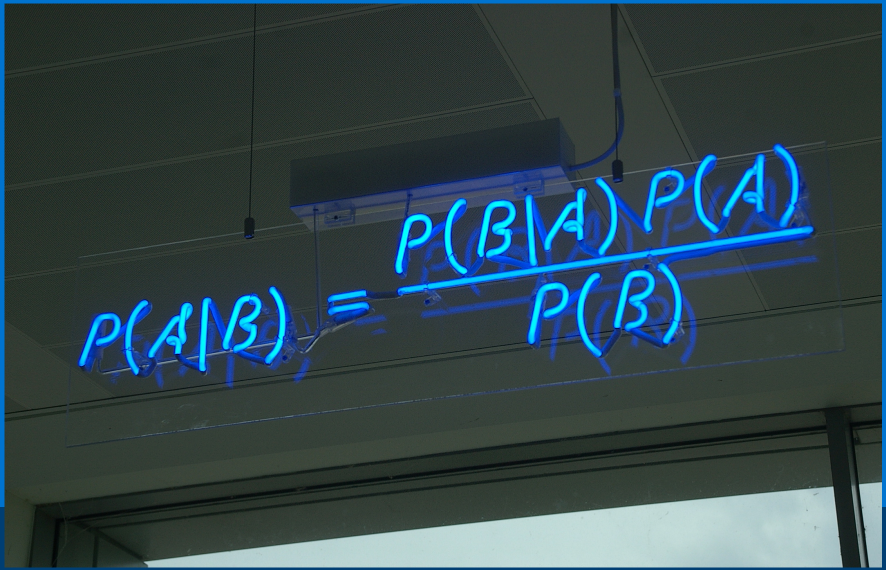
UNIVERSITY OF CAMBRIDGE

Bayesian Methods in Deep Learning

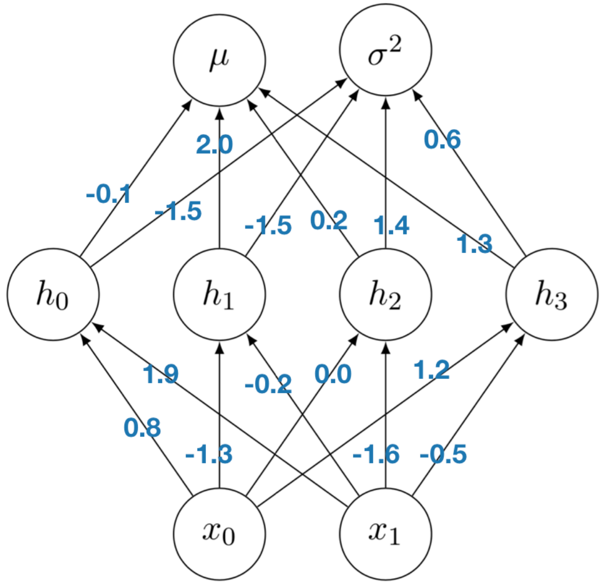
Javier Antorán (ja666@cam.ac.uk)

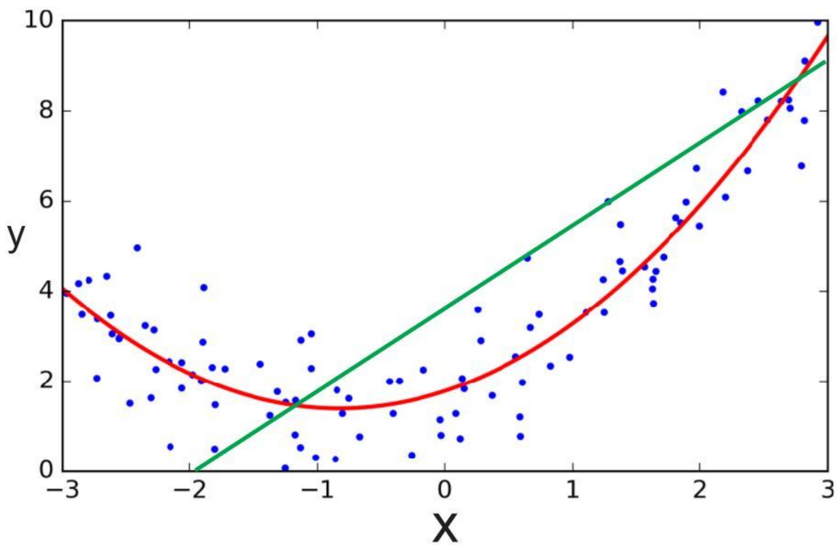


Feel free to interrupt at any time :)

