Skolkovo Institute of Science and Technology

MSc Data Science, 2nd year, Uncertainty Quantification

Project Report

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

Chesakov Daniil

Glazov Vsevolod

Kovalev Evgeny

MOSCOW

2019

1 Introduction

Deep neural networks prove to be a powerful tool to solve many maching learning problems. However, it is hard to tell how certain is model, solely relying on its predictions. Bayesian models can help to estimate it, but they can lead to a big computational cost. Such regularization technique as dropout come to rescue. In this project, we studied a paper [1] where the authors prove that applying a dropout actually approximates the probabilistic deep Gaussian process. The main idea is that, having a trained network with dropout, we may not switch off the dropout for the inference. Instead, we can have multiple runs all having slightly different architecture caused by dropout, and then look and the deviation of the results — if it is high, then the model is uncertain about its outputs. We reimplemented the experiments on two regression datasets and one classification one and also shown the importance of uncertainty estimation in classification task.

2 Regression

2.1 CO_2 dataset

As the first dataset we use a subset (collected from 1965 to 2004) of the monthly atmospheric CO2 concentrations dataset derived from in situ air samples collected at Mauna Loa Observatory, Hawaii [2] to evaluate model extrapolation. The dataset is then processed – trend is extracted and data is normalized so the mean is zero.

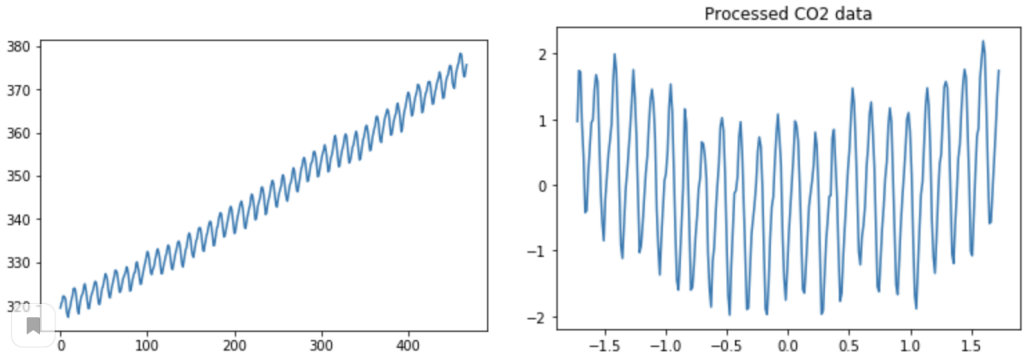

Figure 1: Visualization of CO_2 data

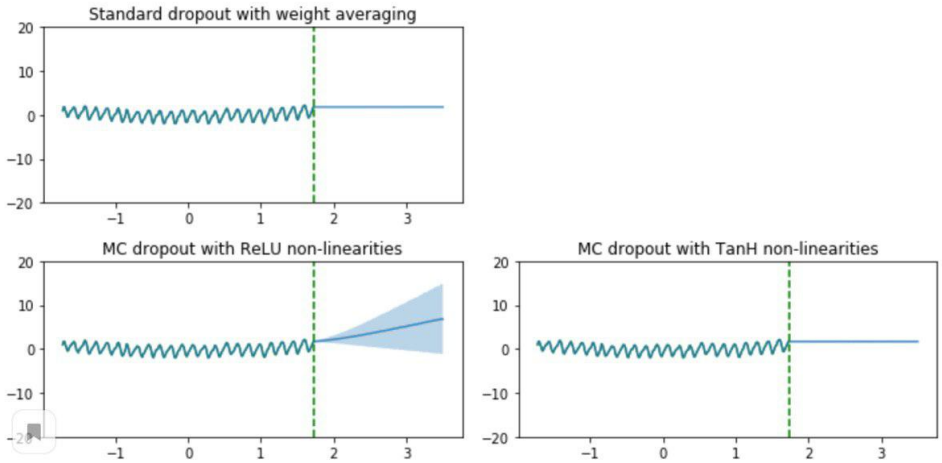

Figure 2: Results of extrapolation on CO_2 data with uncertainty estimation

We trained several models on the CO_2 dataset. We use NNs with 5 hidden layers and 1024 hidden units. We use either ReLU non-linearities or Tanh non-linearities in each network, and use dropout probability $p = 0.2$. As pointed out by paper, similar results can be obtained when using 4 hidden layers or dropout probability of 0.1.

