of the posterior. For Bayesian Neural Networks a full covariance matrix is computationally intractable in memory and compute. A diagonal approximation of the covariance also requires significant computational power in general [1]. Therefore, we limit ourselves to the non pre-conditioned case $M = I$.

2.7.2 Stochastic acceptance function

Asymptotically unbiased samplers can be obtained by letting $\epsilon \rightarrow 0$. However, slow mixing makes such a step size annealing scheme impractical. Various methods have been proposed to introduce a acceptance function into stochastic samplers which use uses only a subset of the data to determine the acceptance probability [3, 4, 38].

Stochastic Metropolis-Hastings test

The MH acceptance function can be rephrased as the hypothesis test $\Delta(\theta', \theta) > \log u$ with $u \sim \mathcal{U}[0, 1)$. The hypothesis can accepted or rejected with some confidence threshold α by estimating $\Delta(\theta', \theta)$ from a large enough sample $P(\theta, \tilde{X})$. The hypothesis test can be performed approximately based on a student-t test assuming CLT [38] or using bounds for finite samples without replacement [3]. The confidence threshold is easily met when $\left\|\Delta(\theta', \theta) - \log u\right\|$ is large. However, in some cases the algorithm will require all N data samples [60].

Stochastic Barker test

Barker's algorithm [5] is an alternative to MH which uses the cumulative density of the logistic distribution which is the sigmoid function

$$\sigma(\Delta) = \frac{1}{1 + \exp(-\Delta)}. \tag{2.55}$$

The Barker acceptance function is equivalent to the test $\Delta + \eta > 0$ where η is sampled from the Logistic distribution. Motivated by the CLT, assume for some mini-batch the posterior ratio is corrupted by additive Gaussian noise $\tilde{\Delta} = \Delta + \kappa$ with $\kappa \sim \mathcal{N}(0, \sigma^2)$. The Barker acceptance test is then given by the decision boundary

$$\tilde{\Delta} + \tilde{\eta} = \Delta + \kappa + \tilde{\eta} > 0, \quad \tilde{\eta} \sim p(\tilde{\eta}) \tag{2.56}$$

where $p(\kappa + \tilde{\eta}) = p(\kappa) * p(\tilde{\eta})$[9]is again the Logistic distribution.

The Logistic distribution is well approximated by a Gaussian distribution $\sigma'(\Delta) \approx \mathcal{N}(0, \pi^2/3)$ (using moment matching). Therefore, the injected noise can be approximated by [4]

$$p(\tilde{\eta}) \approx \mathcal{N}(0, \pi^2/3 - \sigma^2), \quad \sigma^2 \leq \pi^2/3. \tag{2.57}$$

[9] Here $*$ denotes the convolution operator.

A better approximation can be found by numerical optimisation [60]. The stochastic Barker test is limited by the fact that the injected noise always increases the entropy due to the independence of the mini-batch noise. Consequently, the total noise cannot approximate a Logistic distribution when the mini-batch likelihood variance is large $\sigma^2 \gg \pi^2/3$.

Error bounds for acceptance tests

Stochastic acceptance functions introduce bias into the sampler by violating the acceptance function constraint (2.49). Consider an acceptance function A_ϵ with bound error

$$\min_{A_0,\epsilon} \left(|\Delta A_\epsilon(\Delta)| \leq \epsilon \; \forall \; \Delta\right), \;\; \Delta A_\epsilon(\Delta) = A_\epsilon(\Delta) - A_0(\Delta) \tag{2.58}$$

where $A_0(\Delta)$ is a proper acceptance function (2.49). The upper bound error ϵ in the acceptance probability corresponds to an upper bound in the total variation distance[10]

$$D_v[S_0, S_\epsilon] \leq \frac{\epsilon}{1-\eta}, \tag{2.59}$$

where $S_0 = p(\theta|X)$ and S_ϵ are the stationary distributions of the unbiased and biased samplers, respectively [38]. The upper bound (2.59) depends on the rate of convergence of the unbiased sampler

$$D_v[P\mathscr{T}_0, S_\epsilon] \leq \eta D_v[P, S_0], \tag{2.60}$$

where P is an arbitrary distribution and $\mathscr{T}_0$ denotes the transition operator of the unbiased MCMC sampler.

Now follows a proof of (2.59), which is a correction of the proof in [38][11]. Recall (2.61) defines the transition operator

$$\mathscr{T}_\epsilon(\theta'|\theta) = q(\theta'|\theta)A_\epsilon(\Delta(\theta',\theta)) + \left(1 - \int dq(\theta'|\theta)A_\epsilon(\Delta(\theta',\theta))\right)\delta(\theta'-\theta). \tag{2.61}$$

Consider the distance between a single step of the biased and unbiased sampler

[10] The total variation distance is defined as $D_v[p,q] = \frac{1}{2}\int |p(\theta) - q(\theta)| d\Omega(\theta)$, where p and q are Radon-Nikodym derivatives w.r.t. the measure Ω.

[11] The transition operator (2.61) used in the original derivation has an incorrect rejection term. The derived bounds are not affected.

$$\begin{aligned}
&D_v[P\mathcal{T}_0, P\mathcal{T}_\epsilon] = \frac{1}{2}\int \left| \int \left[\mathcal{T}_0(\theta'|\theta) - \mathcal{T}_\epsilon(\theta'|\theta)\right] dp(\theta) \right| d\Omega(\theta') \\
&= \frac{1}{2}\int \left| \left[\Delta A_\epsilon(\theta',\theta)q(\theta'|\theta) - \left(\int \Delta A_\epsilon(\theta'',\theta)dq(\theta''|\theta)\right)\delta(\theta'-\theta)\right] dp(\theta) \right| d\Omega(\theta') \\
&\leq \frac{1}{2}\epsilon \int \left[q(\theta'|\theta) + \delta(\theta'-\theta)\right] dp(\theta)d\Omega(\theta') = \epsilon.
\end{aligned} \tag{2.62}$$

Let $P_\epsilon^{(t+1)} = P_\epsilon^{(t)}\mathcal{T}_\epsilon$ and let $P_\epsilon^{(0)}$ be the initial distribution of the sampler. The triangle inequality $D_v[A,C] \leq D_v[A,B]D_v[B,C]$, together with (2.60) and (2.62) yields

$$D_v[P_\epsilon^{(t+1)}, S_0] \leq D_v[P_\epsilon^{(t+1)}, P_\epsilon^{(t)}\mathcal{T}_0] + D_v[P_\epsilon^{(t)}\mathcal{T}_0, S_0] \leq \epsilon + \eta D_v[P_\epsilon^{(t)}, S_0]. \tag{2.63}$$

For $D_v[P_\epsilon^{(t)}, S_0] \geq \epsilon/r$ with $0 \leq r < 1-\eta$, the biased sampler converges towards the stationary distribution

$$D_v[P_\epsilon^{(t+1)}, S_0] \leq (r+\eta)D_v[P_\epsilon^{(t)}, S_0]. \tag{2.64}$$

Let t_r be the first time instance for which $D_v[P_\epsilon^{(t_r)}, S_0] < \epsilon/r$ and note t_r is finite due to (2.64). Then for $t > t_r$ also $D_v[P_\epsilon^{(t)}, S_0] < \epsilon/r$ because

$$D_v[P_\epsilon^{(t+1)}, S_0] \leq \epsilon + \eta\frac{\epsilon}{r} < \frac{\epsilon}{r}. \tag{2.65}$$

The upper bound in (2.59) is obtained by taking the limit $r \to 1-\eta$.

Unfortunately, the value of η and therefore the upper bound is unknown in practice [38]. The upper bound does motivate the design of a sampler which minimises the maximum error ϵ in the acceptance function and maximises the convergence rate η.

Noise adaptive acceptance test

The MH hypothesis test is efficient when the variance of the posterior ratio σ^2 is large and reduces to a full acceptance test when $\tilde{\Delta}$ is arbitrary close to the decision boundary u. The stochastic Barker test works well irrespective of the decision boundary. However, it can only be used if $\sigma^2 < \pi^2/3$ and might therefore require large batches in practise. Additionally, the Barker acceptance function is less efficient which might increase the bias in the stationary distribution due to slow convergence (2.59).

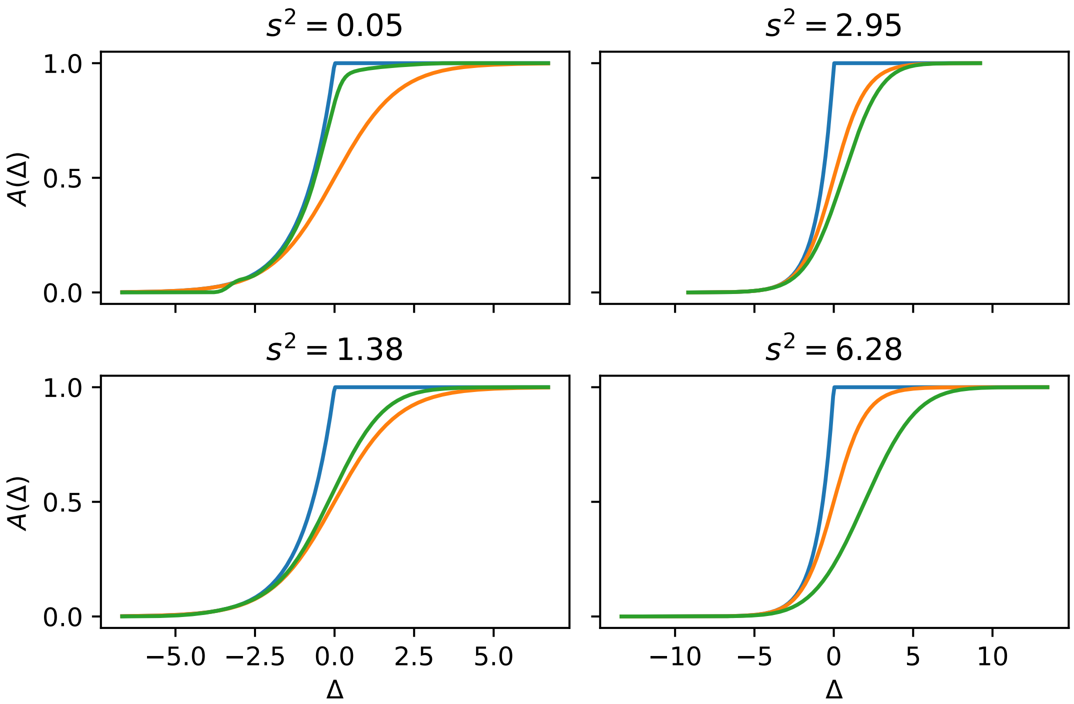


Fig. 2.2 Stochastic acceptance functions for various values of s^2. The acceptance probability of the stochastic acceptance function $\tilde{A}(\Delta)$ (green) is shown together with the MH (blue) and Barker (orange) acceptance functions.

Consider an inference task where it is assumed that the log likelihood ratio follows a Gaussian distribution $\log P(X_i|\theta')/P(X_i|\theta) \sim \mathcal{N}(\Delta_l, s^2)$ where $s^2 = \mathrm{Var}[log P(\tilde{X}_i|\theta')/P(\tilde{X}_i|\theta)]$ and $\tilde{X}$ is a mini-batch of size n. The variance of the estimate $\sigma^2 = N^2 s^2/n$.

The distribution of the sample variance depends on the proposed and current parameters and is reversible

$$p(s^2|\Theta = \theta, \Theta' = \theta') = p(s^2|\Theta = \theta', \Theta' = \theta). \tag{2.66}$$

Let $\tilde{A}_{s^2}(\tilde{\Delta})$ be a stochastic acceptance function for an estimated posterior ratio $p(\tilde{\Delta}) = \mathcal{N}(\Delta, s^2)p(s^2)$. The acceptance function condition (2.49) reduces to

$$\mathbb{E}_{\mathcal{N}(\tilde{\Delta};\Delta,s^2)}[\tilde{A}_{s^2}(\tilde{\Delta})] = \mathbb{E}_{\mathcal{N}(\tilde{\Delta};\Delta,s^2)}[\tilde{A}_{s^2}(-\tilde{\Delta})]\exp(\Delta). \tag{2.67}$$

The stochastic acceptance function $\tilde{A}_{s^2}(\tilde{\Delta}, s^2)$ is restricted to be of the form

$$\tilde{A}_{s^2}(\tilde{\Delta};\theta) = P(\tilde{\Delta} + \xi > 0), \quad p(\xi) = \sum_i p_i(\theta)\delta(\xi - \xi_i). \tag{2.68}$$

The values of ξ_i are fixed on a linear space and the vector $p(\theta) = \text{Softmax}(\theta)$, where θ is optimised numerically using the objective

$$\begin{aligned} \mathscr{L}(\theta) &= \left\| \mathbb{E}_{\tilde{\Delta}}[A_{s^2}(\tilde{\Delta};\theta)] - \hat{A}(\Delta;\theta) \right\|_2 - \lambda \mathbb{E}_{\Delta,\tilde{\Delta}}[A_{s^2}(\tilde{\Delta};\theta)], \\ \hat{A}(\Delta;\theta) &= \begin{cases} \mathbb{E}_{\tilde{\Delta}}[A_{s^2}(\tilde{\Delta};\theta)] & ,\Delta >= 0 \\ \hat{A}(-\Delta)\exp(-\Delta) & ,\Delta < 0 \end{cases}. \end{aligned} \tag{2.69}$$

The expectations in (2.69) are approximated by discretising $p(\Delta)$ and $p(\tilde{\Delta}|s^2)$ into finitely many equally sized intervals. The error w.r.t. a proper acceptance function $\hat{A}(\Delta;\theta)$ is reduced by an L2 norm while simultaneously penalising inefficient acceptance functions with a weighting factor λ.

