Fast Computation of Uncertainty in Deep Learning

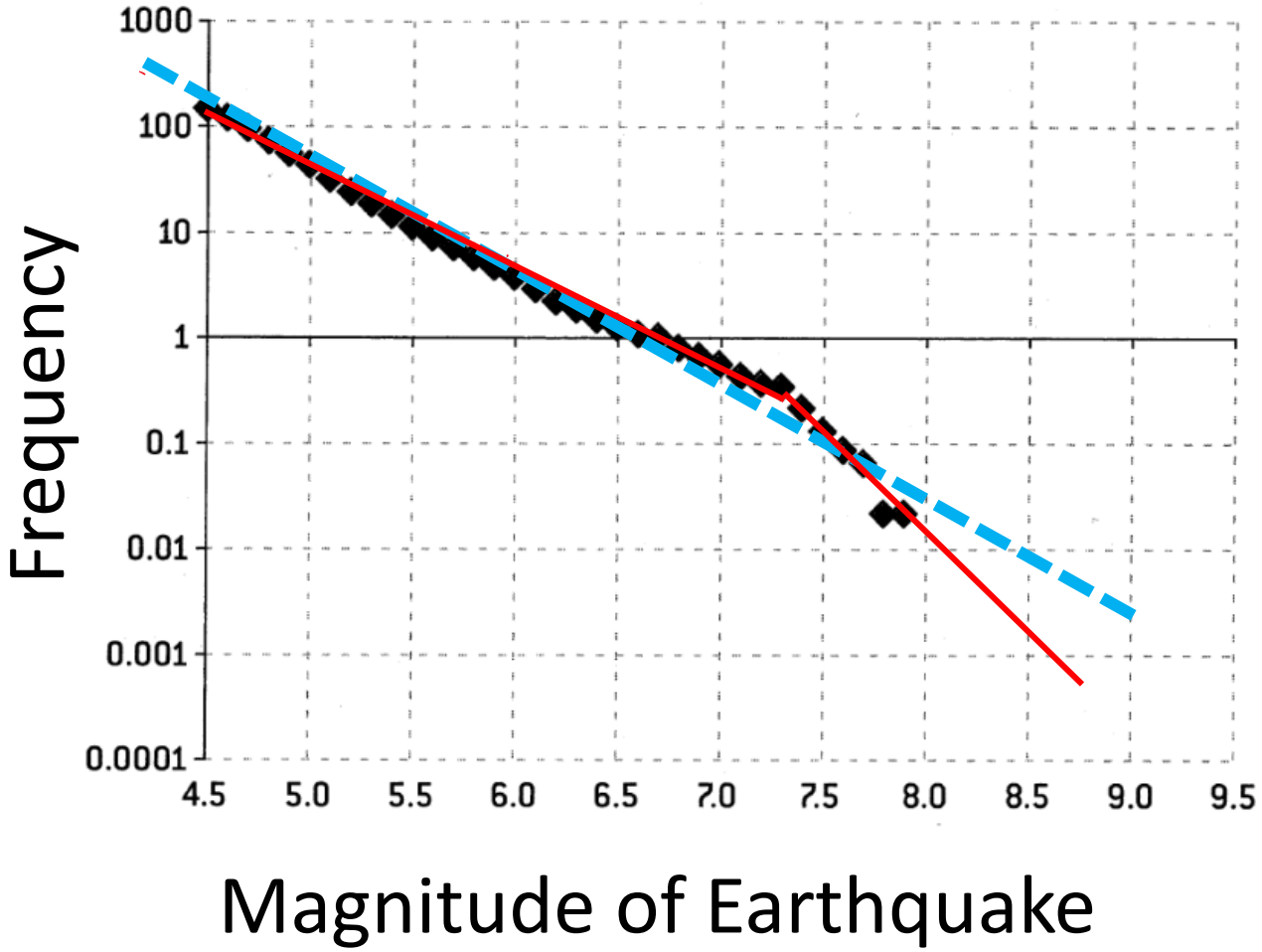
Mohammad Emtiyaz Khan
RIKEN Center for AI Project, Tokyo, Japan

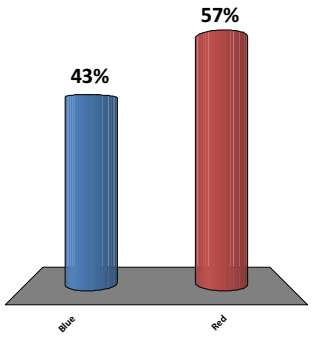
Joint work with
Wu Lin (UBC), Didrik Nielsen (RIKEN), Voot Tangkaratt (RIKEN)
Yarin Gal (University of Oxford), Akash Srivastava (University of Edinburgh)
Zuozhu Liu (SUTD, Singapore)


Uncertainty

Quantifies the confidence in the prediction of a model, i.e., how much it does not know.

