Bayesian Neural Network Ensembles

Tim Pearce
Department of Engineering
University of Cambridge
`tp424@cam.ac.uk`

Mohamed Zaki
Department of Engineering
University of Cambridge

Andy Neely
Department of Engineering
University of Cambridge

1 Introduction

Ensembles of neural networks (NNs) have long been used to estimate predictive uncertainty (Tibshirani, 1996; Heskes, 1996); a small number of NNs are trained from different initialisations and sometimes on differing versions of the dataset. The variance of the ensemble's predictions is interpreted as its epistemic uncertainty. The appeal of ensembling stems from being a collection of regular NNs - this makes them both scalable and easily implementable.

NN ensembles have continued to achieve strong empirical results in recent years, for example in Lakshminarayanan et al. (2017), where it was presented as a practical alternative to more costly Bayesian NNs (BNNs). The departure from Bayesian methodology is of concern since the Bayesian framework provides a principled, widely-accepted approach to handling uncertainty.

Several recent works have explored links between ensembles and Bayesian inference. Variants of an ensembling scheme known to be consistent for Bayesian linear regression have been applied directly to NNs (Lu and Van Roy, 2017; Osband et al., 2017). In this extended abstract we derive and implement a modified ensembling scheme specifically for NNs, which provides a consistent estimator of the Bayesian posterior in wide NNs - regularising parameters about values drawn from a prior distribution.

2 Randomised MAP Sampling

Recent work in the Bayesian community, and independently in the reinforcement learning community, has begun to explore an approach to Bayesian inference that will be novel to many readers. Roughly speaking, it exploits the fact that adding a regularisation term to a loss function returns maximum a posteriori (MAP) estimates of parameters with normally distributed priors centred at zero (MacKay, 1992). For a regression problem this loss is of the form,

$$Loss_{regularise} = \frac{1}{N}||\mathbf{y} - \hat{\mathbf{y}}||_2^2 + \frac{1}{N}||\mathbf{\Gamma}^{1/2}\boldsymbol{\theta}||_2^2, \tag{1}$$

where $\mathbf{y}$ is a vector of targets, $\hat{\mathbf{y}}$ is the NN's predictions, $\boldsymbol{\theta}$ is a flattened vector of NN parameters, and $\mathbf{\Gamma}$ is a diagonal square regularisation matrix with it's k^{th} diagonal element representing the ratio of data noise variance to prior variance for parameter θ_k. Data noise is assumed normally distributed and homoskedastic in this work.

Injecting noise into this loss, either to targets or regularisation term, and sampling repeatedly (i.e. ensembling), produces a *distribution* of MAP solutions which can approximate the true posterior. This can be an efficient method to sample from high-dimensional posteriors (Bardsley et al., 2014).

Whilst it is straightforward to select the noise distribution that produces exact inference in linear regression models, there is difficulty in transferring this idea to NNs. Directly applying the noise distribution from the linear case to NNs has had some empirical success, despite not reproducing the true posterior (Lu and Van Roy, 2017; Osband et al., 2018). A more accurate, though more

computationally demanding solution, is to wrap the optimisation step into an MCMC procedure (Bardsley, 2012; Bardsley et al., 2014). We name this family of schemes *randomised MAP sampling*.

2.1 Normally Distributed Prior and Likelihood

We consider randomised MAP sampling for the case of multivariate normal prior and (normalised) likelihood, $\mathcal{N}(\boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{prior})$, $\mathcal{N}(\boldsymbol{\mu}_{like}, \boldsymbol{\Sigma}_{like})$. The posterior, also multivariate normal, is given by Bayes rule, $\mathcal{N}(\boldsymbol{\mu}_{post}, \boldsymbol{\Sigma}_{post}) \propto \mathcal{N}(\boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{prior}) \cdot \mathcal{N}(\boldsymbol{\mu}_{like}, \boldsymbol{\Sigma}_{like})$. The MAP solution is simply $\boldsymbol{\mu}_{post}$, for which a standard result exists,

$$\boldsymbol{\mu}_{post} = (\boldsymbol{\Sigma}_{like}^{-1} + \boldsymbol{\Sigma}_{prior}^{-1})^{-1}(\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\mu}_{like} + \boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\mu}_{prior}). \tag{2}$$

In randomised MAP sampling we are interested in injecting noise so that $\mathbb{V}ar[\boldsymbol{\mu}_{post}] = \boldsymbol{\Sigma}_{post}$. Previous work analysing linear regression found that injecting noise into both $\boldsymbol{\mu}_{prior}$ and $\boldsymbol{\mu}_{like}$ can provide a consistent estimator of the true posterior. However, beyond the linear case this approach fails as manipulation of $\boldsymbol{\mu}_{like}$ via targets, $\mathbf{y}$, is complex and creates conflicts amongst parameters.

