Journal of Machine Learning Research 1 (2020) 1-48 Submitted 4/00; Published 10/00

1. Uncertainty Methods

This Section is organized as follows: the first Subsection formally presents how uncertainty is quantified in Deep Learning with an introduction to Bayesian modeling. From this, we motivate a cross-category method comparison for subsequently analyzing individual-joint effectiveness in modeling predictive uncertainty. Subsection 1.2 treats predictive uncertainty methods with a focus on the algorithmic procedure, followed by representative method extensions for more reliable uncertainty estimation. Subsection 1.3 is devoted to uncertainty estimation: from what sources uncertainty originates, how to categorize different uncertainty measures, and how to quantify uncertainty at test-time with the methods from Subsection 1.2. In Subsection 1.4 we present the rationale of our study, connecting recent research on how NNs navigate the optimization landscape with the posterior approximation procedure of methods presented in the previous Subsection.

1.1 Quantifying Uncertainty in Deep Learning

In modern Deep Learning, two common uncertainty (or inversely "confidence") estimates are the prediction probability over classes, known as *softmax-score*, and the *predictive entropy* over posterior class probabilities (Shannon, 1948; Zaragoza and d'Alché Buc, 1998). However, Guo et al. (2017)'s work on confidence calibration demonstrated these to be unreliable estimates of Neural Networks' uncertainty.

Bayesian Deep Learning (BDL) methods build on solid mathematical foundations and hold promise for more reliable learned uncertainty estimates (Wilson, 2020). Bayesian Neural Networks (BNN) are in theory able to avoid the pitfalls of stochastic non-convex optimization on non-linear tunable functions with many high-dimensional parameters (MacKay, 1995). In their original formulation, BNNs come with high computational cost, since it involves learning a Gaussian distribution for each weight in the network, effectively doubling the number of parameters. The Bayesian approach consists of casting learning and prediction as an inference task about hypotheses (uncertain quantities, with θ representing all BNN parameters: weights w, biases b, and model structure) from data (measurable quantities, $\mathcal{D} = \left\{\left(\mathbf{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N} = (\mathbf{X}, \mathbf{Y})$). Drawing on the ground-laying works of Denker et al. (1987); MacKay (1992); Neal (1992); Hinton and Van Camp (1993), the "second-generation" in BDL (Ghahramani, 2016) is geared towards finding practical and scalable approximations to the analytically intractable Bayesian posterior (Eq. 1).

$$P(\theta \mid \mathcal{D}, m) = \frac{P(\mathcal{D} \mid \theta, m) P(\theta \mid m)}{P(\mathcal{D} \mid m)} \quad \begin{array}{ll} P(\mathcal{D} \mid \theta, m) & \text{likelihood of } \theta \text{ in model } m \\ P(\theta \mid m) & \text{prior probability of } \theta \\ P(\theta \mid \mathcal{D}, m) & \text{posterior of } \theta \text{ given data } \mathcal{D} \end{array} \tag{1}$$

Generating a prediction for a new test input x^* requires computing the conditional probability of x^* given the data and model.

$$P(x^* \mid \mathcal{D}, m) = \int P(x^* \mid \mathcal{D}, \theta, m) \underbrace{P(\theta \mid \mathcal{D}, m)}_{posterior} d\theta \tag{2}$$

To compute the *predictive distribution*, again the posterior distribution is required, whose integral via marginalization over the parameters is typically very high-dimensional and typically intractable:

$$P(\mathcal{D}, m) = \int P(\mathcal{D}, \theta, m) d\theta \tag{3}$$

Variational Inference (VI) is a Bayesian modeling technique employed by a majority of BDL methods that aim to circumvent the *inference problem.* The key idea consists of approximating the intractable posterior distribution $P(\theta \mid \mathcal{D}, m)$ with a simpler (though conjugate), tractable distribution $q(\theta)$. Specifically, by minimizing $KL(q||p)$, the Kullback–Leibler (KL) divergence between the approximating distribution and the replaced true posterior, one can perform Bayesian approximate inference under the guarantees of maximizing the *evidence lower bound* (ELBO) w.r.t. θ given data.

