Implicit Weight Uncertainty in Neural Networks

Nick Pawlowski
Imperial College London
n.pawlowski16@imperial.ac.uk

Martin Rajchl
Imperial College London
m.rajchl@imperial.ac.uk

Ben Glocker
Imperial College London
b.glocker@imperial.ac.uk

Abstract

We interpret *HyperNetworks* [6] within the framework of variational inference within implicit distributions[16, 8, 13]. Our method, Bayes by Hypernet, is able to model a richer variational distribution than previous methods. Experiments show that it achieves comparable predictive performance on the MNIST classification task while providing higher predictive uncertainties compared to MC-Dropout [3] and regular maximum likelihood training.

1 Introduction

Neural networks achieve state of the art results on a wide variety of tasks [11]. However, networks that are trained with maximum likelihood or MAP-based techniques often output overconfident predictions [5] and do not correctly capture the output uncertainty. This makes them unsuitable for real life decision making, *e.g.* self-driving cars or disease detection employing deep learning methods.

Previous works proposed diverse approximate Bayesian inference methods to alleviate those concerns by obtaining posterior distributions rather than point estimates. We follow works on variational inference for Bayesian neural networks like [1, 12, 16, 3, 10]. We introduce Bayes by Hypernet, a method for approximate Bayesian inference using implicit variational distributions. It combines the work of [16, 8, 13] on implicit variational inference with *HyperNetworks* [6] to implicitly approximate the weight distribution.

2 Variational Inference and Implicit Distributions

Variational inference for Bayesian neural networks aims to approximate the posterior distribution $p(\mathbf{w} \mid \mathcal{D})$, where $\mathbf{w}$ are the weights of the neural network and $\mathcal{D}$ is the given dataset. Given this distribution we can estimate the posterior prediction $\hat{y}$ of a new data point $\hat{x}$ as $p(\hat{y} \mid \hat{x}, \mathcal{D}) = \mathbb{E}_{w \sim p(\mathbf{w}|\mathcal{D})}[p(\hat{y} \mid \hat{x}, \mathbf{w})]$. Because exact Bayesian inference is usually intractable in neural networks we find a variational approximation $q(\mathbf{w} \mid \theta)$ with parameters θ that minimises the Kullback-Leibler (KL) divergence, so that

$$\begin{aligned} \theta^* &= \arg\min_\theta KL\left(q(\mathbf{w} \mid \theta) \| p(\mathbf{w} \mid \mathcal{D})\right) && (1) \\ &= \arg\min_\theta KL\left(q(\mathbf{w} \mid \theta) \| p(\mathbf{w})\right) - \mathbb{E}_{w \sim q(\mathbf{w}|\theta)}\left[\log p(\mathcal{D} \mid \mathbf{w})\right] && (2) \\ &= \arg\min_\theta \mathbb{E}_{w \sim q(\mathbf{w}|\theta)}\left[\log \frac{q(\mathbf{w} \mid \theta)}{p(\mathbf{w})} - \log p(\mathcal{D} \mid \mathbf{w})\right]. && (3) \end{aligned}$$

Second workshop on Bayesian Deep Learning (NIPS 2017), Long Beach, CA, USA.

We propose to use an implicit distribution as variational approximation. An implicit distribution might have an intractable density functions butt is possible to sample from them and calculate expectations of them as well as calculating gradients with respect to its parameters. [13, 8, 16] extended the framework of variational inference to implicit distributions. We follow the notion of prior-contrastive adversarial variational inference [8] where the KL term contrasts approximate posterior with the prior. We estimate the log density ratio in Equation 3 using logistic regression. Following an approach similar to generative adversarial networks [4] we can use a discriminator D for density ratio estimation while a neural network acts as generator $w = G(z \mid \theta)$ and models the variational distribution $q\left(\mathbf{w} \mid \theta\right)$. Here, z is an auxiliary variable. This enables a two-step update procedure[1] with

$$\mathcal{L}\left(D \mid G\right) = \mathbb{E}_{w \sim G(z|\theta)} \log D(\mathbf{w}) - \mathbb{E}_{w \sim p(\mathbf{w})} \log\left(1 - D(\mathbf{w})\right) \tag{4}$$

$$\mathcal{L}\left(G \mid D\right) = \mathbb{E}_z \log \frac{D\left(G(z \mid \theta)\right)}{1 - D\left(G(z \mid \theta)\right)} - \mathbb{E}_z \log p\left(\mathcal{D} \mid G(z \mid \theta)\right) \tag{5}$$

where Equation 4 and Equation 5 are being used to update the discriminator and the generator.