Fig. 2.2 shows how the acceptance functions changes as a function of the noise s^2. As the noise vanishes the stochastic acceptance function reduces to the MH acceptance function. The Barker test is almost optimal if the noise is slightly less than $\pi^2/3$. The acceptance function becomes less efficient as the noise increases.

2.8 Bayesian Dark Knowledge

The posterior predictive of a Bayesian Neural Network can be predicted using the samples generated by SGHMC. Unfortunately, the estimator has high variance due to the complexity of the posterior and the correlation between samples.

The Bayesian Dark Knowledge algorithm aims to reduce the variance of the posterior predictive estimator by training a neural network to mimic the posterior predictive. Once the network is trained, only a single forward pass through the model is needed to evaluate the approximate posterior predictive [39]. This method is similar to student teacher training in ensemble methods [24] where the posterior takes the role of the teacher and newly trained neural network is the student.

The student model $q_\phi(y)$ is optimised by minimising the KL-divergence between the posterior predictive $p(y|X)$ and the student predictions

$$\begin{aligned} \mathscr{L}(\phi) &= \text{KL}[p(y|X)|q_\phi(y)] \\ &= -\mathbb{E}_{p(\theta|X)p(y|\theta)}\left[\log q_\phi(y)\right] + C. \end{aligned} \tag{2.70}$$

Typically the prediction y is conditioned on some context or input $\hat{x}$ in which case the loss becomes an expected KL divergence over the input space

$$\begin{aligned}\mathscr{L}(\phi) &= \mathbb{E}_{p(\hat{x})}\left[\mathrm{KL}[p(y|X,\hat{x})|q_{\phi}(y|\hat{x})]\right] \\ &= -\mathbb{E}_{p(\hat{x})p(\theta|X)p(y|\theta,\hat{x})}\left[\log q_{\phi}(y|\hat{x})\right] + C.\end{aligned} \tag{2.71}$$

Thus, minimising the KL divergence is equivalent to minimising the cross entropy. Because the student objective is to mimic the posterior predictive a small divergence is expected to imply good calibration properties in practise. The inclusive KL divergence does however assign a smaller loss to underconfident approximations compared to overconfident approximations. Consequently, a student network which is not complex enough to faithfully mimic the posterior predictive is expected to produce underconfident predictions.

Chapter 3

Experiments

The experiments conducted as part of this work are summarised in this chapter. Sec. 3.1 describes the calibration plots used for diagnosing the calibratedness of the considered models. First we show the performance of the main algorithms in a toy regression task (Sec. 3.2). Sec. 3.3 and Sec. 3.4 describe the performance of all methods described in Chap. 2 on the MNIST and FashionMNIST tasks, respectively. Finally, Sec. 3.5 shows how online pruning affects training, accuracy, and calibration.

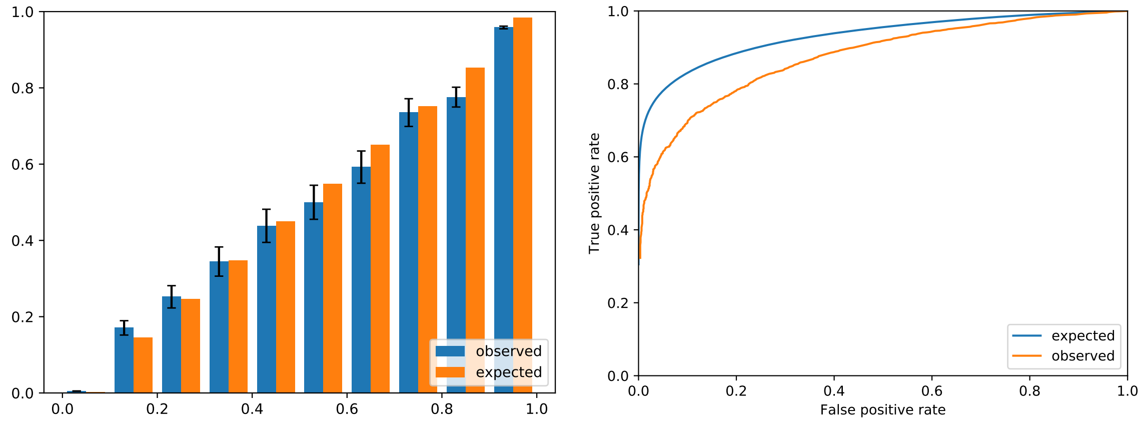
All experiments and algorithms have been implemented in PyTorch [53]. This framework was chosen because it supports GPU computation and Automatic Reverse-mode Differentiation[1]. We did not make use of existence codebases for the implementations of the described algorithms. This reduces the risk that differences in results are caused by implementation details or differences in the amount of effort taken to tune hyper-parameters.

All hyper-parameters were tuned by hand unless explicitly mentioned. Most experiments took between one and two hours to execute on a single K80 GPU. Due to limited computational perform it was impossible to use a more methodical approach like Bayesian Optimisation [62].

3.1 Calibration plots

This chapter uses a variety of approaches to empirically analyse the calibration properties of probabilistic models. Despite the recent attention in the Machine Learning research for calibrated and Bayesian uncertainty approaches, empirical evidence for the quality of uncertainty is rarely provided.

[1] Reverse-mode differentiation is more commonly referred to as back-propagation in ML literature



(a) Bins

(b) ROC

Fig. 3.1 Example of a poorly calibrated Neural Net on MNIST

A simple check for the quality of uncertainty estimates is the expected vs. observed test accuracy. The expected accuracy for a test set $X = \{x^{(i)}\}_{i=1}^{N}$ is given by

$$\frac{1}{N}\sum_{i}^{N}\mathbb{E}[\mathbb{1}(\omega^{(i)} = \hat{\omega}^{(i)})] = \frac{1}{N}\sum_{i=1}^{N} p(\hat{\omega}^{(i)}|x^{(i)}), \tag{3.1}$$

where $\mathbb{1}(.)$ is the indicator function and $\hat{\omega}^{(i)} = \arg\max_{\omega^{(i)}} p(\omega^{(i)}|x^{(i)})$ is the most likely class of $x^{(i)}$. The posterior predictive $p(\omega^{(i)}|x^{(i)})$ is intractable for VI methods on Bayesian Neural Nets. Therefore, a MC estimate is used instead

$$p(\omega|x) \approx \frac{1}{n}\sum_{i}^{M} p(\omega|x,\theta_i), \quad \theta_i \sim q_{\phi}(\theta). \tag{3.2}$$

For the experiments in this report $M = 30$ was found to give good results with little variance.

3.1.1 Calibration Bins

The method can be generalised by considering only the predictions in a certain interval. The observed accuracy in an interval $a < p < b$ is defined as

$$A(a,b) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\mathbb{1}(\omega^{(i)} = w_j, a < p(\omega_j|x^{(i)}) < b)}{\sum_{i=1}^{N}\sum_{j=1}^{M}\mathbb{1}(a < p(\omega_j|x^{(i)}) < b)}. \tag{3.3}$$

Taking the expectation with respect to model predictions yields the expected accuracy

$$\mathbb{E}[A(a,b)] = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} p(\omega_j|x^{(i)})\mathbb{I}(a < p(\omega_j|x^{(i)}) < b)}{\sum_{i=1}^{N}\sum_{j=1}^{M} \mathbb{I}(a < p(\omega_j|x^{(i)}) < b)}. \tag{3.4}$$

In order to determine of the expected accuracy significantly deviates from the observed accuracy it is useful to consider the expected variance of the accuracy

$$\mathrm{Var}[A(a,b)] = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} p(\omega_j|x^{(i)})\left[1 - p(\omega_j)|x^{(i)})\right]\mathbb{I}(a < p(\omega_j|x^{(i)}) < b)}{\left(\sum_{i=1}^{N}\sum_{j=1}^{M} \mathbb{I}(a < p(\omega_j|x^{(i)}) < b)\right)^2}. \tag{3.5}$$

Other authors [46] have plotted the observed accuracy as a line together with $\mathbb{E}[A(a,b)] \approx (a+b)/2$, which approximates the true expected accuracy as $b-a \to 0$. Most predicted class probabilities in the image recognition tasks considered in this report are $p > 0.9$ or $p < 0.1$. This causes the variance of the accuracy to be high for $0.1 < a < b < 0.9$. Consequently, the error bounds are essential for the interpretation of binned calibration plots even when a large test set is available.

Fig. 3.1a shows an example of the expected and observed accuracy together with 95% confidence bounds. The model is a single sample from the posterior of a CNN trained on MNIST. The observed accuracy is significantly less than the expected accuracy in most confidence ranges. Thus, this is an example of a poorly calibrated model.

3.1.2 Calibration ROC

ROC plots are a commonly used diagnostic for the quality of predictions for classification tasks. A ROC plot shows the True Positive Rate (TPR) vs. the False Positive Rate (FPR) as a function of positive classification threshold τ. For a classification tasks with more than two classes the positive case is considered to be correct classification and the negative misclassification

$$\begin{aligned}\mathrm{TPR}(\tau) &= \frac{\sum_{i=1}^{N}\mathbb{I}(\omega^{(i)} = \hat{\omega}^{(i)}, p(\hat{\omega}^{(i)}|x^{(i)}) > \tau)}{\sum_{i=1}^{N}\mathbb{I}(\omega^{(i)} = \hat{\omega}^{(i)})},\\ \mathrm{FPR}(\tau) &= \frac{\sum_{i=1}^{N}\mathbb{I}(\omega^{(i)} \neq \hat{\omega}^{(i)}, p(\hat{\omega}^{(i)}|x^{(i)}) > \tau)}{\sum_{i=1}^{N}\mathbb{I}(\omega^{(i)} \neq \hat{\omega}^{(i)})}.\end{aligned} \tag{3.6}$$

The expected False and True Positive rates are given by

$$
\begin{aligned}
\mathbb{E}[\mathrm{TPR}(\tau)] &= \frac{\sum_{i=1}^{N} p(\hat{\omega}^{(i)})|x^{(i)})\mathbb{I}(p(\hat{\omega}^{(i)}|x^{(i)}) > \tau)}{\sum_{i=1}^{N} \mathbb{I}(\omega^{(i)} = \hat{\omega}^{(i)})}, \\
\mathbb{E}[\mathrm{FPR}(\tau)] &= \frac{\sum_{i=1}^{N} p(\hat{\omega}^{(i)})|x^{(i)})[1 - p(\hat{\omega}^{(i)})|x^{(i)})]\mathbb{I}(p(\hat{\omega}^{(i)}|x^{(i)}) > \tau)}{\sum_{i=1}^{N} \mathbb{I}(\omega^{(i)} \neq \hat{\omega}^{(i)})}.
\end{aligned}
\tag{3.7}
$$

A tradeoff between the TPR and FPR can be made based on the ROC curve, which can be interpreted as a post-calibration procedure. Within the framework of Bayesian decision theory such a tradeoff should be quantified in a utility function

$$
U_{roc}(\omega, \hat{\omega}, a) = a\left[\mathbb{I}(\omega = \hat{\omega}) - \beta\mathbb{I}(\omega \neq \hat{\omega})\right], \tag{3.8}
$$

where $a = 1$ is the action corresponding to positive classification. Maximising the expected utility yields the optimal decision policy

$$
\begin{aligned}
\hat{a} &= \arg\max_{a} \mathbb{E}_{\omega}[U_{roc}(\omega, \hat{\omega}, a)] \\
&= \arg\max_{a} a\left[(1 + \beta)p(\hat{\omega}|x) - \beta\right] \\
&= \mathbb{I}\left(p(\hat{\omega}|x) > \tau = \frac{\beta}{1 + \beta}\right).
\end{aligned}
\tag{3.9}
$$