If instead $\boldsymbol{\mu}_{prior}$ is chosen as the sole noise source, this problem is avoided. In order to inject this noise, let us replace $\boldsymbol{\mu}_{prior}$ with some noisy random variable, $\boldsymbol{\theta}_0$, and denote $\boldsymbol{\mu}_{post}^{MAP}(\boldsymbol{\theta}_0)$ the resulting MAP estimate,

$$\boldsymbol{\mu}_{post}^{MAP}(\boldsymbol{\theta}_0) = (\boldsymbol{\Sigma}_{like}^{-1} + \boldsymbol{\Sigma}_{prior}^{-1})^{-1}(\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\mu}_{like} + \boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\theta}_0), \tag{3}$$

which could be found in practise by minimisation of a slightly modified 'anchored' loss function,

$$Loss_{anchor} = \frac{1}{N}||\mathbf{y} - \hat{\mathbf{y}}||_2^2 + \frac{1}{N}||\boldsymbol{\Gamma}^{1/2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)||_2^2. \tag{4}$$

Derivation of the noise distribution required for $\boldsymbol{\theta}_0$ is found from eq. 3, setting $\mathbb{E}[\boldsymbol{\mu}_{post}^{MAP}(\boldsymbol{\theta}_0)] = \boldsymbol{\mu}_{post}$ and $\mathbb{V}ar[\boldsymbol{\mu}_{post}^{MAP}(\boldsymbol{\theta}_0)] = \boldsymbol{\Sigma}_{post}$. We find $\boldsymbol{\theta}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ with,

$$\boldsymbol{\mu}_0 = \boldsymbol{\mu}_{prior}, \quad \boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_{prior} + \boldsymbol{\Sigma}_{prior}^2 \boldsymbol{\Sigma}_{like}^{-1}. \tag{5}$$

3 Application to NNs

Although the previous section's result is of interest, evaluating eq. 5 requires knowing the likelihood covariance, $\boldsymbol{\Sigma}_{like}$. Estimating this for a NN is far from simple: NNs are unidentifiable, their likelihood variances and correlations vary greatly across parameters, and shift during training. This impasse can be solved in a surprising way. From eq. 5 we see that $diag(\boldsymbol{\Sigma}_0) \geq diag(\boldsymbol{\Sigma}_{prior})$. In fact, with increasing NN width, H, the term $\boldsymbol{\Sigma}_{prior}^2 \boldsymbol{\Sigma}_{like}^{-1}$ tends to a zero matrix (see appendix for proof). Therefore choosing this lower bound and setting $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_{prior}$ is valid for wide NNs.

Stepping back, we note $\boldsymbol{\Sigma}_0 \approx \boldsymbol{\Sigma}_{prior}$ is only true in the case the posterior is dominated by the prior distribution rather than the likelihood. This occurs in BNNs because the role of priors is slightly abused as a source of regularisation in an over-paramatised model. This observation is significant as it allows us to **relax our assumption that the prior and likelihood be normally distributed**. Instead, we can say that our method is valid provided the posterior is dominated by the prior.

A surprisingly simple result remains: a wide NN minimising the loss function in eq. 4, and with $\boldsymbol{\theta}_0 \sim \mathcal{N}(\boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{prior})$, provides a consistent estimator of the posterior.

3.1 Number in the Ensemble

If each NN is a single posterior sample, it might be expected that an inordinate number are required to capture the true posterior parameter distributions. But the parameter distributions themselves are of little interest in the context of a NN, it is the predictive distribution that is of sole interest. In this way we **move from doing inference in parameter space to output space**. Given that each NN provides an independent sample from a posterior predictive distribution, a relatively small number of NNs can
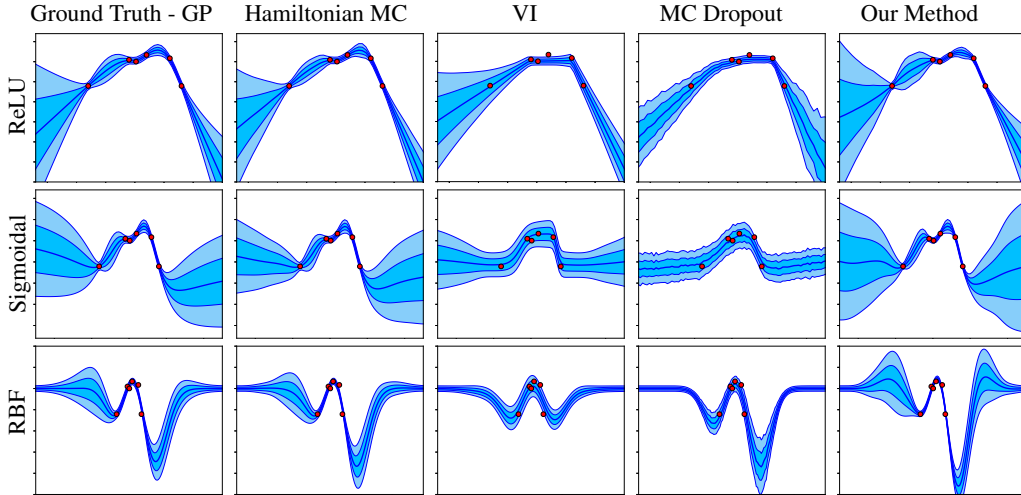
Figure 1: Predictive distributions produced by various inference methods (columns) for various activation functions (rows), e.g. bottom right is a RBF NN with inference by our method.

