Weight Uncertainty in Neural Networks

Authors: Blundell, *et al.* 2015

Presenters: Dana Kianfar, Jose Gallego-Posada and Krsto Proroković (Group 5)

Main Points

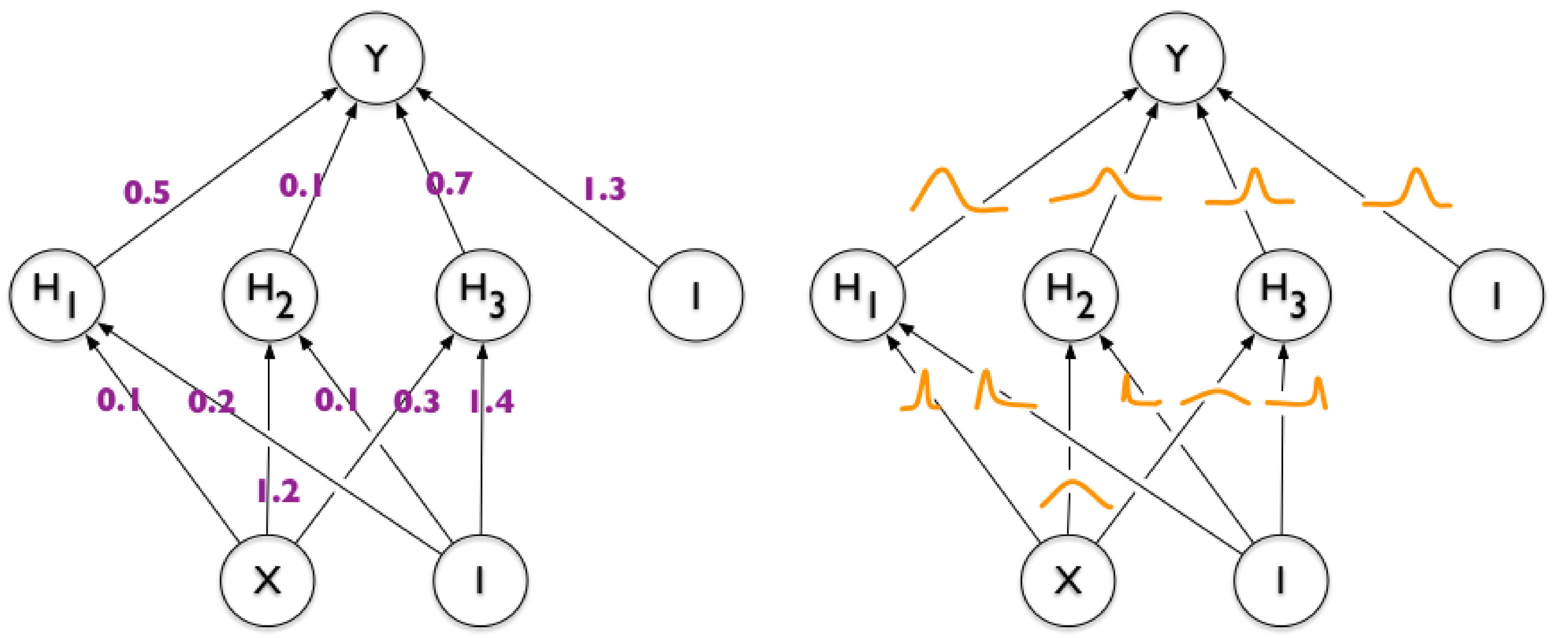
Contribution: An efficient, backpropagation compatible algorithm for bayesian inference on the weights of a neural network

Motivation: Standard neural nets can overfit, and be overconfident in wrong predictions

Bayesian NNs: An *infinite ensemble* of neural networks with good prediction performance and uncertainty estimation

Methodology

All weights in the proposed neural networks are represented by probability distributions over possible values, rather than having a single fixed value as is the norm.



Variational Bayesian inference for neural networks approximates the posterior distribution $P(\mathbf{w}|\mathcal{D})$ with $q(\mathbf{w}|\theta)$ by minimizing the Kullback-Leibler divergence, which is upper bounded by:

$$\mathcal{F}(\mathcal{D},\theta) = \mathbb{KL}\left[q(\mathbf{w}|\theta)||P(\mathbf{w})\right] - \mathbb{E}_{q(\mathbf{w}|\theta)}\left[\log P(\mathcal{D}|\mathbf{w})\right]$$
$$\approx \frac{1}{n}\sum_{i=1}^{n} \log q(\mathbf{w}^{(i)}|\theta) - \log P(\mathbf{w}^{(i)}) - \log P(\mathcal{D}|\mathbf{w}^{(i)}) \,,$$

where $\mathbf{w}^{(i)}$ is drawn from the variational posterior $q(\mathbf{w}^{(i)}|\theta)$.

