Uncertainty Estimation: Not So Bayes-ic

Mizu Nishikawa-Toomey [1,2] Miguel Saavedra [1,2] Alihusein Kuwajerwala [1,2]

[1]Mila - Quebec AI Institute. Montreal, Canada [2]Université de Montréal, Montréal, Canada

Introduction

In this project we provide our own implementation of an approach towards modeling uncertainty in neural networks as proposed in *Weight Uncertainty in Neural Networks* [1].

Plain feedforward neural networks are susceptible to overfitting and are **unable to successfully model uncertainty in their predictions**; leading to overconfident predictions even on out-of-distribution data [2]. The approach in [1] aims to address both issues by proposing an algorithm for learning a *posterior distribution* on the weights of a neural network as opposed to point values (see Fig. 1).

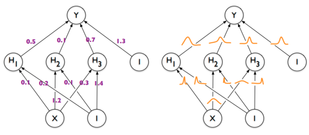

Figure 1. Left: each weight has a fixed value, as provided by classical backpropagation. Right: each weight is assigned a distribution, as provided by Bayes by Backprop. (Original image and caption from [1])

We **re-implement the method in Python using the PyTorch** library and conduct our own experiments in addition to those provided in the original paper, obtaining new results associated with this approach.

For regression, our experiments reproduce the original results, but indicate that they are sensitive to changes in hyper-parameters. For classification, our results confirm the performance gains of the approach over deterministic models, though diverge slightly for other experiments.

Methodology

Our objective function relies on minimising the KL divergence between the approximate variational posterior $q(\mathbf{w}|\theta)$ and the true posterior $P(\mathbf{w}|\mathcal{D})$.

$$\begin{aligned}\theta^* &= \arg\min_\theta \mathrm{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w}|\mathcal{D})] = \arg\min_\theta \mathrm{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})] \\ &= \arg\min_\theta \mathcal{F}(\mathcal{D},\theta)\end{aligned}$$

The loss function is approximated by averaging samples of the weights from our variational posterior $q(\mathbf{w}|\theta)$. This loss is derived using the re-parameterization trick [3].

$$\mathcal{F}(\mathcal{D},\theta) \approx \frac{1}{n}\sum_{i=1}^{n} \log q(\mathbf{w}_i|\theta) - \log P(\mathcal{D}|\mathbf{w}_i)P(\mathbf{w}_i) \tag{1}$$

The posterior predictive mean is computed as the Monte Carlo estimate of the first statistical moment of the likelihood (2) and the **uncertainty** as the second statistical moment (3). The entropy is used as a measure of uncertainty for classification tasks.

$$\mathbb{E}_{P(\hat{\mathbf{y}}|\hat{\mathbf{x}})}[\hat{\mathbf{y}}] \approx \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{P(\hat{\mathbf{y}}|\hat{\mathbf{x}},\mathbf{w}_i)}[\hat{\mathbf{y}}] \tag{2}$$

$$Var_{P(\hat{\mathbf{y}}|\hat{\mathbf{x}})}[\hat{\mathbf{y}}] \approx \frac{1}{n}\sum_{i=1}^{n} Var_{P(\hat{\mathbf{y}}|\hat{\mathbf{x}},\mathbf{w}_i)}[\hat{\mathbf{y}}] \tag{3}$$

Regression Experiments

We experimented using different combinations of priors and assessed the effect of those while capturing the target function using a toy regression experiment with Gaussian noise $\sigma^2 = 0.02$.

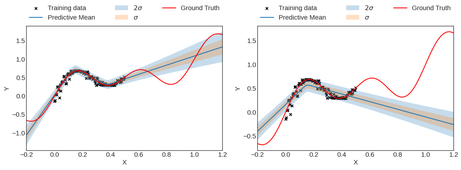

Figure 2. Regression of noisy data with two different scaled mixture priors: (left) Scaled Mixture prior with probability mass concentrated around zero and (right) Scaled mixture prior with spread probability mass.

The method is capable of capturing the complexity of the data and estimate its uncertainty using a mixture prior with most of the probability mass centered around zero whereas a spread prior over-regularizes the model.

Bayes by backprop performs well compared to other baselines on the toy regression function, but struggles to achieve the same result on a real-world dataset because its sensitiveness to changes in hyper-parameters.

Table 1. Comparison against baselines.

Model	MSE ↓	$\bar{\sigma}$
Linear	0.1	-
Dropout	0.23	-
MC Dropout	0.28	0.05
Ensemble	0.09	**0.01**
BBB (Mixture Prior)	**0.05**	0.06

Table 2. Results of BBB on NASA dataset using different likelihood variances

σ_l	MSE ↓	σ	2σ
0.1	**4.14**	6%	11%
0.5	4.54	28%	50%
1.5	7.0	59%	87%
3.0	10.42	77%	97%
5.0	11.26	**91%**	**99%**

Pros 😎 and Cons 🤔 of Bayes by Backprop

Pros 😎 :

- Allows for *uncertainty estimation* in models by introducing stochasticity in the model parameters.
- Inherently self-regularizing via its formulation.
- Significantly computationally cheaper than ensembles with similar performance.

Cons 🤔 :

- Highly sensitive to changes in hyperparameters, especially the choice of variance in the likelihood term, or which prior to use.
- Quantification of uncertainty of prediction not clear for classification. Ensemble methods show higher entropy.

MNIST Classification Experiments

The MNIST experiments consist of two parts. The performance of the Bayes by backprop model is compared to baseline. Then the weights of the Bayes by backprop models are pruned depending on their signal to noise ratio, and test accuracy with the pruned models are examined.

Significant performance gains are observed over baseline models with the 'clean' MNIST dataset, performance across all models is similar on dirty-MNIST. We expected to see some performance gains with Bayes by backprop but since all models reach the same performance, we suspect this is just aleatoric uncertainty of the data which caps the performance.

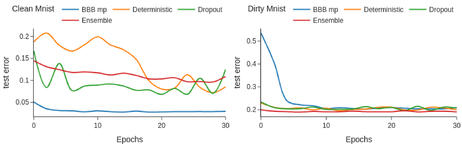

Figure 3. Test error comparing baselines and the Bayes by Backprop model on the MNIST dataset (left) and the dirty MNIST dataset (right).

The weights of the Bayes by backprop model with the lowest signal to noise ratio were removed from the network. These models are trained on clean MNIST. The results are as follows:

Table 3. Entropy of predictions

Dataset	MNIST	dirty-MNIST
BBB model	10k	42k
Ensemble	12k	88k

Table 4. Removing weights with the smallest signal to noise ratio

% of Weights Pruned	0	**75**	90	98	99.5
Test Error (Mixture Prior)	3.4	**3.5**	4.9	25.5	47.9
Test Error (Gaussian Prior)	3.2	**3.9**	4.8	20.4	40.0

Conclusion

- Our project re-implements the approach proposed in *Bayes by Backprop*, confirming the method's performance gains over competing baselines in certain settings.
- Our additional experiments extend the results derived in the original paper with varied hyperparameters to better understand the behavior of such stochastic models.
- Finally, our experiments also raise new questions about uncertainty quantification, which may be used as avenues of inquiry for further research.

References

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

[2] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *CoRR*, abs/1812.05720, 2018.

[3] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.