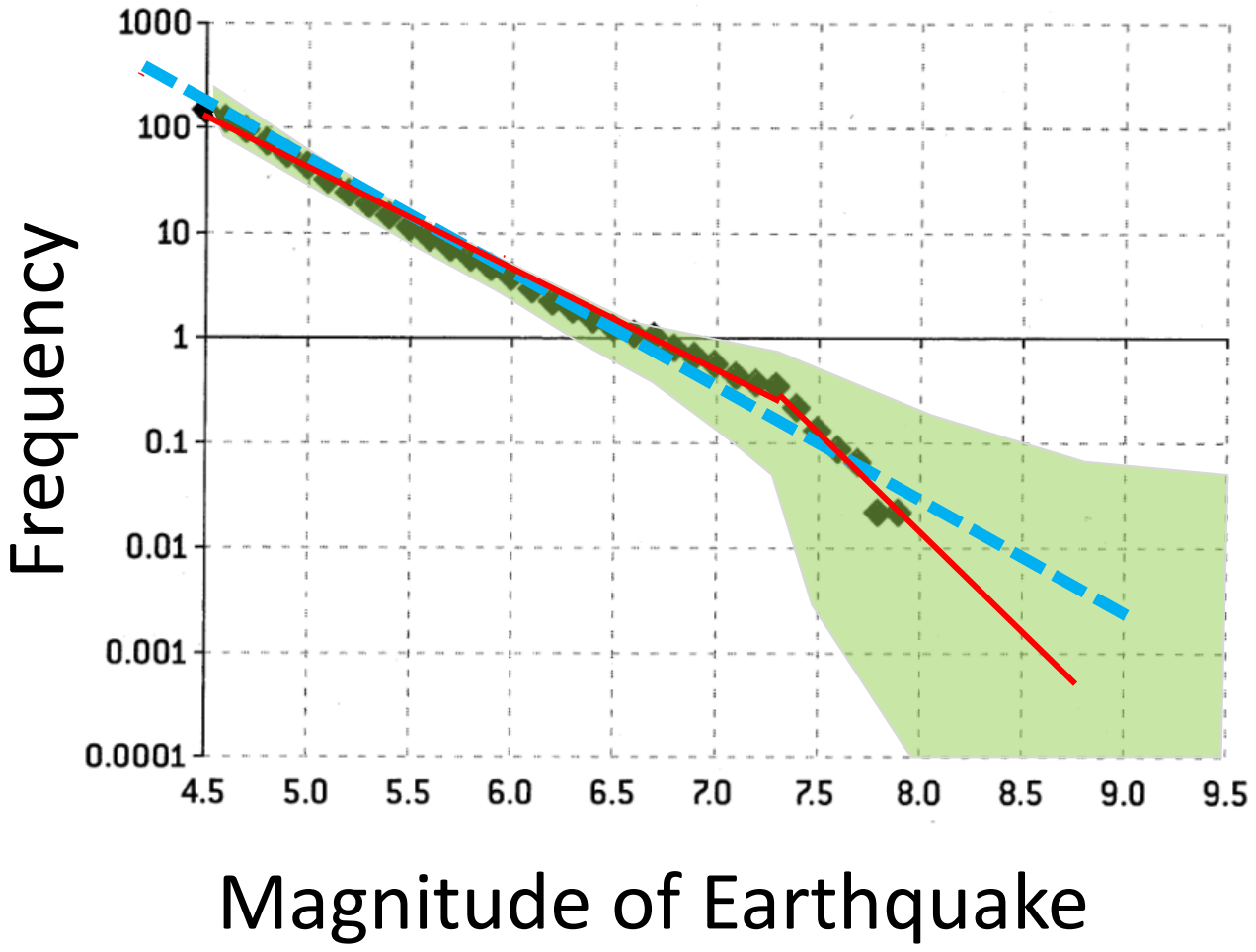
Example: Which is a Better Fit?





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.

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
  - Bayesian deep learning
  - Methods that are extremely easy to implement

Uncertainty in Deep Learning

Why is it difficult to estimate it?

