Bayesian Principles for Learning-Machines

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.

Human Learning at the age of 6 months.

Human Learning at the age of 6 months.

Converged at the age of 12 months

Converged at the age of 12 months

Converged at the age of 12 months

Transfer skills

at the age of 14 months

Transfer skills

at the age of 14 months

Transfer skills

at the age of 14 months

Failure of AI in “dynamic” setting

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

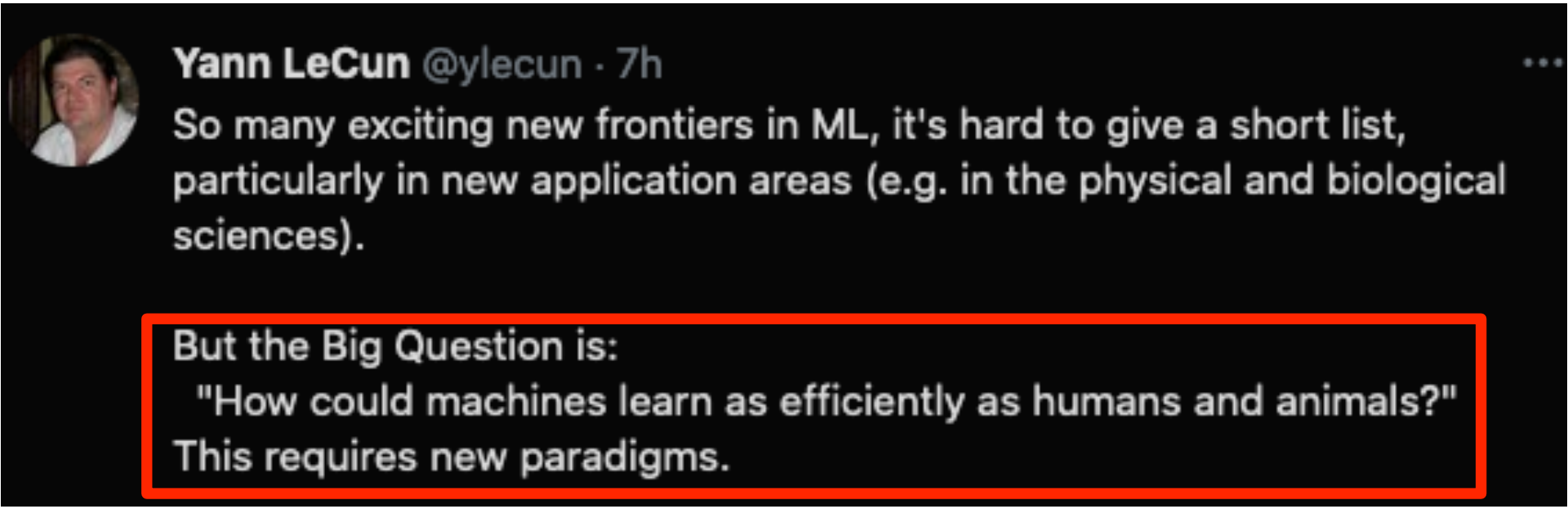
https://www.youtube.com/watch?v=TxobtWAFh8o The video is from 2017

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

July 14, 2021



Towards a new learning paradigm,
based on Bayesian principles

Human learning $\neq$ **Deep learning**

Life-long learning from small chunks of data in a non-stationary world

Bulk learning from a large amount of data in a stationary world

1. Parisi, German I., et al. "Continual lifelong learning with neural networks: A review." *Neural Networks* (2019)

Human learning $\neq$ **Deep learning**

Life-long learning from small chunks of data in a non-stationary world

Bulk learning from a large amount of data in a stationary world

Our current research focuses on reducing this gap!

1. Parisi, German I., et al. "Continual lifelong learning with neural networks: A review." *Neural Networks* (2019)

Bayesian Principles

Human learning $\neq$ **Deep learning**

Life-long learning from small chunks of data in a non-stationary world

Bulk learning from a large amount of data in a stationary world

Our current research focuses on reducing this gap!

1. Parisi, German I., et al. "Continual lifelong learning with neural networks: A review." *Neural Networks* (2019)

2. Geisler, W. S., and Randy L. D. "Bayesian natural selection and the evolution of perceptual systems." *Philosophical Transactions of the Royal Society of London. Biological Sciences* (2002)

Bayesian Principles

Our research

Human learning $\neq$ **Deep learning**

Life-long learning from small chunks of data in a non-stationary world

Bulk learning from a large amount of data in a stationary world

Our current research focuses on reducing this gap!

1. Parisi, German I., et al. "Continual lifelong learning with neural networks: A review." *Neural Networks* (2019)

2. Geisler, W. S., and Randy L. D. "Bayesian natural selection and the evolution of perceptual systems." *Philosophical Transactions of the Royal Society of London. Biological Sciences* (2002)

Deep Learning with Bayesian Principles

Deep Learning with Bayesian Principles

- Bayesian principles as a general principle
  - To unify/generalize/improve learning-algorithms
  - By computing "posterior approximations"

Deep Learning with Bayesian Principles

- Bayesian principles as a general principle
  - 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

Deep Learning with Bayesian Principles

- Bayesian principles as a general principle
  - 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
- Dual perspective of BLR for life-long learning

Deep Learning with Bayesian Principles

- Bayesian principles as a general principle
  - 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
- Dual perspective of BLR for life-long learning
- 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

Principle of Trial-and-Error

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

$$\min_{\theta} \ \ell(\mathcal{D}, \theta)$$

Loss

Data

Model Params

Principle of Trial-and-Error

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

$$\min_{\theta} \ \ell(\mathcal{D}, \theta) \quad = \sum_{i=1}^{N} [y_i - f_\theta(x_i)]^2 + \gamma \theta^T \theta$$

Loss

Data

Model Params

Deep Network

Principle of Trial-and-Error

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

$$\min_{\theta} \ \ell(\mathcal{D}, \theta) \quad = \sum_{i=1}^{N} [y_i - f_\theta(x_i)]^2 + \gamma \theta^T \theta$$

Loss

Data

Model Params

Deep Network

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

Principle of Trial-and-Error

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

$$\min_{\theta} \ \ell(\mathcal{D}, \theta) \quad = \sum_{i=1}^{N} [y_i - f_\theta(x_i)]^2 + \gamma \theta^T \theta$$

Loss

Data

Model Params

Deep Network

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.

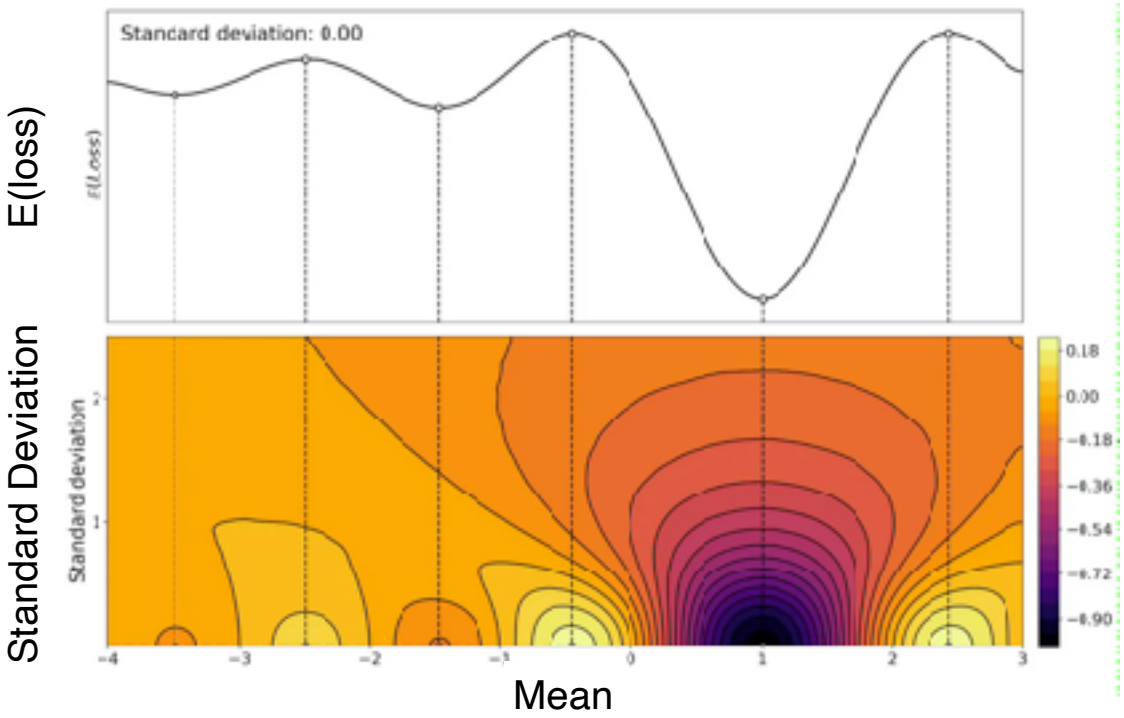
See Section 1.2, Eq 2 in Khan and Rue, 2021

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 very 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

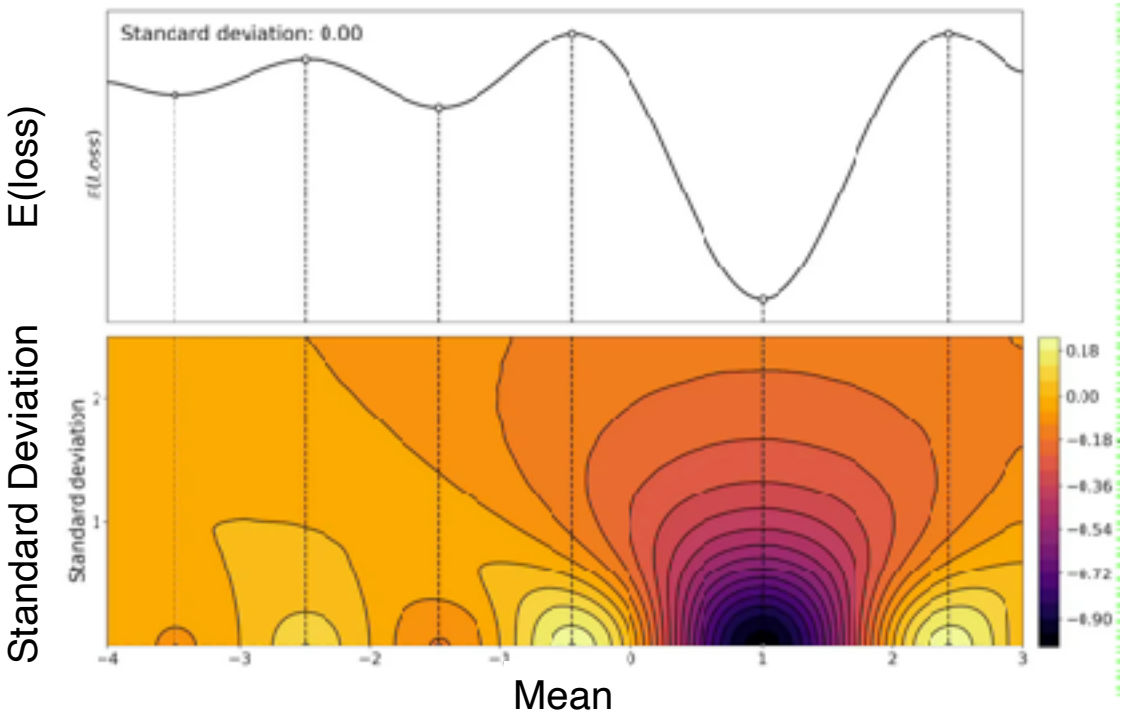
See Section 1.2, Eq 2 in Khan and Rue, 2021

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 very 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

A 2-step Bayesian Scheme

Step 1: Choose an approximation (mix-exp-family)

Natural parameters ↓ Sufficient statistics ↓

$$q(\theta) \propto \exp\left[\lambda^\top T(\theta)\right]$$

See Section 1.2 in Khan and Rue, 2021

A 2-step Bayesian Scheme

Step 1: Choose an approximation (mix-exp-family)

Natural parameters ↓ Sufficient statistics ↓

$$q(\theta) \propto \exp\left[\lambda^\top T(\theta)\right]$$

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

See Section 1.2 in Khan and Rue, 2021

A 2-step Bayesian Scheme

Step 1: Choose an approximation (mix-exp-family)

Natural parameters ↓ Sufficient statistics ↓

$$q(\theta) \propto \exp\left[\lambda^\top T(\theta)\right]$$

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

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

See Section 1.2 in Khan and Rue, 2021

A 2-step Bayesian Scheme

Step 1: Choose an approximation (mix-exp-family)

Natural parameters Sufficient statistics Expectation parameters

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

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

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

See Section 1.2 in Khan and Rue, 2021

A 2-step Bayesian Scheme

Step 1: Choose an approximation (mix-exp-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	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

See Section 1.2 in Khan and Rue, 2021

A 2-step Bayesian Scheme

Step 2: Use the Bayesian Learning Rule to optimize

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

Natural gradient (NatGrad)

See Section 1.2 in Khan and Rue, 2021

A 2-step Bayesian Scheme

Step 2: Use the Bayesian Learning Rule to optimize

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

Natural gradient (NatGrad)

Optimality condition: $\nabla_{\mu} H(q_*) = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$

See Section 1.2 in Khan and Rue, 2021

A 2-step Bayesian Scheme

Step 2: Use the Bayesian Learning Rule to optimize

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

Natural gradient (NatGrad)

Optimality condition: $\nabla_{\mu} H(q_*) = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$

For minimal Exp-Family: $-\lambda_* = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$

See Section 1.2 in Khan and Rue, 2021

A 2-step Bayesian Scheme

Step 2: Use the Bayesian Learning Rule to optimize

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

Natural gradient (NatGrad)

Optimality condition: $\nabla_{\mu} H(q_*) = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$

For minimal Exp-Family: $-\lambda_* = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$

"Information matching" due to the entropy term

1. Natural gradients are essential, & contain higher-order information about the loss, e.g., 1st and 2nd derivatives
2. This info is then assigned to appropriate natural params

See Section 1.2 in Khan and Rue, 2021

A 2-step Bayesian Scheme

Step 2: Use the Bayesian Learning Rule to optimize

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

Natural gradient (NatGrad)

Optimality condition: $\nabla_{\mu} H(q_*) = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$

For minimal Exp-Family: $-\lambda_* = \nabla_{\mu} \mathbb{E}_{q_*}[\ell(\theta)]$

"Information matching" due to the entropy term

1. Natural gradients are essential, & contain higher-order information about the loss, e.g., 1st and 2nd derivatives
2. This info is then assigned to appropriate natural params

By changing *Q* & approx. to natGrads, we can choose the kind of "information" and recover many learning algorithms.

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

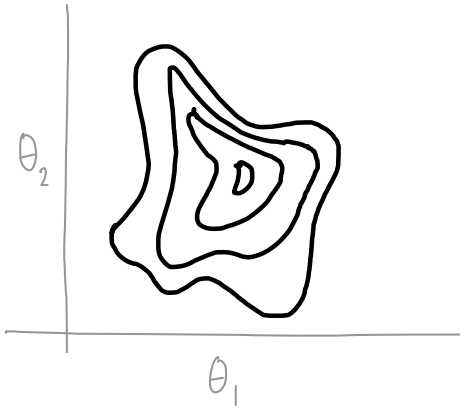
Complex $\longleftrightarrow$ Simple

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 $\longleftrightarrow$ Simple

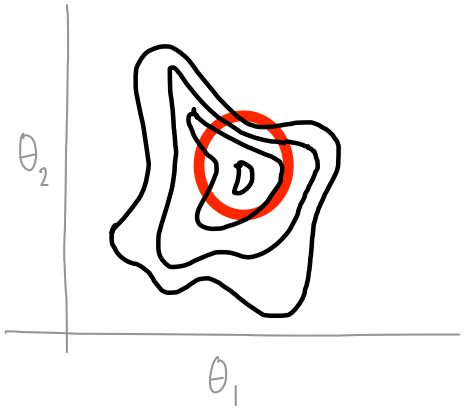


See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple

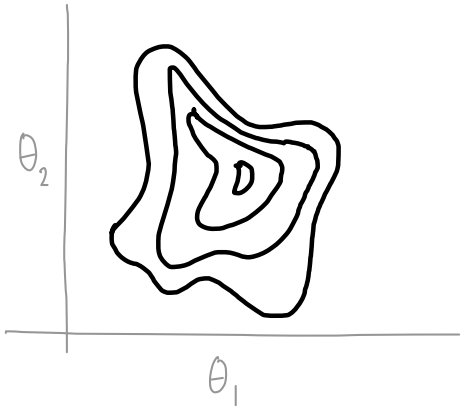


See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple

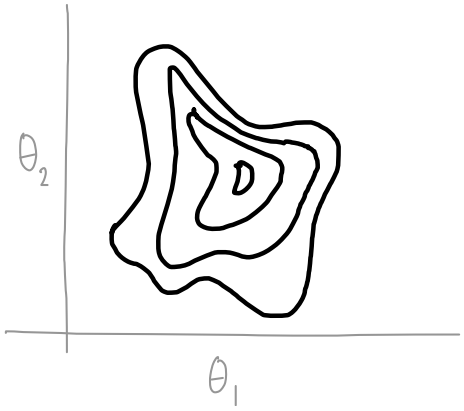


See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple



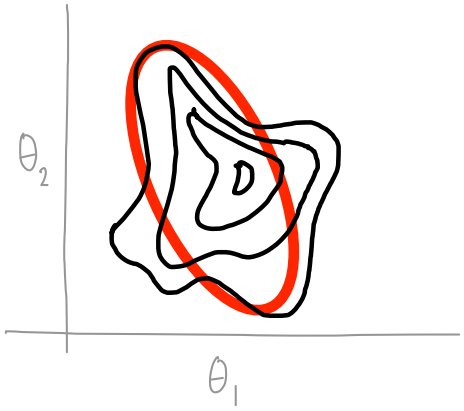
Gradient Descent

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 $\longleftrightarrow$ Simple



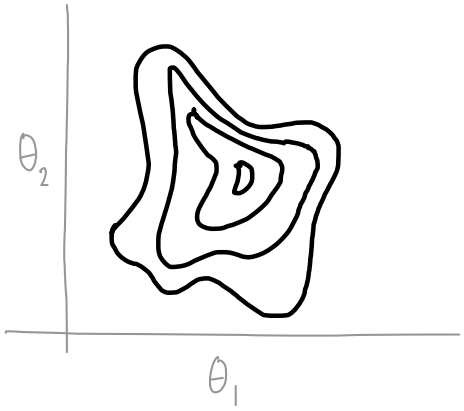
Gradient Descent

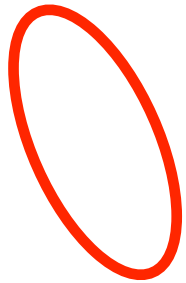
See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple





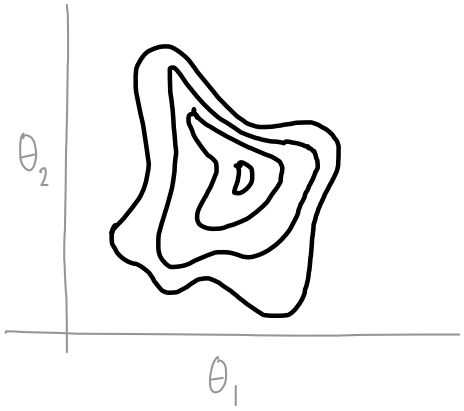
Gradient Descent

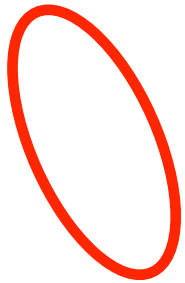
See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple





Newton

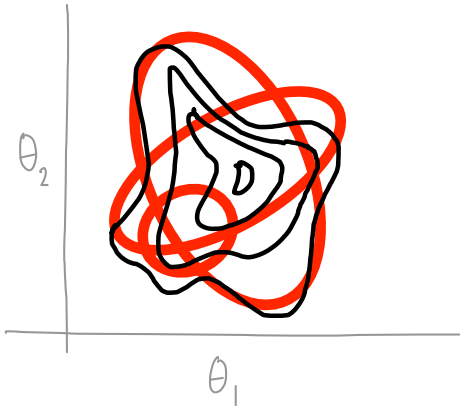
Gradient Descent

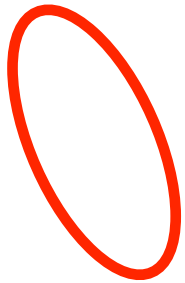
See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple





Newton

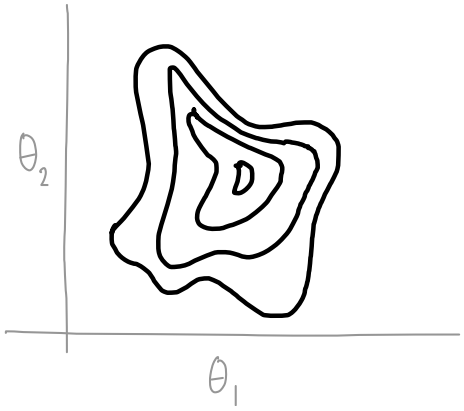
Gradient Descent

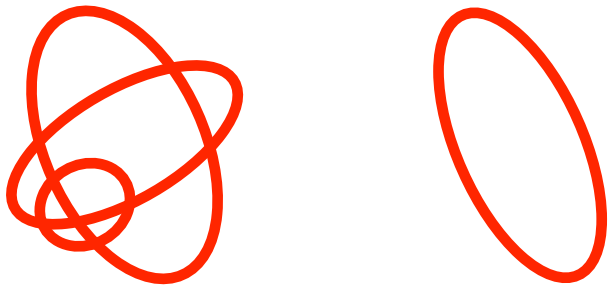
See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftarrow$ Simple





Newton

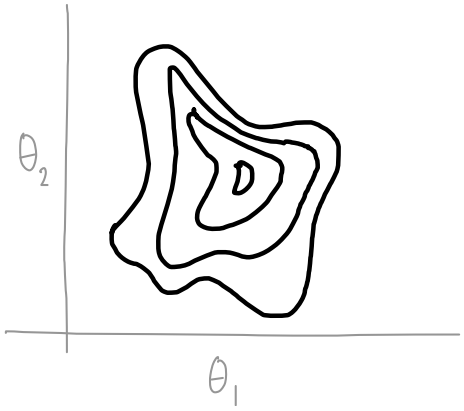
Gradient Descent

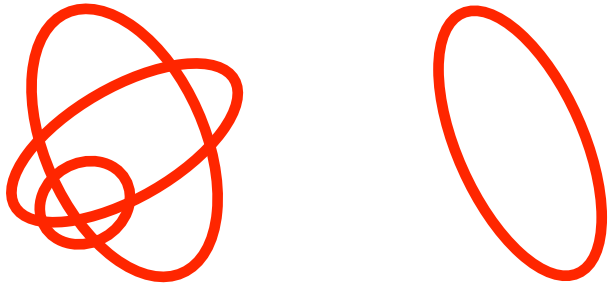
See Section 1.3 and 3.2 in Khan and Rue, 2021

Deriving Learning-Algorithms from the Bayesian Learning Rule

Posterior Approximation $\longleftrightarrow$ Learning-Algorithm

Complex $\longleftrightarrow$ Simple





Mixture of Newton

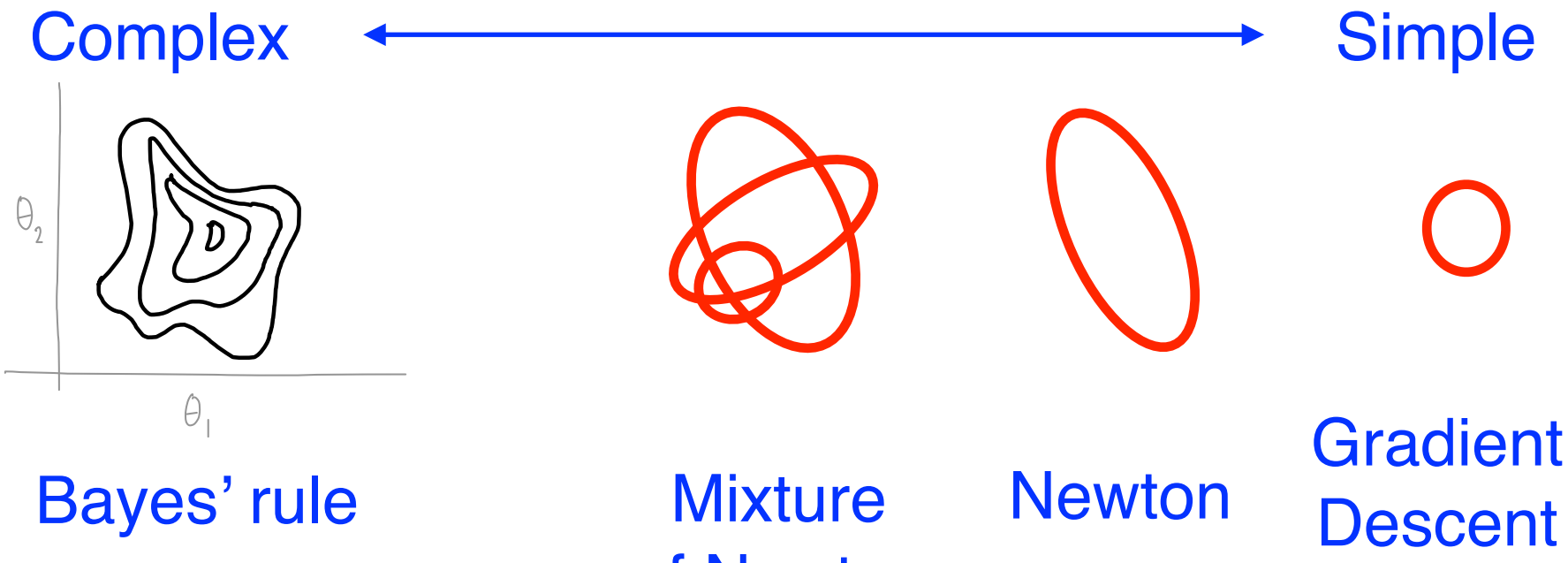
Newton

Gradient Descent

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



See Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

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

See Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

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

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 Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

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

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

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 Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

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

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

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 Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from Bayes

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

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

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

We can compute uncertainty using a variant of Adam.

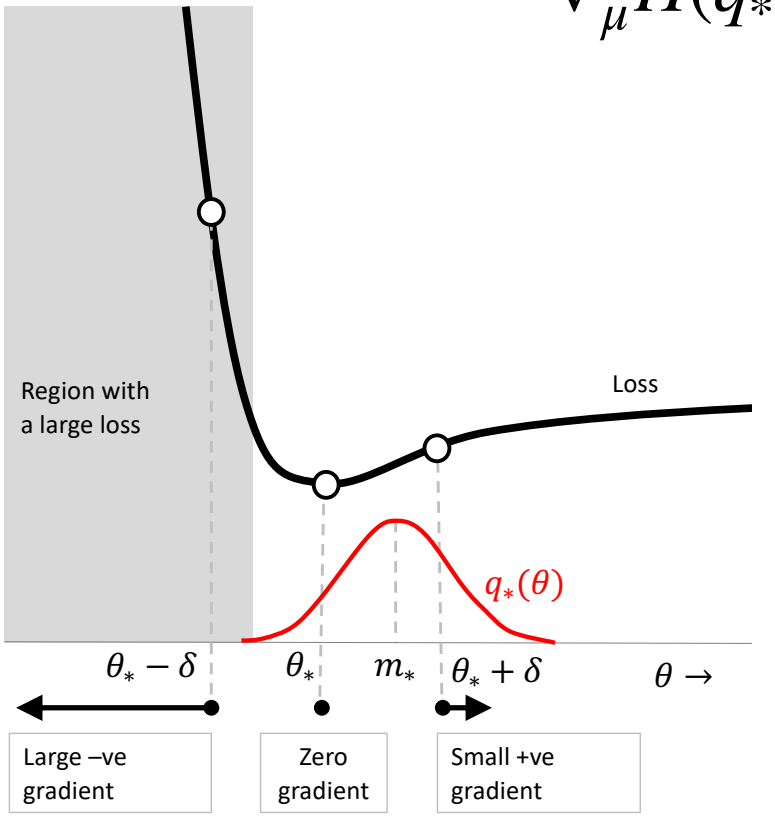
1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." NeurIPS (2019).

See Section 4.4 in Khan and Rue, 2021

Bayes leads to robust solutions

Avoiding large losses

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



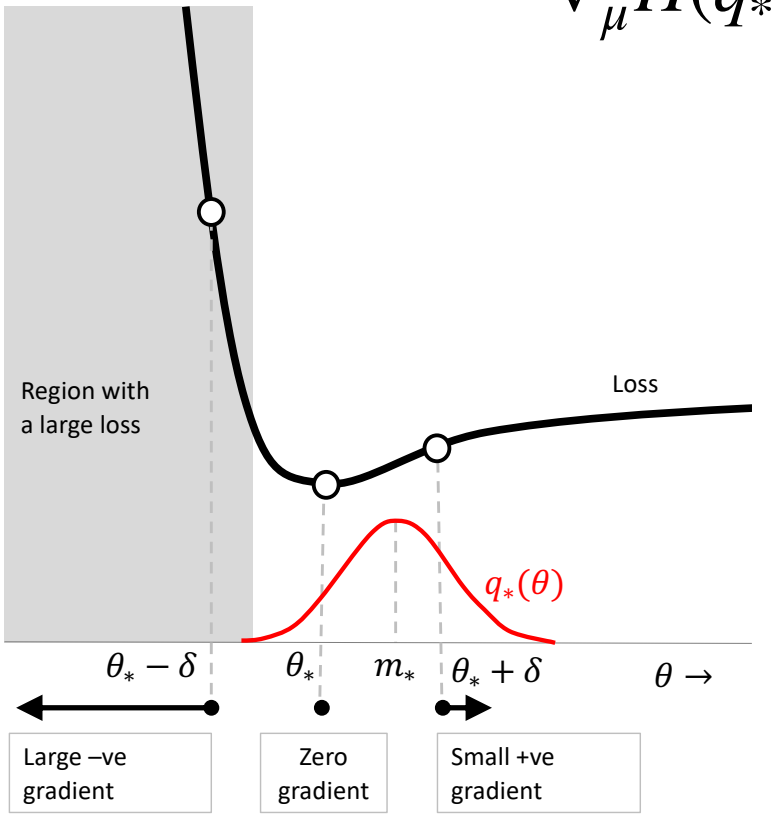
See Section 4.4 in Khan and Rue, 2021

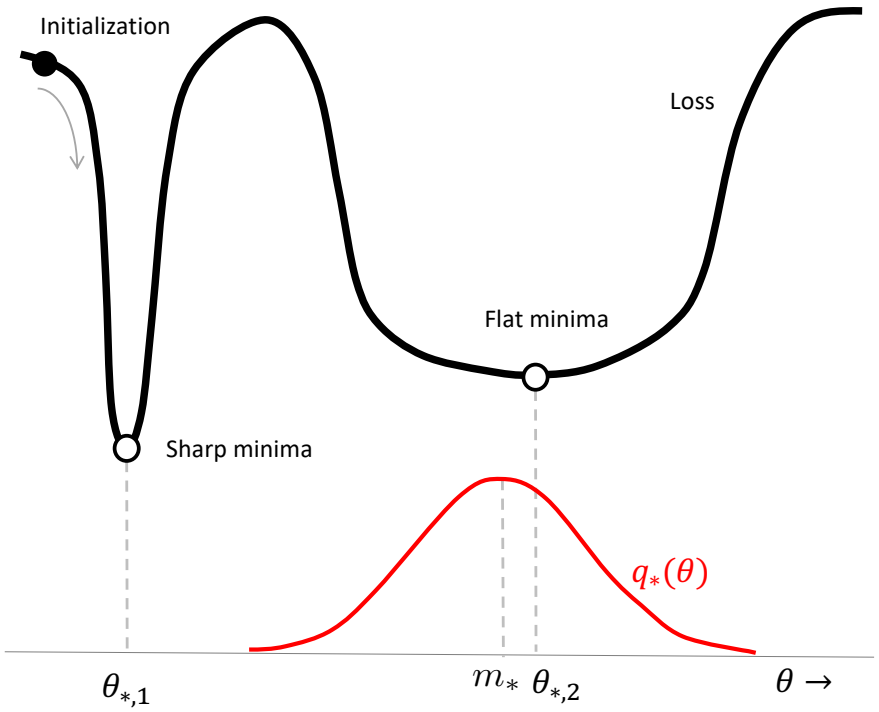
Bayes leads to robust solutions

Avoiding large losses

Avoiding sharp minima

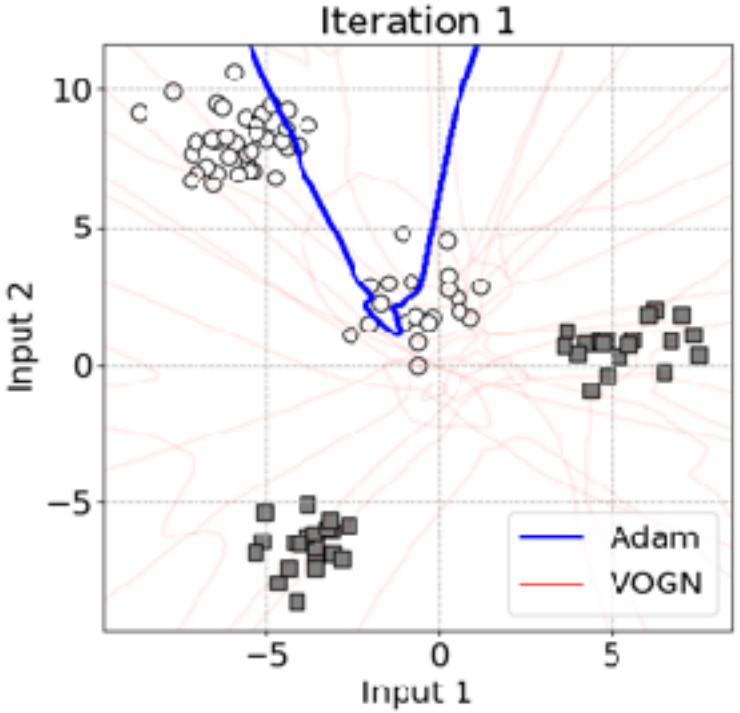
$$\nabla_\mu H(q_*) = \nabla_\mu \mathbb{E}_{q_*}[\ell(\theta)]$$

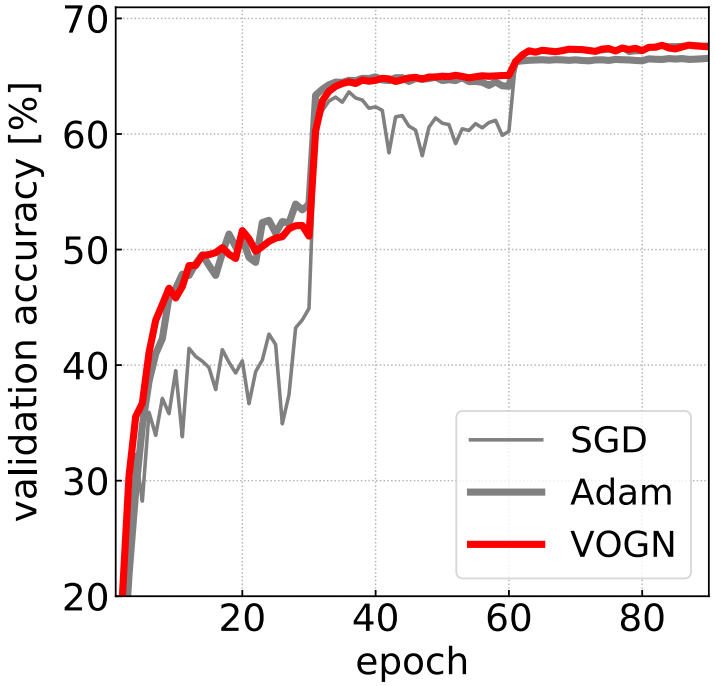




Uncertainty of Deep Nets

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



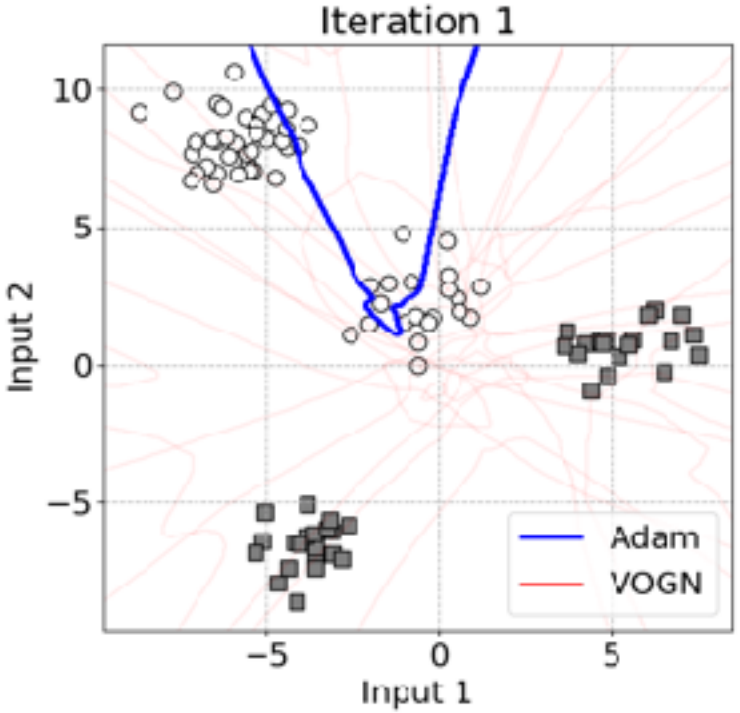


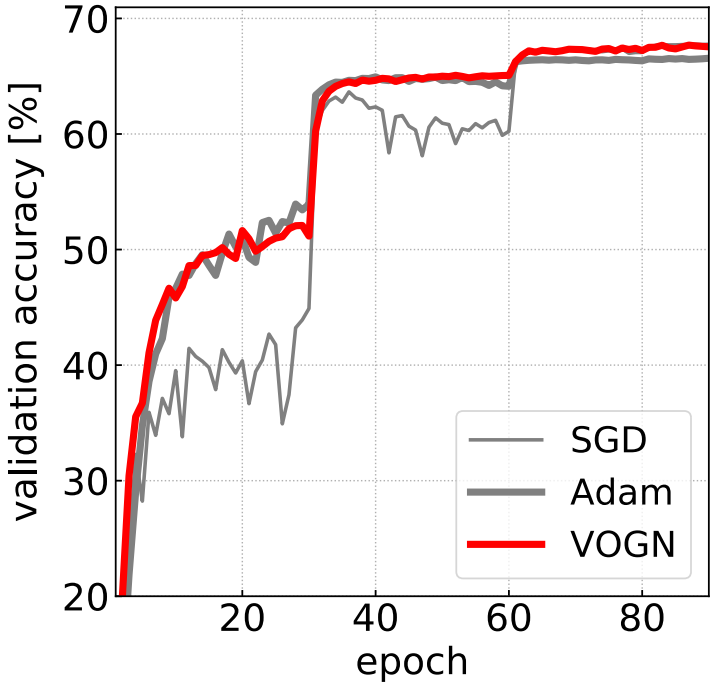
Code available at https://github.com/team-approx-bayes/dl-with-bayes

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).

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).

See Section 4.2 in Khan and Rue, 2021

RMSprop/Adam from Bayes

RMSprop

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

BLR for Gaussian approx

$$S \leftarrow (1-\rho)S + \rho(H_\theta)$$
$$m \leftarrow m - \alpha S^{-1}\nabla_\theta \ell(\theta)$$

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

See Section 4.2 in Khan and Rue, 2021

RMSprop/Adam from Bayes

RMSprop

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

BLR for Gaussian approx

$$S \leftarrow (1-\rho)S + \rho(H_\theta)$$
$$m \leftarrow m - \alpha S^{-1}\nabla_\theta \ell(\theta)$$

To get RMSprop, make the following choices

- Restrict covariance to be diagonal
- Replace Hessian by square of gradients
- Add square root for scaling vector

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

See Section 4.2 in Khan and Rue, 2021

RMSprop/Adam from Bayes

RMSprop

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

BLR for Gaussian approx

$$S \leftarrow (1-\rho)S + \rho(H_\theta)$$
$$m \leftarrow m - \alpha S^{-1}\nabla_\theta \ell(\theta)$$

To get RMSprop, make the following choices

- Restrict covariance to be diagonal
- Replace Hessian by square of gradients
- Add square root for scaling vector

For Adam, use a Heavy-ball term with KL divergence as momentum (Appendix E in [1])

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

Variational Online Gauss-Newton

RMSprop

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

VOGN

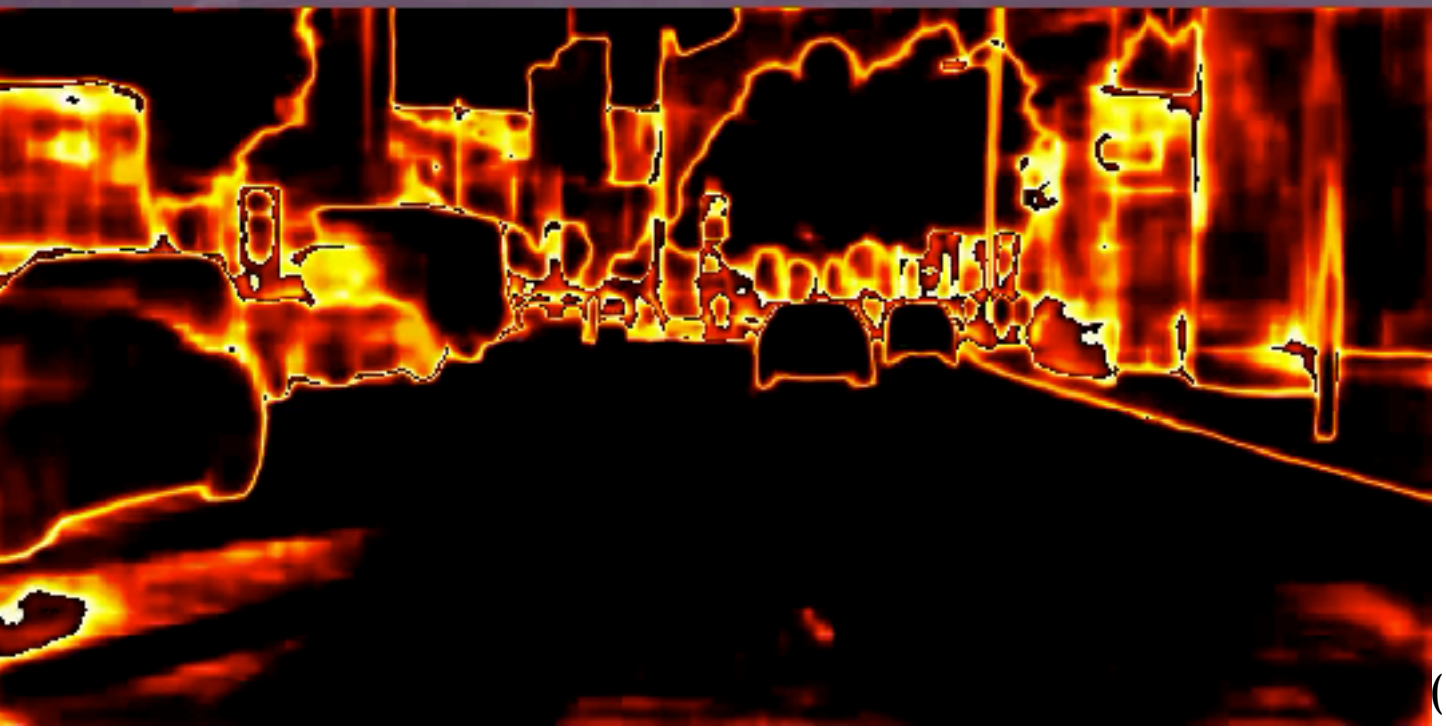
$$g \leftarrow \hat{\nabla}\ell(\theta), \text{ where } \theta \sim \mathcal{N}(m, \sigma^2)$$
$$s \leftarrow (1-\rho)s + \rho(\Sigma_i g_i^2)$$
$$m \leftarrow m - \alpha(s+\gamma)^{-1}\nabla_\theta \ell(\theta)$$
$$\sigma^2 \leftarrow (s+\gamma)^{-1}$$

```
import torch
+import torchsso

train_loader = torch.utils.data.DataLoader(train_dataset)
model = MLP()

-optimizer = torch.optim.Adam(model.parameters())
+optimizer = torchsso.optim.VOGN(model, dataset_size=len(train_loader.dataset))
```

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).

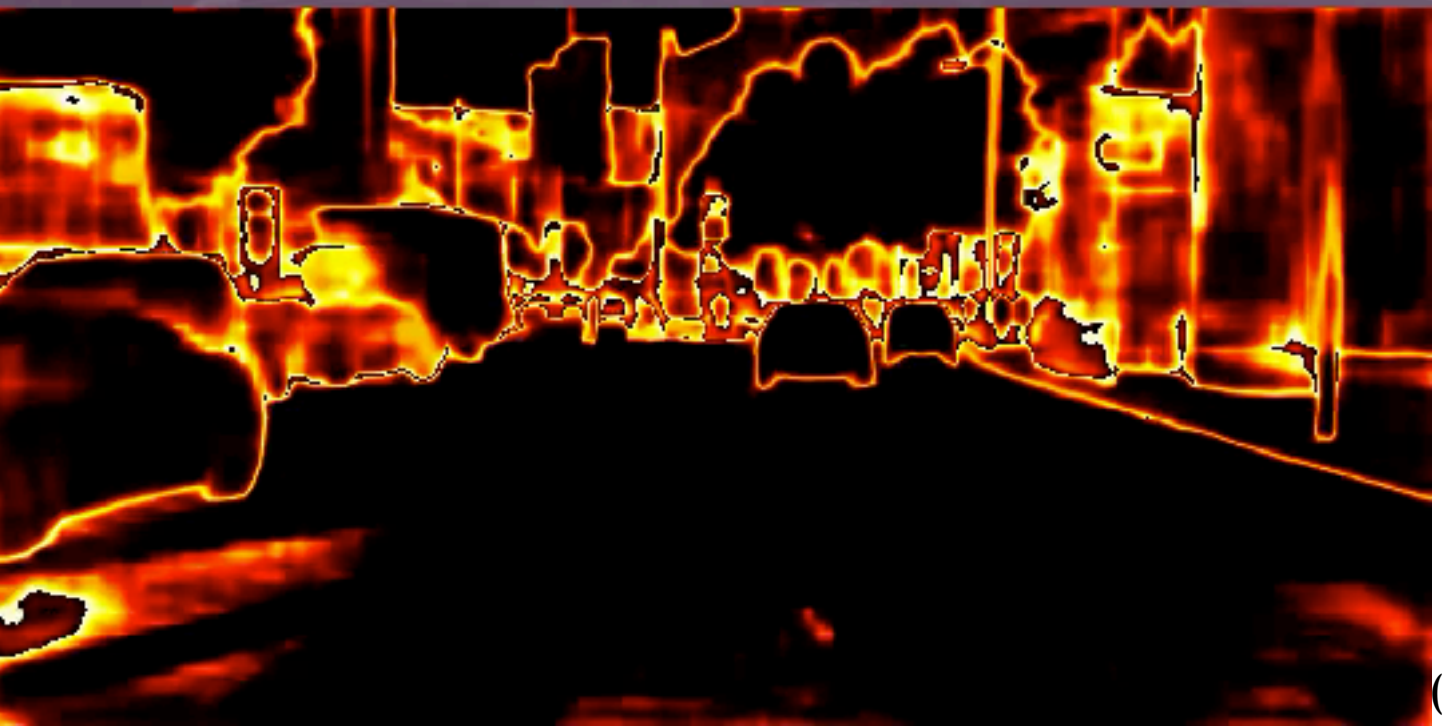
Image Segmentation



Uncertainty (entropy of class probs)

(By Roman Bachmann)

Image Segmentation



Uncertainty (entropy of class probs)

(By Roman Bachmann)

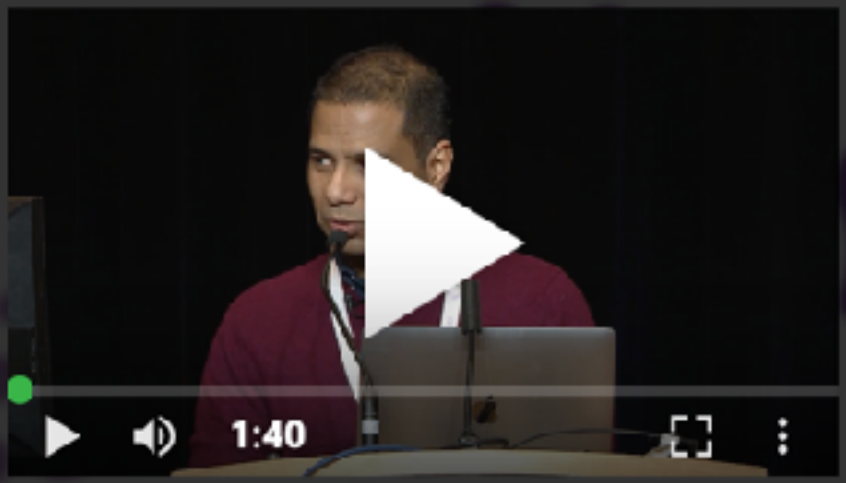
Tuning VOGN

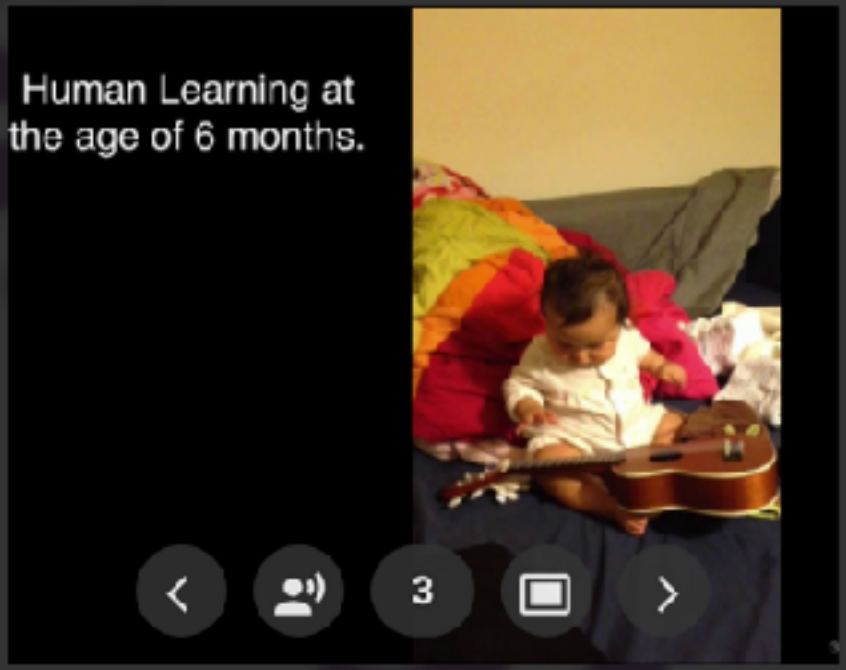
The trick is to mimic Adam's trajectory as closely as possible

Tuning VOGN: Currently, there is no common recipe for tuning the algorithmic hyperparameters for VI, especially for large-scale tasks like ImageNet classification. One key idea we use in our experiments is to start with Adam hyperparameters and then make sure that VOGN training closely follows an Adam-like trajectory in the beginning of training. To achieve this, we divide the tuning into an *optimisation part* and a *regularisation part*. In the *optimisation part*, we first tune the hyperparameters of a deterministic version of VOGN, called the online Gauss-Newton (OGN) method. This method, described in Appendix C, is more stable than VOGN since it does not require MC sampling, and can be used as a stepping stone when moving from Adam/SGD to VOGN. After reaching a competitive performance to Adam/SGD by OGN, we move to the *regularisation part*, where we tune the prior precision δ, the tempering parameter τ, and the number of MC samples K for VOGN. We initialise our search by setting the prior precision δ using the L2-regularisation parameter used for OGN, as well as the dataset size N. Another technique is to warm-up the parameter τ towards $\tau = 1$ (also see the "momentum and initialisation" part). Setting τ to smaller values usually stabilises the training, and increasing it slowly also helps during tuning. We also add an *external damping factor* $\gamma > 0$ to the moving average $\mathbf{s}_t$. This increases the lower bound of the eigenvalues of the diagonal covariance $\mathbf{\Sigma}_t$ and prevents the noise and the step size from becoming too large. We find that a mix of these techniques works well for the problems we considered.

Sec 3, last paragraph in Osawa et al. "Practical Deep Learning with Bayesian Principles." NeurIPS (2019).

NeurIPS 2019 Tutorial





Deep Learning with Bayesian Principles

by **Mohammad Emtiyaz Khan** · Dec 9, 2019 ·

#NeurIPS 2019 Follow

Views 151 807 Presentations 263 Followers 200

Latest Popular ···

From System 1 Deep Learning to System 2 Deep Learning

by Yoshua Bengio

17,953 views · Dec 11, 2019

NeurIPS Workshop on Machine Learning for Creativity and Design...

by Aaron Hertzmann, Adam Roberts, ...

9,656 views · Dec 14, 2019

Deep Learning with Bayesian Principles

by Mohammad Emtiyaz Khan

8,084 views · Dec 9, 2019

Efficient Processing of Deep Neural Network: from Algorithms to...

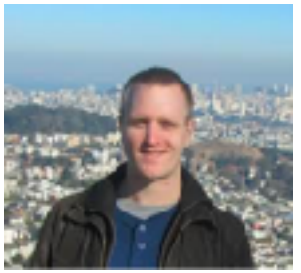
by Vivienne Sze

7,163 views · Dec 9, 2019

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)

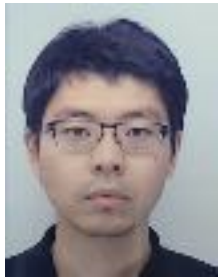


Mark Schmidt (UBC)

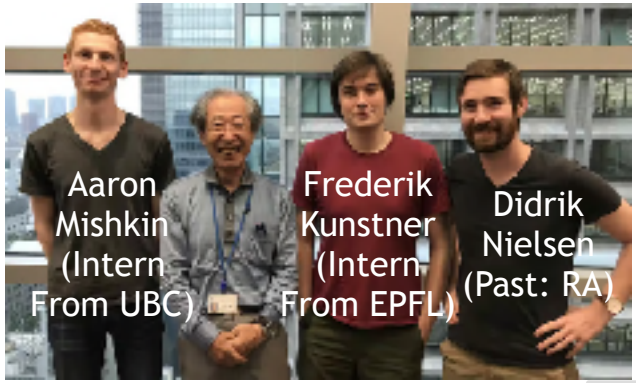
Frank Nielsen (Sony)

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).

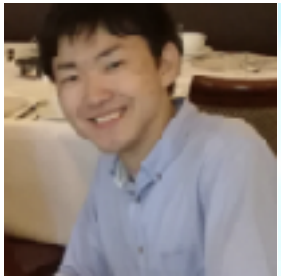


Voot Tangkaratt
(Postdoc, RIKEN-AIP)



Yarin Gal
(UOxford)

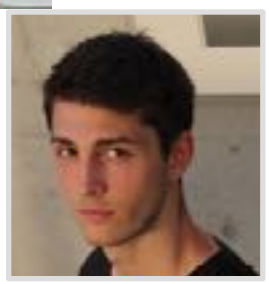
Akash Srivastava
(UEdinburgh)



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)

Dual Perspective of the Bayesian Learning Rule

Memorable Examples

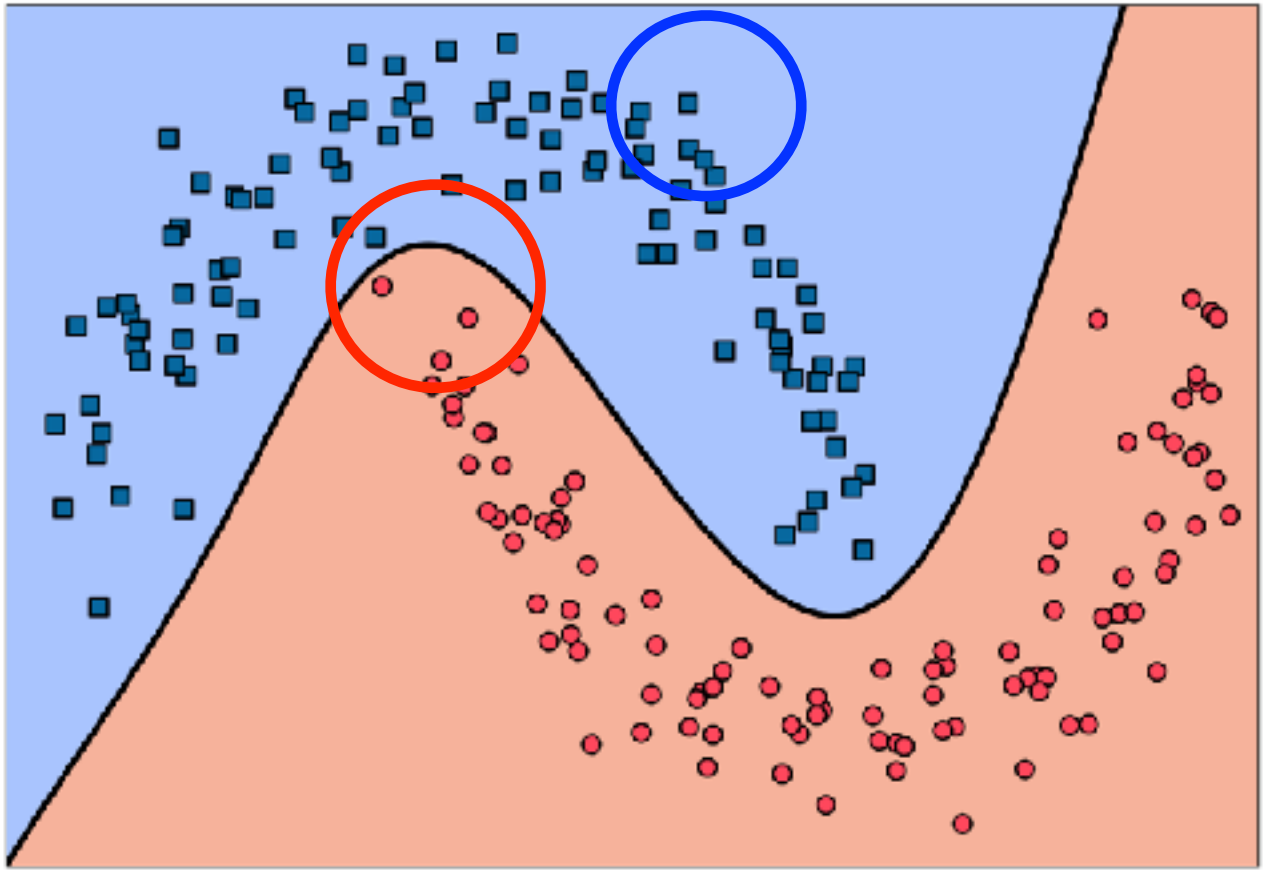
Connections to Gaussian Process

Continual learning

Adaptation with K-priors

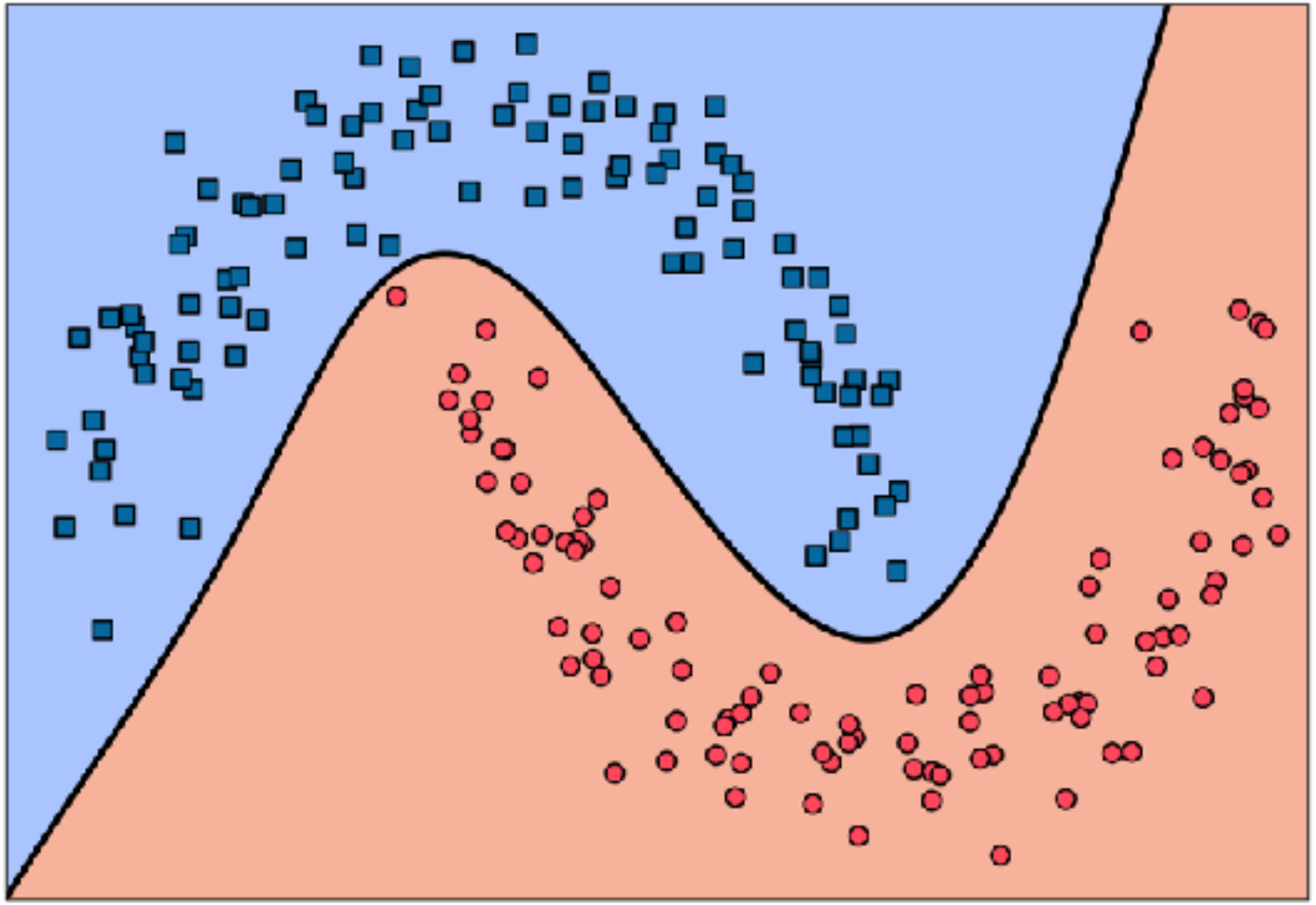
Relevance of Data Examples

Which examples are most relevant for the classifier? Red circle vs Blue circle.



Model view vs Data view

Bayes “automatically” defines data-relevance

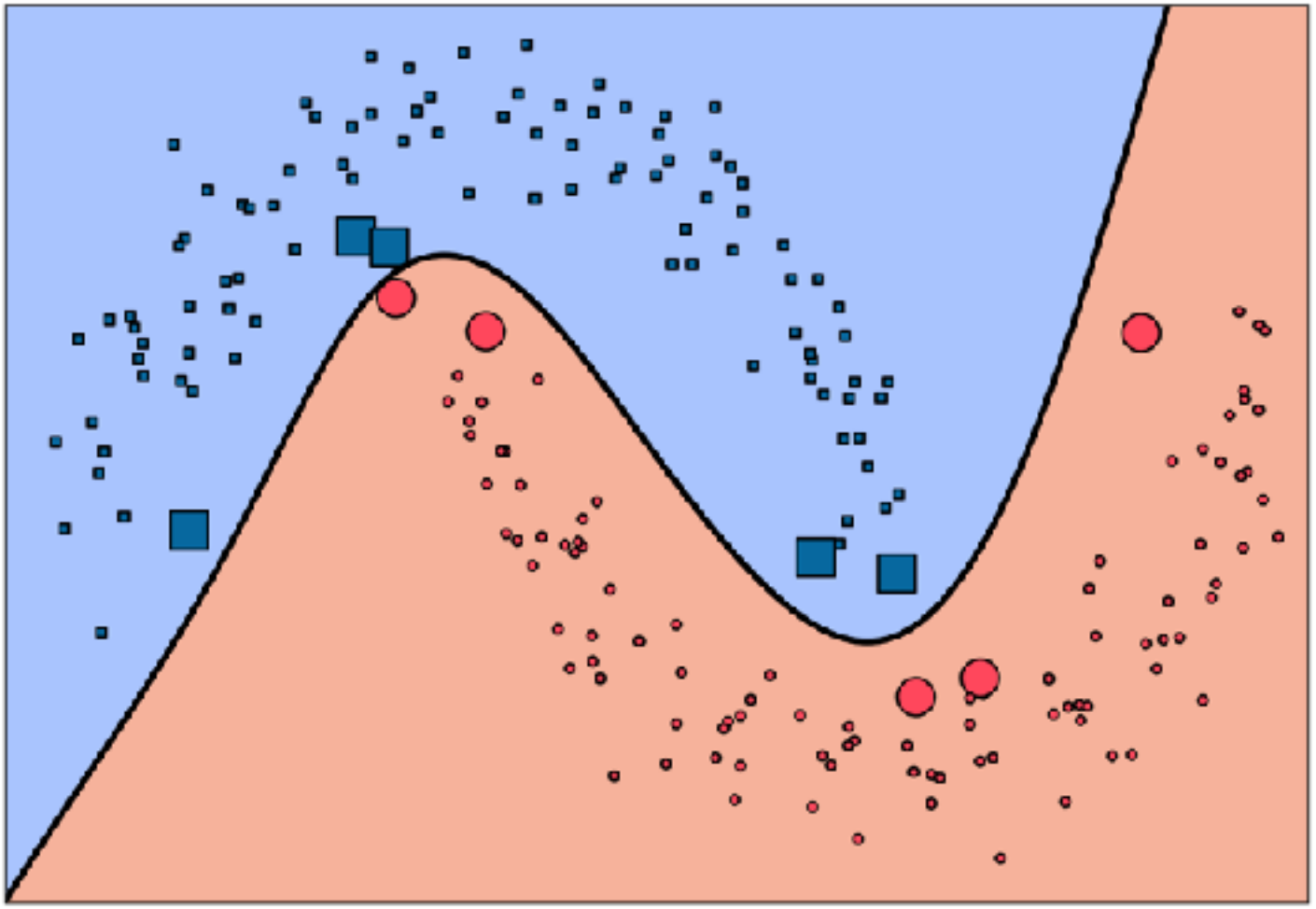


Model view

(By Roman Bachmann)

Model view vs Data view

Bayes “automatically” defines data-relevance



Data view (Very much like SVMs)

(By Roman Bachmann)

Bayes Duality

- Gaussian approx fom Bayes learning rule turn NN into Linear models & Gaussian Process (GPs) [1].

$$\sum_{i=1}^{N} \ell(y_i, f_\theta(x_i))$$

neural network

1. Khan et al. "Approximate Inference Turns Deep Networks into Gaussian Processes." *NeurIPS* (2019).

Bayes Duality

- Gaussian approx fom Bayes learning rule turn NN into Linear models & Gaussian Process (GPs) [1].

$$\sum_{i=1}^{N} \ell(y_i, f_\theta(x_i)) \quad \approx \sum_{i=1}^{N} \frac{1}{\sigma_i^2} [\tilde{y}_i - \phi_i(x_i)^\top \theta]^2$$

neural network

1. Khan et al. "Approximate Inference Turns Deep Networks into Gaussian Processes." *NeurIPS* (2019).

Bayes Duality

- Gaussian approx fom Bayes learning rule turn NN into Linear models & Gaussian Process (GPs) [1].

$$\sum_{i=1}^{N} \ell(y_i, f_\theta(x_i)) \quad \approx \sum_{i=1}^{N} \frac{1}{\sigma_i^2} [\tilde{y}_i - \phi_i(x_i)^\top \theta]^2$$

neural network

"Dual" variables obtained from $\nabla_\mu \mathbb{E}_q[\ell_i(\theta)]$

(For Gaussian approx, obtained from Jacobian, residual etc.)

1. Khan et al. "Approximate Inference Turns Deep Networks into Gaussian Processes." *NeurIPS* (2019).

Bayes Duality

- Gaussian approx fom Bayes learning rule turn NN into Linear models & Gaussian Process (GPs) [1].

$$\sum_{i=1}^{N} \ell(y_i, f_\theta(x_i)) \quad \approx \sum_{i=1}^{N} \frac{1}{\sigma_i^2} [\tilde{y}_i - \phi_i(x_i)^\top \theta]^2$$

neural network

"Dual" variables obtained from $\nabla_\mu \mathbb{E}_q[\ell_i(\theta)]$

(For Gaussian approx, obtained from Jacobian, residual etc.)

- σ_i^2 define the "relevance" of the data examples. We call more relevant ones the "memorable examples".

1. Khan et al. "Approximate Inference Turns Deep Networks into Gaussian Processes." *NeurIPS* (2019).

Bayes Duality

- Gaussian approx fom Bayes learning rule turn NN into Linear models & Gaussian Process (GPs) [1].

$$\sum_{i=1}^{N} \ell(y_i, f_\theta(x_i)) \quad \approx \sum_{i=1}^{N} \frac{1}{\sigma_i^2} [\tilde{y}_i - \phi_i(x_i)^\top \theta]^2$$

neural network

"Dual" variables obtained from $\nabla_\mu \mathbb{E}_q[\ell_i(\theta)]$

(For Gaussian approx, obtained from Jacobian, residual etc.)

- σ_i^2 define the "relevance" of the data examples. We call more relevant ones the "memorable examples".

1. Khan et al. "Approximate Inference Turns Deep Networks into Gaussian Processes." *NeurIPS* (2019).

Bayes Duality

- Gaussian approx fom Bayes learning rule turn NN into Linear models & Gaussian Process (GPs) [1].

$$\sum_{i=1}^{N} \ell(y_i, f_\theta(x_i)) \quad \approx \sum_{i=1}^{N} \frac{1}{\sigma_i^2} [\tilde{y}_i - \phi_i(x_i)^\top \theta]^2$$

neural network

"Dual" variables obtained from $\nabla_\mu \mathbb{E}_q[\ell_i(\theta)]$

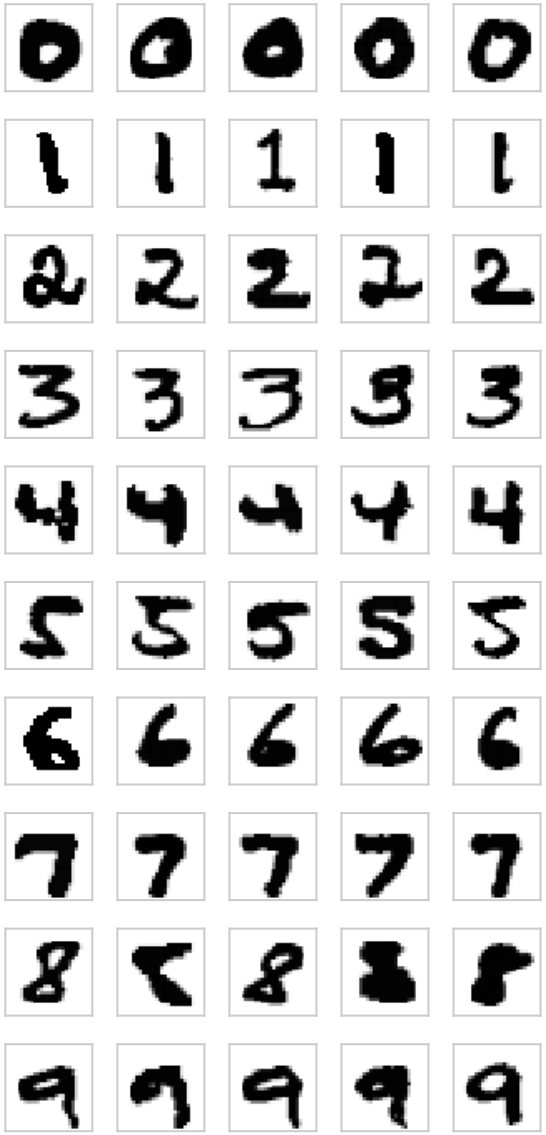
(For Gaussian approx, obtained from Jacobian, residual etc.)

- σ_i^2 define the "relevance" of the data examples. We call more relevant ones the "memorable examples".

- Natural-gradients give "dual variables" (Bayes Duality)

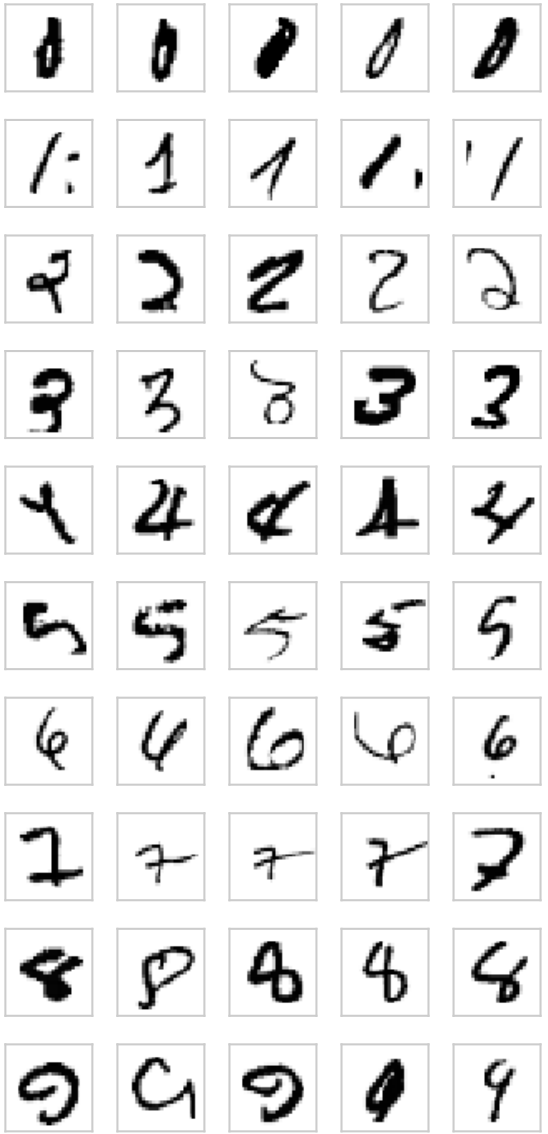
1. Khan et al. "Approximate Inference Turns Deep Networks into Gaussian Processes." *NeurIPS* (2019).

Least Memorable

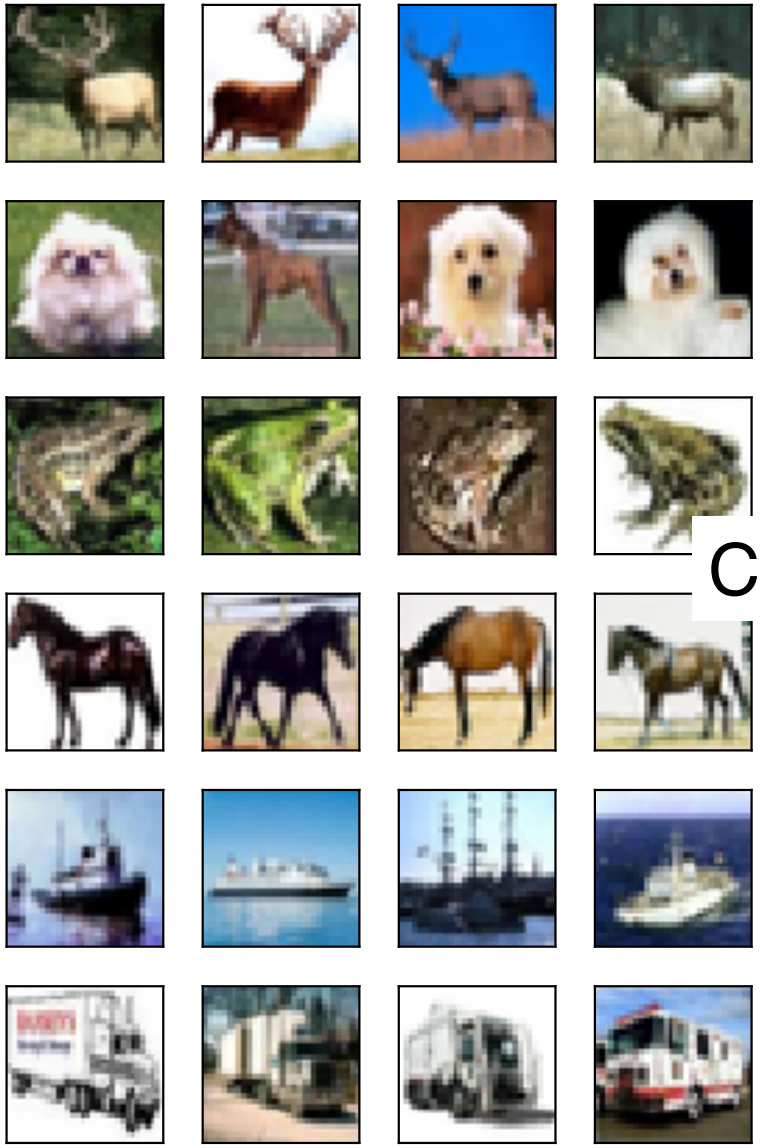


MNIST

Most Memorable

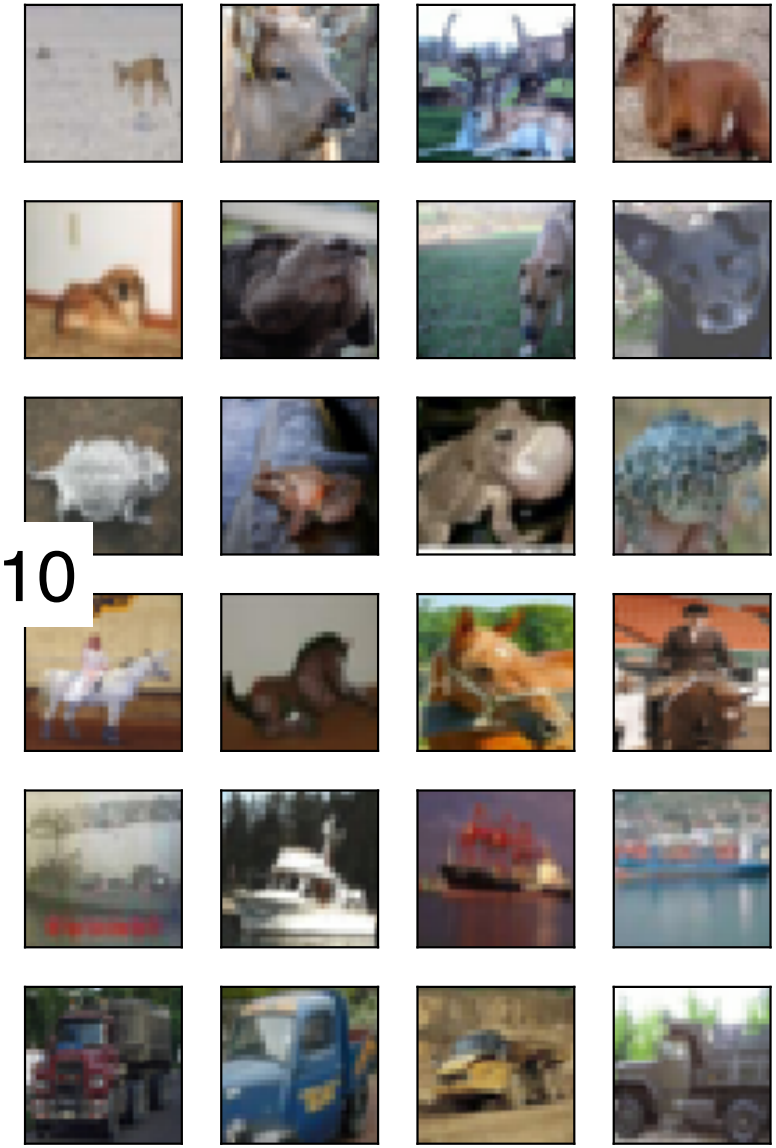


Least Memorable



CIFAR-10

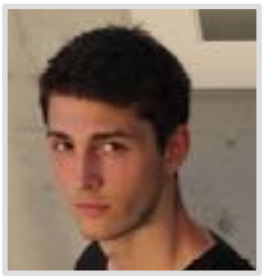
Most Memorable



Continual Learning with Bayes

PingBo Pan
(Intern from UT Sydney)

Siddharth Swaroop
(University of Cambridge)



Runa Eschenhagen
(Intern from University of Osnabruck)

Rich Turner
(University of Cambridge)

Alexander Immer
(Intern from EPFL)

Ehsan Abedi
(Intern from EPFL)

Maciej Korzepa
(Intern from DTU)

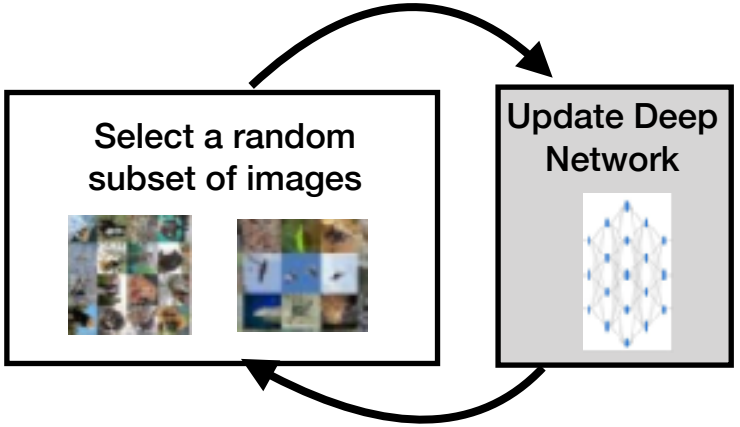
1. Khan et al. “Approximate Inference Turns Deep Networks into Gaussian Process”, NeurIPS, 2019
2. *Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020*

Continual Learning

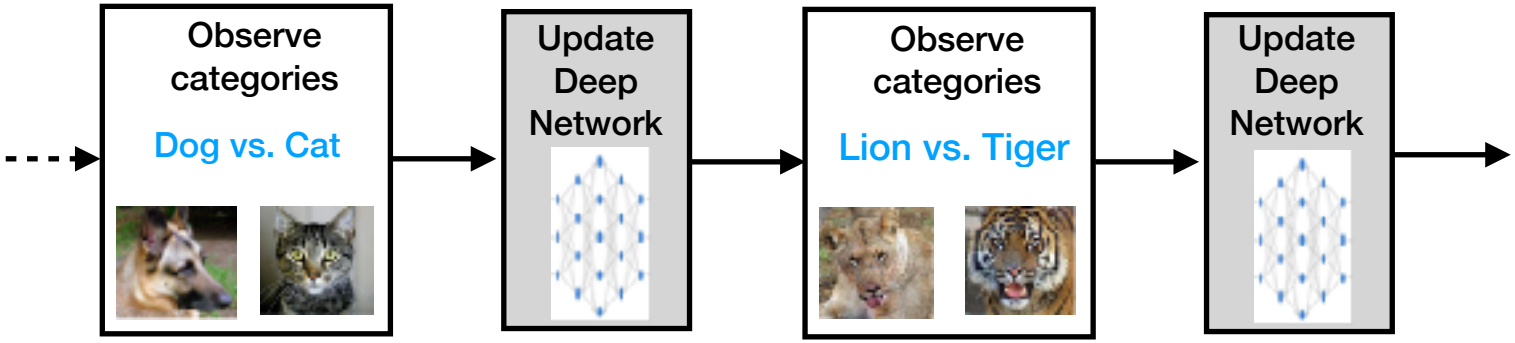
Standard Deep Learning

Continual Learning

Standard Deep Learning



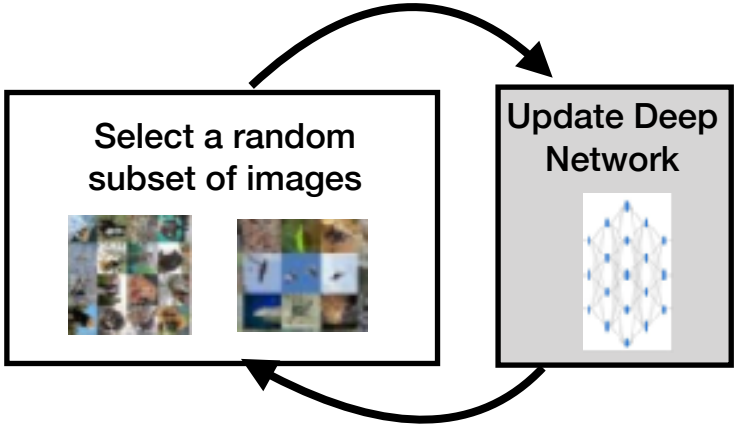
Continual Learning: past classes never revisited



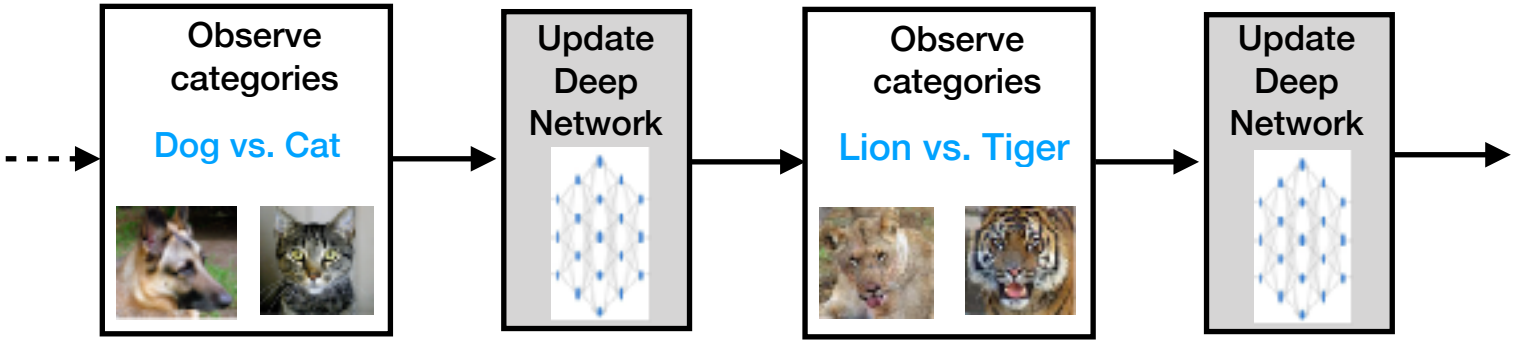
Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *Proceedings of the national academy of sciences* 114.13 (2017): 3521-3526.

Continual Learning

Standard Deep Learning



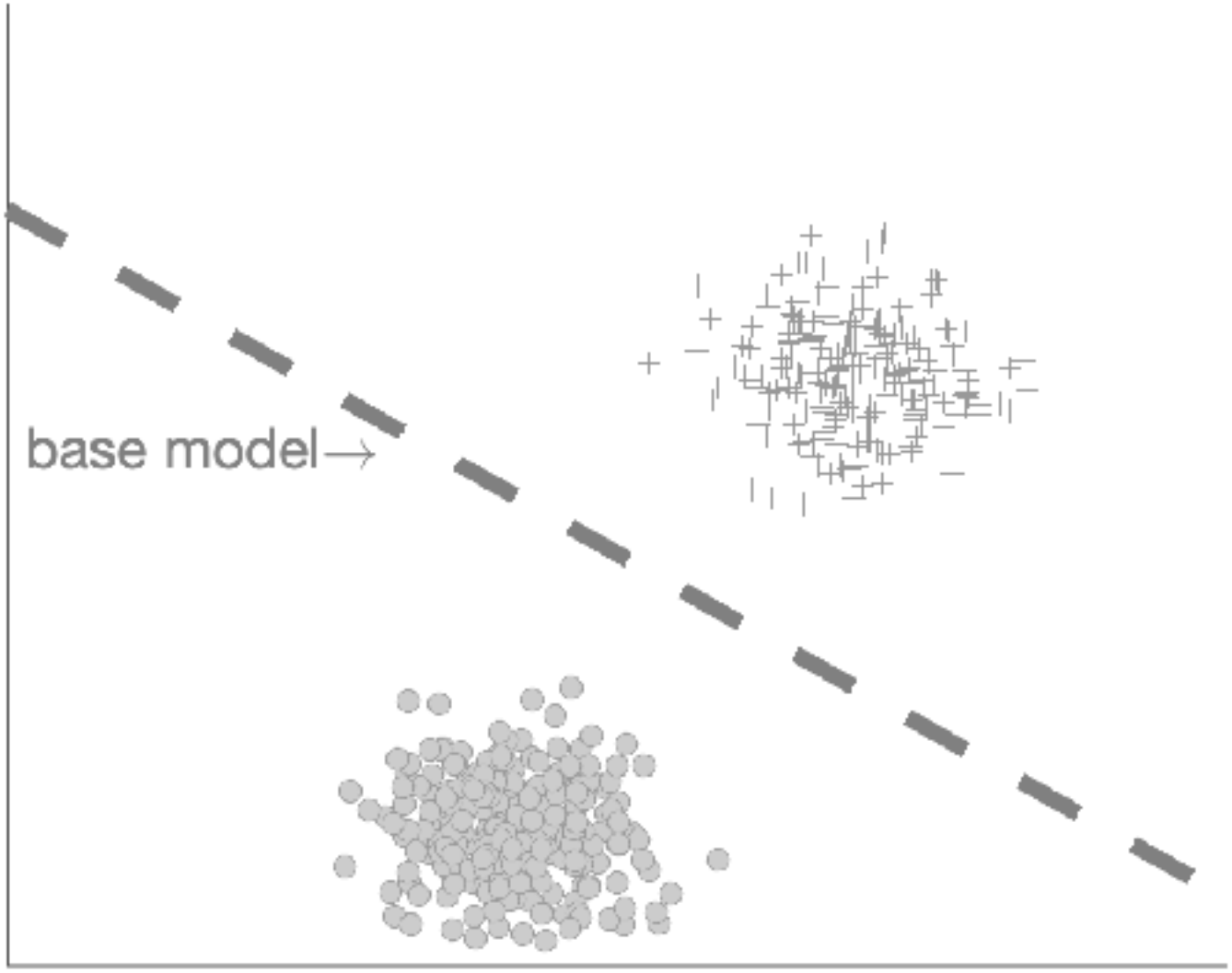
Continual Learning: past classes never revisited



Standard training leads to catastrophic forgetting.

Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *Proceedings of the national academy of sciences* 114.13 (2017): 3521-3526.

Inaccuracy of Weight-Priors



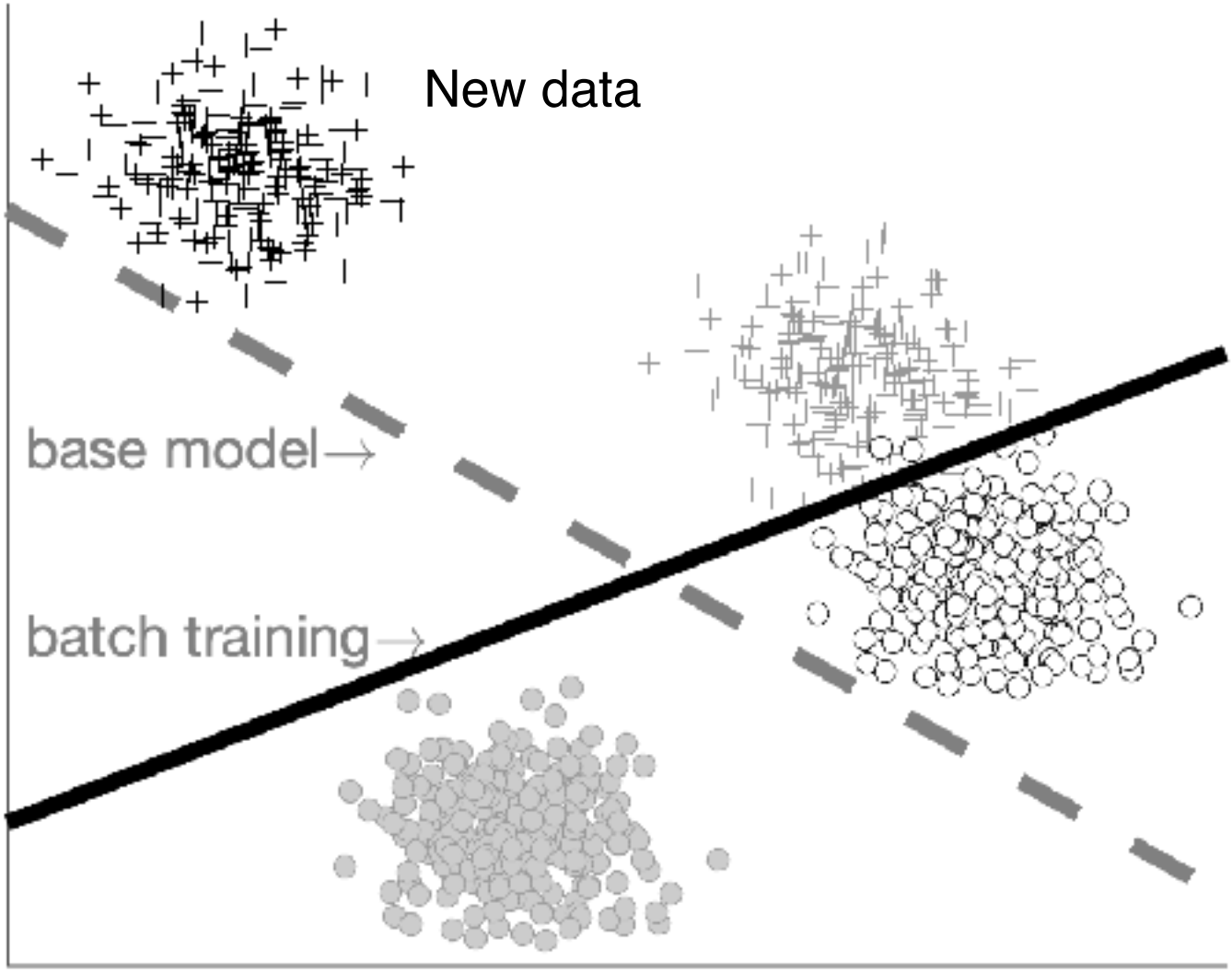
'Add Data' task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.

Inaccuracy of Weight-Priors



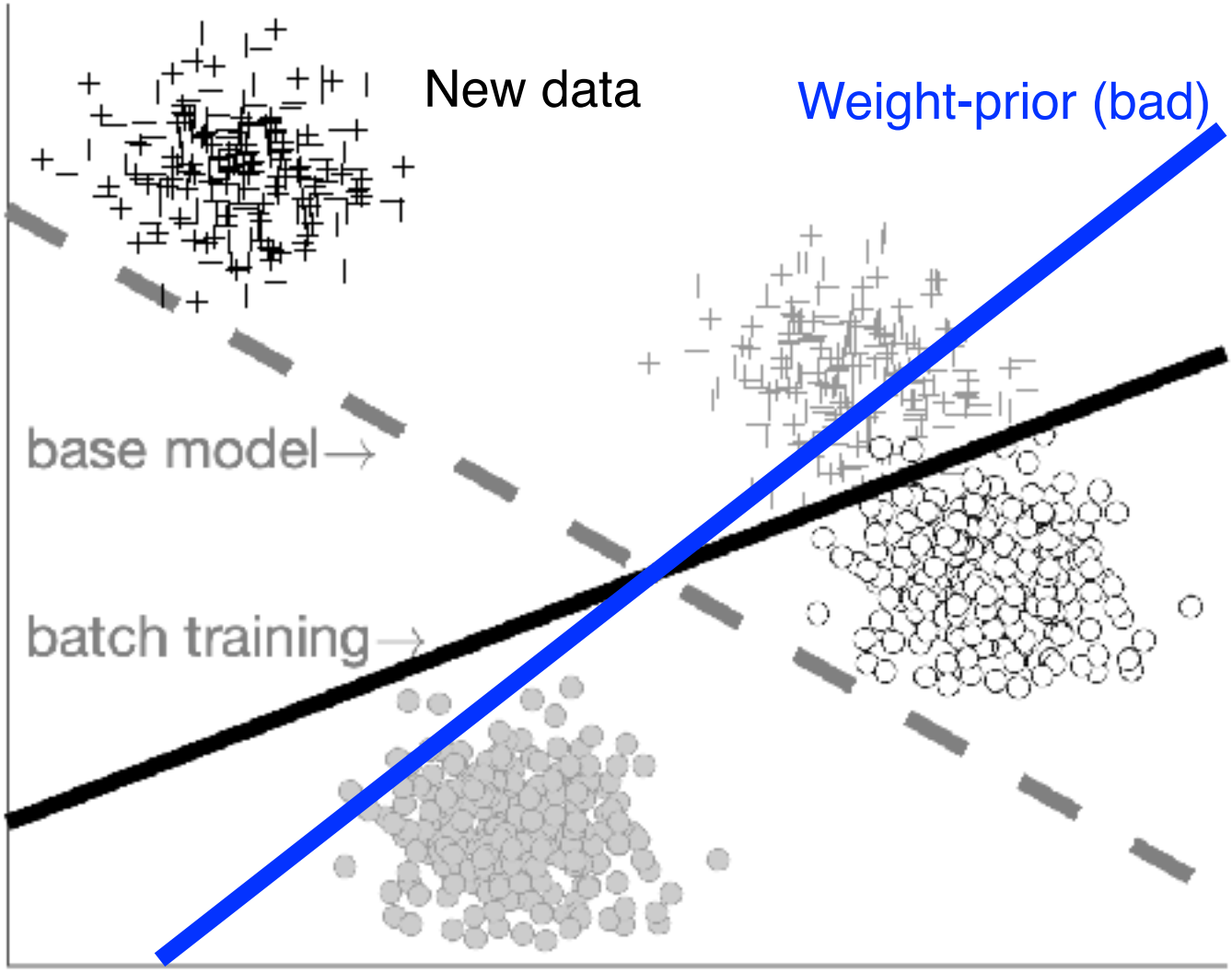
'Add Data' task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.

Inaccuracy of Weight-Priors



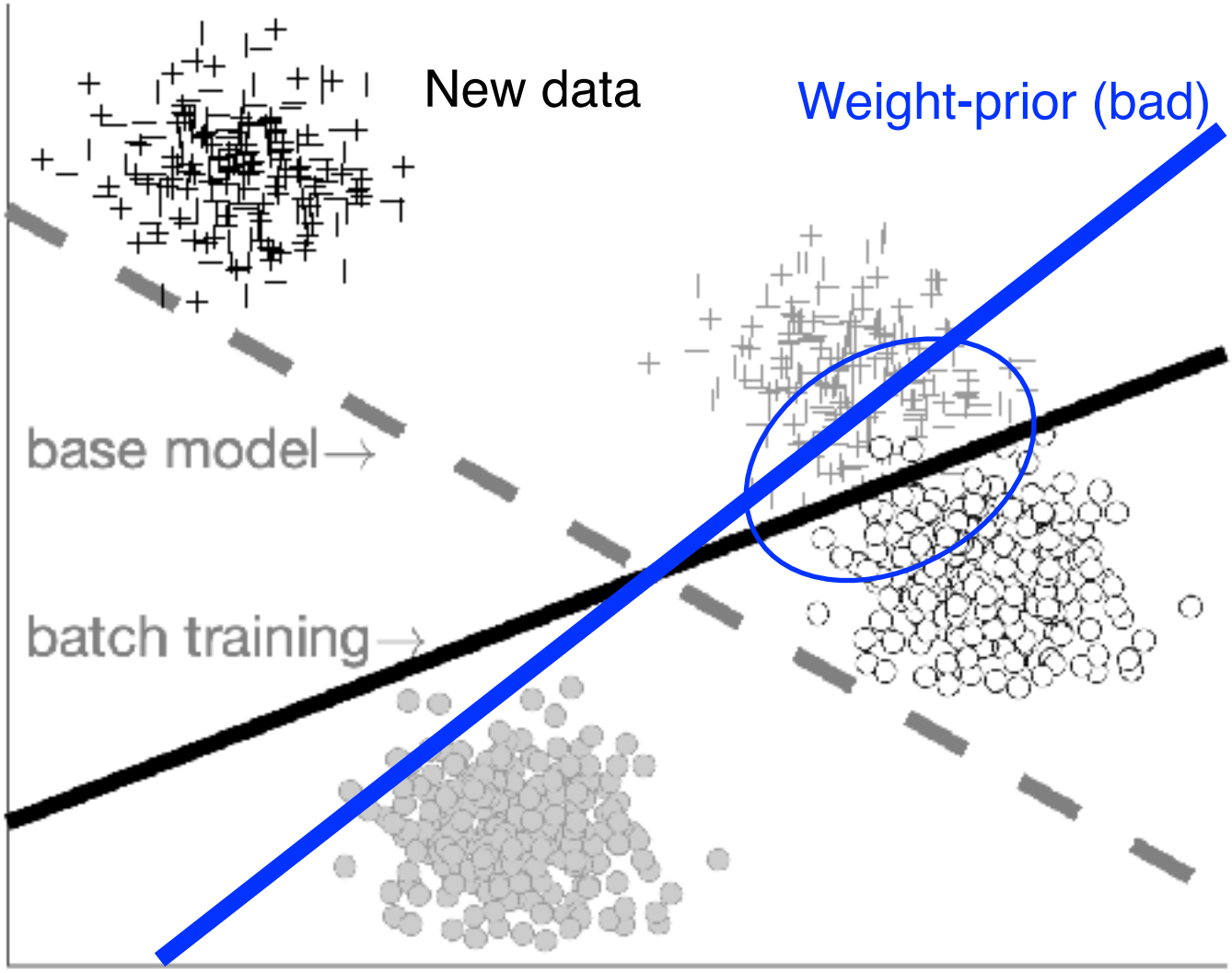
'Add Data' task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.

Inaccuracy of Weight-Priors



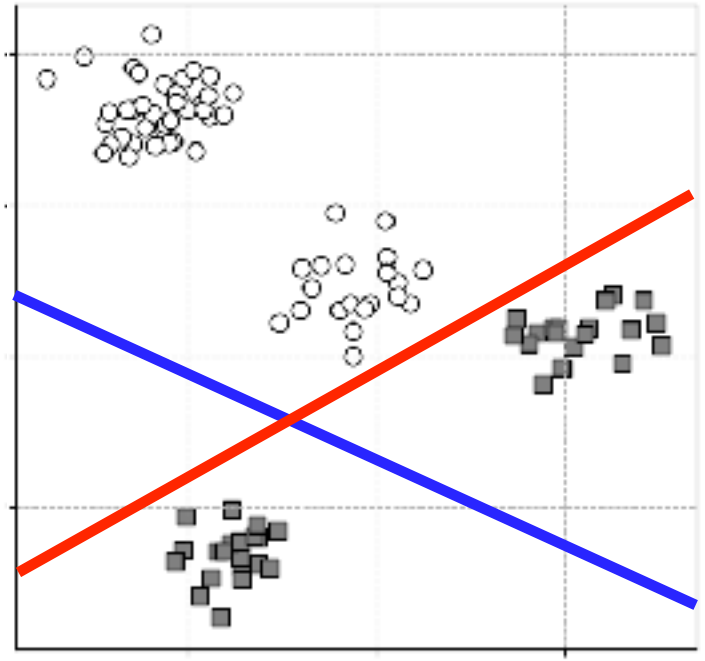
'Add Data' task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.

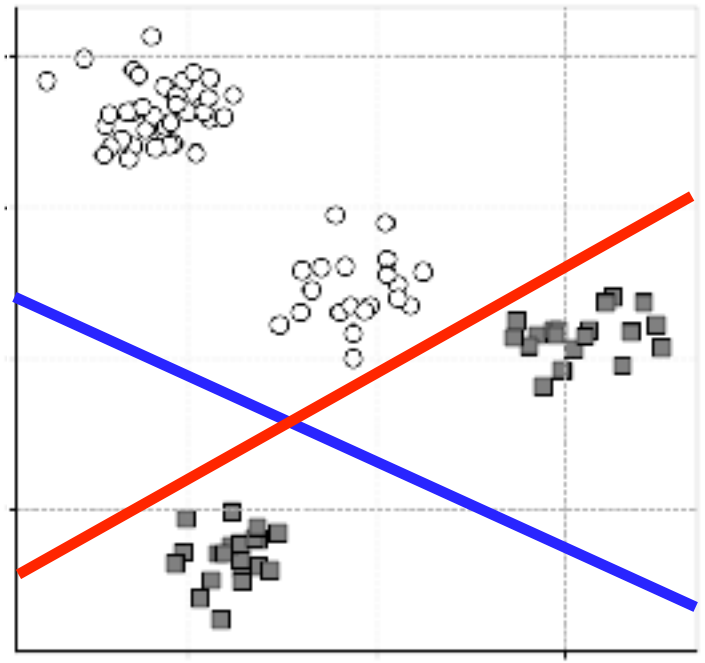
1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.

Continual Learning in function space



1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." PNAS 2017
2. Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020

Continual Learning in function space

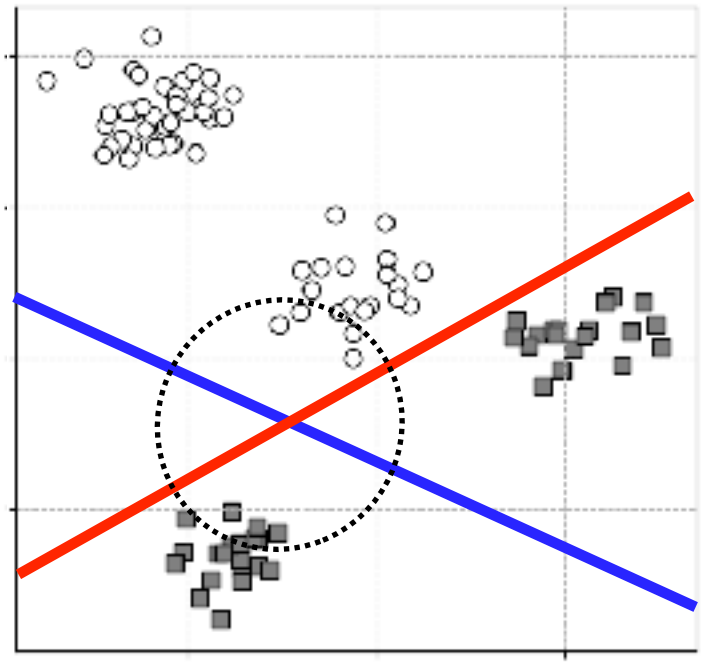


Weights Regularization [1]

$$(\theta - \theta_{old})^{\top}\Sigma_{old}^{-1}(\theta - \theta_{old})$$

1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." PNAS 2017
2. Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020

Continual Learning in function space



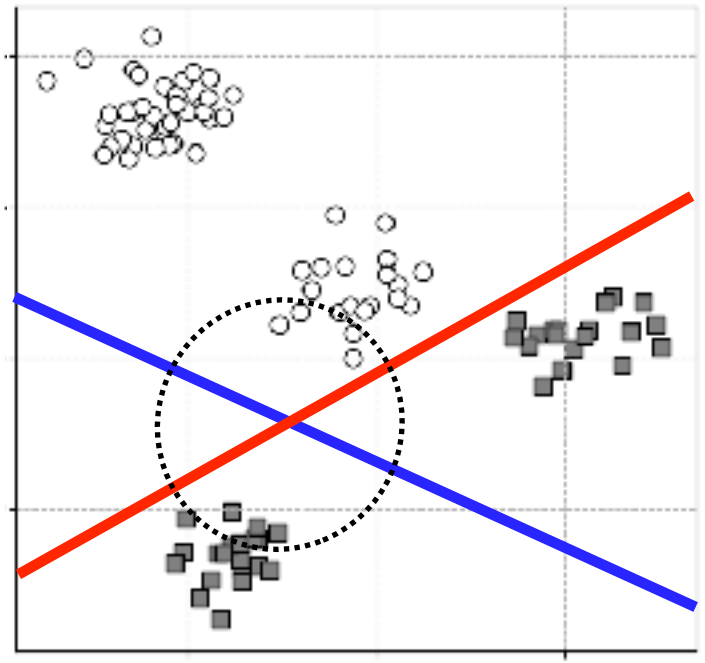
Weights Regularization [1]

$$(\theta - \theta_{old})^{\top}\Sigma_{old}^{-1}(\theta - \theta_{old})$$

Functional Regularization of memorable past (FROMP) [2]

1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." PNAS 2017
2. Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020

Continual Learning in function space



Weights Regularization [1]

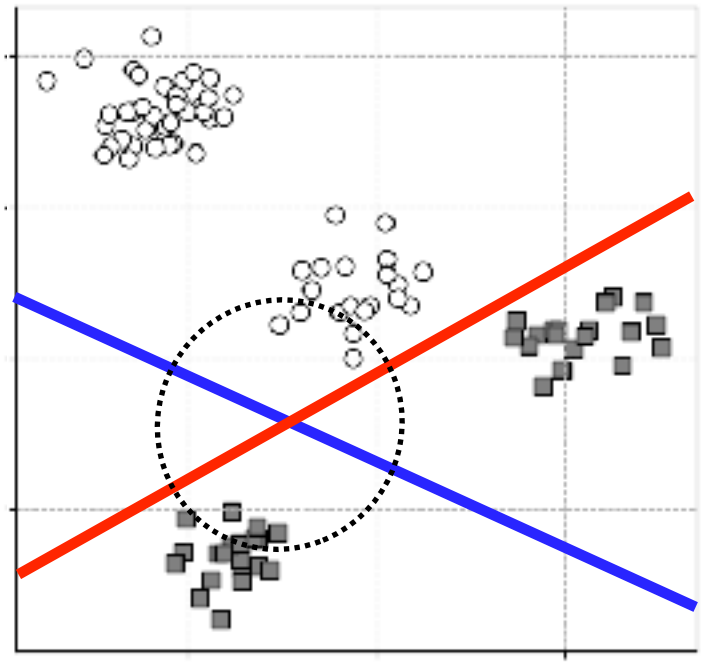
$$(\theta - \theta_{old})^{\top}\Sigma_{old}^{-1}(\theta - \theta_{old})$$

Functional Regularization of memorable past (FROMP) [2]

$$KL(p(\theta)||q(\theta)) \approx KL(p(f)||q(f))$$

1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." PNAS 2017
2. Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020

Continual Learning in function space



Weights Regularization [1]

$$(\theta - \theta_{old})^{\top}\Sigma_{old}^{-1}(\theta - \theta_{old})$$

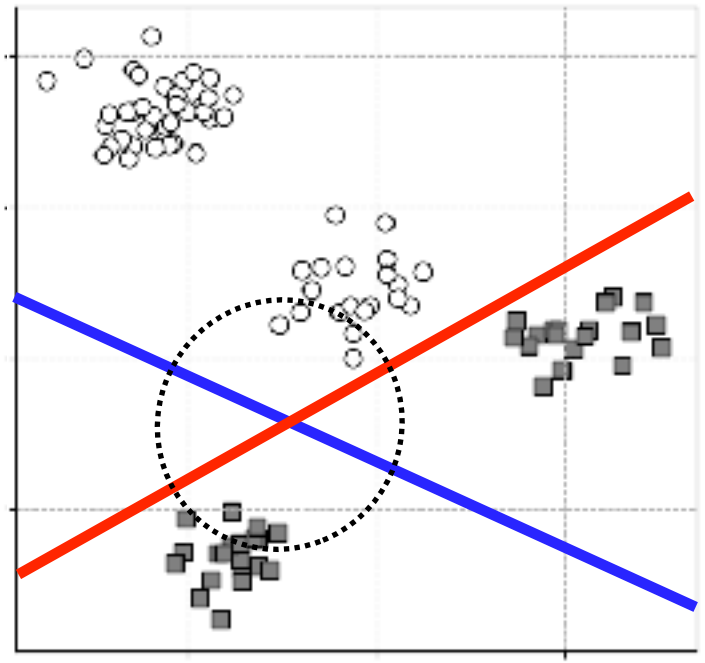
Functional Regularization of memorable past (FROMP) [2]

$$KL(p(\theta)||q(\theta)) \approx KL(p(f)||q(f))$$

$$\left[f(X_m) - f_{old}(X_m)\right]^{\top} K_{old}(X_m, X_m)^{-1}\left[f(X_m) - f_{old}(X_m)\right]$$

1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." PNAS 2017
2. Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020

Continual Learning in function space



Weights Regularization [1]

$$(\theta - \theta_{old})^{\top}\Sigma_{old}^{-1}(\theta - \theta_{old})$$

Functional Regularization of memorable past (FROMP) [2]

$$KL(p(\theta)||q(\theta)) \approx KL(p(f)||q(f))$$

$$\left[f(X_m) - f_{old}(X_m)\right]^{\top} K_{old}(X_m, X_m)^{-1}\left[f(X_m) - f_{old}(X_m)\right]$$

FROMP has the “right form” to reconstruct the “gradient of the past” — it is a type of K-prior!

1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." PNAS 2017
2. Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020

Knowledge-Adaptation Priors

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

Siddharth Swaroop*
University of Cambridge
Cambridge, UK
ss2163@cam.ac.uk

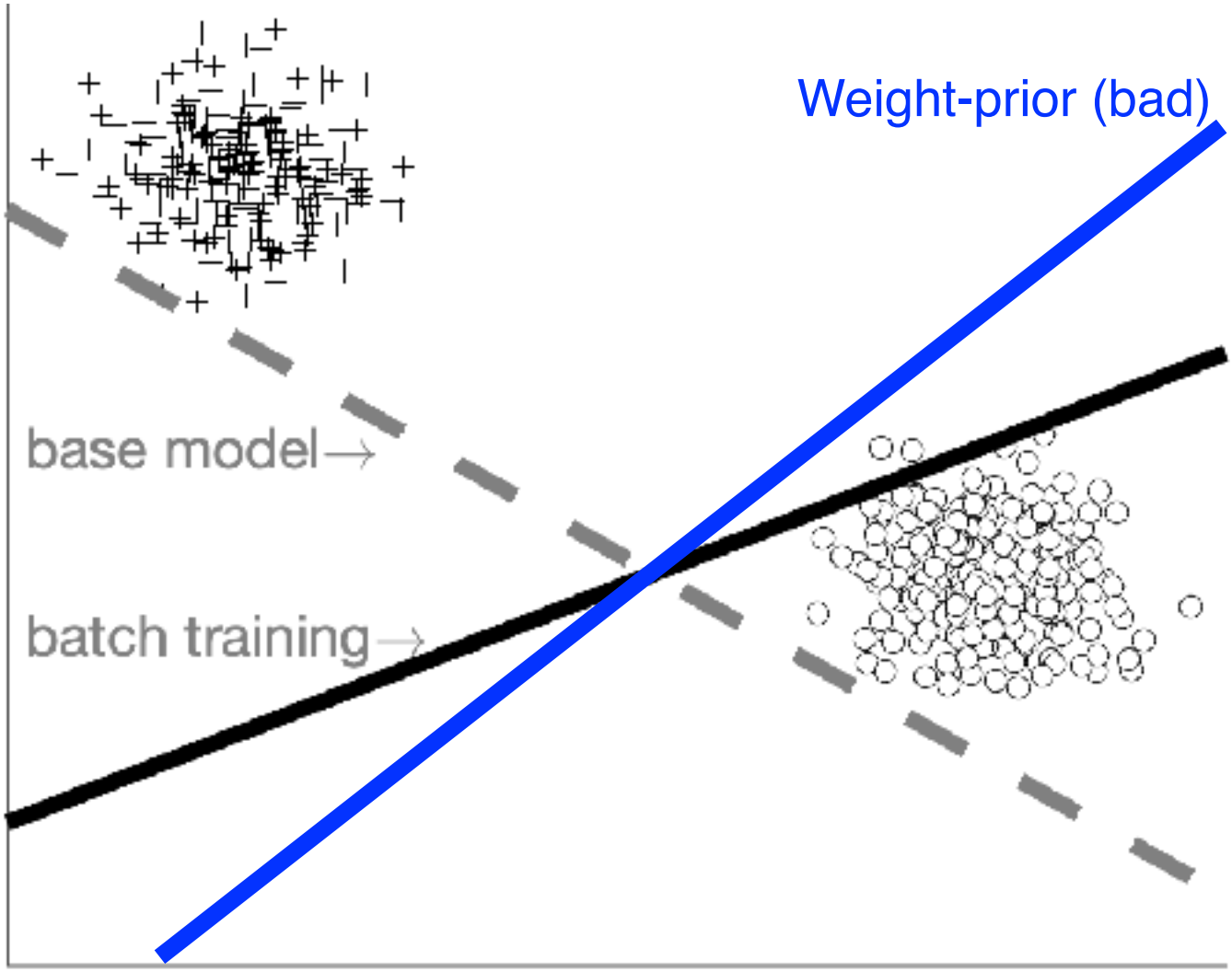
Abstract

Humans and animals have a natural ability to quickly adapt to their surroundings, but machine-learning models, when subjected to changes, often require a complete retraining from scratch. We present Knowledge-adaptation priors (K-priors) to reduce the cost of retraining by enabling quick and accurate adaptation for a wide-variety of tasks and models. This is made possible by a combination of weight and function-space priors to reconstruct the gradients of the past, which recovers and generalizes many existing, but seemingly-unrelated, adaptation strategies. Training with simple first-order gradient methods can often recover the exact retrained model to an arbitrary accuracy by choosing a sufficiently large memory of the past data. Empirical results confirm that the adaptation can be cheap and accurate, and a promising alternative to retraining.

Joint work with Siddharth Swaroop
University of Cambridge, UK

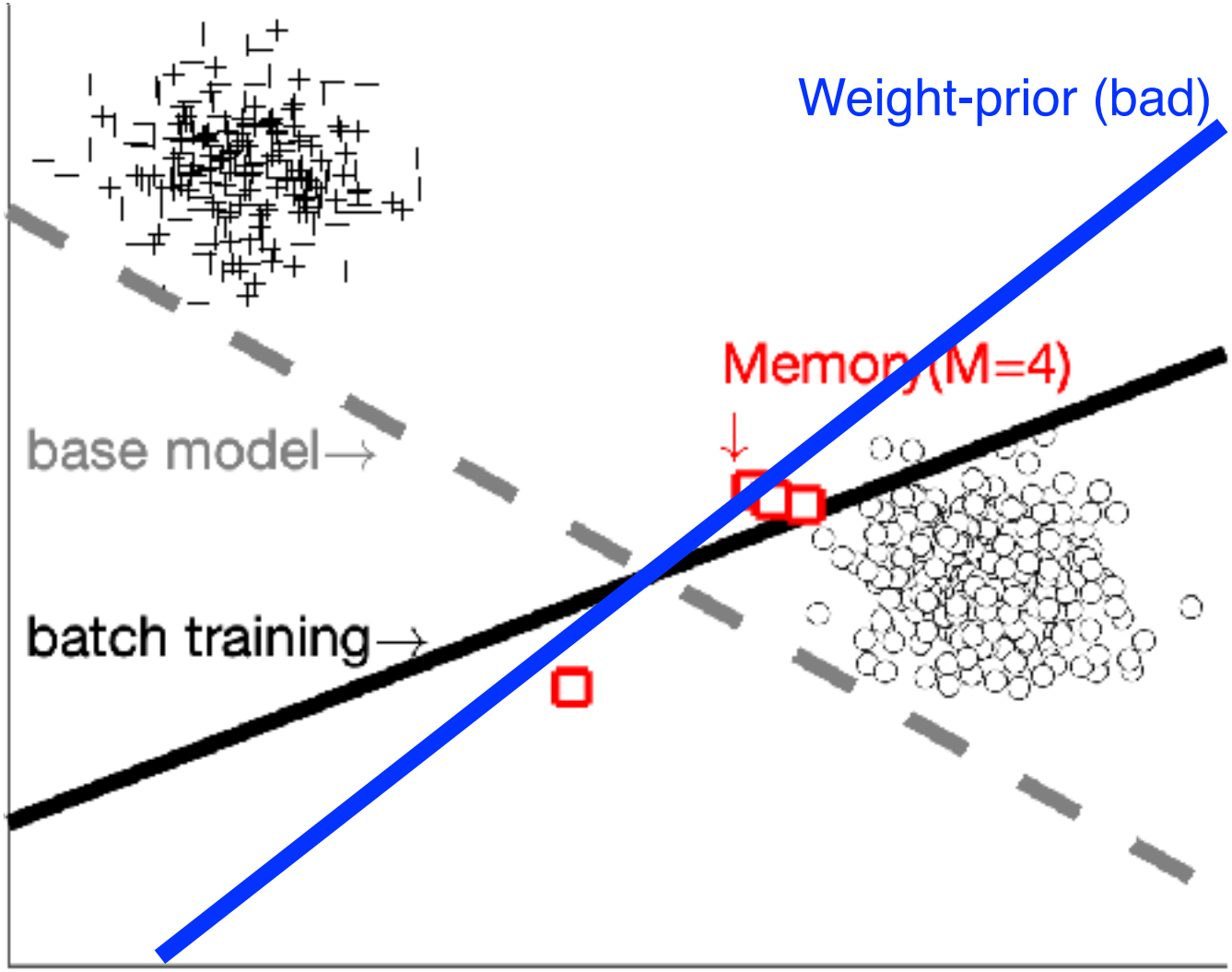
1. Khan and Swaroop. Knowledge-Adaptation Priors, arXiv, 2021 (https://arxiv.org/abs/2106.08769)

Knowledge-Adaptation Priors



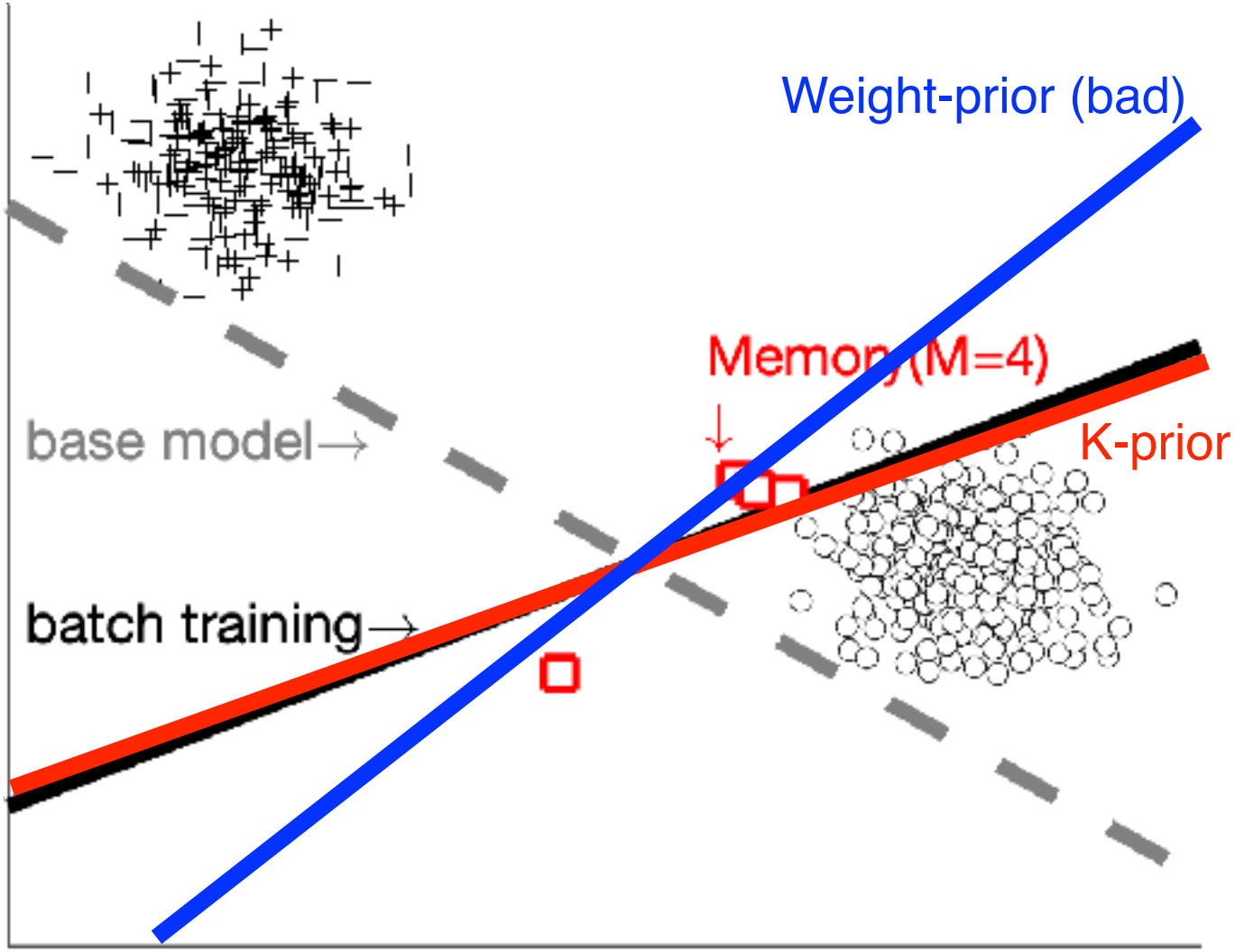
K-priors use past-memory $\mathscr{M}$ (size M) in addition to the base model.

Knowledge-Adaptation Priors



K-priors use past-memory $\mathscr{M}$ (size M) in addition to the base model.

Knowledge-Adaptation Priors



K-priors use past-memory $\mathscr{M}$ (size M) in addition to the base model.

K-prior Construction

Combine weight and function-space divergences

Weight-space Function-space

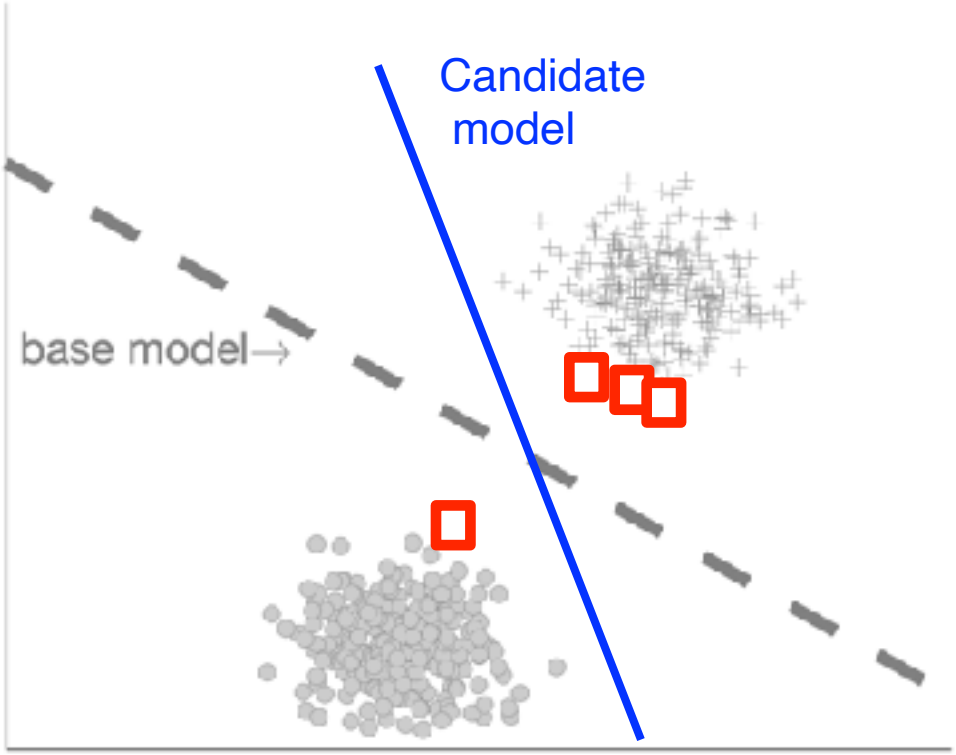
$$\mathcal{K}(w) = \tau \mathbb{D}_w(w \| w_*) + \mathbb{D}_f(\mathbf{f}(w) \| \mathbf{f}(w_*))$$

K-prior Construction

Combine weight and function-space divergences

Weight-space Function-space

$$\mathcal{K}(w) = \tau \mathbb{D}_w(w \| w_*) + \mathbb{D}_f(\mathbf{f}(w) \| \mathbf{f}(w_*))$$



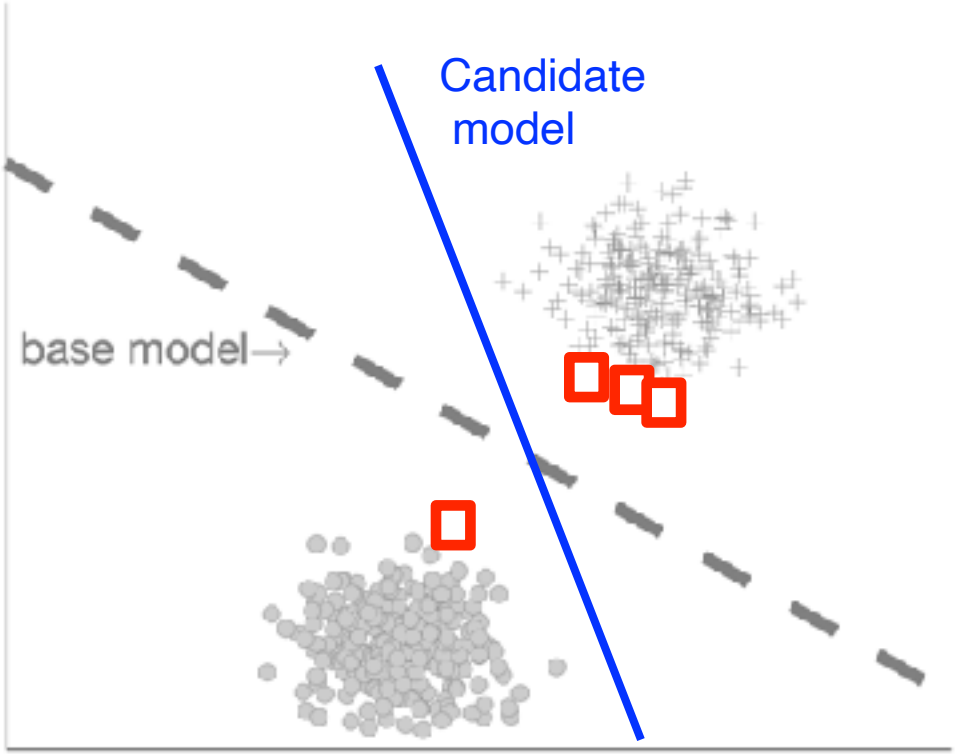
$$\begin{bmatrix} f_w^1 \\ f_w^2 \\ f_w^3 \\ f_w^4 \end{bmatrix} \quad \begin{bmatrix} f_{w_*}^1 \\ f_{w_*}^2 \\ f_{w_*}^3 \\ f_{w_*}^4 \end{bmatrix}$$

K-prior Construction

Combine weight and function-space divergences

Weight-space Function-space

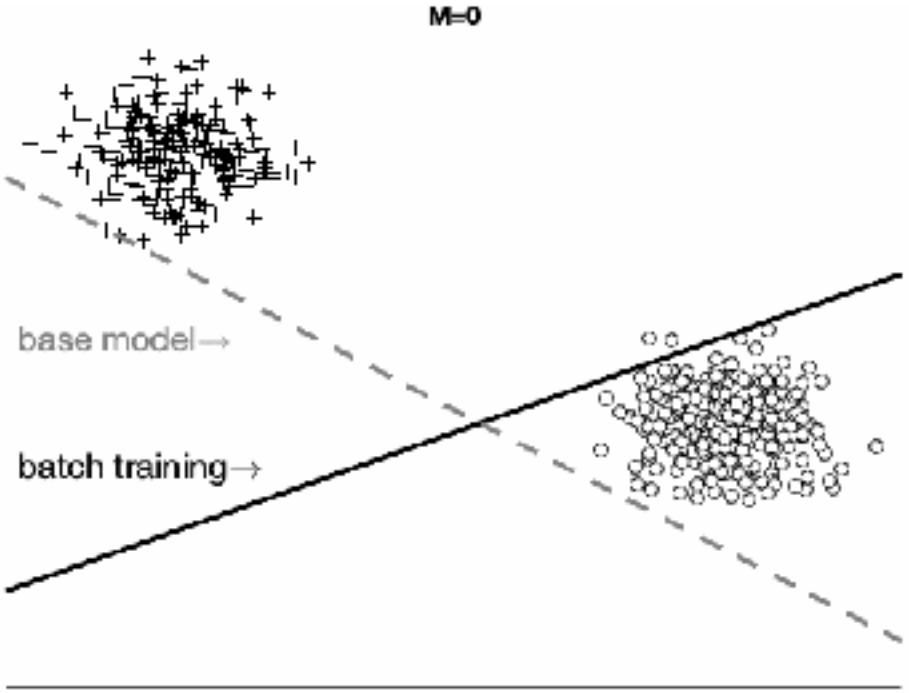
$$\mathcal{K}(w) = \tau \mathbb{D}_w(w \| w_*) + \mathbb{D}_f(\mathbf{f}(w) \| \mathbf{f}(w_*))$$

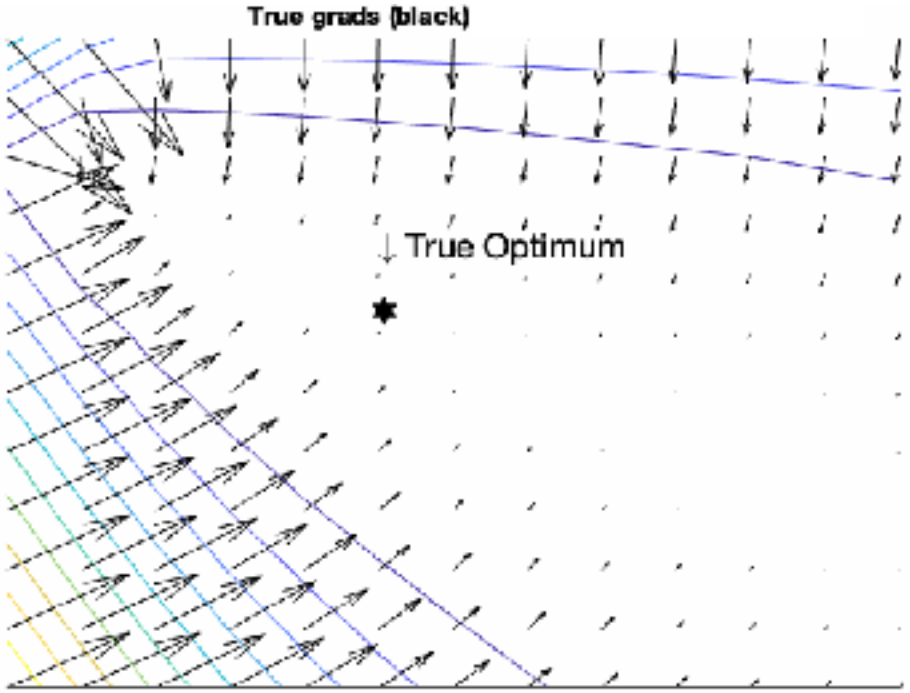


$$\begin{bmatrix} f_w^1 \\ f_w^2 \\ f_w^3 \\ f_w^4 \end{bmatrix} \quad \begin{bmatrix} f_{w_*}^1 \\ f_{w_*}^2 \\ f_{w_*}^3 \\ f_{w_*}^4 \end{bmatrix}$$

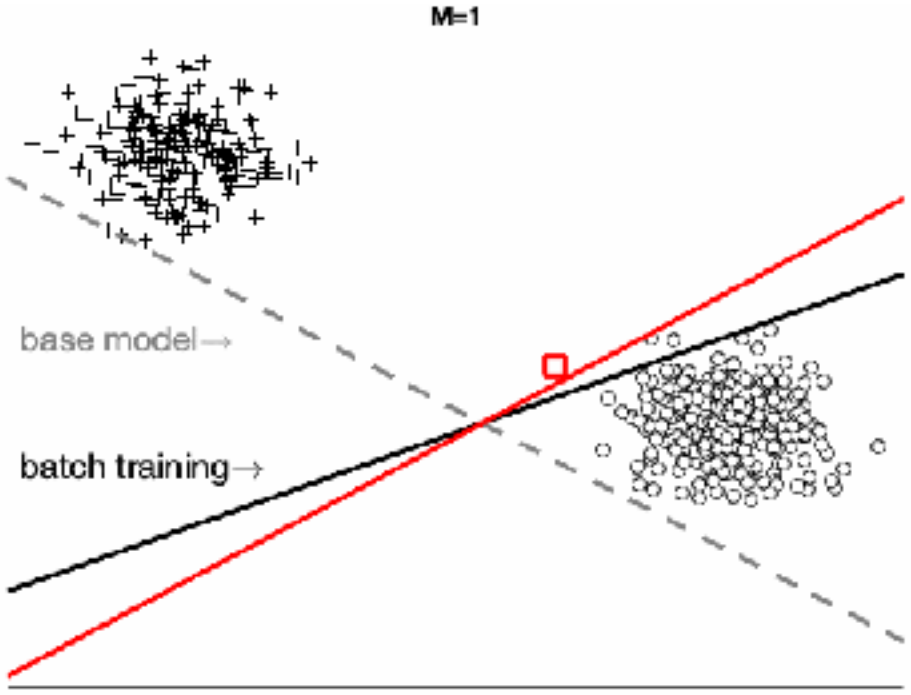
No labels required, so $\mathscr{M}$ can include any inputs!

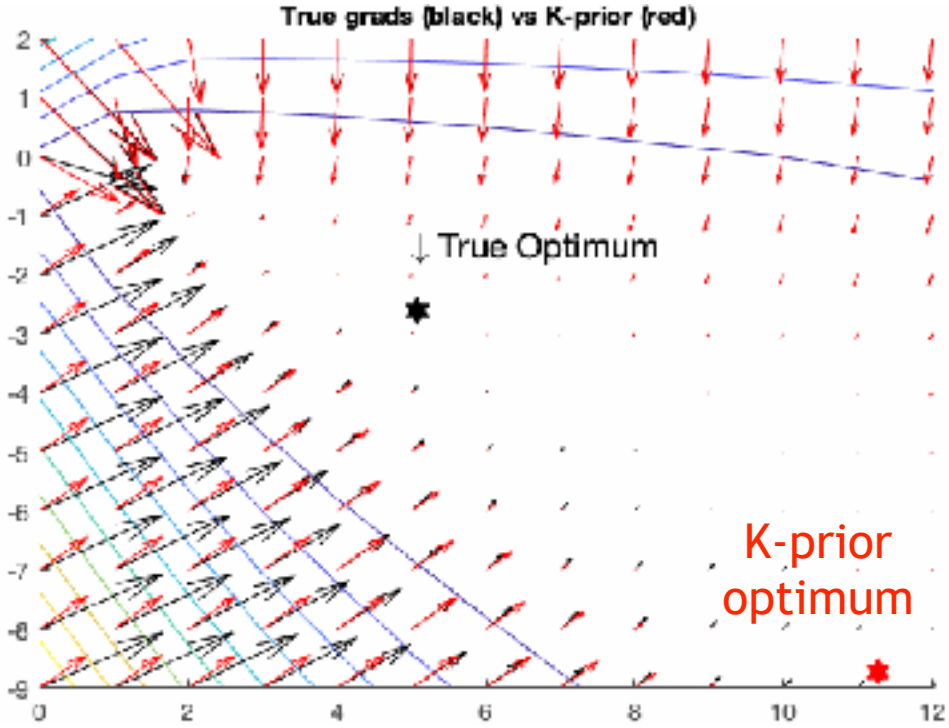
Faithful Gradient Reconstruction



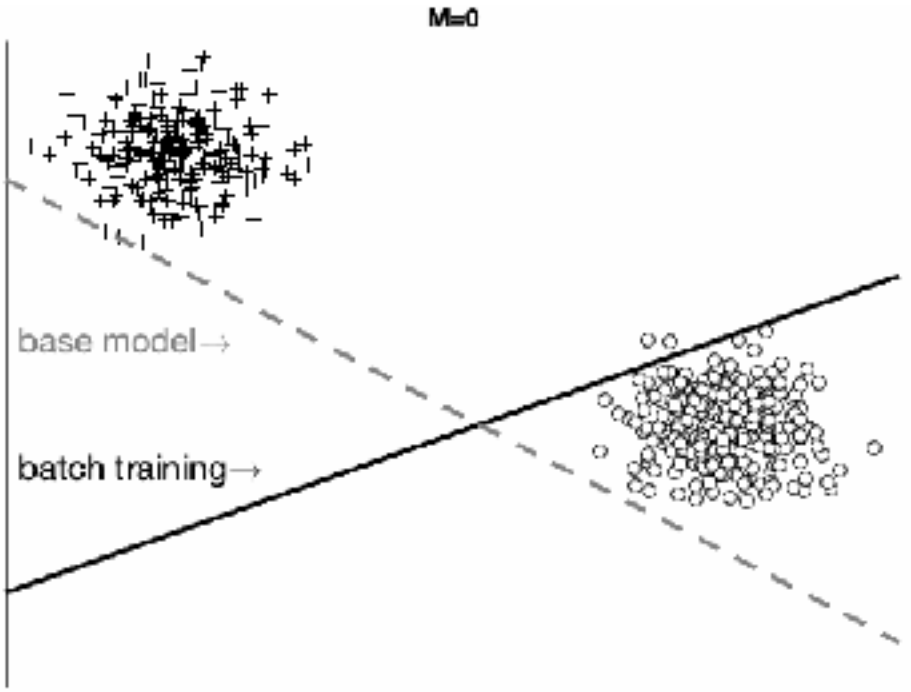


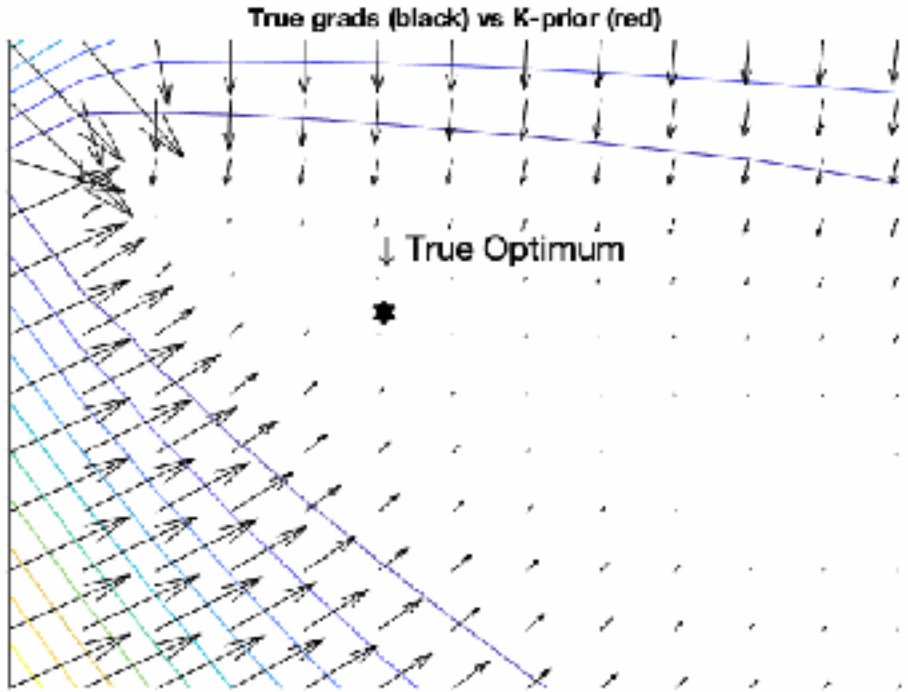
Faithful Gradient Reconstruction





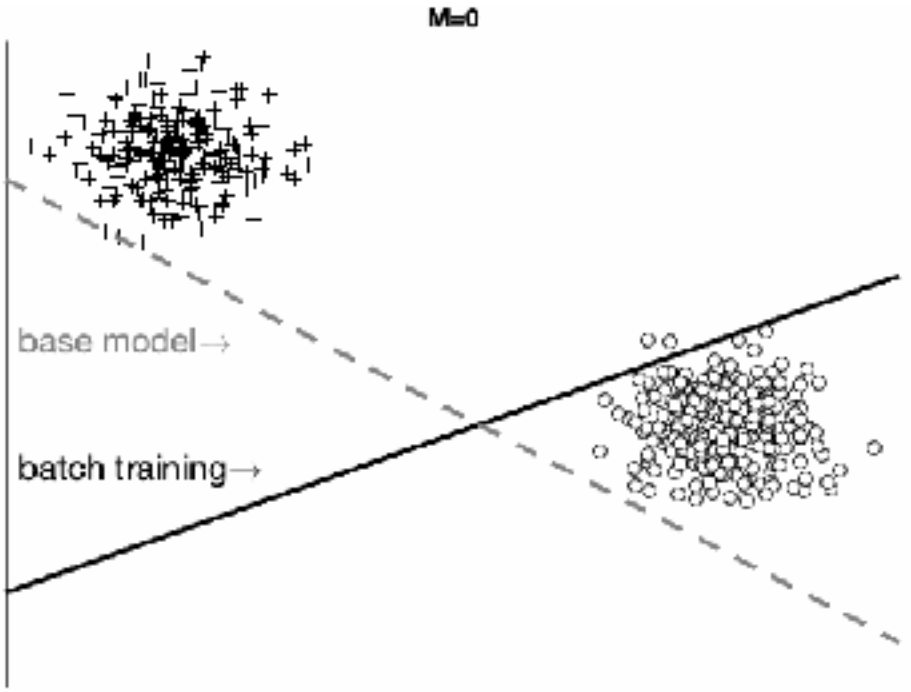
Faithful Gradient Reconstruction

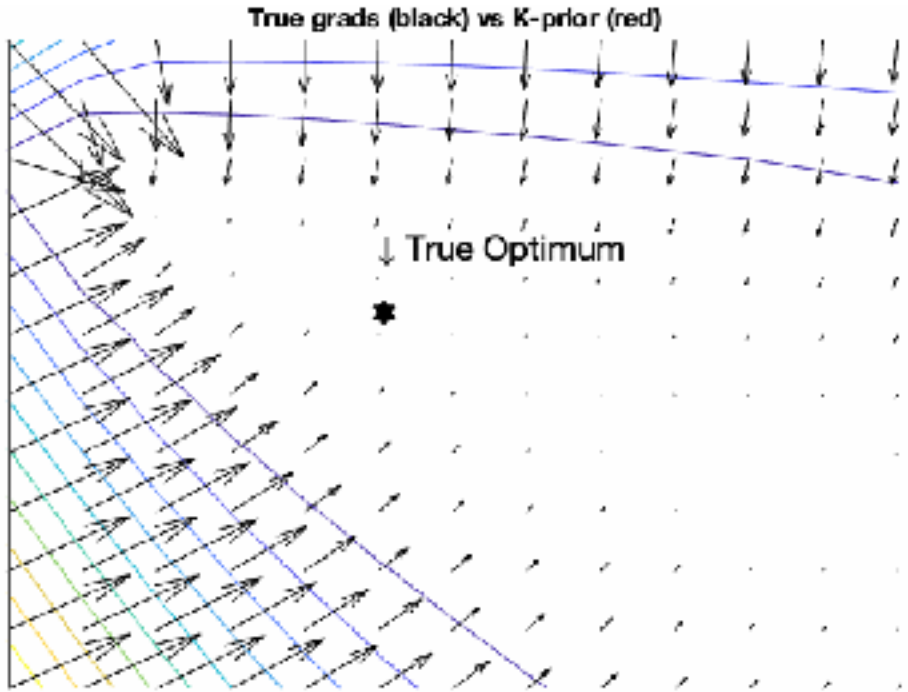




No labels required, so $\mathscr{M}$ can include any inputs!

Faithful Gradient Reconstruction

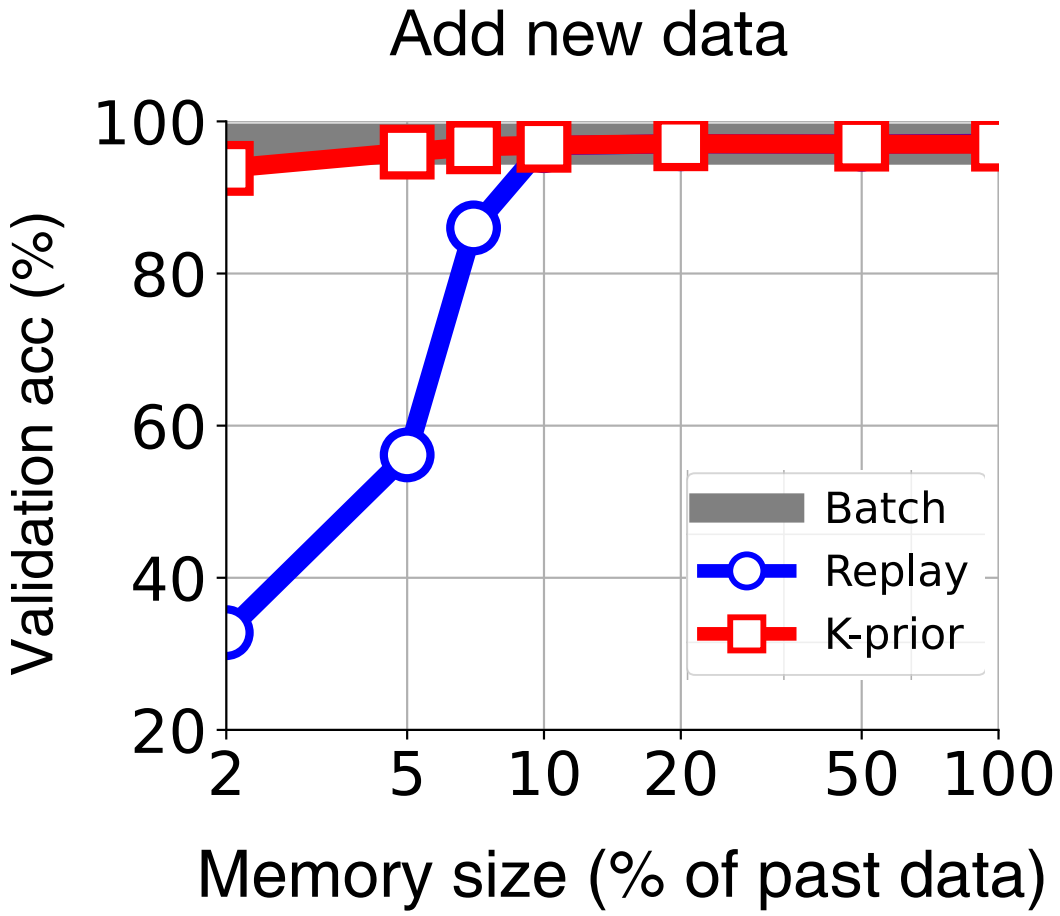




No labels required, so $\mathscr{M}$ can include any inputs!

Results

K-priors need < 2% of past data to match “batch”.

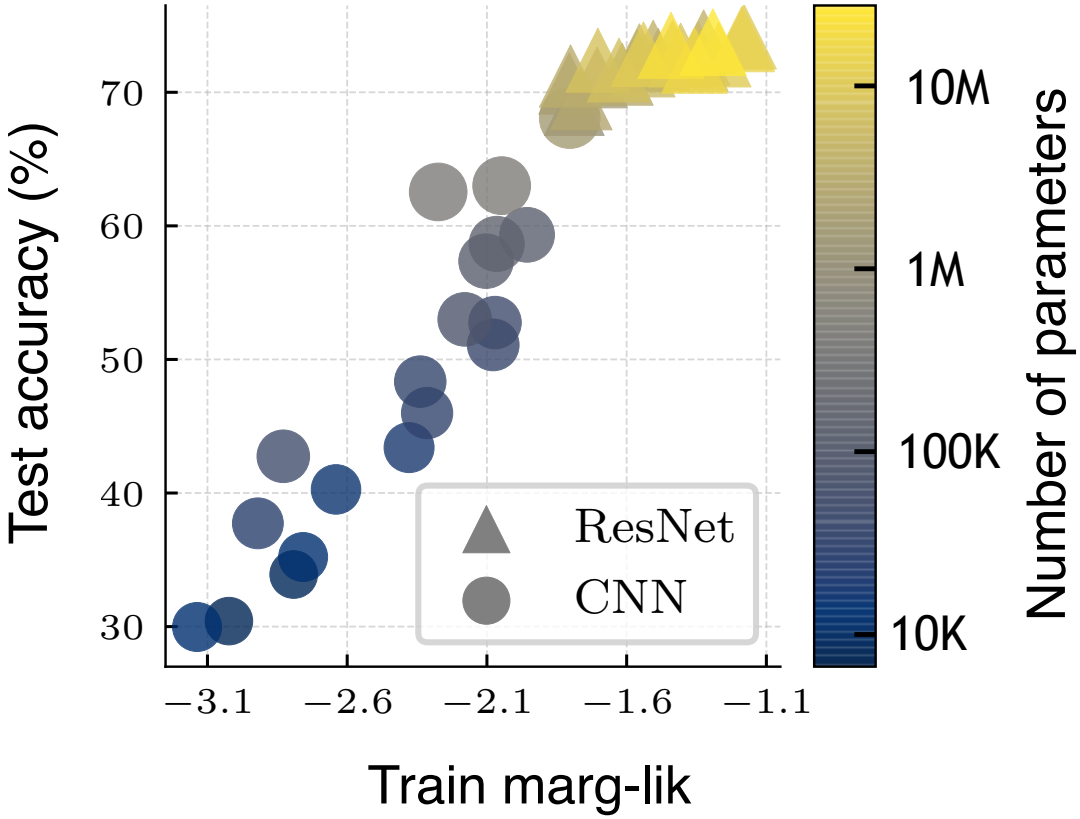


The results are on USPS binary classification with Neural nets.

“Replay” uses the same memory but with true outputs.

Model selection without test set

The “training marginal-likelihood” can be used to select deep-nets, *without* requiring the test set.



Test-accuracy correlates with train marg-lik.

Both increase as the model size is increased.

On CIFAR-100, around 50 models are shown.

Immer et al., Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning, *ICML*, 2021.

How to design AI that learn like us?

How to design AI that learn like us?

- Three questions

How to design AI that learn like us?

- Three questions
  - Q1: What do we know? (model)

How to design AI that learn like us?

- Three questions
  - Q1: What do we know? (model)
  - Q2: What do we not know? (uncertainty)

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)

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

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

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

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

Emtiyaz Khan
Team Leader

Pierre Alquier
Research Scientist

Gian Maria Marconi
Postdoc

Thomas Möllenhoff
Postdoc

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

Wu Lin
PhD Student
University of British Columbia

Dharmesh Tailor
Research Assistant

Fariz Ikhwantri
Part-time Student
Tokyo Institute of Technology

Happy Buzaaba
Part-time Student
University of Tsukuba

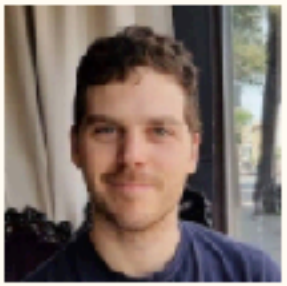
Evgenii Egorov
Remote Collaborator
Skoltech

Siddharth Swaroop
Remote Collaborator
University of Cambridge

Dimitri Meunier
Remote Collaborator
ENSAE Paris

Peter Nickl
Remote Collaborator
TU Darmstadt

Erik Daxberger
Remote Collaborator
University of Cambridge



Alexandre Piché
Remote Collaborator
MILA