Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

AHAMAN ULLAH Efaz
EURECOM
Efaz.Ahaman-Ullah@eurecom.fr

Abstract

Deep learning tools has a huge field of application : regression, classification, etc. Even if the Bayesian approach offers a theoretical framework to compute the model's uncertainty, it is often forgotten because of its computational cost. In this report, I will try to do a summary of the paper "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning", where the authors Yarin Gal and Zoubin Ghahramani, tried to construct a new theoretical framework that uses dropouts in deep neural networks (DNN) to reduce this computational cost. This offers an opportunity to include the uncertainty in DNN's without increasing the computational cost. They showed a considerable improvement in predictive log-likelihood and RMSE compared to existing state-of-the-art method.

1 Introduction

Nowadays, Depp learning (DL) is used in a lot of scientific fields like physics, biology where model uncertainty is critical. But classical DL models like NNs, CNNs with softmax output often misrepresent the uncertainty by giving the confidence of the model. This is problematic in sensible domains like nuclear plants or reinforcement learning (RL). Because there knowing the uncertainty level of the output can have a huge impact on decision making.

The proposed the to interpret the dropout as a Bayesian approximation to a Gaussian Process (GP). Which will give a meaningful uncertainty measurement without changing that much the models and without increasing the computational cost. They provide :

- a theoretical framework between dropout and Bayesian inference.
- show how model uncertainty can improve regression and classification (on MNIST for instance).
- show benefits in RL with a more efficient exploration using uncertainty.

2 Dropout as a Bayesian Approximation

They show that a neural network (NN),with arbitrary depth and non-linearities, with dropout applied before every weight layer, is mathematically equivalent to performing an approximation to the probabilistic deep Gaussian process(GP). The dropout objective minimises the Kullback-Leibler divergence (KL) between an approximation and the posterior and its deep Gaussian process.

The standard loss function with dropout and L_2 regularization :

$$\mathcal{L}_{\text{dropout}} = \frac{1}{N}\sum_{i=1}^{N} E(y_i, \hat{y}_i) + \lambda \sum_{i=1}^{L} \left(\|W_i\|_2^2 + \|b_i\|_2^2\right) \tag{1}$$

where:

- $\hat{y}_i$: model output
- W_i, b_i: weights and biases of layer i
- λ: L_2 regularization coefficient
- $E(\cdot, \cdot)$: loss function (e.g., softmax or square loss)

Define a covariance kernel:

$$K(x, y) = \int p(w)p(b)\,\sigma(w^\top x + b)\,\sigma(w^\top y + b)\,dw\,db \tag{2}$$

where $\sigma(\cdot)$ is a non-linear activation function. This corresponds to a GP with an infinite number of hidden units.

Define a variational distribution $q(\omega)$ with:

$$W_i = M_i \cdot \text{diag}(z_{i,1}, \ldots, z_{i,K_{i-1}}), \quad z_{i,j} \sim \text{Bernoulli}(p_i)$$

This approximates the posterior $p(\omega \mid X, Y)$ in a deep GP.

Bayesian Predictive Distribution :

$$p(y \mid x, X, Y) = \int p(y \mid x, \omega)\, p(\omega \mid X, Y)\, d\omega \tag{3}$$

The integral is intractable so they used Variational approximation with Dropout.

Variational Approximation with Dropout : They minimised the KL divergence between $q(\omega)$ and $p(\omega \mid X, Y)$:

$$\mathcal{L}_{\text{GP-MC}} = \frac{1}{N}\sum_{n=1}^{N} -\log p(y_n \mid x_n, \hat{\omega}_n)/\tau + \sum_{i=1}^{L}\left(\frac{p_i \ell^2}{2\tau N}\|M_i\|_2^2 + \frac{\ell^2}{2\tau N}\|m_i\|_2^2\right) \tag{4}$$

where:

- $\hat{\omega}_n \sim q(\omega)$ is a Monte Carlo sample
- τ: model precision (inverse variance)
- ℓ: prior length-scale

For the square loss, we have:

$$E(y_n, \hat{y}(x_n, \hat{\omega}_n)) = -\log p(y_n \mid x_n, \hat{\omega}_n)/\tau$$

and hence (4) reduces to (1) under appropriate scaling.

Approximation of the predictive distribution using Monte Carlo sampling:

$$p(y \mid x) \approx \frac{1}{T}\sum_{t=1}^{T} p(y \mid x, \hat{\omega}^{(t)}), \quad \hat{\omega}^{(t)} \sim q(\omega) \tag{5}$$

To sum up :

- Dropout performs approximate Bayesian inference.
- Enables epistemic uncertainty estimation.
- Applicable to various dropout variants (e.g., drop-connect, Gaussian noise).

3 Obtaining Model Uncertainty

The approximate predictive distribution is given by:

$$q(y^* \mid x^*) = \int p(y^* \mid x^*, \omega) q(\omega) \, d\omega \tag{6}$$

where $\omega = \{W_i\}_{i=1}^{L}$ are the weight matrices of an L-layer model.

