**Project Assignment - Uncertainty Quantification**

# 1 Introduction

As deep learning model is being widely used to solve a lot of real-world tasks, the question of model confidence has been brought to the fore. With uncertainty estimate, it is possible to have estimates of the models’ prediction uncertainties. Different approaches have been proposed for this type of task. In this paper we will discuss the Bayes by backprop and dropout in regression problems.

# 2 Models

The models explored for this task are the Bayes by backprop and the dropout as a Bayesian approximation. The theory behind these models is detailed in this section.

## 2.1 Bayes by backpropagation

The first deep probabilistic regression model selected for the uncertainty estimation is Bayes by backpropagation (Blundell et al. 2015). Given a deep probabilistic model p(y|x,w) where y is a continuous variable from a regression set. With the training dataset D={x(i), y(i)}, we construct a likelihood function p(D|w). By maximizing p(D|w) we obtain the maximum likelihood function (MLE). The negative log likelihood is the optimization objective during training. MLE can be used to obtain the point estimates of parameters. Maximum a Posteriori (MAP) can also be used as it is better at reducing overfitting. By Bayes rule, multiplying the likelihood with the prior approximates to the posterior distribution p(w|D)∝p(D|w)p(w).

Optimising p(D|w)p(w) gives the Maximum a posteriori (MAP). MAP has a regularisation effect and thus helps to prevent overfitting. Both MAP and MLE are point estimates. To make predictions that take the weight uncertainty into account, we would need to have a full posterior distribution over the parameters.

The posterior in a deep network is intractable. Thus, it is approximated with a variational posterior from a known distribution . The parameters for is estimated by minimising the Kullback-Leibler divergence between the true posterior and the variational posterior .

The optimisation objective or loss function is expressed as:

The loss function is approximated by drawing Monte Carlo samples w(i) from :

The aim is to minimise this loss function using backpropagation.

## 2.2 Dropout as Bayesian optimisation

Dropout is a regularisation technique applied to deep neural network (Srivastava et al. 2014). It works by adding multiplicative noise to the input of the neural network layers. For a fully connected neural network, the dropout is expressed as:

with

Where the symbol represents the element-wise multiplication of Z input matrix of M x K input features with a matrix of M x K independent noise variables . The idea behind this technique is that by adding noise to the weight of the layers, it would help to prevent overfitting of the weight parameters to the training data.

Gal and Ghahramani (2015) proposed using dropout as Bayesian optimisation. By using dropout in a regression network during training and testing, the network is able to generate a different output for every forward pass of the same input. The authors demonstrated in their paper that these multiple passes approximate to the Monte Carlo sampling in Bayesian statistics.

# 3 Experiment

For the regression task, ten (10) datasets were used for the experiment. The dataset was split into 90% train set and 10% test set. Each of the datasets had target outputs, y.

# 4 Evaluation

For the evaluation, we used the point predictions were measured using root mean squared error, The prediction intervals prediction interval coverage probability (PICP) and mean prediction interval width (MPIW) was measured for 95% interval. PICP and MPIW are metrics proposed by Pearce et al. (2018). The results of the empirical evaluation for each of the datasets are shown in this section.

The prediction interval (PI) is defined as where is 0.95 PI.

This became for 95% prediction interval. Where are the mean and standard deviations respectively of the predicted output.

The RMSE was normalised by multiplying with the standard deviation of the training data.

## 4.2 Results from empirical evaluation

### 4.2.1 Bayes by Backprop

For each of the datasets, the model was trained for 100 epochs. Table I shows the values of RMSE, PICP and MPIW for the regression datasets for Bayes by backprop.

Table I: Empirical evaluation of Bayes by Backprop model on the regression datasets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Housing | Concrete | Energy | Wine | Yatch | Kin8nm | Power |
| RMSE | 3.112 | 5.221 | 2.140 | 0.610 | 1.515 | 0.139 | 4.309 |
| PICP | 0.955 | 0.902 | 0.973 | 0.935 | 1.0 | 0.981 | 0.979 |
| MPIW | 1.433 | 0.898 | 0.525 | 2.828 | 0.489 | 1.867 | 1.115 |

### 3.2.2 Dropout as Bayesian approximation

Table II shows the values of RMSE, PICP and MPIW for the regression datasets for dropout as Bayesian approximation.