give a good approximation. An ensemble size of 5-10 worked well in experiments. This number does not increase with dimensionality of input or output.

4   Experiments

In figure 1 we compare predictive distributions produced by popular Bayesian inference methods in wide (100 node) single-layer NNs, with our method, on a toy regression problem. We used several non-linearities for which analytical GP kernels exist - ReLU, ERF (sigmoidal) and RBF. GP and HMC produce 'gold standard' Bayesian inference, and we judge the remaining methods, which are scalable approximations, to them. 'VI' denotes mean-field variational inference with Gaussian approximating distributions. 'MC Dropout' refers to the popular method proposed in Gal & Ghahramani (2015). 'Our method' implements the scheme described in this work, with ten NNs per ensemble.

The predictive distributions produced by our method appear good, if slightly wavy, approximations of gold standard inference. However, there does appear to be a tendency to over predict the variance. It captures uncertainty in interpolated regions significantly better than VI and MC Dropout, neither of which account for correlations between parameters.

These plots illustrate one more important point. An example of when ensembling fails to perform Bayesian inference was provided by Gal (2016) [p. 27]: an ensemble of RBF NNs would output zero with high confidence when predicting far from the training data, and this would not be the case for the equivalent RBF GP which was the squared exponential (SE) kernel. However, the RBF GP is *not* the SE kernel except in the special case of infinite variance priors (Williams, 1996). Figure 1, bottom left, shows an actual RBF GP with finite variance. In fact the GP outputs zero with high confidence far from the data, as do all methods.

Table 1 (appendix) gives results of our method on ten standard benchmarking datasets (Hernández-Lobato and Adams, 2015). Our method outperforms Deep Ensembles (Lakshminarayanan et al., 2017) on datasets where the primary source of uncertainty was epistemic. Code of our implementation is available at `https://github.com/TeaPearce`.

5   Conclusion

This paper considered a method to produce Bayesian behaviour in NN ensembles by leveraging randomised MAP sampling. It departs only slightly from the usual handling of NNs, with parameters regularised around values drawn from a prior distribution. We showed that for NNs of sufficient width, each produces a sample from the posterior predictive distribution. Qualitative and benchmarking experiments were encouraging. Our ongoing work considers extending the presented theory to classification tasks as well as other architectures such as convolutional NNs.

References

Bardsley, J. M. (2012). MCMC-based image reconstruction with uncertainty quantification. *SIAM Journal on Scientific Computing*, 34(3):1316–1332.

Bardsley, J. M., Solonen, A., Haario, H., and Laine, M. (2014). Randomize-Then-Optimize: A Method for Sampling from Posterior Distributions in Nonlinear Inverse Problems. *SIAM Journal on Scientific Computing*, 36(4).

Cheng, X., Khomtchouk, B., Matloff, N., and Mohanty, P. (2018). Polynomial Regression As an Alternative to Neural Nets. pages 1–28.

Gal, Y. (2016). *Uncertainty in Deep Learning*. PhD thesis.

Gal, Y. and Ghahramani, Z. (2015). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning*.

Hernández-Lobato, J. M. and Adams, R. P. (2015). Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *Proceedings of the 32nd International Conference on Machine Learning*.

Heskes, T. (1996). Practical confidence and prediction intervals. In *Advances in Neural Information Processing Systems 9*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *31st Conference on Neural Information Processing Systems*.

Lu, X. and Van Roy, B. (2017). Ensemble Sampling. In *31st Conference on Neural Information Processing Systems*.

MacKay, D. J. C. (1992). A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3):448–472.

Neal, R. M. (1997). *Bayesian Learning for Neural Networks*. PhD thesis.

Osband, I., Aslanides, J., and Cassirer, A. (2018). Randomized Prior Functions for Deep Reinforcement Learning. In *32nd Conference on Neural Information Processing Systems (NIPS 2018)*.

