Dropout Inference in Bayesian Neural Networks with Alpha-divergences

Yingzhen Li [1]   **Yarin Gal** [1] [2]

Abstract

To obtain uncertainty estimates with real-world Bayesian deep learning models, practical inference approximations are needed. Dropout variational inference (VI) for example has been used for machine vision and medical applications, but VI can severely underestimates model uncertainty. Alpha-divergences are alternative divergences to VI's KL objective, which are able to avoid VI's uncertainty underestimation. But these are hard to use in practice: existing techniques can only use Gaussian approximating distributions, and require existing models to be changed radically, thus are of limited use for practitioners. We propose a re-parametrisation of the alpha-divergence objectives, deriving a simple inference technique which, together with dropout, can be easily implemented with existing models by simply changing the loss of the model. We demonstrate improved uncertainty estimates and accuracy compared to VI in dropout networks. We study our model's epistemic uncertainty far away from the data using adversarial images, showing that these can be distinguished from non-adversarial images by examining our model's uncertainty.

1. Introduction

Deep learning models have been used to obtain state-of-the-art results on many tasks (Krizhevsky et al., 2012; Szegedy et al., 2014; Sutskever et al., 2014; Sundermeyer et al., 2012; Mikolov et al., 2010; Kalchbrenner & Blunsom, 2013), and in many pipelines these models have replaced the more traditional *Bayesian probabilistic* models (Sennrich et al., 2016). But unlike deep learning models, Bayesian probabilistic models can capture parameter uncertainty and its induced effects over predictions, capturing the models' ignorance about the world, and able to convey their increased uncertainty on out-of-data examples. This

information can be used, for example, to identify when a vision model is given an adversarial image (studied below), or to tackle many problems in AI safety (Amodei et al., 2016). With model uncertainty at hand, applications as far-reaching as safety in self-driving cars can be explored, using models which can propagate their uncertainty up the decision making pipeline (Gal, 2016). With deterministic deep learning models this invaluable uncertainty information is often lost.

Bayesian deep learning – an approach to combining Bayesian probability theory together with deep learning – allows us to use state-of-the-art models and at the same time obtain model uncertainty (Gal, 2016; Gal & Ghahramani, 2016a). Originating in the 90s (Neal, 1995; MacKay, 1992; Denker & LeCun, 1991), Bayesian neural networks (BNNs) in particular have started gaining in popularity again (Graves, 2011; Blundell et al., 2015; Hernandez-Lobato & Adams, 2015). BNNs are standard neural networks (NNs) with prior probability distributions placed over their weights. Given observed data, inference is then performed to find what are the more likely and less likely weights to explain the data. But as easy it is to formulate BNNs, is as difficult to perform inference in them. Many approximations have been proposed over the years (Denker & LeCun, 1991; Neal, 1995; Graves, 2011; Blundell et al., 2015; Hernandez-Lobato & Adams, 2015; Hernández-Lobato et al., 2016), some more practical and some less practical. A practical approximation for inference in Bayesian neural networks should be able to scale well to large data and complex models (such as convolutional neural networks (CNNs) (Rumelhart et al., 1985; LeCun et al., 1989)). Much more importantly perhaps, it would be impractical to change existing model architectures that have been well studied, and it is often impractical to work with complex and cumbersome techniques which are difficult to explain to non-experts. Many existing approaches to obtain model confidence often do not scale to complex models or large amounts of data, and require us to develop new models for existing tasks for which we already have well performing tools (Gal, 2016).

One possible solution for practical inference in BNNs is variational inference (VI) (Jordan et al., 1999), a ubiquitous technique for approximate inference. Dropout variational distributions in particular (a mixture of two Gaussians with

---

[1]University of Cambridge, UK [2]The Alan Turing Institute, UK. Correspondence to: Yingzhen Li <yl494@cam.ac.uk>.

small standard deviations, and with one component fixed at zero) can be used to obtain a practical inference technique (Gal & Ghahramani, 2016b). These have been used for machine vision and medical applications (Kendall & Cipolla, 2016; Kendall et al., 2015; Angermueller & Stegle, 2015; Yang et al., 2016). Dropout variational inference can be implemented by adding dropout layers (Hinton et al., 2012; Srivastava et al., 2014) before every weight layer in the NN model. Inference is then carried out by Monte Carlo (MC) integration over the variational distribution, in practice implemented by simulating stochastic forward passes through the model at test time (referred to as MC dropout). Although dropout VI is a practical technique for approximate inference, it also has some major limitations. Dropout VI can severely underestimate model uncertainty (Gal, 2016, Section 3.3.2) – a property many VI methods share (Turner & Sahani, 2011). This can lead to devastating results in applications that *must rely* on good uncertainty estimates such as AI safety applications.

Alternative objectives to VI's objective are therefore needed. Black-box α-divergence minimisation (Hernández-Lobato et al., 2016; Li & Turner, 2016; Minka, 2005) is a class of approximate inference methods extending on VI, approximating EP's energy function (Minka, 2001) as well as the Hellinger distance (Hellinger, 1909). These were proposed as a solution to some of the difficulties encountered with VI. However, the main difficulty with α-divergences is that the divergences are hard to use in practice. Existing inference techniques only use Gaussian approximating distributions, with the density over the approximation having to be evaluated explicitly many times. The objective offers a limited intuitive interpretation which is difficult to explain to non-experts, and of limited use for engineers (Gal, 2016, Section 2.2.2). Perhaps more importantly, current α-divergence inference techniques require existing models and code-bases to be changed radically to perform inference in the Bayesian counterpart to these models. To implement a complex CNN structure with the inference and code of (Hernández-Lobato et al., 2016), for example, one would be required to re-implement many already-implemented software tools.

