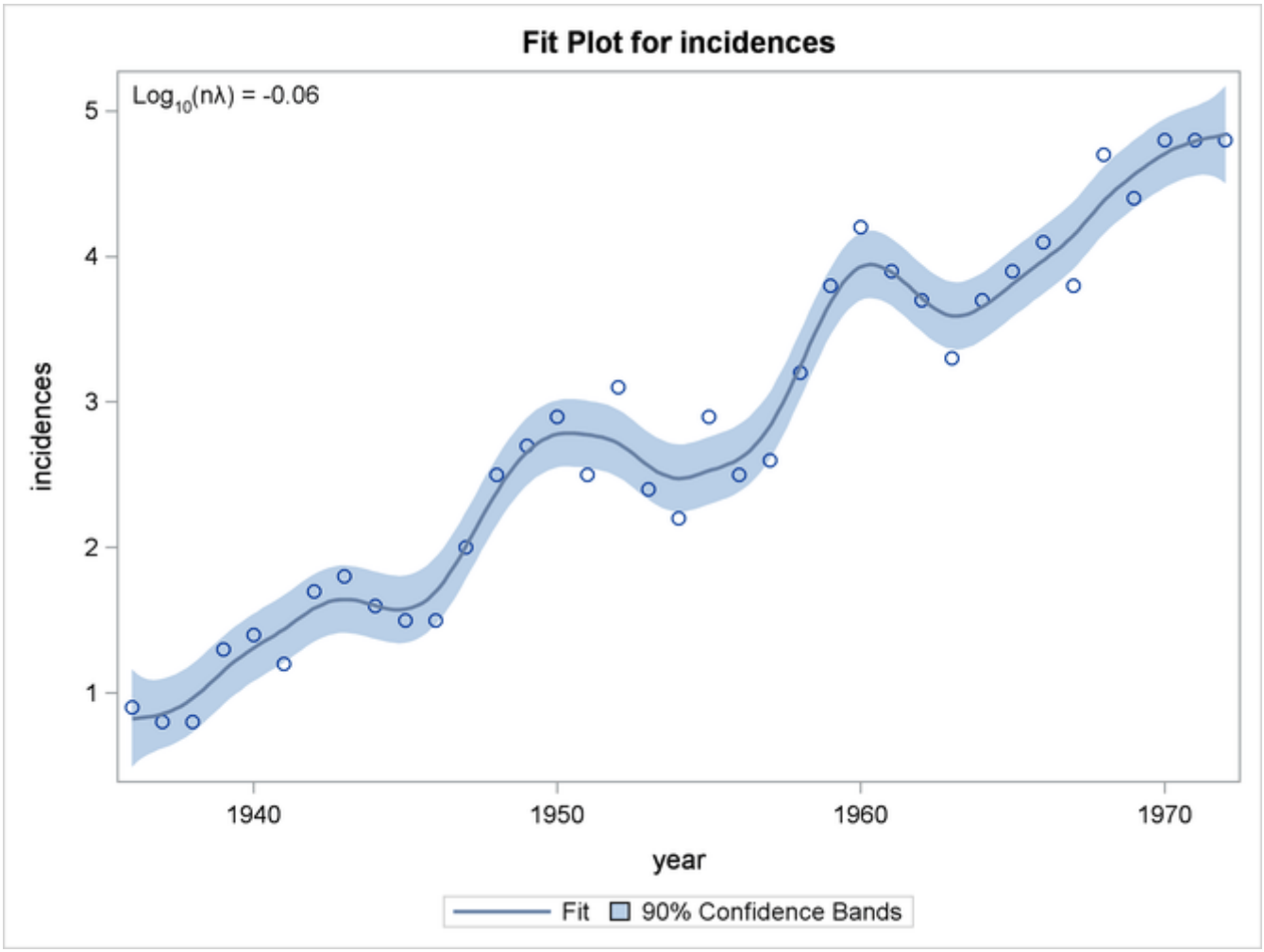
Uncertainty estimation for Neural Network — Dropout as Bayesian Approximation

Nok
Jan 28 · 6 min read ★



> *The key theme of this article is, you can use dropout to create prediction confidence.*

. . .