Deep learning

- Overparametrised non-linear models
- ML / MAP inference + Stochastic optimisation
- Needs lots of data



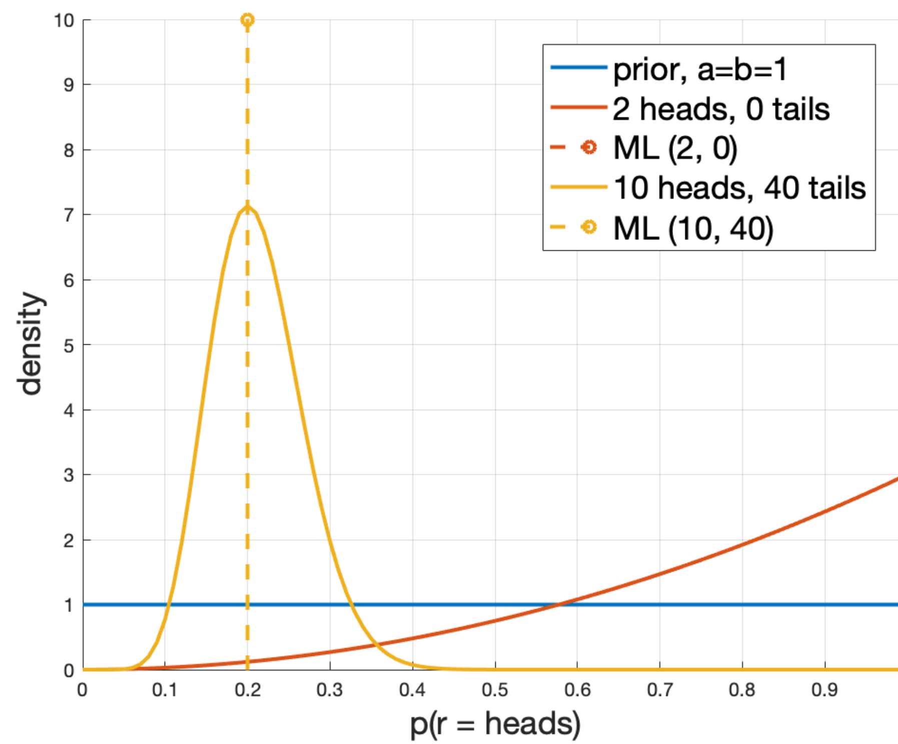


Probabilistic Inference: A biased coin

Likelihood

Prior

$$p(\mathbf{w}|\mathscr{D}) = \frac{p(\mathscr{D}|\mathbf{w})p(\mathbf{w})}{p(\mathscr{D})}$$

