Efficient variational Bayesian neural network ensembles for outlier detection

Nick Pawlowski*[1], Miguel Jaques*[2], Ben Glocker[1]

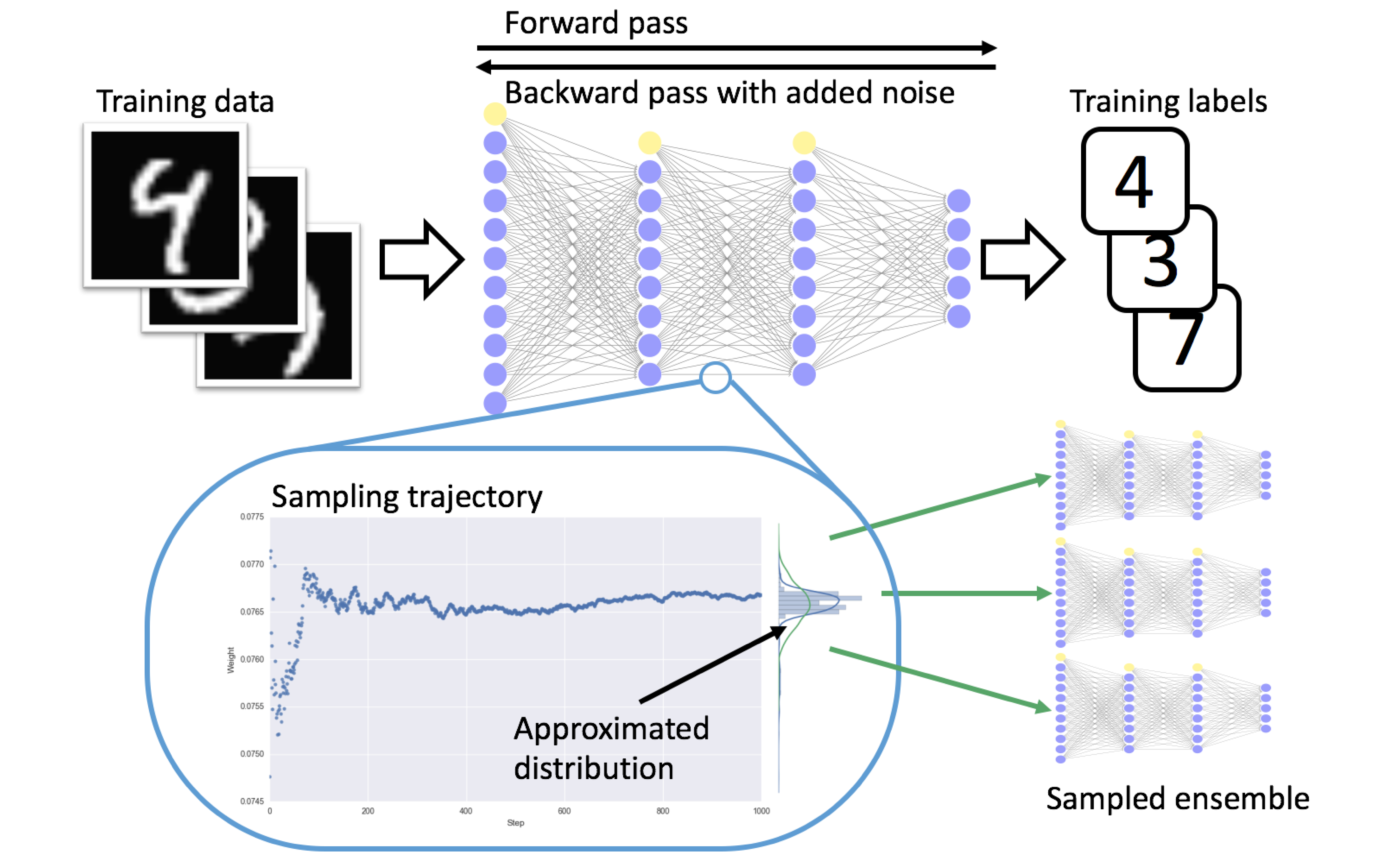
(1) Imperial College, (2) No affiliation

BioMedIA Imperial College London

1: Introduction

Various recent works proposed efficient approaches to Bayesian neural networks like MC-Dropout [1]. We propose to approximate the posterior distribution of weights of a neural network by sampling and approximating the sampled distribution with diagonal Gaussians. Those can be fitted with incremental updates, reducing the memory overhead. We use SGLD [3] as sampling method.

