Deep Probabilistic Forecasting
with Monte-Carlo Dropout in Neural Networks

Novin Shahroudi
Institute of Computer Science, University of Tartu
novin@ut.ee — github.com/novinsh/master_thesis — @novinsh

Forecasting is an act of predicting the future that comes with a degree of *uncertainty*. That is why a probabilistic expression of the future suits its remote and uncertain nature better than other forms. *Deep Learning* is taking a prevalent role in today's industries. It is being employed in more mission-critical tasks such as *self-driving cars* as well as tasks that may not require a safety measure, yet require some measure of reliability such as in *energy power forecasting*. Taking uncertainty into account in these models is a key enabler to account for safety and reliability in these models. This poster demonstrates some of the experiments performed for my master thesis to perform probabilistic forecasting using a type of *Variational Bayesian Approximation* for neural networks.

Reliable forecasting models play a crucial role in operations of electricity grid. Wind power is expected to provide at least 50% of the electricity production in Estonia and Sweden in the upcoming decade. Therefore, this work was motivated to be applied on Wind Power Forecasting.

Background

Time-series Forecasting

In the *Time series* forecasting, a history of data up to time is given and a prediction should be made for the future for a given number of steps. Data points are dependent on each other and may not be identically distributed. Each step in the forecast referred to as a *lead time*, and the whole forecast steps referred to as *horizon*. Equation (1) denotes the observed values of a time series from the past to the present and the future.

$$y_0, y_1, \ldots, y_t, y_{t+1}, \ldots, y_{t+h} \qquad h \in \mathbb{N}_{>0} \tag{1}$$

Additional features could be incorporated in univariate time series forecasting that it is referred to as exogenous variables.

Bayesian Deep Learning

A *Bayesian neural network (BNN)* [1] considers a distribution over its parameters as well as its outputs as opposed to a generic *Artificial Neural Network (ANN)* which may be referred to as *Non-bayesian Neural Network* as well. Calculating the posterior of a Bayesian model is intractable for neural networks due to humongous parameter space even with relatively small networks. In [3] authors introduce Monte-Carlo Dropout (MCDO) network as an approximation of BNN that gives reasonably good results to approximate a Bayesian Neural Network.

Probabilistic Forecasting

Probabilistic or *Dense Forecast* is the most informative form of forecast that estimates a probability distribution rather a point forecast for each time step. It is common to demonstrate a probabilistic forecast using *fan-charts*, each fan representing a quantile of the probability distribution as depicted in Fig. 5.

Evaluation metrics

CRPS [7] evaluates quality of a probabilistic forecast represented by a predictive cumulative distribution function $F_t(y)$ against the true value or observed probability passed through a *heaviside* function $H(y, \hat{y})$ as shown in Eq. (2). An important property of CRPS is that it measures the accuracy, as well as the sharpness of the predictions which means *calibratedness* of the forecasts is also reflected in the score. When CRPS normalized, it is denoted as NCRPS.

$$NCRPS(y) = \frac{1}{n(\mathbb{T})} \sum_{t \in \mathbb{T}} \int_{-\infty}^{\infty} \left(F_{t+h|t}(y) - \mathbb{H}(y - y_{t+h})\right)^2 \tag{2}$$

Mean Squared Error (MSE) has the same measurement unit as of the squared of the series evaluated for. It is also equivalent to the variance of the model output if the model is unbiased and it is traditionally used for point estimate forecasts.

$$MSE(h) = \frac{1}{n(\mathbb{T})} \sum_{t \in \mathbb{T}} (\hat{y}_{t+h|t} - y_{t+h})^2 \tag{3}$$

Data

For the experiments with Synthetic data, the `TimeSynth` was used to generate and sample from a sinusoidal signal with frequency.

For the experiments with real data, wind power dataset from Global Energy Forecasting Competition 2014 [5] was used. Setup of the data performed according to [4]. Two sets of experiments conducted on this dataset, a univariate forecasting, and univariate forecasting with 4 wind speed features as other predictors.

Baseline Models

Persistent or naive forecast is a well-known time-series forecasting baseline. It uses the last observation as the forecast.

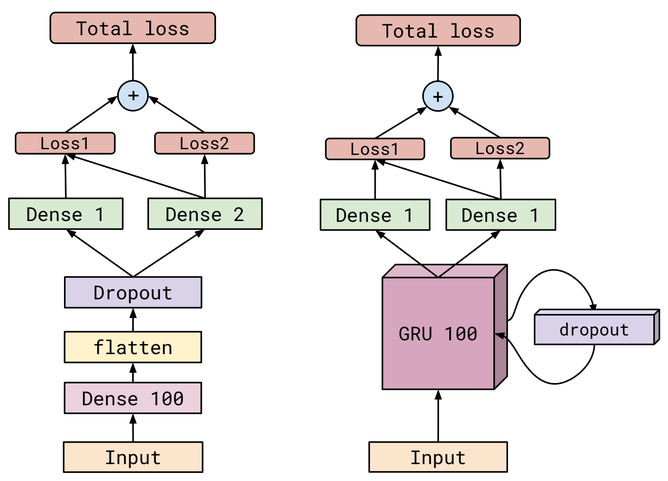
Methods

Probabilistic Baseline

Naive method extended to provide probabilistic baseline where a given quantile of the training data is being used as the forecast.