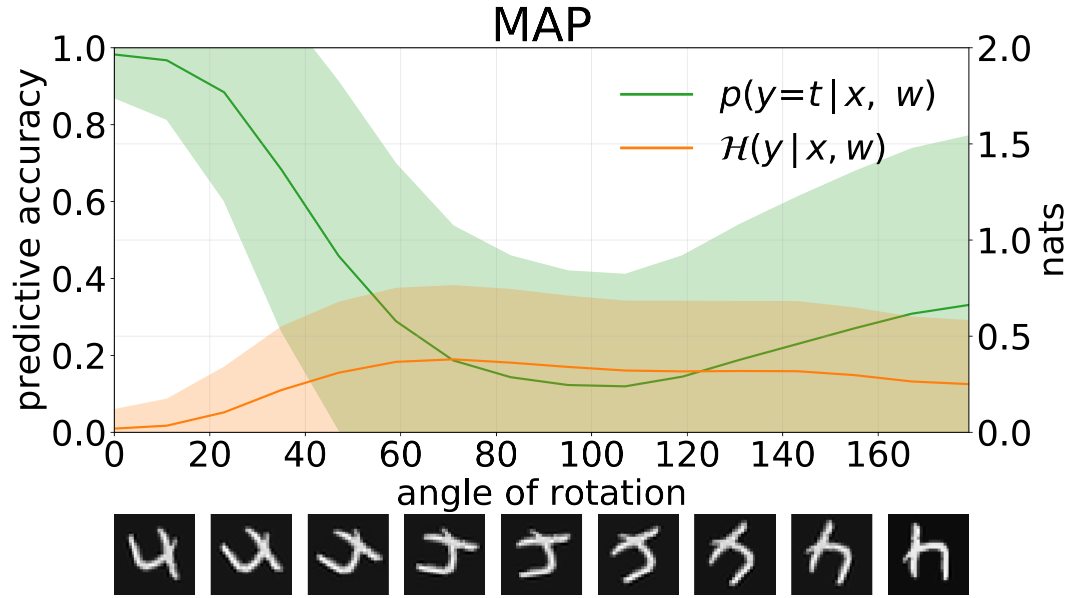


The Problem With MAP / ML

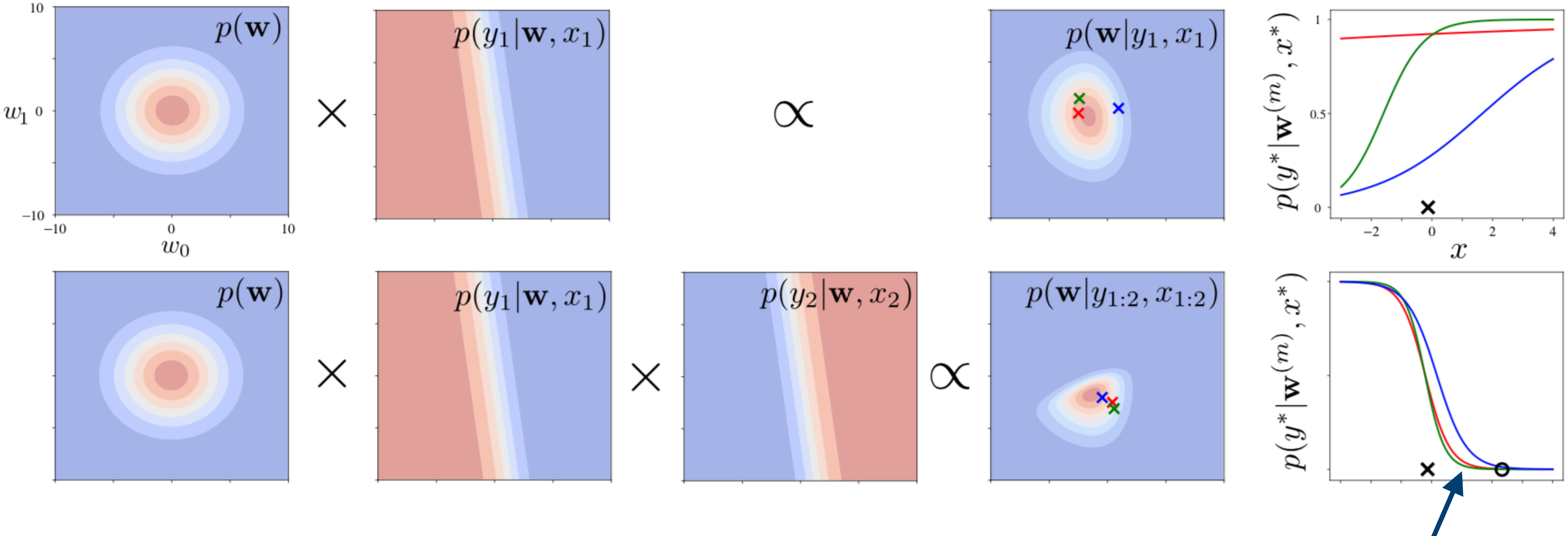
Given a dataset $\mathscr{D} = \{\mathbf{X}, \mathbf{Y}\}$ composed of inputs $\mathbf{X} = \{\mathbf{x}_1 \dots \mathbf{x}_n\}$ and targets $\mathbf{Y} = \{\mathbf{y}_1 \dots \mathbf{y}_n\}$, and a neural network parametrised by $\mathbf{w}$, the maximum likelihood criterion:

$$\mathbf{w}_{ml} = \arg\max_{\mathbf{w}} p(\mathscr{D}|\mathbf{w}) = \arg\max_{\mathbf{w}} \log p(\mathscr{D}|\mathbf{w}) \tag{2.1}$$

- MAP / ML returns the parameter setting that best explains the data.
- It may not be the only good explanation.
- No guarantees about unseen data. These must be introduced through priors.



