Fast Computation of Uncertainty in Deep Learning

Mohammad Emtiyaz Khan
RIKEN Center for AI Project, Tokyo, Japan
https://emtiyaz.github.io

Joint work with
Wu Lin (UBC), Didrik Nielsen (RIKEN), Voot Tangkaratt (RIKEN)
Yarin Gal (University of Oxford), Akash Srivastava (University of Edinburgh)
Zuozhu Liu (SUTD, Singapore)
The Goal of My Research

“To understand the fundamental principles of learning from data and use them to develop algorithms that can learn like living beings.”
Learning by exploring at the age of 6 months
Converged at the age of 12 months
Transfer Learning at 14 months
The Goal of My Research

“To understand the fundamental principles of learning from data and use them to develop algorithms that can learn like living beings.”
Uncertainty in Deep Learning

To estimate the confidence in the predictions of a deep-learning system
Example: Which is a Better Fit?

Real data from Tohoku (Japan). Example taken from Nate Silver’s book “The signal and noise”
Example: Which is a Better Fit?

When the data is scarce and noisy, e.g., in medicine, and robotics.
Uncertainty for Image Segmentation

The main contributions of this work are:

- We study the trade-offs between modeling aleatoric or epistemic uncertainty by characterizing the process of uncertainty propagation from input data to aleatoric uncertainty and composing these together with epistemic uncertainty approximations. We derive our framework for both regression and classification applications.

- We capture an accurate understanding of aleatoric and epistemic uncertainties, in particular, the effect of noisy data with the implied attenuation obtained from explicitly representing heteroscedastic uncertainty.

- We show how modeling aleatoric uncertainty alone comes at a cost. Out-of-data examples, which can be explained away given enough data, and is often referred to as heteroscedastic uncertainty.

- Heteroscedastic uncertainty depends on the number of training samples, and we observe that the uncertainty can further be categorized into aleatoric and epistemic uncertainty.

- Aleatoric uncertainty captures noise inherent in the observations. In (d) our model exhibits increased aleatoric uncertainty.

- Epistemic uncertainty accounts for our ignorance about which model generated our collected data. This is a notably heterogeneous process, and in (e) our model exhibits increased epistemic uncertainty.

- Model failure can be explained away with the large amounts of data often available in machine vision. We further show that semantic segmentation exhibits increased aleatoric uncertainty on object boundaries and for objects far from the camera.

- For the CamVid dataset, we observe that the segmentation model fails to segment the footpath due to increased epistemic uncertainty, but not aleatoric uncertainty.

- We illustrate the difference between aleatoric and epistemic uncertainty with a novel approach for classification, and visually challenging pixels.

- The bottom row shows a failure case of the segmentation model when the model fails to segment the footpath due to increased epistemic uncertainty, but not aleatoric uncertainty.

- We present results for per-pixel depth regression and semantic segmentation tasks (see Figure 1).

- We show how modeling aleatoric uncertainty in regression and classification can be used to learn loss attenuation, and develop a complimentary approach for the classification task.

- This demonstrates the efficacy of our approach on difficult and large scale tasks.

(taken from Kendall et al. 2017)
Outline of the Talk

• Uncertainty is important
  – E.g., when data are scarce, missing, unreliable etc.

• Uncertainty computation is difficult
  – Due to large model and data used in deep learning

• This talk: fast computation of uncertainty
  – Ideas from Bayesian Inference, Optimization, information geometry
  – Methods that are extremely easy to implement
Uncertainty in Deep Learning

Why is it difficult to estimate it?
A Naïve Method

\[ p(\mathcal{D} | \theta) = \prod_{i=1}^{N} p(y_i | f_\theta(x_i)) \]

- **Data** → Parameters
- **Input** → Neural network
- **Output** → Generate Prior distribution

\[ \theta \sim p(\theta) \]

- **Draws from a distribution**
- **Variance of the draws**
- **Mean of the draws** (Uncertainty)

**Magnitude of Earthquake**

**Frequency**

13
Bayesian Inference

Bayes’ rule:

\[ p(\theta|\mathcal{D}) = \frac{p(\mathcal{D} | \theta) p(\theta)}{\int p(\mathcal{D} | \theta) p(\theta) d\theta} \]

Posterior distribution

Intractable integral

Narrow

Wide
Variational Inference with Gradients

\[ p(\theta|\mathcal{D}) \approx q(\theta) = \mathcal{N}(\theta|\mu, \sigma^2) \]

\[
\max \mathcal{L}(\mu, \sigma^2) := \mathbb{E}_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \sum_{i=1}^{N} \mathbb{E}_q[\log p(D_i|\theta)]
\]

Regularizer \hspace{1cm} Data-fit term

\[
\mu \leftarrow \mu + \rho \nabla_{\mu} \mathcal{L} \\
\sigma \leftarrow \sigma + \rho \nabla_{\sigma} \mathcal{L}
\]

Bayes by Backprop (Blundell et al. 2015), Practical VI (Graves et al. 2011), Black-box VI (Rangnathan et al. 2014) etc.