Osband, I., Russo, D., Wen, Z., and Van Roy, B. (2017). Deep Exploration via Randomized Value Functions.

Tibshirani, R. (1996). A Comparison of Some Error Estimates for Neural Network Models. *Neural Computation*, 8:152–163.

Williams, C. K. I. (1996). Computing with infinite networks. In *Advances in Neural Information Processing Systems 9*.

A Appendix

A.1 Proofs

Theorem A.1. $\Sigma^2_{prior}\Sigma^{-1}_{like}$ *tends to a zero matrix with increasing* H.

Proof. First we consider priors, Σ_{prior}. It is usual to scale prior covariance in BNNs according to $1/H$ (Neal, 1997). This means the term of interest, $\Sigma^2_{prior}\Sigma^{-1}_{like} \propto \frac{1}{H^2}\Sigma^{-1}_{like}$, which clearly decreases with H.

Secondly, increasing H creates more parameters and hence a higher probability of strong correlations amongst them - a phenomenon known as multicollinearity. This has the effect of increasing the magnitude of Σ_{like} (see also Cheng et al. (2018)). Hence Σ^{-1}_{like} decreases.

Both these results suggest, $\lim_{H\to\infty} \Sigma^2_{prior}\Sigma^{-1}_{like} \to 0$. $\qquad\square$

A.2 Benchmark Results

Table 1: Regression benchmark results for a Bayesian ensemble of five NNs

	N	D	$\hat{\sigma}^2_\epsilon$	RMSE			NLL		
				Deep Ens. *State-Of-Art*	Bay. Ens. *Our Method*	GP[1] *Gold Standard*	Deep Ens. *State-Of-Art*	Bay. Ens. *Our Method*	GP[1] *Gold Standard*
High Epistemic Uncertainty									
Energy	768	8	1e-7	2.09 ± 0.29	$\mathbf{0.35 \pm 0.01}$	0.60 ± 0.02	1.38 ± 0.22	$\mathbf{0.96 \pm 0.13}$	0.86 ± 0.02
Naval	11,934	16	1e-7	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.00	-5.63 ± 0.05	$\mathbf{-7.17 \pm 0.03}$	-10.05 ± 0.02
Yacht	308	6	1e-7	1.58 ± 0.48	$\mathbf{0.57 \pm 0.05}$	0.60 ± 0.08	1.18 ± 0.21	$\mathbf{0.37 \pm 0.08}$	0.49 ± 0.07
Medium Epistemic & Aleatoric Uncertainty									
Kin8nm	8,192	8	0.02	0.09 ± 0.00	$\mathbf{0.07 \pm 0.00}$	0.07 ± 0.00	-1.20 ± 0.02	$\mathbf{-1.09 \pm 0.01}$	-1.22 ± 0.01
Power	9,568	4	0.05	4.11 ± 0.17	$\mathbf{4.07 \pm 0.04}$	3.97 ± 0.04	$\mathbf{2.79 \pm 0.04}$	2.83 ± 0.01	2.80 ± 0.01
Concrete	1,030	8	0.05	6.03 ± 0.58	$\mathbf{4.87 \pm 0.11}$	4.88 ± 0.13	3.06 ± 0.18	$\mathbf{2.97 \pm 0.02}$	2.96 ± 0.02
Boston	506	13	0.08	3.28 ± 1.00	$\mathbf{3.09 \pm 0.17}$	2.86 ± 0.16	$\mathbf{2.41 \pm 0.25}$	2.52 ± 0.05	2.45 ± 0.05
High Aleatoric Uncertainty									
Protein	45,730	9	0.5	4.71 ± 0.06	$\mathbf{4.36 \pm 0.02}$	$*4.34 \pm 0.02$	$\mathbf{2.83 \pm 0.02}$	2.89 ± 0.01	$*2.88 \pm 0.00$
Wine	1,599	11	0.5	$\mathbf{0.64 \pm 0.04}$	$\mathbf{0.63 \pm 0.01}$	0.61 ± 0.01	$\mathbf{0.94 \pm 0.12}$	$\mathbf{0.95 \pm 0.01}$	0.92 ± 0.01
Song Year	515,345	90	0.7	$8.89 \pm \text{NA}$	$\mathbf{8.82 \pm \text{NA}}$	$**9.01 \pm \text{NA}$	$\mathbf{3.35 \pm \text{NA}}$	$3.60 \pm \text{NA}$	$**3.62 \pm \text{NA}$

[1] For reference only (not a scalable method). * Trained on $10,000$ rows of data. ** Trained on $20,000$ rows of data, tested on $5,000$ data points.