1-d Probabilistic Logistic Regression



$p(\mathbf{w})$ $\times$ $P(\mathcal{Y}|\mathbf{w}, \mathcal{X})$ $\propto$ $p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$

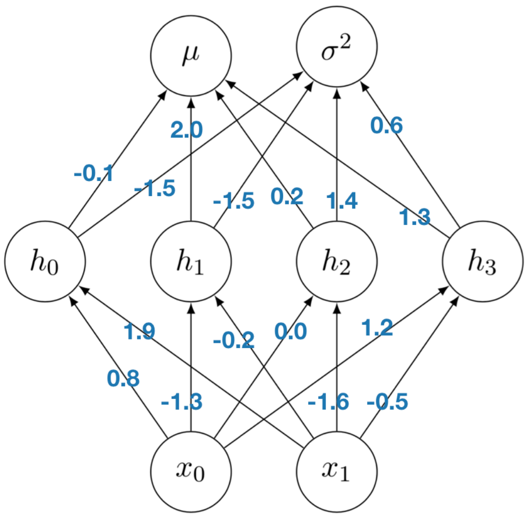
prior
what we knew
before seeing data

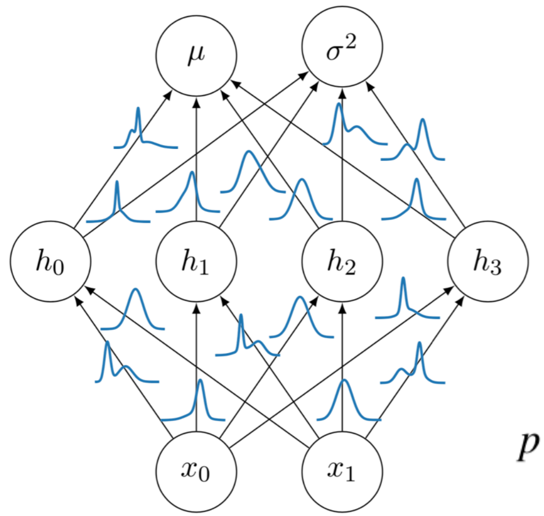
likelihood of $\mathbf{w}$
what the
data told us

posterior
what we know
after seeing data

Variability among samples of **w** is model uncertainty

Bayesian Neural Networks (BNNs)

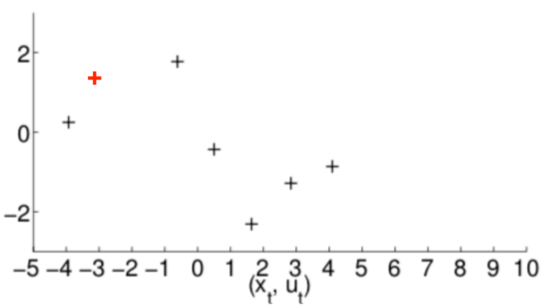


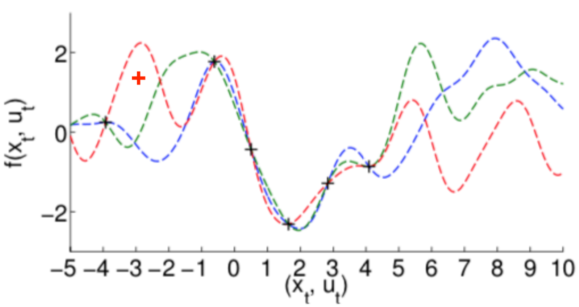


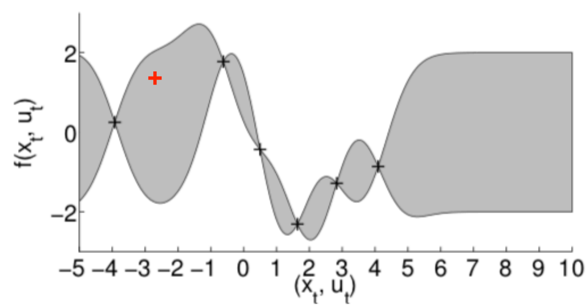
- Predictions incorporate model uncertainty!