Diverse strategies have been proposed to principally and practically approximate the posterior distribution for beneficial uncertainty estimation. In what follows we will focus on methods that have seen more widespread adoption given their ability to scale both in network architecture and dataset size. In this, we follow the ensembling categorization of Ashukha et al. (2020), who discern two main strategies:

1. Obtaining snapshots of model parameters

2. Introducing stochasticity in the computation graph

The weight snapshots direction aims to find different sets of NN weights during training. Within this group, three sub-strategies exist: (i) the traditionally resource-expensive yet empirically effective ***Deep Ensemble*** trains independent sets of weights (Lakshminarayanan et al., 2016), (ii) snapshots are connected during different stages of training (Huang et al., 2017; Garipov et al., 2018; Maddox et al., 2019), or (iii) a sampling process such as Markov Chain Monte-Carlo (Gilks et al., 1995; Zhang et al., 2019; Hoffman, 2019) is used. Predictions are averaged across the snapshots during evaluation where model uncertainty can be estimated.
The stochastic computation-graph direction involves the introduction of noise over weights and/or activations during training, where each stochastic noise variant represents a model within the ensemble, and at inference time predictions are averaged over the members of the ensemble. Notable examples include ***Monte Carlo Dropout*** (Gal and Ghahramani, 2016), Batch Normalization (Atanov et al., 2018; Teye et al., 2018), and Stochastic Variational Inference (Graves, 2011; Blei et al., 2017; Farquhar et al., 2019).
In our benchmarking study we select at least 1 representative method (denoted by cursive emphasis) from each of the above categories, motivating a cross-category comparison and analyzing their individual-joint effectiveness in modeling predictive uncertainty.

1.2 Predictive Uncertainty Methods

We will first introduce each method by explaining the algorithm, followed by advantages or identified shortcomings, with subsequent method extensions from the same procedure category. Finally, we will zoom in on how to quantify uncertainty using each method.

1.2.1 Monte Carlo Dropout

The seminal work of Gal and Ghahramani (2016) on Monte Carlo Dropout (MC Dropout, MCD) proposes efficient model uncertainty estimation by exploiting dropout regularization as an approximate VI method. The authors reason that dropout training approximately integrates over model parameters by randomly masking (setting to 0) weight matrices, which is equivalent to drawing samples from a Bernoulli distribution given a fixed dropout probability. Concretely, for an L layers deep NN with weights W_l they define a variational Bernoulli distribution with Φ dropout rates (Eq. 4), where a Gaussian matrix G_l is multiplied with a diagonal matrix of Bernoulli random variable realizations (dropout masks), $diag[\mathbf{z}]$, drawn from Bernoulli($\boldsymbol{\theta}; \Phi$) for each set of weights $W_l = G_l diag[\mathbf{z}]$.

$$P(\boldsymbol{\theta} \mid \mathcal{D}) \approx q(\boldsymbol{\theta}; \Phi) = \text{Bernoulli}(\boldsymbol{\theta}; \Phi) \tag{4}$$

Additionally, they show that a Deep NN with dropout applied before every weight layer mathematically approximates a deep Gaussian Process (GP) (Rasmussen, 2003). This is an important comparison as GPs have desirable properties for principled uncertainty estimation.

In practice, the MCD procedure boils down to (i) applying dropout on all non-linear layers' weights, and (ii) activating dropout both during training and evaluation, wherein the latter predictions are averaged from T stochastic forward passes with different dropout masks to obtain an approximate posterior.

