Chapter 2

The Language of Uncertainty

To formalise our discussion of model uncertainty we will rely on probabilistic modelling, and more specifically on Bayesian modelling. Bayesian probability theory offers us the machinery we need to develop our tools. Together with techniques for approximate inference in Bayesian models, in the next chapter we will present the main results of this work. But prior to that, let us review the main ideas underlying Bayesian modelling, approximate inference, and a model of key importance to us: the Bayesian neural network.

2.1 Bayesian modelling

Given training inputs $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and their corresponding outputs $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$, in Bayesian (parametric) regression we would like to find the parameters $\boldsymbol{\omega}$ of a function $\mathbf{y} = \mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x})$ that are *likely to have generated* our outputs. What parameters are likely to have generated our data? Following the Bayesian approach we would put some *prior* distribution over the space of parameters, $p(\boldsymbol{\omega})$. This distribution represents our prior belief as to which parameters are likely to have generated our data before we observe any data points. Once some data is observed, this distribution will be transformed to capture the more likely and less likely parameters given the observed data points. For this we further need to define a likelihood distribution $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega})$—the probabilistic model by which the inputs generate the outputs given some parameter setting $\boldsymbol{\omega}$.

For classification tasks we may assume a softmax likelihood,

$$p\big(y = d|\mathbf{x}, \boldsymbol{\omega}\big) = \frac{\exp(f_d^{\boldsymbol{\omega}}(\mathbf{x}))}{\sum_{d'} \exp(f_{d'}^{\boldsymbol{\omega}}(\mathbf{x}))}$$

or a Gaussian likelihood for regression:

$$p\Big(\mathbf{y}|\mathbf{x},\boldsymbol{\omega}\Big) = \mathcal{N}(\mathbf{y};\mathbf{f}^{\omega}(\mathbf{x}),\tau^{-1}I) \tag{2.1}$$

with model precision τ. This can be seen as corrupting the model output with observation noise with variance τ^{-1}.

Given a dataset $\mathbf{X},\mathbf{Y}$, we then look for the *posterior* distribution over the space of parameters by invoking Bayes' theorem:

$$p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega})p(\boldsymbol{\omega})}{p(\mathbf{Y}|\mathbf{X})}.$$

This distribution captures the most probable function parameters given our observed data. With it we can predict an output for a new input point $\mathbf{x}^*$ by integrating

$$p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{X},\mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{x}^*,\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\mathrm{d}\boldsymbol{\omega}.$$

This process is known as *inference*[1].

A key component in posterior evaluation is the normaliser, also called *model evidence*:

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega})p(\boldsymbol{\omega})\mathrm{d}\boldsymbol{\omega}. \tag{2.2}$$

Performing this integration is also referred to as *marginalising* the likelihood over $\boldsymbol{\omega}$, which is the origin of the alternative name for the model evidence: *marginal likelihood*. Marginalisation can be done analytically for simple models such as Bayesian linear regression. In such models the likelihood is *conjugate* to the prior, and the integral can be solved with known tools in calculus. Marginalisation is the bread and butter of Bayesian modelling, and ideally we would want to marginalise over all uncertain quantities—i.e. average w.r.t. all possible model parameter values $\boldsymbol{\omega}$, each weighted by its plausibility $p(\boldsymbol{\omega})$.

But with more interesting models (even basis function regression when the basis functions are not fixed) this marginalisation cannot be done analytically. In such cases an approximation is needed.

[1]Note that "inference" in Bayesian modelling has a different meaning to that in deep learning. In Bayesian modelling "inference" is the process of integration over model parameters. This means that "approximate inference" can involve optimisation at training time (approximating this integral). This is in contrast to deep learning literature where "inference" often means model evaluation at test time alone.

2.1.1 Variational inference

The true posterior $p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})$ cannot usually be evaluated analytically. Instead we define an approximating *variational* distribution $q_\theta(\boldsymbol{\omega})$, parametrised by θ, whose structure is easy to evaluate. We would like our approximating distribution to be as close as possible to the posterior distribution obtained from the original model. We thus minimise the Kullback–Leibler (KL) divergence [Kullback, 1959; Kullback and Leibler, 1951] w.r.t. θ, intuitively a measure of similarity between two distributions:

$$\mathrm{KL}(q_\theta(\boldsymbol{\omega}) \mid\mid p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})) = \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})} \mathrm{d}\boldsymbol{\omega}. \tag{2.3}$$

Note that this integral is only defined when $q_\theta(\boldsymbol{\omega})$ is absolutely continuous w.r.t. $p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})$ (i.e. for every measurable set A, $p(A|\mathbf{X}, \mathbf{Y}) = 0$ implies $q_\theta(A) = 0$). We denote by $q^*_\theta(\boldsymbol{\omega})$ the minimum of this optimisation objective (often a local minimum).