$$p(\mathbf{Y}^*|\mathbf{X}^*, \mathscr{D}) = \int p(\mathbf{Y}^*|\mathbf{X}^*, \mathbf{w}) p(\mathbf{w}|\mathscr{D})\, d\mathbf{w}$$

$\mu(x_t)$
$\sigma = 0$

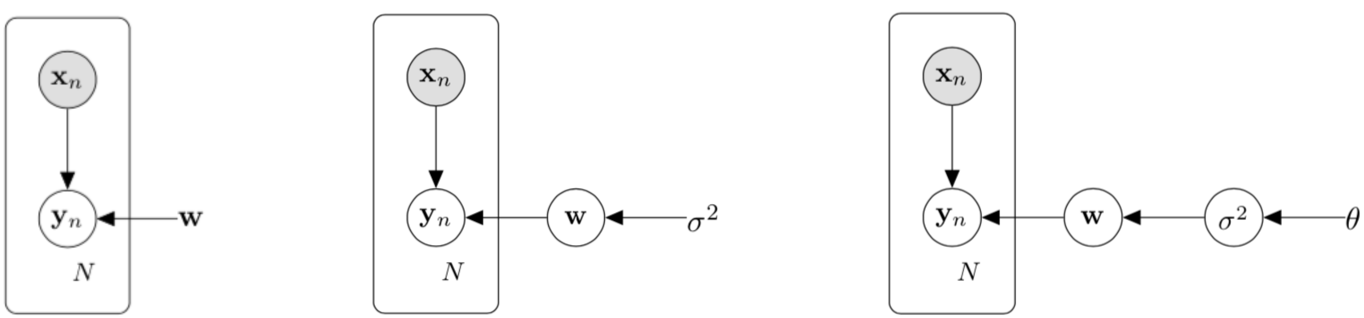






Practical Forms of Prior

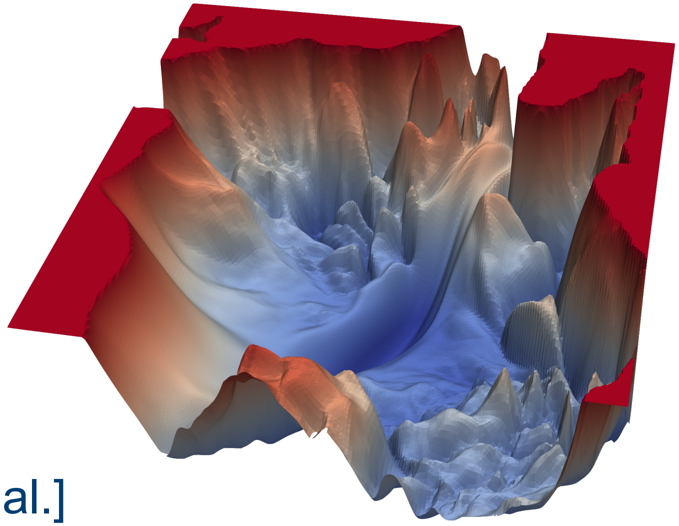
- ML / MAP **a)**
- Symmetric tractable prior **b)**
  - Gaussian, Laplace, Student-T, etc
  - Type-2 Maximum Likelihood
  - Previous posterior (Bayesian update)
- Conjugate hyper-prior + Gibbs sampling (hierarchical model) **c)**



a) **b)** **c)**

Inference with BNNs

- The data's likelihood under a BNN model is a very complex, high dimensional and multimodal function.



[Li et al.]