Quantifying "epistemic" *model uncertainty* using MCD involves applying dropout both during training and evaluation. In the latter case, T stochastic weights are sampled from the variational Bernoulli distribution $\hat{\theta}_t \sim q(\boldsymbol{\theta})$ to calculate the lower-order moments of the approximate Gaussian posterior, respectively the predictive mean and variance (Eq. 5).

$$\begin{aligned} \hat{\mu}_{pred}(\mathbf{x}^*) &= \frac{1}{T}\sum_{t=1}^{T} P(y^*|\mathbf{x}^*, \hat{\theta}_t) \\ \hat{\sigma}_{pred}(\mathbf{x}^*) &= \frac{1}{T}\sum_{t=1}^{T} [P(y^*|\mathbf{x}^*, \hat{\theta}_t) - \hat{\mu}_{pred}]^2 \end{aligned} \tag{5}$$

MCD's simplicity and computational tractability, i.e. dropout training is a standard DL practice and prediction only requires 1 model from which to sample (in parallel), has made it one of the most popular predictive uncertainty methods. However, there are some known limitations with MCD that have been explored in further works. An important shortcoming of VI, and in consequence MCD in Gal and Ghahramani (2016)'s formulation, is that they are known to underestimate predictive variance (Turner and Sahani, 2011). Whereas Gal and Ghahramani (2016) originally grounded the Bayesian interpretation of dropout in Variational Inference, Nalisnick et al. (2018) decouple dropout from inference and suggests a generalization of multiplicative noise to structured shrinkage priors. Their extension is dependent on the efficiency of Markov chain Monte Carlo (MCMC) sampling algorithms, which is why we do not consider it. We will touch on a selection of representative extensions in further subsections (1.2.3, 1.2.4).

1.2.2 Deep Ensemble

Deep Ensemble (Lakshminarayanan et al., 2016) (DE) involves independently training multiple probabilistic NNs with different random weight initializations and aggregating predictions from individual models. The empirical success of DE demonstrated that combinations of NNs trade-off computational resources for beneficial uncertainty estimation, robustness to dataset shift, and model quality improvements. As a downside, it has higher computational and memory complexity, since you need to train and store M models. A recent benchmarking survey (Ovadia et al., 2019) found DEs of a relatively small size (M=5) to be more robust to dataset shift and perform the best all-around compared to other popular BDL methods. In comparison to MC Dropout, DEs are treated as a uniformly-weighted Gaussian Mixture model, to which the formula for predictive variance is adapted:

$$\hat{\sigma}_{pred}(\mathbf{x}^*) = \frac{1}{M}\sum_{m}\left(\sigma^2_{\theta_m}(\mathbf{x}^*) + \mu^2_{\theta_m}(\mathbf{x}^*)\right) - \mu^2_*(\mathbf{x}^*) \tag{6}$$

The empirical performance increase of ensembles can be attributed to the diversity of (uncorrelated) errors between ensemble members. In the absence of diversity, ensembles lack good posterior approximation and for this reason, ensemble diversity promotion is a promising avenue for further improvements (Jain et al., 2020; Brazowski and Schneidman, 2020). The interplay between ensembling and regularization, "the effect of a prior", warrants more thought since not regularizing risks overfitting, while too strong regularization risks constraining diversity (cf. Subsection 1.4).

1.2.3 Concrete Dropout

As referred to in Section 1.2.1, the original MC Dropout definition with fixed-rate Bernoulli variational distribution suffers from uncertainty underestimation and miscalibration. To obtain well-calibrated uncertainty estimates, Osband (2016) made a case for manually tuning layer-wise dropout probability rates, since the dropout probability characterizes the overall posterior uncertainty. However, this grid-search is prohibitively expensive for deeper models.

Gal et al. (2017) proposes a **Con**tinuous-dis**crete** distribution relaxation to adapt and optimize the dropout probability p as a variational parameter using standard gradient descent. By taking advantage of the reparametrization trick, the Concrete distribution approximation $\tilde{\mathbf{z}}$ of the original Bernouilli random variable $\mathbf{z}$ conveniently parametrizes to a simple sigmoid distribution allowing for gradient-based optimization. Given a low temperature r (0.1) and a uniform random noise variable $\mathbf{u}$, the expression varies with respect to p, which if p >>0.5, sigmoid produces accelerated by a factor 10 a value approaching 1.

$$\tilde{\mathbf{z}} = \text{sigmoid}\left(\frac{1}{r}\cdot(\log p - \log(1-p) + \log \mathbf{u} - \log(1-\mathbf{u}))\right) \tag{7}$$

Concrete Dropout promises better-calibrated uncertainties at an almost negligible cost, consequently reducing experimentation time.

