# *Supplementary material for presentation*[1]

*Jialin Lu*

*26 Feb 2020*

[1] This handout contains the summary and references for the presentation on the lab meeting of Martin Ester's group, at Simon Fraser University, Feb 26 2020

I view this presentation as a short survey into the topic of **Bayesian Deep Learning**.

- First I give the motivations of combing Bayesian methods for deep learning.
- Then in Part 2 introduce the main theme of approximating the posterior distribution of NN's parameter and talk about two technical approaches for tackling it.
- In part 3 I show, by referring a simple experiment, that the BDL thing is a little bit frustrating in practice and does not really work yet, compared with a simple baseline of ensemble.
- I will then end with some more personal opinions.

## *Part 1: Promises of combining Bayesian Learning (BL) and Deep Learning (DL)*

Combing the best from both worlds seems to be promising, but the problem is how to do so. Mainly categorized into two ways:

**1. DL↦BL**: use a DNN to model an edge function in probabilistic graphical models, example: Variational Autoencoders [Kingma and Welling, 2013].

**2. BL↦DL**: Bayesian Treatment of neural networks. This is what *Bayesian Deep Learning* or *Bayesian Neural Network* is dealing with.

Here we only talk about the **latter**. Of course this idea itself is not really new [MacKay, 1992, Neal, 1996], but current state of DL research certainly is a different situation compared to NN research in 90s, that we start to treat overparameterization as a serious issue.

BL and DL are good, but in different aspects:

- Bayesian Learning is such a flexible framework, we can use to model many things, even the most flexible human learning. Bulks of works on computational modelling of cognition and psychology are now Bayesian, one example is the PhD thesis of [Tenenbaum, 1999].
- Deep learning is efficient and scalable using Batch-based backprop gradient update. Flexible design of architecture further makes it applicable in various applications.

## *Part 2: Bayesian Treatment of Deep Learning*

In Bayesian learning, we do not just obtain one single model $\theta = argmax\, P(\theta|D)$, but a distribution of models, i.e., the posterior distribution of parameters $P(\theta|D)$. When we have $P(\theta|D)$ we can do a lot of things, model comparison, uncertainty [Kendall and Gal, 2017] but the technical problem is how to obtain it. There are two options. [2]

[2] We view these two options as two directions that try to approximate $P(\theta|D)$ from the weight space or the function space.

[3] Further see the NeurIPS 2019 tutorial [Osawa et al., 2019].

Option 1: Approximate the Posterior using Variational Inference [3] This approach requires to first predefine the form of the posterior, and then make it a parameterized distribution $q$. We want $q$ to approximate $P(\theta|D)$. For example, we say that $q$ is a Gaussian with two parameters: mean and variance. Then we can adjust the mean and variance to approximate the true posterior.

- By choosing the right form of $q$, you can derive standard deep learning optimization algorithms like SGD, RMSprop and Adam.
- Go beyond basic choices, you get more complicated algorithms [Osawa et al., 2019].[4]

[4] Sad thing, the performance is not really great because it is hard. Making it work nearly as good as Adam is already very impressive.

The above mentioned approach, viewed as variational inference, will only work if the true posterior can really be described by the parameterized distribution, the choice of $q$. We can of course make more complicated choices, for example, to use mixtures as the posterior [Lin et al., 2019][5].

[5] But will this be practical?

Option 2: Approximate the Posterior using Interpolation[6] However you define the form of posterior, it is not flexible and will never to close to the true posterior. The alternative approach is to use interpolation-based approximation of the posterior, i.e. collect and store multiple set of parameters during the trajectory of optimization, and use these to interpolate, in order to approximate the true posterior. The term 'interpolation' is my own understanding and thus it might not be so precise, can further see the paper [Garipov et al., 2018, Izmailov et al., 2018, Maddox et al., 2019, Izmailov et al., 2019].

[6] It actually works better. You just need to collect and store points during an optimization trajectory.

And yes, it makes a lot more sense to make it interpolate with multiple trajectories [Wilson and Izmailov, 2020], not single one. But it will take linearly more time...

## *Part 3: A simple baseline and What goes wrong?*

All these methods mentioned above, actually have a very simple baseline called **Deep Ensemble**, basically, with different random initialization, train multiple models and then call this pool of models the samples from the true posterior distribution. This is so simple and should not be so great, either the accuracy or the calibration. However, Deep Ensemble is really good. (Further details see [Snoek et al., 2019] ) [7] The deep ensemble paper [Lakshminarayanan et al., 2017] is from 2017 but I guess it is still the state of art.

[7] this recent paper (also in NeurIPS 2019) mainly talks about ensemble is good in terms of the metrics of both accuracy and uncertainty qualification.

## *Part 4: A New Hope*

Consider Prior on Architecture. Stop thinking only about the distribution on weights, also start considering the architecture of a DNN. The architecture should bring us more diversity of model. And going back to the old days, it should be possible to obtain the posterior, not only over the weights, but also over the possible architectures.

I think this is theoretically interesting If we focus on improving the Bayesian Deep Learning research. But perhaps we are not really interested in doing it.

Think about the usage of $P(\theta|D)$ and just use ensemble. Sometimes we are not actually interested in $P(\theta|D)$ itself, but rather the things we can do with the posterior I believe this should have some impact on transfer or meta-learning, which are the exact case where we need posterior of parameters.

This is more practical to our lab members, as we do not really care about the art of Bayesian Deep Learning, we are only motivated by the advantages. Some of the lab members focus on transfer and meta learning.

Suppose we have a posterior, and we might be able to determine when it can transfer useful information, and when it transfers non-sense. There are really some interesting things we can do with a posterior (approximated using ensemble). In fact, there is an interesting paper on how we can use uncertainty to detect at test time, whether the test input is out-of-distribution [Madras et al., 2019]. [8]

[8] Note that It makes a post-hoc ensemble to approximate the posterior

At last, I would like to name two really important reference, extremely thoughtful and clear-written. **The Case for Bayesian Deep Learning** [Wilson, 2020] and **Bayesian Deep Learning and a Probabilistic Perspective of Generalization** [Wilson and Izmailov, 2020]. The former is a dicussion paper, the latter a slightly more technical one.

## *References*


Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, pages 8789–8798, 2018.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Pavel Izmailov, Wesley J Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace inference for bayesian deep learning. *arXiv preprint arXiv:1907.07504*, 2019.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

Wu Lin, Mohammad Emtiyaz Khan, and Mark Schmidt. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. *arXiv preprint arXiv:1906.02914*, 2019.

David JC MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13132–13143, 2019.

David Madras, James Atwood, and Alex D'Amour. Detecting extrapolation with local ensembles. *arXiv preprint arXiv:1910.09573*, 2019.

Radford M Neal. Bayesian learning for neural networks. 1996.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with bayesian principles. In *Advances in Neural Information Processing Systems*, pages 4289–4301, 2019.

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13969–13980, 2019.

Joshua Brett Tenenbaum. *A Bayesian framework for concept learning*. PhD thesis, Massachusetts Institute of Technology, 1999.

Andrew Gordon Wilson. The case for bayesian deep learning. *arXiv preprint arXiv:2001.10995*, 2020.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.