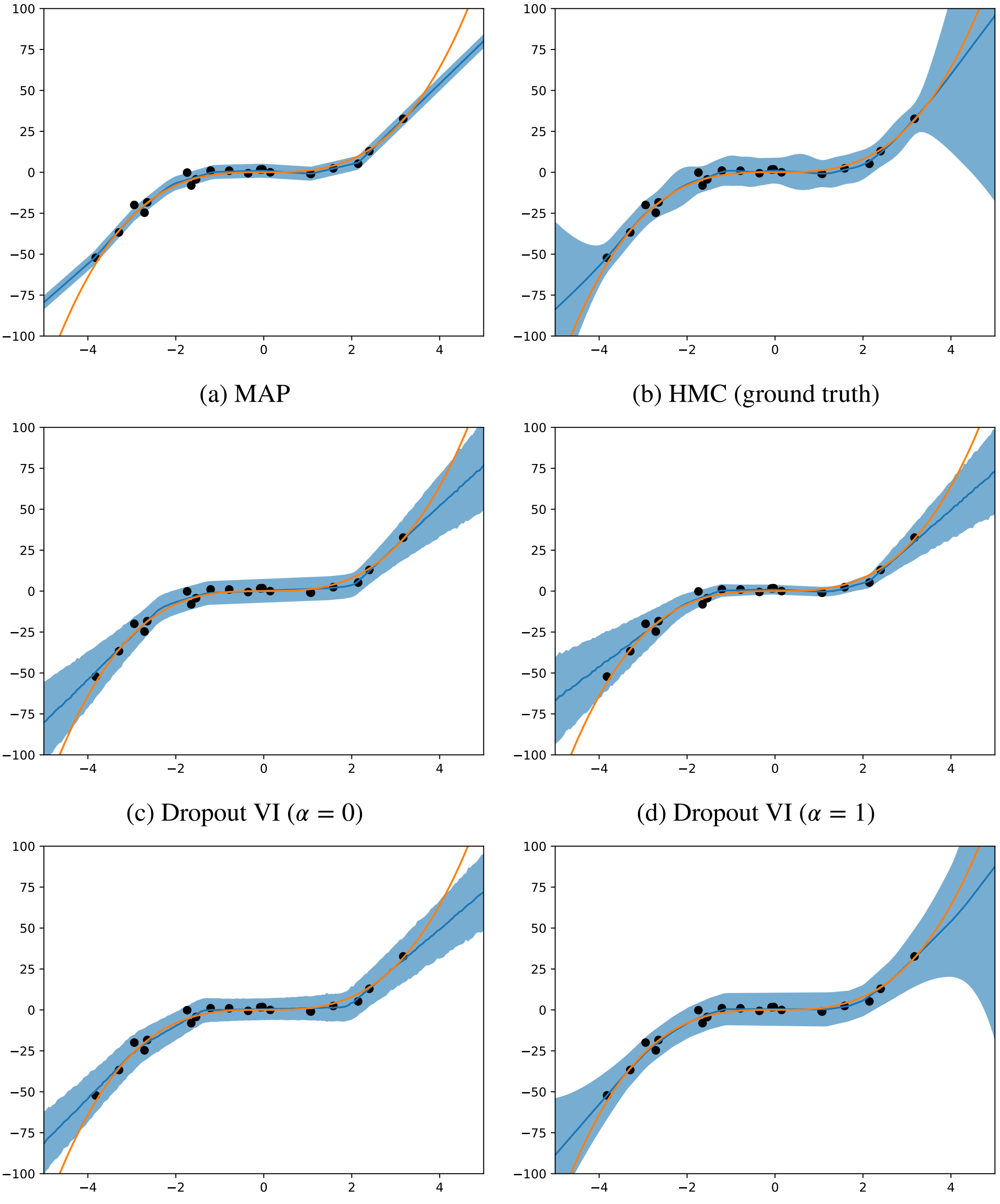
Substituting the optimal policy (3.9) into (3.8) and considering the entire test set yields

$$
\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\omega^{(i)}}[U_{roc}(\omega^{(i)}, \hat{\omega}^{(i)}, \hat{a})] = \mathbb{E}[\mathrm{TPR}(\tau) - \beta\mathrm{FPR}(\tau)] \tag{3.10}
$$

In the Bayesian decision framework, a tradeoff between the TPR and FPR leads to a classification threshold τ based on the expectations of the TPR and the FPR. If the uncertainty estimates are poor, the Bayesian policy might be sub-optimal compared to the policy based on the observed TPR and FPR. Consequently, it is beneficial for a classifier to have an expected ROC curve that closely matches the observed ROC curve. This way a TPR/FPR tradeoff can be made without the need of a validation dataset.

Fig. 3.1b shows an example of a calibration ROC plot which shows significantly overconfident predictions. Another advantage of the calibration ROC plot is that systematic underconfidence/overconfidence will accumulate which is not the case in the calibration bins plot.

3.2 Toy regression



(a) MAP

(b) HMC (ground truth)

(c) Dropout VI ($\alpha = 0$)

(d) Dropout VI ($\alpha = 1$)

(e) Bayes By Backprop

(f) Bayesian Dark Knowledge

Fig. 3.2 Bayesian Regression with uncertainty estimates

Sampling, VI, and traditional MAP algorithms were used on a toy regression task equivalent to the one used in [22, 39]. The dataset consists of $N = 20$ points i.i.d. from $p(x) = \mathscr{U}[-4, 4]$ and $p(y|x) = \mathscr{N}(y; x^3, 9)$. The model used is a fully connected neural network with a single hidden layer of 100 units with ReLU non-linearity. The output layer contains a single layer used as the mean such that the likelihood is $\mathscr{N}(y; f(x), \sigma^2)$ where the variance σ^2 is a model parameter.

Fig. 3.2 shows the line of the generating function without noise $y = x^3$ and the posterior predictive with 95% uncertainty bounds. Because the dataset is small, a HMC sampler with Metropolis-Hastings step can be used to sample from the posterior. The uncertainty estimates given by the HMC sample (3.2b) are considered to be the ground-truth for true posterior predictive.

The MAP solution (3.2a) clearly fails to capture any meaningful estimate of uncertainty outside of densely sampled regions. Dropout VI (Fig. 3.2c & 3.2d) approximates the true uncertainty reasonably well. However, the type of α-divergence seems to have little effect on the uncertainty estimates. The uncertainty estimates did not vary much with the dropout rate. The default dropout probability $p = 0.5$ worked well.

Bayesian Dark Knowledge (3.2f) produced uncertainty estimates that almost perfectly match those of HMC. The uncertainty estimates are slightly underconfident as expected from the inclusive KL loss. The network correctly learns to increase uncertainty as it extrapolates further away from the training data but does not have the capacity to increase uncertainty during interpolation between the larger gaps within the training set.

Bayes By Backprop (3.2e) learns reasonable uncertainty estimates similar to those of Dropout VI. However, BBB did not converge unless the prior was set carefully and the variance was fixed to the true value $\sigma^2 = 3$. Fig. 3.3 shows the negative result obtained when BBB is optimised with a learned variance which is similar to the results reported for BBB earlier [22, 39]. When σ^2 is optimised as a model parameter the data term in the ELBO (2.8) becomes too small compared to the regulariser term resulting in a variational distribution that converges towards the prior.

Looking closely at the results for BBB reveals that the mean function is essentially a piecewise function consisting of 4 almost linear pieces with smoothed out discontinuities. Inspection of the hidden layer's weight distribution reveals that ELBO optimisation forces most of the weights to have a small mean with large variance and a large negative bias with large variance causing the activations of these neurons to be zero with high probability and the entropy of the variational distribution term to be large. The ELBO objective thus leads to (over-)pruning of features/filters which was first observed in [64]. The ELBO increases due to large variance in weights that cause no variance in the activation or output of the network.

This result motivated the use of compression priors in subsequent experiments as these high variance weights were found to destabilise training for more complex models.

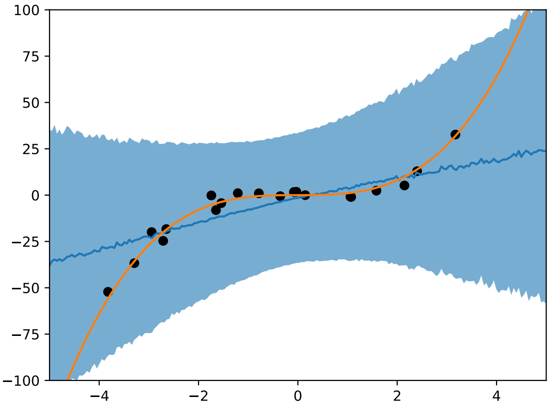


Fig. 3.3 Bayes By Backprop with trained noise

3.3 MNIST

α	Observed error (%)	Expected error (%)
0	**0.46**	0.50
0.5	0.50	0.40
1.0	0.52	0.46

Table 3.1 Observed and expected test accuracies on MNIST for various α-divergences

	Observed err. (%)	Expected err. (%)	τ
Dropout ($p = 0.35$)	0.46	0.50	
Dropout ($p = 0.5$)	**0.44**	0.73	
Gaussian Dropout	0.81	0.74	0.33
BBB (compress)	0.79	1.4	0.25
BBB (constraint)	0.67	0.68	1
Dropout-BBB (compress)	0.86	0.98	0.25
Dropout-BBB (constraint)	0.61	0.56	1
Gauss-Gamma	0.52	0.63	0.01
Bayesian Compression	0.58	0.65	0.1
Bayesian Dark Knowledge	0.57	0.70	
Bayesian Dark Knowledge (accept)	0.69	0.57	

Table 3.2 Observed and expected test accuracies on MNIST

For MNIST a Convolutional architecture was used with two convolution layers with 32 and 64 output channels, respectively. After each convolution max pooling is used. The output of the convolutions is fed into a fully connected network with a single hidden layer of 500 features. The ReLU activation function is used after each hidden layer in the network. Tbl. 3.2 shows the test set error together with the expected error.

Despite the lack of posterior concentration (Sec. 2.2.2), Dropout VI is found to have the lowest test error and an expected error that matches the observed error closely. Fig. 3.5a and Fig. 3.6a show that the model is almost perfectly calibrated for the given calibration tests. The dropout rate was optimised such that the deviation between observed and expected error was minimal. Dropout was only applied after the second convolutions and the fully connected hidden layer. Because the dropout rates cannot be optimised using Gradient Ascend directly, a grid search was performed with a granularity of 0.05. With the ”standard” dropout rate $p = 0.5$ the model is underconfident (Fig. 3.7a & 3.7b).

Dropout VI was also tested using various α-divergences. The results are reported in Tbl. 3.1. For $\alpha > 0$ the inferred model is overconfident whereas it is underconfident for

$\alpha = 0$. This result contradicts the intuition that approximating the inclusive KL should result in a less concentrated approximation and consequently less confident approximations. The calibration plots (Fig. 3.4) confirm that for $\alpha = 0.5$ and $\alpha = 1.0$ the inferred models is slightly too confident for $p > 0.9$ in particular.

Gaussian dropout was found to result in significantly underconfident predictions unless the KL term is discounted by some factor τ. This agrees with earlier observation that without discounting the regularisation term Gaussian dropout tends to underfit the data [35]. In this case $\tau = 1/3$ resulted in fairly good results.

Training the original BBB algorithm was found to be strongly sensitive to hyper-parameters when calibration is concerned. The variance of the weights continues to increase leading to underconfidence and poor generalisation. Unless the regularisation term is discounted, the variances are initialised to a small value, and/or the training is stopped early.

(Dropout-)BBB was trained in two variations. A constraint version which used the fact that the variance tend to stay small when the variances are initialised at a small value and trained with SGD. The variances steadily increase during the 100 training epochs but does not reach an optimum. When training with Adam or using a much larger step size for the variances, the ELBO will be an order of magnitude larger but the model finds a very poor fit of the data with low test accuracy as a result. Thus, the constraint BBB variant relies on early stopping to keep the variational distribution from containing too much noise. The prior for the constraint model $p(\theta) = \mathcal{N}(0, 10)$.

The compression variant does not constrain the optimisation and instead uses a KL discount $\tau = 1/4$ and compression priors to make sure a good approximation can be found without relying on early stopping. The constraint approach does however show lower test error and better calibration plots (Fig. 3.8c & 3.9c) compared to the unconstrained approach with compression priors (Fig. 3.8a & 3.9a). The compression prior is defined according to (2.40) with $\sigma = \{10^3, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}\}$ and $T = 0.1$.

The Dropout-BBB yields a model with better calibration properties compared to BBB (Fig. 3.8d & 3.9d). However, maximising the ELBO directly still results in too much noise and unconfined predictions. Without compression priors or constraint optimisation the model did not learn to use dropout noise and simply increased the independent noise for (almost) every weight. Using compression priors results in underconfidence (Fig. 3.8b & 3.9b) and highest observed error of all methods.

We did not find a significant difference in performance between the local and global parameterisation of BBB. However, the cost of sampling Gaussian noise for each weight far outweighs the cost of computing the additional convolution or vector matrix product required to determine the variance of an activation. Therefore, the reported results for (Dropout-)BBB use global parameterisation. For the compression variants the KL term cannot be computed analytically. This further increases the performance gap between global and local because the sampled weights is reused when sampling the regularisation term.

Gauss-Gamma VI is the best performing proper method although it requires significant KL discounting $\tau = 0.01$. Gauss-Gamma results in slightly underconfident uncertainty estimates (Fig. 3.5c & 3.6c).

