The Bayesian Learning Rule

Mohammad Emtiyaz Khan

RIKEN Center for AI Project, Tokyo

http://emtiyaz.github.io

AI that learn like humans

Quickly adapt to learn new skills, throughout their lives

Human Learning at the age of 6 months.

Converged at the age of 12 months

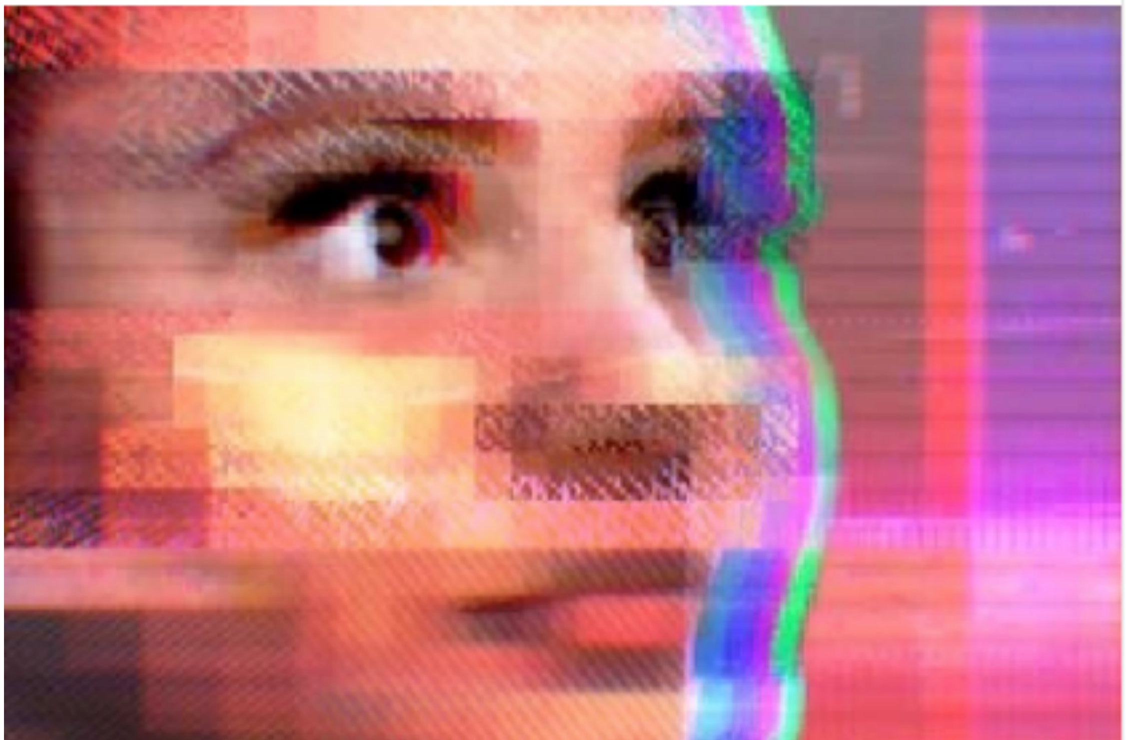
Transfer skills

at the age of 14 months

Fail because too quick to adapt

TayTweets: Microsoft AI bot manipulated into being extreme racist upon release

Posted Fri 25 Mar 2016 at 4:38am, updated Fri 25 Mar 2016 at 9:17am



TayTweets is programmed to converse like a teenage girl who has "zero chill", according to Microsoft. (Twitter: TayTweets)

https://www.abc.net.au/news/2016-03-25/microsoft-created-ai-bot-becomes-racist/7276266

Failure of AI in “dynamic” setting

Robots need quick adaptation to be deployed
(for example, at homes for elderly care)

https://www.youtube.com/watch?v=TxobtWAFh8o The video is from 2017

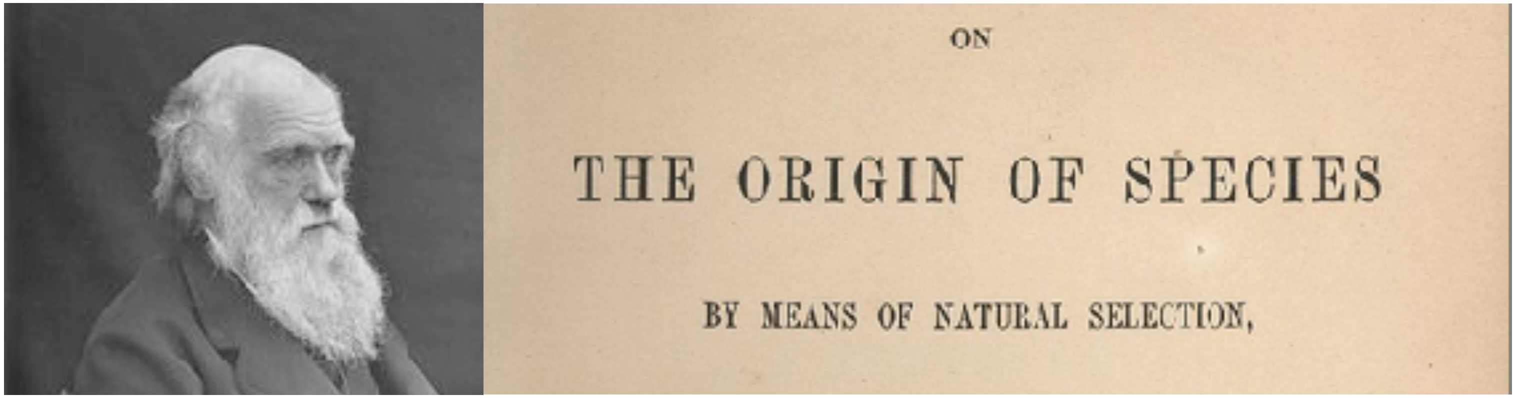
AI that learn like humans

Quickly adapt to learn new skills, throughout their lives

See Section 6 (discussion) in Khan and Rue, 2021

Principles of "good" algorithms?

- What are (some) common principles of good algorithms?
- Common origin of Algorithms
  - Revise past belief using new data



Principles of "good" algorithms?

- Bayesian principles
  - To unify/generalize/improve learning-algorithms
  - By computing "posterior approximations"
- Bayesian Learning rule (BLR)
  - Derive many existing algorithms
  - Deep Learning (SGD, RMSprop, Adam)
  - Design new algorithms for uncertainty in DL
- Impact: Everything with the same principle

The Bayesian Learning Rule

Mohammad Emtiyaz Khan
RIKEN Center for AI Project
Tokyo, Japan
emtiyaz.khan@riken.jp

Håvard Rue
CEMSE Division, KAUST
Thuwal, Saudi Arabia
haavard.rue@kaust.edu.sa

Abstract

We show that many machine-learning algorithms are specific instances of a single algorithm called the *Bayesian learning rule*. The rule, derived from Bayesian principles, yields a wide-range of algorithms from fields such as optimization, deep learning, and graphical models. This includes classical algorithms such as ridge regression, Newton's method, and Kalman filter, as well as modern deep-learning algorithms such as stochastic-gradient descent, RMSprop, and Dropout. The key idea in deriving such algorithms is to approximate the posterior using candidate distributions estimated by using natural gradients. Different candidate distributions result in different algorithms and further approximations to natural gradients give rise to variants of those algorithms. Our work not only unifies, generalizes, and improves existing algorithms, but also helps us design new ones.

Khan and Rue, The Bayesian Learning Rule, arXiv, https://arxiv.org/abs/2107.04562, 2021

Bayesian learning rule

See Table 1 in Khan and Rue, 2021

Learning Algorithm	Posterior Approx.	Natural-Gradient Approx.	Sec.
Optimization Algorithms			
Gradient Descent	Gaussian (fixed cov.)	Delta method	1.3
Newton's method	Gaussian	—"—	1.3
Multimodal optimization (New)	Mixture of Gaussians	—"—	3.2
Deep-Learning Algorithms			
Stochastic Gradient Descent	Gaussian (fixed cov.)	Delta method, stochastic approx.	4.1
RMSprop/Adam	Gaussian (diagonal cov.)	Delta method, stochastic approx., Hessian approx., square-root scaling, slow-moving scale vectors	4.2
Dropout	Mixture of Gaussians	Delta method, stochastic approx., responsibility approx.	4.3
STE	Bernoulli	Delta method, stochastic approx.	4.5
Online Gauss-Newton (OGN) (New)	Gaussian (diagonal cov.)	Gauss-Newton Hessian approx. in Adam & no square-root scaling	4.4
Variational OGN (New)	—"—	Remove delta method from OGN	4.4
BayesBiNN (New)	Bernoulli	Remove delta method from STE	4.5
Approximate Bayesian Inference Algorithms			
Conjugate Bayes	Exp-family	Set learning rate $\rho_t = 1$	5.1
Laplace's method	Gaussian	Delta method	4.4
Expectation-Maximization	Exp-Family + Gaussian	Delta method for the parameters	5.2
Stochastic VI (SVI)	Exp-family (mean-field)	Stochastic approx., local $\rho_t = 1$	5.3
VMP	—"—	$\rho_t = 1$ for all nodes	5.3
Non-Conjugate VMP	—"—	—"—	5.3
Non-Conjugate VI (New)	Mixture of Exp-family	None	5.4

Principle of Trial-and-Error

Frequentist: Empirical Risk Minimization (ERM) or Maximum Likelihood Principle, etc.

$$\min_{\theta} \underbrace{\ell}_{\text{Loss}}(\underbrace{\mathcal{D}}_{\text{Data}}, \underbrace{\theta}_{\text{Model Params}}) = \sum_{i=1}^{N} [y_i - \underbrace{f_\theta}_{\text{Deep Network}}(x_i)]^2 + \gamma \theta^T \theta$$

Deep Learning Algorithms: $\theta \leftarrow \theta - \rho H_\theta^{-1} \nabla_\theta \ell(\theta)$

Scales well to large data and complex model, and very good performance in practice.

A Bayesian Origin

$$\min_{\theta} \ \ell(\theta) \qquad \text{vs} \qquad \min_{q \in \mathcal{Q}} \ \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

Entropy

Posterior approximation (expo-family)

Bayesian Learning Rule [1,2]

Natural and Expectation parameters of q

$$\lambda \leftarrow \underbrace{\lambda -}_{\text{Old belief}} \rho \nabla_{\mu} \underbrace{\left(\mathbb{E}_q[\ell(\theta)]\right.}_{\text{Revise using new information through natural gradients}} \left. - \mathcal{H}(q)\right)$$

Old belief

Revise using new information
through natural gradients

By changing *Q*, we can recover DL algorithms (and more)

1. Khan and Rue, The Bayesian Learning Rule, arXiv, https://arxiv.org/abs/2107.04562, 2021
2. Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." AIstats (2017).

Bayesian learning rule

See Table 1 in Khan and Rue, 2021

Learning Algorithm	Posterior Approx.	Natural-Gradient Approx.	Sec.
Optimization Algorithms			
Gradient Descent	Gaussian (fixed cov.)	Delta method	1.3
Newton's method	Gaussian	——"——	1.3
Multimodal optimization (New)	Mixture of Gaussians	——"——	3.2
Deep-Learning Algorithms			
Stochastic Gradient Descent	Gaussian (fixed cov.)	Delta method, stochastic approx.	4.1
RMSprop/Adam	Gaussian (diagonal cov.)	Delta method, stochastic approx., Hessian approx., square-root scaling, slow-moving scale vectors	4.2
Dropout	Mixture of Gaussians	Delta method, stochastic approx., responsibility approx.	4.3
STE	Bernoulli	Delta method, stochastic approx.	4.5
Online Gauss-Newton (OGN) (New)	Gaussian (diagonal cov.)	Gauss-Newton Hessian approx. in Adam & no square-root scaling	4.4
Variational OGN (New)	——"——	Remove delta method from OGN	4.4
BayesBiNN (New)	Bernoulli	Remove delta method from STE	4.5
Approximate Bayesian Inference Algorithms			
Conjugate Bayes	Exp-family	Set learning rate $\rho_t = 1$	5.1
Laplace's method	Gaussian	Delta method	4.4
Expectation-Maximization	Exp-Family + Gaussian	Delta method for the parameters	5.2
Stochastic VI (SVI)	Exp-family (mean-field)	Stochastic approx., local $\rho_t = 1$	5.3
VMP	——"——	$\rho_t = 1$ for all nodes	5.3
Non-Conjugate VMP	——"——	——"——	5.3
Non-Conjugate VI (New)	Mixture of Exp-family	None	5.4

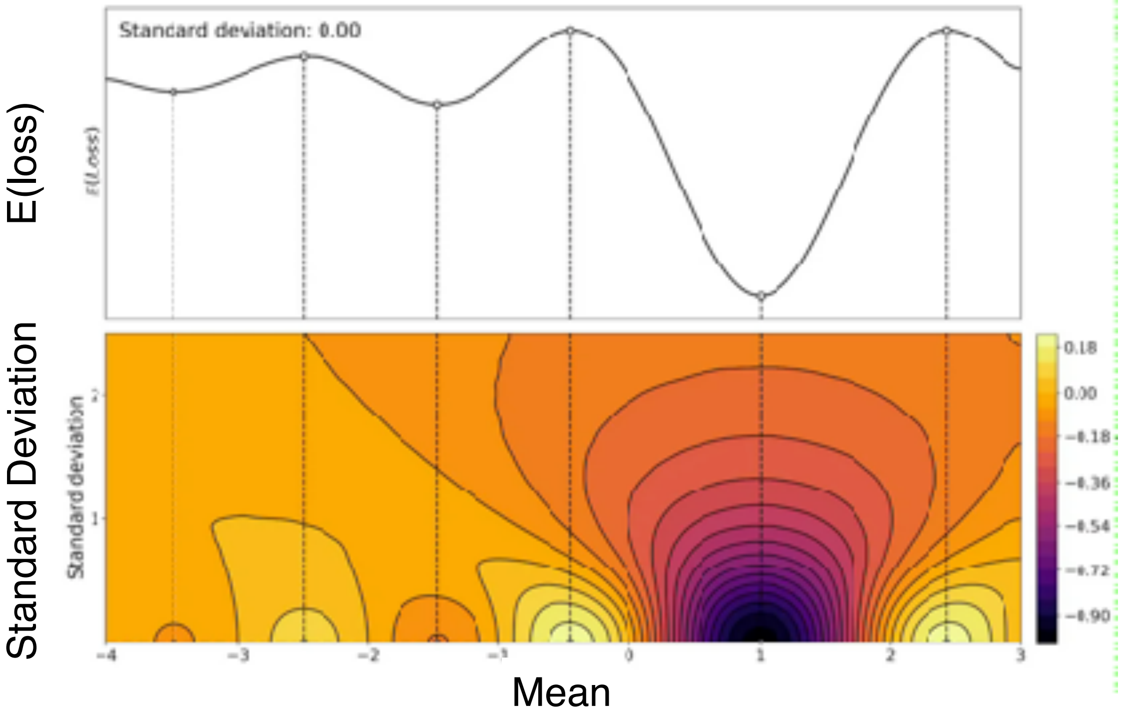
See Section 1.2, Eq 2 in Khan and Rue, 2021

Bayes Objective

$$\min_{\theta} \ \ell(\theta) \quad \text{vs} \quad \min_{q \in \mathcal{Q}} \ \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

Entropy

Generalized-Posterior approx.



Instead of the original loss, optimize a different (smoothed) one (a popular idea now for DL theory [4]).

A common idea in Inference, optimization, online learning, Reinforcement learning

1. Zellner, A. "Optimal information processing and Bayes's theorem." *The American Statistician* (1988)
2. Many other: Bissiri, et al. (2016), Shawe-Taylor and Williamson (1997), Cesa-Bianchi and Lugosi (2006)
3. Huszar's blog, Evolution Strategies, Variational Optimisation and Natural ES (2017)
4. Smith et al., On the Origin of Implicit Regularization in Stochastic Gradient Descent, ICLR, 2021

See Section 1.2 in Khan and Rue, 2021

Exponential Family

Natural parameters — Sufficient statistics — Expectation parameters

$$q(\theta) \propto \exp\left[\lambda^\top T(\theta)\right] \qquad \mu := \mathbb{E}_q[T(\theta)]$$

$$\mathcal{N}(\theta|m, S^{-1}) \propto \exp\left[-\frac{1}{2}(\theta - m)^\top S(\theta - m)\right]$$

$$\propto \exp\left[(Sm)^\top \theta + \mathrm{Tr}\left(-\frac{S}{2}\theta\theta^\top\right)\right]$$

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta\|m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

Bayesian learning rule: $\lambda \leftarrow \lambda - \rho \nabla_{\mu} \left(\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q) \right)$

Learning Algorithm	Posterior Approx.	Natural-Gradient Approx.	Sec.
Optimization Algorithms			
Gradient Descent	Gaussian (fixed cov.)	Delta method	1.3
Newton's method	Gaussian	——"——	1.3
Multimodal optimization (New)	Mixture of Gaussians	——"——	3.2
Deep-Learning Algorithms			
Stochastic Gradient Descent	Gaussian (fixed cov.)	Delta method, stochastic approx.	4.1
RMSprop/Adam	Gaussian (diagonal cov.)	Delta method, stochastic approx., Hessian approx., square-root scaling, slow-moving scale vectors	4.2
Dropout	Mixture of Gaussians	Delta method, stochastic approx., responsibility approx.	4.3
STE	Bernoulli	Delta method, stochastic approx.	4.5
Online Gauss-Newton (OGN) (New)	Gaussian (diagonal cov.)	Gauss-Newton Hessian approx. in Adam & no square-root scaling	4.4
Variational OGN (New)	——"——	Remove delta method from OGN	4.4
BayesBiNN (New)	Bernoulli	Remove delta method from STE	4.5
Approximate Bayesian Inference Algorithms			
Conjugate Bayes	Exp-family	Set learning rate $\rho_t = 1$	5.1
Laplace's method	Gaussian	Delta method	4.4
Expectation-Maximization	Exp-Family + Gaussian	Delta method for the parameters	5.2
Stochastic VI (SVI)	Exp-family (mean-field)	Stochastic approx., local $\rho_t = 1$	5.3
VMP	——"——	$\rho_t = 1$ for all nodes	5.3
Non-Conjugate VMP	——"——	——"——	5.3
Non-Conjugate VI (New)	Mixture of Exp-family	None	5.4

See Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

Gradient descent: $\theta \leftarrow \theta - \rho \nabla_\theta \ell(\theta)$

Bayes Learn Rule: $m \leftarrow m - \rho \nabla_m \ell(m)$

"Global" to "local"
(the delta method)
$\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$

$$m \leftarrow m - \rho \nabla_m \mathbb{E}_q[\ell(\theta)]$$

$$\lambda \leftarrow \lambda - \rho \nabla_\mu \left(\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q)\right)$$

Derived by choosing Gaussian with fixed covariance

Gaussian distribution $q(\theta) := \mathcal{N}(m, 1)$

Natural parameters $\lambda := m$

Expectation parameters $\mu := \mathbb{E}_q[\theta] = m$

Entropy $\mathcal{H}(q) := \log(2\pi)/2$

See Eq 25 in Khan and Rue, 2021 (Bonnet's theorem)

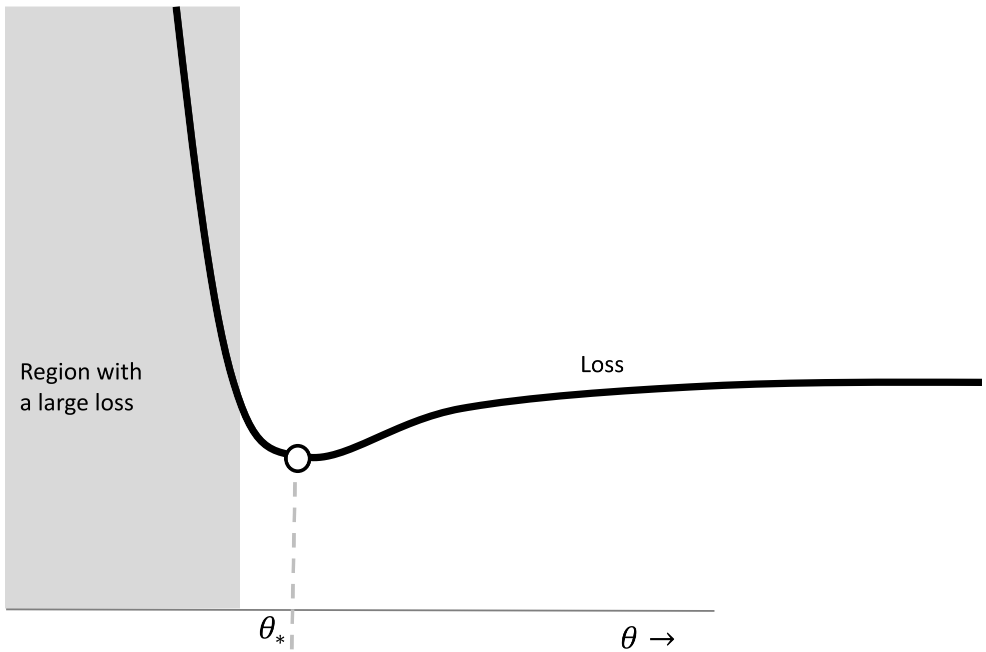
Bayes Prefers Flatter directions

GD: $\theta \leftarrow \theta - \rho \nabla_\theta \ell(\theta) \quad \Longrightarrow \quad \nabla_\theta \ell(\theta_*) = 0$

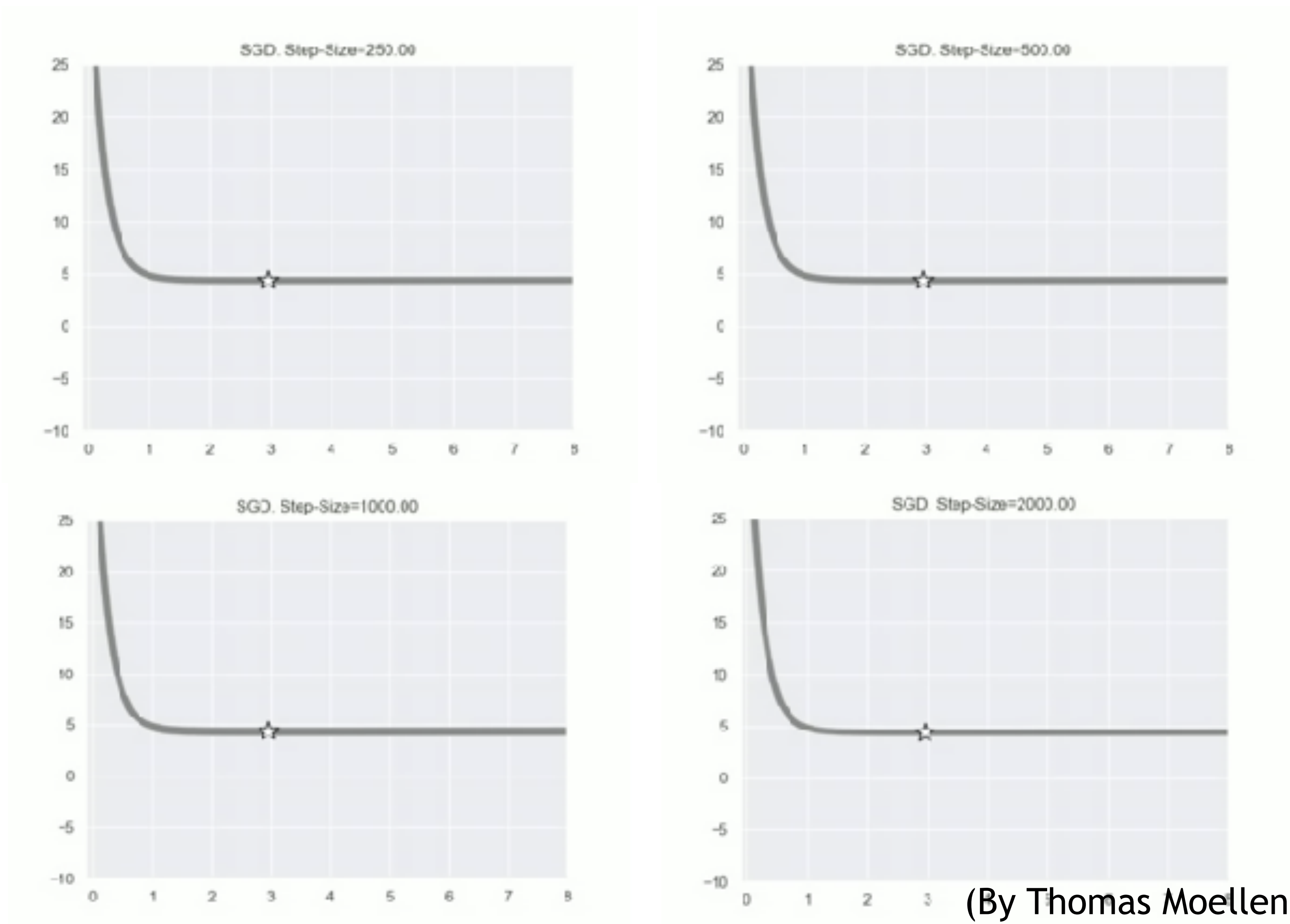
BLR: $m \leftarrow m - \rho \nabla_{m} \mathbb{E}_q[\ell(\theta)]$

$$\Longrightarrow \nabla_m \mathbb{E}_{q_*}[\ell(\theta)] = 0 \quad \Longrightarrow \mathbb{E}_{q_*}[\nabla_\theta \ell(\theta)] = 0$$

Bayesian solution injects "noise" which has a similar regularization effect to noise in Stochastic GD. It prefers "flatter" directions.

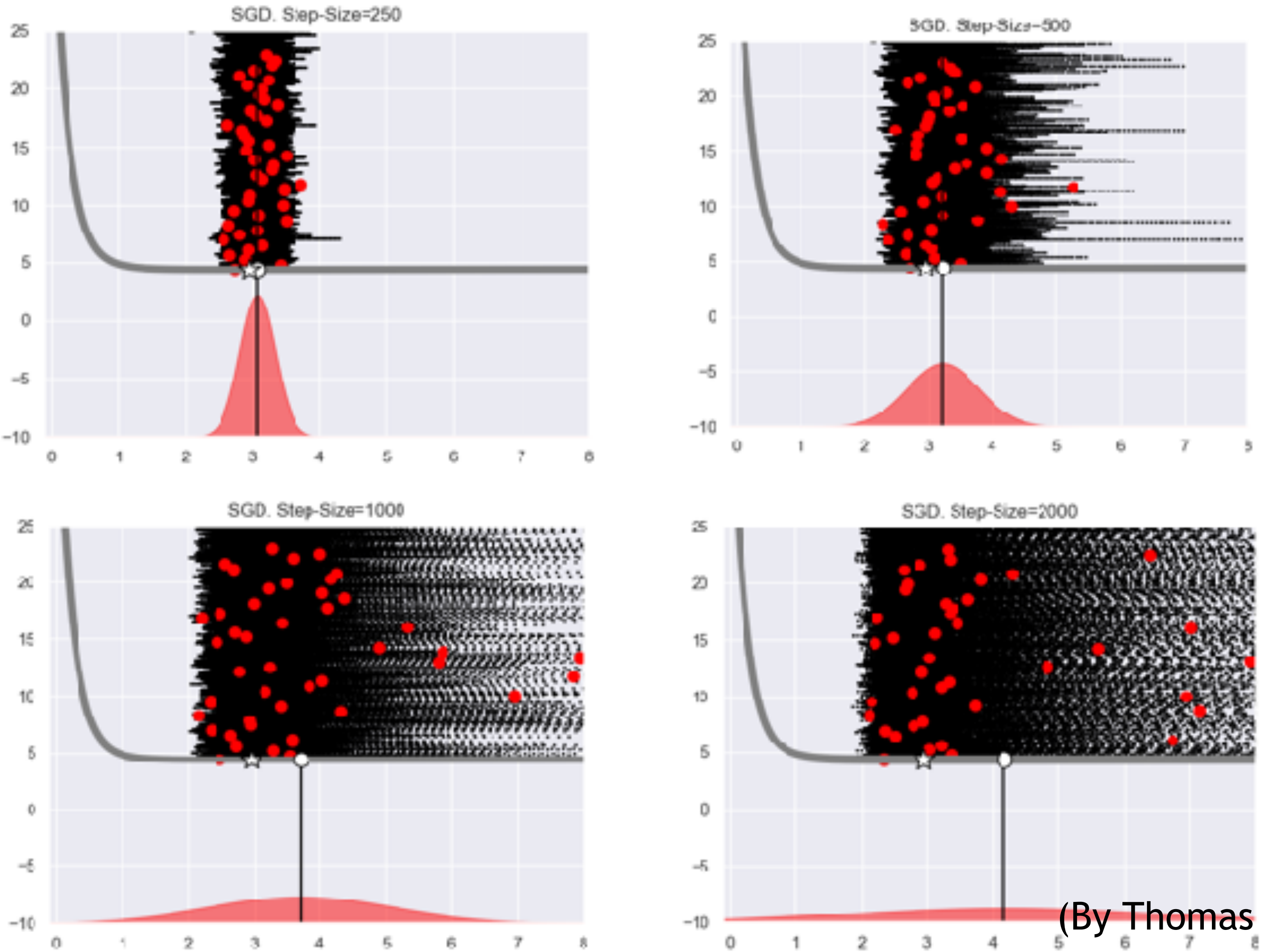


SGD: Implicit Regularization



(By Thomas Moellenhoff)

SGD: Implicit Regularization

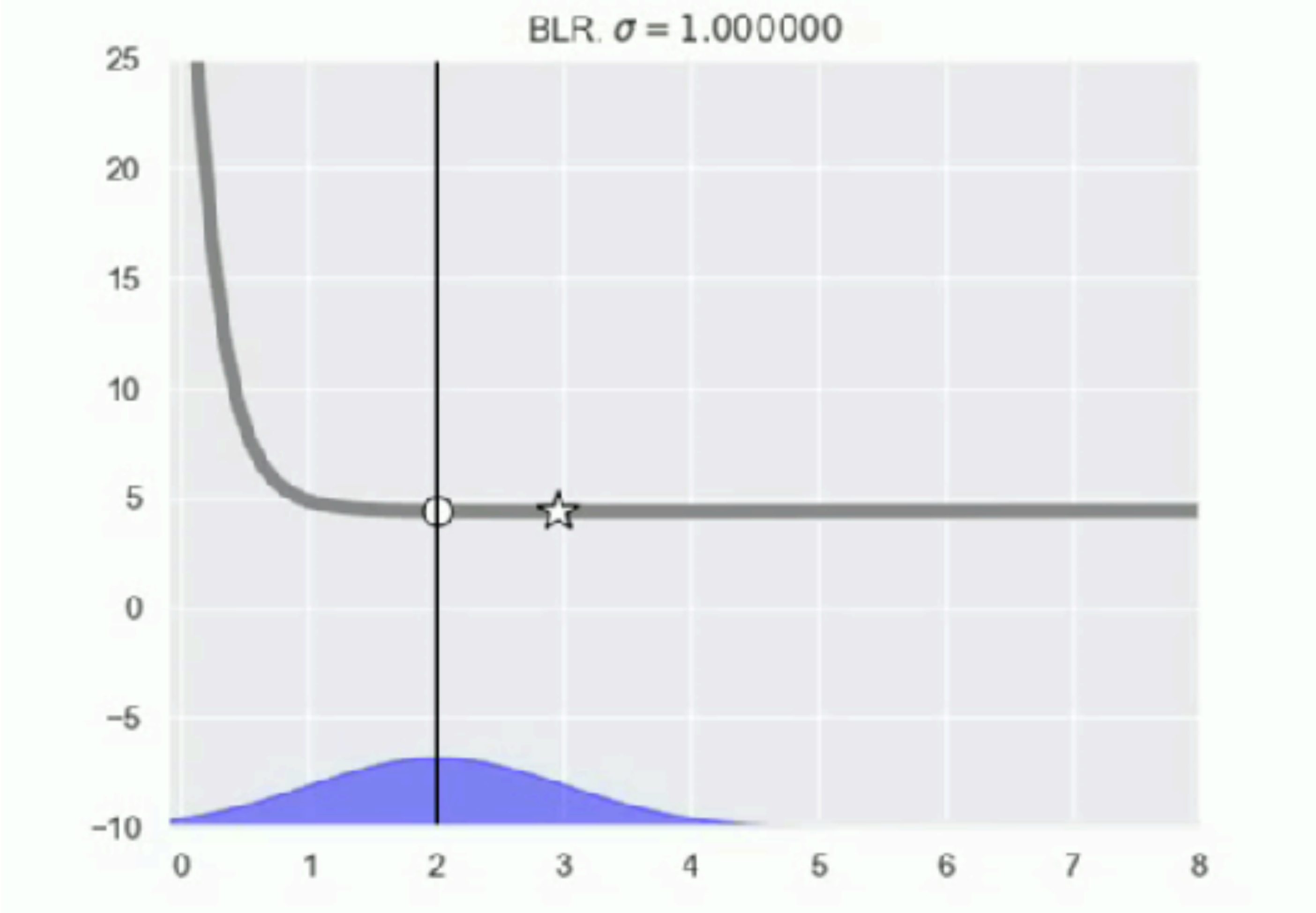


(By Thomas Moellenhoff)

Bayes: Implicit Regularization

Estimating Gaussian posteriors where the variance is fixed, and only the mean is estimated

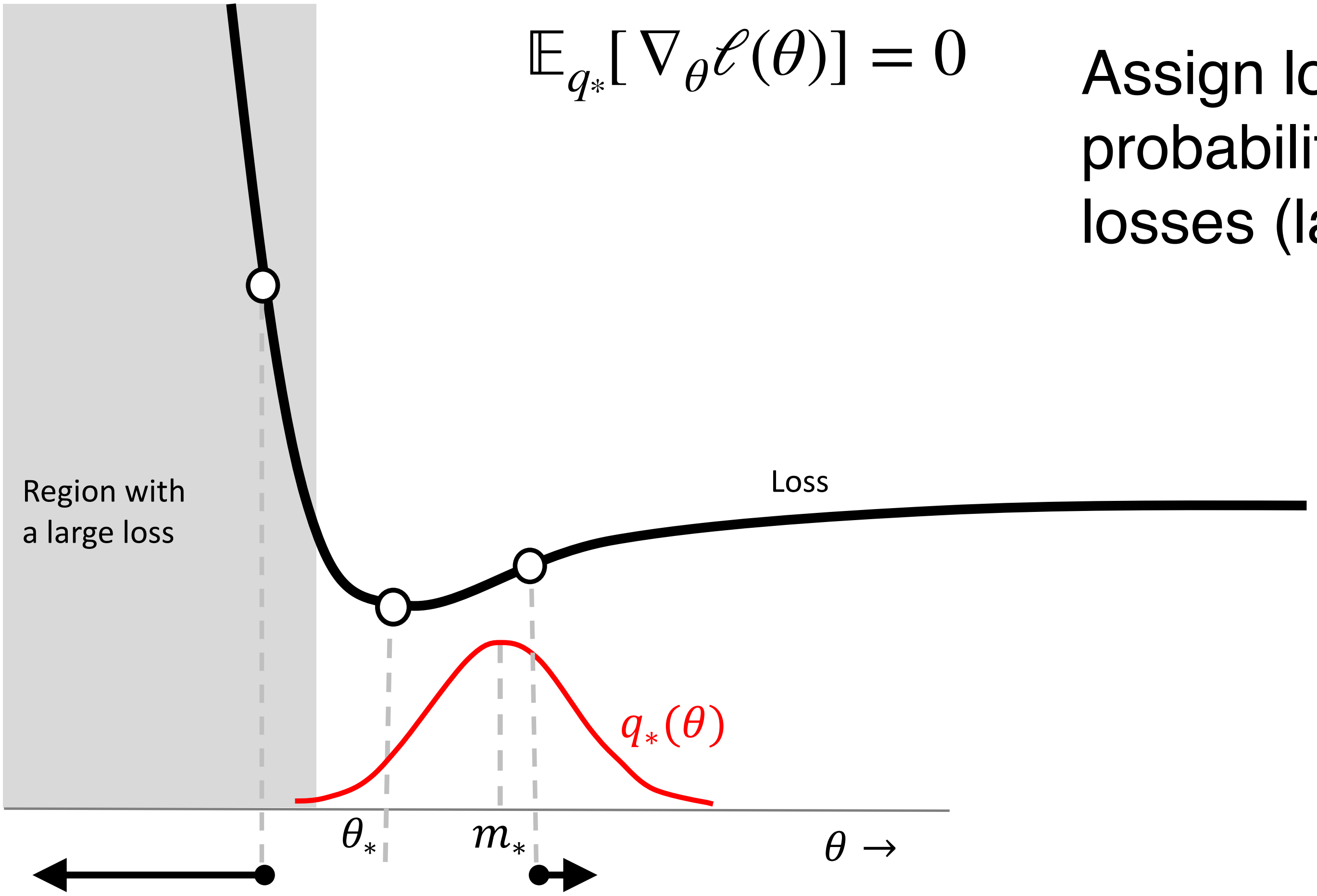
$$\mathbb{E}_{q_*}[\nabla_\theta \ell(\theta)] = 0$$



See Section 4.4, Fig. 1 in Khan and Rue, 2021

Bayes: Implicit Regularization

$$\mathbb{E}_{q_*}[\nabla_\theta \ell(\theta)] = 0$$



Assign lower probability to higher losses (large -ve grad)

Bayes: Implicit Regularization

Bayes solutions (blue) with different variances vs
SGD solutions (red lines) with different learning rates.

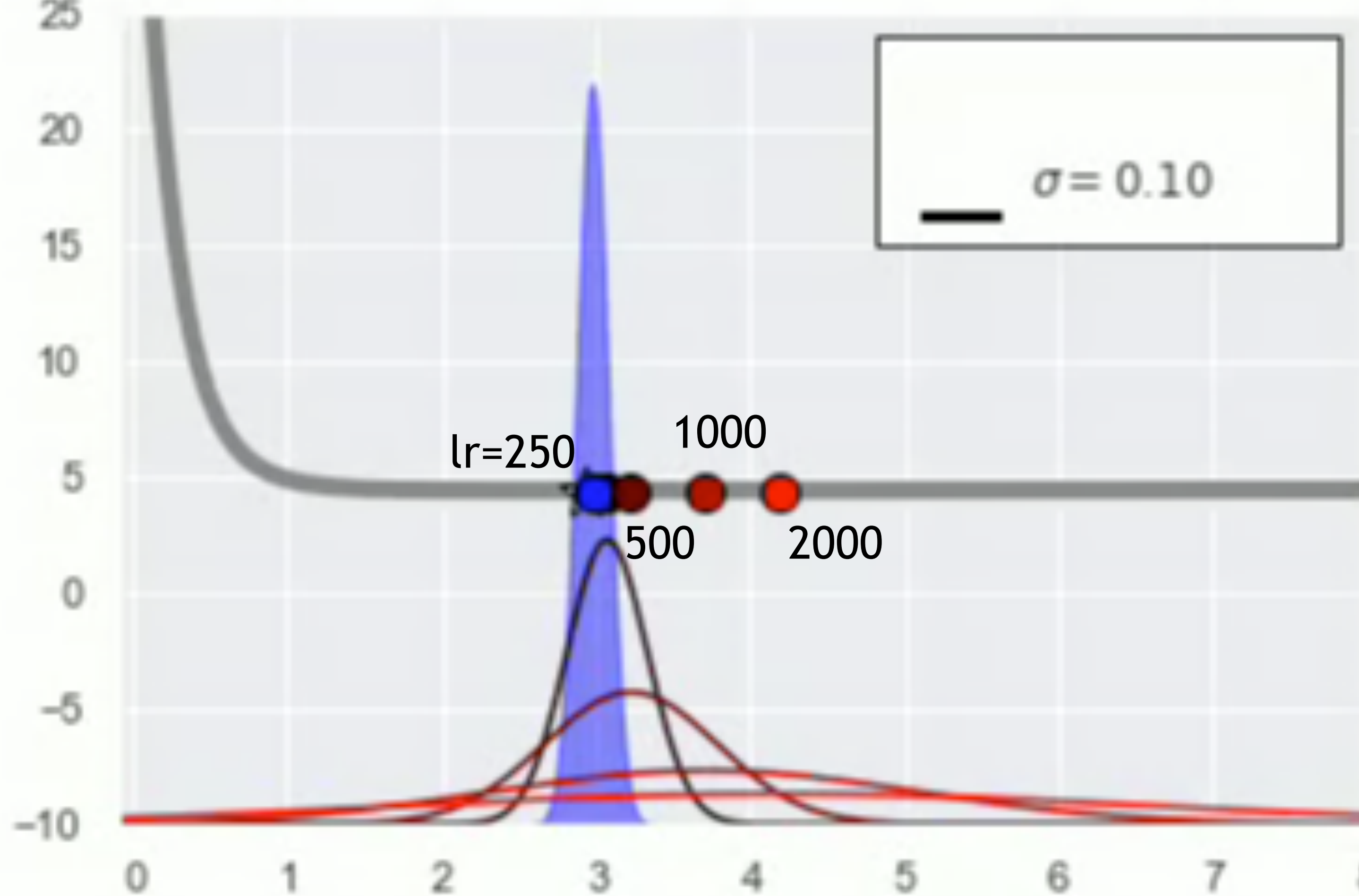


Image from https://www.buzzfeed.com/daniellaemanuel/bernie-sanders-once-again-asking

Bayesian learning rule: $\lambda \leftarrow (1-\rho)\lambda - \rho \nabla_{\mu} \mathbb{E}_q[\ell(\theta)]$

Learning Algorithm	Posterior Approx.	Natural-Gradient Approx.	Sec.
Optimization Algorithms			
Gradient Descent	Gaussian (fixed cov.)	Delta method	1.3
Newton's method	Gaussian	—"—	1.3
Multimodal optimization (New)	Mixture of Gaussians	—"—	3.2
Deep-Learning Algorithms			
Stochastic Gradient Descent	Gaussian (fixed cov.)	Delta method, stochastic approx.	4.1
RMSprop/Adam	Gaussian (diagonal cov.)	Delta method, stochastic approx., Hessian approx., square-root scaling, slow-moving scale vectors	4.2
Dropout	Mixture of Gaussians	Delta method, stochastic approx., responsibility approx.	4.3
STE	Bernoulli	Delta method, stochastic approx.	4.5
Online Gauss-Newton (OGN) (New)	Gaussian (diagonal cov.)	Gauss-Newton Hessian approx. in Adam & no square-root scaling	4.4
Variational OGN (New)	—"—	Remove delta method from OGN	4.4
BayesBiNN (New)	Bernoulli	Remove delta method from STE	4.5
Approximate Bayesian Inference Algorithms			
Conjugate Bayes	Exp-family	Set learning rate $\rho_t = 1$	5.1
Laplace's method	Gaussian	Delta method	4.4
Expectation-Maximization	Exp-Family + Gaussian	Delta method for the parameters	5.2
Stochastic VI (SVI)	Exp-family (mean-field)	Stochastic approx., local $\rho_t = 1$	5.3
VMP	—"—	$\rho_t = 1$ for all nodes	5.3
Non-Conjugate VMP	—"—	—"—	5.3
Non-Conjugate VI (New)	Mixture of Exp-family	None	5.4

Put the expectation (Bayes) back in!

The BLR variants [1,2,3] led to the winning solution for the NeurIPS 2021 challenge for "approximate inference in BDL". Watch Thomas Moellenhoff's talk at https://www.youtube.com/watch?v=LQInlN5EU7E

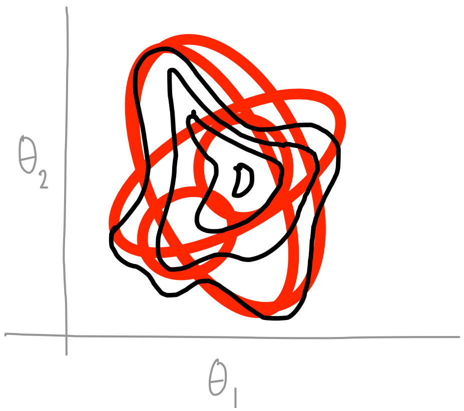
1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." NeurIPS (2019).
3. Lin et al. "Handling the positive-definite constraints in the BLR." ICML (2020).

See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex ⟷ Simple



Bayes' rule

Mixture of Newton

Newton

Gradient Descent

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_\theta^{-1}\left[\nabla_\theta \ell(\theta)\right]$

$$Sm \leftarrow (1-\rho)Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)]$$

$$-\frac{1}{2}S \leftarrow (1-\rho)\left(-\frac{1}{2}S\right) - \rho \nabla_{\mathbb{E}_q(\theta\theta^\top)} \mathbb{E}_q[\ell(\theta)]$$

$$\lambda \leftarrow (1-\rho)\lambda - \rho \nabla_\mu \mathbb{E}_q[\ell(\theta)]$$

$$\boxed{-\nabla_\mu \mathcal{H}(q) = \lambda}$$

Derived by choosing a multivariate Gaussian

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta \mid m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from Bayes

Newton's method: $\theta \leftarrow \theta - H_\theta^{-1}\left[\nabla_\theta \ell(\theta)\right]$

Set ρ =1 to get $\quad m \leftarrow m - H_m^{-1}[\nabla_m \ell(m)]$

$$m \leftarrow m - \rho S^{-1} \nabla_m \ell(m)$$
$$S \leftarrow (1-\rho) S + \rho H_m$$

Delta Method
$$\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$$

Express in terms of gradient and Hessian of loss:

$$\nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[\nabla_\theta \ell(\theta)] - 2\mathbb{E}_q[H_\theta] m$$

$$\nabla_{\mathbb{E}_q(\theta\theta^\top)} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[H_\theta]$$

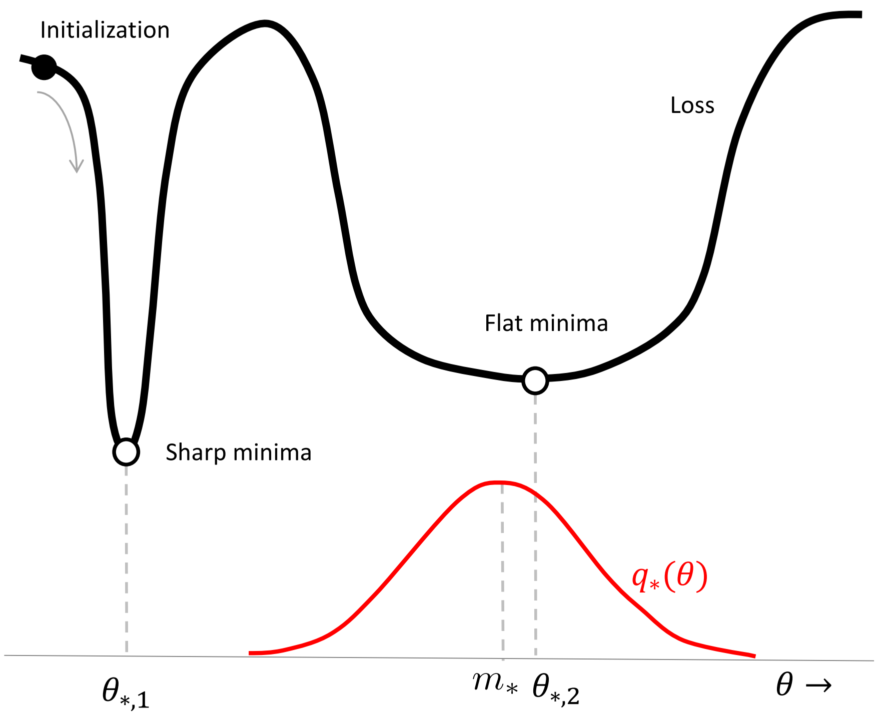
$$Sm \leftarrow (1-\rho) Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)]$$
$$S \leftarrow (1-\rho) S - \rho 2 \nabla_{\mathbb{E}_q(\theta\theta^\top)} \mathbb{E}_q[\ell(\theta)]$$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 4.4, Fig. 1 in Khan and Rue, 2021

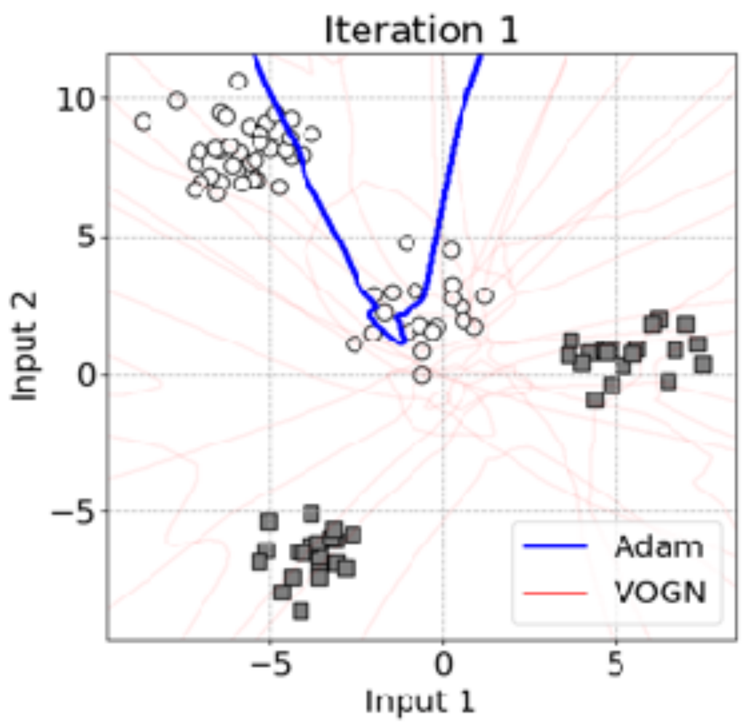
Bayes leads to robust solutions

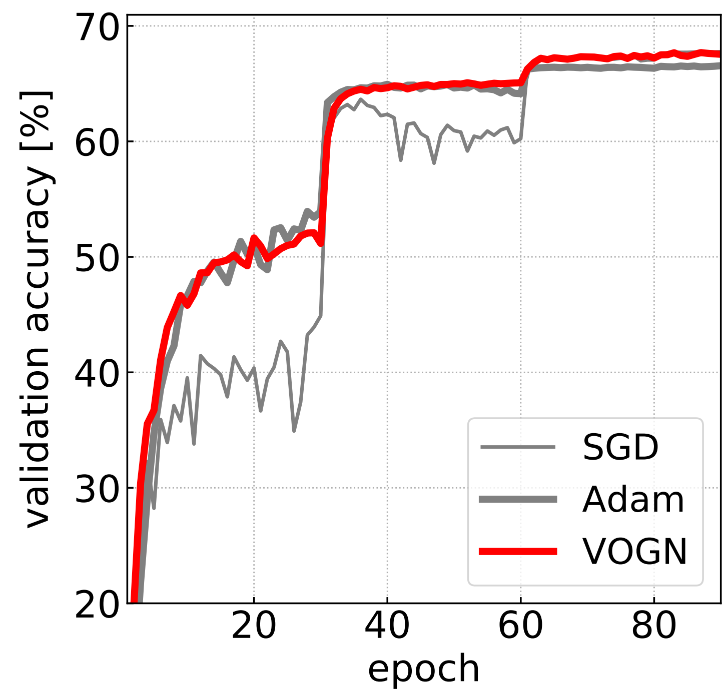
Avoiding sharp minima



Uncertainty of Deep Nets

VOGN: A modification of Adam but match the performance on ImageNet





Code available at https://github.com/team-approx-bayes/dl-with-bayes

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. “Practical Deep Learning with Bayesian Principles.” NeurIPS (2019).

BLR Variants

RMSprop

$$g \leftarrow \hat{\nabla}\ell(\theta)$$
$$s \leftarrow (1-\rho)s + \rho g^2$$
$$\theta \leftarrow \theta - \alpha(\sqrt{s}+\delta)^{-1} g$$

Variational Online Gauss-Newton (VOGN)

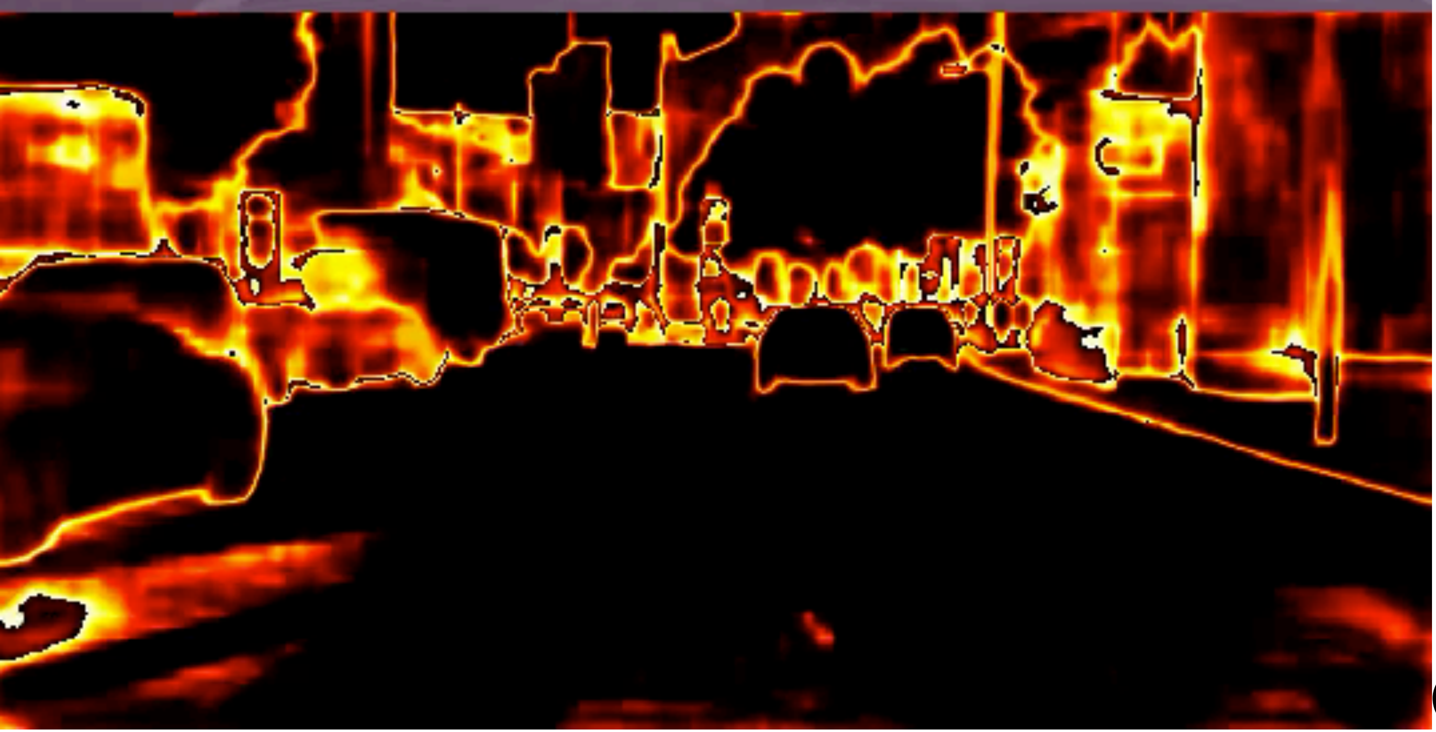
$$g \leftarrow \hat{\nabla}\ell(\theta), \text{ where } \theta \sim \mathcal{N}(m, \sigma^2)$$
$$s \leftarrow (1-\rho)s + \rho(\Sigma_i g_i^2)$$
$$m \leftarrow m - \alpha(s+\gamma)^{-1}\nabla_\theta \ell(\theta)$$
$$\sigma^2 \leftarrow (s+\gamma)^{-1}$$

Available at https://github.com/team-approx-bayes/dl-with-bayes

The BLR variant from [3] led to the winning solution for the NeurIPS 2021 challenge for "approximate inference in deep learning". Watch Thomas Moellenhoff's talk at https://www.youtube.com/watch?v=LQInlN5EU7E.

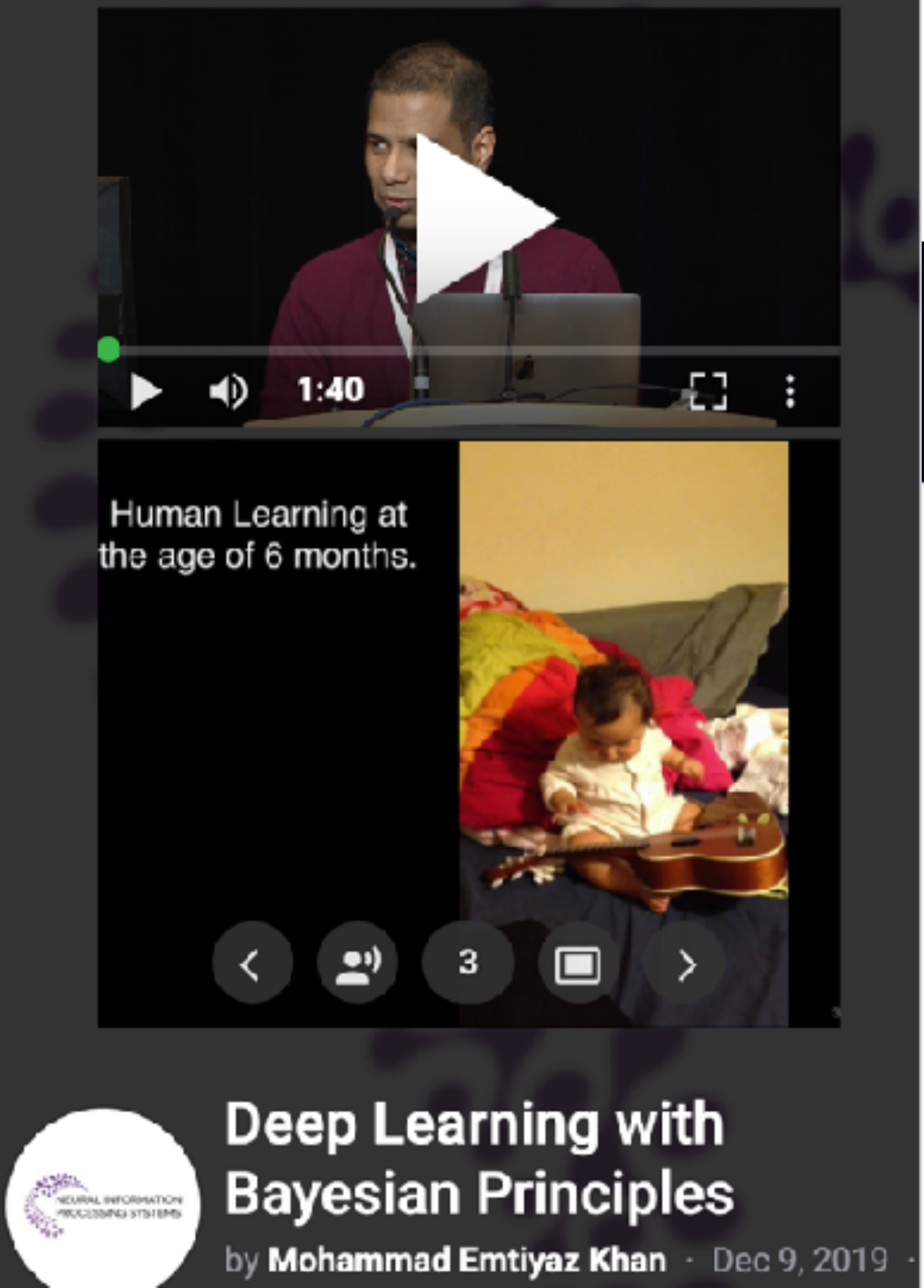
1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." NeurIPS (2019).
3. Lin et al. "Handling the positive-definite constraints in the BLR." ICML (2020).

Image Segmentation

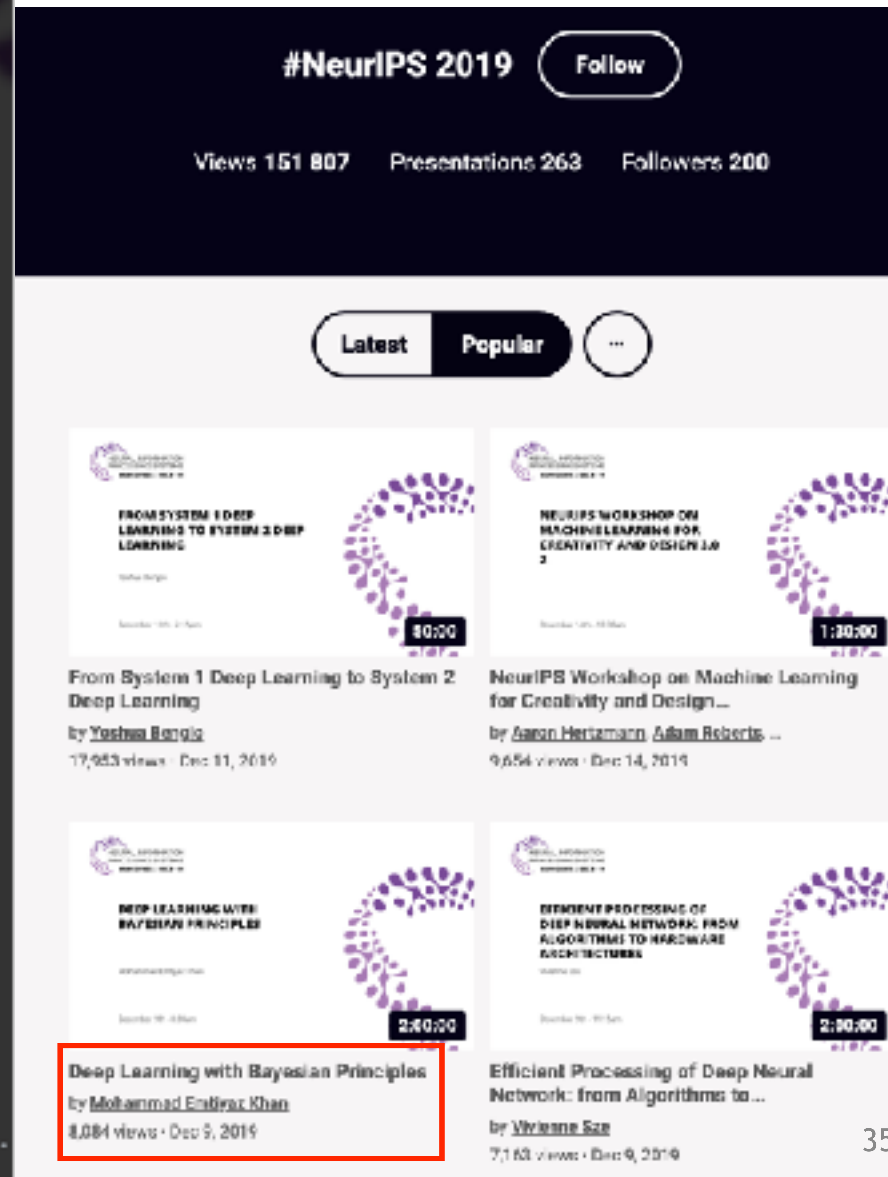


Uncertainty (entropy of class probs)

(By Roman Bachmann)



NeurIPS 2019 Tutorial



Past and New Work

- Natural Gradient Variational Inference
    1. Khan and Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models." AIstats (2017).
    2. Khan and Nielsen. "Fast yet simple natural-gradient descent for variational inference in complex models." *(2018) ISITA*.
- Mixture of Exponential family
    3. Lin et al. "Fast and Simple Natural-Gradient Variational Inference with Mixture of Exponential-family Approximations," ICML (2019).
- Generalization of natural gradients
    4. Lin et al. "Handling the Positive-Definite Constraint in the Bayesian Learning Rule", ICML (2020)
    5. Lin et al. "Tractable structured natural gradient descent using local parameterizations", ICML, (2021)
- Gaussian approx $\longleftrightarrow$ Newton-variants

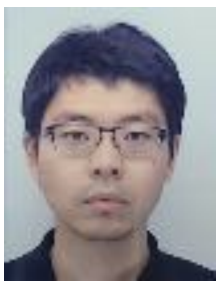
Wu Lin (UBC)

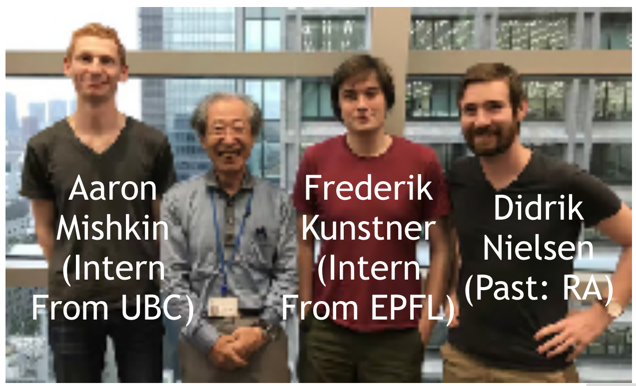
Mark Schmidt (UBC)

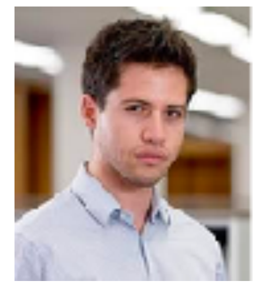
Frank Nielsen (Sony)

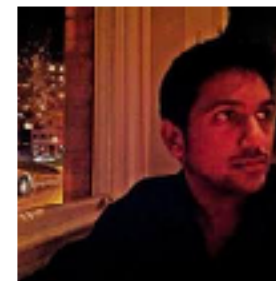
Gaussian Approximation and DL

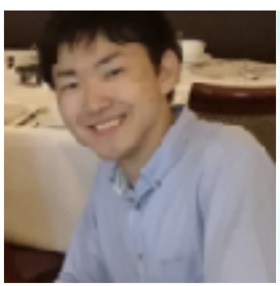
1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Mishkin et al. “SLANG: Fast Structured Covariance Approximations for Bayesian Deep Learning with Natural Gradient” NeurIPS (2018).
3. Osawa et al. “Practical Deep Learning with Bayesian Principles.” NeurIPS (2019).


Voot Tangkaratt (Postdoc, RIKEN-AIP)



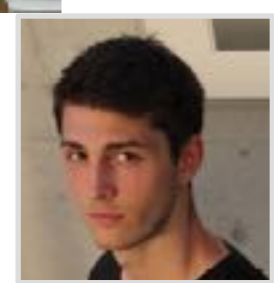

Yarin Gal (UOxford)


Akash Srivastava (UEdinburgh)


Kazuki Osawa (Tokyo Tech)


Rio Yokota (Tokyo Tech)


Anirudh Jain (Intern from IIT-ISM, India)


Runa Eschenhagen (Intern from U Osnabruck)


Siddharth Swaroop (UCambridge)


Rich Turner (UCambridge)

Extensions

- Binary Neural Networks (Bernoulli approx)
  1. Meng, et al. "Training Binary Neural Networks using the Bayesian Learning Rule." *ICML* (2020).
- Gaussian Process
  2. Chang et al. "Fast Variational Learning in State-Space GP Models", MLSP (2020)
  - For sparse GPs, BLR is a generalization of [1]

Roman Bachmann (Intern from EPFL)

Xiangming Meng (RIKEN-AIP)

Paul Chang (Aalto University)

W. J. Wilkinson (Aalto University)

Arno Solin (Aalto University)

1. Hensman et al. "Gaussian Process for Big Data", UAI (2013)

How to design AI that learn like us?

- Three questions
  - Q1: What do we know? (model)
  - Q2: What do we not know? (uncertainty)
  - Q3: What do we need to know? (action & exploration)
- Posterior approximation is the key
  - (Q1) Models == representation of the world
  - (Q2) Posterior approximations == representation of the model
  - (Q3) Use posterior approximations for knowledge representation, transfer, and collection.

Approximate Bayesian Inference Team

https://team-approx-bayes.github.io/

We have many open positions!
Come, join us.

Emtiyaz Khan
Team Leader

Pierre Alquier
Research Scientist

Gian Maria Marconi
Postdoc

Thomas Möllenhoff
Postdoc

Lu Xu
Postdoc

Jooyeon Kim
Postdoc

Wu Lin
PhD Student
University of British Columbia

David Tomàs Cuesta
Rotation Student,
Okinawa Institute of Science and Technology

Dharmesh Tailor
Remote Collaborator
University of Amsterdam

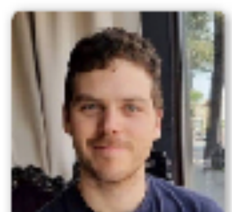
Erik Daxberger
Remote Collaborator
University of Cambridge

Tojo Rakotoaritina
Rotation Student,
Okinawa Institute of Science and Technology

Peter Nickl
Research Assistant

Happy Buzaaba
Part-time Student
University of Tsukuba

Siddharth Swaroop
Remote Collaborator
University of Cambridge



Alexandre Piché
Remote Collaborator
MILA

Paul Chang
Remote Collaborator
Aalto University