- **Intractable evidence:**

$$p(\mathscr{D}) = \int p(\mathscr{D}|\mathbf{w})p(\mathbf{w})\,d\mathbf{w}$$

- **Intractable predictive:**

$$p(\mathbf{Y}^*|\mathbf{X}^*,\mathscr{D}) = \int p(\mathbf{Y}^*|\mathbf{X}^*,\mathbf{w})p(\mathbf{w}|\mathscr{D})\,d\mathbf{w}$$

- Must resort to **Approximate Inference**

Variational Inference

Calculus of variations is like finding optima of a function but instead of finding a value of (x, y) which yields the optima, you find a function which yields the optima.

Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w}$

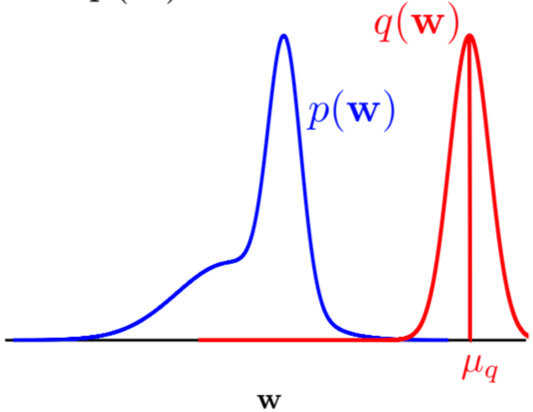
1. non-negative

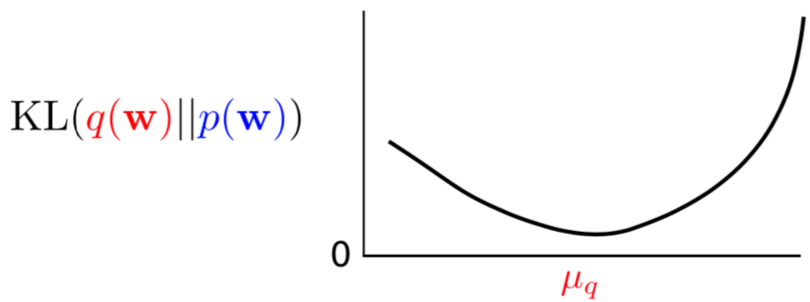
$$\frac{\delta^2}{\delta q^2}\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \frac{1}{q(\mathbf{w})} \geq 0$$

2. zero (minimised) when $q(\mathbf{w}) = p(\mathbf{w})$

$$\mathrm{KL}(p(\mathbf{w})||p(\mathbf{w})) = \int p(\mathbf{w}) \log \frac{p(\mathbf{w})}{p(\mathbf{w})} \mathrm{d}\mathbf{w} = 0$$