As we can see from Figure 2, the dropout network with ReLU non-linearities successfully show the uncertainty for points away from the training data points. The uncertainty of dropout network with Tanh non-linearities doesn't increase far from the data, presumably because Tanh saturates whereas ReLU does not, as explained by in paper.

2.2 Irradiance dataset

We also repeated our experiment on solar activity data from 1610 to 2000. See Figure 3. Data was collected by World Data Center for Paleoclimatology, Boulder and containce new reconstructions of spectral irradiance [3]. For the experiment the data was normalized to have zero mean and unit variance. We estimated uncertanty for extrapolation task for years: 2000 - 2100.

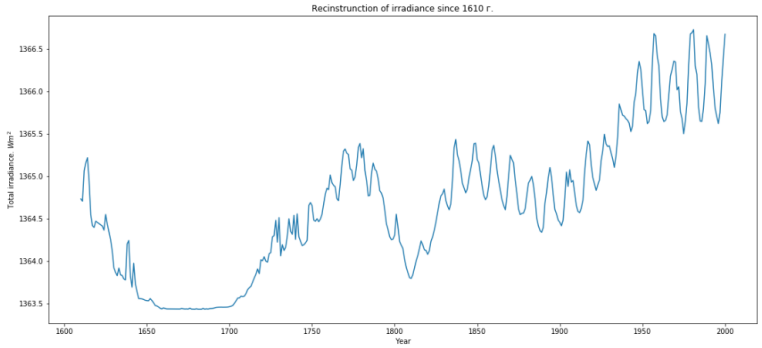

Figure 3: Visualization of Solar Irradiance Reconstruction dataset

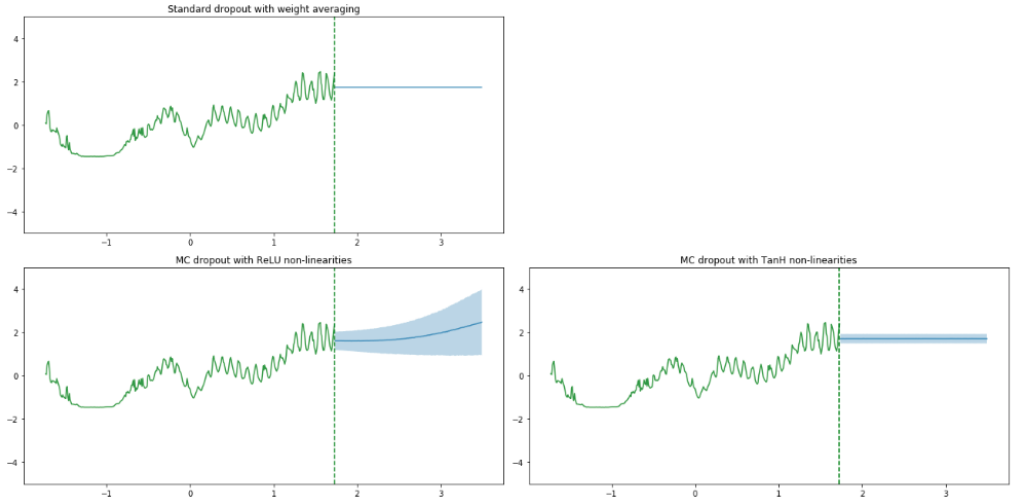

Figure 4: Results of extrapolation on irradiance data with uncertainty estimation

For this task we also trained the MLP with 1024 hidden units in a layer. We tried 4 and 5 hidden layers, as well as different non-linear ReLU and Tanh functions. After each hidden layer we applied Dropout layer with $p = 0.2$ or $p = 0.1$. As it was mentioned in the paper, there was no any visual differences in results depending on number of layers or p. To obtain uncertanty we applied the model to the dataset including it's extrapolation part of $T = 1000$ times. As we can see on Figure 4 the expected results were repeated. The model didn't catch complicated structure of the dataset but predicted the trend. For ReLU function uncertanty of the model irceasing with with a distance from the training examples, while for Tanh function, the uncertanty reachs up the limit due to restricted nature of it.

2.3 RMSE and LL estimation

To estimate how well model fits the data we computed predictive log-likelihood and RMSE. We repeated the experiment hold in the paper on our Irradiance dataset. We compared RMSE and predictive log-likelihood of

			RMSE		*LL*	
Dataset	N	Q	**PBP**	**Dropout**	**PBP**	**Dropout**
Lean2000 irradiance	391	1	0.25	0.28	−1.79	−1.87

Table 1: RMSE and Log-likelihood

our dropout approach to the Probabilistic Back-propagation (PBP) proposed in [4].