2: Overview



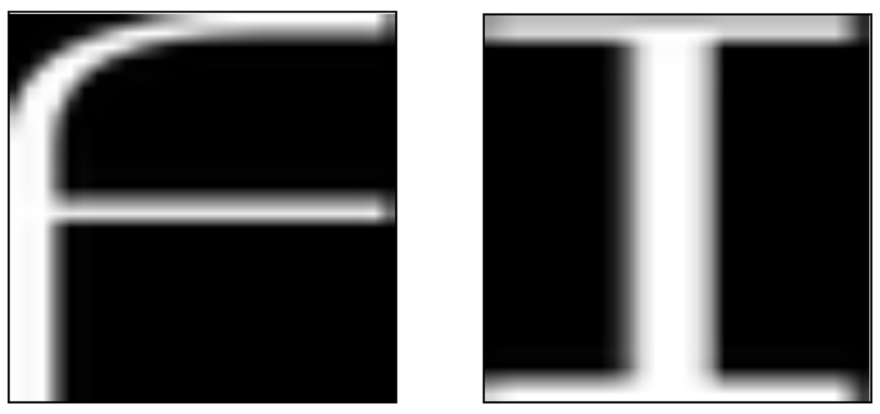
3: Ensemble Classification

We tested our method on the MNIST dataset against regular and Dropout ensembles. Tests with snapshot ensembles or variational inference performed poorly. All networks have the same architectures and are trained for 1000 steps using the same learning rate schedule used for SGLD. The following table shows the classification accuracy for the different methods for different ensemble sizes.

Size	1	5	10
Regular	80.7 ± 1.1	86.3 ± 0.6	87.0 ± 0.6
Dropout	61.4 ± 3.2	71.4 ± 2.9	72.9 ± 2.7
Ours	76.0 ± 0.6	76.1 ± 0.7	76.1 ± 0.7

4: Outlier Dataset

For outlier detection we compare the MNIST test set against the notMNIST test set. Examples of notMNIST are:



5: Outlier Detection

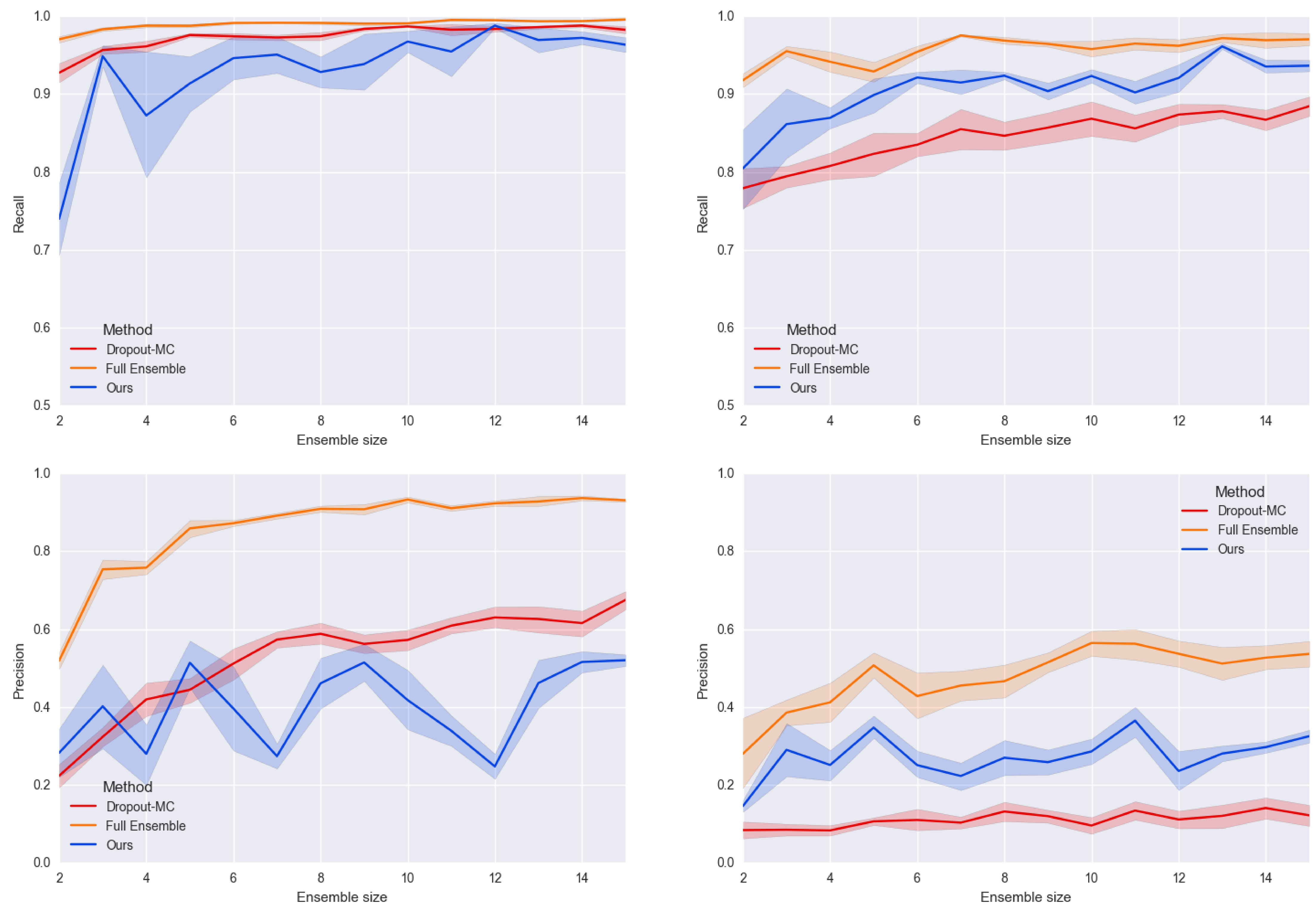
We use the disagreement of the different ensemble components $p_i(y \mid x)$ and the ensemble prediction $\bar{p}(y \mid x) = \frac{1}{N}\sum_i p_i(y \mid x)$ as defined by [2]:

$$d_x = \frac{1}{N}\sum_i \mathrm{KL}(p_i(y \mid x) \mid \bar{p}(y \mid x))$$

Given the average $\mu(d_{train})$ and standard deviation $\sigma(d_{train})$ of the disagreement on the training set, we use this to define the probability of x being an outlier as:

$$p(\text{outlier} \mid x) = d_x \geq \mu(d_{train}) + 3\sigma(d_{train})$$

There we chose a margin of $3\sigma(d_{train})$ to reduce the false positive rate. This yield the following recall and precision curves for outlier detection using different methods, learning rate schedules (left: SGLD, right: Adam) and different ensemble sizes:



6: Conclusion

We proposed a new, efficient method for training Bayesian neural networks. We show that our method outperforms Dropout on the MNIST classification task, when the SGLD learning rate schedule is used. However, we slightly underperform on the outlier detection task. Interestingly, we still obtain good uncertainty estimates when approximating SGLD with a noisy version of Adam. We then outperform Dropout which was trained with regular Adam.

Code and paper

The code is available at `https://github.com/pawni/sgld_online_approximation`.
The paper can be found at `https://arxiv.org/abs/1703.06749`.

Acknowledgements

NP is supported by Microsoft Research through its PhD Scholarship Programme and the EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (HiPEDS, Grant Reference EP/L016796/1).

References

[1] Y. Gal *et al.*, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," 2015.

[2] B. Lakshminarayanan *et al.*, "Simple and scalable predictive uncertainty estimation using deep ensembles," *ArXiv preprint arXiv:1612.01474*, 2016.

[3] M. Welling *et al.*, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th International Conference on Machine Learning*, 2011.