Presentation at the UNIST AI Technology Open Workshop
Sep 21, 2023

Bayesian Learning Rule for Adaptive AI

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

Summary of recent research at https://emtiyaz.github.io/papers/symposium_2022.pdf

How to make AI that can adapt quickly?

Reasoning is crucial for this!

Human Learning at the age of 6 months.

Converged at the age of 12 months

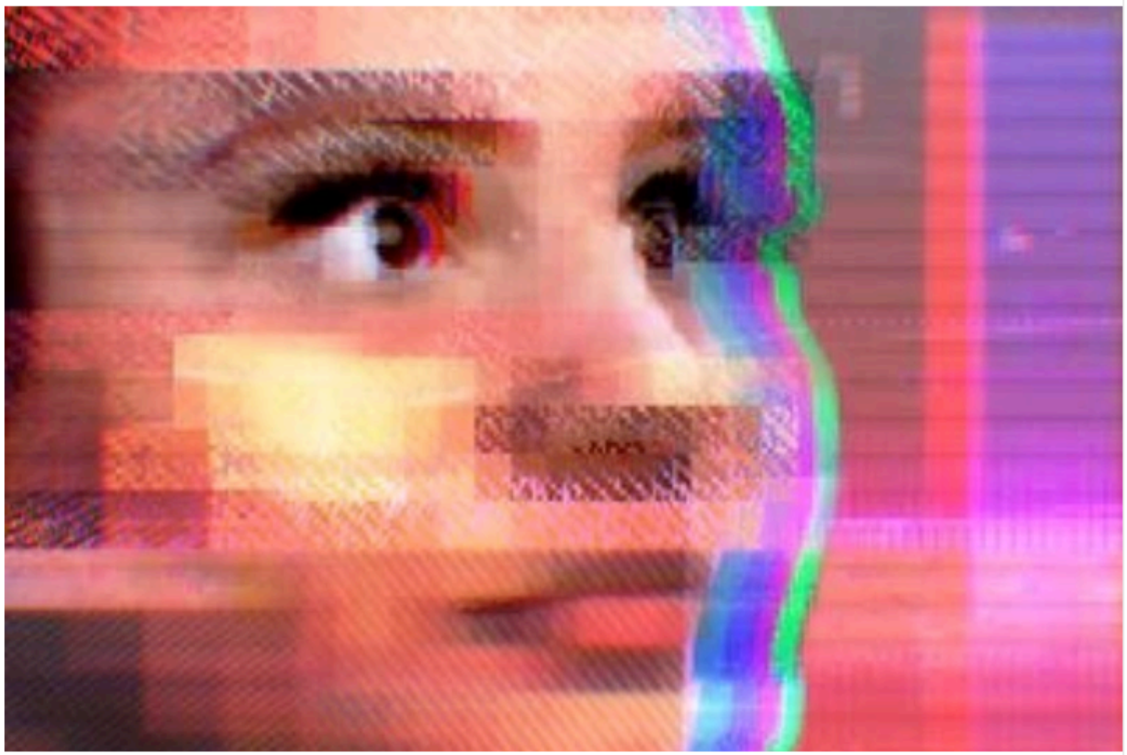
Transfer skills

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

Fail because too slow to adapt

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

Adaptation in Machine Learning

- Even a small change may need retraining
- Huge amount of resources are required only few can afford (costly & unsustainable) [1,2, 3]
- Difficult to apply in "dynamic" settings (robotics, medicine, epidemiology, climate science, etc.)
- Our goal is to solve such challenges
- Also to reduce "magic" in deep learning

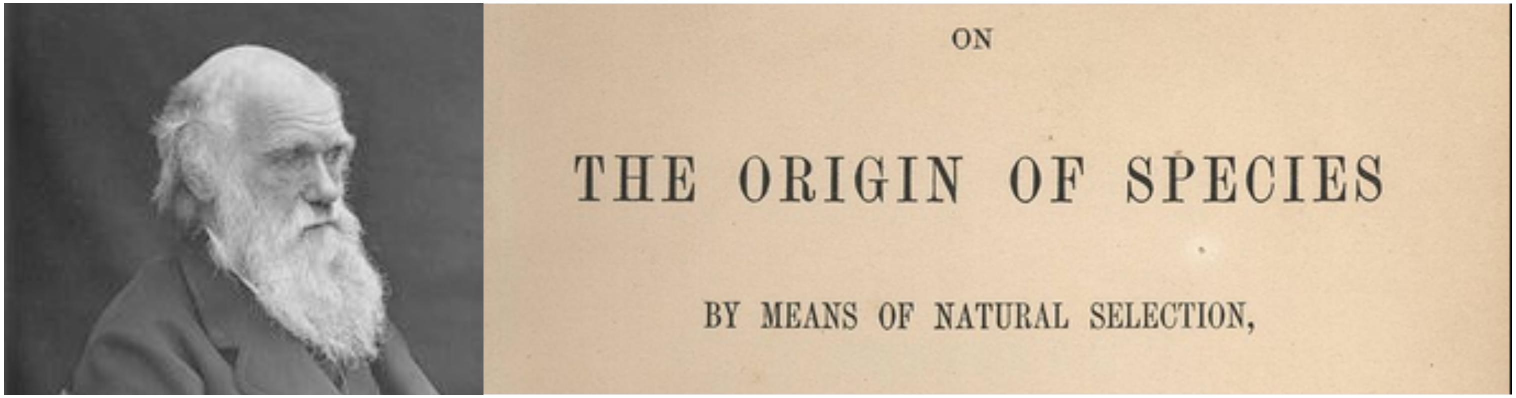
1. Diethe et al. Continual learning in practice, arXiv, 2019.
2. Paleyes et al. Challenges in deploying machine learning: a survey of case studies, arXiv, 2021.
3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s

Towards Quick Adaptation

- Unify, generalize and improve algorithms
  - Bayesian Learning rule (BLR)
- Memory (or representation)
  - Sensitivity and dual view of the BLR
- Adaptation (or transfer)
  - Continual learning and K-priors
  - Use sensitivity to adapt quickly

Bayesian Learning Rule

Unify, generalize, and improve
learning algorithms



The Origin of Algorithms

What are the common principles behind popular algorithms?

1. 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

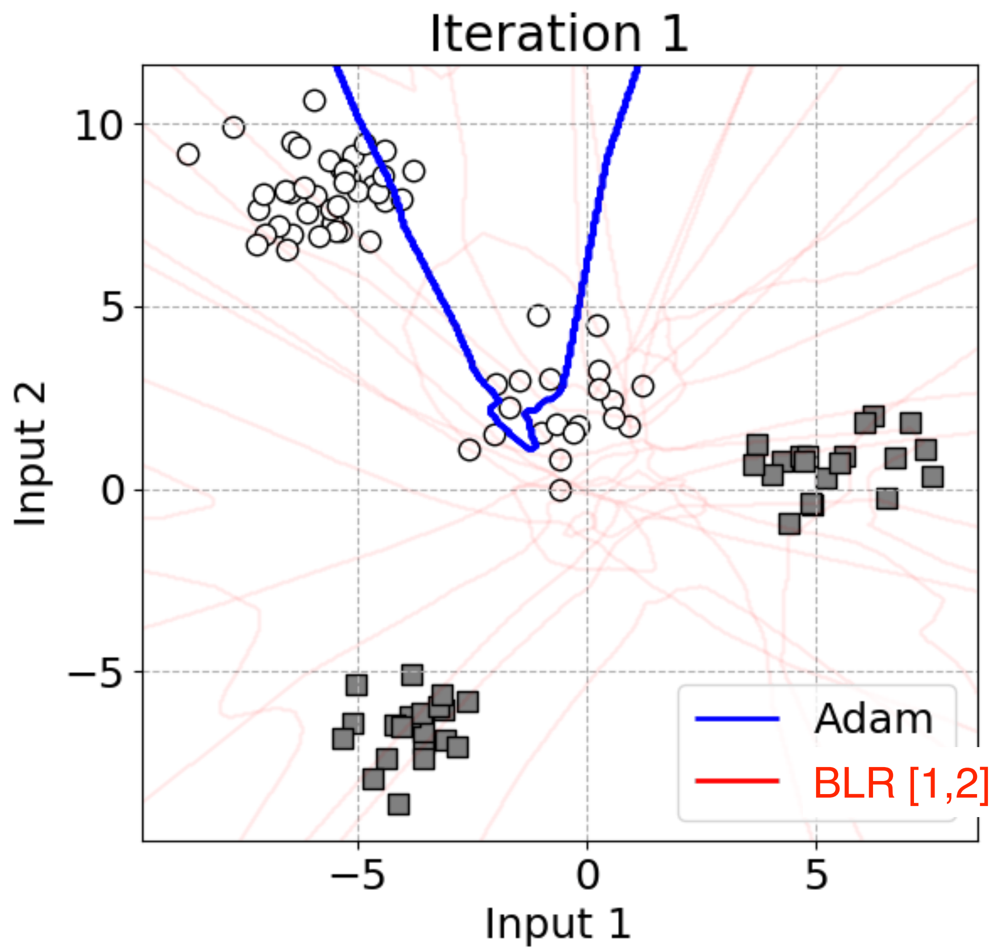
Learning Algorithm	Posterior Approx.	Natural-Gradient Approx.	Sec.
Optimization Algorithms			
Gradient Descent	Gaussian (fixed cov.)	Delta method	1.3
Newton's method	Gaussian	—"—	1.3
Multimodal optimization (New)	Mixture of Gaussians	—"—	3.2
Deep-Learning Algorithms			
Stochastic Gradient Descent	Gaussian (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

All sorts of algorithms can be derived by using two sets of approximations.

By relaxing the approximations, we get an improvement, for example, uncertainty aware deep learning optimizers

1. Khan and Rue, The Bayesian Learning Rule, arXiv, https://arxiv.org/abs/2107.04562, 2021
2. Khan and Lin. "Conjugate-computation variational inference…." AIstats (2017).

Uncertainty in Deep Learning



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).

Practical Deep Learning with Bayes

How to estimate uncertainty with DL optimizers?

RMSprop

$$
\begin{aligned}
g &\leftarrow \hat{\nabla}\ell(\theta) \\
h &\leftarrow g \cdot g \\
s &\leftarrow (1-\rho)s + \rho h \\
\theta &\leftarrow \theta - \alpha\, g/\sqrt{s} \\
\sigma^2 &\leftarrow 1/\sqrt{s}???
\end{aligned}
$$

Costs are exactly the same, but uncertainty quality is much better!!

(Improved) Bayesian Learning Rule [3]

$$
\begin{aligned}
g &\leftarrow \hat{\nabla}\ell(\theta) \\
h &\leftarrow g \cdot \sqrt{s} \cdot \epsilon \\
s &\leftarrow (1-\rho)s + \rho h + \rho^2 h^2/(2s) \\
m &\leftarrow m - \alpha\; g/s \\
\sigma^2 &\leftarrow 1/s, \;\; \theta \leftarrow m + \epsilon \sim \mathcal{N}(0, 1/s)
\end{aligned}
$$

Perturb the gradients to get Hessian
Perturb according to the posterior
Ensure s is always +ve

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).

The Bayesian Learning Rule

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

Entropy

Posterior approximation (eg Gaussian)

Natural gradient descent (or equivalently mirror descent)

Natural and Expectation parameters of q

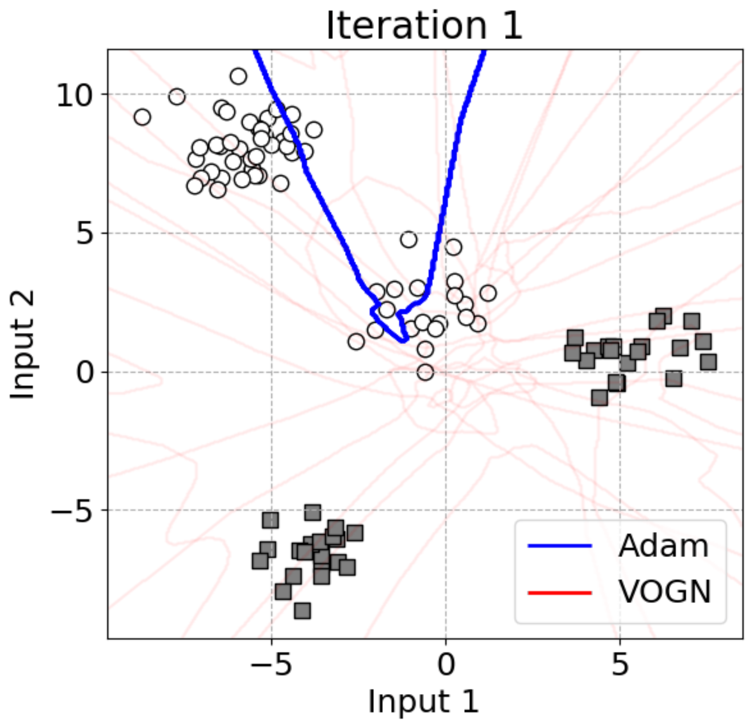
$$\lambda \leftarrow \lambda - \rho \nabla_{\mu} \Big\{ \mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q) \Big\}$$

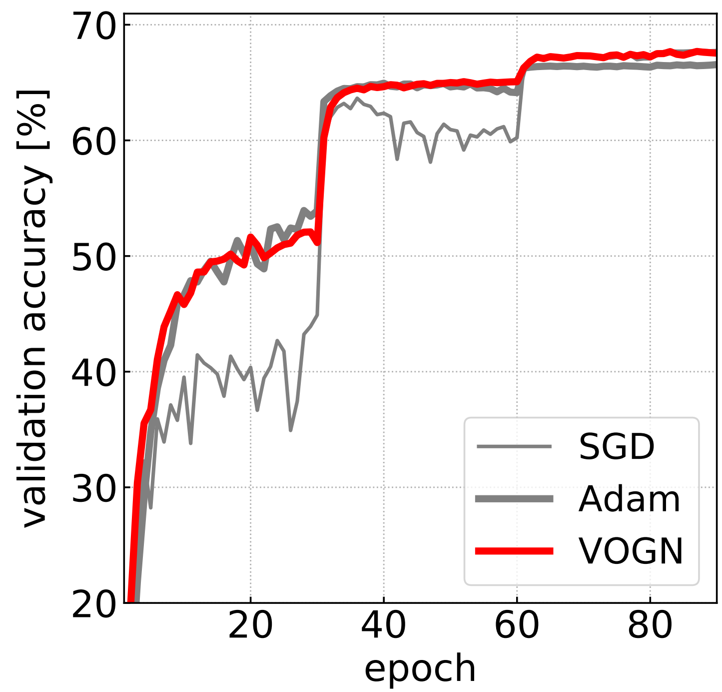
Exploiting posterior's information geometry to derive existing algorithms as special instances

1. Khan and Rue, The Bayesian Learning Rule, arXiv, https://arxiv.org/abs/2107.04562, 2021
2. Khan and Lin. "Conjugate-computation variational inference…." AIstats (2017).

Uncertainty of Deep Nets

VOGN: A modification of Adam with similar performance on ImageNet, but better uncertainty





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 variant [3] got 1st prize in NeurIPS 2021 Approximate Inference Challenge

Watch Thomas Moellenhoff's talk at https://www.youtube.com/watch?v=LQInlN5EU7E.

Mixture-of-Gaussian Posteriors with an Improved Bayesian Learning Rule

Thomas Möllenhoff[1], Yuesong Shen[2], Gian Maria Marconi[1]
Peter Nickl[1], Mohammad Emtiyaz Khan[1]

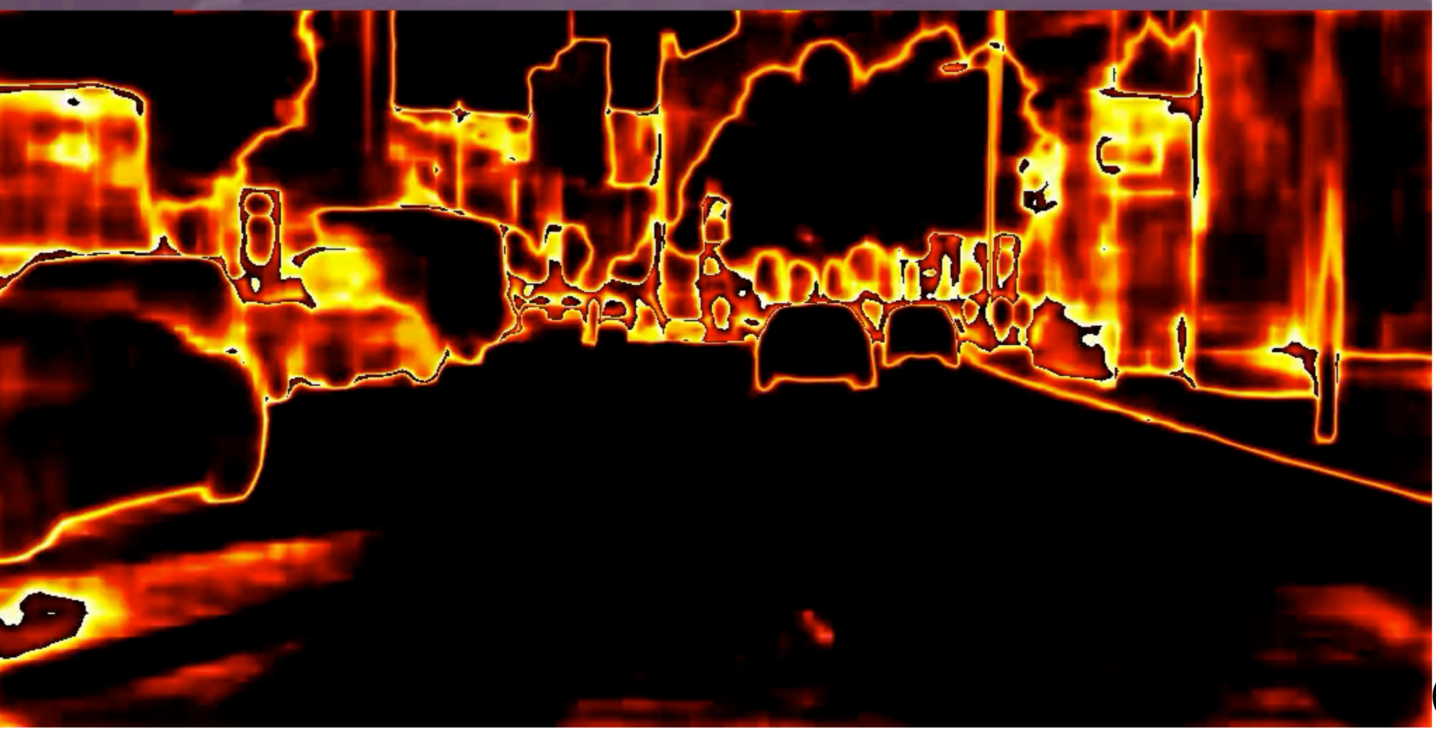
1 Approximate Bayesian Inference Team
RIKEN Center for AI Project, Tokyo, Japan

2 Computer Vision Group
Technical University of Munich, Germany

Dec 14th, 2021 — NeurIPS Workshop on Bayesian Deep Learning

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).

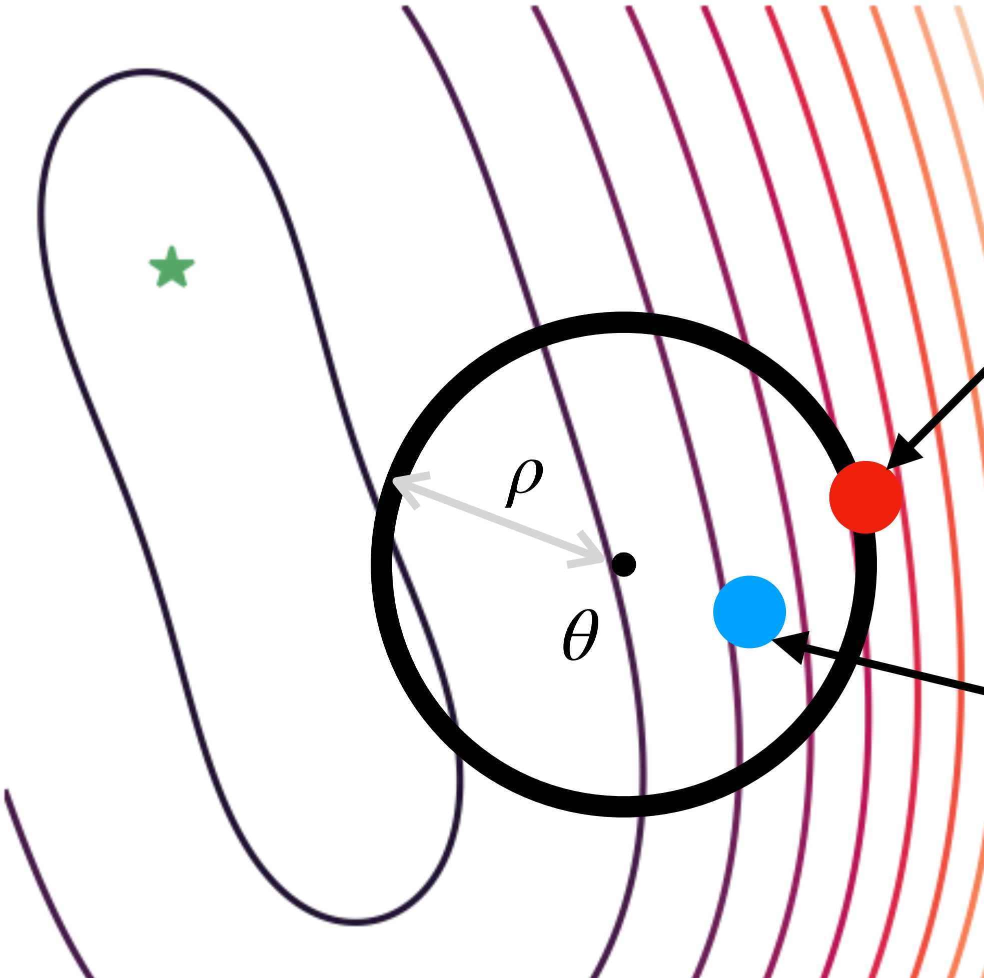
Image Segmentation



Uncertainty (entropy of class probs)

(By Roman Bachmann)

Sharpness-Aware Minimization (SAM) as an Optimal relaxation of Bayes



SAM: $\sup_{|\epsilon|<\rho} \ell(\theta + \epsilon)$

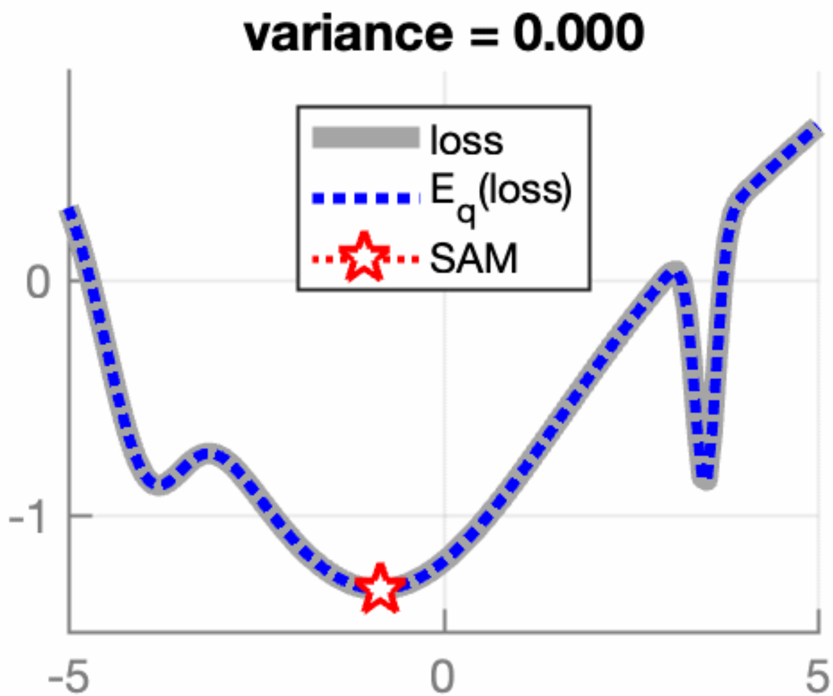
Our work: Fenchel Biconjugate

Bayes: $\mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2)}[\ell(\theta + \epsilon)]$

1. Moellenhoff and Khan, SAM as optimal relaxation of Bayes, ICLR 2023 (top 5%)

SAM as a relaxation of Bayes

SAM (red star) upper bounds the Bayesian $\mathbb{E}_q[\ell]$



Bayesian-SAM

An Adam-style algorithm, derived using the BLR, where variances are automatically learned.

SAM with RMSprop

$$
\begin{aligned}
g_1 &\leftarrow \hat{\nabla}\ell(\theta) \\
\epsilon &\leftarrow \rho \frac{g_1}{\|g_1\|} \\
g &\leftarrow \hat{\nabla}\ell(\theta+\epsilon) \\
s &\leftarrow (1-\rho)s + \rho g^2 \\
\theta &\leftarrow \theta - \alpha(\sqrt{s}+\delta)^{-1} g
\end{aligned}
$$

SAM with BLR

$$
\begin{aligned}
g_1 &\leftarrow \hat{\nabla}\ell(\theta) \\
\epsilon &\leftarrow \frac{\rho'}{s} g_1 \\
g &\leftarrow \hat{\nabla}\ell(\theta+\epsilon) \\
s &\leftarrow (1-\rho)s + \rho\sqrt{s}|g_1| \\
\theta &\leftarrow \theta - \alpha(s+\gamma)^{-1} g \\
\sigma^2 &\leftarrow (s+\gamma)^{-1}, \;\; \theta \leftarrow m + \epsilon'\sigma
\end{aligned}
$$

1. Foret et al. Sharpness-Aware Minimization for Efficiently Improving Generalization, ICLR, 2021
2. Moellenhoff and Khan, SAM as an optimal relaxation of Bayes, https://arxiv.org/abs/2210.01620, 2022

Uncertainty Improves Performance

CIFAR-100 with ResNet-20 (270K params).

	Accuracy		AUROC
SGD	$55.82_{(0.97)}$	+8%	$0.811_{(0.004)}$
SAM-SGD	$58.58_{(0.59)}$		$0.827_{(0.003)}$
SWAG	$56.53_{(0.40)}$		$0.814_{(0.004)}$
VOGN	$59.83_{(0.75)}$		$0.830_{(0.002)}$
Adam	$39.73_{(0.97)}$	+22%	$0.775_{(0.004)}$
SAM-Adam	$53.25_{(0.80)}$	+10%	$0.818_{(0.005)}$
bSAM (ours)	$\mathbf{62.64}_{(0.33)}$		$\mathbf{0.841}_{(0.004)}$

Memory

What is relevant from the past?

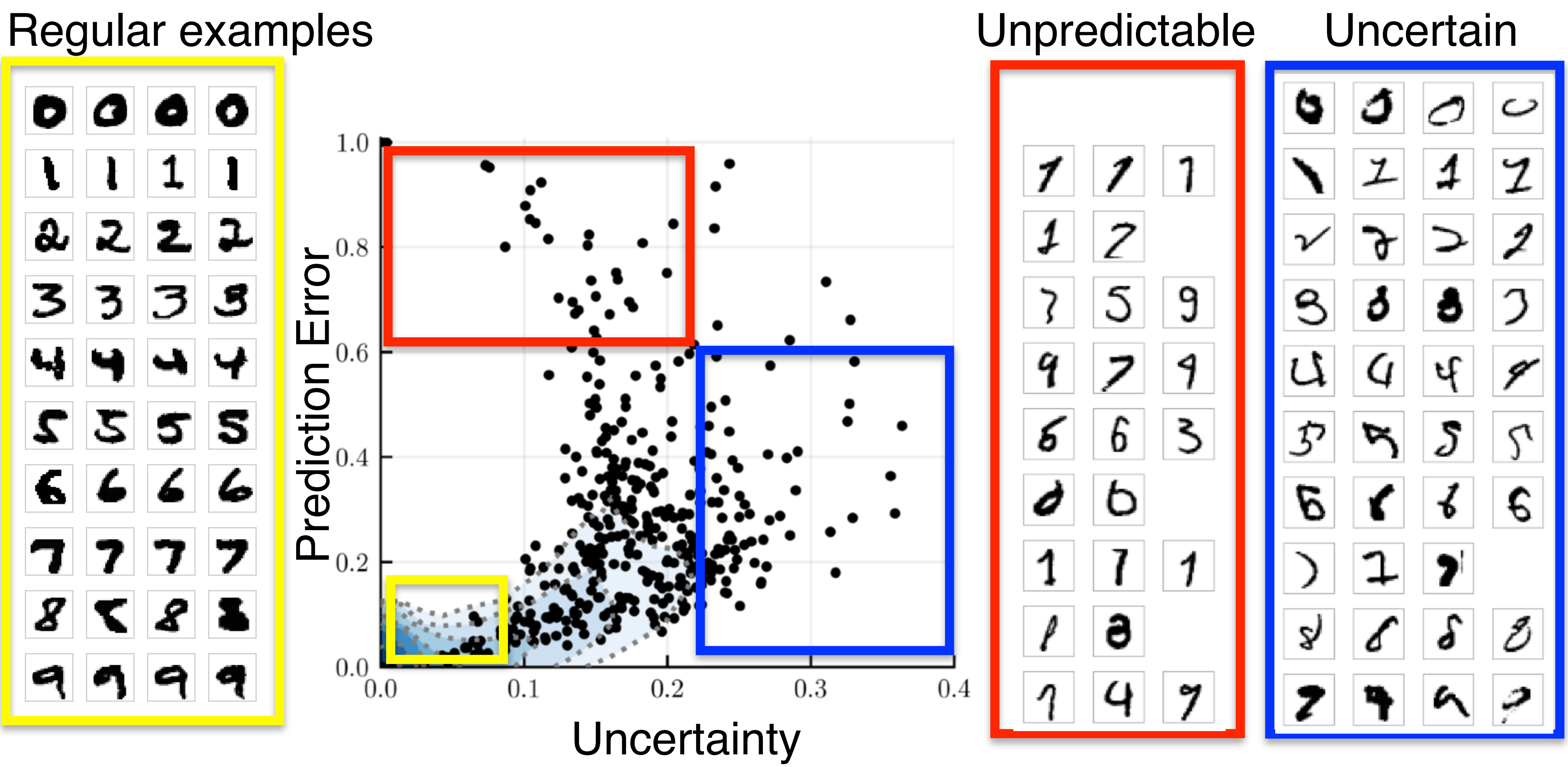
How to represent and adapt the knowledge? Perturbation, Sensitivity, and Duality

Bayes-Duality

https://tenor.com/view/clockwork-gears-brain-gif-16784329

Memory Maps using the BLR

Understand generic ML models and algorithms.



1. Tailor, Chang, Swaroop, Nalisnick, Solin, Khan, Memory maps to understand models (under review)

See Section 5.4 in Khan and Rue, 2021 for local parameterization

BLR Solutions & Their Duality

$$\ell(\theta) = \sum_{i=0}^{N} \ell_i(\theta) \qquad \lambda \leftarrow (1-\rho)\lambda - \sum_{i=0}^{N} \rho \nabla_{\mu} \mathbb{E}_q[\ell_i(\theta)]$$

Global natural parameter $\longrightarrow$ $$\lambda^* = \sum_{i=0}^{N} \underbrace{\nabla_{\mu^*} \mathbb{E}_{q^*}[-\ell_i(\theta)]}_{}$$

Local natural parameter $\longrightarrow$ $\widetilde{\lambda}_i^*$

Local parameters are Lagrange Multipliers, measuring the sensitivity of BLR solutions to local perturbation [1]. They can be used to tell apart relevant vs irrelevant data.

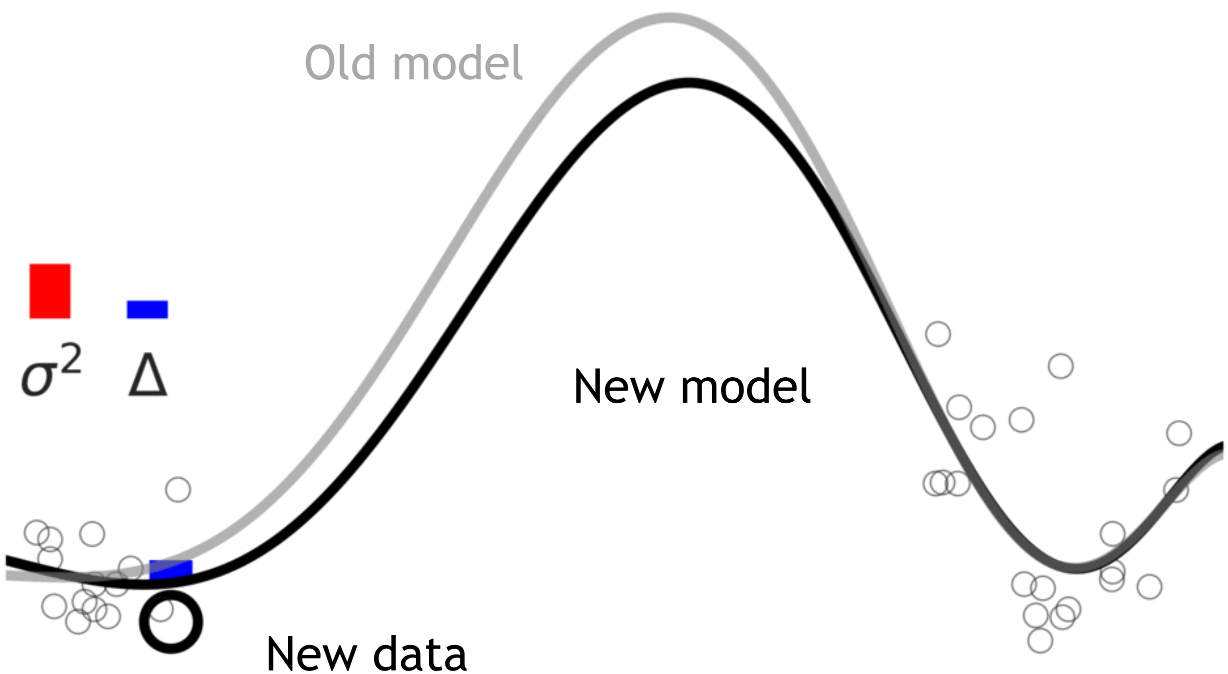
The main contribution is that we can do this "during training" for a wide-variety of ML algorithms and models.

Memory Perturbation

How sensitive is a model to its training data?

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

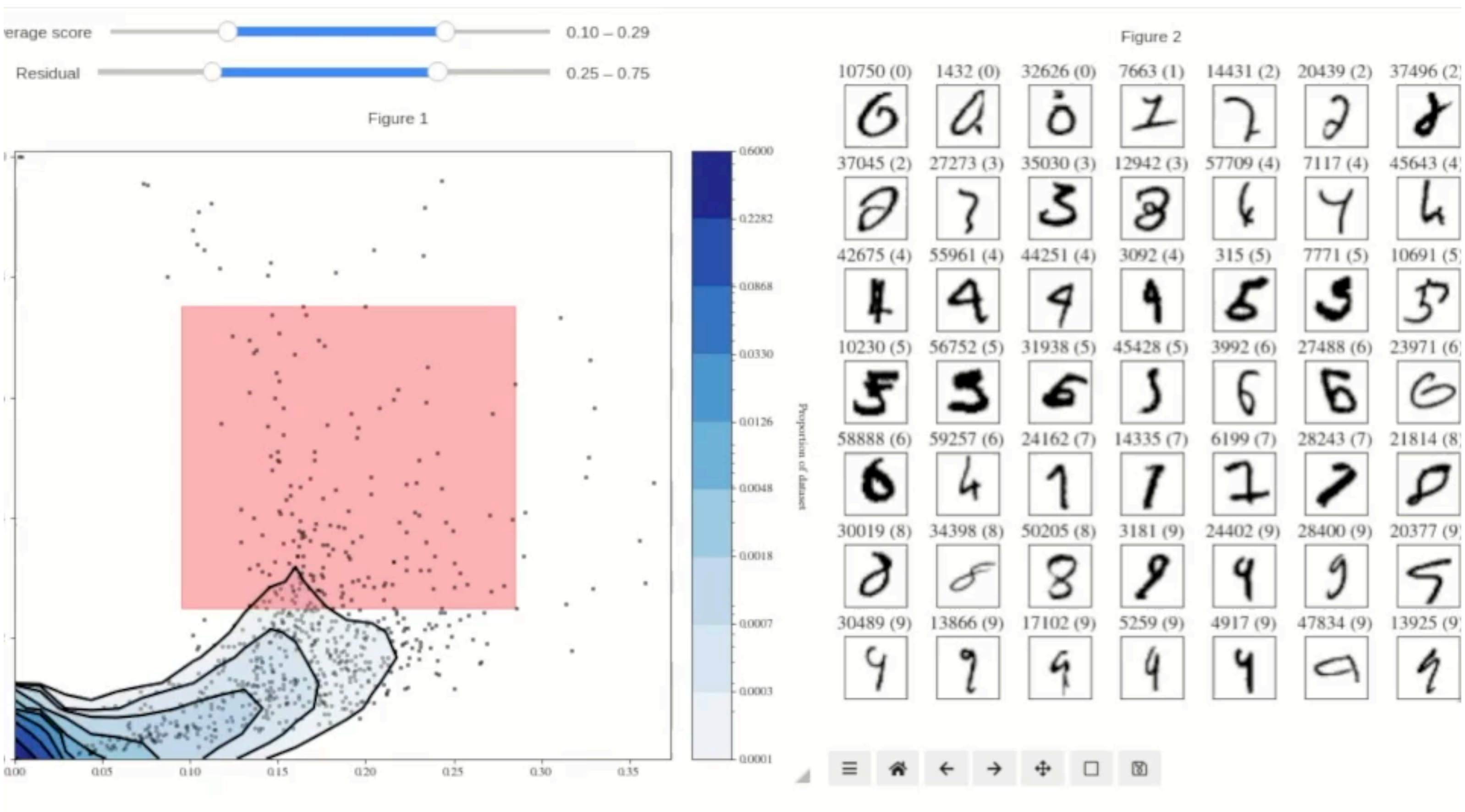
Model-deviation (Δ) = predictability * Uncertainty



1. Cook. Detection of Influential Observations in Linear Regression. Technometrics. ASA 1977
2. Nickl, Xu, Tailor, Moellenhoff, Khan, The memory-perturbation equation (under review)

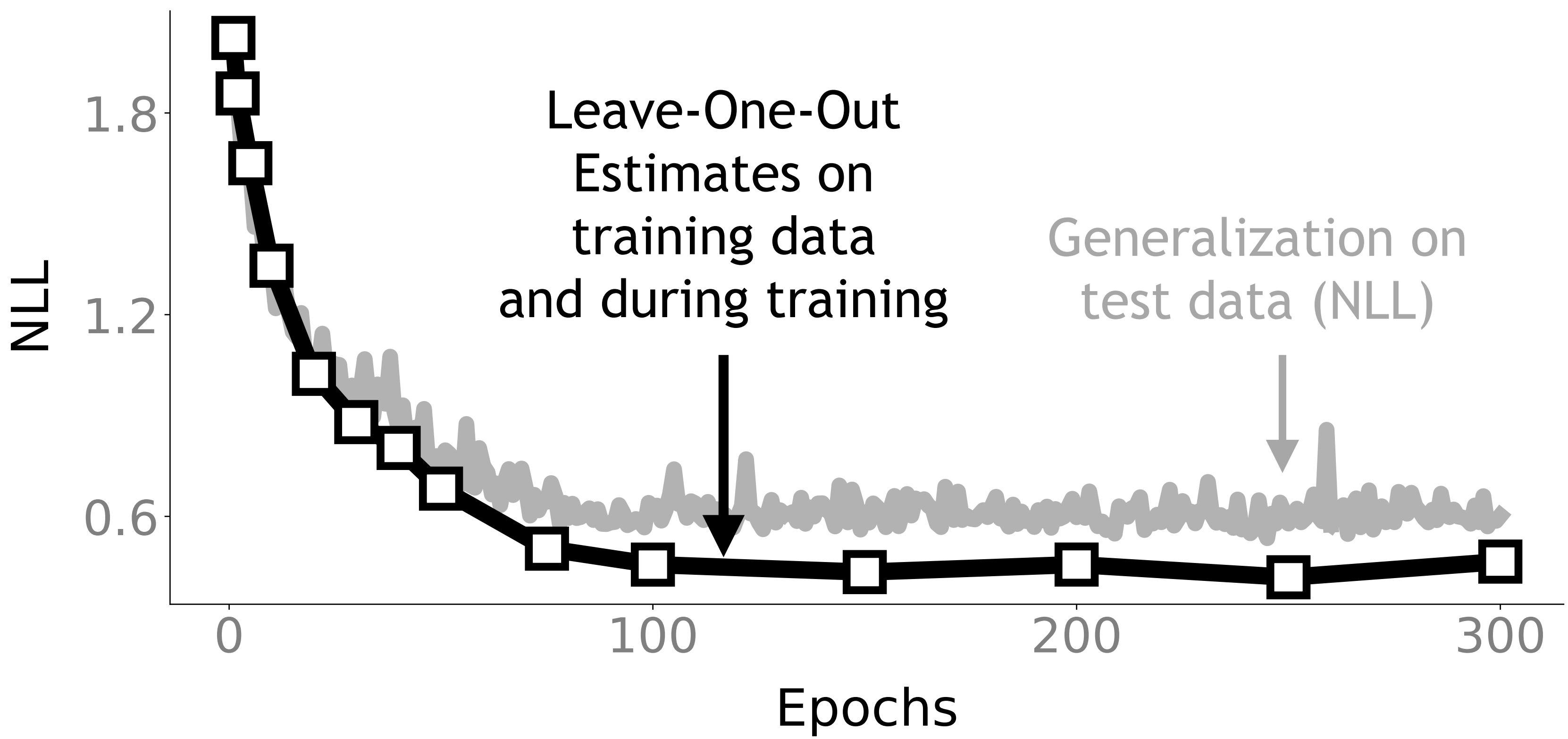
A Tool for Data-Scientists

Understand the memory of a model.



Predict Generalization during Training

CIFAR10 on ResNet-20 using iVON [1]. Adam also works but better uncertainty gives better estimates.



1. Lin et al. "Handling the positive-definite constraints in the BLR." ICML (2020).

Summary

- Through posterior approximations, the criteria to categorize examples naturally emerges
  - Generalizes existing concepts such as support vectors, influence functions, inducing inputs etc
- Applies to almost all ML problem
  - Supervised, unsupervised, RL
  - Discrete/continuous losses and parameters
- No extra computation needed
- A measure of generalization (model complexity)
- The sensitivity of posterior leads to “Bayes Duality”

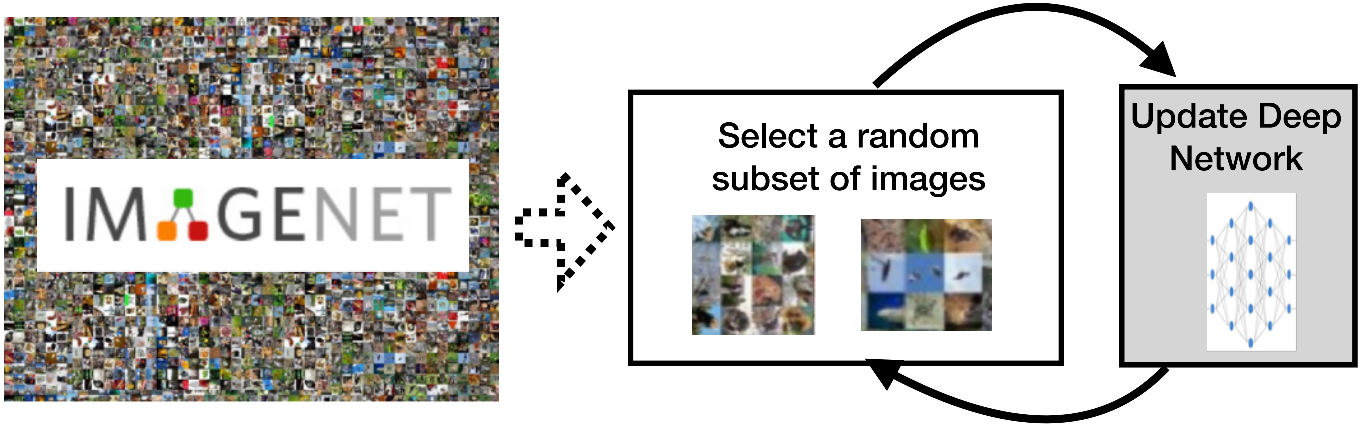
1. Nickl, Xu, Tailor, Moellenhoff, Khan, The memory-perturbation equation (under review)

Adaptation

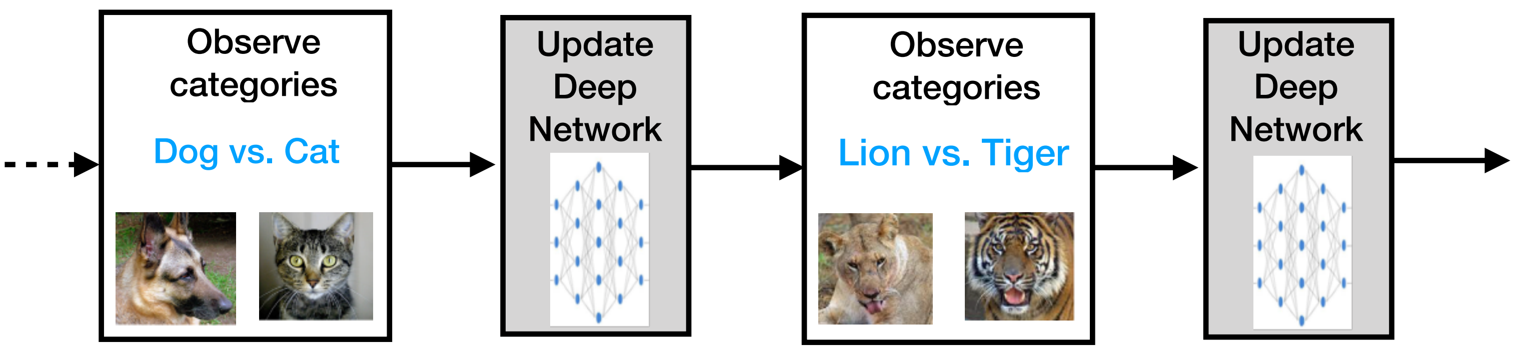
Transfer knowledge without
forgetting the past

Example: Continual Learning

Standard Deep Learning



Continual Learning: past classes never revisited

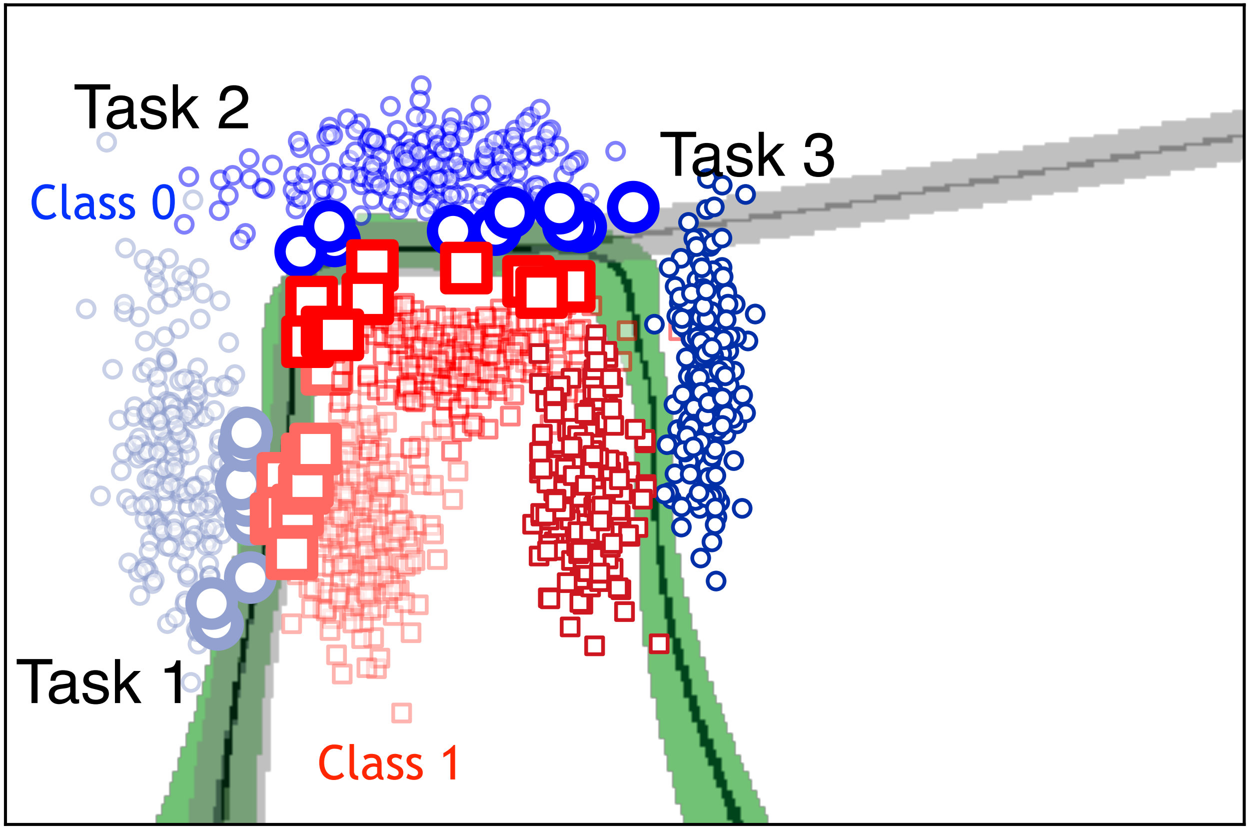


Standard training leads to catastrophic forgetting.

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

Continual Learning

Avoid forgetting by using “memorable examples” [1,2]



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

Functional Regularization of Memorable Past (FROMP) [4]

Standard way to is to add a weight-regularizer [1]

$$(\theta - \theta_{\text{old}})^\top F_{\text{old}}(\theta - \theta_{\text{old}})$$

Weight uncertainty

We add functional regularizer [2]

$$[\sigma(\mathbf{f}(\theta)) - \sigma(\mathbf{f}_{old})]^\top K_{old}^{-1}[\sigma(\mathbf{f}(\theta)) - \sigma(\mathbf{f}_{old})]$$

Uncertainty

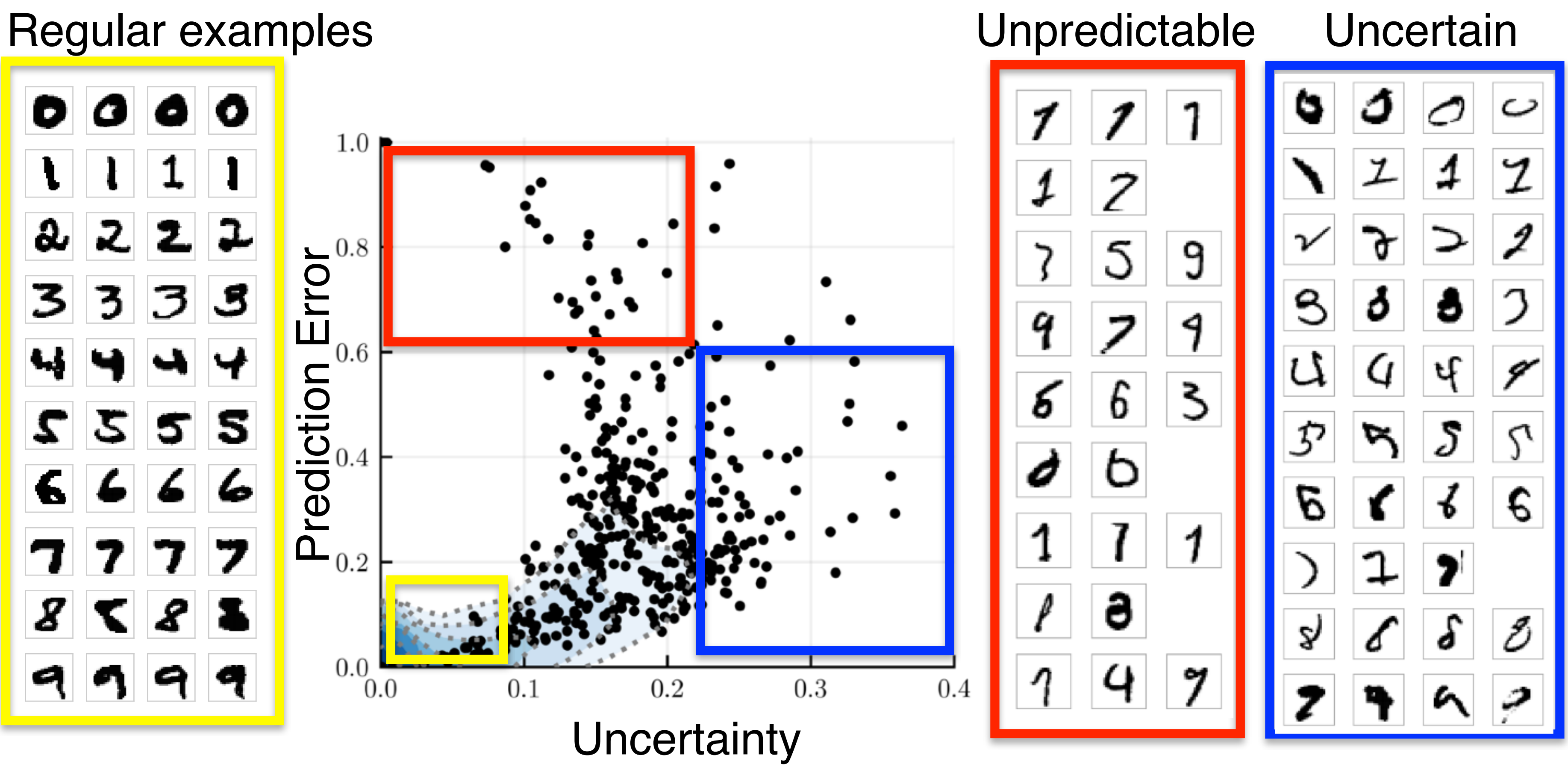
Predictions

Why does this work?

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

Back to the Memory Map

Highly sensitive examples are crucial for adaptation



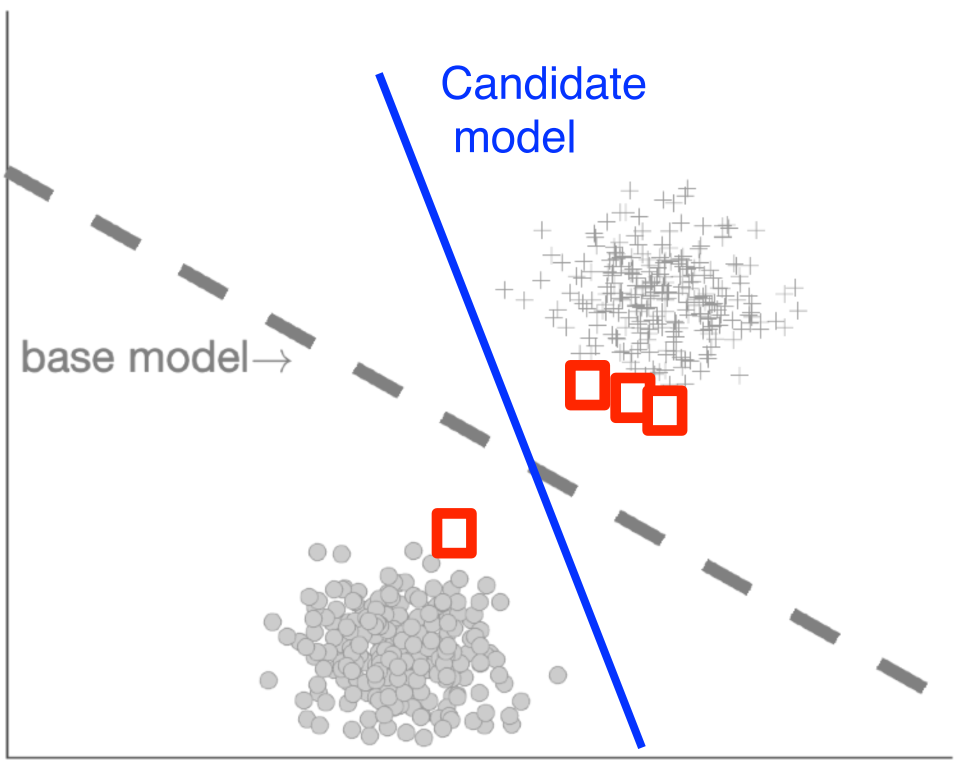
1. Tailor, Chang, Swaroop, Nalisnick, Solin, Khan, Memory maps to understand models (under review)

Knowledge-Adaptation Priors

Combine weight and function-space divergences

Weight-space Function-space

$$\mathcal{K}(\theta) = \tau \mathbb{D}_w(\theta \| \theta_{\text{old}}) + \mathbb{D}_f(\mathbf{f}(\theta) \| \mathbf{f}(\theta_{\text{old}}))$$



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

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

How to Choose Memory?

Minimize the error in the gradients

$$\nabla l_{\text{old}}(\theta) - \nabla K(\theta)$$

$$= \sum_{i \in \mathcal{D} \setminus \mathcal{M}} \nabla f_i(\theta) \left[\sigma(f_i(\theta)) - \sigma(f_i(\theta_{\text{old}})) \right]$$

Prediction disagreement

Past and future should agree. There are some general rules to ensure this, but no magic. In general, we must understand sensitivity of the past and future using natural gradients.

1. Pan et al. Continual deep learning by functional regularisation of memorable past. NeurIPS, 2020.

Towards Quick Adaptation

- Unify, generalize and improve algorithms
  - Bayesian Learning rule (BLR)
- Memory (or representation)
  - Sensitivity and dual view of the BLR
- Adaptation (or transfer)
  - Continual learning and K-priors
  - Use sensitivity to adapt quickly

The webpage is available at https://bayesduality.github.io/, and Twitter account @BayesDuality

The Bayes-Duality Project

Toward AI that learns adaptively, robustly, and continuously, like humans

Emtiyaz Khan

Research director
(Japan side)

Approx-Bayes team at RIKEN-AIP and OIST

Julyan Arbel

Research director
(France side)

Statify-team, Inria Grenoble Rhône-Alpes

Kenichi Bannai

Co-PI (Japan side)

Math-Science Team at RIKEN-AIP and Keio University

Rio Yokota

Co-PI
(Japan side)

Tokyo Institute of Technology

Received total funding of around USD 3 million through JST's CREST-ANR and Kakenhi Grants.

Approximate Bayesian Inference Team

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

Emtiyaz Khan
Team Leader

Thomas Möllenhoff
Research Scientist

Geoffrey Wolfer
Special Postdoctoral Resesarcher

Hugo Monzón Maldonado
Postdoctoral Researcher

Keigo Nishida
Postdoctoral Researcher
RIKEN BDR

Gian Maria Marconi
Postdoctoral Researcher

Lu Xu
Postdoctoral Researcher

Peter Nickl
Research Assistant

Etash Guha
Intern
Georgia Tech

Joseph Austerweil
Visiting Scientist
University of Winsconsin-Madison

Pierre Alquier
Visiting Scientist
ESSEC Business School

Dharmesh Tailor
Remote Collaborator
University of Amsterdam

Many thanks to our group members and collaborators (many not on this slide).

We have open positions and are always looking for new collaborations.