The goal was to compare the uncertainty quality obtained from a naive application of dropout in NNs to the specialized method developed to capture uncertainty. Following Bayesian interpretation of dropout as it was proposed in the paper, we had to define a prior length-scale, and find an optimal model precision parameter τ in terms of optimizing LL.

Similar to [4] to estimate PBP we used Bayesian optimisation over validation log-likelihood to find optimal τ, with length-scale set to 10^{-2}. In the estimation of Dropout approach we used grid-search over τ from $[0.05, 0.1, 0.15]$ and dropout rates from $[0.005, 0.01, 0.05, 0.1]$.

The obtained values are listed in the Table 1. While in the paper authors stated that Dropout approach gives better results in terms of RMSE and LL we got opposite results. Probably that happened due to the data complexity.

3 Classification

For the classification task, we used LeNet [5] architecture trained on MNIST dataset. The dropout layer was added before the last output layer of the network. Four different values of p (probabilities of leaving a neuron) were used for dropout: 0, 0.25, 0.5, 0.75. The setup for the training was the following. Batch size was set to 32, initial learning rate was equal to 0.005, the number of epochs was 500, the optimizer used was SGD with momentum equal to 0.9 and weight decay equal to 10^{-6}, also early stopping was used (model from the best epoch according to the test set was taken). Such learning rate policy was applied:

$$\mathrm{lr}_i = \mathrm{lr}_{\mathrm{init}}(1 + \alpha i)^{-\beta},$$

with $\alpha = 1e-4$, $\beta = 0.75$, i is the number of iteration, $lr_{\mathrm{init}} = 0.005$ is the initial learning rate.

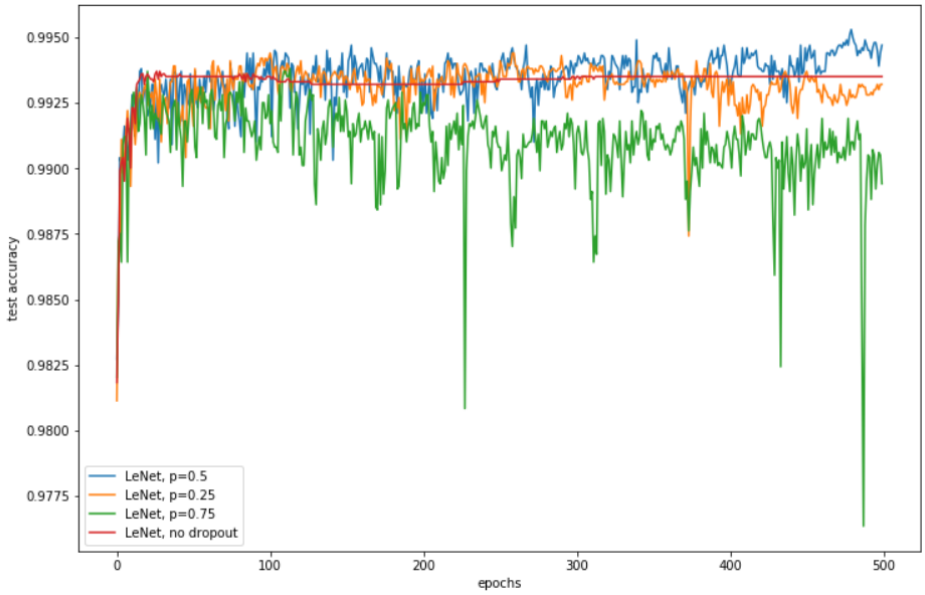

Figure 5: Test accuracy logs for different dropout rates

The results of training are shown on Figure 5 and Table 2. We can see that the model with $p = 0.5$ is the best. Also the results show that dropout proves to be a good regularization technique in this case, enabling the model to learn for much more epochs.