A Naïve Method

Data

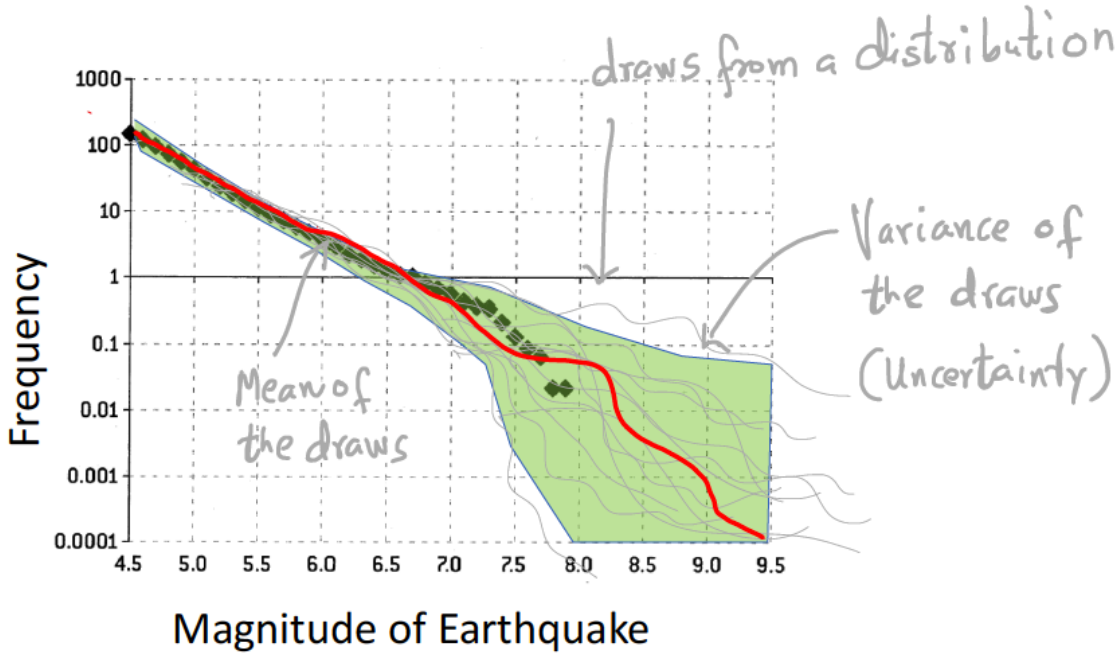
Output

Input

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

Parameters

Neural network



Generate

$$\theta \sim p(\theta)$$

Prior distribution

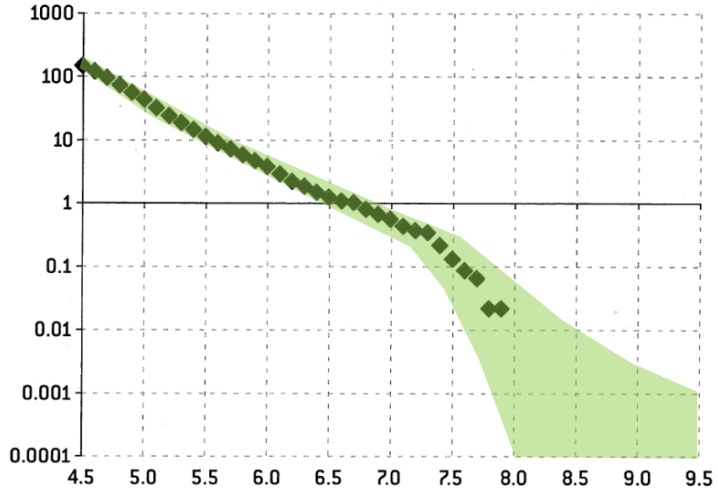
Bayesian Inference

Bayes' rule : $$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$$

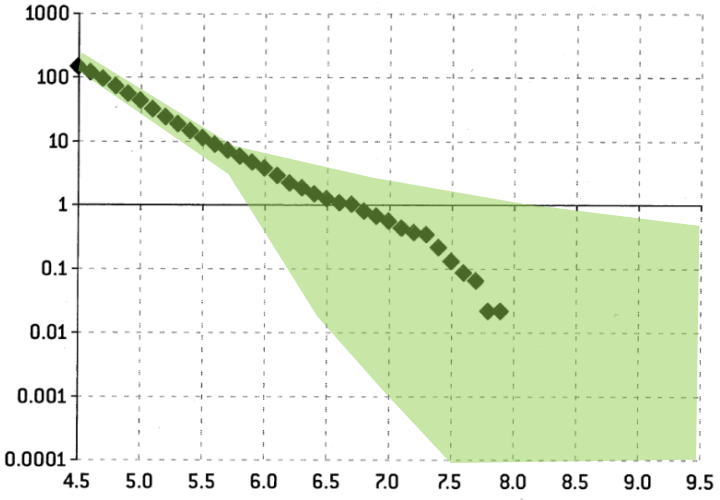
Posterior distribution

Intractable integral

Narrow



Wide



Approximate Bayesian Inference

Variational Inference: Approximate the posterior by a Gaussian distribution

$$\min_{\mu,\sigma^2} D\left[\mathcal{N}(\theta|\mu,\sigma^2)\|p(\theta|\mathcal{D})\right]$$

Mean Variance

Optimize using gradient methods (SGD/Adam)

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

Computation and memory intensive, and require substantial implementation effort

Fast Computation of (Approximate) Uncertainty

Approximate by a Gaussian distribution, and find it by “perturbing” the parameters during backpropagation

Fast Computation of Uncertainty

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

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

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

$$\theta \leftarrow \theta + \text{learning\_rate} * \frac{\text{gradient}}{\sqrt{\text{scale}} + 10^{-8}}$$

Fast Computation of Uncertainty

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

Variational Adam (Vadam)