Models

Methods explained in this section were mainly inspired by [6]. A *Multi-layer Perceptron (MLP)*, and a *Recurrent Neural Network (RNN)* with GRU cells [2] as depicted in Fig. 1 were used. Both models provided two outputs, one estimating the mean and the other the variance of the forecast. 100 units were used with a dropout rate of 0.3 for both models. For the RNN, recurrent dropout was used. The batch size for all experiments set to 32. All models were run for 10 epochs with Adam optimizer and Cyclical Learning Rate Scheduler [8]. Activation function for the first layer set to *linear* for the MLP and *tanh* for the RNN. Activation of the first dense output that estimate the `mean` set to *hard sigmoid*, and the second dense output that estimates the logarithm of the variance set to linear.



Architecture of MLP (left) and RNN (right)

Models were trained to forecast one-step ahead and to achieve multi-step ahead forecast *roll-forward forecasting* was used.

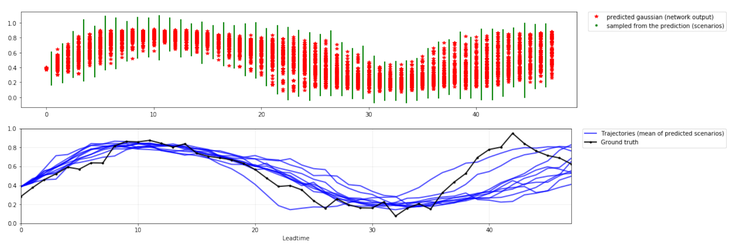
Uncertainty in the data, also referred to as **Aleatoric uncertainty**, is mainly caused by partial observability and stochasticity in the underlying process generating the data, and/or errors in measurements. It is estimated by the second output of the models. A new loss as shown in Eq. 4 is then used. In this way, the network is encouraged not only minimize the mean but also the variance of the predictions.

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sigma(\vec{x_i})^2} \mid\mid \vec{y_i} - f(\vec{x_i}) \mid\mid^2 + log\sigma(\vec{x_i})^2 \tag{4}$$

Uncertainty in the model also referred to as **Epistemic uncertainty**. It can be caused by model misspecification (a wrong parametric family), error in the estimation of the model's parameters when modeling, exposing the model to novel examples and any other systematic error in the model. This can be estimated by Dropout at test time. It produces different possible outputs that can be thought to be produced by different sub-networks which would imply the variance of the model. A neural network can also be thought of as an ensemble method for reducing the variance of prediction by using a number of unbiased models.

Scenario Forecasting

The roll-forward forecasting produces biased results as a result of the model's ignorance caused by treating forecast values as new observations. To tackle this issue scenario forecasting was used which also enabled the model to produce different forecast trajectories as another informative form of forecasting. Since the forecasts were probabilistic, one could sample from the forecast distribution and the sampled values could be used instead as the next value in the input sequence forecasting for all possible alternative futures.



Demonstration of constructing trajectories with scenario forecasting. On the top, possible samples drew from the forecast distribution (denoted by red) in order to perform scenario forecasting on these samples (denoted by blue). On the bottom, generated trajectories as a result of scenario forecasting demonstrated. Scenarios reveal possible multimodality in the data. Each of these scenarios forecasts performed as part of a MCDO simulation and so were averaged over all to obtain the final result as shown Figure 5

Results

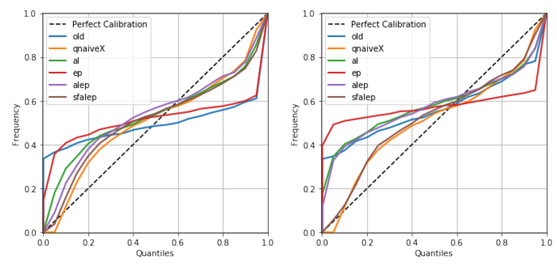
All results obtained by evaluation on 20 different splits of the validation set. Input size for the Synthetic data was 24 and of 4 for the *GEFCom'14* with forecast horizon of size 48. Results of 5 variation of each model per each architecture are being reported here. 4 of which correspond to the Bayesian models and to a non-Bayesian model (`old`). The Bayesian models are denoted as `AL` for Aleatoric, the `EP` for Epistemic, the `ALEP` for Aleatoric+Epistemic, and `SFALEP` for the Scenario Forecast+`ALEP`. Finally, the probabilistic baseline denoted by `QNaive`.