This article is mainly about how I start with the Uber's paper *Deep and Confident Prediction for Time Series at Uber.* Model interpretation with neural networks has not been an easy task, knowing the confidence of a neural network could be very important for business. Despite a lot of complicated proof in these series of papers, they all trying to answer a simple question

> ***How confident is my model about a particular prediction?***

Table of Content

Background

We are dealing with some forecasting problem for business, so we are researching new methods for forecasting especially new approach to neural networks. (I am aware of the LSTM/encoder-decoder/seq2seq but I have not heard a lot of exciting performance from it). There are not many public discussions about **Time Series** from the big company that I can find, the closest thing I can find is **Prophet** from Facebook, which is a purely statistical base forecasting method.

Uber has a paper ***Deep and Confident Prediction for Time Series at Uber*** *that* draw our attention as they are one of the industry players that put a lot resource on probabilistic programming. After some research, I find that Uber has also won the M4 competition (a well-known time series competition) The M4 competition does not only requires an accurate prediction, but also the confidence interval of it. They have also open source Pyro, a probabilistic programming library, so combining these facts, I have reasonable faith that they are doing some interesting(and practically useful) works.

I do not have a deep statistic background and the word “Bayesian” is almost meaningless to me, all I know is that it relates to conditional probability, that’s it. I try very hard to recall my STAT1000 about Bayesian vs frequentist, this fantastic thread explains it quite well.

Deep and Confident Prediction for Time Series at Uber

Uber has already written a blog post about this work, I start with the 1st paper but end up reading a few more papers (some are earlier groundworks, some are follow up works). I will not spend too many words to explain the papers, as I could not understand every derives steps. Instead, I will only **highlight** the important parts of my research journey and I encourage you to go through the paper.

> *We show that the use of dropout (and its variants) in NNs can be interpreted as a Bayesian approximation of a well known probabilistic model: the Gaussian process (GP)*[1]

I personally do not 100% convinced by the work, but they have shown great practical result and that is the most important part. (Like BatchNorm people have been thinking why it works for a wrong reason for a long time, but that does not stop anyone using it as it **did** improve model convergence)

Studied Papers:

1. *Deep and Confident Prediction for Time Series at Uber*
2. *Time-series Extreme Event Forecasting with Neural Networks at Uber*
3. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*
4. *Variational Bayesian dropout: pitfalls and fixes*
5. *Variational Gaussian Dropout is not Bayesian*
6. Risk versus Uncertainty in Deep Learning: Bayes, Bootstrap and the Dangers of Dropout

Other Resources:

1. *The M4 Competition: Results, findings, conclusion and way forward*
2. *Uncertainty in Deep Learning* (Yarin Gal’s thesis)

It's useful to have a timeline when reading a series of materials.

Paper	Publish Time
Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning	Jun 2015
Time-series Extreme Event Forecasting with Neural Networks at Uber	June 2017
Deep and Confident Prediction for Time Series at Uber	Sep 2017
Variational Gaussian Dropout is not Bayesian	Nov 2017
Risk versus Uncertainty in Deep Learning: Bayes, Bootstrap and the Dangers of Dropout	Nov 2017
M4 Forecasting Competition: Introducing a New Hybrid ES-RNN Model	Jun 2018
Variational Bayesian dropout: pitfalls and fixes	Jul 2018

Timeline

. . .

Uncertainty Estimation

One of the key distinction about Bayesian is that parameters are distributions instead of fixed weights.

> $Error = Model\ Uncertainty + Model\ misspecification + inherent\ noise$

The Bayesian neural network decomposes uncertainty into **model uncertainty**, **model misspecification**, and **inherent noise**.

MCDropout

Algorithm 1: MCdropout

Input: data x^*, encoder $g(\cdot)$, prediction network $h(\cdot)$, dropout probability p, number of iterations B

Output: prediction $\hat{y}^*_{mc}$, uncertainty η_1

for $b = 1$ to B **do**
  $e^*_{(b)} \leftarrow VariationalDropout(g(x^*), p)$
  $z^*_{(b)} \leftarrow \text{Concatenate}(e^*_{(b)}, \text{extFeatures})$
  $\hat{y}^*_{(b)} \leftarrow Dropout\ (h(z^*_{(b)}), p)$
end for

// prediction

// prediction

6: $\hat{y}^*_{mc} \leftarrow \frac{1}{B}\sum_{b=1}^{B} \hat{y}^*_{(b)}$

// model uncertainty and misspecification

7: $\eta_1^2 \leftarrow \frac{1}{B}\sum_{b=1}^{B}(\hat{y}^*_{(b)} - \hat{y}^*)^2$

8: **return** $\hat{y}^*_{mc}, \eta_1$

MCDropout

One of the key here in Bayesian is that everything is a probabilistic distribution but not a point estimate. This means there are uncertainties about your weight as well.