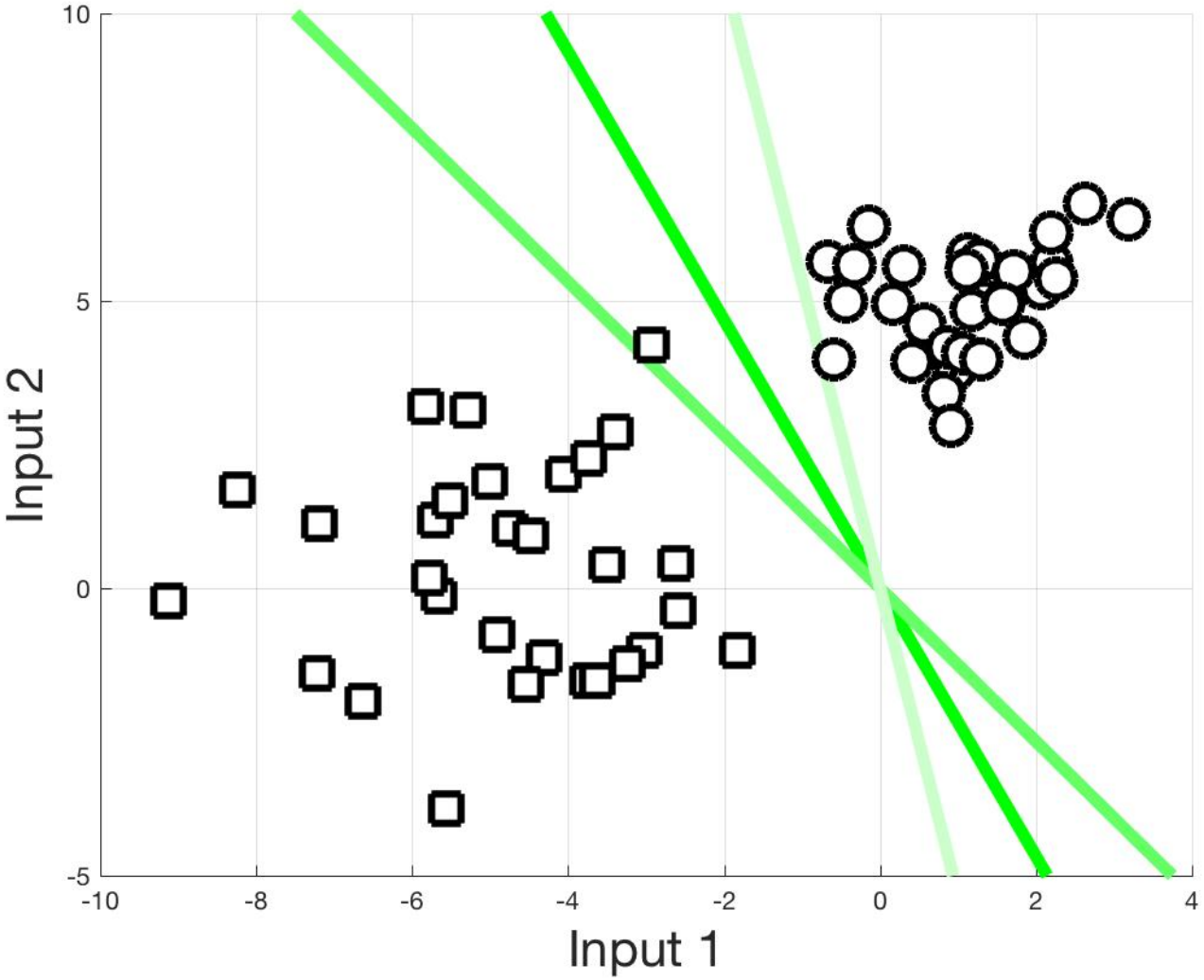
0. Sample ϵ from a standard normal distribution

$$\theta_{\text{temp}} \leftarrow \theta + \epsilon * \sqrt{N * \text{scale} + 1}$$

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

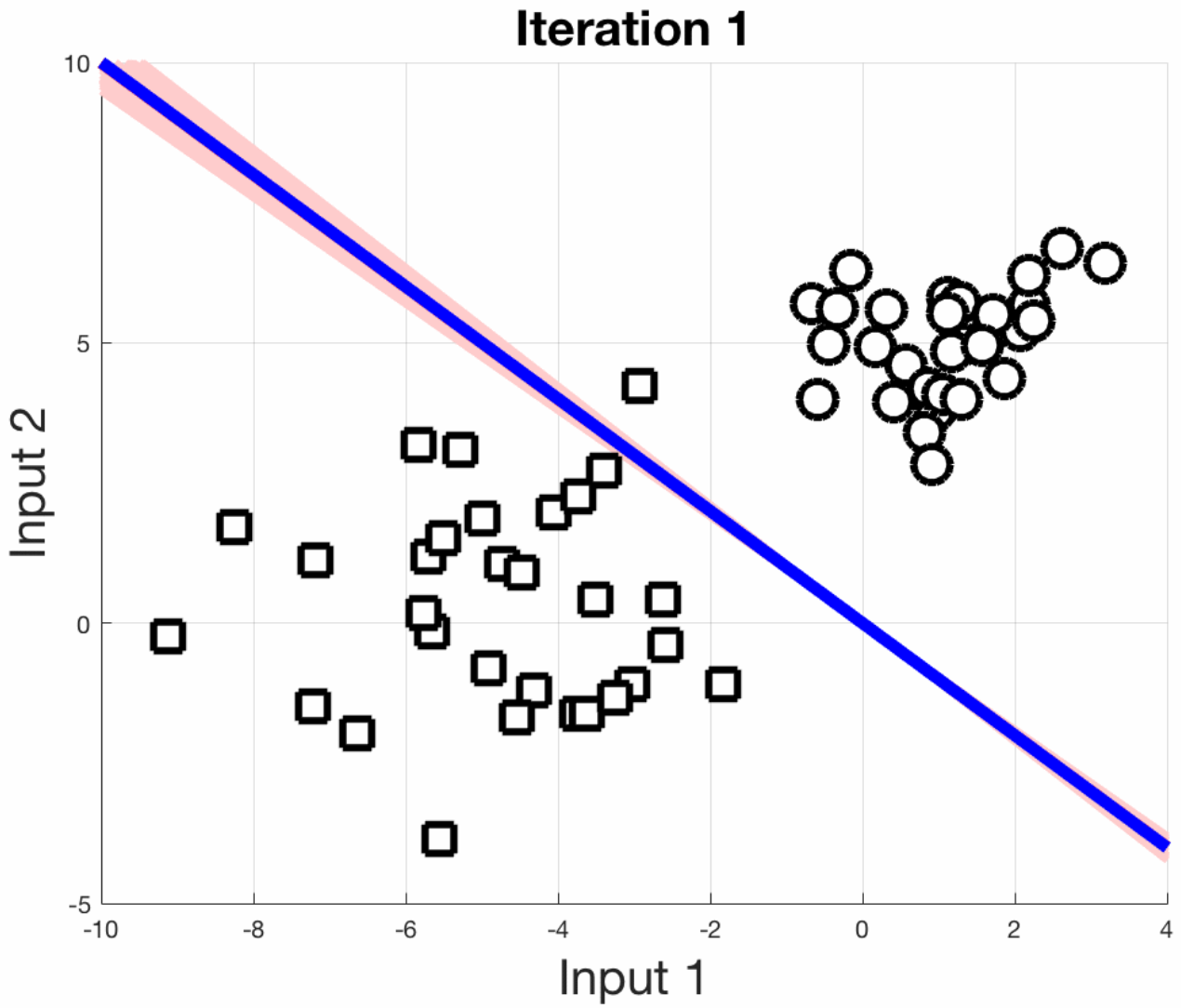
$$\theta \leftarrow \theta + \text{learning\_rate} * \frac{\text{gradient} + \theta/N}{\sqrt{\text{scale}} + 1/N}$$

