Weight Uncertainty in Neural Networks

Gergely Flamich, Tatiana Matejovičová, Luca Biggio

March 15, 2019

UNIVERSITY OF CAMBRIDGE

Introduction

Problem

Traditional feedforward neural networks do not provide a measure of the **uncertainty** in their predictions resulting in being prone to **overfitting**.

Proposed Solution

Bayes by Backprop (BBB) [1] introduces uncertainty in the weights (Figure 1) and predictions and thus improves generalisation. This is achieved by sampling from the **weight posterior distribution**.

Imple-mentation

Exact Bayesian inference is intractable because of the large number of network parameters. A **variational approximation** based on free energy minimization and the reparametrisation trick estimates the parameters of the weights posterior.

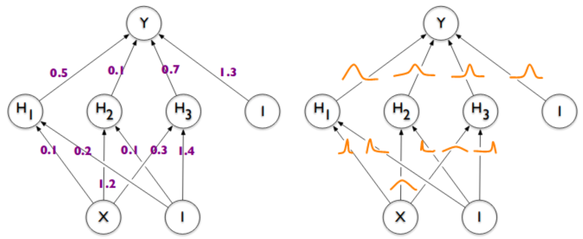

Figure 1: Each weight is represented by a single value on the left (standard neural network) and by a distribution on the right (BBB). Image taken from [1].

- Prediction is made as an expected value over the posterior distribution of the weights:

$$P(\hat{\mathbf{y}} \mid \hat{\mathbf{x}}) = \mathbb{E}_{P(\mathbf{w}|\mathcal{D})}[P(\hat{\mathbf{y}} \mid \hat{\mathbf{x}}, \mathbf{w})] \tag{1}$$

Bayes By Backprop: A Variational Approach

- The parameters, θ^*, defining the posterior approximation are obtained by minimizing the Expectation Lower Bound (ELBO), defined as

$$\mathcal{F}(\mathcal{D}, \theta) = \underbrace{\mathcal{KL}\left[q(\mathbf{w} \mid \theta) \,||\, P(\mathbf{w})\right]}_{\text{Compression cost}} - \underbrace{\mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D} \mid \mathbf{w})]}_{\text{Likelihood cost}}. \tag{2}$$

- Calculating this quantity is analytically intractable, hence we resort to Monte Carlo (MC) approximations:

$$\begin{aligned} \mathcal{F}(\mathcal{D}, \theta) &\approx \frac{1}{n}\sum_{i=1}^{n} \log q(\mathbf{w}^{(i)} \mid \theta) - \log P(\mathbf{w}^{(i)}) - \log P(\mathcal{D} \mid \mathbf{w}^{(i)}) \\ &= \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{w}^{(i)}, \theta) \quad \text{where } \mathbf{w}^{(i)} \sim q(\mathbf{w} \mid \theta) \end{aligned} \tag{3}$$

- To calculate $\nabla_\theta \mathcal{F}(\mathcal{D}, \theta)$, which is the gradient of an expectation over $q(\mathbf{w} \mid \theta)$, we utilise the reparametrisation trick [2]:

$$\mathbf{w} = \mu + \sigma \circ \epsilon \quad \text{where } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{4}$$

- Sampling $\mathbf{w}^{(i)}$ according to Eq (4) allows us to obtain unbiased MC estimates for the gradient of $\mathcal{F}$ using the samples from Eq (3):

$$\nabla_\theta \mathcal{F}(\mathcal{D}, \theta) = \frac{\partial}{\partial \theta}\mathbb{E}_{q(\mathbf{w}|\theta)}\left[f(\mathbf{w}, \theta)\right] \approx \sum_{i}^{n} \frac{\partial f(\mathbf{w}^{(i)}, \theta)}{\partial \mathbf{w}^{(i)}}\frac{\partial \mathbf{w}^{(i)}}{\partial \theta} + \frac{\partial f(\mathbf{w}^{(i)}, \theta)}{\partial \theta}.$$

Classification

Model	#Units	Test Error
BBB	400	1.79%
Dropout	400	1.72%

Model	#Units	Test Error
BBB	800	1.92%
Dropout	800	1.45%

Table 1: BBB and Dropout are applied to a feedforward neural network with two 400/800 unit layers (MNIST dataset).

- An **advantage** of BBB is that it returns the posterior over the weights.
- A **drawback** of BBB is that it has twice as many parameters to train.