Training Sample a weight configuration from the current distributions. Then perform forward and back-propagation steps.

procedure $\textsc{Bayes-by-Backprop Step}(\mathcal{D}, \mu, \rho)$

Sample $\epsilon \sim \mathcal{N}(0, \mathbf{I})$

Let $\mathbf{w} = \mu + \log(1+\exp(\rho)) \odot \epsilon$

Let $\theta = (\mu, \rho)$

Let $f(\mathbf{w},\theta) = \log q(\mathbf{w}|\theta) - \log P(\mathbf{w})P(\mathcal{D}|\mathbf{w})$

$\Delta_\mu = \frac{\partial f(\mathbf{w},\theta)}{\partial \mathbf{w}} + \frac{\partial f(\mathbf{w},\theta)}{\partial \mu}$

$\Delta_\rho = \frac{\partial f(\mathbf{w},\theta)}{\partial \mathbf{w}} \frac{\epsilon}{1+\exp(-\rho)} + \frac{\partial f(\mathbf{w},\theta)}{\partial \rho}$

Update parameters

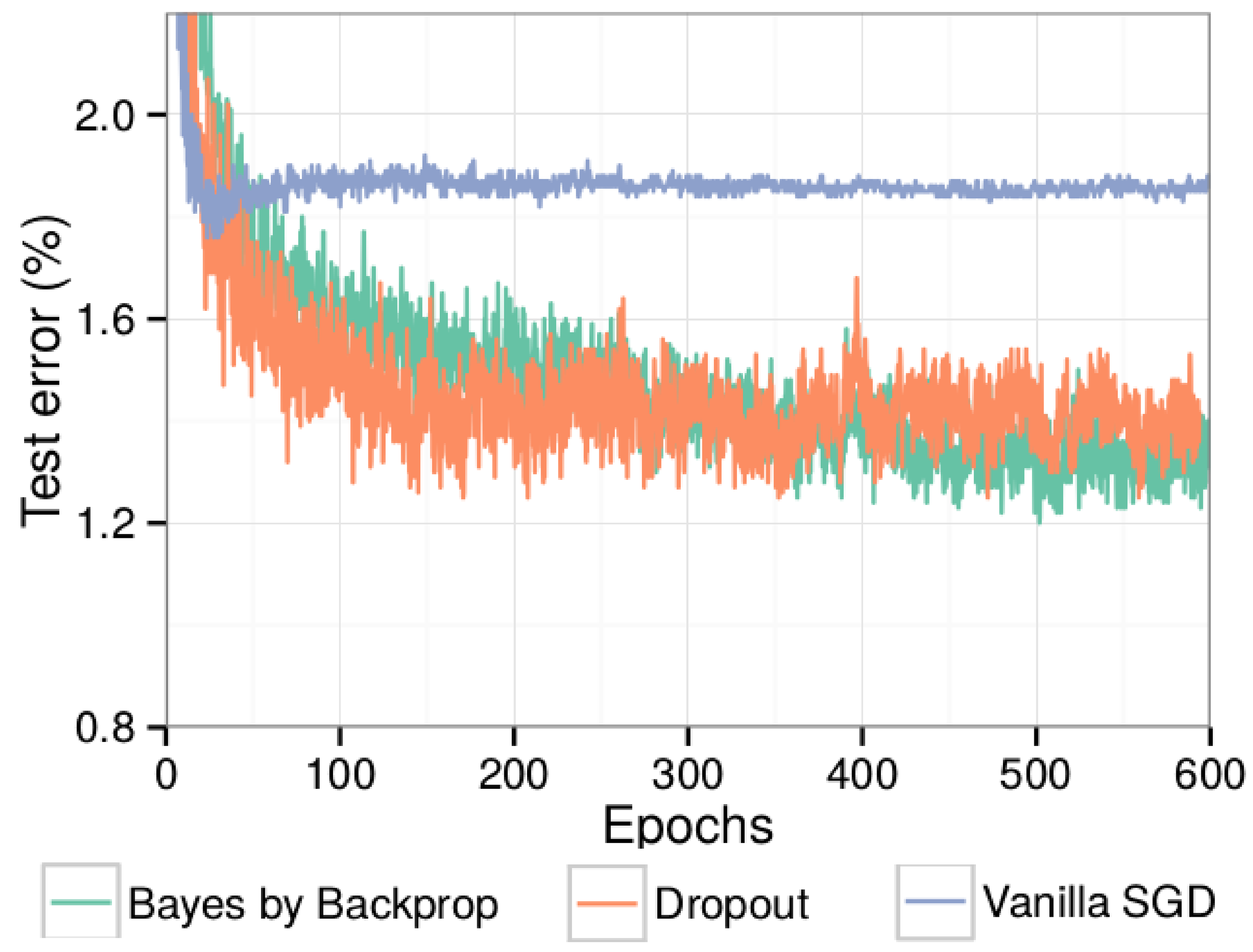
$$\mu \leftarrow \mu - \eta\Delta_\mu$$
$$\rho \leftarrow \rho - \eta\Delta_\rho$$

end procedure

Testing Confidence intervals for the prediction given a input data point can be obtained by doing multiple forwards passes through the network using different weight samples.