Minimising the KL divergence allows us to approximate the predictive distribution as

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) \approx \int p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\omega}) q^*_\theta(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} =: q^*_\theta(\mathbf{y}^*|\mathbf{x}^*). \tag{2.4}$$

KL divergence minimisation is also equivalent to maximising the *evidence lower bound* (ELBO) w.r.t. the variational parameters defining $q_\theta(\boldsymbol{\omega})$,

$$\mathcal{L}_{\mathrm{VI}}(\theta) := \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} - \mathrm{KL}(q_\theta(\boldsymbol{\omega}) || p(\boldsymbol{\omega})) \leq \log p(\mathbf{Y}|\mathbf{X}) = \log \text{ evidence}, \tag{2.5}$$

which defines the objective we will refer to henceforth. Maximising the first term in this last equation (referred to as the *expected log likelihood*) encourages $q_\theta(\boldsymbol{\omega})$ to explain the data well, while minimising the second term (referred to as the *prior KL*) encourages $q_\theta(\boldsymbol{\omega})$ to be as close as possible to the prior. This acts as an "Occam razor" term and penalises complex distributions $q_\theta(\boldsymbol{\omega})$.

This procedure is known as *variational inference* (VI), a standard technique in Bayesian modelling [Jordan et al., 1999]. Variational inference replaces the Bayesian modelling marginalisation with optimisation, i.e. we replace the calculation of integrals with that of derivatives. But compared to the optimisation approaches often used in deep learning, in this setting we optimise over distributions instead of point estimates[2]. This approach preserves many of the advantages of Bayesian modelling (such as the

[2]Note that optimisation in the deep learning sense can be recovered by setting the approximating distribution as a delta $q_\theta(\boldsymbol{\omega}) := \delta(\boldsymbol{\omega} - \theta)$.

balance between complex models and models that explain the data well), and results in probabilistic models that capture model uncertainty.

The calculation of derivatives is often much easier than that of integrals, which makes many approximations tractable. But even though this procedure makes inference analytical for a large class of models, it still lacks in many ways. This technique does not scale to large data (evaluating $\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}$ requires calculations over the entire dataset), and the approach does not adapt to complex models (models in which this last integral cannot be evaluated analytically). Recent advances in VI allow us to circumvent these difficulties, and we will get back to this topic later in §3.1. But first we review a model of key importance to us: the Bayesian neural network.

2.2 Bayesian neural networks

First suggested in the '90s and studied extensively since then [MacKay, 1992b; Neal, 1995], Bayesian neural networks (BNNs, Bayesian NNs) offer a probabilistic interpretation of deep learning models by inferring distributions over the models' weights. The model offers robustness to over-fitting, uncertainty estimates, and can easily learn from small datasets.

Bayesian NNs place a prior distribution over a neural network's weights, which induces a distribution over a parametric set of functions. Given weight matrices $\mathbf{W}_i$ and bias vectors $\mathbf{b}_i$ for layer i, we often place standard matrix Gaussian prior distributions over the weight matrices, $p(\mathbf{W}_i) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and often assume a point estimate for the bias vectors for simplicity. Likelihood specification often follows the standard Bayesian literature (such as the softmax likelihood or Gaussian likelihood, §2.1).

Bayesian neural networks are easy to formulate, but difficult to perform inference in. Many have tried over the years, to varying degrees of success. This is surveyed next.