In this paper we propose a re-parametrisation of the induced α-divergence objectives, and by relying on some mild assumptions (justified below), derive a simple approximate inference technique which can easily be implemented with existing models. Further, we rely on the dropout approximate variational distribution and demonstrate how inference can be done in a practical way – requiring us to *only change the loss of the NN, $\mathcal{L}(\theta)$, and to perform multiple stochastic forward passes at training time*. Precisely, given $l(\cdot, \cdot)$ some standard NN loss such as cross entropy or the Euclidean loss, and $\{\mathbf{f}^{\widehat{\omega}_k}(\boldsymbol{x}_n)\}_{k=1}^{K}$ a set of K stochastic dropout network outputs on input $\mathbf{x}_n$

with randomly masked weights $\widehat{\omega}_k$, our proposed objective is:

$$\mathcal{L}(\theta) = -\frac{1}{\alpha} \sum_n \text{log-sum-exp} \left[-\alpha l(y_n, \mathbf{f}^{\widehat{\omega}_k}(\boldsymbol{x}_n)) \right] + L_2(\theta)$$

with α a real number, θ the set of network weights to be optimised, and an L_2 regulariser over θ. By selecting $\alpha = 1$ this objective directly optimises the per-point predictive log-likelihood, while picking $\alpha \to 0$ would focus on increasing the training accuracy, recovering VI.

Specific choices of α will result in improved uncertainty estimates (and accuracy) compared to VI in dropout BNNs, without slowing convergence time. We demonstrate this through a myriad of applications, including an assessment of fully connected NNs in regression and classification, and an assessment of Bayesian CNNs. Finally, we study the uncertainty estimates resulting from our approximate inference technique. We show that our models' uncertainty increases on adversarial images generated from the MNIST dataset, suggesting that these lie outside of the training data distribution. This in practice allows us to tell-apart such adversarial images from non-adversarial images by examining epistemic model uncertainty.

2. Background

We review background in Bayesian neural networks and approximate variational inference. In the next section we discuss α-divergences.

2.1. Bayesian Neural Networks

Given training inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and their corresponding outputs $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, in parametric Bayesian regression we would like to infer a distribution over parameters ω of a function $\mathbf{y} = \mathbf{f}^{\omega}(\mathbf{x})$ that could have generated the outputs. Following the Bayesian approach, to find parameters that could have generated our data, we put some *prior* distribution over the space of parameters $p_0(\omega)$. This distribution captures our prior belief as to which parameters are likely to have generated our outputs before observing any data. We further need to define a probability distribution over the outputs given the inputs $p(\mathbf{y}|\mathbf{x}, \omega)$. For classification tasks we assume a softmax likelihood,

$$p(y|\mathbf{x}, \omega) = \text{Softmax}(\mathbf{f}^{\omega}(\mathbf{x}))$$

or a Gaussian likelihood for regression. Given a dataset $\mathbf{X}, \mathbf{Y}$, we then look for the *posterior* distribution over the space of parameters: $p(\omega|\mathbf{X}, \mathbf{Y})$. This distribution captures how likely the function parameters are, given our observed data. With it we can predict an output for a new input point $\mathbf{x}^*$ by integrating

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \omega) p(\omega|\mathbf{X}, \mathbf{Y}) \text{d}\omega. \quad (1)$$

One way to define a distribution over a parametric set of functions is to place a prior distribution over a *neural network's* weights $\omega = \{\mathbf{W}_i\}_{i=1}^L$, resulting in a *Bayesian NN* (MacKay, 1992; Neal, 1995). 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, $p_0(\mathbf{W}_i) = \mathcal{N}(\mathbf{W}_i; \mathbf{0}, \mathbf{I})$ and often assume a point estimate for the bias vectors for simplicity.

2.2. Approximate Variational Inference in Bayesian Neural Networks

In approximate inference, 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 $\mathbf{X}, \mathbf{Y}$: $p(\omega|\mathbf{X}, \mathbf{Y})$, which is not tractable in general. Existing approaches to approximate this posterior are through *variational inference* (as was done in Hinton & Van Camp (1993); Barber & Bishop (1998); Graves (2011); Blundell et al. (2015)). We need to define an approximating variational distribution $q_\theta(\omega)$ (parametrised by variational parameters θ), and then minimise w.r.t. θ the KL divergence (Kullback & Leibler, 1951; Kullback, 1959) between the approximating distribution and the full posterior:

$$
\begin{aligned}
&\mathrm{KL}\big(q_\theta(\omega)||p(\omega|\mathbf{X}, \mathbf{Y})\big) \\
&\propto -\int q_\theta(\omega) \log p(\mathbf{Y}|\mathbf{X}, \omega)\mathrm{d}\omega + \mathrm{KL}(q_\theta(\omega)||p_0(\omega)) \\
&= -\sum_{i=1}^N \int q_\theta(\omega) \log p(\mathbf{y}_i|\mathbf{f}^\omega(\mathbf{x}_i))\mathrm{d}\omega + \mathrm{KL}(q_\theta(\omega)||p_0(\omega)),
\end{aligned}
$$

where $A \propto B$ is slightly abused here to denote equality up to an additive constant (w.r.t. variational parameters θ).

2.3. Dropout Approximate Inference

Given a (deterministic) neural network, stochastic regularisation techniques in the model (such as dropout (Hinton et al., 2012; Srivastava et al., 2014)) can be interpreted as variational Bayesian approximations in a Bayesian NN with the same network structure (Gal & Ghahramani, 2016b). This is because applying a stochastic regularisation technique is equivalent to multiplying the NN weight matrices $\mathbf{M}_i$ by some random noise $\boldsymbol{\epsilon}_i$ (with a new noise realisation for each data point). The resulting stochastic weight matrices $\mathbf{W}_i = \boldsymbol{\epsilon}_i \mathbf{M}_i$ can be seen as draws from the approximate posterior over the BNN weights, replacing the deterministic NN's weight matrices $\mathbf{M}_i$. Our set of variational parameters is then the set of matrices $\theta = \{\mathbf{M}_i\}_{i=1}^L$. For example, dropout can be seen as an approximation to Bayesian NN inference with *dropout approximating distributions*, where the rows of the matrices $\mathbf{W}_i$ distribute according to a mixture of two Gaussians with small variances

and the mean of one of the Gaussians fixed at zero. The uncertainty in the weights induces prediction uncertainty by marginalising over the approximate posterior using Monte Carlo integration:

$$
\begin{aligned}
p(y = c|\mathbf{x}, \mathbf{X}, \mathbf{Y}) &= \int p(y = c|\mathbf{x}, \omega)p(\omega|\mathbf{X}, \mathbf{Y})\mathrm{d}\omega \\
&\approx \int p(y = c|\mathbf{x}, \omega)q_\theta(\omega)\mathrm{d}\omega \\
&\approx \frac{1}{K}\sum_{k=1}^K p(y = c|\mathbf{x}, \widehat{\boldsymbol{\omega}}_k)
\end{aligned}
$$

with $\widehat{\boldsymbol{\omega}}_k \sim q_\theta(\omega)$, where $q_\theta(\omega)$ is the Dropout distribution (Gal, 2016). Given its popularity, we concentrate on the dropout stochastic regularisation technique throughout the rest of the paper, although any other stochastic regularisation technique could be used instead (such as multiplicative Gaussian noise (Srivastava et al., 2014) or dropConnect (Wan et al., 2013)).