First grad = 0 only if q=p.
Second grad -> strictly convex

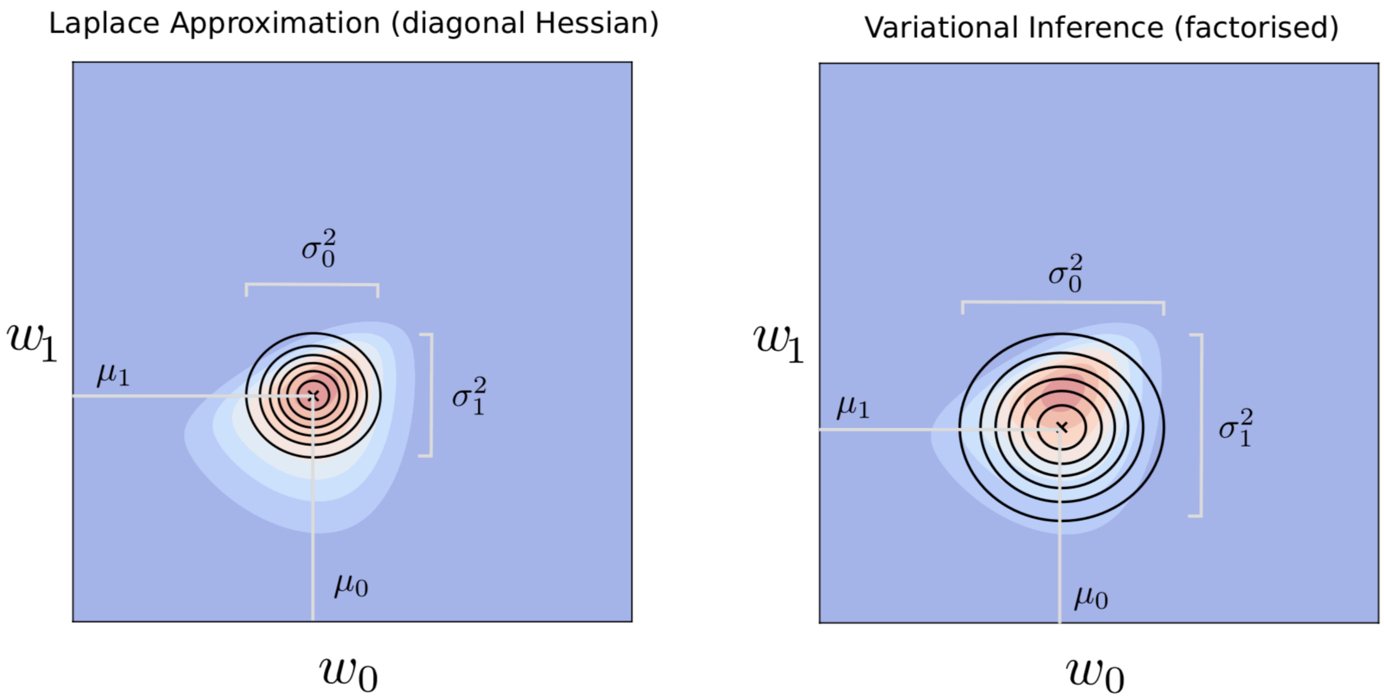




3. can be computed up to an additive constant w/o needing normalisation for $p(\mathbf{w})$

Issues with VI

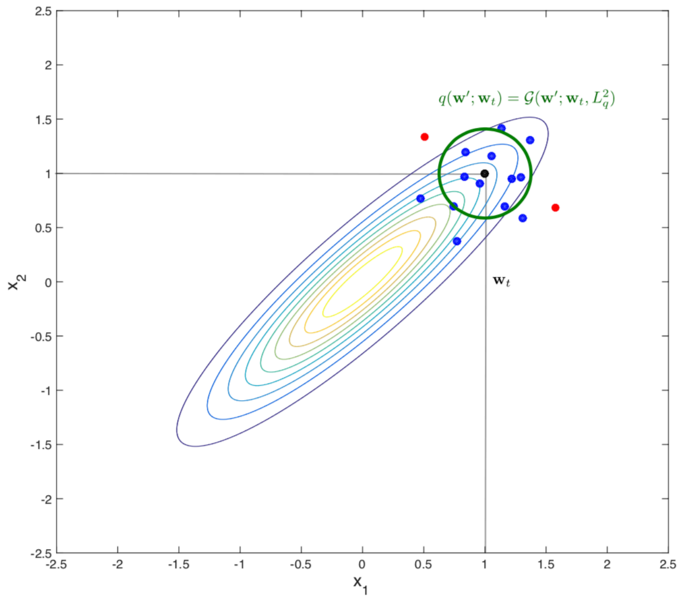
- Mode seeking behaviour
- Overly simple approximate posteriors (complexity vs tractability)