Weight Distribution

- The Dropout and BBB weight distributions have a larger variance than a standard neural network, possibly leading to regularisation.
- **Signal-to-noise ratio** (SNR) i.e. $\frac{|\mu|}{\sigma}$ is computed to prune weights in BBB.

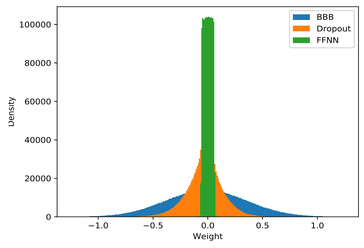

Figure 2: Comparison between the weight distributions of BBB, Dropout and a standard feed-forward neural network.

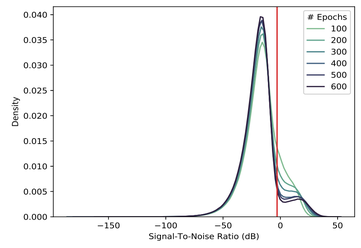

Figure 3: Evolution of weight distribution as a function of the training epochs and 90% pruning threshold.

- Up to 90% of the weights can be removed in BBB without significantly affecting the accuracy resulting in a faster and smaller network.
- Removing the same number of weights based on their absolute value in the Dropout NN reduces accuracy considerably.

Pruning	#Parameters	BBB
0%	2.6m	1.92%
50%	1.3k	1.86%
75%	650k	1.85%
90%	260k	1.82%
95%	130k	2.16%

Pruning	#Parameters	Dropout
0%	1.3m	1.45%
50%	650k	1.45%
75%	325k	1.75%
90%	130k	17.01%
95%	65k	41.48%

Table 2: Weights are pruned based on SNR (left) and absolute value (right) to obtain a compact network.

References

[1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. *arXiv e-prints*, page arXiv:1505.05424, May 2015.

[2] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv e-prints*, page arXiv:1312.6114, Dec 2013.

[3] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

Regression

- We generated the homoscedastic regression task described in [1] (Figure 4).
- While the regular NN is confident in its fit far away from the region where we have data, BBB naturally becomes more uncertain as we move farther into unseen regions.

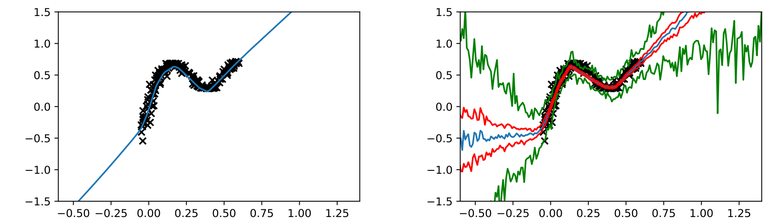

Figure 4: Comparison of fit between a standard feed-forward neural network (left) with BBB (right) on the regression task given above. For BBB the result of 30 samples is given. The curves show the interquartile ranges (blue is the median).

Reinforcement Learning

- We compared the performance of BBB to other NN-based agents on the UCI Mushroom Dataset [3], cast as a RL task. We measured the agents' cumulative regret relative to an oracle (Figure 5).
- We see that the ϵ-greedy agents over-explore. The greedy agent does better, but BBB's regret flattens out quicker and to a larger extent than the others'.

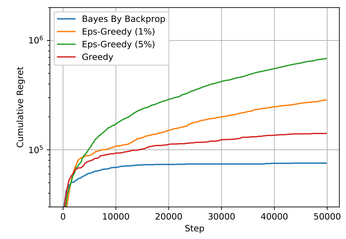

Figure 5: Cumulative regrets of different agents.

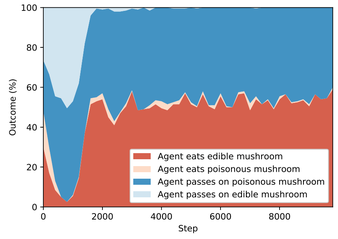

Figure 6: Evolution of the policy of the BBB agent over the first 10000 steps.

Conclusion and Future Investigation

- Neural networks are regularised in a principled fashion in BBB by using the Kullback-Leibler divergence in the cost function.
- BBB allows cheap averaging of NNs, which in the limit is equivalent to an ensemble of uncountably many models.
- The uncertainty introduced by sampling from the variational posterior naturally leads to exploration in RL tasks, which outperforms the simpler ϵ-greedy methods.
- The hyperparameters ought to be systematically tuned to achieve better performance.
- An interesting problem is to extend BBB to other types of neural networks, especially CNNs and RNNs.