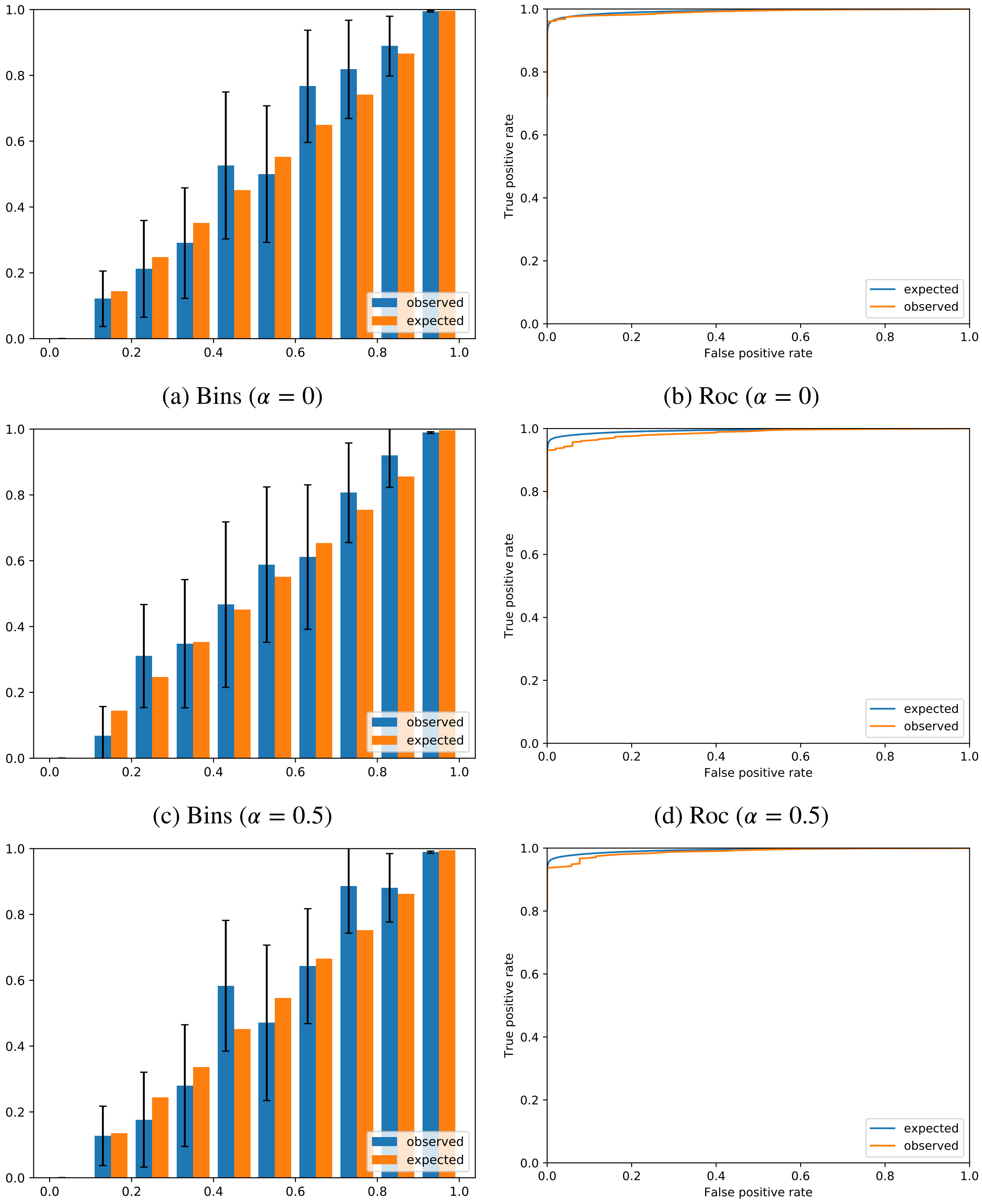
Bayesian Compression is competitive with Gauss-Gamma and outperforms Gaussian Dropout and BBB variants despite converging to a significantly smaller network (see Sec. 3.5). The model predictions are slightly more underconfident (Fig. 3.5d & 3.6d) compared to Gauss-Gamma with significantly smaller KL discount factor $\tau = 0.1$.

Guass-Gamma VI and Bayesian Compression are the methods which could be successfully trained using the Adam optimiser [34] and have adaptive amounts of noise. The Adam optimiser reweighs the gradients such that the amount of noise increases rapidly even when the variance is initialised at a small value.

Sampling approaches using Bayesian Dark Knowledge performs slightly worse than the dropout models but better than the Gaussian VI methods in terms of test accuracy and calibration. First a sampler without acceptance function was simulated for 50 epochs to find a good starting point for the Markov Chain. The student was trained in 100 epochs using all samples from the Markov Chain. The binned calibration test (Fig. 3.5e) shows no significant miscalibration. However, The expected test accuracy and ROC plot (Fig. 3.6e) shows that the approximation of the posterior predictive is slightly underconfident.

Using a stochastic acceptance step did not result in better performance. The learning rate had to be reduced by a factor 10 for samples to be accepted at a reasonable rate. Furthermore, rejection causes the chain to move more slowly due to additional momentum updates. Consequently, the sampler with acceptance step produces samples with much higher correlation. This likely caused the student to mimic the behaviour of just a small region of the posterior.

We also considered the posterior predictive accuracy based on a MC estimate of the posterior predictive. The sample consisted of the weights obtained after each training epoch. The resulting posterior predictive estimate did not result in a competitive test error. The uncertainty estimates are significantly underconfident without acceptance function and extremely overconfident with acceptance test.



(a) Bins ($\alpha = 0$)

(b) Roc ($\alpha = 0$)

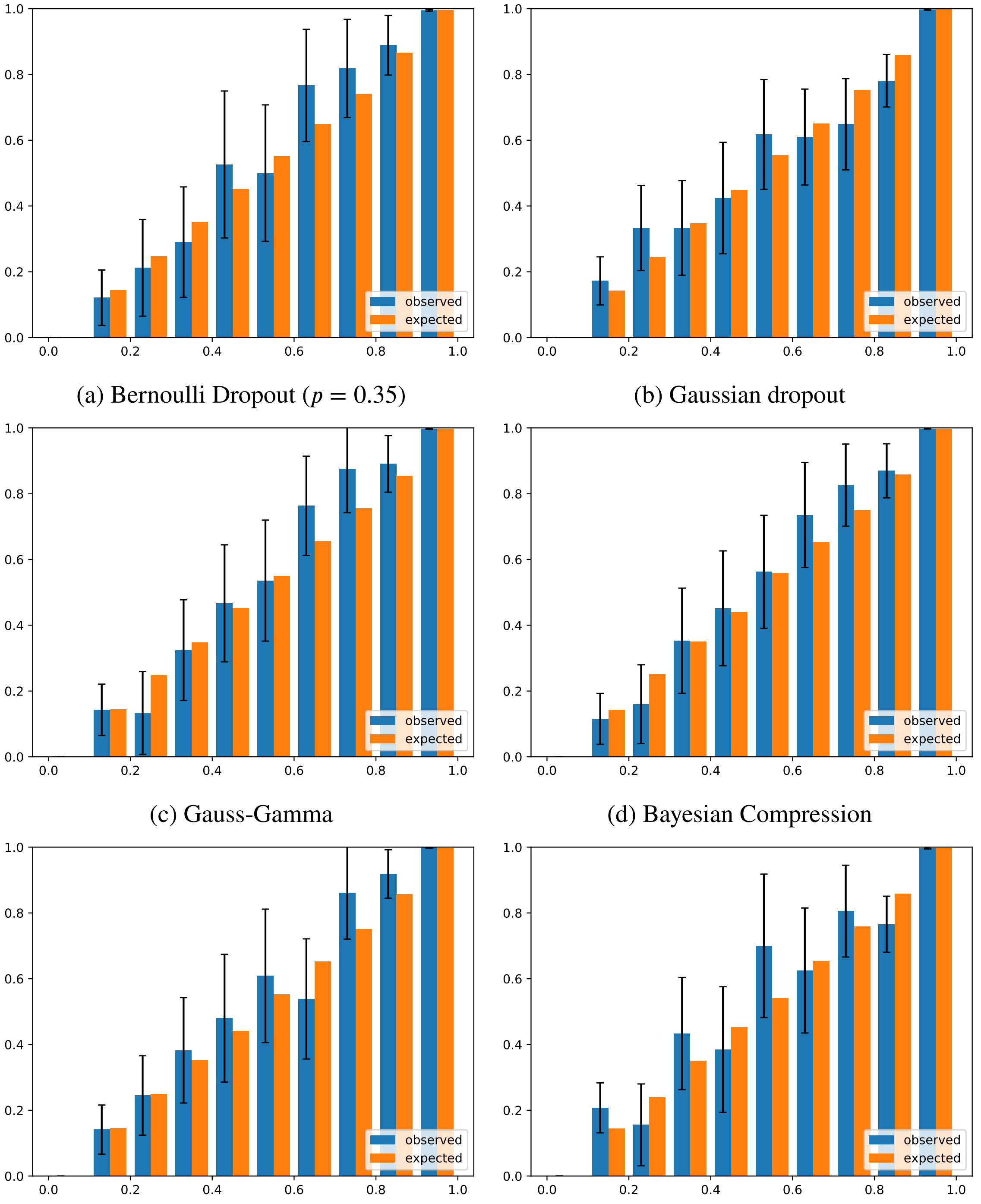
(c) Bins ($\alpha = 0.5$)

(d) Roc ($\alpha = 0.5$)

(e) Bins ($\alpha = 1$)

(f) Roc ($\alpha = 1$)

Fig. 3.4 Calibration bins for MNIST classification for various α-divergences



(a) Bernoulli Dropout ($p = 0.35$)

(b) Gaussian dropout

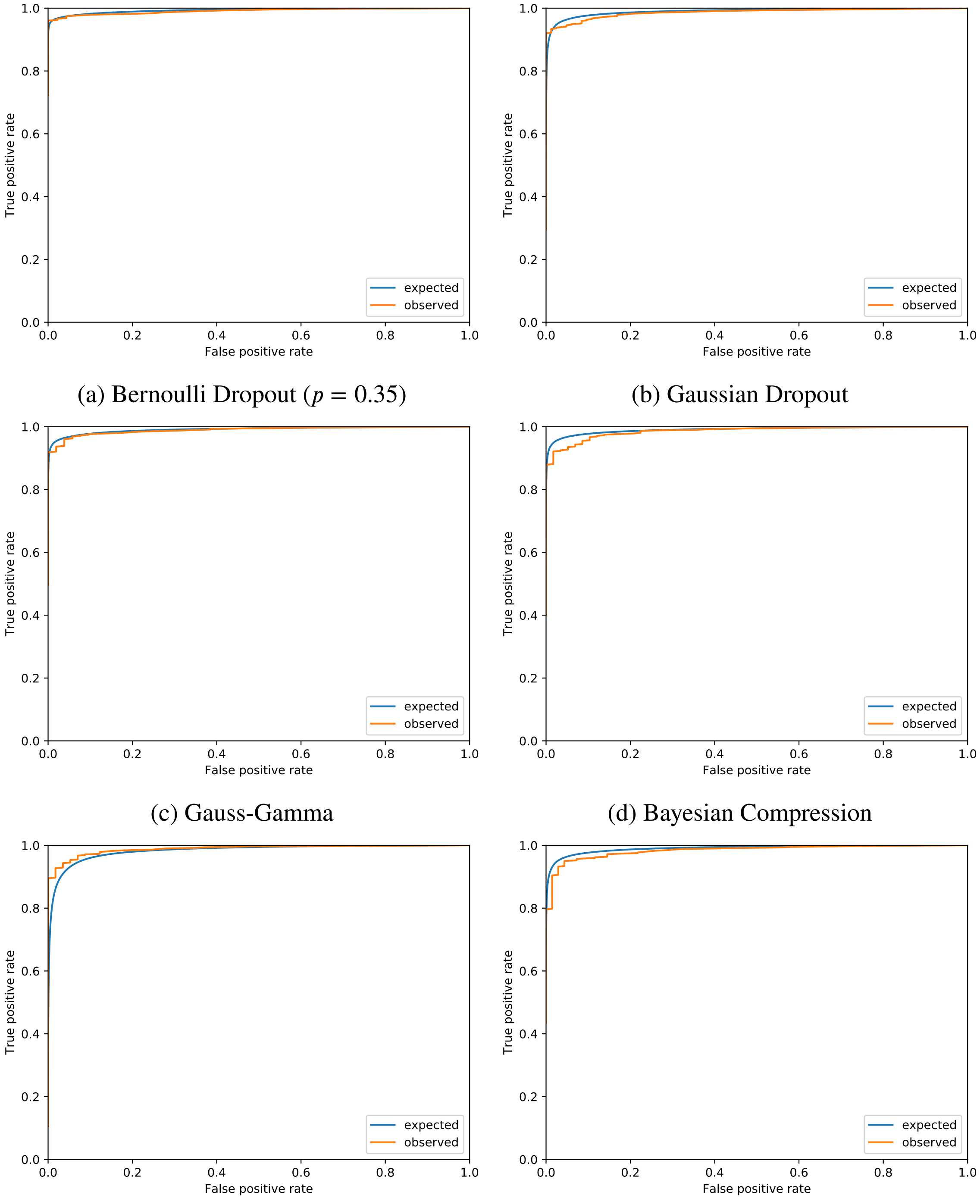
(c) Gauss-Gamma

(d) Bayesian Compression

(e) Bayesian Dark Knowledge

(f) Bayesian Dark Knowledge (accept function)

Fig. 3.5 Calibration bins for MNIST classification



(a) Bernoulli Dropout ($p = 0.35$)