1.2.4 Heteroscedastic Extensions

Kendall and Gal (2017); Kwon et al. (2018); Xiao and Wang (2019) proposed similar approaches to extend MC Dropout predictive uncertainty to allow measuring uncertainty information from different sources.

Estimating input-dependent, “heteroscedastic aleatoric”, *data uncertainty* (detail Subsection 1.3.3) requires slightly modifying the model's architecture and objective function following Kendall and Gal (2017). Firstly, the output layer of model $f_{\hat{\theta}}$ is extended with a set of learnable variance variables $\boldsymbol{\sigma}$ per unique class output. The model's output logits, $\mathbf{v}$, are sampled from the stochastic output layer parametrized by $\mathcal{N}(f_{\hat{\theta}}(x), diag(\boldsymbol{\sigma}(x)^2))$. This model adaptation will be referred to as the *heteroscedastic model*. *Fig.* 1 visualizes the difference in output layer design.

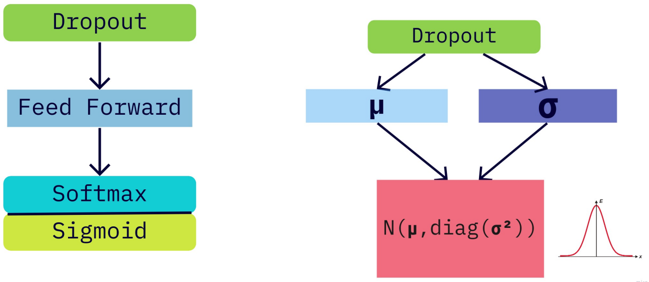

Figure 1: Visualization of output layer blocks. The left block denotes standard *softmax* (multi-class) or *sigmoid* (binary/multi-label) output. On the right, the *heteroscedastic* model outputs a normal distribution $\mathcal{N}(\boldsymbol{\mu}(x), diag(\boldsymbol{\sigma}(x)^2)$ parametrizing mean and variance by the logits coming from two separate preceding feedforward layers.

Next, it requires incorporating a residual *heteroscedastic loss*:

$$\mathcal{L}_{\text{HET}}(\hat{\theta}) = \sum_{i=1}^{N} \log \frac{1}{T} \sum_{t=1}^{T} \exp\left(\mathbf{v}_{i,c}^{(t)} - \log \sum_{k} \exp \mathbf{v}_{i,k}^{(t)} \right) + \log T \tag{8}$$

with N the number of training examples passing through an instance t of the model $f_{\hat{\theta}_t}(x)$ + $\boldsymbol{\sigma}^{(t)}$ to generate for example i a sampled logit vector $\mathbf{v}_i^t$, where predicted value for class k, $\mathbf{v}_{i,k}^{(t)}$, and c the index of the ground truth class. By learning to predict log variance with T dropout-masked samples, the model will be able to predict high variance (uncertainty) for inputs where the predictive mean is far removed from the true observation, which by design has a smaller effect on the total loss. This uncertainty modeling method is referred to as *Learned Loss Attenuation*.

1.3 Uncertainty Estimation

In this Subsection, we will introduce sources of uncertainty, how uncertainty is quantified in practice, followed by a categorization of uncertainty.

1.3.1 Total Uncertainty

Classification models trained by minimizing negative log-likelihood (i.e. cross-entropy) quantify global uncertainty over class outcomes with entropy (H) over logits. Therefore, the entropy of the posterior predictive distribution can be determined a measure of both data uncertainty and model uncertainty (Hüllermeier and Waegeman, 2019). In our definition of predictive uncertainty, we consider both **calibration** and **robustness**. Model misspecification, noisy data or supervision all contribute to the total uncertainty.

Decomposing total uncertainty into the different sources is beneficial for determining actions to evaluate the room for improvement. Model uncertainty is reducible by collecting more data, whereas data uncertainty cannot be decreased by model refinements or including more data.