Table 2: Results of training on MNIST

Model	accuracy	best epoch
LeNet, p=0.5	**0.995308**	479
LeNet, p=0.25	0.994409	100
LeNet, p=0.75	0.993810	113
LeNet, no dropout	0.993710	26

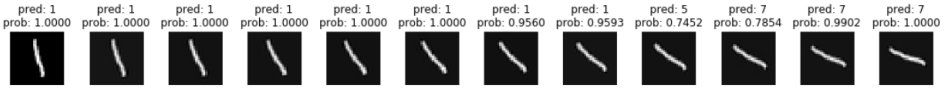

Figure 6: Predictions for rotated digit, $p = 0.5$

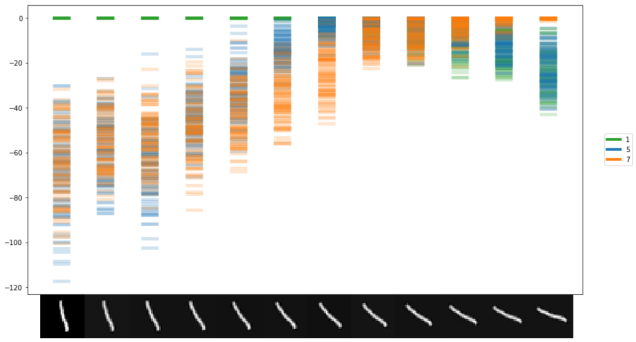

Figure 7: Softmax inputs for rotated digit, $p = 0.5$

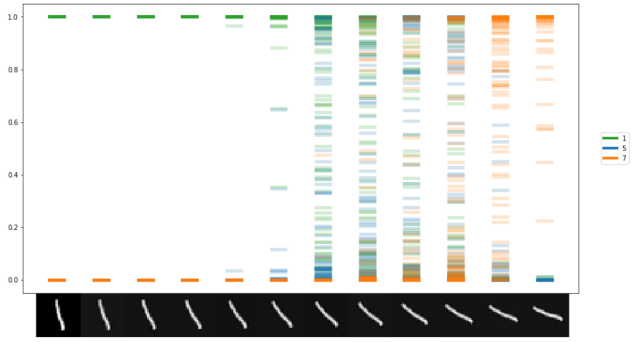

Figure 8: Softmax outputs for rotated digit, $p = 0.5$

Then one of the digits was taken from the test data and it was rotated on twelve angles uniformly from 0 to 60 degrees (see Figure 6). The predictions of network with switched off dropout with rate $p = 0.5$ show that with the rotation of the image the network starts to predict the wrong class 7. Then we turned on the dropout and ran 100 forward passes of the network for each rotated digit. The visualizations are shown on Figure 7 and Figure 8. Softmax values for the true class 1 are shown in green, for wrong classes 5 and 7 in blue and orange, respectively. We can see that with the rotation of the image the model uncertainty grows. At first, the model predicts the true class correctly with nearly to deviation at all. Then the deviation of softmax values considerably grows, which means that the model becomes uncertain about its predictions. However, on Figure 6 it is shown that the class for the penultimate digit is predicted with "probability" of approximately 99%, while the Figure 8 clearly shows that the model is uncertain in this case. Such observation implies that here it might be wrong to trust the model softmax outputs blindly, and it is a good idea to check the model uncertainty in the described way and probably give such examples to the human assessors.

The results of the experiment with model with dropout rate $p = 0.25$ are shown on Figures 9, 10 and 11. Now the blue identified wrong class 2, other classes have the same colors. We also see that the model becomes uncertain with the rotation of the image, however, outputting relatively big numbers from the softmax layer for the wrong class.

The results of the experiment with model with dropout rate $p = 0.75$ are shown on Figures 12, 13 and 14. The colors are the same as in the previous case. We can see that the model become even more uncertain with the rotation of the digit. Probably this is because the value of dropout is too big, so the resulting architectures become very different on the forward passes, which leads to the big deviation of the results.