Dropout VI is an example of practical approximate inference, but it also underestimates model uncertainty (Gal, 2016, Section 3.3.2). This is because minimising the KL divergence between $q(\omega)$ and $p(\omega|\mathbf{X}, \mathbf{Y})$ penalises $q(\omega)$ for placing probability mass where $p(\omega|\mathbf{X}, \mathbf{Y})$ has no mass, but does not penalise $q(\omega)$ for not placing probability mass at locations where $p(\omega|\mathbf{X}, \mathbf{Y})$ *does have mass*. We next discuss α-divergences as an alternative to the VI objective.

3. Black-box α-divergence minimisation

In this section we provide a brief review of the *black box alpha* (BB-α, Hernández-Lobato et al. (2016)) method upon which the main derivation in this paper is based. Consider approximating the following distribution:

$$
p(\omega) = \frac{1}{Z}p_0(\omega)\prod_n f_n(\omega).
$$

In Bayesian neural networks context, these factors $f_n(\omega)$ represent the likelihood terms $p(\mathbf{y}_n|\mathbf{x}_n, \omega)$, $Z = p(\mathbf{Y}|\mathbf{X})$, and the approximation target $p(\omega)$ is the exact posterior $p(\omega|\mathbf{X}, \mathbf{Y})$. Popular methods of approximate inference include variational inference (VI) (Jordan et al., 1999) and expectation propagation (EP) (Minka, 2001), where these two algorithms are special cases of power EP (Minka, 2004) that minimises Amari's α-divergence (Amari, 1985) $\mathrm{D}_\alpha[p||q]$ in a *local* way:

$$
\mathrm{D}_\alpha[p||q] = \frac{1}{\alpha(1-\alpha)}\left(1 - \int p(\omega)^\alpha q(\omega)^{1-\alpha}d\omega\right).
$$

We provide details of α-divergences and local approximation methods in the appendix, and in the rest of the paper we consider three special cases in this rich family:

1. Exclusive KL divergence:

$$D_0[p||q] = KL[q||p] = \mathbb{E}_q\left[\log\frac{q(\omega)}{p(\omega)}\right];$$

2. Hellinger distance:

$$D_{0.5}[p||q] = 4\text{Hel}^2[q||p] = 2\int\left(\sqrt{p(\omega)} - \sqrt{q(\omega)}\right)^2 d\omega;$$

3. Inclusive KL divergence:

$$D_1[p||q] = KL[p||q] = \mathbb{E}_p\left[\log\frac{p(\omega)}{q(\omega)}\right].$$

Since $\alpha = 0$ is used in VI and $\alpha = 1.0$ is used in EP, in later sections we will also refer to these alpha settings as the VI value, Hellinger value, and EP value, respectively.

Power-EP, though providing a generic variational framework, does not scale with big data. It maintains approximating factors attached to every likelihood term $f_n(\omega)$, resulting in space complexity $\mathcal{O}(N)$ for the posterior approximation which is clearly undesirable. The recently proposed stochastic EP (Li et al., 2015) and BB-α (Hernández-Lobato et al., 2016) inference methods reduce this memory overhead to $\mathcal{O}(1)$ by sharing these approximating factors. Moreover, optimisation in BB-α is done by descending the so called BB-α energy function, where Monte Carlo (MC) methods and automatic differentiation are also deployed to allow fast prototyping.

BB-α has been successfully applied to Bayesian neural networks for regression, classification (Hernández-Lobato et al., 2016) and model-based reinforcement learning (Depeweg et al., 2016). They all found that using $\alpha \neq 0$ often returns better approximations than the VI case. The reasons for the worse results of VI are two fold. From the perspective of inference, due to the zero-forcing behaviour of exclusive KL discussed before, VI often fits to a local mode of the exact posterior and is over-confident in prediction. On hyper-parameter learning point of view, as the variational lower-bound is used as a (biased) approximation to the maximum likelihood objective, the learned model could be biased towards over-simplified cases (Turner & Sahani, 2011). These problems could potentially be addressed by using α-divergences. For example, inclusive KL encourages the coverage of the support set (referred as mass-covering), and when used in local divergence minimisation (Minka, 2005), it can fit an approximation to a mode of $p(\omega)$ with better estimates of uncertainty. Moreover the BB-α energy provides a better approximation to the marginal likelihood as well, meaning that the learned model will be less biased and thus fitting the data distribution better (Li & Turner, 2016). Hellinger distance seems to provide a good balance between zero-forcing and mass-covering, and empirically it has been found to achieve the best performance.

Given the success of α-divergence methods, it is a natural idea to extend these algorithms to other classes of approximations such as dropout. However this task is non-trivial. First, the original formulation of BB-α energy is an ad hoc adaptation of power-EP energy (see appendix), which applies to exponential family q distributions only. Second, the energy function offers a limited intuitive interpretation to non-experts, thus of limited use for practitioners. Third and most importantly, a naive implementation of BB-α using dropout would bring in a prohibitive computational burden. To see this, we first review the BB-α energy function in the general case (Li & Turner, 2016) given $\alpha \neq 0$:

$$\mathcal{L}_\alpha(q) = -\frac{1}{\alpha}\sum_n \log\mathbb{E}_q\left[\left(\frac{f_n(\omega)p_0(\omega)^{\frac{1}{N}}}{q(\omega)^{\frac{1}{N}}}\right)^\alpha\right]. \quad (2)$$