The idea of using a generator to predict weights resembles the idea of *HyperNetworks* [6], which uses a neural network to deterministically predict the weights of a bigger network. It lends our method its name, Bayes by Hypernet. Here, we only employ static Hypernets that lead to a global variable model and propose to parametrise the auxiliary variable z similar to [2] and conditional GANs [14] by combining conditioning that encodes the position of the generated weight with an auxiliary noise vector. In contrast to [10], our method works with arbitrary *HyperNetworks* and relies on density ratio estimation for variational inference with implicit distributions.

3 Experiments

For all experiments, we employ a two layer fully-connected neural network with 64 and 256 hidden units as *HyperNetwork*. The output size is dependent on the maximum size of the generated weights. The output vector is sliced if a smaller weights vector is needed. The density ratio is estimated via a discriminator with 2 hidden layers with 20 units each. A diagonal scale-mixture prior similar to [1] is employed and the weights treated independently of each other. All experiments use *Adam* [9] for optimisation. The source code to reproduce the proposed method is publicly available at `https://github.com/pawni/bayesbyhypernet`.

Regression on a Toy Example: We follow the setup from [7] to build a toy dataset for a simple regression task. A fully-connected network with 100 hidden units and ReLU activation is used and apply a dropout with $\pi = 0.5$ to the hidden layer. The proposed method models the posterior of the first layer weights and bias. We compare the proposed method with MC-Dropout [3], Bayes by Backprop [1], and maximum likelihood training. Figure 1 shows a comparison of the results. The error bands correspond to 1σ, 2σ and 3σ. We argue that our method provides the best fit as it also fits the saddle point that is ignored by all other methods.

MNIST Digit Classification: We perform digit classification on MNIST with a fully-connected network with two hidden layers with each 800 units. We employ the weight norm [15] parametrization as used by [10] to implicitly learn the scaling factors of every layer. The rest of the weights are learned using maximum likelihood training. We compare our method to MC-Dropout with $\pi = 0.7$ applied to all hidden layers and maximum likelihood training. After 50 epochs of training, all networks exhibit comparable test accuracy performance of 98.47%, 98.56% and 98.51%, for the proposed method, maximum likelihood and MC-Dropout, respectively.

We follow [3, 12] and test the predictive uncertainty by evaluating the output probabilities of a rotated image of a digit 3. We plot the output probabilities of maximum likelihood training in Figure 2a, ours, Bayes by Hypernet, in Figure 2b, and MC-Dropout in Figure 2c. Maximum likelihood exhibits poorly calibrated predictive uncertainties, while our method and MC-Dropout show uncertain predictions for some rotation angles. Whereas MC-Dropout stays certain about the correct label for small rotations, our method exhibits an overall higher predictive uncertainty which intuitively seems to make sense.

[1] This is closely related to `http://www.inference.vc/variational-inference-with-implicit-probabilistic-models-part-1-2/#variationalinferenceanddensityratios`