2.2.1 Brief history

In their work at AT&T Bell Laboratories in 1987, Denker, Schwartz, Wittner, Solla, Howard, Jackel, and Hopfield [1987] looked at the general problem of learning from examples. They extended on an already vast existing literature in neural networks research, and proposed a new way to train these. In essence, Denker et al. [1987, page 904] proposed placing a prior distribution over the space of weights (and used a uniform distribution on a compact space). Denoting all possible (binary) inputs as $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, they then mapped each weight configuration to a set of corresponding network outputs

$\{\hat{\mathbf{y}}_1, ..., \hat{\mathbf{y}}_N\}$. This allowed them to integrate over the weights and obtain a marginal probability for each set $\{\hat{\mathbf{y}}_1, ..., \hat{\mathbf{y}}_N\}$. Given observed training data, the probabilities of inconsistent network weights were set to zero, leading to an updated marginal probability for each set $\{\hat{\mathbf{y}}_1, ..., \hat{\mathbf{y}}_N\}$. These marginal probabilities were then used to calculate the efficiency of a network (by calculating its entropy).

Working from AT&T Bell Laboratories as well, Tishby, Levin, and Solla [1989] extended on the ideas of Denker et al. [1987] and developed a statistical framework to reason about network generalisation error. As far as I could find, this is the earliest citation for what one would consider today to be a "Bayesian neural network". Tishby et al. [1989] showed that the only statistical interpretation of a NN Euclidean loss is as maximum likelihood w.r.t. a Gaussian likelihood over the network outputs. They proceeded to define a prior distribution over the network weights, and showed that inference could theoretically be performed through the invocation of Bayes' theorem. This was used to propose a quantity dependent on the training set alone that would correlate to the network's generalisation error on a test set (although the problem of inference practicality was overlooked).

In more work coming out from AT&T Bell Laboratories, Denker and LeCun [1991] extended on Tishby et al. [1989]'s ideas and suggested the use of Laplace's method to approximate the posterior of the Bayesian NN. Denker and LeCun [1991] used the (back then) newly suggested "backpropagation" technique, and optimised the neural network weights to find a mode. They then fitted a Gaussian to the discovered mode, with the width of the Gaussian determined by the Hessian of the likelihood at that mode.

Working from California Institute of Technology, MacKay [1992b] performed an extensive study of Bayesian NNs. MacKay [1992b] advocated the use of model evidence for model comparison, and obtained this quantity following the approximation of Denker and LeCun [1991]. Using an array of experiments with different model sizes and model configurations, MacKay [1992b] showed that model evidence correlates to generalisation error, and thus can be used to select model size. MacKay [1992b] further showed that model misspecification can lead to Bayes failure, where model evidence does not indicate model generalisation. As he showed, this could happen for example when the priors of multiple weight layers are tied together, and thus low probability could be given to models with large input weight magnitude together with small output weight magnitude unjustifiably.

As a way of model regularisation, Hinton and Van Camp [1993] (working from Toronto) suggested the use of minimum description length (MDL) to penalise high amounts of information contained in a network's weights. They showed that for a single hidden

layer NN it is possible to compute their suggested (and somewhat ad-hoc) objective analytically. This technique can be seen as the first variational inference approximation to Bayesian NNs—even though the suggested technique is motivated through information theoretic foundations, the resulting optimisation objective is identical to VI's ELBO (as will be explained below).

In his thesis, Neal [1995] developed alternative inference approximations for Bayesian NNs based on Monte Carlo (MC) techniques. Hamiltonian Monte Carlo (HMC, also *Hybrid Monte Carlo*) was suggested for posterior inference, a technique based on dynamical simulation that does not rely on any prior assumptions about the form of the posterior distribution. Neal [1995] attempted to reproduce MacKay [1992b]'s experiments which relied on Laplace's method [Denker and LeCun, 1991]. Neal [1995, page 122] validated some of MacKay [1992b]'s experiments, but could not reproduce some others, presumably because of the approximation error of Laplace's method. In addition to this work, Neal [1995] further studied different prior distributions in Bayesian NNs, and showed that in the limit of the number of units the model would converge to various stable processes, depending on the prior used (for example, the model would converge to a Gaussian process when a Gaussian prior is used).

One last key development relevant to our setting stems from work by Barber and Bishop [1998]. In the work, Barber and Bishop [1998] developed Hinton and Van Camp [1993]'s MDL approximation under a VI interpretation, and replaced Hinton and Van Camp [1993]'s diagonal covariance matrices with full covariance matrices. Barber and Bishop [1998] further highlighted the fact that the obtained objective forms a lower bound to the model evidence. Lastly, Barber and Bishop [1998] placed gamma priors over the network hyper-parameters. They then performed VI with free-form variational distributions over the hyper-parameters, and derived their optimal form. This allowed the model evidence (the quantity that the ELBO bounds) to remain constant w.r.t. the changing hyper-parameter values, since the hyper-parameters are always averaged w.r.t. their approximating distribution.