One could verify that this is the same energy function as presented in (Hernández-Lobato et al., 2016) by considering q an exponential family distribution. In practice (2) might be intractable, hence an MC approximation is introduced:

$$\mathcal{L}_\alpha^{\text{MC}}(q) = -\frac{1}{\alpha}\sum_n \log\frac{1}{K}\sum_k\left[\left(\frac{f_n(\widehat{\omega}_k)p_0(\widehat{\omega}_k)^{\frac{1}{N}}}{q(\widehat{\omega}_k)^{\frac{1}{N}}}\right)^\alpha\right] \quad (3)$$

with $\widehat{\omega}_k \sim q(\omega)$. This is a biased approximation as the expectation in (2) is computed before taking the logarithm. But empirically Hernández-Lobato et al. (2016) showed that the bias introduced by the MC approximation is often dominated by the variance of the samples, meaning that the effect of the bias is negligible. When $\alpha \to 0$ it returns the *variational free energy* (the VI objective)

$$\mathcal{L}_0(q) = \mathcal{L}_{\text{VFE}}(q) = KL[q||p_0] - \sum_n\mathbb{E}_q\left[\log f_n(\omega)\right], \quad (4)$$

and the corresponding MC approximation $\mathcal{L}_{\text{VFE}}^{\text{MC}}$ becomes an unbiased estimator of $\mathcal{L}_{\text{VFE}}$. Also $\mathcal{L}_\alpha^{\text{MC}} \to \mathcal{L}_{\text{VFE}}^{\text{MC}}$ as the number of samples $K \to 1$.

The original paper (Hernández-Lobato et al., 2016) proposed a naive implementation which directly evaluates the MC estimation (3) with samples $\widehat{\omega}_k \sim q(\omega)$. However as discussed before, dropout implicitly samples different masked weight matrices $\widehat{\omega} \sim q$ for different data points. This indicates that the naive approach, when applied to dropout approximation, would gather all these samples for all M datapoints in a mini-batch (i.e. MK sets of neural network weight matrices in total), which brings prohibitive cost if the network is wide and deep. Interestingly, the minimisation of the variational free energy ($\alpha = 0$) with the dropout approximation can be computed very efficiently. The main reason for this success is due to the additive structure of the variational free energy: no evaluation of q density is required if the "regulariser" $KL[q||p_0]$ can be computed/approximated efficiently. In the following section we

propose an improved version of BB-α energy to allow applications with dropout and other flexible approximation structures.

4. A New Reparameterisation of BB-α Energy

We propose a reparamterisation of the BB-α energy to reduce the computational overhead. First we denote $\tilde{q}(\omega)$ as a free-form "cavity distribution" (see appendix), and write the approximate posterior q as

$$q(\omega) = \frac{1}{Z_q} \tilde{q}(\omega) \left(\frac{\tilde{q}(\omega)}{p_0(\omega)} \right)^{\frac{\alpha}{N-\alpha}}, \qquad (5)$$

where we assume $Z_q < +\infty$ is the normalising constant to ensure q a valid distribution. When $\alpha/N \to 0$, the unnormalised density in (5) converges to $\tilde{q}(\omega)$ for every ω, and $Z_q \to 1$ by the assumption of $Z_q < +\infty$ (Van Erven & Harremoës, 2014). Hence $q \to \tilde{q}$ when $\alpha/N \to 0$, and this happens for example when we choose $\alpha \to 0$, or $N \to +\infty$ as well as when α grows sub-linearly to N. Now we rewrite the BB-alpha energy in terms of $\tilde{q}$:

$$\mathcal{L}_\alpha(q) = -\frac{1}{\alpha} \sum_n \log \int \left(\frac{1}{Z_q} \tilde{q}(\omega) \left(\frac{\tilde{q}(\omega)}{p_0(\omega)} \right)^{\frac{\alpha}{N-\alpha}} \right)^{1-\frac{\alpha}{N}}$$
$$p_0(\omega)^{\frac{\alpha}{N}} f_n(\omega)^\alpha d\omega$$
$$= -\frac{1}{\alpha} \sum_n \left(\int \tilde{q}(\omega) f_n(\omega)^\alpha - \left(1 - \frac{\alpha}{N} \right) \log Z_q \right)$$
$$= \frac{N}{\alpha} \left(1 - \frac{\alpha}{N} \right) \log \int \tilde{q}(\omega) \left(\frac{\tilde{q}(\omega)}{p_0(\omega)} \right)^{\frac{\alpha}{N-\alpha}} d\omega$$
$$- \frac{1}{\alpha} \sum_n \log \mathbb{E}_{\tilde{q}} [f_n(\omega)^\alpha]$$
$$= \mathrm{R}_\beta[\tilde{q}||p_0] - \frac{1}{\alpha} \sum_n \log \mathbb{E}_{\tilde{q}} [f_n(\omega)^\alpha], \; \beta = \frac{N}{N-\alpha},$$

where $\mathrm{R}_\beta[\tilde{q}||p_0]$ represents the *Rényi divergence* (Rényi (1961), see appendix) of order β. Furthermore, provided $\mathrm{R}_\beta[\tilde{q}||p_0] < +\infty$ (which holds when assuming $Z_q < +\infty$), we have $\mathrm{R}_\beta[\tilde{q}||p_0] \to \mathrm{KL}[\tilde{q}||p_0] = \mathrm{KL}[q||p_0]$ as $\frac{\alpha}{N} \to 0$. This means that for a constant α that scales sublinearly with N, in large data settings we can further approximate the BB-α energy as

$$\mathcal{L}_\alpha(q) \approx \tilde{\mathcal{L}}_\alpha(q) = \mathrm{KL}[q||p_0] - \frac{1}{\alpha} \sum_n \log \mathbb{E}_q [f_n(\omega)^\alpha].$$

Note that here we also use the fact that now $q \approx \tilde{q}$. Critically, the proposed reparameterisation is continuous in α, and by taking $\alpha \to 0$ the variational free-energy (4) is recovered.