1.3.2 Model Uncertainty

Epistemic uncertainty presents the inherent "uncertainty" (Osband, 2016) of the model with regards to the true values for its parameters and structure after having seen the training data. The model will communicate ignorance because of lack of knowledge, evidenced by a broad posterior over parameters. In principle, this quantity of uncertainty can be reduced by feeding more data, choosing a more expressive model, or by finding more appropriate values for the hyperparameters. The quantity is hypothesized (e.g. Seedat and Kanan (2019)) to increase when presented with test inputs far removed from the training distribution.
Mutual Information (MI) (Smith and Gal, 2018) has been proposed as a separate measure of epistemic uncertainty, contrasting to predictive entropy which is high given any model uncertainty or data noise. Intuitively, the measure captures the amount of information that would be gained about model parameters through knowledge of the true outcome (Eq. 9).

$$\mathrm{MI}(\boldsymbol{\theta}, y \mid \mathcal{D}, \mathbf{x}^*) = H(y \mid \mathbf{x}^*, \mathcal{D}) - \mathop{\mathbb{E}}_{p(\theta \mid \mathcal{D})}[H(y \mid \mathbf{x}^*, \theta)] \tag{9}$$

1.3.3 Data Uncertainty

Aleatoric uncertainty captures the inherent stochasticity and noise in data. Moreover, it cannot be explained away when feeding the model more data. It can be further decomposed into a *homoscedastic* component, which represents constant noise over inputs such as the numerical accurateness of a measuring device, and *heteroscedastic* uncertainty representing input-dependent noise generated by the data by, for instance, class overlap, complex decision boundaries or label noise. Including the quantification of heteroscedastic data uncertainty allows for the expression of instance-level uncertainty. Even when having frequently observed a sample during training, instance-level data noise should be expressed together with the best possible prediction.

1.3.4 Uncertainty categorization

Below follows a categorization of the uncertainty quantities within the scope of the experiments. To estimate for a new test sample x^* the prediction and uncertainty of model $f_{\hat{\theta}}(x^*)$ we typically seek to obtain the predictive posterior distribution $P(y^*|x^*, \hat{\theta})$ over class membership probabilities with $y^*_k \in \{1, \ldots, K\}$.

For MC Dropout at inference time, we presume $P(y^*|x^*, \hat{\theta}) \approx \frac{1}{T}\sum_{t=1}^{T} P(y^*|x^*, \hat{\theta}_t)$, with prediction obtained after applying softmax/sigmoid function for sample t, $\hat{p}_t = P(y^*|x^*, \hat{\theta}_t)$, and predictive mean $\bar{p} = \frac{1}{T}\sum_{t=1}^{T} \hat{p}_t$. For Deep Ensemble, the above notations would require a change from T to M, but for consistency over quantity formulas, we maintain T to denote posterior sampling. For ease of notation, we define a helper entropy function on $H(x^*, z) = -\sum_{k=1}^{K} P(y_k|x^*, z) \log P(y_k|x^*, z)$.

Quantity	*Formula*
Softmax-score	$S = \arg\max_{k} \frac{\exp f_{\hat{\theta},k}(x^*)}{\sum_{i=1}^{K} \exp f_{\hat{\theta},i}(x^*)}$
Predictive Entropy	$H_{pred} = H(x^*, \hat{\theta})$
Mutual Information	$I = H_{pred} - \frac{1}{T} \sum_{t=1}^{T} H(x^*, \hat{\theta}_t)$
Model Uncertainty	$\hat{\sigma}_{model} = \frac{1}{T} \sum_{t=1}^{T} (\hat{p}_t - \bar{p})^2$
Data Uncertainty	$\hat{\sigma}_{data} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\sigma}_k^{(t)}(x^*)$

For any classification model, it is possible to compute the softmax-score and predictive entropy. For multi-label classification, softmax-score does not take into account multiple winning classes and a standard approximation would be to average over the sigmoid-scaled probabilities of predicted classes.

