Weight Uncertainty in Neural Networks

UNIVERSITY OF CAMBRIDGE

A. Macijauskas M. Sadiq V. S. Bøe

Team 10

Motivation

- Classical feed-forward NNs use **point estimates** for weights.

Overfitting | Overconfident

- Bayesian neural networks (BNNs) tackle this by learning a posterior **distribution over weights**.

Uncertainty estimates | Regularization

Bayes by Backprop (BBB) [1]

- **Variational approximation** to infer the posterior $q(\mathbf{w}|\theta)$:

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

- Approximate $\mathcal{F}$ using **MC estimation:**

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

- Averaging outputs for **multiple weight samples** for inference:

$$P(\hat{y}|\hat{x}) = \mathbb{E}_{P(\mathbf{w}|\mathcal{D})} P(\hat{y}|\hat{x}, \mathbf{w}) \approx \frac{1}{n}\sum_{i=1}^{n} P(\hat{y}|\hat{x}, \mathbf{w}_i), \quad \mathbf{w}_i \sim q(\mathbf{w}_i|\theta)$$

Regression on nonlinear data

- Fit a network to **noisy**, **nonlinear** data.

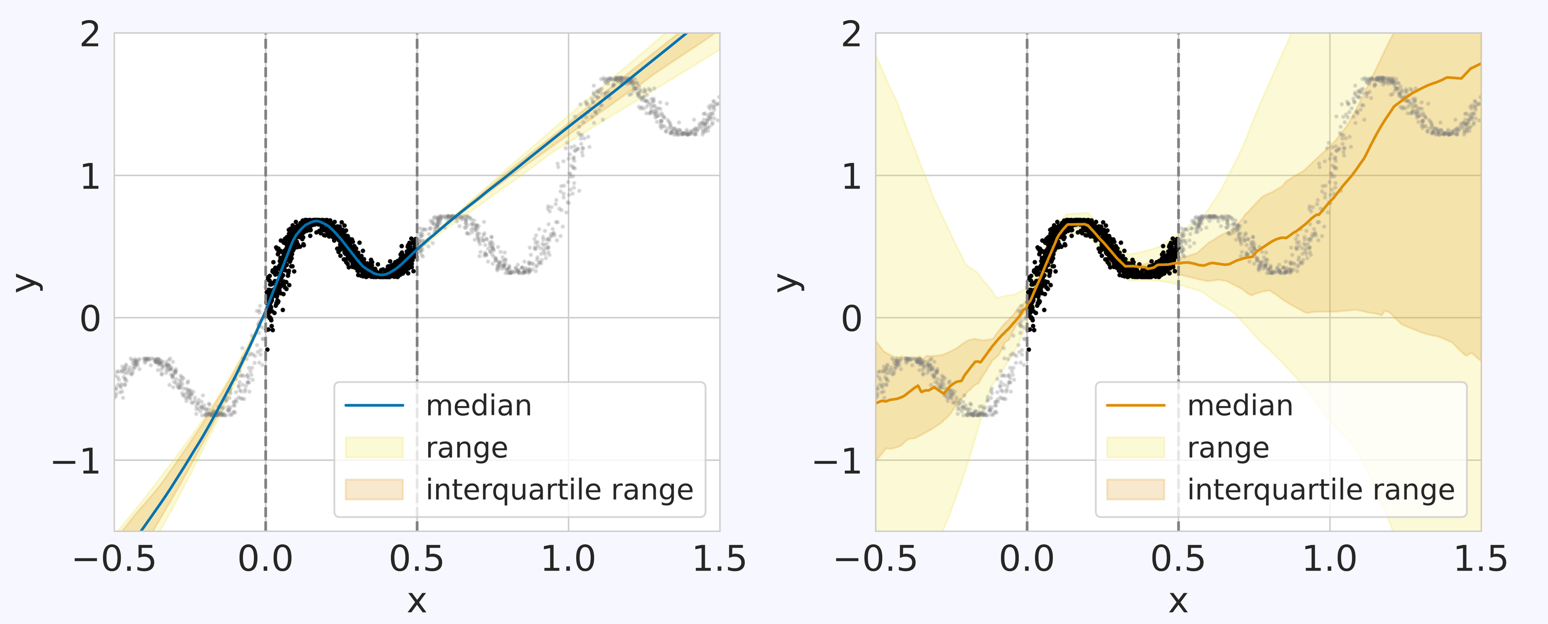

Figure 1. Vanilla NN (left), BBB (right).