Given a loss function $l(\cdot, \cdot)$, e.g. l_2 loss in regression or cross entropy in classification, we can define the (unnormalised) likelihood term $f_n(\omega) \propto p(\mathbf{y}_n | \boldsymbol{x}_n, \omega) \propto$

$\exp[-l(\mathbf{y}_n, \mathbf{f}^\omega(\boldsymbol{x}_n))]$, e.g. see (LeCun et al., 2006)[1]. Swapping $f_n(\omega)$ for this last expression, and approximating the expectation over q using Monte Carlo sampling, we obtain our proposed minimisation objective:

$$\tilde{\mathcal{L}}_\alpha^{\mathrm{MC}}(q) = \mathrm{KL}[q||p_0] + \mathrm{const} \qquad (6)$$
$$- \frac{1}{\alpha} \sum_n \text{log-sum-exp}[-\alpha l(y_n, \mathbf{f}^{\widehat{\omega}_k}(\boldsymbol{x}_n))]$$

with log-sum-exp being the log-sum-exp operator over K samples from the approximate posterior $\widehat{\omega}_k \sim q(\omega)$. This objective function also approximates the marginal likelihood. Therefore, compared to the original formulation (2), the improved version (6) is considerably simpler (both to implement and to understand), has a similar form to standard objective functions used in deep learning research, yet remains an approximate Bayesian inference algorithm.

To gain some intuitive understanding of this objective, we observe what it reduces to for different α and K settings. By selecting $\alpha = 1$ the per-point predictive log-likelihood $\log \mathbb{E}_q[p(y_n | \mathbf{x}_n, \omega)]$ is directly optimised. On the other hand, picking the VI value ($\alpha \to 0$) would focus on increasing the training accuracy $\mathbb{E}_q[\log p(y_n | \mathbf{x}_n, \omega)]$. The Hellinger value could be used to achieve a balance between reducing training error and improving predictive likelihood, which has been found to be desirable (Hernández-Lobato et al., 2016; Depeweg et al., 2016). Lastly, for $K = 1$ the log-sum-exp disappears, the α's cancel out, and the original (stochastic) VI objective is recovered.

In summary, our proposal modifies the loss function by multiplying it by α and then performing log-sum-exp with a sum over multiple stochastic forward passes sampled from the BNN approximate posterior. The remaining KL-divergence term (between q and the prior p) can often be approximated. It can be viewed as a regulariser added to the objective function, and reduces to L_2-norm regulariser for certain popular q choices (Gal, 2016).

4.1. Dropout BB-α

We now provide a concrete example where the approximate distribution is defined by dropout. With dropout VI, MC samples are used to approximate the expectation w.r.t. q, which in practice is implemented as performing *stochastic forward passes* through the dropout network – i.e. given an input $\mathbf{x}$, the input is fed through the network and a new dropout mask is sampled and applied at each dropout layer. This gives a stochastic output – a sample from the dropout network on the input $\mathbf{x}$. A similar approximation is used in our case as well, where to implement the MC sampling in eq. (6) we perform multiple stochastic forward passes

---

[1] $f_n(\omega)$ does not need to be a normalised density of y_n unless one would like to optimise the associated hyper parameters.

through the network.

Recall the neural network $\mathbf{f}^\omega(\mathbf{x})$ is parameterised by the variable ω. In classification, cross entropy is often used as the loss function $l(\mathbf{y}, \mathbf{f}^\omega(\boldsymbol{x})) = -\mathbf{y}^T \log \mathbf{p}^\omega(\mathbf{x})$, where the label $\mathbf{y}_n$ is a one-hot binary vector, and the network output $\mathbf{p}^\omega(\mathbf{x}_n) = \text{Softmax}(\mathbf{f}^\omega(\boldsymbol{x}_n))$ encodes the probability vector of class assignments. Applying the re-formulated BB-α energy (6) with a Bayesian equivalent of the network, we arrive at the objective function

$$\tilde{\mathcal{L}}_\alpha^{\text{MC}}(q) = \sum_i p_i ||\mathbf{M}_i||_2^2 - \frac{1}{\alpha} \sum_n \mathbf{y}_n^T \log \frac{1}{K} \sum_k (\mathbf{p}^{\widehat{\omega}_k}(\mathbf{x}_n))^\alpha$$

$$= \frac{1}{\alpha} \sum_n l\left(\mathbf{y}_n, \frac{1}{K} \sum_k \mathbf{p}^{\widehat{\omega}_k}(\mathbf{x}_n)^\alpha\right) + \sum_i L_2(\mathbf{M}_i)$$

with $\{\mathbf{p}^{\widehat{\omega}_k}(\mathbf{x}_n)\}_{k=1}^K$ being K stochastic network outputs on input $\mathbf{x}_n$, p_i equals to one minus the dropout rate of the ith layer, and the L_2 regularization terms coming from an approximation to the KL-divergence (Gal, 2016). I.e. we raise network probability outputs to the power α and average them as an input to the standard cross entropy loss. Taking $\alpha \neq 1$ can be viewed as training the neural network with an adjusted "power" loss, regularized by an L_2 norm. Implementing this induced loss with Keras (Chollet, 2015) is as simple as a few lines of Python. A code snippet is given in Figure 1, with more details in the appendix.

In regression problems, the loss function is defined as $l(\boldsymbol{y}, \mathbf{f}^\omega(\boldsymbol{x})) = \frac{\tau}{2}||\boldsymbol{y} - \mathbf{f}^\omega(\boldsymbol{x})||_2^2$ and the likelihood term can be interpreted as $\mathbf{y} \sim \mathcal{N}(\boldsymbol{y}; \mathbf{f}^\omega(\boldsymbol{x}), \tau^{-1}\boldsymbol{I})$. Plugging this into the energy function returns the following objective

$$\tilde{\mathcal{L}}_\alpha^{\text{MC}}(q) = -\frac{1}{\alpha} \sum_n \text{log-sum-exp} \left[-\frac{\alpha\tau}{2}||\mathbf{y}_n - \mathbf{f}^{\widehat{\omega}_k}(\boldsymbol{x}_n)||_2^2\right]$$

$$+ \frac{ND}{2} \log \tau + \sum_i p_i ||\boldsymbol{M}_i||_2^2, \tag{7}$$

with $\{\mathbf{f}^{\widehat{\omega}_k}(\mathbf{x}_n)\}_{k=1}^K$ being K stochastic forward passes on input $\mathbf{x}_n$. Again, this is reminiscent of the l_2 objective in standard deep learning, and can be implemented by simply passing the input through the dropout network multiple times, collecting the stochastic outputs, and feeding the set of outputs through our new BB-alpha loss function.