Model uncertainty can be quantified with Monte Carlo integration or the aggregation of individual models. In practice, it is quantified by either (a) calculating the average sigmoid/softmax variance over the predictive mean from MC samples or (b) computing the total variance from an ensemble mixture distribution (Eq. 6). Changing to the heteroscedastic extensions allows to quantify data uncertainty. More specifically, data uncertainty is quantified with as surrogate the average over variance logits $\boldsymbol{\sigma}$ (see *Fig.* 1). Whenever ensembling is applied where a single model estimates a quantity, one typically averages over the ensemble components' uncertainty.

1.4 Motivating Hybrid Approaches

This Subsection will motivate the theorized complementarity of VI-based and ensembling methods for improved uncertainty estimation and robustness. It has been hinted at by prior work (Fort et al., 2019; Wilson, 2020) but never empirically analyzed, on no account in a text classification benchmarking setting.

All predictive uncertainty methods in scope share the goal of approximating the posterior distribution over model parameters θ, which represent hypotheses learned from the data. However, in light of the empirical success of Deep Ensembles, recent research (Fort et al., 2019) raises an important question concerning the difference in function-space between variational Bayesian NNs (MC Dropout and extensions) and Deep Ensemble. We rebuild their argument below and couple it to our hypothesis on why the combination of both approaches can improve uncertainty reliability.

NNs are parametrized stochastic functions presenting a non-convex optimization problem, which concerns multiple feasible regions with multiple locally optimal points within each. With maximum-a-posteriori (MAP) estimation, the network is expected to interpolate to

the training data, finding parameter/weights values (*hypotheses*) for which the loss function is low by navigating the high-dimensional loss landscape. Once model training converges to optimized parameter values, one ends up with a weight-space *solution*, representing a single *mode* (unique functions f_θ).
The true posterior is generally a highly complex and multimodal distribution, with multiple possible but not necessarily equivalent parametrizations θ able to fit the training data. To accurately quantify posterior uncertainty, it is wishful to capture as many modes or separated regions as possible.

Correspondingly, the common goal is to achieve reliable uncertainty and, following the BDL paradigm, one resorts to modeling a Bayesian posterior. What differs among the selected predictive uncertainty methods, is the form of the prior and likelihood, from which to determine a procedure. For text classification, the likelihood model is typically defined by a cross-entropy loss and the same data-generating process is shared for all procedures. Below we expound on the **difference in posterior approximation procedure**:

- MC Dropout (and extensions) is a common VI procedure (Hron et al., 2018), which applies Bernoulli dropout and Gaussian (L2) priors on weight-space, assuming a posterior Gaussian distribution from which to draw stochastic samples at test-time. VI-based methods tend to locally approximate uncertainty surrounding a single mode, **intra-modal** posterior approximation. More specifically, the procedure followed by MC Dropout approximates to a spike-and-slab marginal prior with typically very peaked variance, a plausible reason for approximated uncertainty centered tightly around 1 single mode.
- An ensemble of NNs makes no direct assumptions on the form or distribution of the prior and just "expects" to sample well from the posterior. It generates a series of MAP estimates which through inherent stochasticity in weight initialization and optimization end up at different regions in weight space, leading to functionally dissimilar but equally accurate modes of the solution space. Ensembles are very effective at exploration in weight-space and happen to converge to multiple modes, allowing for **inter-modal** posterior approximation. Furthermore, by considering more possible hypotheses it will be better at approximating multimodal posterior distributions and avoid the collapse to a single mode.

Combining both procedures is to generate a mixture over priors, which in itself is again a prior, all under the same likelihood function. There is no guarantee that a combination of methods from both procedures captures the true posterior, yet in our work we will empirically analyse if and why combining inter and intra-modal posterior approximation offers the hypothesized complementary benefits.

References

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.

Andrei Atanov, Arsenii Ashukha, Dmitry Molchanov, Kirill Neklyudov, and Dmitry Vetrov. Uncertainty estimation via stochastic batch normalization. *arXiv:1802.04893 [cs, stat]*, March 2018. arXiv: 1802.04893.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859–877, Feb 2017.

