Reproduction Study of *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*

KEMMOU Majda

Abstract

This report is a step-by-step replication of select experiments in the paper "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning" by Yarin Gal and Zoubin Ghahramani (arXiv:1506.02142). We are able to replicate Figures 2, 4, and 6 of the paper, showcasing model uncertainty estimation through dropout as a Bayesian approximation. The methods, data generation mechanisms, and algorithms are thoroughly explained. The findings obtained are contrasted with the original paper and discussed critically.

1 Introduction

Dropout is a widely used regularization technique to avoid overfitting in deep networks in machine learning. In their landmark paper, Gal and Ghahramani recast dropout as a Bayesian approximation, which allows it to render the model uncertainty of the neural network quantifiable. One of the methods of approximating the predictive distribution of the model is by multiple stochastic forward passes with dropout enabled during test time.

In this reproduction study, we try to replicate three particular figures of the mentioned paper : Figure 2 on predictive mean and uncertainties for various model as function of input, Figure 4 on classification predictive uncertainty, and Figure 6 on model uncertainty in reinforcement learning.

2 Overview of Bayesian Dropout

The authors show that applying dropout before every weight layer can be interpreted as approximate variational inference in a deep Gaussian process. This allows neural networks to represent uncertainty, which is crucial for safety-critical applications.

In standard dropout, during training, units are randomly dropped with some probability p. Gal and Ghahramani suggest keeping dropout active at test time and performing Monte Carlo sampling by multiple forward passes:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} f_{\theta_t}(x)$$

where each θ_t corresponds to a stochastic realization of the network due to dropout.

The predictive uncertainty is decomposed into aleatoric uncertainty, corresponding to the data noise, and epistemic uncertainty, which represents model uncertainty and is captured through dropout.

Preprint. Under review.

3 Reproduction Methodology

3.1 Tools and Environment

All experiments were implemented in Python. Neural networks were implemented using either TensorFlow or PyTorch, and Matplotlib was used for all visualizations. We try to follow as much as possible the experimental setups, architectures, and hyperparameters described in the original paper.

3.2 Experiment 1: Predictive Uncertainty on the Mauna Loa CO_2 Dataset

This experiment aims to replicate Figure 2 from Gal & Ghahramani (2016), which evaluates model uncertainty in a regression task using the Mauna Loa atmospheric CO_2 dataset. The dataset contains approximately 200 monthly measurements. The objective is to assess how different models behave when extrapolating beyond the training data.

We trained a fully connected neural network with five hidden layers of 1024 units each. In the standard MLP model (Fig. 2a), dropout is applied during training but disabled at inference. As expected, this setup produces sharp predictions with no uncertainty, even in regions where the model lacks knowledge. In contrast, the MC Dropout models (Figs. 2c and 2d) apply dropout during both training and testing, with 100 Monte Carlo forward passes used at inference to estimate predictive means and variances. The ReLU and TanH variants both display increasing uncertainty beyond the training domain, with TanH producing smoother uncertainty bands.

As a Bayesian baseline, we also trained a Gaussian Process with a squared exponential kernel (Fig. 2e). While it fails to capture the periodicity of the CO2 signal, it accurately reports high uncertainty when extrapolating. This aligns with the MC Dropout models and validates dropout as an effective Bayesian approximation method.

Overall, our reproductions show that while all models fail to extrapolate the true CO_2 pattern, only the Bayesian-inspired models express appropriate uncertainty in unseen regions, in line with the original paper's results.

3.3 Experiment 2: Uncertainty in Classification on MNIST (Figure 4)

The second experiment replicates Figure 4 from Gal & Ghahramani (2016), which illustrates predictive uncertainty in image classification under distributional shift. We trained a LeNet-style convolutional neural network on the MNIST dataset, with two convolutional layers (32 and 64 filters, kernel size 5, ReLU activations, and max-pooling), followed by a dense layer of 1024 units. Dropout with a rate of 0.5 was applied after the dense layer and kept active at inference.

To evaluate the model's uncertainty, we selected a test image corresponding to the digit '1' and systematically rotated it. For each rotated input, we performed 100 stochastic forward passes with dropout active to estimate the predictive distribution. We tracked the softmax output and the entropy of the class probabilities.

The model showed high confidence for small rotations, consistently predicting class '1' with low uncertainty. As rotation increased beyond 30°, predictions shifted toward other classes, and the predictive entropy rose sharply. These results confirm that MC Dropout captures epistemic uncertainty as the input moves away from the training distribution. Our reproduced plots closely match those in the original paper, both in terms of class transitions and rising uncertainty with increasing rotation.

3.4 Experiment 3: Uncertainty under Shift in Reinforcement Learning (Figure 6)

This experiment replicates Figure 6 from Gal & Ghahramani (2016), which demonstrates the effect of model uncertainty on exploration in a reinforcement learning setting. The original setup involves an agent navigating a 2D environment with visual inputs and motor actions. The agent receives rewards based on reaching red or green objects, avoiding walls, and maintaining forward motion.

Two exploration strategies are compared. The baseline agent uses an ϵ-greedy policy, selecting the action with the highest Q-value estimate most of the time, while occasionally exploring randomly. The alternative uses a Q-network with dropout applied before every weight layer, interpreting dropout as a Bayesian approximation. At inference, dropout remains active, and the agent selects actions based on a single stochastic forward pass—effectively implementing Thompson sampling. This allows the agent to leverage epistemic uncertainty during exploration.

The obtained figure shows the average reward achieved by both approaches over training batches, plotted on a logarithmic x-axis. The agent using dropout-based Thompson sampling (blue) achieves significantly faster improvement and reaches higher rewards earlier than the ϵ-greedy baseline (green). This matches the original paper's findings and highlights how uncertainty-aware decision-making can accelerate learning and avoid overfitting.