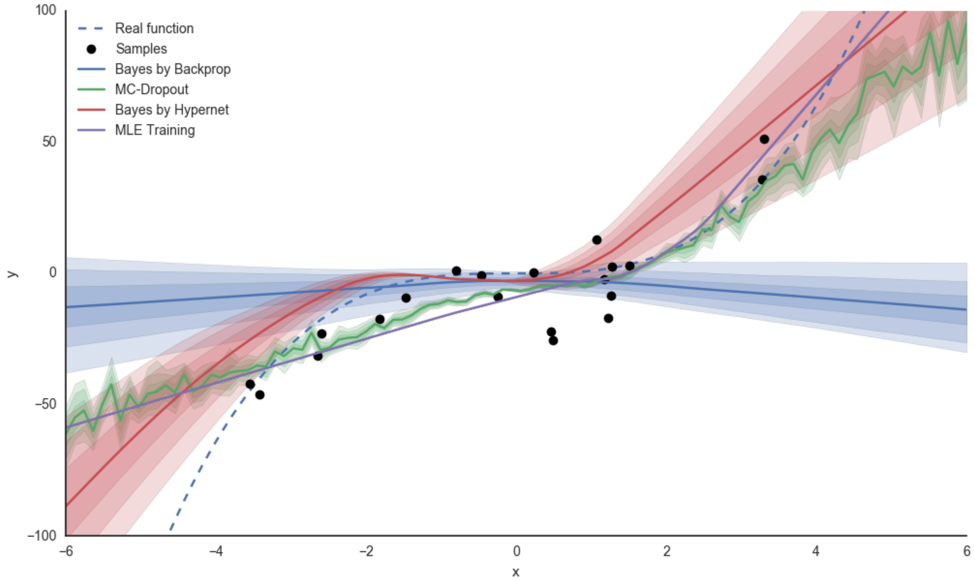
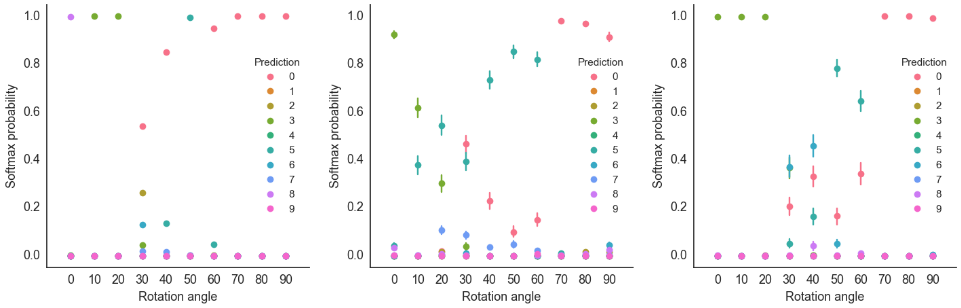


Figure 1: Toy example inspired by [7]. The proposed method exhibits the highest uncertainty and is the only method to correctly model the saddle point. MC-Dropout strongly follows the maximum likelihood result, while Bayes by Backprop fails at this task.



(a) Maximum Likelihood (b) Bayes by Hypernet (c) Dropout

Figure 2: Predictive distribution of a rotated 3 from the MNIST dataset. The points correspond to the mean class probability and the error bars show the standard deviation of the prediction. Figure inspired by [3, 12].

4 Discussion & Conclusions

We propose Bayes by Hypernet, interpreting *HyperNetworks* [6] as an implicit distribution which we use as approximate distribution within variational inference. We show that on a toy task this approximation yields better uncertainties than Dropout. On MNIST it yields comparable predictive performance to other methods and similar uncertainties as MC-Dropout.

However, our experiments seem to not fully reproduce results of previous works that attributed better performance on the toy dataset to MC-Dropout [12]. Further, the predictive uncertainty on the rotated digit task seems to be inferior to [12] that uses CNNs rather than fully-connected networks. In contrast to the work by [10] that holds a similar name as our work, we believe that the *Bayesian GANs* presented in [16] are the closest work to ours.

Future work should explore the importance of the dimensionality of the auxiliary variables as well as the complexity of the used HyperNetwork and density estimator. Further, this method could be extended to dynamic *HyperNetworks*, which would lead to implicit distributions with latent rather than global variables.

Acknowledgements

The authors would like to thank Andrew Brock for useful discussions and Miguel Jaques for comments on an earlier draft of this work. NP is supported by Microsoft Research PhD Scholarship and the EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (HiPEDS, Grant Reference EP/L016796/1). MR is supported by an Imperial College Research Fellowship. We gratefully acknowledge the support of NVIDIA with the donation of one Titan X GPU for our research.

References

[1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[2] Andrew Brock, Theodore Lim, JM Ritchie, and Nick Weston. Smash: One-shot model architecture search through hypernetworks. *arXiv preprint arXiv:1708.05344*, 2017.

[3] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

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

[6] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[7] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

[8] Ferenc Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

[9] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[12] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. *arXiv preprint arXiv:1703.01961*, 2017.

[13] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*, 2017.

[14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[15] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.

[16] Dustin Tran, Rajesh Ranganath, and David M Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 2017.