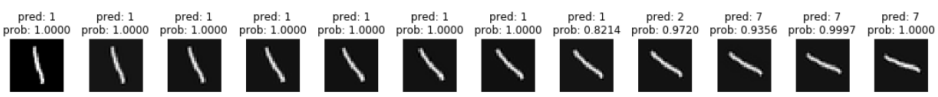

Figure 9: Predictions for rotated digit, $p = 0.25$

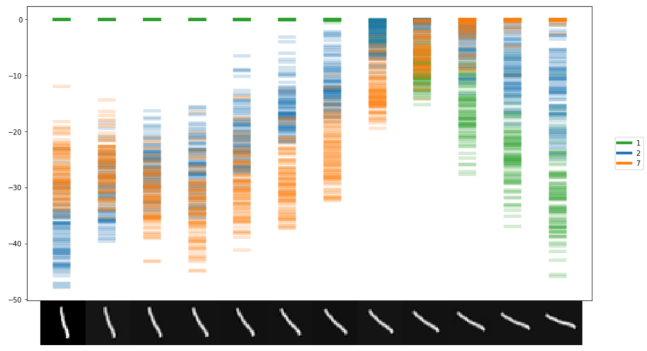

Figure 10: Softmax inputs for rotated digit, $p = 0.25$

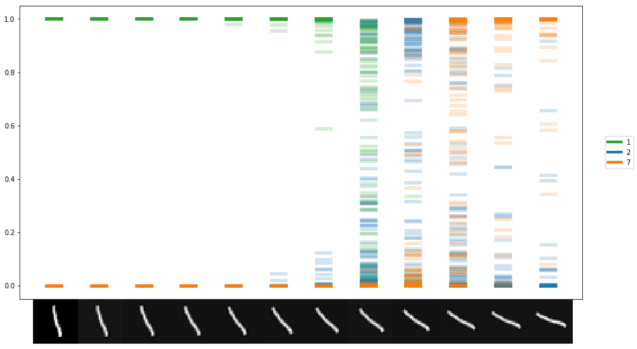

Figure 11: Softmax outputs for rotated digit, $p = 0.25$

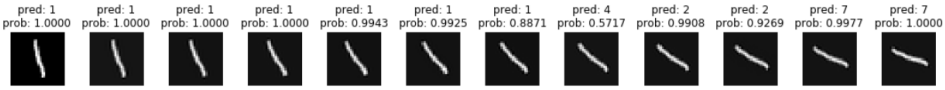

Figure 12: Predictions for rotated digit, $p = 0.75$

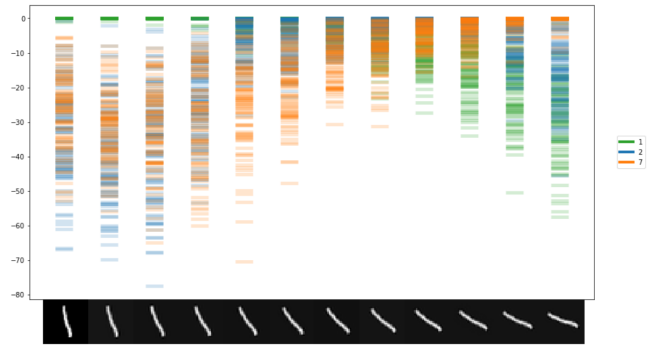

Figure 13: Softmax inputs for rotated digit, $p = 0.75$

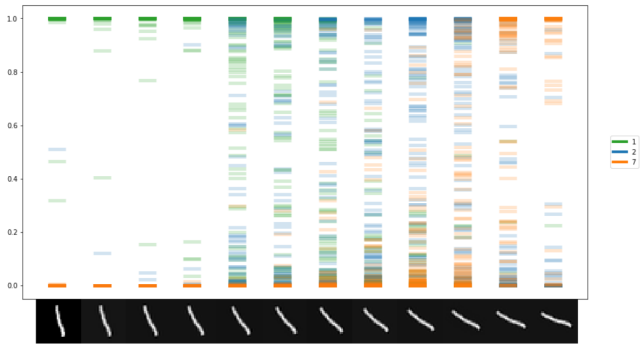

Figure 14: Softmax outputs for rotated digit, $p = 0.75$

4 Conclusion

In this work we studied how such regularization technique as dropout can be applied in order to represent model uncertainty. We implemented the experiments from the studied paper and compared the uncertainty of different models in regression and classification tasks. We also explicitly shown the importance of uncertainty estimation in classification task.

5 Team Contribution

- Chesakov Daniil: regression task for CO_2 dataset
- Glazov Vsevolod: regression task for Irradiance dataset
- Kovalev Evgeny: classification task

6 GitHub

Here you can find a code to reproduce our experiments:
`https://github.com/blacKitten13/Dropout-for-Model-Uncertainty`

References

[1] Y. Gal, Z. Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *arXiv:1506.02142*, 2016.

[2] C. Keeling, T. Whorf. Atmospheric CO2 concentrations (ppmv) derived from in situ air samples collected at Mauna Loa Observatory, Hawaii. 2004.

[3] J. Lean. Solar Irradiance Reconstruction. *IGBP PAGES/World Data Center for Paleoclimatology*, 2004.

[4] J. Hernandez-Lobato, R. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. *ICML-15*, 2015.

[5] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proc. of the IEEE*, 1998.