Experiments

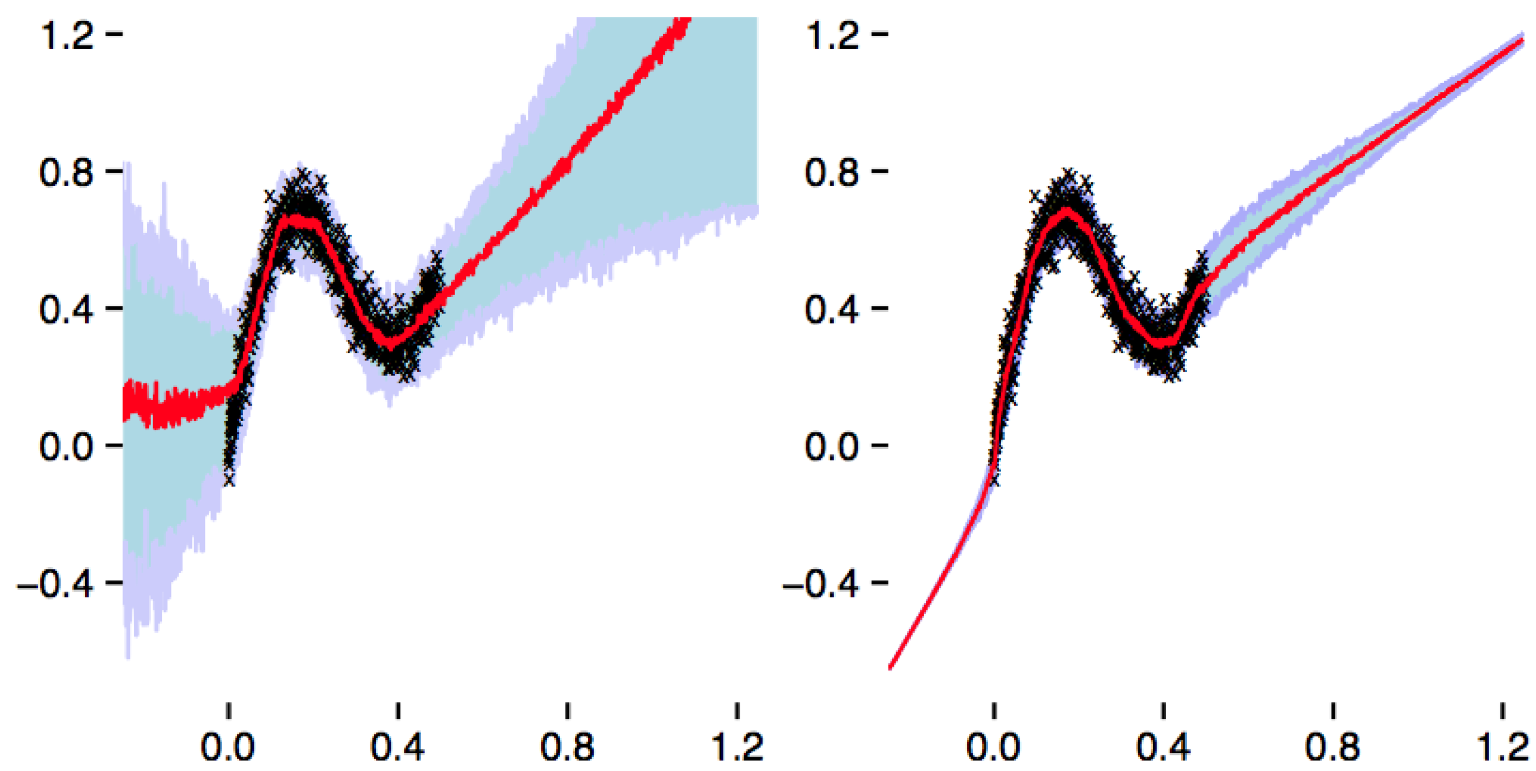
Test performance is comparable to dropout!



Pruning weights minimally reduces test error!

Proportion removed	# Weights	Test Error
0%	2.4m	1.24%
50%	1.2m	1.24%
75%	600k	1.24%
95%	120k	1.29%
98%	48k	1.39%

Appropriate uncertainty estimation
current work (left) vs. standard NN (right)



Discussion

Performance: Comparable to non-Bayesian state-of-the-art on MNIST

Computational Complexity: Two-fold increase in the number of trainable parameters

Compression: A large majority of weights can be removed without losing test-time performance

Uncertainty Estimation: Performed through ensembling many samples of network parameters

Reinforcement Learning: Exploration is naturally encouraged by weight sampling