(b) Gaussian Dropout

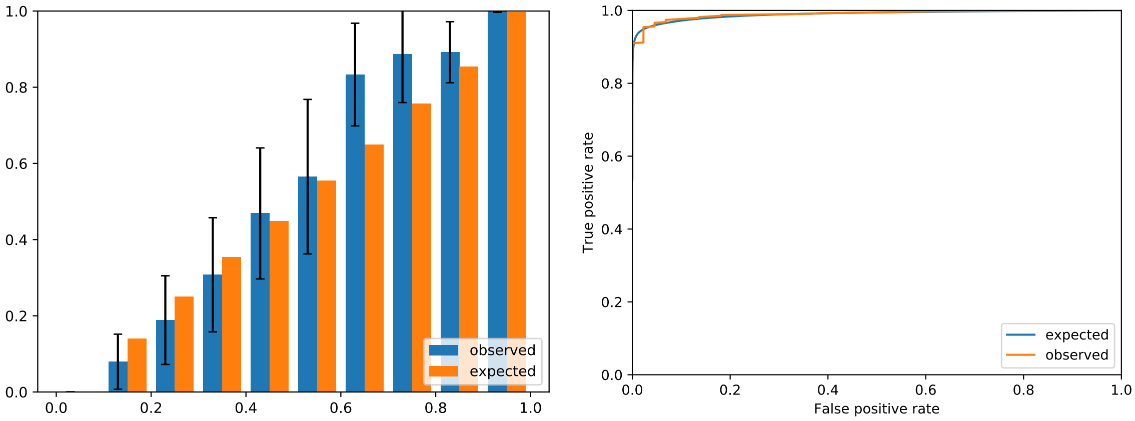
(c) Gauss-Gamma

(d) Bayesian Compression

(e) Bayesian Dark Knowledge

(f) Bayesian Dark Knowledge (accept function)

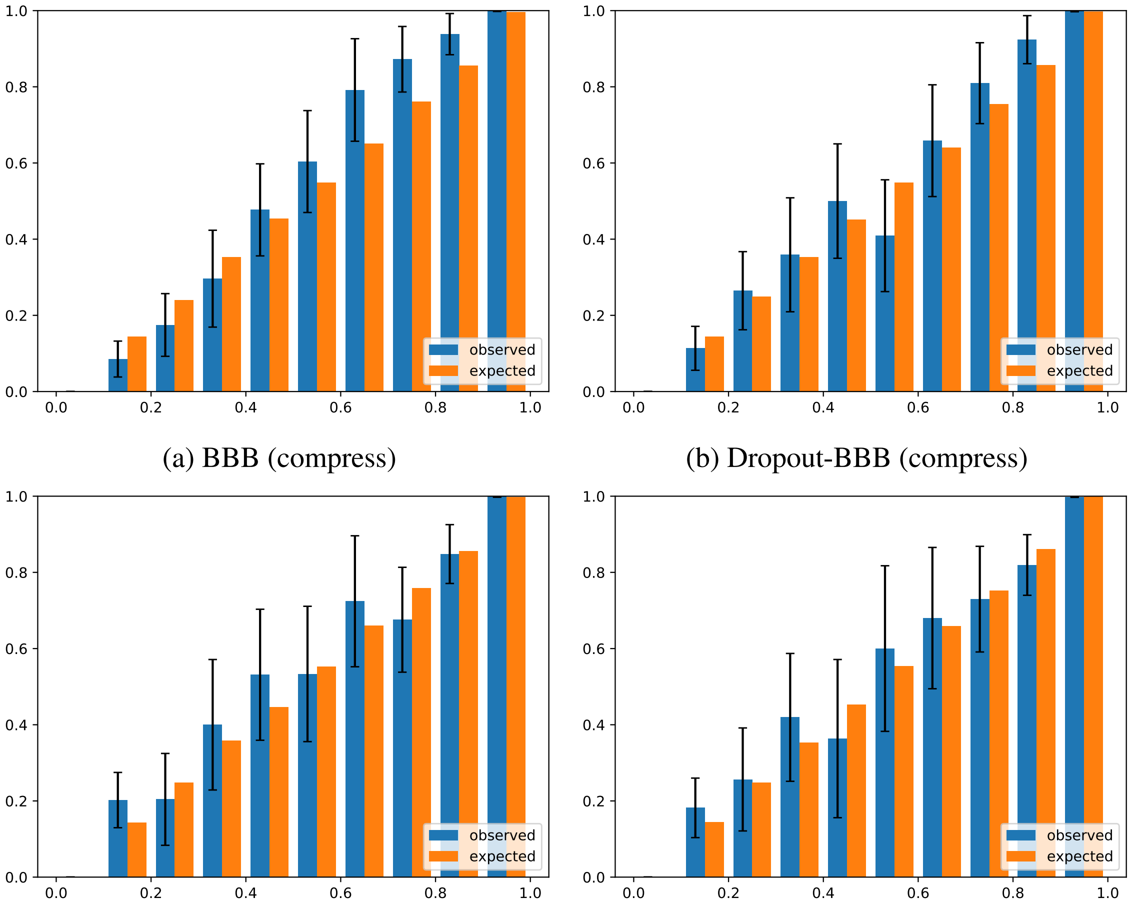
Fig. 3.6 Calibration ROC plots for MNIST classification



(a) Bins

(b) ROC

Fig. 3.7 Calibration ROC for MNIST classification with Dropout VI ($p = 0.5$)



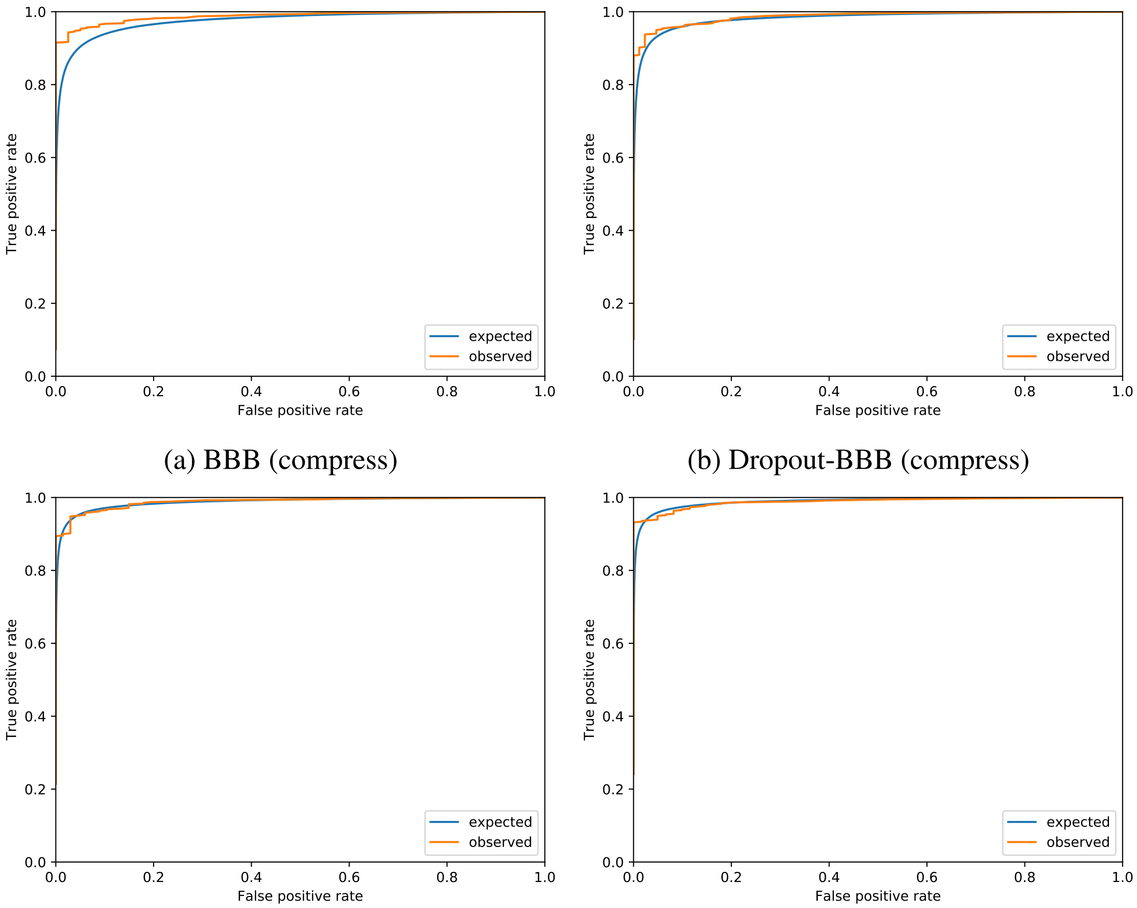
(a) BBB (compress)

(b) Dropout-BBB (compress)

(c) BBB (constraint)

(d) Dropout-BBB (constraint)

Fig. 3.8 Calibration Bins for MNIST classification with BBB variants



(a) BBB (compress)

(b) Dropout-BBB (compress)

(c) BBB (constraint)

(d) Dropout-BBB (constraint)

Fig. 3.9 Calibration ROC for MNIST classification with BBB variants

3.4 FashionMNIST

	Observed Error (%)	Expected Error (%)	τ
Dropout (p=0.6)	8.4	8.1	
Gaussian dropout ($\sigma^2 = 2$)	8.6	8.0	0.5
Gaussian dropout (scale invariant)	8.8	5.1	
BBB (compression)	9.0	9.9	0.5
BBB (constraint)	9.1	9.2	1
Dropout-BBB (compression)	8.7	8.9	0.5
Dropout-BBB (constraint)	8.1	8.3	1
Gauss-Gamma	7.8	7.5	0.02
Bayesian Compression	**7.5**	7.7	0.4
Bayesian Dark Knowledge	9.8	10.0	
Bayesian Dark Knowledge (accept)	10.5	10.2	

Table 3.3 Observed and expected test accuracies on FashionMNIST

FashionMNIST is a drop-in replacement for MNIST containing grayscale images of clothing products in 10 categories. FashionMNIST is a significantly more difficult problem compared to MNIST and more closely resembles modern computer vision tasks [67].

Due to the increased complexity, the convolutional layers where increased to have 64 and 128 channels, respectively. The results on the test set can be found in Tbl. 3.3. Despite the similarity of the two tasks, the calibratedness of the various algorithms is significantly different for FashionMNIST. Even though all algorithms use the same architecture there is a significant spread in observed error ($7.5 - 10.5$).

Dropout VI with tuned dropout rate results in a good test accuracy. Unlike the MNIST task, the dropout rate could not be increased until predictions were no longer overconfident. For dropout rates $p > 0.6$ the training procedure quickly becomes unstable. Consequently, even for well-tuned dropout rates the model is still significantly overconfident (Fig. 3.10a & 3.12a).

Similar behaviour is observed for Gaussian dropout which again results in overconfident results. We found that it is useful to introduce additional regularisation by adding some dependency on scale into the prior. This was done by multiplying the scale invariant prior with an L2 penalty $p(\theta) \propto \exp(\theta^2/2\sigma^2)/\theta$ with $\sigma^2 = 2$. Despite the rather large gap between the observed (8.6%) and expected error (8.1%), the calibration bins (Fig. 3.10b) and calibration ROC curve (Fig. 3.12b) indicate a well-calibrated model. Thus, overall the model is not well-calibrated but the bias does not concentrate at predictions of a certain confidence or affect the tradeoff between the True and False Positive Rate. Using the scale invariant prior the model is severely miscalibrated (Fig. 3.12).

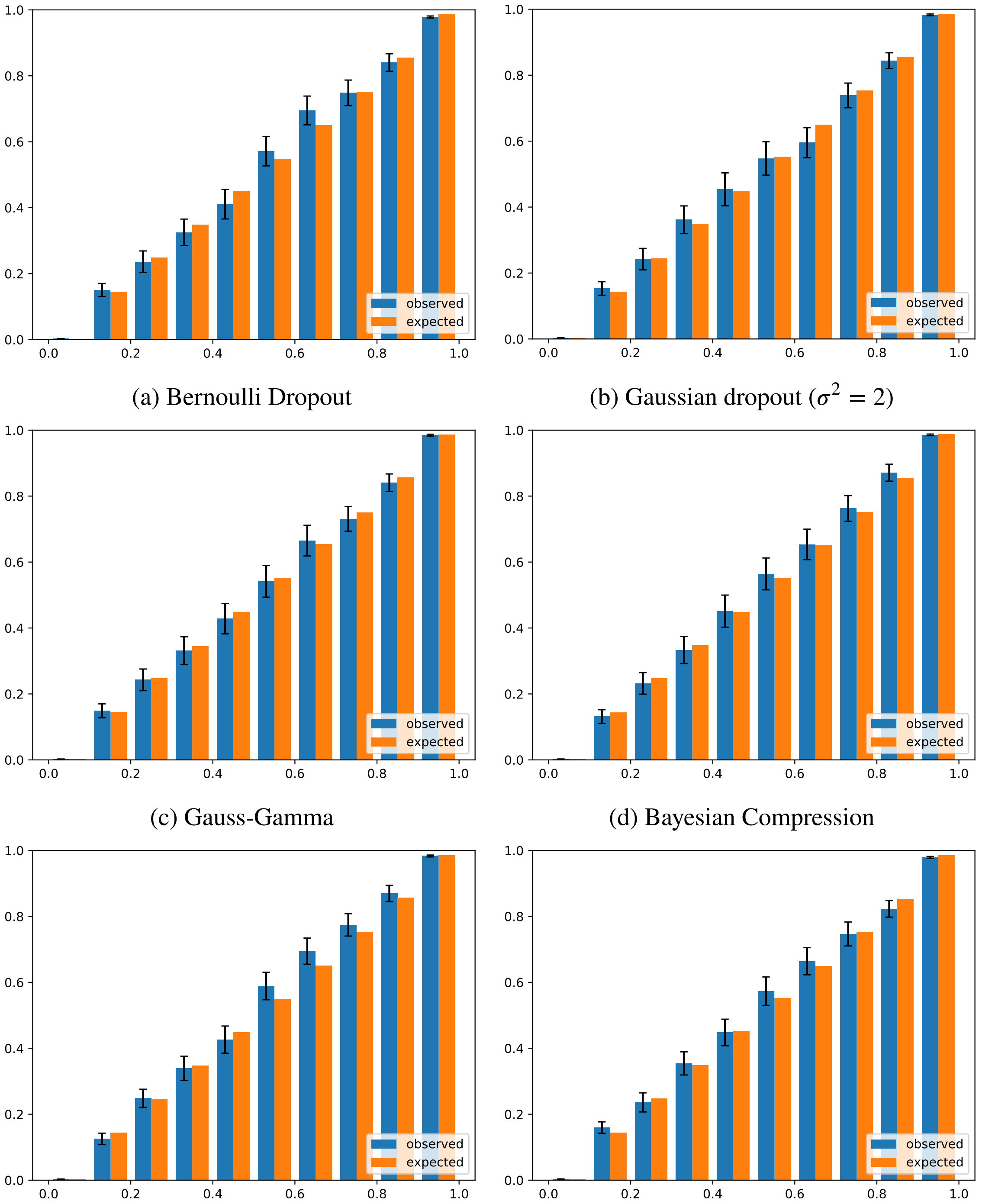
Both variants of BBB are unable to outperform dropout VI in terms of classification accuracy. Constraint BBB is well-calibrated (Fig. 3.13c & Fig. 3.14c) when the initialisation and training is tuned for calibration. The compression variant of BBB suffers from underconfidence (Fig. 3.13 & Fig. 3.14a).