Illustration: Classification



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

Adam vs Vadam



Adam

Vadam (mean)

Vadam (samples)

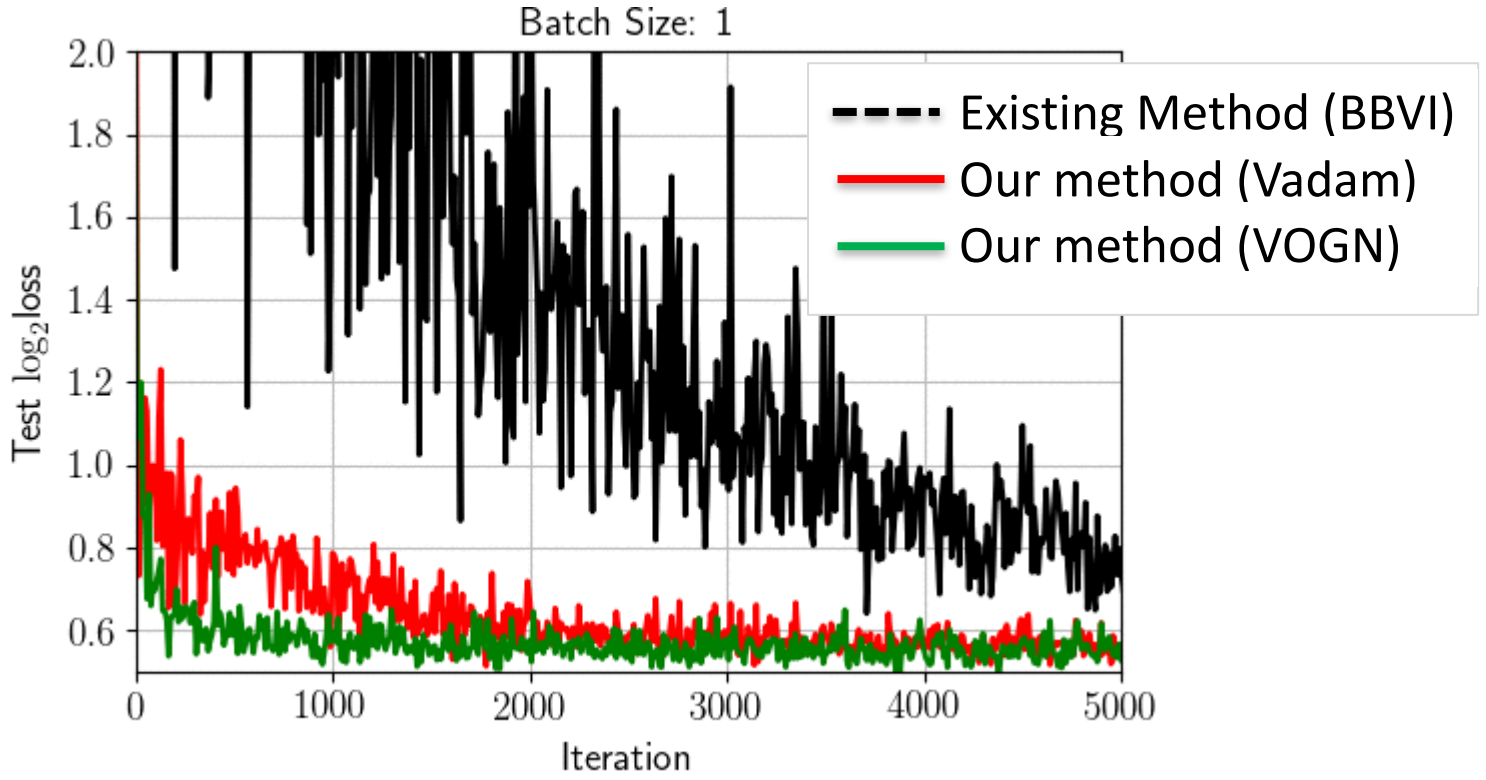
For both algorithms,
Minibatch of 5
Learning_rate = 0.01
Prior precision = 0.01

Why does this work?

- This algorithm is obtained by replacing "gradients" by "natural gradients".
  - See our ICML 2018 paper.
- The scaling in natural gradient is related to the scaling in Newton method.
- An approximation to the Hessian results in Adam.
- Some caveats: Choose small minibatches, better results are obtained with VOGN.