- Vanilla NN has almost **no variance**, **over-confident**.
- BBB shows **more uncertainty** further away from training data.

Classification on MNIST

- 50k/10k/10k data split, trained using **SGD optimizer**.

Model	# Units	Test Error (reported)	Test Error (achieved)
SGD	400	1.83%	2.15%
	800	1.84%	1.82%
	1200	1.88%	1.98%
SGD with Dropout	400	1.51%	1.48%
	800	1.33%	1.60%
	1200	1.36%	1.43%
BBB Gaussian	400	1.82%	1.61%
BBB Scale Mixture	800	1.34%	1.37%
BBB Scale Mixture	1200	1.32%	1.48%

Table 1. Comparative study of different models.

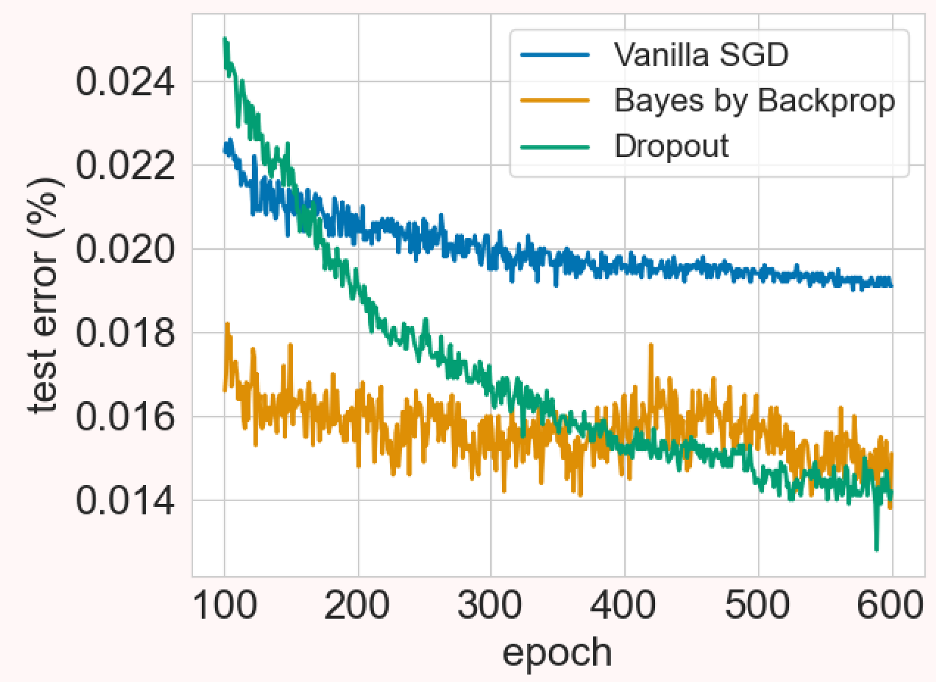

Figure 2. Test error vs epoch as training progresses.