5. Experiments

We test the reparameterised BB-α on Bayesian NNs with the dropout approximation. We assess the proposed inference in regression and classification tasks on standard benchmarking datasets, comparing different values of α. This last experiment leads us to propose a technique that could be used to identify adversarial image attacks. In the appendix we further provide a study of run time trade-off.

```python
def softmax_cross_ent_with_mc_logits(alpha):
  def loss(y_true, mc_logits):
    # mc_logits: MC samples of shape MxKxD
    mc_log_softmax = mc_logits \
      - K.max(mc_logits, axis=2, keepdims=True)
    mc_log_softmax = mc_log_softmax - \
      logsumexp(mc_log_softmax, 2)
    mc_ll = K.sum(y_true*mc_log_softmax,-1)
    return -1./alpha * (logsumexp(alpha * \
      mc_ll, 1) + K.log(1.0 / K_mc))
  return loss
```

Figure 1. Code snippet for our induced classification loss.

5.1. Regression

The first experiment considers Bayesian neural network regression with approximate posterior induced by dropout. We use benchmark UCI datasets[2] that have been tested in related literature. The model is a single-layer neural network with 50 ReLU units for all datasets except for Protein and Year, which use 100 units. We consider $\alpha \in \{0.0, 0.5, 1.0\}$ in order to examine the effect of mass-covering/zero-forcing behaviour in dropout. MC approximation with $K = 10$ samples is also deployed to compute the energy function. Other initialisation settings are largely taken from (Li & Turner, 2016).

We summarise the test negative log-likelihood (LL) and RMSE with standard error (across different random splits, the lower the better) for selected datasets in Figure 2 and 3, respectively. The full results are provided in the appendix. Although optimal α may vary for different datasets, using non-VI values has significantly improved the test-LL performances, while remaining comparable in test error metric. In particular, $\alpha = 0.5$ produced overall good results for both test LL and RMSE, which is consistent with previous findings. We also compare with a BNN with a Gaussian approximation (VI-G) (Li & Turner, 2016), a BNN with HMC, and a sparse Gaussian process model with 50 inducing points (Bui et al., 2016). In test-LL metric our best dropout model out-performs the Gaussian approximation method on almost all datasets, and for some datasets is on par with HMC which is the current gold standard for Bayesian neural works, and with the GP model that is known to be superior in regression.

5.2. Classification

We further experiment with a classification task, comparing the accuracy of the various α values on the MNIST benchmark (LeCun & Cortes, 1998). We assessed a fully connect NN with 2 hidden layers and 100 units in each layer. We used dropout probability 0.5 and $\alpha \in \{0, 0.5, 1\}$. Again, we use $K = 10$ samples at training time for all α values,

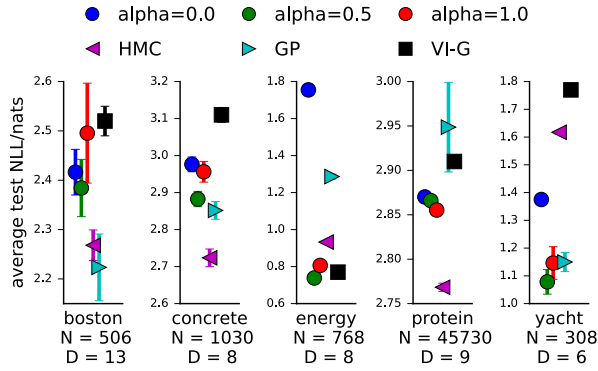[2] http://archive.ics.uci.edu/ml/datasets. html

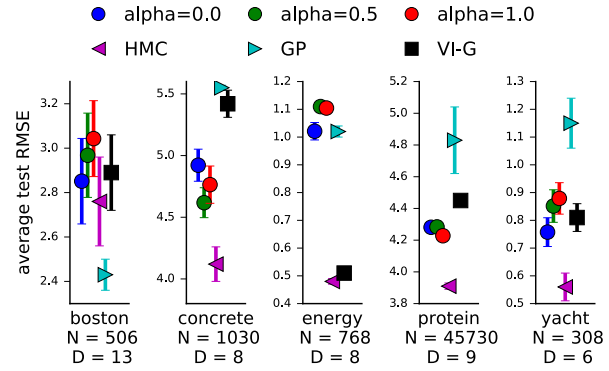*Figure 2.* Negative test-LL results for Bayesian NN regression.

Figure 3. Test RMSE results for Bayesian NN regression.

and $K_{\text{test}} = 100$ samples at test time. We use weight decay 10^{-6}, which is equivalent to prior lengthscale $l^2 = 0.1$ (Gal & Ghahramani, 2016b). We repeat each experiment three times and plot mean and standard error. Test RMSE as well as test log likelihood are given in Figure 4. As can be seen, Hellinger value $\alpha = 0.5$ gives best test RMSE, with test log likelihood matching that of the EP value $\alpha = 1$. The VI value $\alpha = 0$ under-performs according to both metrics.

We next assess a convolutional neural network model (CNN). For this experiment we use the standard CNN example given in (Chollet, 2015) with 32 convolution filters, 100 hidden units at the top layer, and dropout probability 0.5 before each fully-connected layer. Other settings are as before. Average test accuracy and test log likelihood are given in Figure 5. In this case, VI value $\alpha = 0$ seems to supersede the EP value $\alpha = 1$, and performs similarly to the Hellinger value $\alpha = 0.5$ according to both metrics.

5.3. Detecting Adversarial Examples

The third set of experiments considers adversarial attacks on dropout-trained Bayesian neural networks. We test the hypothesis that certain techniques for generating adversarial examples will give images that lie outside of the image

manifold, i.e. far from the data distribution (note though that there exist techniques that will guarantee the images staying near the data manifold, by minimising the perturbation used to construct the adversarial example). By assessing the BNN uncertainty, we should see increased uncertainty for adversarial images if they indeed lie outside of the training data distribution. The tested models are fully connected networks with 3 hidden layers of 1000 units trained using dropout rate 0.5 and different alpha values. These models are also compared to a benchmark MLP with the same architecture but trained by maximum likelihood. The adversarial examples are generated on MNIST test data that is normalised to be in the range $[0, 1]$. For the dropout trained networks we perform MC dropout at test time with $K_{\text{test}} = 10$ MC samples.