NCRSP results obtained on the GEFCom'14 Dataset

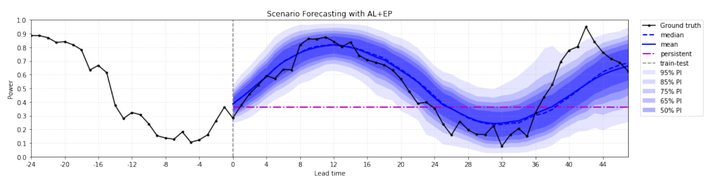
NCRPS	Univariate		with Exogenous Variables	
Architectures	MLP	GRU	MLP	GRU
`QNaive`	0.211 ± 0.103	0.211 ± 0.103	0.209 ± 0.092	0.209 ± 0.092
`AL`	0.215 ± 0.157	0.218 ± 0.155	0.167 ± 0.121	0.169 ± 0.150
`EP`	0.247 ± 0.197	0.263 ± 0.186	0.196 ± 0.154	0.157 ± 0.141
`ALEP`	0.208 ± 0.147	0.217 ± 0.150	$\mathbf{0.166 \pm 0.097}$	0.159 ± 0.137
`SFALEP`	$\mathbf{0.203 \pm 0.149}$	$\mathbf{0.183 \pm 0.097}$	0.173 ± 0.104	$\mathbf{0.144 \pm 0.099}$

Results obtained on the Synthetic Dataset

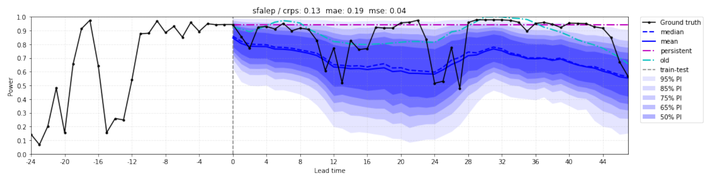
Metrics	CRPS		MSE	
Architectures	MLP	GRU	MLP	GRU
`old`	N/A	N/A	0.033 ± 0.020	0.035 ± 0.021
`QNaive`	0.163 ± 0.055	0.163 ± 0.055	0.082 ± 0.014	0.082 ± 0.014
`AL`	0.116 ± 0.100	0.103 ± 0.105	0.033 ± 0.019	0.030 ± 0.024
`EP`	0.131 ± 0.106	0.132 ± 0.111	0.033 ± 0.019	0.035 ± 0.022
`ALEP`	0.112 ± 0.095	0.103 ± 0.104	0.033 ± 0.019	0.030 ± 0.024
`SFALEP`	0.107 ± 0.086	$\mathbf{0.098 \pm 0.085}$	0.034 ± 0.019	$\mathbf{0.031 \pm 0.024}$



Calibration of models evaluated on GEFCom data with exogenous variables



Forecast with RNN on Synthetic data



Forecast with RNN on GEFCom data with exogenous variables

Conclusions

Successful application of the method proposed in [6] for probabilistic forecasting. A naive baseline proposed by us which helps with diagnosis and it is a good baseline for evaluation of models calibration. The proposed model, `SFALEP`, improved the results both from the perspective of performance, and calibration. RNN was more effective especially in the case of the real dataset with multiple variables involved in the forecasting.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[2] Junyoung Chung, Yoshua Bengio, and et.al. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[3] Yarin Gal and Zoubin Ghahramani. Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1050–1059. JMLR.org, 2016.

[4] Sebastian Haglund and El Gaidi. Bounded Probabilistic Wind Power Forecasting using Mixture Density Recurrent Neural Network. In *Workshop on the Meaning of 42*, 2018.

[5] Tao Hong, Rob J. Hyndman, and et.al. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *International Journal of Forecasting*, 32(3):896 – 913, 2016.

[6] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *CoRR*, abs/1703.0, 2017.

[7] Juan.M Morales, Conejo, and et.al. *Integrating Renewables in Electricity Markets. International Series in Operations Research & Management Science*. Springer, 2011.

[8] Leslie N. Smith. No more pesky learning rate guessing games. *CoRR*, abs/1506.01186, 2015.

Acknowledgements

I would like to gratefully acknowledge my master thesis supervisors Prof. Dr. Meelis Kull of the *Insitute of Computer Science* at *University of Tartu*, Erik Ylip of the *Swedish Institute of Computer Science* at *Research Institutes of Sweden (RISE)*, and Sebastian Haglund El Gaidi at *Greenlytics AB*. I am also grateful for the encouraging advice and support from Sofiya Demchuck.