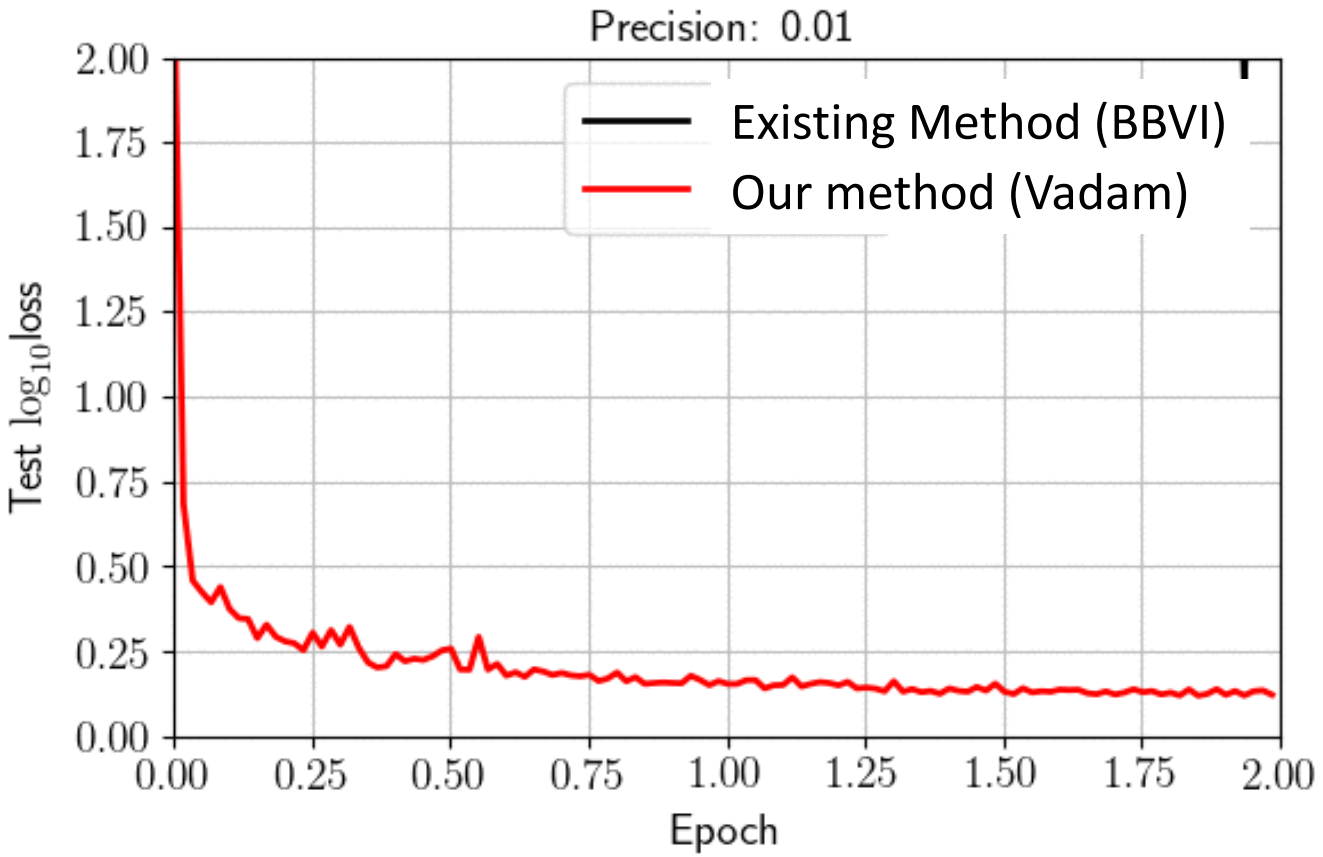
Faster, Simpler, and More Robust

Regression on Australian-Scale dataset using deep neural nets for various number of minibatch size.



Faster, Simpler, and More Robust

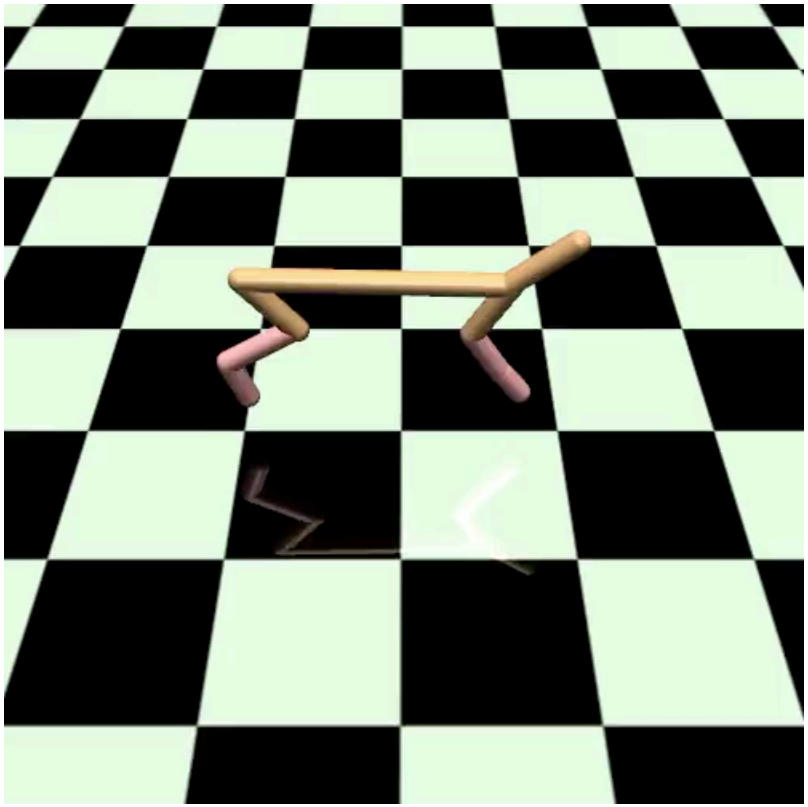
Results on MNIST digit classification (for various values of Gaussian prior precision parameter λ)



Deep Reinforcement Learning

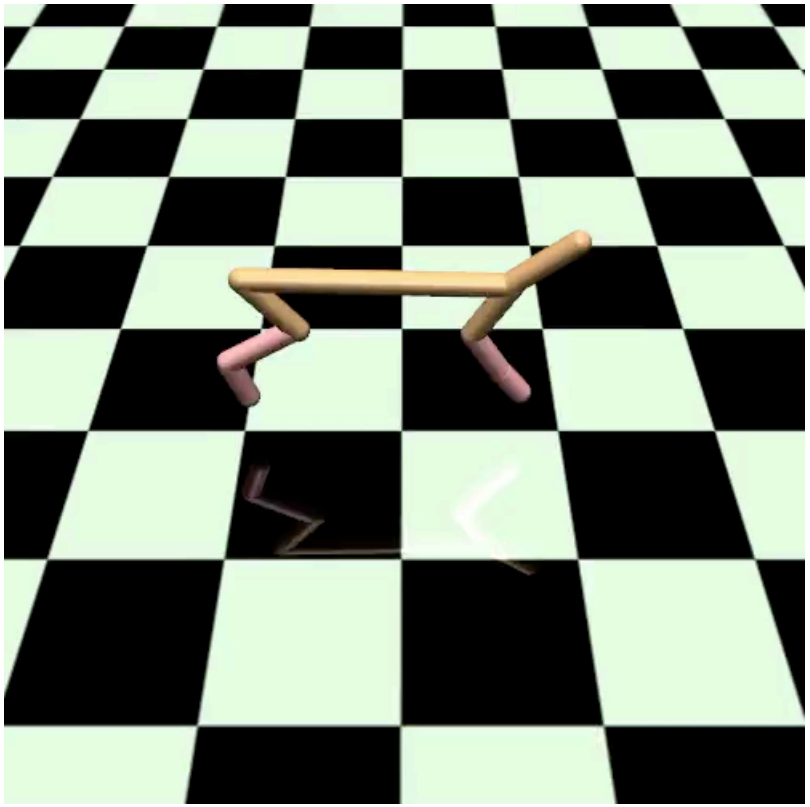
No Exploration (SGD)

Reward = 2860

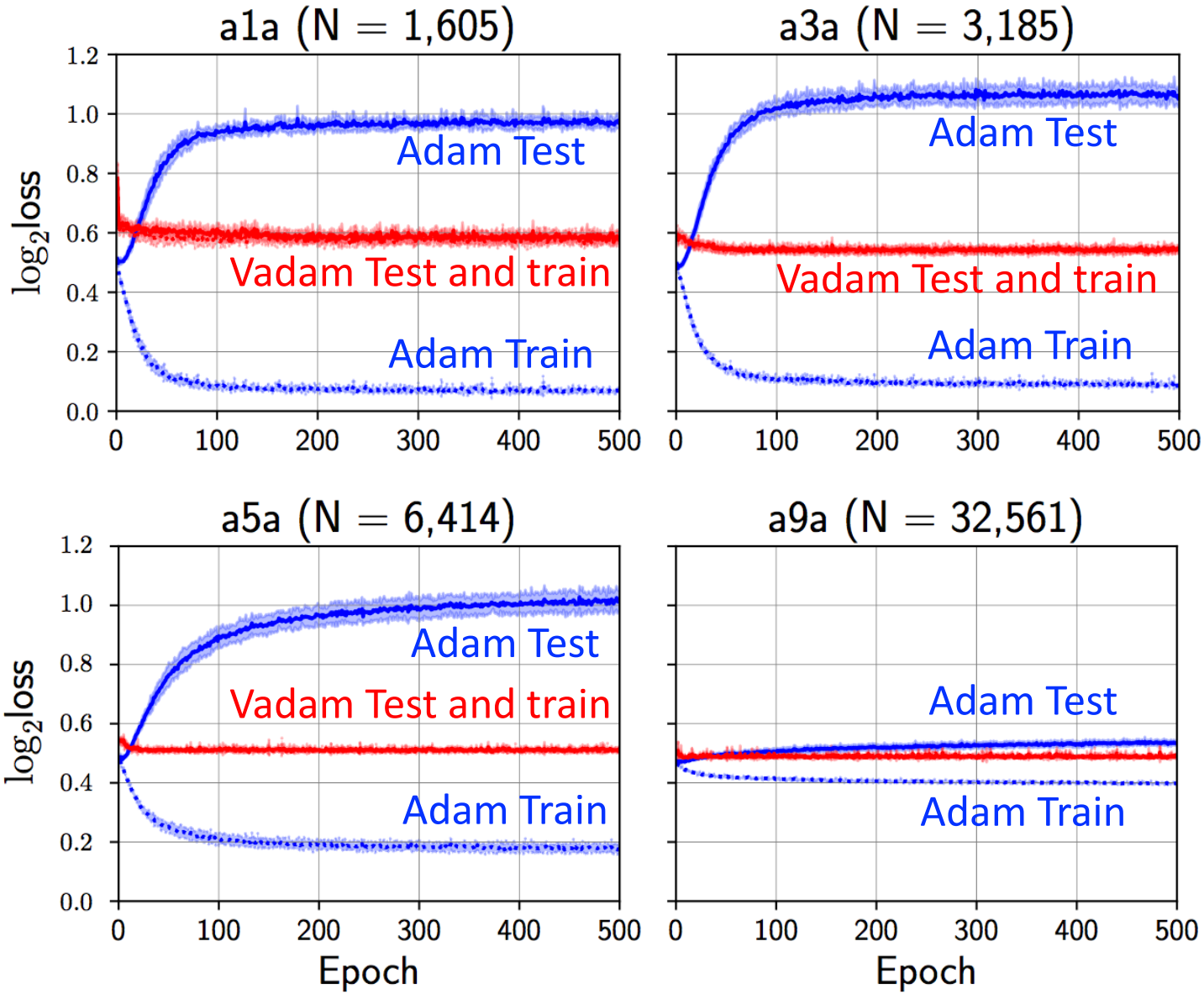


Exploration using Vadam

Reward = 5264

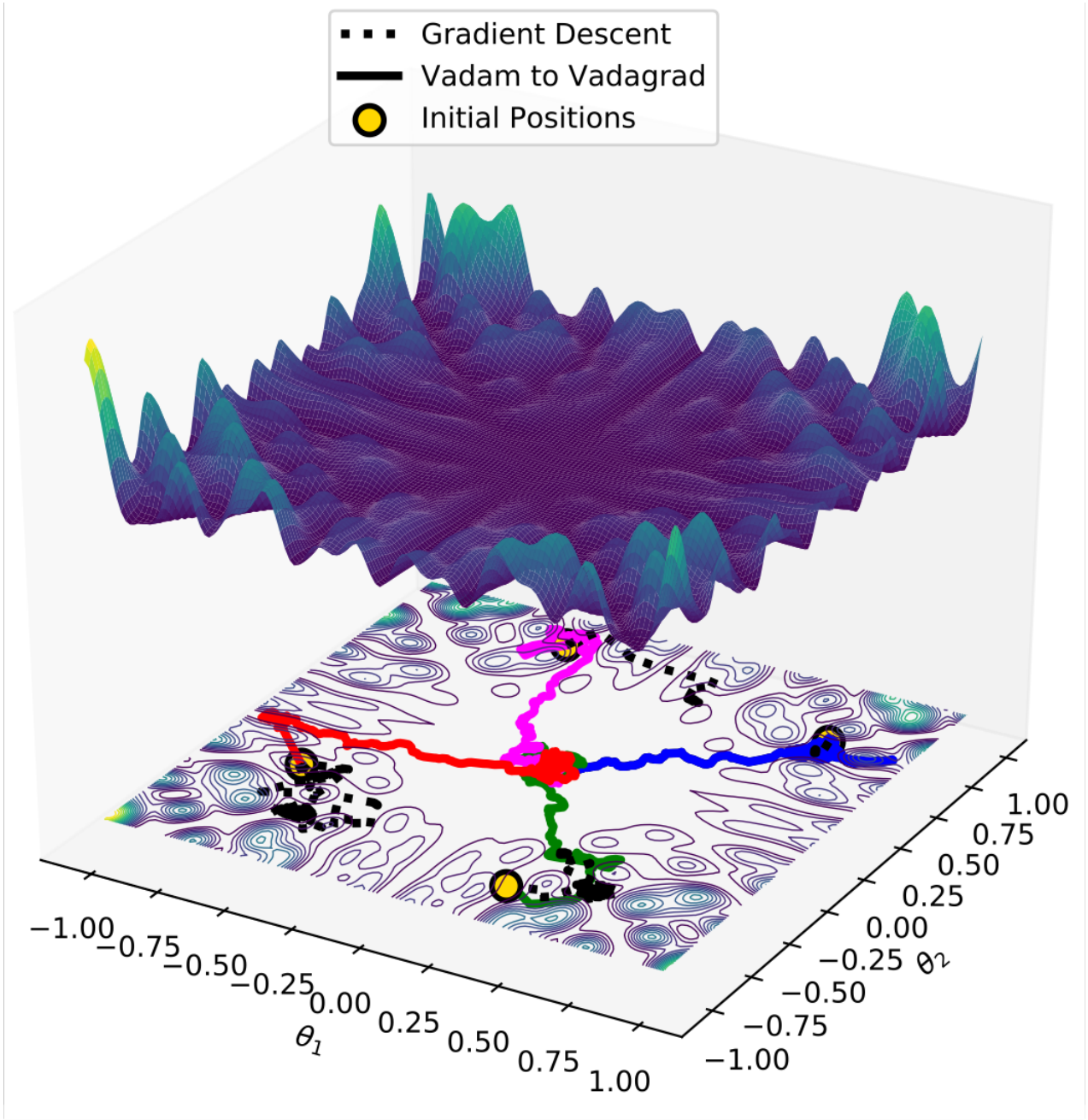


Reduce Overfitting with Vadam



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

- Quality of uncertainty estimates
  - Application to life science?
  - Check out the “Bayesian deep learning” workshop at NIPS 2018.
- Estimating various types of uncertainty
  - Model uncertainty vs data uncertainty
  - Applications play a big role here
- Is uncertainty in deep learning useful?
  - Multiple local minima make it difficult to establish

References

https://emtiyaz.github.io

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]

Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam,
(ICML 2018) **M.E. KHAN**, D. NIELSEN, V. TANGKARATT, W. LIN, Y. GAL, AND A. SRIVASTAVA, [ArXiv Version] [Code]

Conjugate-Computation Variational Inference : Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models,
(AISTATS 2017) **M.E. KHAN** AND W. LIN [Paper] [Code for Logistic Reg + GPs] [Code for Correlated Topic Model]

Thanks!

https://emtiyaz.github.io