Table II: Empirical evaluation of Dropout model on the regression datasets

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Housing | Concrete | Energy | Wine | Yatch | Kin8nm |
| RMSE | 4.081 | 10.938 | 4.446 | 0.637 | 3.703 | 0.199 |
| PICP | 0.503 | 0.685 | 0.501 | 0.229 | 0.779 | 0.264 |
| MPIW | 1.532 | 2.164 | 1.731 | 0.977 | 2.341 | 0.752 |

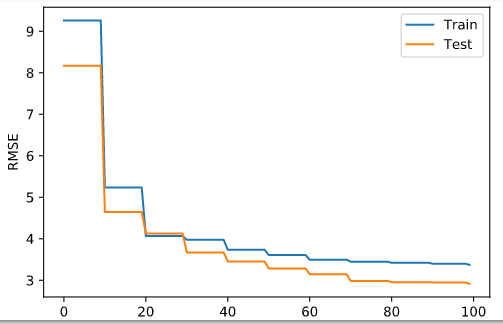


Fig 1: RMSE scores for the train and test set of Housing data (Bayes by Backprop)

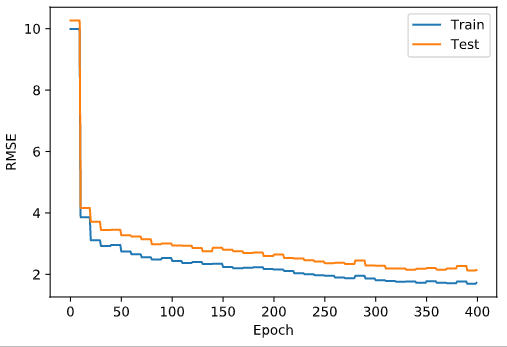


Fig 2: RMSE scores for the train and test set of Energy data (Bayes by Backprop)

Fig 1 shows the graph of the RMSE against the epochs for train and test data. The error for training data appears to be higher than the test data. One possible explanation could be as a result of how the data was split. The test data is relatively much smaller than the training data with test set being only 10% of the entire dataset. The energy dataset has lower training error than the test error as shown in Fig. 2. The model does not overfit in this case and it reflects on the RMSE score which is lower than most of the other datasets.

# 5 Findings and discussion

It was observed that the dropout approximation depends on the dropout rates. Different values have different effects on the point estimates and PIs. The Bayes by backprop method appears to have more stability than the Dropout method. For the dropout method, it was observed for most of the datasets that the error scores were fluctuating beginning from 10 epochs. The Bayes by backprop performed better than the dropout on the point predictions and PIs across the datasets.

## 5.1 Improvements

The local reparametrisation trick (Kingma et al. 2015) was proposed to improve the bayesian model. It works by transforming ρ to a positive vector σ via the softplus function:

The “reparametrization trick” is then performed that separates the randomness from the parameters of .

w=μ +σ ∘ ϵ

where the ∘ operator represents the element-wise multiplication.

An extension of the regular dropout, named the variational dropout, was proposed by Kingma et al. (2015). They suggested the variational dropout improves the adaptability of the normally fixed dropout rates p, to the data. During the experiment, different dropout rates were also used to see the effect on the model uncertainty as shown in Table 3.

Table 3: Evaluation at different dropout rates (housing dataset)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0.1** | **0.2** | **0.5** | **0.8** |
| PICP | 0.552 | 0.649 | 0.525 | 0.426 |
| MPIW | 2.063 | 2.128 | 1.431 | 1.075 |
| RMSE | 3.276 | 3.642 | 3.178 | 5.402 |

# 6 Conclusion

In sensitive domains like healthcare and autonomous driving, there is a need to query not just the accuracy of the model but also the confidence of the model’s predictions. In this report, uncertainty estimation of non-linear models has been explored in a regression task. Ten regression datasets were used to determine the point prediction and the prediction interval of the models.

# References

Blundell, C., Cornebise, J., Kavukcuoglu, K. and Wierstra, D., 2015. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Gal, Y. and Ghahramani, Z., 2016, June. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*(pp. 1050-1059).

Kingma, D.P., Salimans, T. and Welling, M., 2015. Variational dropout and the local reparameterization trick. In Advances in Neural Information Processing Systems (pp. 2575-2583).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), pp.1929-1958.

Pearce, T., Zaki, M., Brintrup, A. and Neely, A., 2018. High-quality prediction intervals for deep learning: a distribution-free, ensembled approach. *arXiv preprint arXiv:1802.07167*.