Estimation of the first moment (mean) of this predictive distribution by performing T stochastic forward passes:

$$\mathbb{E}_{q(y^*|x^*)}[y^*] \approx \frac{1}{T} \sum_{t=1}^{T} \hat{y}^*(x^*, W_1^{(t)}, \ldots, W_L^{(t)}) \tag{7}$$

Also known as MC dropout.

Estimation of the second raw moment:

$$\mathbb{E}_{q(y^*|x^*)}[(y^*)^T y^*] \approx \tau^{-1} I_D + \frac{1}{T} \sum_{t=1}^{T} \hat{y}^{*(t)T} \hat{y}^{*(t)} \tag{8}$$

The predictive variance is then approximated as:

$$\mathrm{Var}_{q(y^*|x^*)}(y^*) \approx \tau^{-1} I_D + \frac{1}{T} \sum_{t=1}^{T} \hat{y}^{*(t)T} \hat{y}^{*(t)} - \left(\mathbb{E}_{q(y^*|x^*)}[y^*]\right)^T \left(\mathbb{E}_{q(y^*|x^*)}[y^*]\right) \tag{9}$$

The model precision τ is given by:

$$\tau = \frac{p\ell^2}{2N\lambda} \tag{10}$$

where p is the keep probability, ℓ is the prior length-scale, N is the number of training points, and λ is the weight decay.

The predictive log-likelihood can be approximated by:

$$\log p(y^* \mid x^*, X, Y) \approx \log \sum_{t=1}^{T} \exp\left(-\frac{1}{2}\tau \|y^* - \hat{y}^{*(t)}\|^2\right) - \log T - \frac{1}{2}\log 2\pi - \frac{1}{2}\log \tau^{-1} \tag{11}$$

These approximations allowed them to derive uncertainty estimates from dropout models.

4 Experiments

In this I will try to a comparison between what the authors did and what we did. We focused ourselves in doing two experiments : Model uncertainty in Regression Tasks and Model uncertainty in classification tasks.

4.1 Model Uncertainty in Regression Tasks

We tried to do a first experiment using the Mauna Loa atmospheric CO_2 dataset. We used a NN with 5 hidden layers (each with 1024 units). They were trained using ReLU or TanH activations and with a dropout rate of 0.1 and 1e6 iterations, 100 forward passes.

- **Standard Dropout (Fig. 1a)**: Here our model predicts point estimates via weight averaging. But it fails to quantify uncertainty and gives overconfident wrong predictions.
- **Gaussian Process (GP, Fig. 1b)**: Predicts similar values but with increasing uncertainty in regions far from data.
- **MC Dropout with ReLU/tanH (Fig. 1c,d)**: Here the model captures uncertainty better, where shades of blue represent confidence intervals .

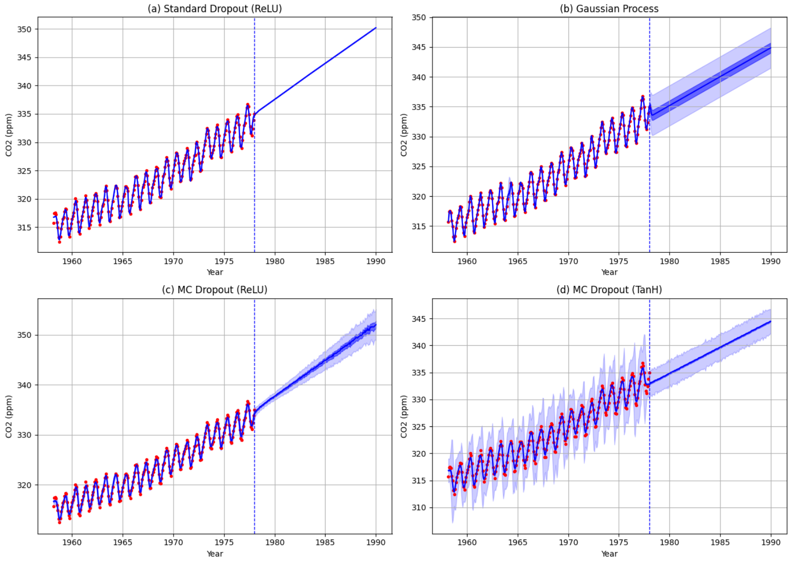

Figure 1: Predictive mean and uncertainties on the Mauna Loa CO2 concentrations dataset, for various models.

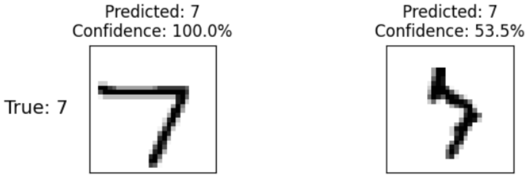

Figure 2: Comparison of a 7 classified with high and low confidence scores from MC dropout

- **Key Finding**: Finally the model uncertainty increases with distance from training data (especially with ReLU). At the same time the tanH models remain bounded in uncertainty due to their saturating nature.