The first attack in consideration is the Fast Gradient Sign (FGS) method (Goodfellow et al., 2014). This is an untargeted attack, which attempts to reduces the maximum value of the predicted class label probability

$$\mathbf{x}_{\text{adv}} = \mathbf{x} - \eta \cdot \text{sgn}(\nabla_{\mathbf{x}} \max_y \log p(y|\mathbf{x})).$$

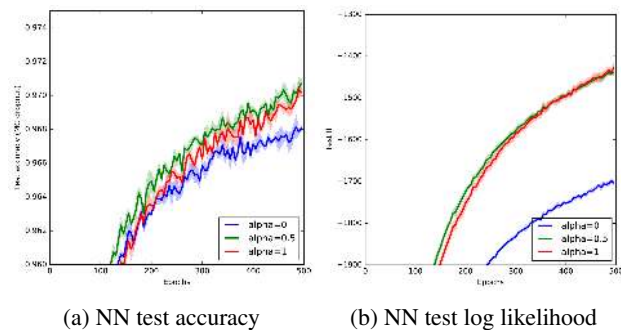We use the single gradient step FGS implemented in Cleverhans (Papernot et al., 2016) with the stepsize η varied

(a) NN test accuracy

(b) NN test log likelihood

Figure 4. MNIST test accuracy and test log likelihood for a fully connected NN in a classification task.
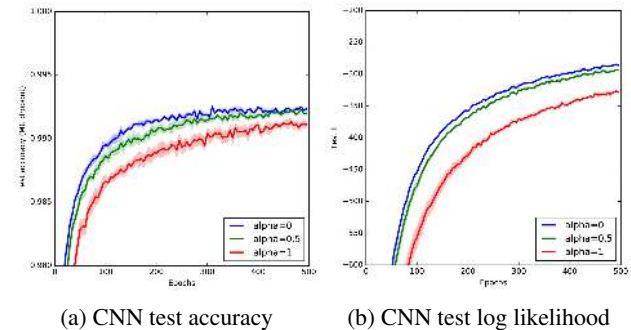

(a) CNN test accuracy

(b) CNN test log likelihood

Figure 5. MNIST test accuracy and test log likelihood for a convolutional neural network in a classification task.

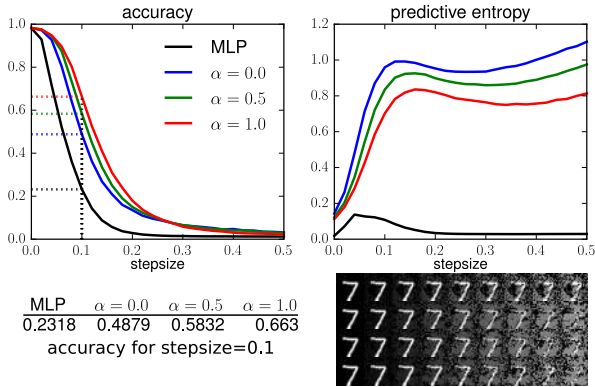| accuracy			
MLP	$\alpha = 0.0$	$\alpha = 0.5$	$\alpha = 1.0$
0.2318	0.4879	0.5832	0.663

accuracy for stepsize=0.1

Figure 6. Un-targeted attack: classification accuracy results as a function of perturbation stepsize. The adversarial examples are shown for (from top to bottom) MLP and BNN trained with dropout and $\alpha = 0.0, 0.5, 1.0$.

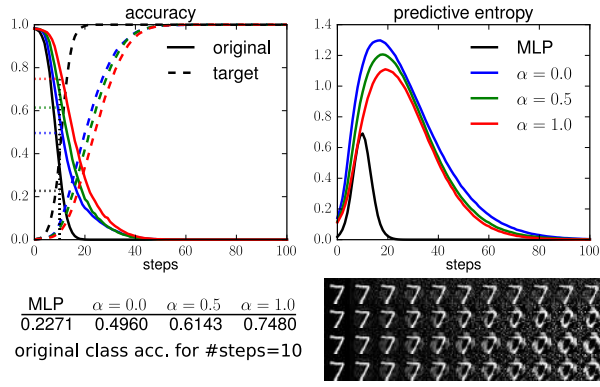| accuracy			
MLP	$\alpha = 0.0$	$\alpha = 0.5$	$\alpha = 1.0$
0.2271	0.4960	0.6143	0.7480

original class acc. for #steps=10

Figure 7. Targeted attack: classification accuracy results (on both original and target class) as a function of the number of iterative gradient steps. Note the log scale x-axis in the left panel.

between 0.0 and 0.5. The left panel in Figure 6 demonstrates the classification accuracy on adversarial examples, which shows that the dropout networks, especially the one trained with $\alpha = 1.0$, are significantly more robust to adversarial attacks compared to the deterministic NN. For example, for $\eta = 0.1$ the adversarial samples still visually close to the original class, and the BNN trained with $\alpha = 0.0$ achieves an accuracy level almost 3 times higher than the MLP and around 20% higher than the VI-trained version. More interestingly, the test data examples and adversarial images can be told-apart by investigating the uncertainty representation of the dropout models. In the right panel of Figure 6 we depict the predictive entropy computed on the neural network output probability vector, and show example corresponding adversarial images below the axis for each corresponding stepsize. Clearly the deterministic NN model produces over-confident predictions on adversarial samples, e.g. it predicts the wrong label very confidently even when the input is still visually close to digit "7" ($\eta = 0.2$). While dropout models, though producing wrong labels, are very uncertain about their predictions. This uncertainty keeps increasing as we move away from the data manifold. Hence the dropout networks are much more immunised from noise-corrupted inputs, as they can be detected using uncertainty estimates in this example.

The second attack we consider is a targeted version of FGS (Goodfellow et al., 2014; Carlini & Wagner, 2016), which maximises the predictive probability of a selected class instead. As an example, we fix class 0 as the target and apply the iterative gradient-base attack to all non-zero digits in test data. At step t, the adversarial output is computed as

$$\mathbf{x}_{\text{adv}}^t = \mathbf{x}_{\text{adv}}^{t-1} + \eta \cdot \text{sgn}(\nabla_{\mathbf{x}} \log p(y_{\text{target}} | \mathbf{x}_{\text{adv}}^{t-1})),$$

where the stepsize η is fixed at 0.01 in this case. Results are presented in the left panel of Figure 7, and again dropout

trained models are more robust to this attack compared with the MLP. Similarly these adversarial examples could be detected by the Bayesian neural networks' uncertainty, by examining the predictive entropy. By visually inspecting the generated adversarial examples in the right panel of Figure 7, it is clear that the MLP overconfidently classifies a digit 7 to class 0. On the other hand, the dropout models are still fairly uncertain about their predictions even after 40 gradient steps. More interestingly, running this iterative attack on dropout models produces a smooth interpolation between different digits, and when the model is confident on predicting the target class, the corresponding adversarial images are visually close to digit zero.