They use **MCDropout** to deal with model uncertainty and misspecification. Basically, they have claimed that using Dropout at **inference** time is equivalent to doing Bayesian approximation. The key idea here is letting dropout doing the same thing in **both training and testing time.** At test time, you will repeat B times (Few hundreds of times as the paper said), i.e. passing the same input to the network with random dropout. You then take means of your prediction and you can generate a prediction interval with these # of B predictions. MC is referring to Monte Carlo as the dropout process is similar to sampling the neurons.

Inherent Noise

Algorithm 2: Inference

Input: data x^*, encoder $g(\cdot)$, prediction network $h(\cdot)$, dropout probability p, number of iterations B

Output: prediction $\hat{y}^*$, predictive uncertainty η

// prediction, model uncertainty and misspecification

1: $\hat{y}^*, \eta_1 \leftarrow$ *MCdropout* (x^*, g, h, p, B)

// Inherent noise

2: **for** x'_v **in** validation set $\{x'_1, ..., x'_V\}$ **do**

3: $\hat{y'}_v \leftarrow h(g(x'_v))$

4: **end for**

5: $\eta_2^2 \leftarrow \frac{1}{V}\sum_{v=1}^{V}\left(\hat{y'}_v - y'_v\right)^2$

// total prediction uncertainty

6: $\eta \leftarrow \sqrt{\eta_1^2 + \eta_2^2}$

7: **return** $\hat{y}^*, \eta$

Inherent Noise

They also introduce the term **Inherent Noise** which refer to noise that is irreducible. In short, they use a very common technique to model this error — held-out validation. They call this an adaptive approach and talk about smoothness and prior, but I don't see any difference from the standard train/validation practice that the ML community is familiar with. In the end, you will combine the two error terms and get the final uncertainty term.

Discussion

You can find an interesting discussion on Reddit, which provide some counterargument from a theoretical standpoint. In fact, I am not 100% convinced by Uber's paper. However, they have shown good result in both internally and M4 competition. Like a lot of advance in deep learning, theory come later than practical results. If you are interested, feel free to try it out, the implementation should be relatively easy as you just need to keep dropout at inference time.

Conclusion

The takeaway is, uncertainty exists not only in your model, but your weight as well. Bayesian Neural Network tries to model the weights as distributions.

MCDropout offer a new and handy way to estimate uncertainty with minimal changes in most existing networks. In the simplest case, you just need to keep your dropout on at test time, then pass the data multiple times and store all the predictions. The downside is, this could be computationally expensive, although Uber claims this adds less than ten milliseconds. They didn't discuss how they achieve this, but my guess is they do heavily parallel computation as you can imagine the multiple passes of data does not come in sequential order, so this process can be parallel easily.

I am very eager to hear more discussion about the latest time series forecast and method estimating uncertainty with NN, let me know if you know some better way!

Appendix

Where to start? Key Name to google at:

- Yarin Gal (He propose using dropout for Bayesian approximation in his thesis)

- Slawek Smyl (winner of M4 competition)

If you want to know about application only. Have a quick look at [2] then [1], as [1] is the groundwork for [1], some of the background is duplicate.

[4] is a great overview of some of the pitfalls of using dropout

1. *Deep and Confident Prediction for Time Series at Uber*
2. *Time-series Extreme Event Forecasting with Neural Networks at Uber*
3. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*
4. *Variational Bayesian dropout: pitfalls and fixes*
5. *Variational Gaussian Dropout is not Bayesian*
6. Risk versus Uncertainty in Deep Learning: Bayes, Bootstrap and the Dangers of Dropout

https://tensorchiefs.github.io/bbs/files/dropouts-brownbag.pdf (Yarin Gal)

Machine Learning Neural Networks Artificial Intelligence Bayesian Machine Learning Timeseries