Constraint Dropout-BBB is the best performing BBB variant on FashionMNIST in terms of test error. The uncertainty estimates are of high quality (Fig. 3.13d & Fig. 3.14d) after explicit tuning of the initialisation. Compression Dropout-BBB outputs slightly underconfident predictions (Fig. 3.13b & Fig. 3.14b).

Gauss-Gamma VI attains competitive test accuracy compared to the best performing method and is well-calibrated (Fig. 3.10c & 3.11c) with some overconfidence when the KL is significantly discounted $\tau = 0.02$. The large discounting that is needed can be attributed to over-parameterisation and the use of Adam as an optimiser which again lead to a quick increase in the amount of injected noise during training.

Bayesian Compression is the best performing inference method on FashionMNIST with a test error of 7.5%. Using a KL discount factor $\tau = 0.4$, the is slightly underconfident (Fig. 3.10d & 3.11d). Thus, Bayesian Compression yields a well-calibrated model with the lowest test accuracy and least amount of KL discounting.

We were unable to reach a test accuracy with Bayesian Dark Knowledge that is competitive with VI methods. A sampler without acceptance function results in slightly underconfident predictions (Fig. 3.10e & 3.11e). With acceptance function, the observed accuracy reduces further and uncertainty estimates become overconfident (Fig. 3.10f & 3.11f).



(a) Bernoulli Dropout

(b) Gaussian dropout ($\sigma^2 = 2$)

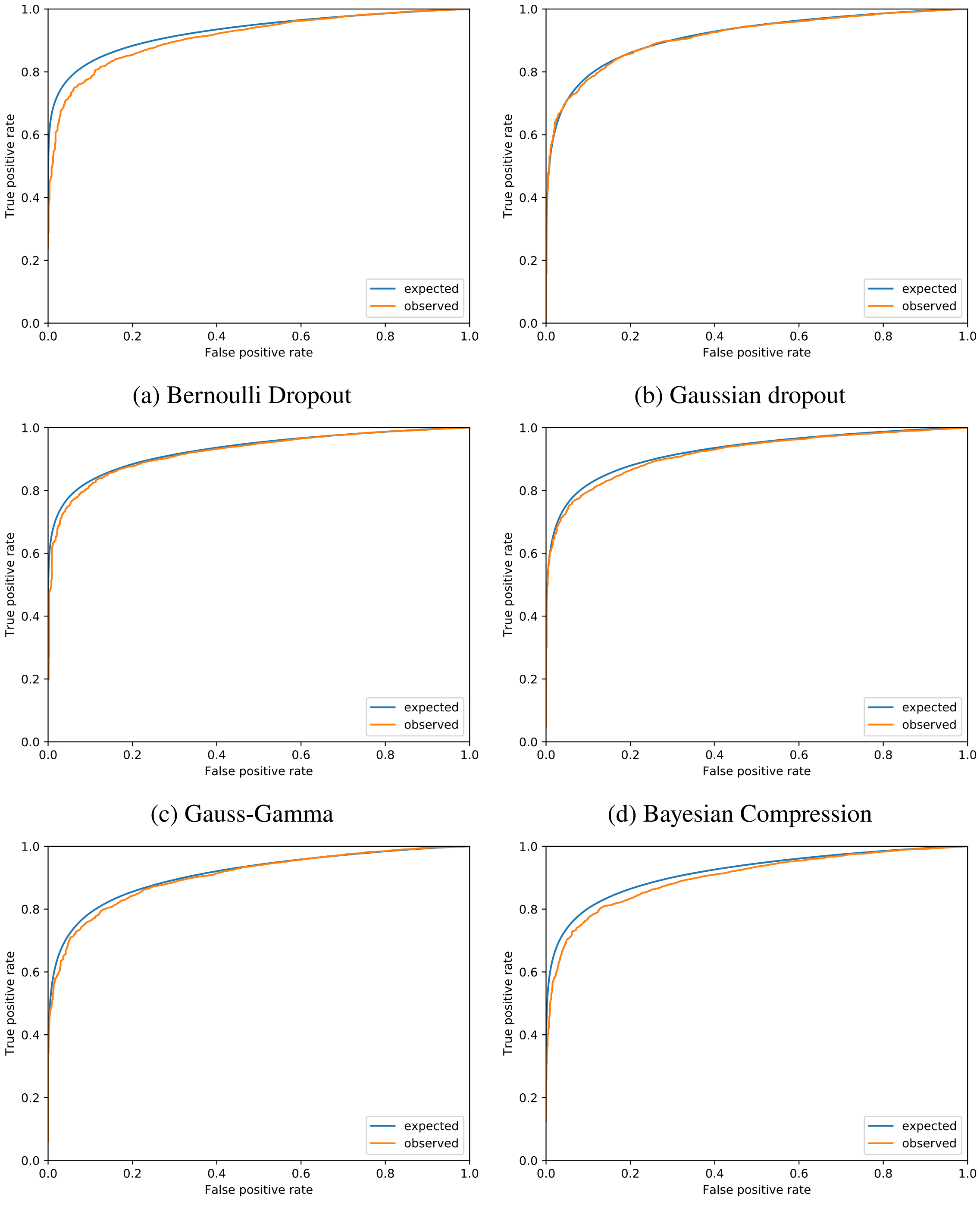
(c) Gauss-Gamma

(d) Bayesian Compression

(e) Bayesian Dark Knowledge

(f) Bayesian Dark Knowledge (accept function)

Fig. 3.10 Calibration bins for FashionMNIST classification



(a) Bernoulli Dropout

(b) Gaussian dropout

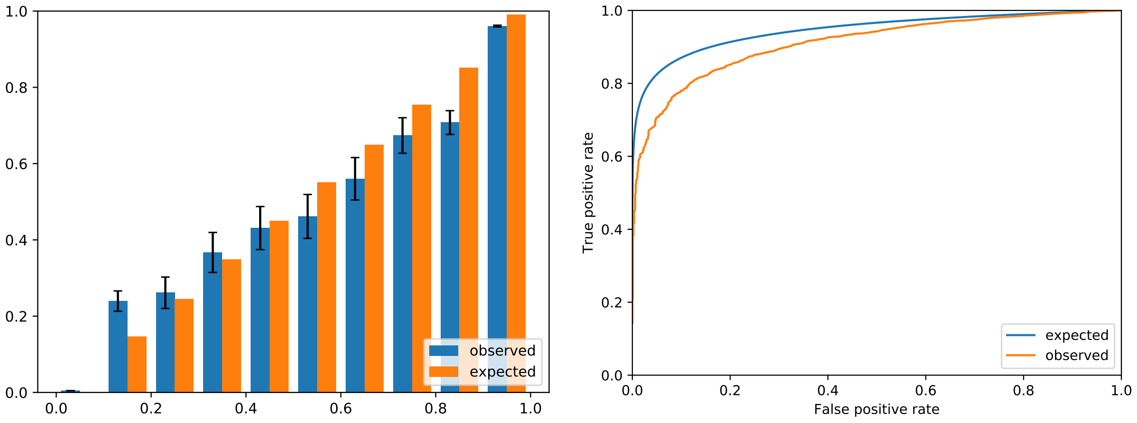
(c) Gauss-Gamma

(d) Bayesian Compression

(e) Bayesian Dark Knowledge

(f) Bayesian Dark Knowledge (accept function)

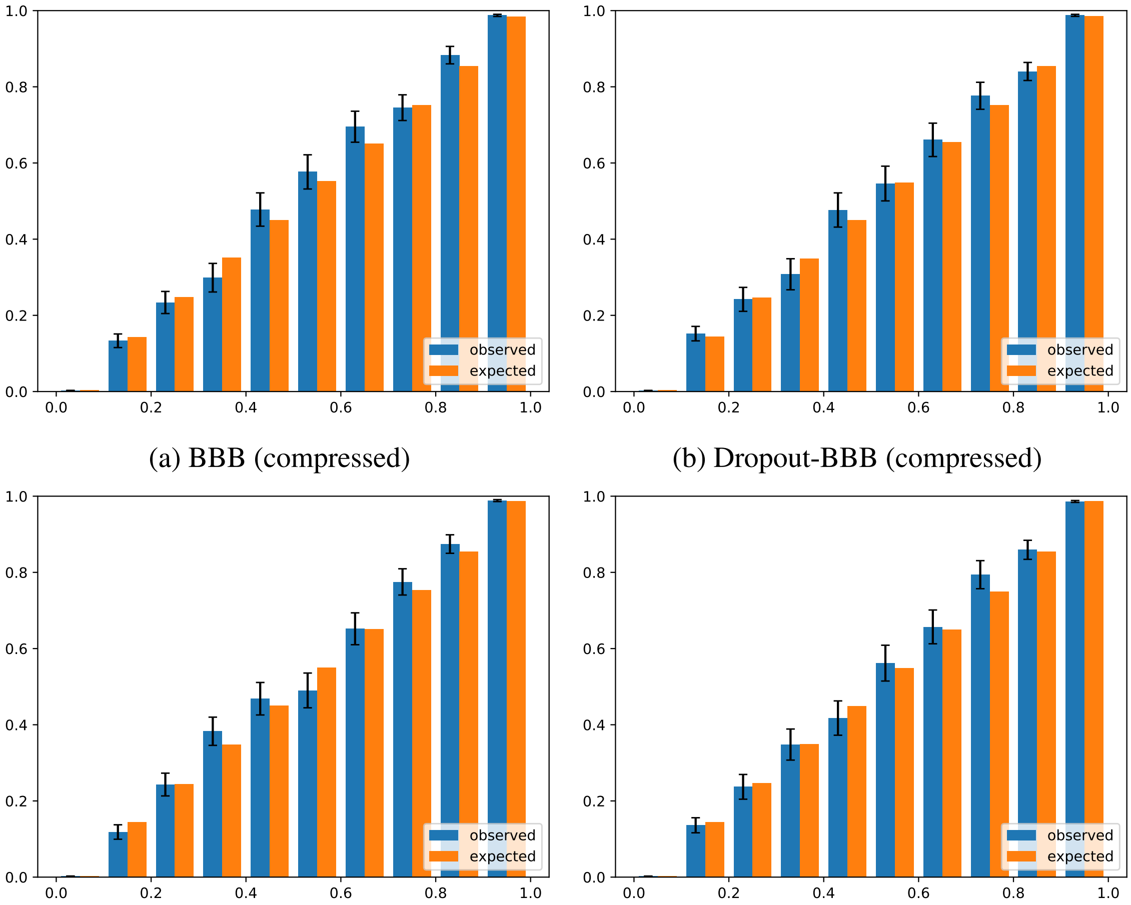
Fig. 3.11 Calibration ROC plots for FashionMNIST classification



