Dropout as A Variational Approximation to Bayesian Neural Networks

David Burt, Sigurjón Ísaksson, Raj Shah, Minglong Sun

Introduction

As compared to standard neural networks, Bayesian neural networks better quantify the uncertainty in predictions and are more robust to over-fitting on small datasets. However, approximating the posterior over weights in these networks usually comes at a high computational cost. Gal and Ghahramani [1] approximate the posterior efficiently with a Bernoulli variational distribution, using dropout in the network. We recreate and extend their experiments with this architecture on MNIST, focusing on the issues of network certainty and overfitting.

Dropout Training as Approximate Variational Inference

Training Bayesian networks amounts to the intractable problem of learning the posterior over weights in the network, given inputs $\mathbf{X}$ and labels $\mathbf{Y}$. Assume a Gaussian prior over weights with mean $\mathbf{0}$ and precision matrix λI, and assume categorical likelihoods. The full posterior for this network is intractable, but we can use variational methods to approximate it. Define,

$$q(\mathbf{W}_i) = \mathbf{M}_i \cdot (\mathrm{diag}[z_{i,j}]_{j=1}^{K_i}), \qquad (1)$$
$$z_{i,j} \sim \mathrm{Bernoulli}(p_i).$$

where $\mathbf{W}_i$ are the weights in layer i and the $\mathbf{M}_i$ are the parameters we seek to maximize. Taking this as our variational approximation, maximising the variational lower bound is equivalent to minimising,

$$\mathbb{E}_{\omega\sim q(\omega)}[E(\mathbf{Y}, f(\mathbf{X},\omega))] - \mathrm{KL}(q(\omega)\|p(\omega))$$
$$\approx \frac{1}{N}\sum_{n=1}^{N} E(\mathbf{y}_n, f(\mathbf{x}_n, \hat{\omega}_n)) + \frac{\lambda}{2p}\|\omega\|_2^2, \qquad (2)$$
$$\omega_n \sim q(\omega).$$

where the approximation is an unbiased Monte Carlo estimate for the variational lower bound. This MC integration is equivalent to performing dropout after every trainable layer with cross entropy loss and ℓ^2 weight decay.