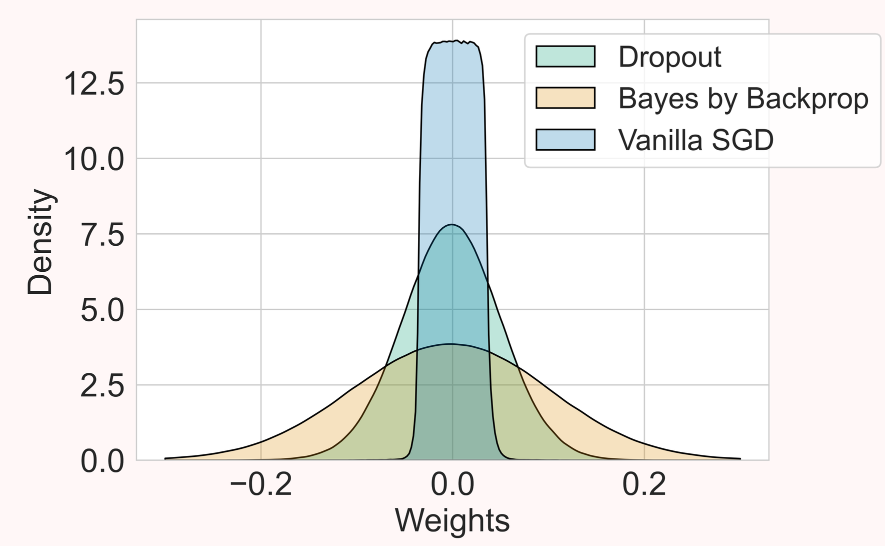

Figure 3. Histograms of weights for different models.

- Natural **weight pruning** using **signal-to-noise ratio**.

Proportion removed	0%	75%	95%	99.5%
# Weights	2.4	600k	120k	12k
Test error	1.58%	1.62%	1.75%	1.84%

Table 2. Classification error after weight pruning.

Reinforcement learning: contextual bandits

- UCI Mushroom dataset as contextual bandit task.
- **Thompson sampling** allows the BBB network to naturally trade-off exploration and exploitation.

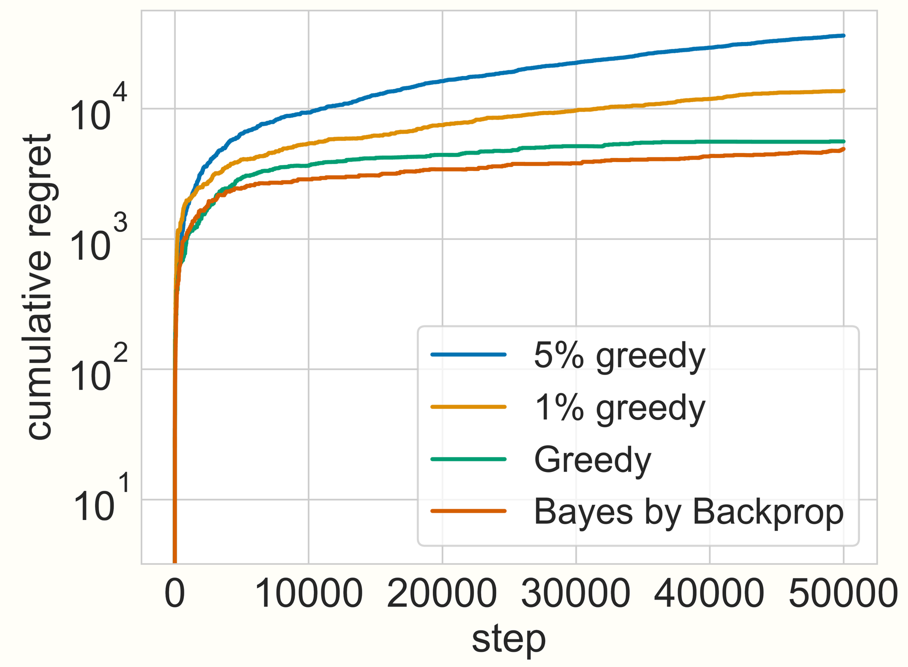

Figure 4. Comparison of cumulative regret values of various agents on the mushroom bandit task.

Takeaways

- Introduced **custom weighting** on the complexity term for BBB regression to work.
- **Weight initialization** matters a lot.
- Our pure greedy agent **alternated actions** from the beginning.

Further work

- Finish training models using **different configurations**.
- Check if BBB can be used to **regularize GATs** [2].
- Evaluate BBB on more **complex datasets**.

Reflection

- Restrain from writing all of the code from scratch on our own.
- Be more confident in our own ideas.

References

[1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[2] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2017.