Remark (Multimodal or unimodal approximating distributions?)**.** MacKay [1992b] used a unimodal Gaussian distribution to fit the posterior and reasoned that this already gives us much more than a maximum likelihood estimate (MLE). That's because the width of the fitted Gaussian acts as an "Occam razor" and penalises complex models (in essence capturing the ratio between the posterior parameters' distribution volume to prior parameters' distribution volume). Neal [1995] criticised the simplistic unimodal approximation though, and argued that the approximation

only works in low dimensional problems (and therefore we should use HMC that doesn't place any assumptions over the posterior structure, and able to capture complicated multimodal posterior and predictive distributions).
But fitting the posterior over the weights of a Bayesian NN with a unimodal approximating distribution does not mean the predictive distribution would be unimodal! imagine for simplicity that the intermediate feature output from the first layer is a unimodal distribution (a uniform for example) and let's say, for the sake of argument, that the layers following that are modelled with delta distributions (or Gaussians with very small variances). Given enough follow-up layers we can capture any function to arbitrary precision—including the inverse cumulative distribution function (CDF) of any multimodal distribution. Passing our uniform output from the first layer through the rest of the layers—in effect transforming the uniform with this inverse CDF—would give a multimodal predictive distribution.

These approaches represent important first steps towards practical inference in Bayesian NNs. But these are difficult to adapt to modern needs found in the field. Next we review more modern approaches to approximate inference in Bayesian NNs.

2.2.2 Modern approximate inference

Modern research in Bayesian NNs often relies on either different flavours of variational inference, or sampling based techniques. Each approach has its merits, but has its limitations as well. Next we will survey some of the recently suggested approaches.

For variational inference, modern approaches follow the work of Hinton and Van Camp [1993] closely. This work relied on a fully factorised approximation—the approximating distribution assumes independence of each weight scalar in each layer from all other weights. We will go over this approach in more detail from the VI perspective. This will be followed by a survey of extensions to this approach.

Recall that we are interested in finding the distribution of weight matrices (parametrising our functions) that have generated our data. This is the posterior over the weights given our observables $\mathbf{X}, \mathbf{Y}$: $p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})$. This posterior is not tractable in general, and we use variational inference to approximate it. For this we need to define an approximating variational distribution $q_\theta(\boldsymbol{\omega})$, and then minimise the KL divergence between the

approximating distribution and the full posterior[3]:

$$\begin{aligned}\mathrm{KL}\Big(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\Big) &\propto -\int q_\theta(\boldsymbol{\omega})\log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega})\mathrm{d}\boldsymbol{\omega} + \mathrm{KL}(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega})) \\ &= -\sum_{i=1}^{N}\int q_\theta(\boldsymbol{\omega})\log p(\mathbf{y}_i|\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i))\mathrm{d}\boldsymbol{\omega} + \mathrm{KL}(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega}))\end{aligned} \tag{2.6}$$

with $\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i)$ the model output on input $\mathbf{x}_i$ and with the terms summed-over in the last equation referred to as expected log likelihoods. Hinton and Van Camp [1993] defined $q_\theta(\boldsymbol{\omega})$ to factorise over the weights:

$$q_\theta(\boldsymbol{\omega}) = \prod_{i=1}^{L} q_\theta(\mathbf{W}_i) = \prod_{i=1}^{L}\prod_{j=1}^{K_i}\prod_{k=1}^{K_{i+1}} q_{m_{ijk},\sigma_{ijk}}(w_{ijk}) = \prod_{i,j,k}\mathcal{N}(w_{ijk}; m_{ijk}, \sigma_{ijk}^2).$$

Optimising the objective is challenging, since the expected log likelihood is intractable for most BNN model structures. Therefore Hinton and Van Camp [1993] only demonstrated the technique with a single hidden layer in which case the optimisation objective is analytical.