These initial results suggest that assessing the epistemic uncertainty of classification models can be used as a viable technique to identify adversarial examples. We would note though that we used this experiment to demonstrate our techniques' uncertainty estimates, and much more research is needed to solve the difficulties faced with adversarial inputs.

6. Conclusions

We presented a practical extension of the BB-alpha objective which allows us to use the technique with dropout approximating distributions. The technique often supersedes existing approximate inference techniques (even sparse Gaussian processes), and is easy to implement. A code snippet for our induced loss is given in the appendix.

Acknowledgements

References

Amari, Shun-ichi. *Differential-Geometrical Methods in Statistic*. Springer, New York, 1985.

Amodei, Dario, Olah, Chris, Steinhardt, Jacob, Christiano, Paul, Schulman, John, and Mane, Dan. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Angermueller, C and Stegle, O. Multi-task deep neural network to predict CpG methylation profiles from low-coverage sequencing data. In *NIPS MLCB workshop*, 2015.

Barber, David and Bishop, Christopher M. Ensemble learning in Bayesian neural networks. *NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES*, 168:215–238, 1998.

Blundell, Charles, Cornebise, Julien, Kavukcuoglu, Koray, and Wierstra, Daan. Weight uncertainty in neural network. In *ICML*, 2015.

Bui, Thang D, Hernández-Lobato, Daniel, Li, Yingzhen, Hernández-Lobato, José Miguel, and Turner, Richard E. Deep gaussian processes for regression using approximate expectation propagation. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, 2016.

Carlini, Nicholas and Wagner, David. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*, 2016.

Chollet, Francois. Keras. https://github.com/fchollet/keras, 2015.

Denker, John and LeCun, Yann. Transforming neural-net output levels to probability distributions. In *Advances in Neural Information Processing Systems 3*. Citeseer, 1991.

Depeweg, Stefan, Hernández-Lobato, José Miguel, Doshi-Velez, Finale, and Udluft, Steffen. Learning and policy search in stochastic dynamical systems with bayesian neural networks. *arXiv preprint arXiv:1605.07127*, 2016.

Gal, Yarin. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Gal, Yarin and Ghahramani, Zoubin. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *ICLR workshop track*, 2016a.

Gal, Yarin and Ghahramani, Zoubin. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *ICML*, 2016b.

Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Graves, Alex. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, pp. 2348–2356, 2011.

Hellinger, Ernst. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 136:210–271, 1909.

Hernandez-Lobato, Jose Miguel and Adams, Ryan. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *ICML*, 2015.

Hernández-Lobato, José Miguel, Li, Yingzhen, Hernández-Lobato, Daniel, Bui, Thang, and Turner, Richard E. Black-box alpha divergence minimization. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1511–1520, 2016.

Hinton, Geoffrey E and Van Camp, Drew. Keeping the neural networks simple by minimizing the description length of the weights. In *COLT*, pp. 5–13. ACM, 1993.

Hinton, Geoffrey E, Srivastava, Nitish, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Jordan, Michael I, Ghahramani, Zoubin, Jaakkola, Tommi S, and Saul, Lawrence K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Kalchbrenner, Nal and Blunsom, Phil. Recurrent continuous translation models. In *EMNLP*, 2013.

Kendall, Alex and Cipolla, Roberto. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4762–4769. IEEE, 2016.

Kendall, Alex, Badrinarayanan, Vijay, and Cipolla, Roberto. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Kullback, Solomon. *Information theory and statistics*. John Wiley & Sons, 1959.

Kullback, Solomon and Leibler, Richard A. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

LeCun, Yann and Cortes, Corinna. The mnist database of handwritten digits, 1998.

LeCun, Yann, Boser, Bernhard, Denker, John S, Henderson, Donnie, Howard, Richard E, Hubbard, Wayne, and Jackel, Lawrence D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

LeCun, Yann, Chopra, Sumit, Hadsell, Raia, Ranzato, M, and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1:0, 2006.

Li, Yingzhen and Turner, Richard E. Rényi divergence variational inference. In *NIPS*, 2016.

Li, Yingzhen, Hernández-Lobato, José Miguel, and Turner, Richard E. Stochastic expectation propagation. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

MacKay, David JC. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

Mikolov, Tomáš, Karafiát, Martin, Burget, Lukáš, Černockỳ, Jan, and Khudanpur, Sanjeev. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

Minka, Tom. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

Minka, T.P. Expectation propagation for approximate Bayesian inference. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001.

Minka, T.P. Power EP. Technical Report MSR-TR-2004-149, Microsoft Research, 2004.

Neal, Radford M. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

Papernot, Nicolas, Goodfellow, Ian, Sheatsley, Ryan, Feinman, Reuben, and McDaniel, Patrick. cleverhans v1.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2016.

Rényi, Alfréd. On measures of entropy and information. *Fourth Berkeley symposium on mathematical statistics and probability*, 1, 1961.

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

Sennrich, Rico, Haddow, Barry, and Birch, Alexandra. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pp. 371–376, Berlin, Germany, August 2016. Association for Computational Linguistics.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Sundermeyer, Martin, Schlüter, Ralf, and Ney, Hermann. LSTM neural networks for language modeling. In *INTERSPEECH*, 2012.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc VV. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

Turner, RE and Sahani, M. Two problems with variational expectation maximisation for time-series models. *Inference and Estimation in Probabilistic Time-Series Models*, 2011.

Van Erven, Tim and Harremoës, Peter. Rényi divergence and Kullback-Leibler divergence. *Information Theory, IEEE Transactions on*, 60(7):3797–3820, 2014.

Wan, L, Zeiler, M, Zhang, S, LeCun, Y, and Fergus, R. Regularization of neural networks using dropconnect. In *ICML-13*, 2013.

Yang, Xiao, Kwitt, Roland, and Niethammer, Marc. Fast predictive image registration. *arXiv preprint arXiv:1607.02504*, 2016.