(a) Calibration bins

(b) Calibration ROC

Fig. 3.12 Calibration of Gaussian Dropout with scale invariant prior



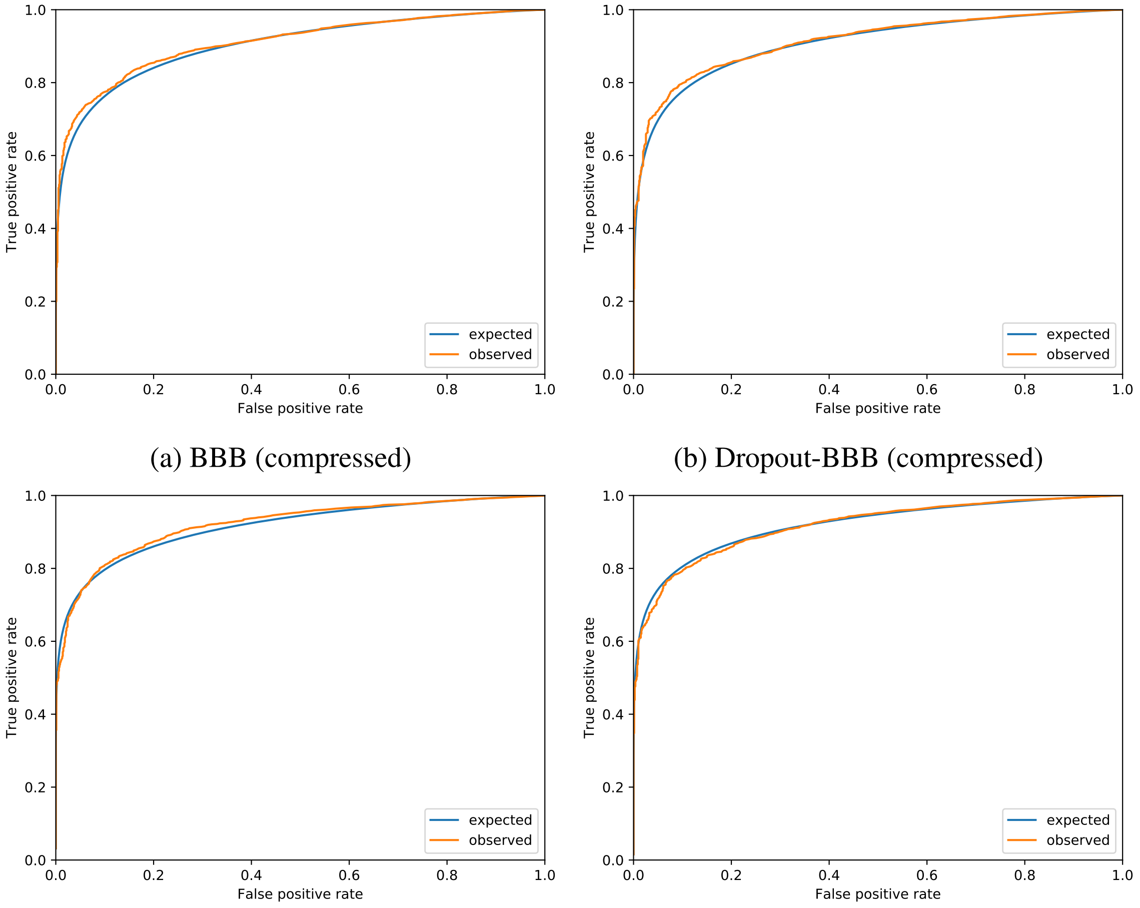
(a) BBB (compressed)

(b) Dropout-BBB (compressed)

(c) BBB (constraint)

(d) Dropout-BBB (constraint)

Fig. 3.13 Calibration Bins for FashionMNIST classification with BBB variants



(a) BBB (compressed)

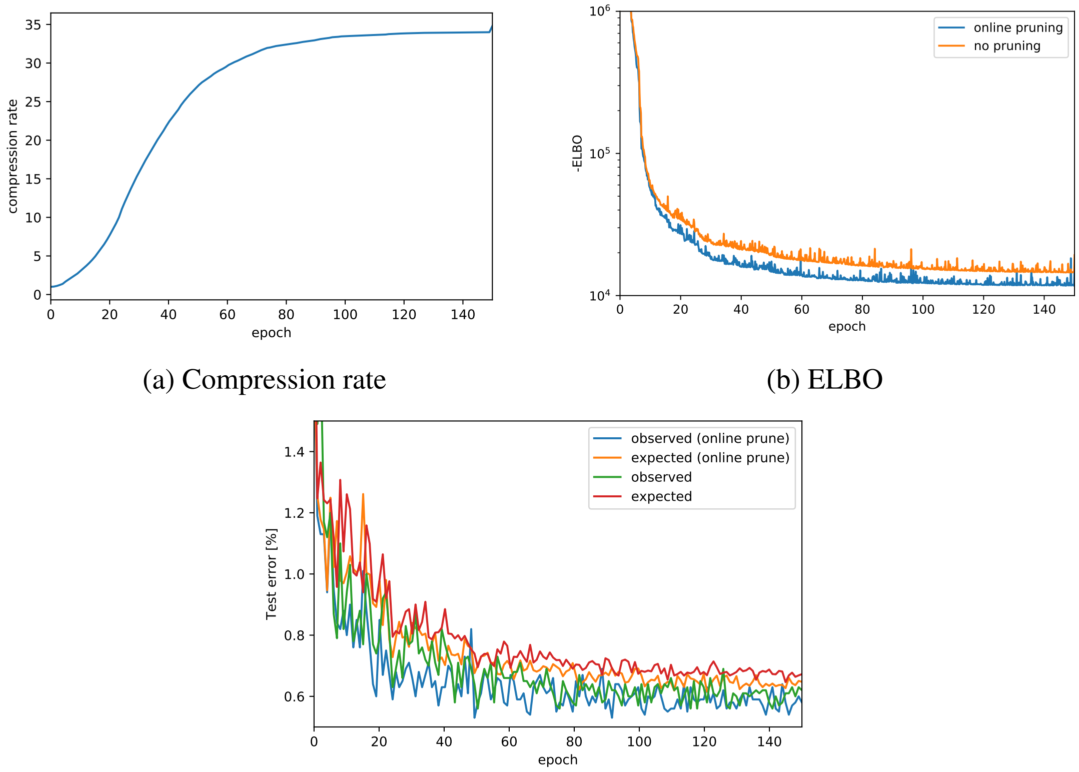
(b) Dropout-BBB (compressed)

(c) BBB (constraint)

(d) Dropout-BBB (constraint)

Fig. 3.14 Calibration ROC for FashionMNIST classification with BBB variants

3.5 Online pruning



(a) Compression rate

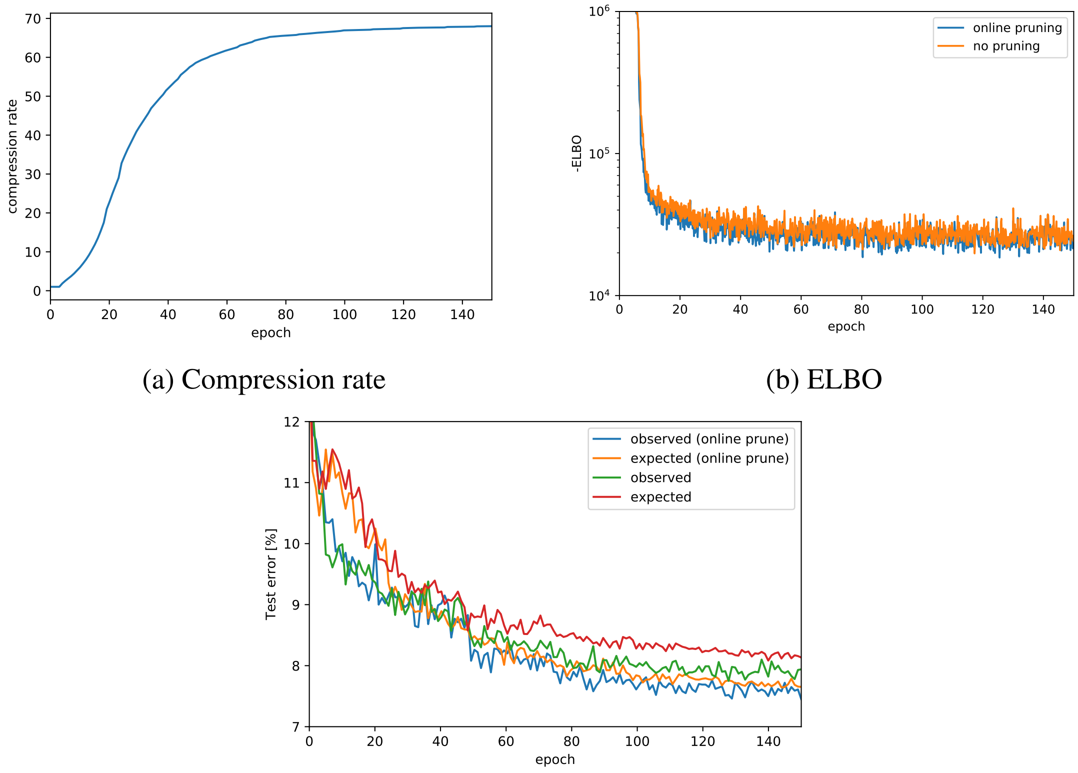
(b) ELBO

(c) Expected and Observed test error

Fig. 3.15 ELBO, test error, and compression rate during training of online Bayesian Compression for MNIST classification

Online Bayesian compression was performed by pruning the weights and features with a dropout thresholds of $0,99$ and 0.95, respectively. Additionally, weights were only pruned if $\sigma \geq 0.5\sigma_p$ where σ_p is the prior standard deviation. The compression rate of a network is defined as the ratio of the original number of weights over the non-zero weights of the pruned network. In the last 50 epochs the threshold for feature compression is reduced linearly from 0.95 to 0.05.

Fig. 3.15a shows that the compression rate starts to increase within the first few training epochs even though the ELBO and test accuracy are for from reaching a local optimum. Fig. 3.15b shows that the ELBO only increases marginally due to pruning and that the pruned weights and features were thus simply fitting the prior distribution. Both the expected and observed accuracy are slightly higher for online pruning compared to Bayesian Compression without pruning (Fig. 3.15c). The result is surprising when we consider that only 3% of the original weights are kept in the final network. The pruned architecture has 26 filters in both convolutional layers and 50 hidden units in the fully connected layers. The fully connected features seem to be pruned more aggressively which can be explained by their increased connectivity which makes it more expensive to maintain such features.



(a) Compression rate

(b) ELBO

(c) Expected and Observed test error

Fig. 3.16 ELBO, test error, and compression rate during training of online Bayesian Compression for FashionMNIST classification

For FashionMNIST similar patterns are observed. Only 1.5% of the original weights are kept and the pruned architecture consists of 41 and 27 filters in the convolutional layers and 40 hidden units in the fully connected layers. Interestingly, the second convolutional layer now has fewer features compared to the first layer. Such a choice would be uncommon for human engineered networks because typically the number of features is increased to counterbalance the reduction of information caused by pooling.

Fig. 3.16c shows that online pruning results in significantly better test accuracy and similar calibration. This suggest that the positive effect of online pruning becomes more pronounced as the network scales in size.

Chapter 4

Discussion

In the first place, this report aims to motivate the use of calibration as a direct metric for the quality of uncertainty estimates. Therefore, we will reflect on the advantages of these diagnostics and propose some potential improvements. Secondly, the experiments described in this report show the calibration characteristics of various Bayesian Inference methods. This chapter reflects on the performance, bottlenecks, and potential for future improvement of these methods.

4.1 Calibration tests

Calibration test can be performed using any random variable for which the model can predict the outcome, and on arbitrary subsets of the data (Sec. 1.3). It is therefore difficult to make general statements about the calibratedness of a model on arbitrary calibration tasks. Instead, the appropriate tests can be constructed based on the properties good uncertainty estimates should have depending on the context. For example, the ROC calibration test was designed to facilitate an application where a tradeoff must be made between true positives and false negatives.

The conducted experiments did not include post-calibration methods. The weakness of these methods is that they assume a limited definition of calibratedness. Usually these methods transform the binned probabilities such that perfect calibration is obtained on the binned calibration test [20, 50].

A potential focus for future research would be to define more general calibration tests. One way to achieve this is to learn a calibration test. This way both the selection procedure (1.4) and the random variable of interest would be parameterised. The parameterised calibration test should then maximise some definition of poor-calibration on a finite test sample.

The problem of generalising calibration tests is similar to the Fairness gerrymandering problem. In this case a selection procedure is used to find a subpopulation which suffers most from some pre-defined negative bias [30]. In fact, miscalibraiton itself is commonly used as the measure of unfairness [37].

A topic for future research could be to determine whether models that are well-calibrated perform better on tasks where good uncertainty estimates are essential. For example, Reinforcement Learning algorithms using Thompson Sampling, and detecting out-of-distribution or adversarial examples.

4.2 Variational Inference

The Variational Inference methods considered in this research share the ELBO as their objective. However, the characteristics of the obtained posterior approximation differ significantly.

Dropout VI has a constant entropy term resulting in efficient and robust training. However, the dropout rates must be tuned carefully as the approximation will not automatically increase concentration around a sharper mode (Sec. 2.2.2). It is further observed that for complex tasks, the training procedure can stop to converge when the dropout rate is increased too much, despite the fact that the model is still overconfident.

Gaussian Dropout does have a flexible variational distribution with trainable entropy. However, the variational distribution does not share the same support as the posterior. The ELBO diverges for both Bernoulli and Gaussian dropout VI and it is therefore non-trivial to what extend these methods are truly approximate Bayesian Inference methods. See [29] for a detailed analysis on the shortcomings and behaviour of Dropout VI methods.