Benjamin Brazowski and Elad Schneidman. Collective learning by ensembles of altruistic diversifying neural networks. *arXiv:2006.11671 [cs, stat]*, June 2020. arXiv: 2006.11671.

John Denker, Daniel Schwartz, Ben Wittner, Sara Solla, Richard Howard, Lawrence Jackel, and John Hopfield. Large automatic learning, rule extraction, and generalization. *Complex systems*, 1(5):877–922, 1987.

Sebastian Farquhar, Michael Osborne, and Yarin Gal. Radial Bayesian neural networks: Beyond discrete support in large-scale Bayesian deep learning, 2019. arXiv:1907.00865.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete Dropout. In *Advances in neural information processing systems*, pages 3581–3590, 2017.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. *arXiv:1802.10026 [cs, stat]*, October 2018. arXiv: 1802.10026.

Zoubin Ghahramani. A history of bayesian neural networks. In *NIPS Workshop on Bayesian Deep Learning*, 2016.

W.R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics. Taylor & Francis, 1995. ISBN 9780412055515. URL `http://books.google.com/books?id=TRXrMWY_i2IC`.

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

Matt Hoffman. Langevin Dynamics as Nonparametric Variational Inference. 2019.

Jiri Hron, Alexander G. de G. Matthews, and Zoubin Ghahramani. Variational bayesian dropout: pitfalls and fixes. In *ICML*, 2018.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get m for free, 2017.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction. *arXiv preprint arXiv:1910.09457*, 2019.

Siddhartha Jain, Ge Liu, Jonas Mueller, and David Gifford. Maximizing overall diversity for improved uncertainty estimates in deep ensembles. In *AAAI*, pages 4264–4271, 2020.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. Uncertainty quantification using bayesian neural networks in classification. page 13, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2016.

David JC MacKay. *Bayesian methods for adaptive models.* PhD thesis, California Institute of Technology, 1992.

David JC MacKay. Probable networks and plausible predictions—a review of practical bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469–505, 1995.

Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning, 2019.

Eric Nalisnick, José Miguel Hernández-Lobato, and Padhraic Smyth. Dropout as a Structured Shrinkage Prior, 2018.

Radford M Neal. Bayesian mixture modeling. In *Maximum Entropy and Bayesian Methods*, pages 197–211. Springer, 1992.

Ian Osband. Risk versus uncertainty in deep learning : Bayes , bootstrap and the dangers of dropout. 2016.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you Trust your Model's Uncertainty? Evaluating Predictive Uncertainty under Dataset Shift. In *Advances in Neural Information Processing Systems*, pages 13991–14002, 2019.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

Nabeel Seedat and Christopher Kanan. Towards calibrated and scalable uncertainty representations for neural networks, 2019.

Claude E Shannon. A Mathematical Theory of Communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. 2018. URL `http://arxiv.org/abs/1803.08533`.

Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian Uncertainty Estimation for Batch Normalized Deep Networks. *arXiv:1802.06455 [stat]*, July 2018. URL `http://arxiv.org/abs/1802.06455`. arXiv: 1802.06455.

R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, T. Cemgil, and S. Chiappa, editors, *Bayesian Time series models*, chapter 5, pages 109–130. Cambridge University Press, 2011.

Andrew Gordon Wilson. The Case for Bayesian Deep Learning. *arXiv preprint arXiv:2001.10995*, 2020.

Yijun Xiao and William Yang Wang. Quantifying Uncertainties in Natural Language Processing Tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7322–7329, 2019.

Hugo Zaragoza and Florence d'Alché Buc. Confidence Measures for Neural Network Classifiers. In *Proceedings of the Seventh Int. Conf. Information Processing and Management of Uncertainty in Knowlegde Based Systems*, 1998.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. *arXiv:1902.03932 [cs, stat]*, February 2019. URL `http://arxiv.org/abs/1902.03932`. arXiv: 1902.03932.