Our contribution: Using natural-gradients leads to faster and simpler algorithm than gradients methods

- Khan & Lin (Alstats 2017), Khan et al. (ICML 2018), Khan & Nielsen (ISITA2018)
VI using Natural-Gradient Descent

Gradient Descent: \[ \lambda \leftarrow \lambda + \rho \nabla_{\lambda} \mathcal{L}(\lambda) \]

Natural-Gradient Descent: \[ \lambda \leftarrow \lambda + \rho F(\lambda)^{-1} \nabla_{\lambda} \mathcal{L}(\lambda) \]

Fisher Information Matrix (FIM)

\[ F(\lambda) := \mathbb{E}_{q_{\lambda}} \left[ \nabla \log q_{\lambda}(w) \nabla \log q_{\lambda}(w)^{\top} \right] \]
Euclidean Distance is inappropriate!

Two Gaussians with mean 1 and 10 respectively and variances equal to $\sigma_1$ have Euclidean distance = 10

Same as the top row but with the variance $\sigma_2 > \sigma_1$ but still Euclidean distance = 10

Natural-gradient vs gradients

Natural-Gradient VI

\[
\mu \leftarrow \mu - \beta \sigma^2 \nabla_\mu \mathcal{L}
\]

\[
\frac{1}{\sigma^2} \leftarrow \frac{1}{\sigma^2} + 2\beta \nabla_{\sigma^2} \mathcal{L}
\]

(Graves et al. 2011, Blundell et al. 2015)

Gradient-based VI

\[
\mu \leftarrow \mu + \alpha \frac{\hat{\nabla}_\mu \mathcal{L}}{\sqrt{s_\mu} + \delta}
\]

\[
\sigma \leftarrow \sigma + \alpha \frac{\hat{\nabla}_\sigma \mathcal{L}}{\sqrt{s_\sigma} + \delta}
\]

This type of update can be derived when q is an ExpFamily. It is also a generalization of methods such as, Kalman filtering, Sum-product, etc., Variational Message Passing (Winn and Bishop 2005), Stochastic variational inference (Hoffman et al. 2013). See Khan and Nielsen, 2018 for a summary.
Fast Computation of (Approximate) Uncertainty

Approximate by a Gaussian distribution, and find it by “perturbing” the parameters during backpropagation.
Fast Computation of Uncertainty

\[ \prod_{i=1}^{N} p(y_i | f_{\theta}(x_i)) \quad \theta \sim \mathcal{N}(\theta|0, I) \]

Adaptive learning-rate method (e.g., Adam)

0. Sample \( \epsilon \) from a standard normal distribution

\[ \theta_{\text{temp}} \leftarrow \theta + \epsilon \times \sqrt{N \times \text{scale} + 1} \]

1. Select a minibatch
2. Compute gradient using backpropagation
3. Compute a scale vector to adapt the learning rate
4. Take a gradient step

Mean

\[ \theta \leftarrow \theta + \text{learning rate} \times \frac{\text{gradient} \times \theta/N}{\sqrt{\text{scale} + 10/N^8}} \]

Variance

\[ N \sum_{i=1}^{N} p(y_i | f_{\theta}(x_i)) \]
Illustration: Classification

Logistic regression (30 data points, 2 dimensional input). Sampled from Gaussian mixture with 2 components.
Adam vs Vadam

For both algorithms,
Minibatch of 5
Learning_rate = 0.01
Prior precision = 0.01
Why does this work?

• This algorithm is obtained by replacing “gradients” by “natural gradients” (using information geometry)
  – See our ICML 2018 paper.
  – The scaling in natural gradient is related to the scaling in Newton method.
  – Our method is a more principled approach than the Bayesian dropout (Gal and Ghahramani, 2016).
  – Some caveats: Choose small minibatches, better results are obtained with VOGN.
Faster, Simpler, and More Robust

Regression on Australian-Scale dataset using deep neural nets for various number of minibatch size.
Faster, Simpler, and More Robust

Results on MNIST digit classification (for various values of Gaussian prior precision parameter $\lambda$)

![Graph showing comparison between Existing Method (BBVI) and Our method (Vadam)]
Parameter-Space Noise for Deep RL

On OpenAI Gym Cheetah with DDPG with DNN with [400,300] ReLU

- Vadam (noise using natural-grads) - Reward 5264
- SGD (noise using standard gradients) - Reward 2038
- SGD (no noise)

Ruckstiesh et.al. 2010, Fortunato et.al. 2017, Plapper et.al. 2017
Reduce Overfitting with Vadam

Vadam shows consistent train-test performance, while Adam overfits when $N$ is small.

BNN classification on a1a - a9a datasets.
Avoiding Local Minima

An example taken from Casella and Robert’s book.

Vadam reaches the flat minima, but GD gets stuck at a local minima.

Optimization by smoothing, Gaussian homotopy/blurring etc., Entropy SGLD etc.
Summary

• Uncertainty is important, especially when the data is scarce, missing, unreliable etc.
• We can obtain uncertainty cheaply with very little effort
  – Bayesian deep learning
• It works reasonably well on our benchmarks.
Open Questions

• Extensions to other types of distributions
• Quality and usefulness of uncertainty
  – Multiple local minima make it difficult to establish
• Estimating various types of uncertainty
  – Model uncertainty vs data uncertainty
  – Applications play a big role here
• Application to active learning, reinforcement learning, continual learning
Available at https://emtiyaz.github.io/publications.html


Fast yet Simple Natural-Gradient Descent for Variational Inference in Complex Models, INVITED PAPER AT (ISITA 2018) M.E. Khan and D. Nielsen, [ Pre-print ]


Thanks!

Slides, papers, and code available at https://emtiyaz.github.io
We are looking for post-docs, RAs, and interns
Bayesian Inference for Classification

Sampled decision boundaries

Samples from the posterior

Map Estimate
RMSprop vs Vprop