Small Data Sets: Overfitting

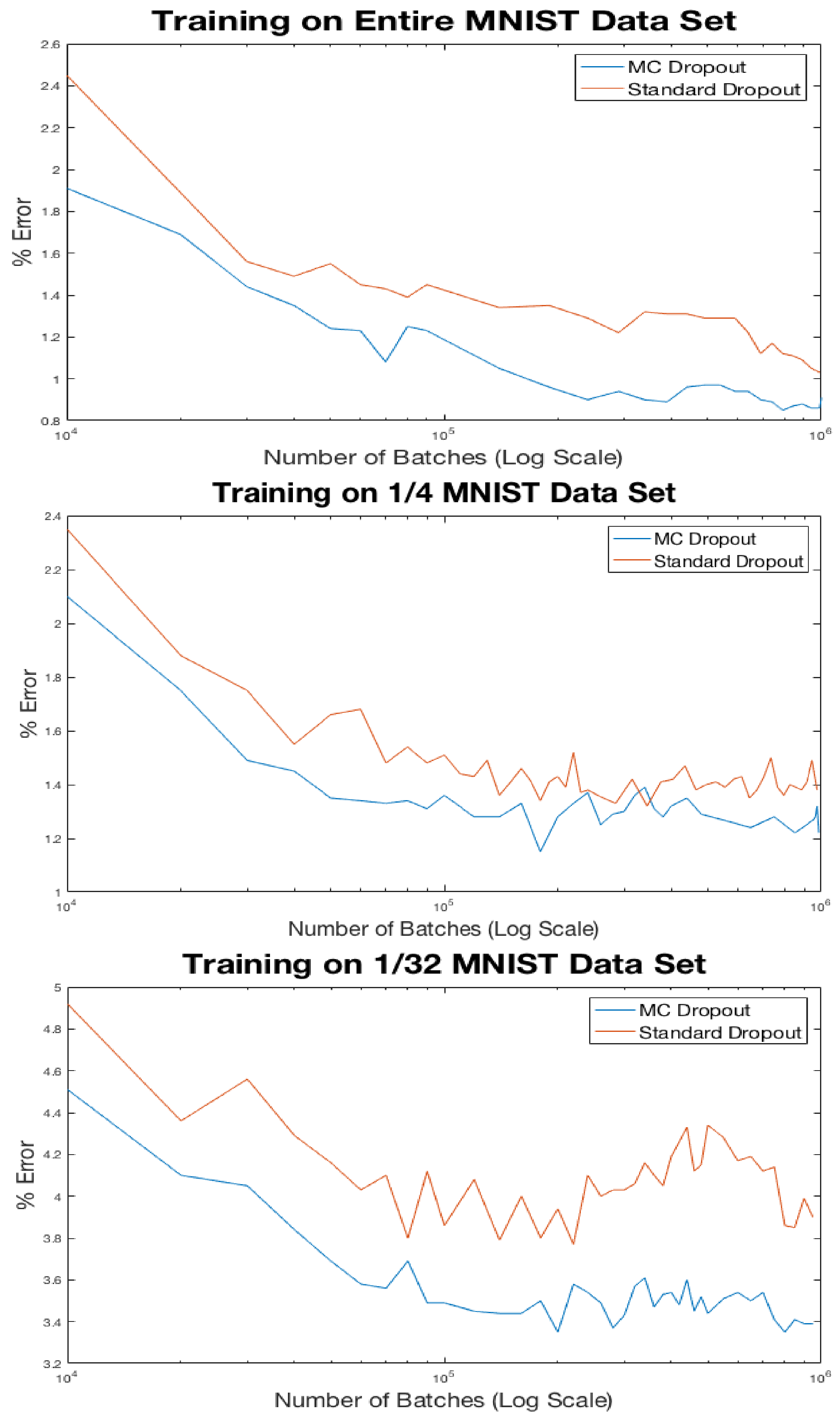


Figure: From top to bottom, training accuracies on all of MNIST, 1/4 of MNIST and 1/32 of MNIST

Noisy Digits

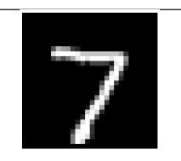
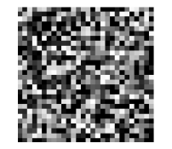
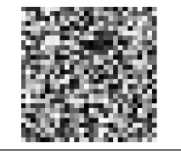
Corrupt Pixels (%)	Example Image	Softmax (%) Std.	Softmax (%) MC	Accuracy (%) Std.	Accuracy (%) MC
0		96.58	95.91	96.29	97.62
20		91.12	88.80	93.21	94.07
40		77.86	72.99	77.72	77.55
60		67.43	59.12	40.95	43.95
80		68.33	56.80	17.32	18.53
100		63.44	55.20	9.55	9.68

Table: On noisy data, MC dropout achieves higher levels of accuracy, but gives lower confidence in its predictions when compared to standard dropout.

Uncertainty

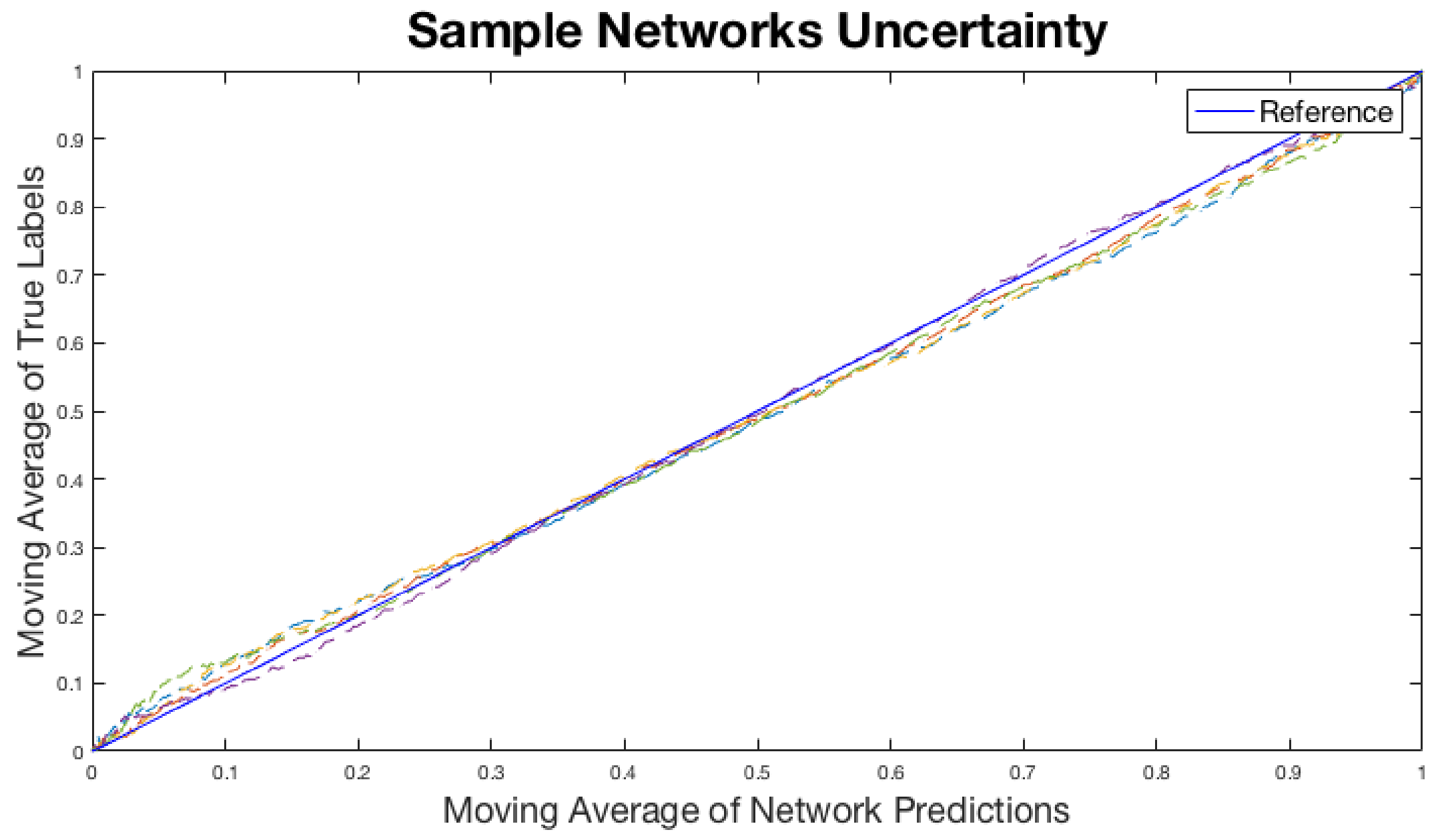


Figure: Samples from networks using dropout gives good estimates to posterior probabilities.

Bayesian Inference and MC Dropout

When $p = 0.5$, the variational distribution $q(\omega)$ defines a uniform distribution over the corners of a hyperrectangle with a one corner at $\mathbf{0}$ and the opposite corner at $\mathbf{M}$, where $\mathbf{M}$ was optimized in the training procedure to maximize the sum of the posterior probability assigned to the rectangle's vertices. Bayesian inference for a new x_n involves averaging the predictions at each corner, i.e.,

$$\bar{y}_n = \mathbb{E}_{q(\omega)}[f(x_n, \omega)]. \qquad (3)$$

This is intractable. Standard dropout uses a biased estimate, interchanging the expectation with f, yielding:

$$\bar{y}_n \approx f\left(x_n, \mathbb{E}_{q(\omega)}[\omega]\right) = f(x_n, \frac{1}{p}\mathbf{M}).$$

Geometrically, this means evaluating the network at the center of the optimized hyperrectangle. An alternative known as MC dropout is to use a Monte Carlo approximation (3),

$$\bar{y}_n \approx \frac{1}{T}\sum_{i=1}^{T} f(x_n, \omega_t), \qquad \omega_t \sim q(\omega). \qquad (4)$$

Bias and Variance

By adjusting the probability of dropout at test time, we can control the bias variance trade-off in the network evaluation.

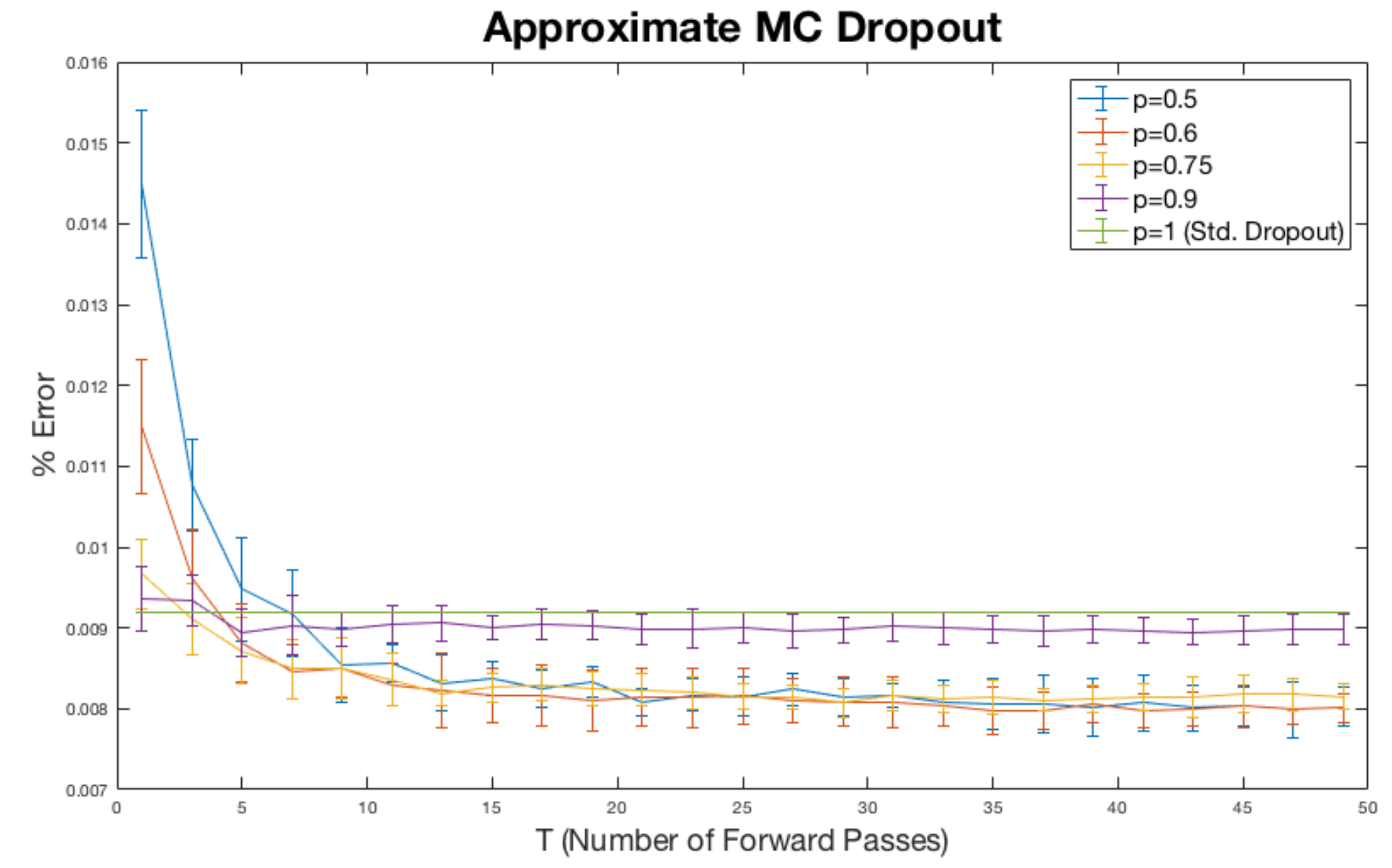


Figure: Mean and standard deviations for 10 trials plotted against the number of forward passes for various dropout probabilities at test time.

Future Experiments

- Impact of different p during training.
- Train on noisier and more complicated data.

References

[1] Yarin Gal and Zoubin Ghahramani.
Bayesian convolutional neural networks with Bernoulli approximate variational inference.
arXiv preprint arXiv:1506.02158, 2015.

Contact Information

- Git: https://github.com/raj-shah/MLSALT4
- Emails: drb62@cam.ac.uk, si318@cam.ac.uk, rns38@cam.ac.uk, ms2471@cam.ac.uk

UNIVERSITY OF CAMBRIDGE
DEPARTMENT OF ENGINEERING