Overall our results are very close to the the paper . We observed that the dropout approximates the behaviour of Gaussian Processes with different covariance functions. The moments (mean and variance) of the dropout models converge to those of the corresponding GP. A small number of forward passes ($T = 10$) is often sufficient to estimate uncertainty accurately.

4.2 Model Uncertainty in Classification Tasks

Secondly the authors tried to use the MC dropout for classification tasks. They used an experiment of digit classification with the MNIST dataset. They used the LeNet convolutional neural network model (Le-Cun et al., 1998) with a Dropout probability of 0.5. In order to reproduce this experiment, we used the same model, parameters

The Figure 2 shows the difference in confidence when classifying a digit from a very straightforward picture and then from a contentious one. The first 7 seems to be classifiable with low uncertainty and our model had a perfect confidence score. The second one is much harder to classify, and as a result our model gives a much smaller confidence score. With a classical CNN model, we would have the same output. Here, with MC dropout we obtain a more nuanced result.

We tried to plot like the Figure 4 from the paper. It represents a scatter plot of the softmax input and output for each forward pass of our model, with different images of a one rotated to different degrees.

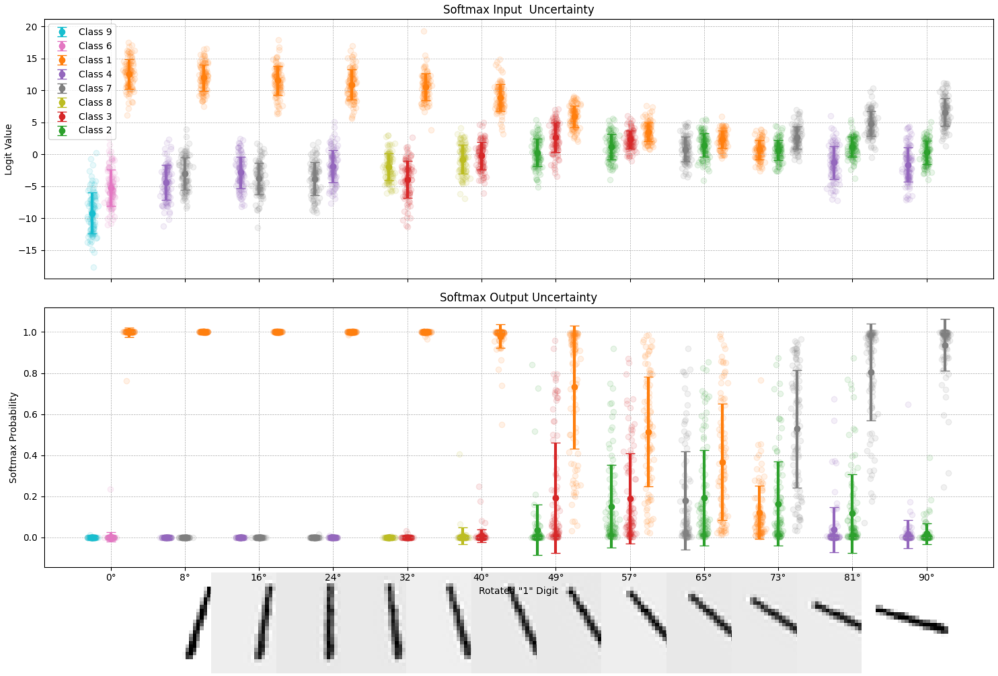

Figure 3: A scatter plot of 100 forward passes of the softmax input and output for dropout LeNet

We obtained, finally, a very similar result like the authors.Our results on the chosen class differ, with [1 1 1 1 1 1 1 1 1 7 7 7] for us and [1 1 1 1 1 5 5 7 7 7 7 7] for the original paper, but we can reach the same conclusion. When the rotation is over 50 degrees, the uncertainty in the softmax output overlaps multiple classes. In a critical scenario, simply returning the output choice when the uncertainty is so high would not be reasonable.

4.3 Predictive Performance Comparison

We did not do this part of the experiment. Here, the authors compared dropout-based models with two Bayesian inference methods :

- **VI**: Variational Inference (Graves, 2011)
- **PBP**: Probabilistic Backpropagation (Hernández-Lobato & Adams, 2015)

They evaluated RMSE and predictive log-likelihood on standard UCI datasets.

5 Conclusion

To conclude, the authors developed a probabilistic interpretation of dropouts, enabling DL models to capture model uncertainty without increasing computational cost. Their study showcased both in theoretical and practical demonstration that Bayesian modeling principles can be integrated into standard neural networks using dropout.

Other dropout variants, like dropconnect, etc, can also be understood under this framework.

Ongoing research continues non-linearity–regularization combinations on predictive distributions and uncertainty estimation.

P.S : Wanted to put a "References" section, but don't have enough space...