Even when the optimisation objective is analytical, this approximation can work quite badly in practice. This could possibly be attributed to the method losing important information about weight correlations. Barber and Bishop [1998], by modelling correlations between the weights, managed to improve on [Hinton and Van Camp, 1993]. But this came with the price of increased computational complexity. The method now required the representation of covariance matrices with a number of entries quadratic in the number of weights in the model. This is impractical for most modern models, since the number of model parameters in modern deep learning tends to be as large as modern hardware would allow. With the additional constraint of having to handle large amounts of data, plain VI could not scale to modern needs and was thus mostly forgotten.

In recent work, Graves [2011] has attempted to answer the difficulties above. Graves [2011] used *data sub-sampling* techniques in a fully factorised VI objective (although this was developed in the context of MDL). This allowed the technique to scale to large amounts of data. Graves [2011] approximated the intractable expected log likelihood with Monte Carlo estimates [Opper and Archambeau, 2009], which allowed the technique to scale to more complex models, going beyond the single layer restriction[4]. Ironically, even though Bayesian NNs were not mentioned in the paper even once, Graves [2011]'s

[3]We slightly abuse standard notation here, and use $A \propto B$ to mean that A is identical to B up to some constant c: $A = B + c$ rather than $A = cB$.

[4]These are two techniques that started gaining in popularity within VI at the time, and will be reviewed in more detail in the next chapter.

work can be seen as a big step forward from previous research on approximate inference in BNNs. For the first time a *practical* technique was proposed, which scaled to complex models and big data. Sadly, similarly to Hinton and Van Camp [1993], the technique still performed badly in practice [Hernandez-Lobato and Adams, 2015], perhaps because of the lack of correlations over the weights. As a result it was not picked-up and actively extended-on by the community for a long period of time.

In more recent work, done in parallel to the one presented in this thesis, Blundell et al. [2015] has built on Graves [2011]'s approach, re-parametrising the expected log likelihood MC estimates following Kingma and Welling [2013]. Blundell et al. [2015] further changed the BNN model itself by putting a mixture of Gaussians prior over each weight and optimised the mixture components. This allowed them to improve model performance compared to Graves [2011], and match that of existing approaches in the field. But even this variational approach can still be computationally expensive—the use of Gaussian approximating distributions increases the number of model parameters considerably, without increasing model capacity by much. Blundell et al. [2015] for example used Gaussian distributions for Bayesian NN posterior approximation, doubling the number of model parameters. This makes the approach difficult for use with large complex models since the increase in the number of parameters can be too costly, especially on systems with limited memory.

An alternative approach to variational inference makes use of expectation propagation [Hernandez-Lobato and Adams, 2015]. Hernandez-Lobato and Adams [2015]'s probabilistic back propagation (PBP) has improved considerably on VI approaches such as [Graves, 2011] both in root mean square error (RMSE) and in uncertainty estimation. Closely related to PBP are the recently suggested approximate inference techniques based on α-divergence minimisation [Hernández-Lobato et al., 2016; Li and Turner, 2016; Minka, 2005]. Most of these are still under active development, and not used in practice yet (with the exception of [Hernández-Lobato et al., 2016], which was used in reinforcement learning [Depeweg et al., 2016]). These approximate inference techniques offer alternative minimisation objectives to VI's ELBO, relying on various forms of Rényi's α-divergence [Rényi et al., 1961]. The approximating distributions used in these papers are still rather simple, with [Depeweg et al., 2016; Hernández-Lobato et al., 2016] for example making use of fully factorised Gaussian distributions. The main motivation behind the techniques is to avoid VI's mode-seeking behaviour, and indeed in fitting a more dispersed distribution (a distribution spreading its mass on larger parts of the support), these techniques seem to sacrifice their estimation of the dominant modes of the posterior. But in performing

predictions we often care about finding modes and exploring around them, rather than interpolating between different modes.

An alternative line of research has extended on Neal [1995]'s HMC rather than the work of Hinton and Van Camp [1993]. HMC makes use of Hamiltonian dynamics in MCMC [Duane et al., 1987], following Newton's laws of motion [Newton, 1687]. Neal [1995]'s use of HMC in statistics was to generate samples from a model's posterior which would be difficult to sample from directly. But HMC can be difficult to work with in practice, as setting the leapfrog step sizes can be somewhat of an art. Further, the method does not scale to large data. This is because the method requires us to calculate gradients for the likelihood evaluated on the entire dataset [Neal, 2011]. The Langevin method is a simplification of Hamiltonian dynamics where only a single leapfrog step is used. The use of a single step simplifies the inference method itself considerably, and allows it to be scaled to large data. The latter is achieved through the use of stochastic estimates of the gradient of the likelihood, replacing the gradients over the entire dataset [Welling and Teh, 2011]. These stochastic estimates of the likelihood are similar to the ones used to scale-up VI used by Graves [2011]. Welling and Teh [2011]'s scalable Langevin method was named Stochastic Gradient Langevin Dynamics (SGLD). The SGLD technique generates a set of samples $\{\widehat{\boldsymbol{\omega}}_t\}$ from the model's posterior over the random variable $\boldsymbol{\omega}$ by adding stochastic gradient steps to the previously generated samples:

$$\begin{aligned}\Delta\boldsymbol{\omega} &= \frac{\epsilon}{2}\left(\nabla \log p(\boldsymbol{\omega}) + \frac{N}{M}\sum_{i\in S}\nabla \log p(\mathbf{y}_i|\mathbf{x}_i,\boldsymbol{\omega})\right) + \eta \\ \eta &\sim \mathcal{N}(0,\epsilon).\end{aligned}$$

Here S is a randomly sampled set of M indices from $\{1,...,N\}$ and ϵ is decreased in magnitude following the Robbins–Monro equations [Robbins and Monro, 1951]. Unlike fully-factorised VI, this approach can capture weight correlations since the approximate posterior structure need not be specified. However, the main difficulty with SGLD is that it often collapses to a single mode and explores that mode alone. This is because ϵ decreases so rapidly that the probability of the method to jump outside of the region of a mode and converge to another mode is extremely small. Even though that in the limit of time the entire support of the distribution will be explored, in practice the method explores a single mode, similar to VI. Further, the method generates correlated samples—which means that many samples need to be generated, since many intermediate samples have to be discarded. These factors somewhat limit SGLD's practicality.

One last approach I shall review here cannot technically be considered as approximate inference in BNNs, but nonetheless can be used to estimate model uncertainty. The technique uses an ensemble of deterministic models, meaning that each model in the ensemble produces a point estimate rather than a distribution. It works by independently training many randomly initialised instances of a model on the same dataset (or different random subsets in the case of bootstrapping), and given an input test point, evaluating the sample variance of the outputs from all deterministic models [Osband et al., 2016]. Even though this approach is more computationally efficient than many Bayesian approaches to model uncertainty (apart from the need to represent the parameters of multiple models), its produced uncertainty estimates lack in many ways as explained in the next illustrative example. To see this, let's see what would happen if each deterministic model were to be given by an RBF network (whose predictions coincide with the predictive mean of a Gaussian process with a squared-exponential (SE) covariance function). An RBF network predicts zero for test points far from the training data. This means that in an ensemble of RBF networks, each and every network will predict zero for a given test point far from the training data. As a result, the sample variance of this technique will be zero at the given test point. The ensemble of models will have very high confidence in its prediction of zero even though the test point lies far from the data! This limitation can be alleviated by using an ensemble of *probabilistic* models instead of *deterministic* models. Even though the RBF network's predictions coincide with the predictive mean of the SE Gaussian process, by using a Gaussian process we could also make use of its *predictive variance*. The Gaussian process predictions far from the training data will have large model uncertainty. In the ensemble, we would thus wish to take into account each model's confidence as well as its mean (by sampling an output from each model's predictive distribution before calculating our sample variance). These ideas are assessed below in the context of Bayesian recurrent neural networks (§4.5).

2.2.3 Challenges

In the introduction (§1.5) we discussed what makes an approximate inference technique practical. We concluded that a technique should:

1. scale well to large data,
2. easily adapt to complex models (that can be as big as modern architecture would allow, or complex pipelines combining many building blocks),
3. not necessitate the change of existing model architectures (or objectives),

4. and should be easy for non-experts to use and understand.

The practicality of the techniques above is mixed, as discussed per technique. HMC for example, even though shown to obtain good results, does not scale to large data [Neal, 1995], and it is difficult to explain the technique to non-experts. α-divergence based techniques share this latter difficulty as well, limiting their use by non-experts.

In the next chapter we will develop an approximate inference technique for Bayesian NNs which will satisfy the requirements above. This will be demonstrated in the following chapters, where a large number of real-world use cases will be presented. Even more interesting, it will be shown that most modern deep learning models have been performing approximate Bayesian inference all along.