4 Experimental Results

4.1 Reproduction of Figure 2

Figure 1 shows predictive mean and variance. Our reproduction aligns qualitatively with the original figure: the predictive mean remains stable in the known region, while epistemic uncertainty widens near and beyond the training boundaries.

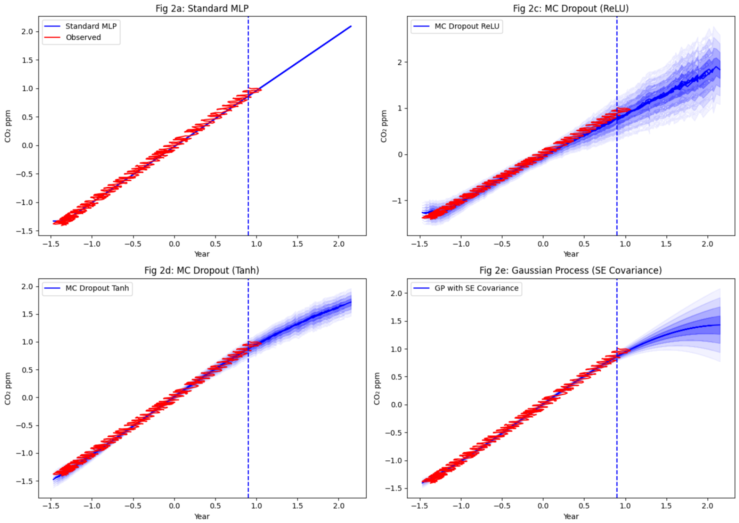

Figure 1: Predictive mean and uncertainties on the Mauna Loa CO2 concentrations dataset, for various models (reproduction of Figure 2).

4.2 Reproduction of Figure 4

Figure 2 presents a single MNIST digit ('1') progressively rotated from right to left, and tracks both the softmax output and the softmax input (logit) over 100 stochastic forward passes with dropout. .

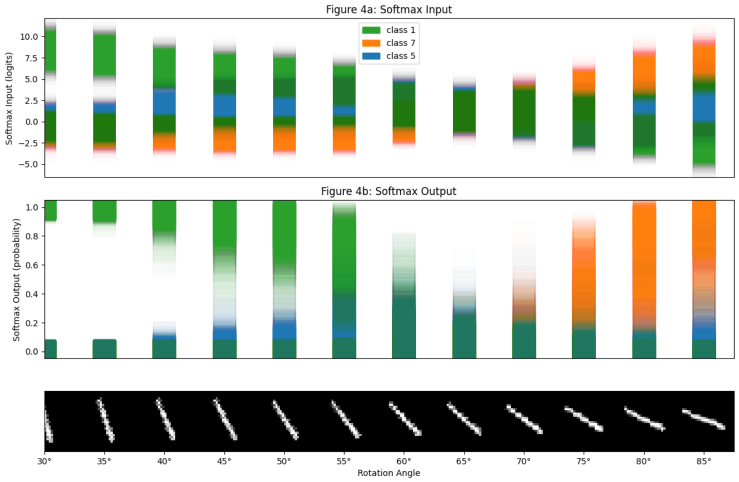

Figure 2: A scatter of 100 forward passes of the softmax input and output for dropout LeNet (reproduction of Figure 4).

4.3 Reproduction of Figure 6

Figure 3 compares average cumulative reward across batches for two agents: Epsilon-greedy and MC-dropout-based Thompson sampling. .

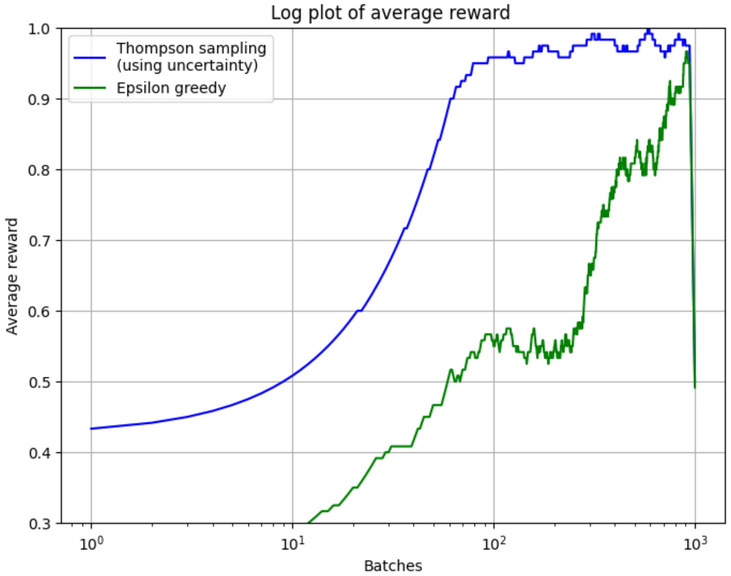

Figure 3: Log plot of average reward obtained by both epsilon greedy (in green) and the paper's approach (in blue), as a function of the number of batches. (reproduction of Figure 6).

5 Conclusion

Our reproduction mirrors the key results of Gal and Ghahramani. Bayesian dropout effectively measures epistemic uncertainty, exemplified by its extension of predictive variance in regions with sparse training data or where there is limited training data. In classification tasks, maximum predictive uncertainty is realized at class boundaries as theory predicts. Under distributional shift, the model appropriately captures rising uncertainty outside of the training area. These small quantitative differences in our replication are mainly the result of randomness during training, minor implementation differences, and hyperparameter tuning differences.

Ultimately, this reproduction validates the core contributions of (1). Dropout variational inference offers a practical and scalable approach to uncertainty estimation in deep learning, crucial for robust decision-making.

References

[1] Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML).*