BBB and the variations discussed in this report offer an alternative to Dropout VI with a well-defined ELBO and full support. Using the ELBO objective (2.24) naively results in underconfident models. Good calibration is only achieved when the we carefully tune the optimisation procedure, initialisation, and introduce some discount factor for the KL term.

Sparsity inducing priors are found to have a positive impact on calibration, generalisation, and training stability. In fact, pruning during training is found to have a positive impact the calibration and test accuracy by reducing the amount of weights that do not help fitting the data. Using Bayesian Compression we can prune entire features which also reduces the computation requirements of the model which consequently speeds up training.

VI methods designed to exploit the redundancy of modern neural networks improve both accuracy and calibratedness. However, sparse variational distributions like Gauss-Gamma are empirically found to require significant regularisation discounting to work well. A much simpler approach is to use pruning for weights with a small signal-to-noise ratio or dropout rate which is also found to require much less KL discounting in order to work well. In fact, when some underconfidence is permissible, Bayesian Compression with online pruning can be used with little extra tuning.

4.3 MCMC methods

We derived stochastic gradient HMC methods with an optional acceptance function with an adjustable tradeoff between efficiency/bias (Sec. 2.7.2). These methods allow for (nearly) unbiased but correlated sample to be obtained by annealing the step size to zero or by using an acceptance step with little bias. However, accepting all proposals in a stochastic HMC sampler with fixed step size resulted in better performance approximations of the posterior predictive. Thus, for Bayesian Neural Networks the autocorrelation of the sample is the bottleneck for obtaining a finite ensemble that is representative of the posterior predictive.

Although it proved useful to empirically show that variance is the bottleneck for well-calibrated MCMC approaches, the considered tasks might not demonstrate the full potential of the proposed stochastic acceptance function. Future research could focus on problems with smaller datasets and less complex models. The considered tasks also did not allow the proposed acceptance test to be compared to alternatives because existing acceptance tests require too large batch sizes to run experiments for MNIST or FashionMNIST in a reasonable amount of time.

Bayesian Dark Knowledge significantly improves the test accuracy and speed of inference for sampling methods. The student consistently outperformed an ensemble of networks, based on the weights at the end of each training epoch, in both test accuracy and calibration. Even though the finite ensemble of posterior samples requires 100 times as much compute and memory during inference.

A disadvantage of training student network is its tendency to produce underconfident predictions. This might be explained by the inclusive KL loss (2.71) used to train the student. In the student-teacher literature, various alternative losses have been proposed. For example, in [24] it is proposed to scale the logits of both teacher and student with a temperature parameter. However, in our experiments adjusting temperature was found to have a unstable effect on the calibration properties. More recently, discriminator networks have been used as a loss for the student in knowledge distillation [6]. A discriminator could potentially learn to distinguish the teacher and student based on their different levels of confidence. Consequently, end-to-end optimisation of the student could promote matching the level of confidence for the teacher and the student.

Another area for improvement is the input distribution $p(x)$. We followed the same approach as [39] by adding Gaussian noise to the original training set. Alternatively, a more realistic distribution $p(x)$ could be used based on a pre-trained VAE or GAN. Some experiments with a GAN based input distribution distribution were conducted. However, this let to a further increase in underconfidence of the student. This suggest a more complex input distribution is only worthwhile when the loss function is first improved to penalise underconfidence. More elaborate experiments will be needed to determine the interaction of Bayesian Dark Knowledge with the input distribution.

4.4 Conclusion

The presented work shows how the quality of uncertainty estimates cannot be taken for granted even when a model exhibits the ability to generalise well to unseen data. Calibration tests form a well grounded framework for assessing the quality of uncertainty estimates. A major difficulty lies in the the choice of an appropriate calibration test. Nonetheless, we find that most calibration tests sketch a similar image: Most approximate Bayesian inference methods cause either significant underconfidence or overconfidence in the model's predictions. In fact it is often sufficient to consider the difference in expected and observed accuracy to diagnose a poorly calibrated model. If the SOTA in calibration improves however, more specific or even adversely trained calibration tests might be useful to acquire even stronger guarantees about the quality of predictions.

Variational Inference provides a scalable solution for obtaining well-calibrated models. Bernoulli Dropout VI is found to be surprisingly robust when the dropout rate is optimised for calibration. However, Bernoulli and Gaussian Dropout have technical issues which makes it difficult to interpret them as proper Bayesian Inference methods. BBB provides a sound alternative where the amount of noise is part of the variational parameterisation and should thus not have to be chosen by the ML researcher. Unfortunately, we find that BBB suffers from significant underconfidence when trained naively.

Various techniques to reduce underconfidence where considered: compression priors, more expressive variational distributions like Dropout-BBB and Gauss-Gamma, and online pruning. All of these methods improve test accuracy. We do however find that hyperparameters must be tuned explicitly for calibration in order to get optimal results on calibration tests for all of these methods. The most robust approach is online pruning which only requires a small amount of KL discounting and has the additional benefits of faster training and providing a model that is both compressed and well-calibrated. On larger tasks like

FashionMNIST, Bayesian Compression with online pruning also results in the highest test accuracy.

On small problems sampling methods can be used to approximate the posterior predictive well. Bayesian Dark Knowledge offers a powerful tool to compress the posterior predictive into a single network allowing fast and well-calibrated inference on small tasks. The bias of stochastic gradient HMC samplers is negligible compared to the variance or autocorrelation in the produced sample for the parameters of a Bayesian Neural Network. However, sampling methods and Bayesian Dark Knowledge in particular are unable to compete with Variational Inference methods on more complex tasks.

In conclusion, this report has motivated the importance of empirically testing the quality of uncertainty estimates. Calibration tests provide a powerful way to identify bottlenecks in Bayesian Neural Networks and approximate inference methods. Sparsity inducing VI methods and compression in conjunction with carefully chosen priors can help to perform Bayesian Inference at scale with well-calibrated models as a result.

References

[1] Ahn, S., Korattikara, A., and Welling, M. (2012). Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring. *ICML*.

[2] Amari, S.-i. and Fellow, L. (2009). Alpha?-Divergence Is Unique, Belonging to Both f-Divergence and Bregman Divergence Classes. *IEEE Transactions on information theory*, 55(11):4925–4931.

[3] Bardenet, R., Doucet, A., and Holmes, C. (2014). Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. *ICML*, (4):405–413.

[4] Bardenet, R., Doucet, A., and Holmes, C. (2017). On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18:1–43.

[5] Barker, A. (1965). Monte Carlo Calculations of the Radial Distribution Functions for a Proton Electron Plasma. *Australian Journal of Physics*, 18(2):119.

[6] Belagiannis, V., Farshad, A., and Galasso, F. (2018). Adversarial Network Compression.

[7] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight Uncertainty in Neural Networks. 37.

[8] Chen, T., Fox, E. B., and Guestrin, C. (2014). Stochastic Gradient Hamiltonian Monte Carlo. 32.

[9] Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. pages 1–14.

[10] Dawid, A. P. (1982). The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610.

[11] Denil, M., Shakibi, B., Dinh, L., Aurelio, M., and Nando, R. (2014). Predicting Parameters in Deep Learning.

[12] Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. (2017). Sharp Minima Can Generalize For Deep Nets.

[13] Feller, W. (1971). An Introduction to Probability Theory and Its Applications - Vol. II.

[14] Gal, Y. (2016). Uncertainty in Deep Learning.

[15] Gal, Y. and Ghahramani, Z. (2015). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *ICML*, 48:1–10.

[16] Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. *The IEEE conference on computer vision and pattern recognition*, pages 2414–2423.

[17] Ge, R., Huang, F., Jin, C., and Yuan, Y. (2015). Escaping From Saddle Points – Online Stochastic Gradient for Tensor Decomposition. *Journal of Machine Learning Research*, 40:1–46.

[18] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and Harnassing ADVERSARIAL EXAMPLES. *ICLR*, pages 1–11.

[19] Graves, A. (2011). Practical Variational Inference for Neural Networks. *NIPS*.

[20] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks.

[21] Hastings, W. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109.

[22] Hernández-Lobato, J. M. and Adams, R. P. (2015). Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks.

[23] Hernandez-Lobato, J. M., Li, Y., Rowland, M., Hernández-lobato, D., Bui, T. D., and Turner, R. E. (2016). Black-Box α-Divergence Minimization. *ICML*.

[24] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the Knowledge in a Neural Network.

[25] Hinton, G. E. and van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *COLT*, pages 5–13.

[26] Hochreiter, S. and Schmidhuber, J. (1997). Flat Minima. *Neural Computation*, 9(1):1–42.

[27] Hron, J., Matthews, A. G. d. G., and Ghahramani, Z. (2017). Variational Gaussian Dropout is not Bayesian.

[28] Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

[29] Jiri Hron (2018). Variational Bayesian dropout: pitfalls and fixes.

[30] Kearns, M., Neel, S., Roth, A., and Wu, Z. S. (2017). Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness.

[31] Kendall, A. and Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *NIPS*.

[32] Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.-M., Lam, V.-D., Bewley, A., and Shah, A. (2018). Learning to Drive in a Day.

[33] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. pages 1–16.

[34] Kingma, D. P. and Ba, J. L. (2015). Adam: a Method for Stochastic Optimization. *ICLR*, pages 1–15.

[35] Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational Dropout and the Local Reparameterization Trick.

[36] Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes.

[37] Kleinberg, J., Mullainathan, S., and Raghavan, M. (2016). Inherent Trade-Offs in the Fair Determination of Risk Scores.

[38] Korattikara, A., Chen, Y., and Welling, M. (2013). Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget. pages 1–23.

[39] Korattikara, A., Rathod, V., Murphy, K., and Welling, M. (2015). Bayesian Dark Knowledge.

[40] Krueger, D., Ballas, N., Jastrzebski, S., Arpit, D., Kanwal, M. S., Maharaj, T., Bengio, E., Fischer, A., and Courville, A. (2017). Deep Nets Don't Learn via Memorization. *ICLR*, pages 1–4.

[41] Li, Y. and Gal, Y. (2017). Dropout Inference in Bayesian Neural Networks with Alpha-divergences.

[42] Li, Y., Turner, R. E., and Liu, Q. (2017). Approximate Inference with Amortised MCMC.

[43] Louizos, C., Ullrich, K., and Welling, M. (2017). Bayesian Compression for Deep Learning. *NIPS*.

[44] Louizos, C. and Welling, M. (2017). Multiplicative Normalizing Flows for Variational Bayesian Neural Networks.

[45] Mandt, S., Hoffman, M. D., and Blei, D. M. (2017). Stochastic Gradient Descent as Approximate Bayesian Inference. 18:1–35.

[46] McClure, P. and Kriegeskorte, N. (2016). Robustly representing uncertainty in deep neural networks through sampling. *NIPS*.

[47] Molchanov, D., Ashukha, A., and Vetrov, D. (2017). Variational Dropout Sparsifies Deep Neural Networks.

[48] Mousavi, H. S., Monga, V., and Tran, T. D. (2015). ICR: Iterative Convex Refinement for Sparse Signal Recovery Using Spike and Slab Priors.

[49] Murphy, A. H. and Winkler, R. L. (2010). Reliability of Subjective Probability Forecasts of Preciptation and Temperature. *Royal Statistical Society*, 26(1):41–47.

[50] Naeini, M. P., Cooper, G. F., and Hauskrecht, M. (2015). Obtaining Well Calibrated Probabilities Using Bayesian Binning. *AAAI*, pages 2901–2907.

[51] Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, pages 113–162.

[52] Osband, I. (2016). Risk versus Uncertainty in Deep Learning: Bayes, Bootstrap and the Dangers of Dropout. *NIPS 2016 Bayesian Deep Learning Workshop*.

[53] Paszke, A., Chanan, G., Lin, Z., Gross, S., Yang, E., Antiga, L., and Devito, Z. (2017). Automatic differentiation in PyTorch. *NIPS*, pages 1–4.

[54] Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., and Weinberger, K. Q. (2017). On Fairness and Calibration.

[55] Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2018). Do CIFAR-10 Classifiers Generalize to CIFAR-10?

[56] Rényi, A. (1961). On measures of entropy and information. In *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 547–561.

[57] Riquelme, C., Tucker, G., Snoek, J., and Brain, G. (2017). Deep Bayesian Bandits Showdown. *NIPS 2017 Bayesian Deep Learning Workshop*.

[58] Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407.

[59] Scott, W. A. (2002). Maximum likelihood estimation using the empirical fisher information matrix. *Journal of Statistical Computation and Simulation*, 72(8):599–611.

[60] Seita, D., Pan, X., Chen, H., and Canny, J. (2016). An Efficient Minibatch Acceptance Test for Metropolis-Hastings.

[61] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359.

[62] Snoek, J., Larochelle, H., and Adams, R. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *NIPS*, pages 1–9.

[63] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

[64] Trippe, B. and Turner, R. (2018). Overpruning in Variational Bayesian Neural Networks.

[65] Turner, R. E. and Sahani, M. (2011). Two problems with variational expectation maximisation for time series models. In *Bayesian Time Series Models*, pages 104–124.

[66] Welling, M. and Teh, Y. W. (2011). Bayesian Learning via Stochastic Gradient Langevin Dynamics. *ICML*, pages 681–688.

[67] Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.

[68] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization.