Lifelong Learning with Bayes

Emtiyaz Khan
Gian Maria Marconi
& Lu Xu

RIKEN Center for AI Project, Tokyo
https://team-approx-bayes.github.io/

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

How to make AI that can adapt quickly?

Humans and animals are extremely good at this

Human Learning at the age of 6 months.

Converged at the
age of 12 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

Adaptation in Machine Learning

- Machines are bad in quickly adapting to changes
  - Even small changes require a complete retraining-from-scratch
  - This is expensive, time consuming [1,2]
  - Example: Tesla AI Data-Engine for “self-driving cars” takes 70000 GPU hrs [3]
- Difficult to apply to domains with “dynamic” setting
  - Robotics, medicine, user interaction, epidemiology, climate science, etc.

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

Summary

- Why Bayes?
- Lifelong learning with Bayes
  - Use simple estimates of uncertainty
  - Use memory, sensitivity etc.
- A (simple) method to get good uncertainty out of Deep-Learning optimizers

Why Bayes?

Because uncertainty!

Principle of Trial-and-Error

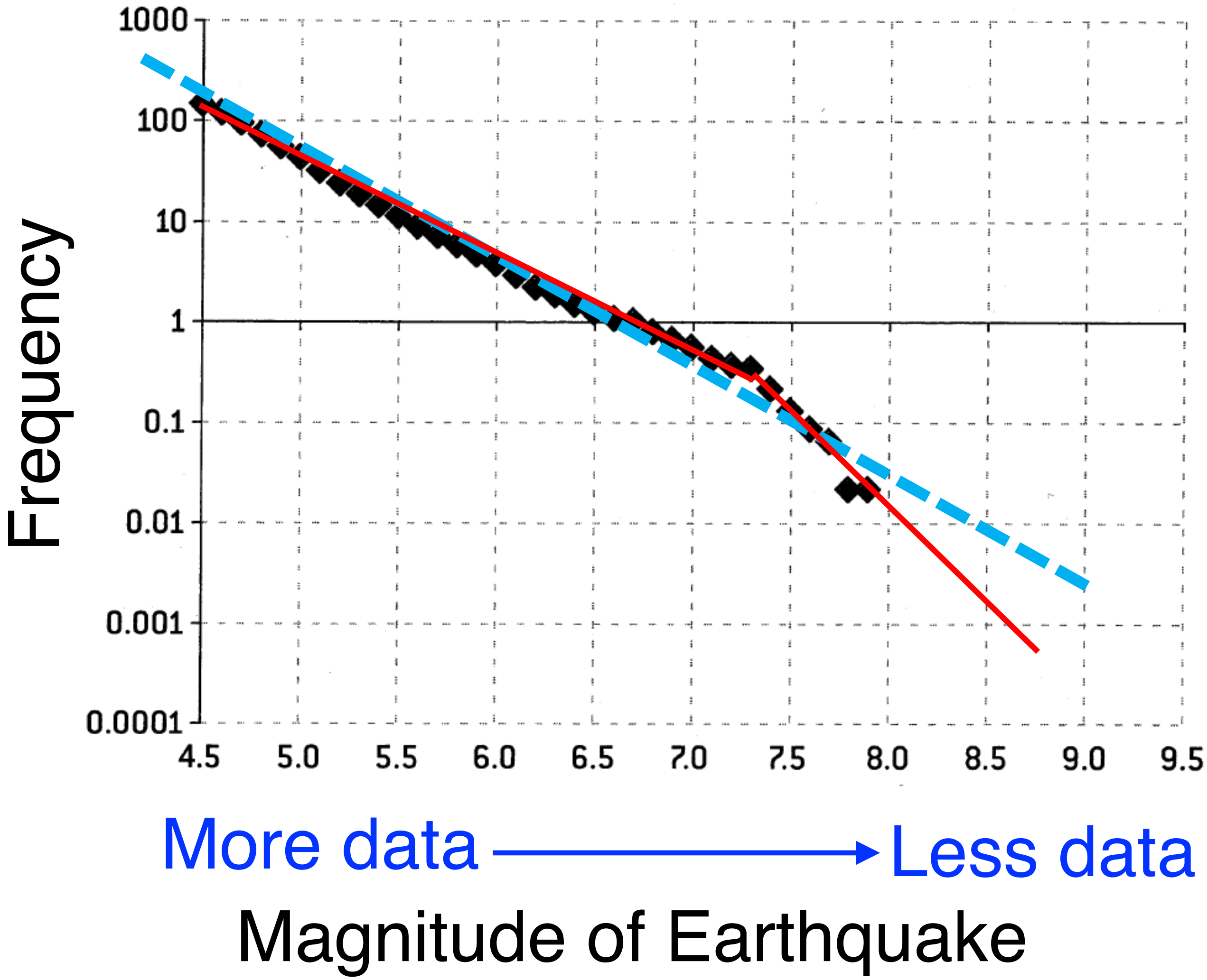
Frequentist: Empirical Risk Minimization (ERM) or Maximum Likelihood Principle, etc.

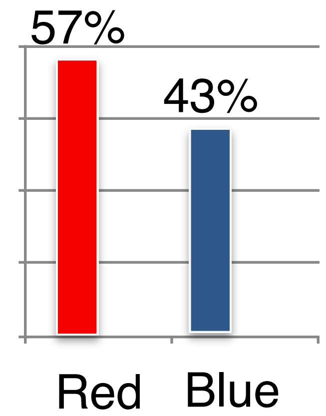
$$\min_{\theta} \ \underbrace{\ell}_{\text{Loss}}(\underbrace{\mathcal{D}}_{\text{Data}}, \underbrace{\theta}_{\text{Model Params}}) = \sum_{i=1}^{N} [y_i - \underbrace{f_\theta}_{\text{Deep Network}}(x_i)]^2 + \gamma \theta^T \theta$$

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.

Example: Which is a Better Fit?

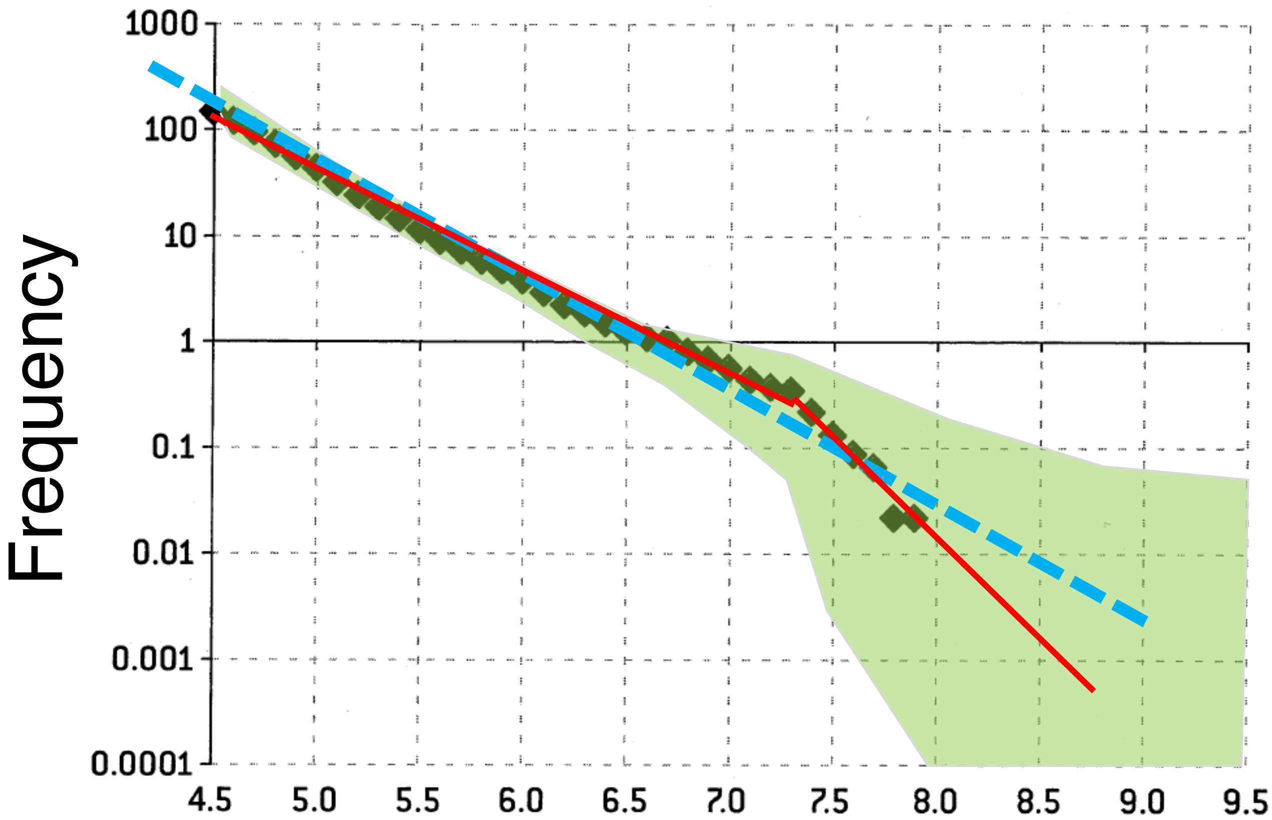




Red is more risky than the blue

Real data from Tohoku (Japan). Example taken from Nate Silver's book "The signal and noise"

Example: Which is a Better Fit?



More data → Less data

Magnitude of Earthquake

Uncertainty: "What the model does not know"

Choose less risky options!

Avoid data bias with uncertainty!

Real data from Tohoku (Japan). Example taken from Nate Silver's book "The signal and noise"

Bayesian Principles

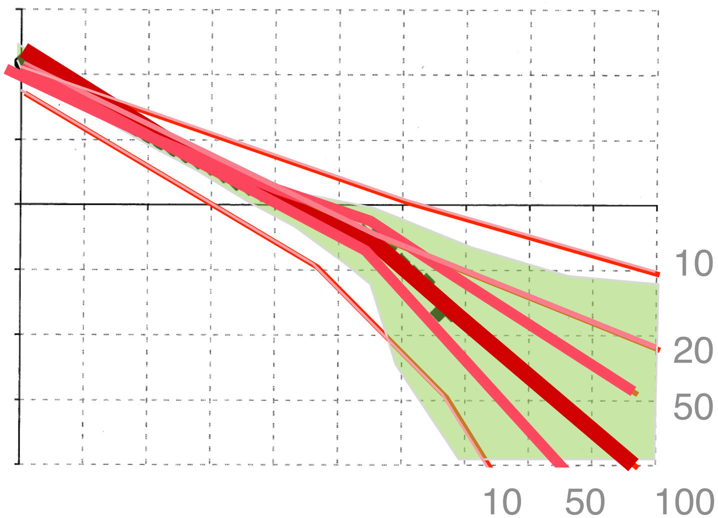
1. Sample $\theta \sim p(\theta)$ prior

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

3. Normalize

Posterior Likelihood x Prior

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

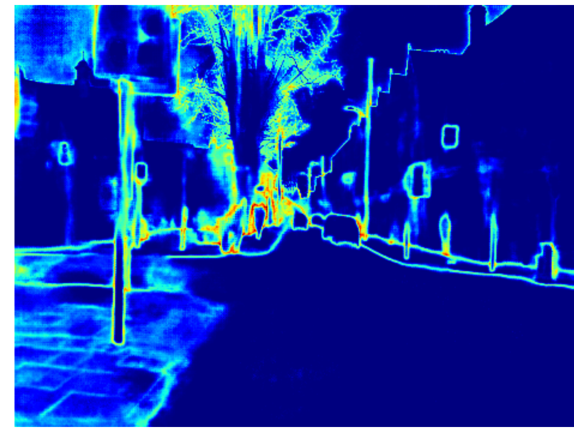


A global method: Integrates over all models
Does not scale to large problem

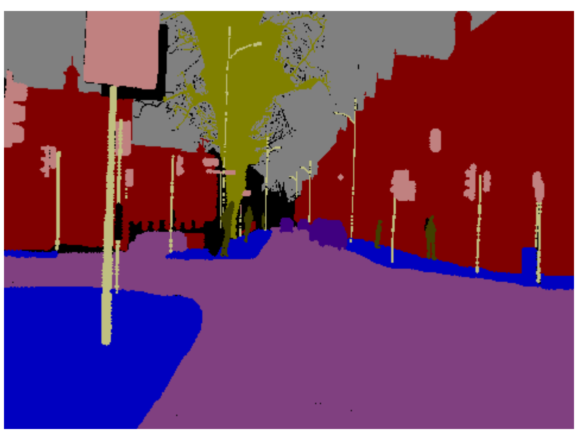
Uncertainty Estimates for Image Segmentation

Image

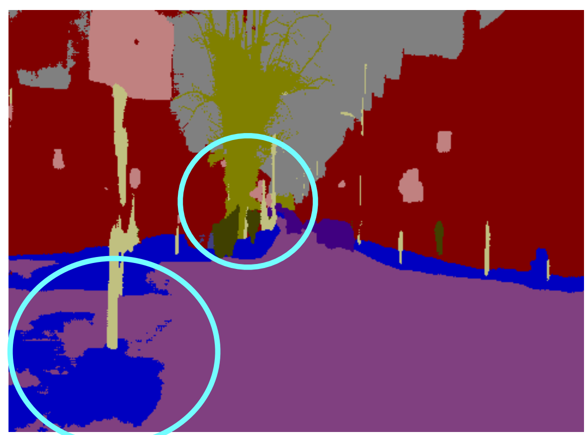
Uncertainty

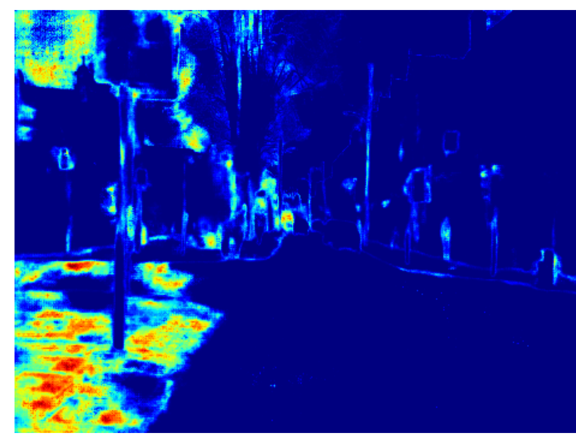


True Segments



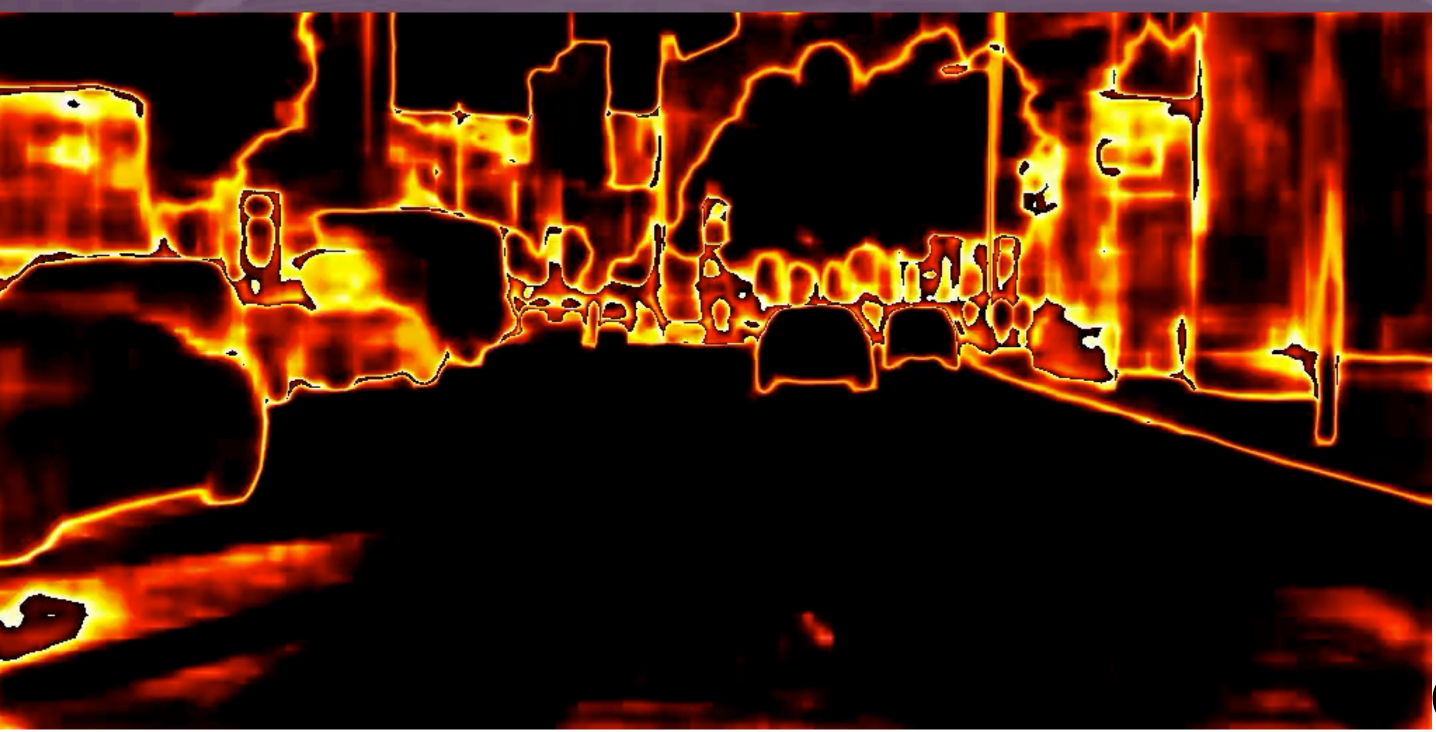
Prediction





Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." *CVPR*. 2018.

Image Segmentation



Uncertainty (entropy of class probs)

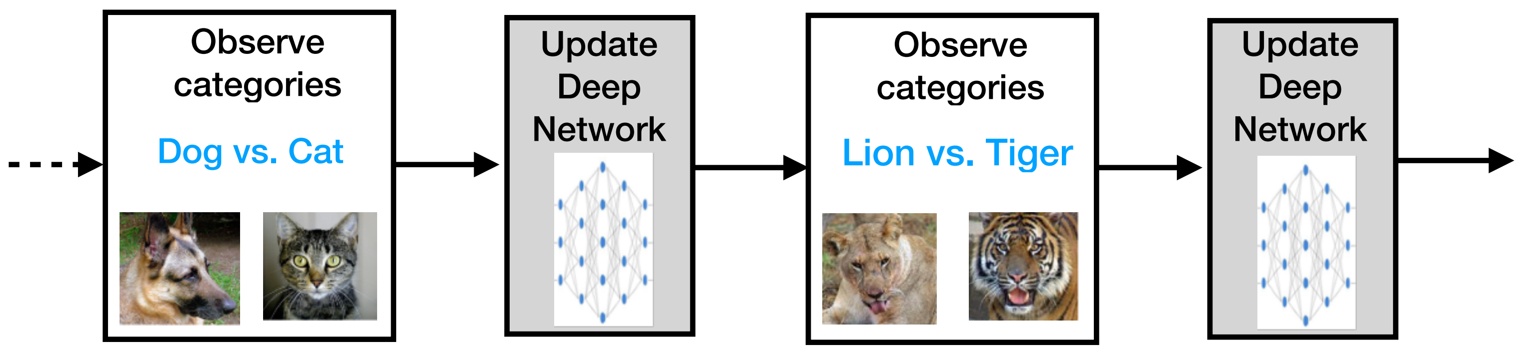
(By Roman Bachmann)

What about lifelong continual learning?

Lifelong Continual Learning

Standard Deep Learning

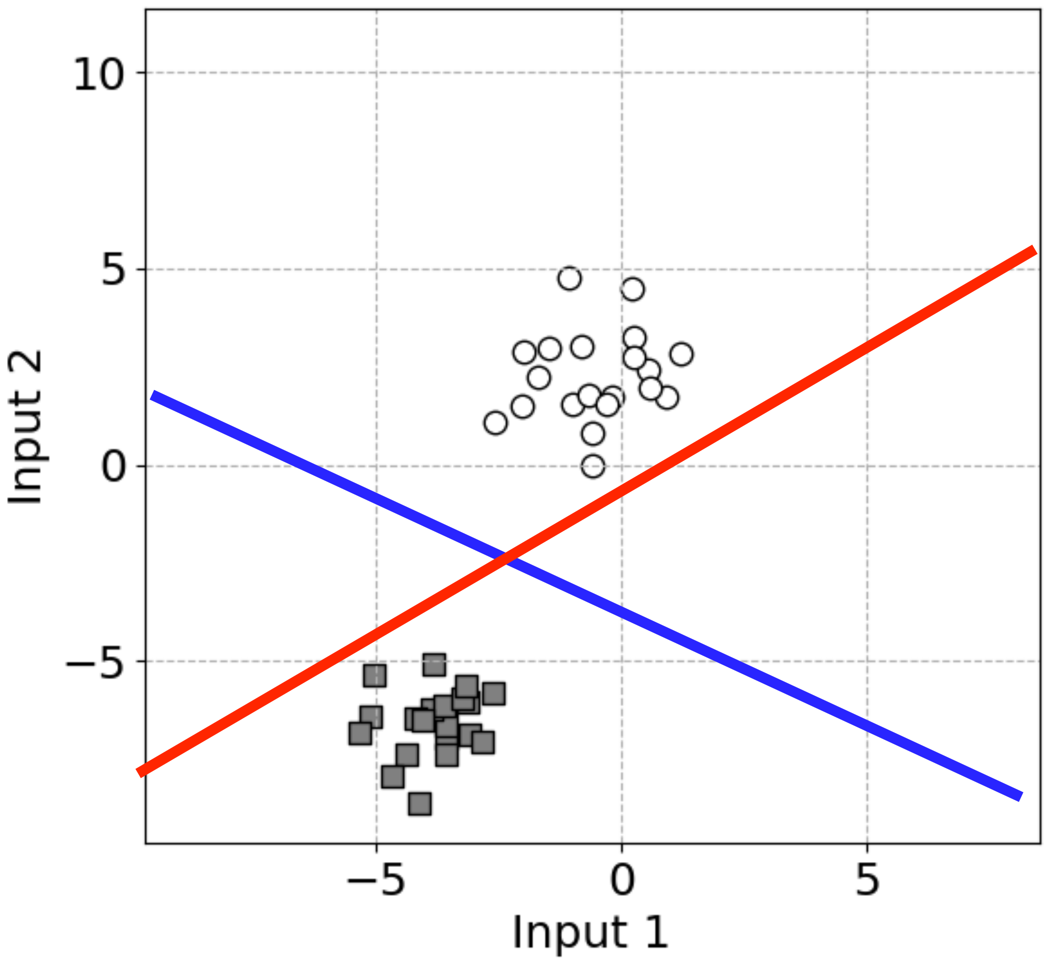
Continual Learning: past classes never revisited



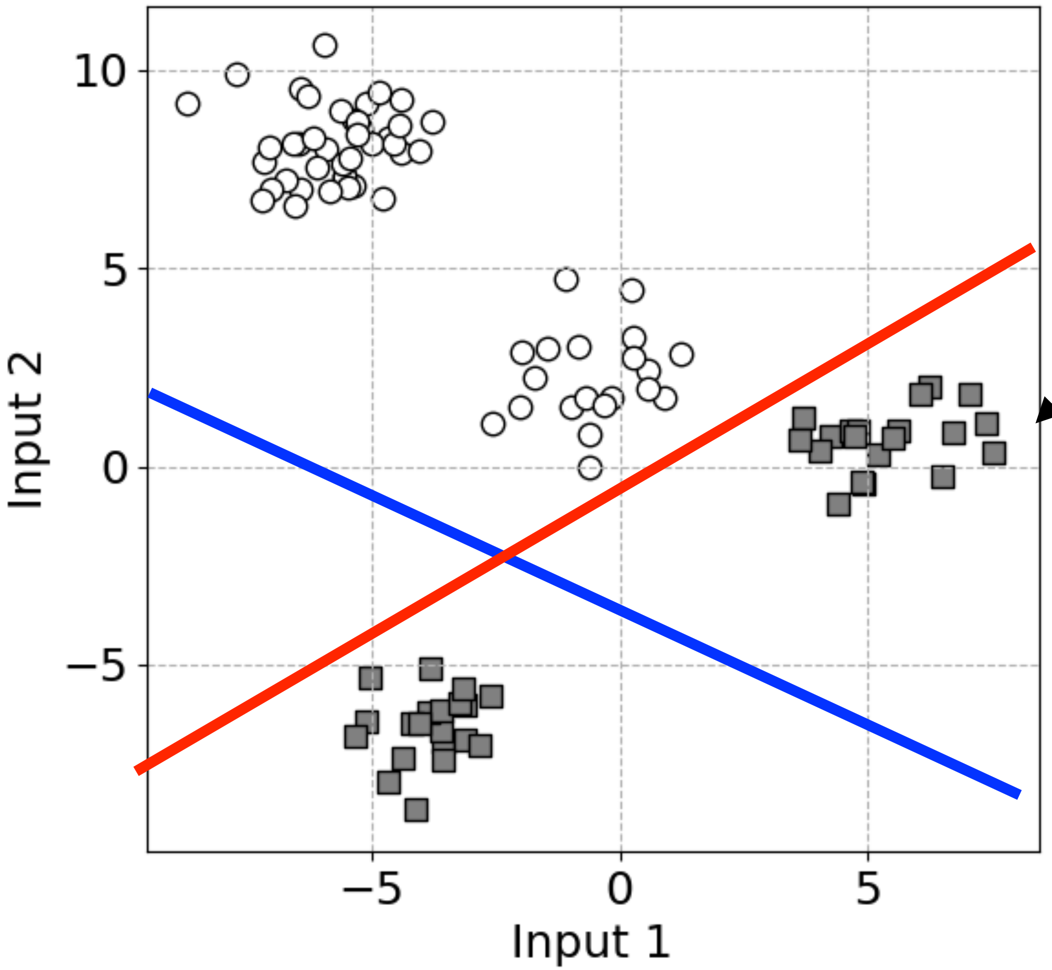
Standard training leads to catastrophic forgetting.

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

Which is a good classifier?

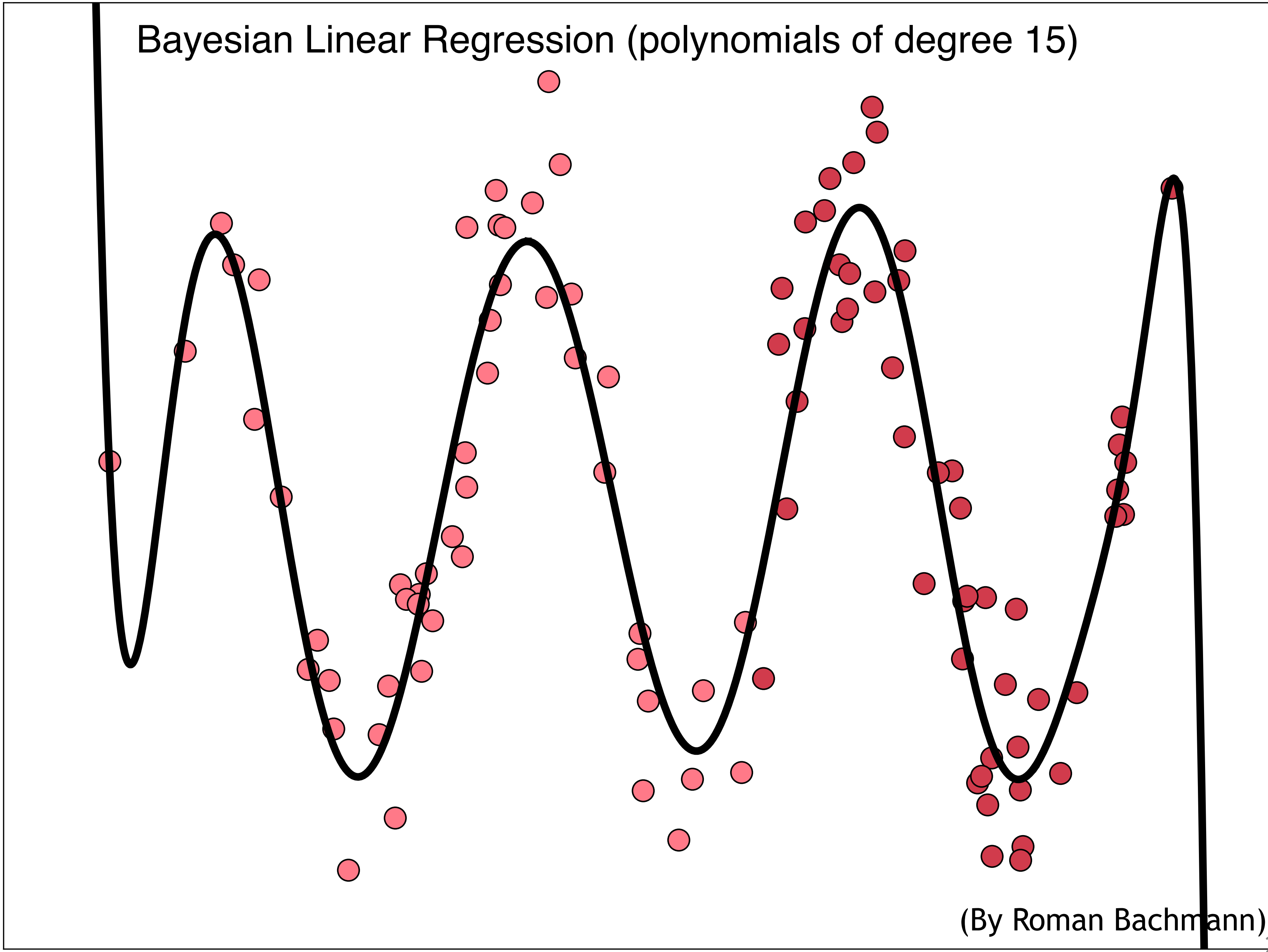


Which is a good classifier?

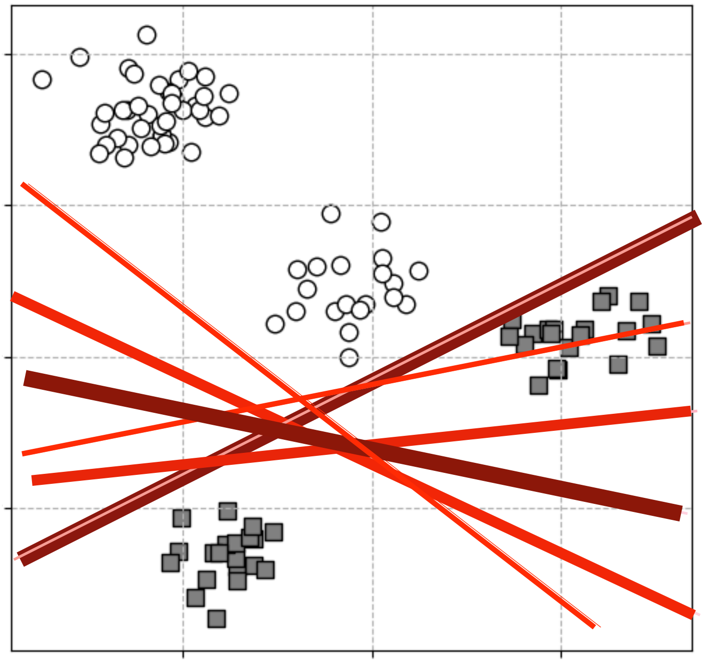


Misclassified by the red line, but not by the blue

What you don't know now, can hurt you later
"Uncertainty matters"


Bayesian Linear Regression (polynomials of degree 15)
(By Roman Bachmann)

Bayesian Principles



(1) Keep your options open

$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta)p(\theta)}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

(2) Revise with new evidence

$$p(\theta|\mathcal{D}_2, \mathcal{D}_1) = \frac{p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)}{\int p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)d\theta}$$

Similar ideas in sequential/online decision-making (uncertainty/randomization). Computation is infeasible.

Weight regularizers

Computing posteriors exactly is infeasible, but we could approximate them [1]. One option is to use weight regularizer known as the Elastic-Weight Consolidation (EWC)

$$\log p(\theta|\mathcal{D}_{\text{old}}) \approx -\frac{1}{2}(\theta - \theta_{\text{old}})^\top S_{\text{old}}(\theta - \theta_{\text{old}})$$

Weight uncertainty
(Hessian/Fisher etc.)

Gianma and Lu will show later how to compute S_old within a deep-learning optimizer.

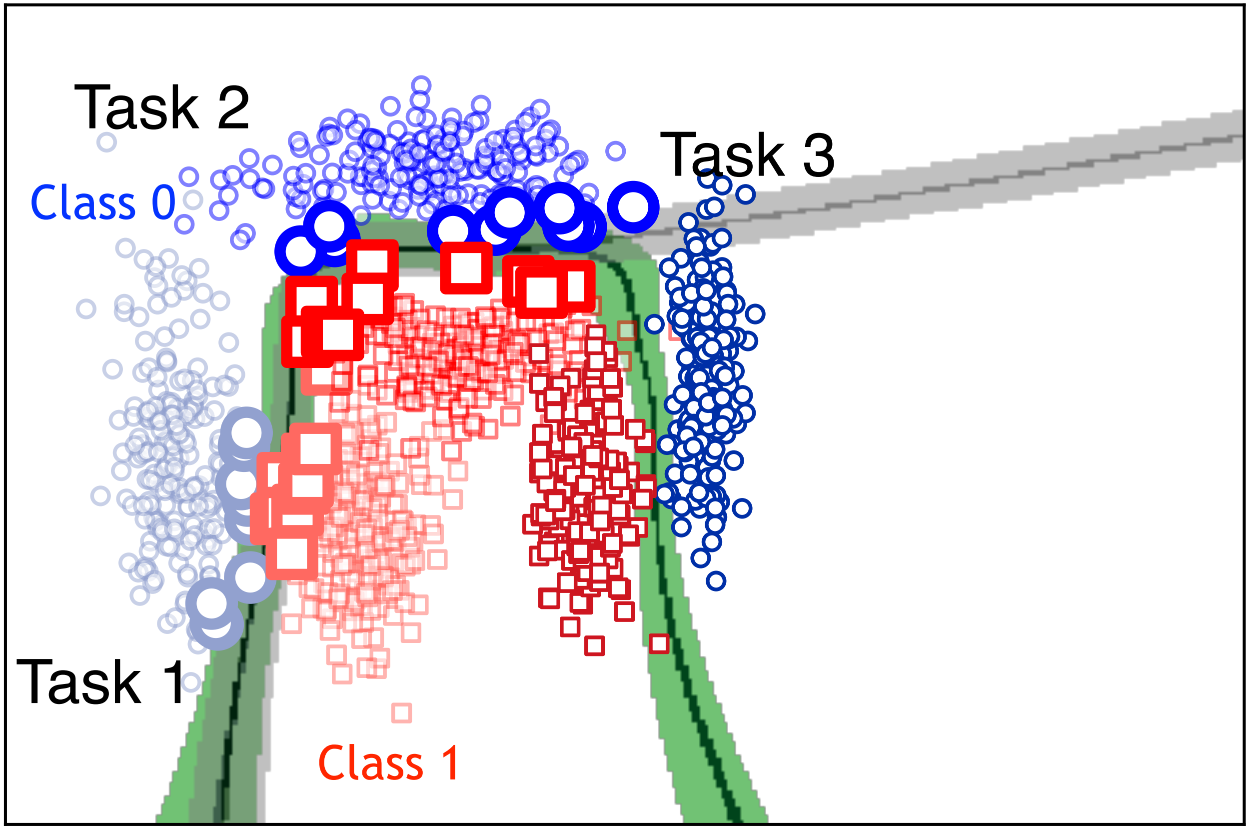
1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS* 2017

Uncertainty = Memory = Sensitivity

An out of the box idea!

Memory-based Methods

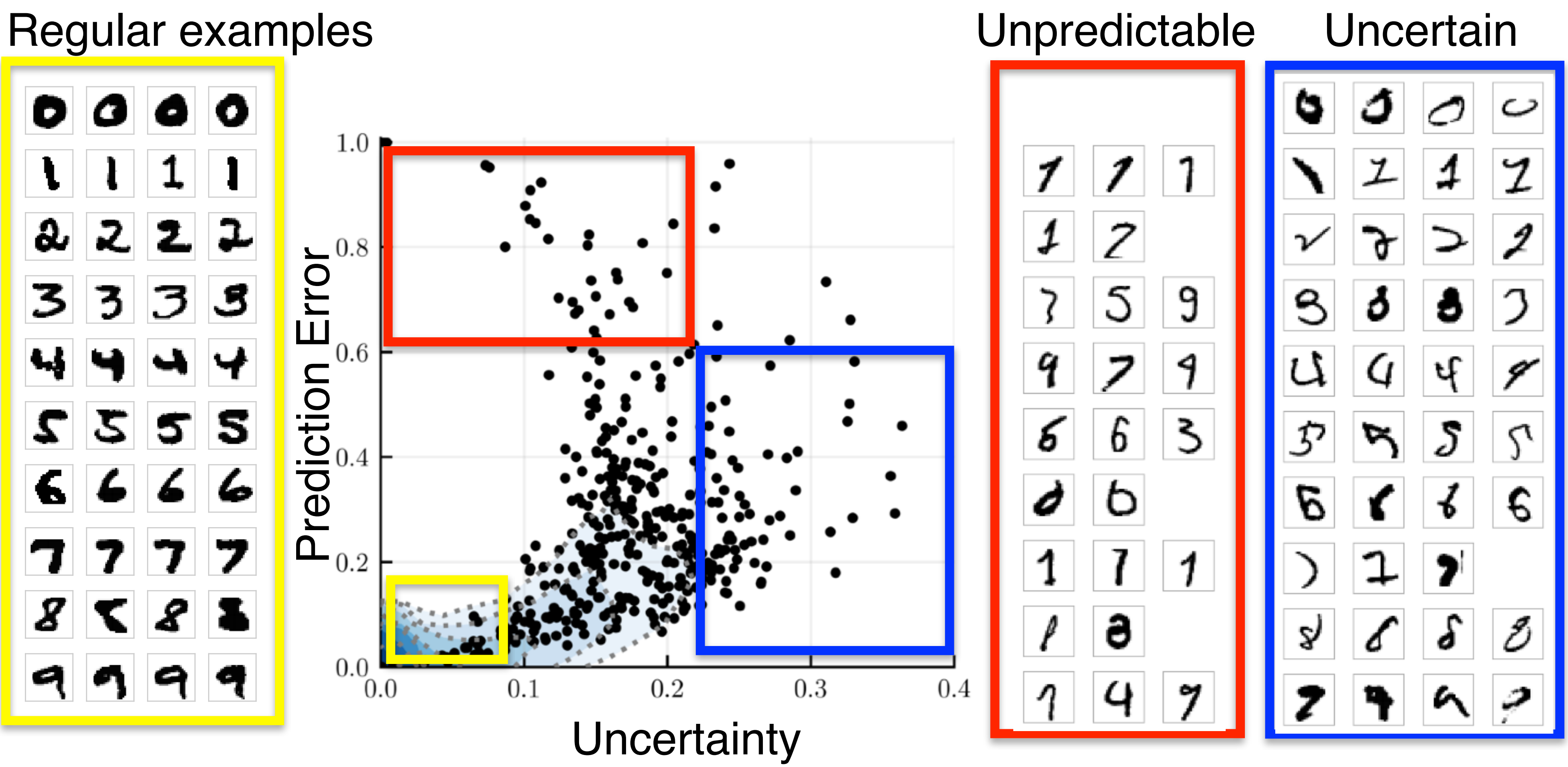
Avoid forgetting by using "memorable examples" [1,2]



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

Memory (as sensitivity) Maps

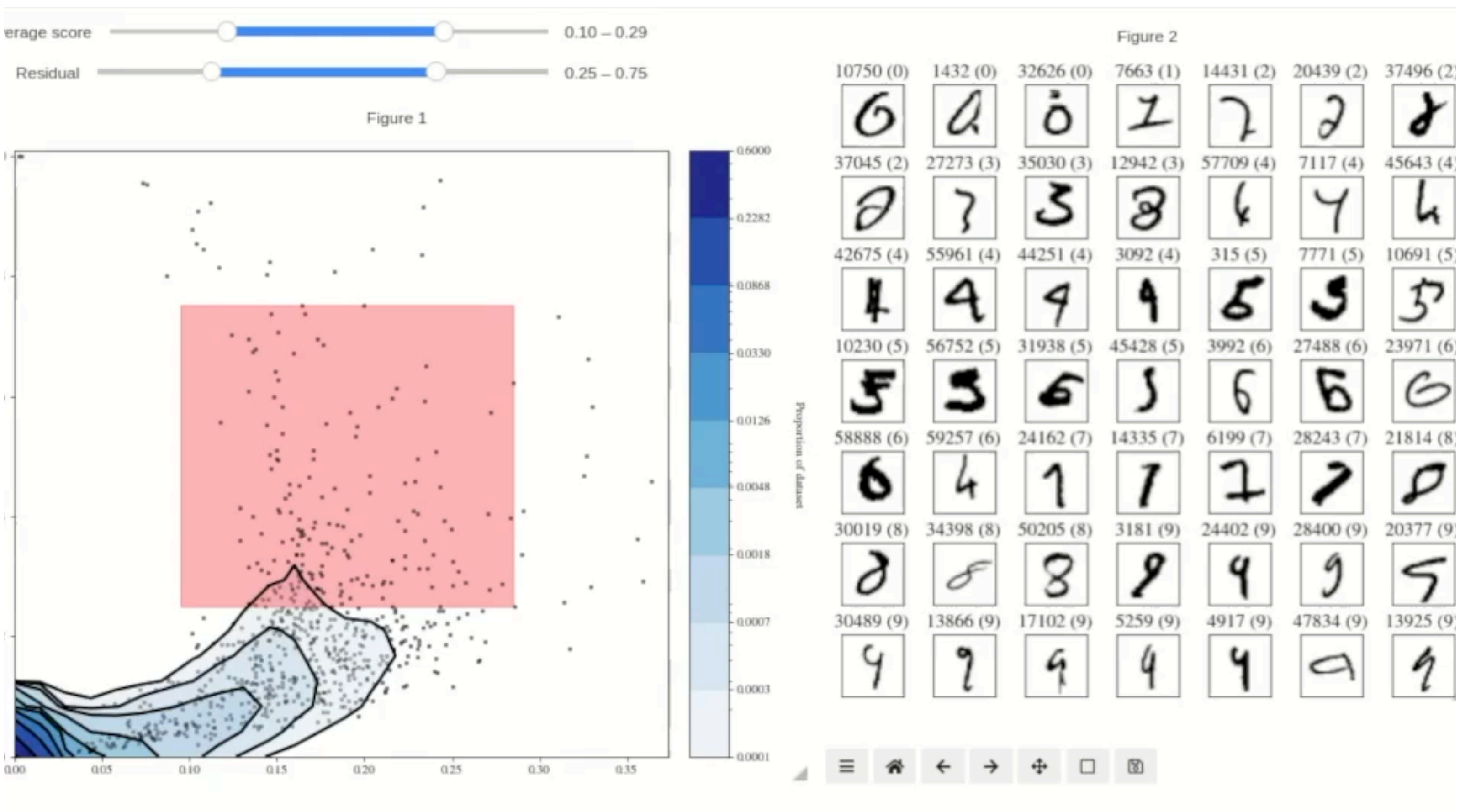
Highly sensitive examples: crucial for lifelong learning



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

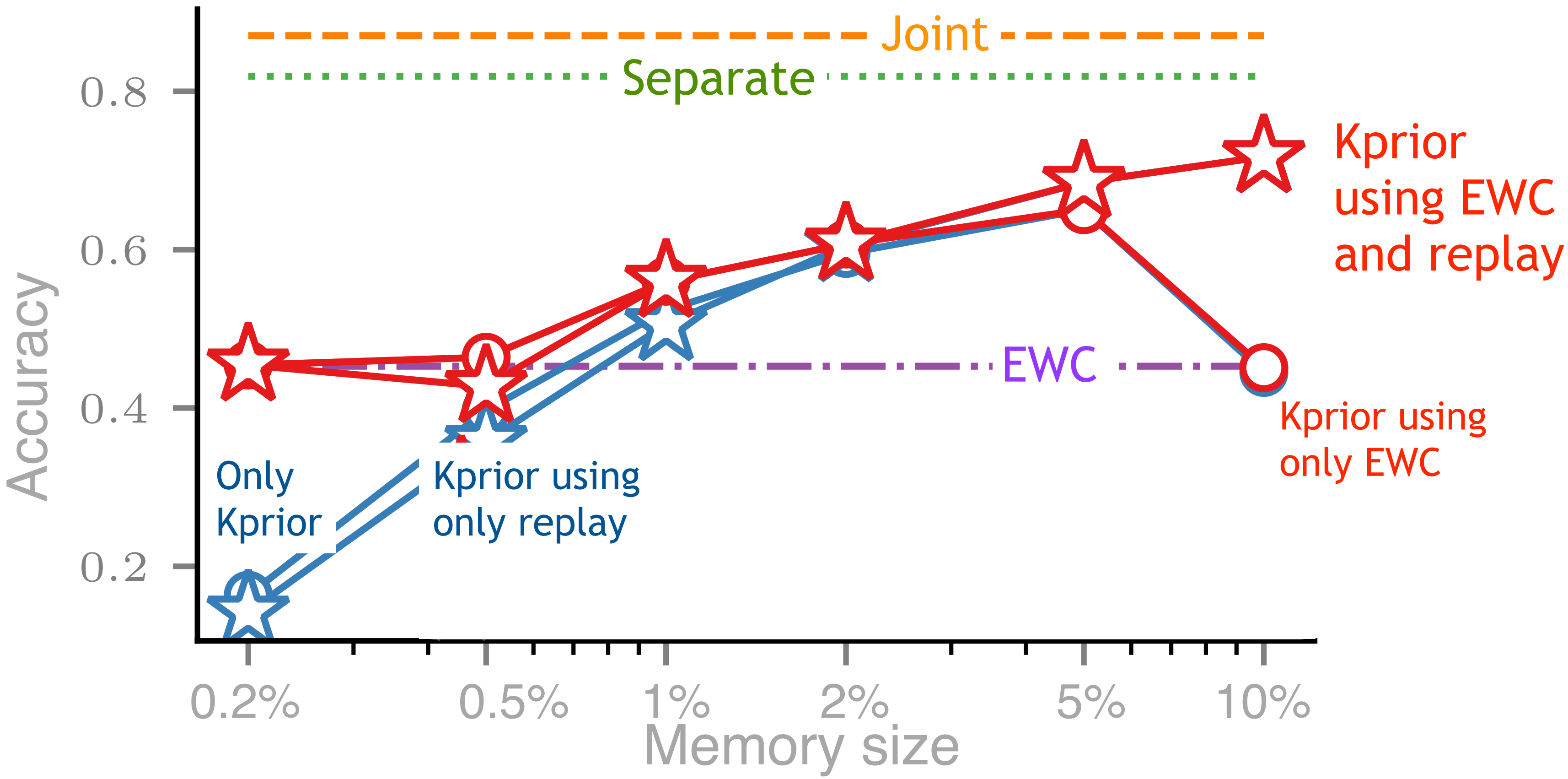
A Tool for Data-Scientists

Understand the memory of a model.



Continual Learning on ImageNet

K-prior allows us to optimally combine model and data to get good accuracy with little memory.



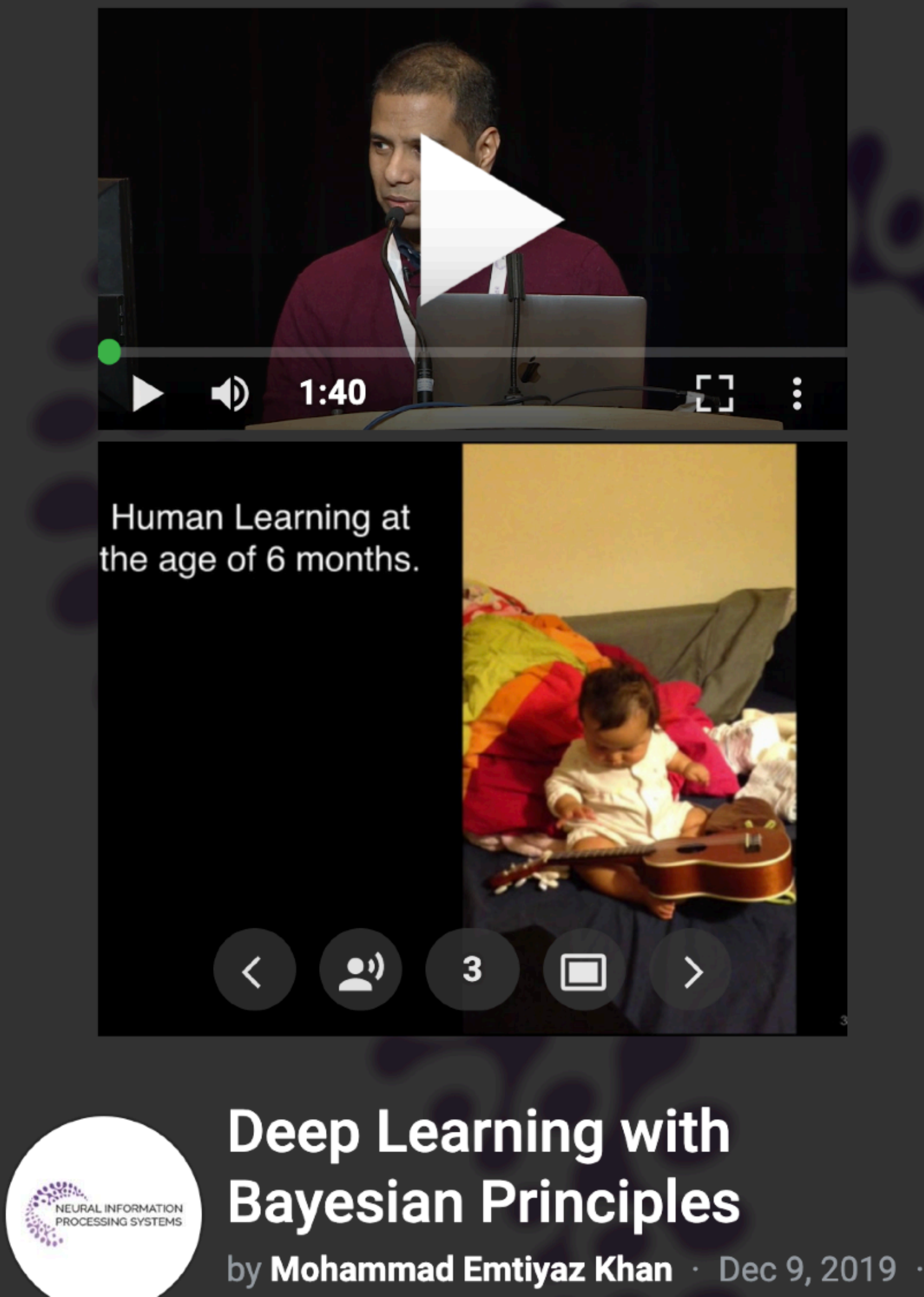
1. Khan and Swaroop, Knoweldge-Adaptation Priors, NeurIPS 2021
2. Daxberger et al. Improving CL by Accurate Gradient Reconstruction of the Past (under review).

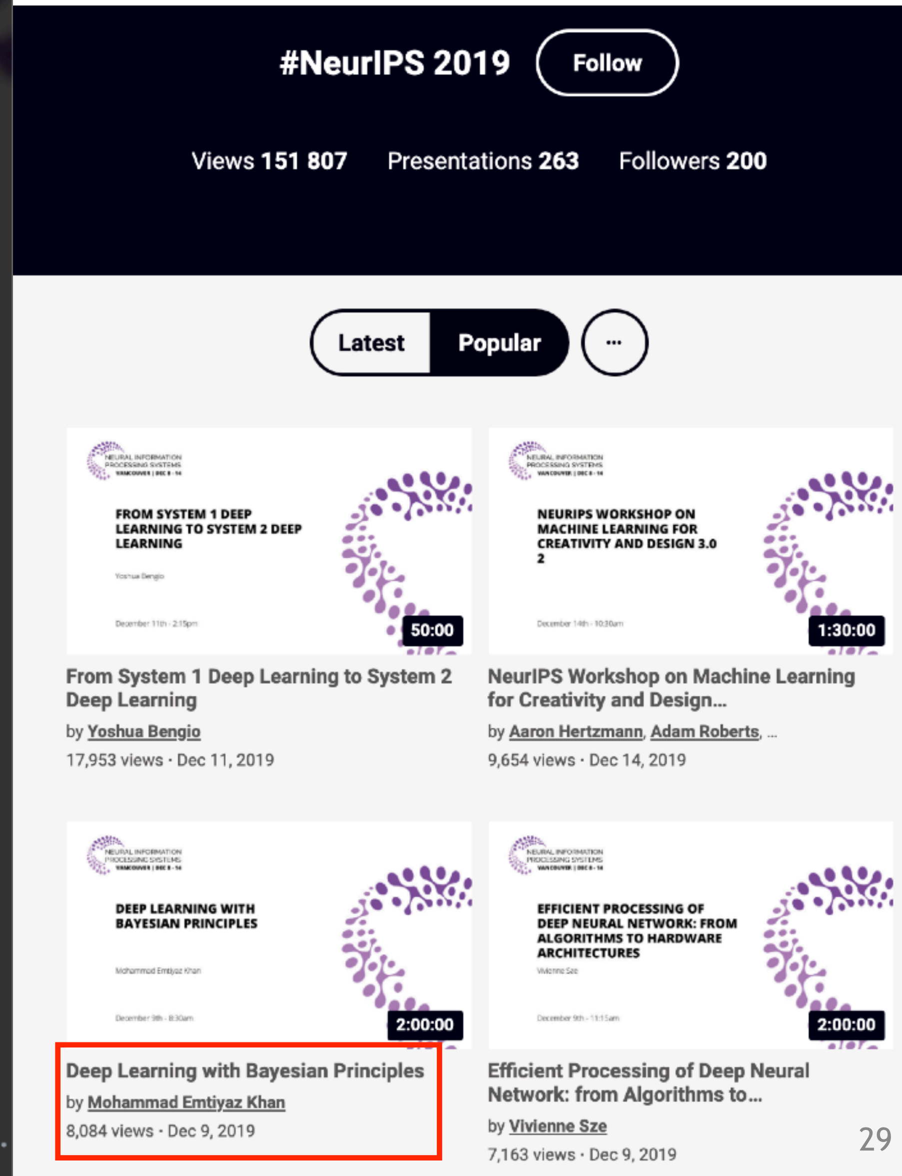
How to compute uncertainty for deep learning?

Algorithms as special cases of the Bayesian Learning Rule [1], which allows us to add uncertainty for free

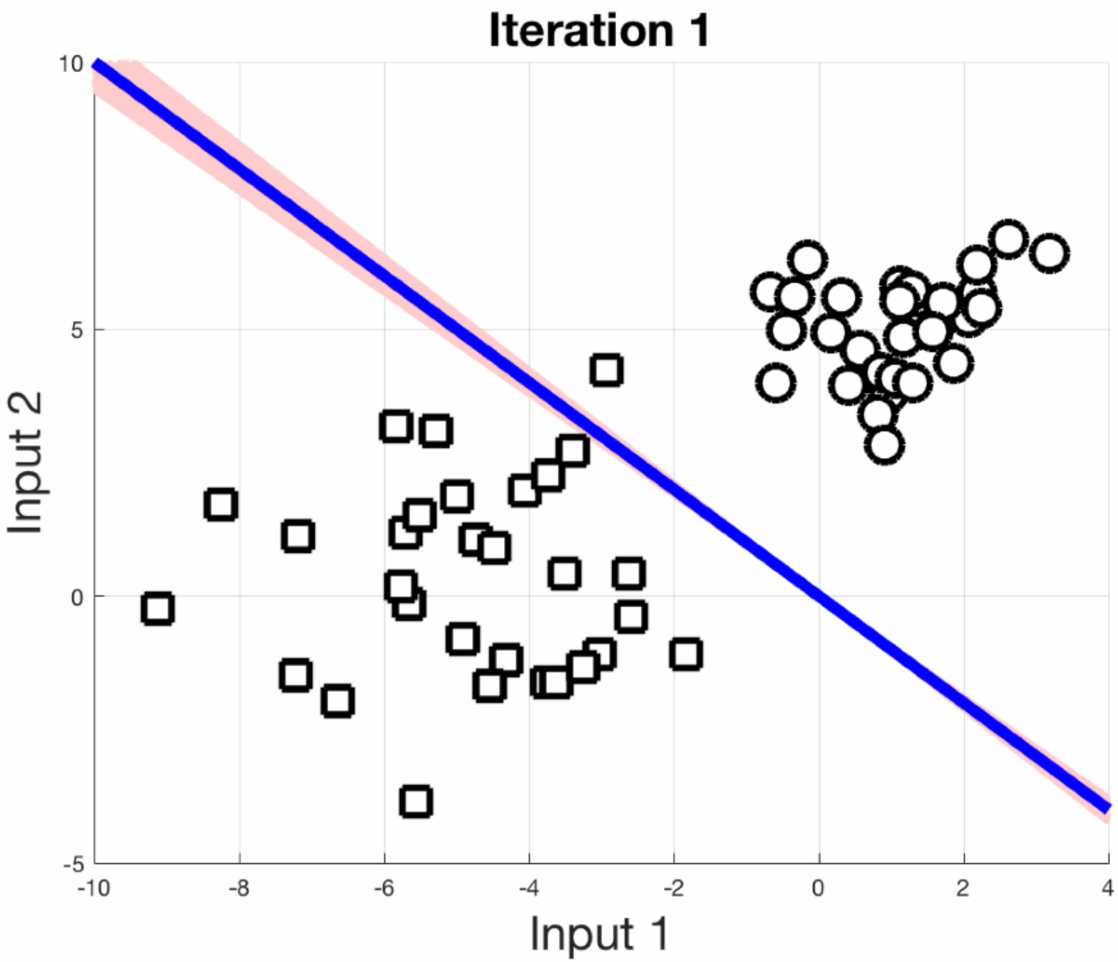
1. Khan and Rue, The Bayesian Learning Rule, arXiv, https://arxiv.org/abs/2107.04562, 2021

NeurIPS 2019 Tutorial

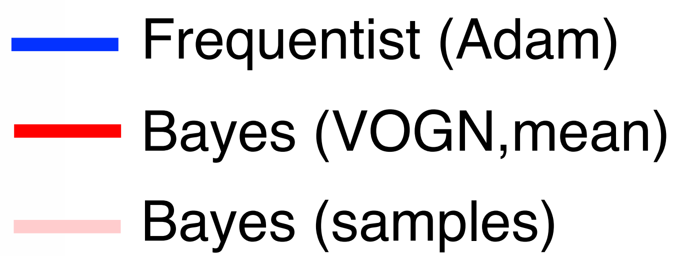




Uncertainty in Logistic Regression



Variational Online Gauss-Newton method



Logistic Regression

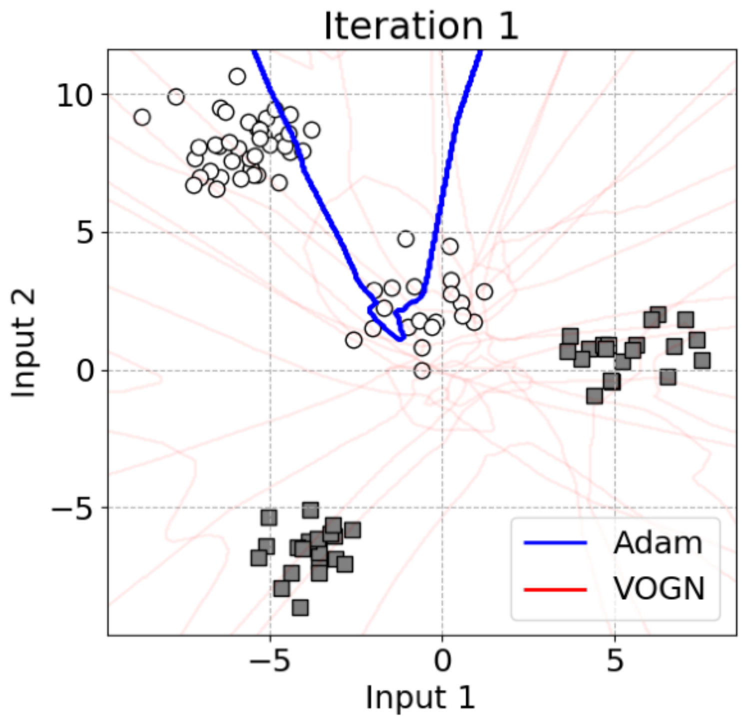
Minibatch = 5,

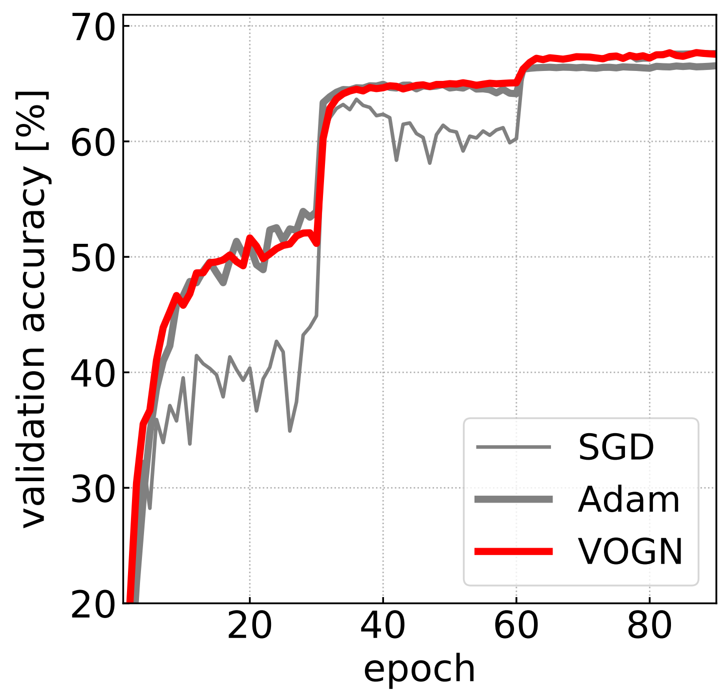
Learning rates = (0.01, 0.01)

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

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

Watch Thomas Moellenhoff's talk at
https://www.youtube.com/watch?v=LQInIN5EU7E.

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).

Practical Deep Learning with Bayes

How to estimate uncertainty with DL optimizers?

RMSprop

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

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

Second-order BAyes (SOBA) [3]

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

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).

Summary

- Why Bayes?
- Lifelong learning with Bayes
  - Use simple estimates of uncertainty
  - Use memory, sensitivity etc.
- A (simple) method to get good uncertainty out of Deep-Learning optimizers