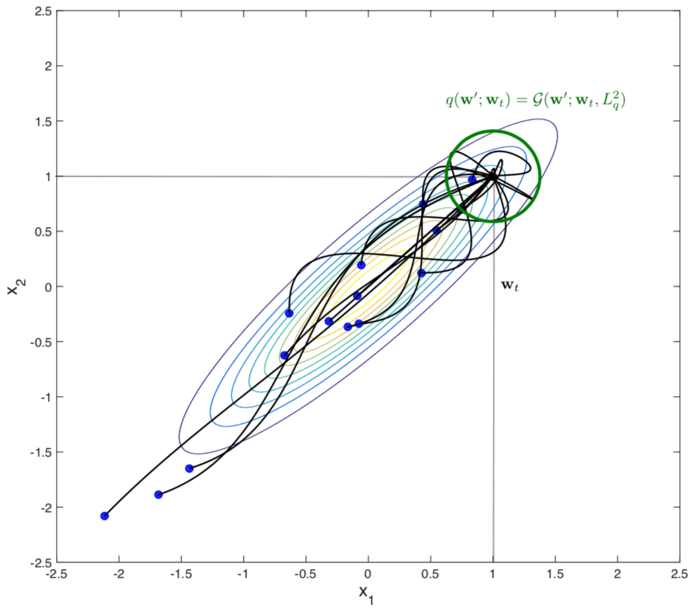
Markov Chain Monte Carlo

$$p(\mathbf{Y}^*|\mathbf{X}^*,\mathscr{D}) \approx \sum_{k=1}^{K} p(\mathbf{Y}^*|\mathbf{X}^*,\mathbf{w}_k); \quad \mathbf{w}_k \sim p(\mathbf{w}|\mathscr{D})$$

- Robust uncertainty requires sample diversity: HarleMCMC shake

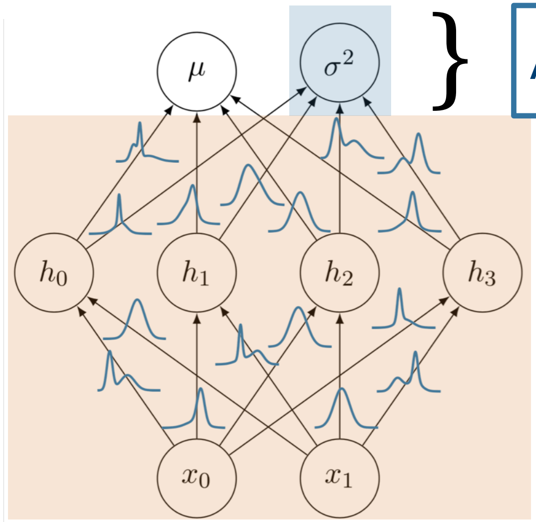


Metropolis



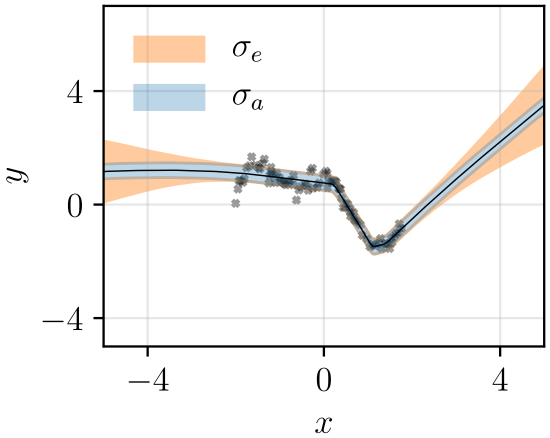
Hybrid Monte Carlo

Uncertainty Decomposition

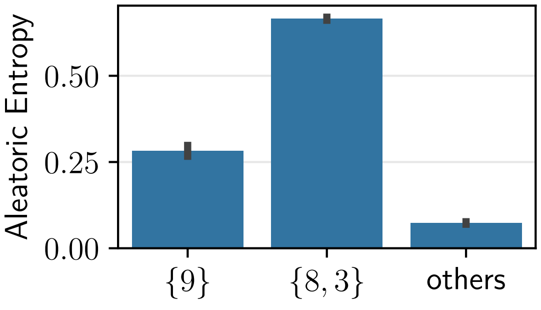


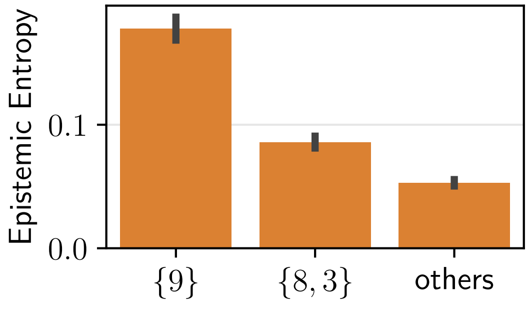
Aleatoric Uncertainty

Model Uncertainty



MNIST; Remove 80% of {9} and permute 30% {8,3}

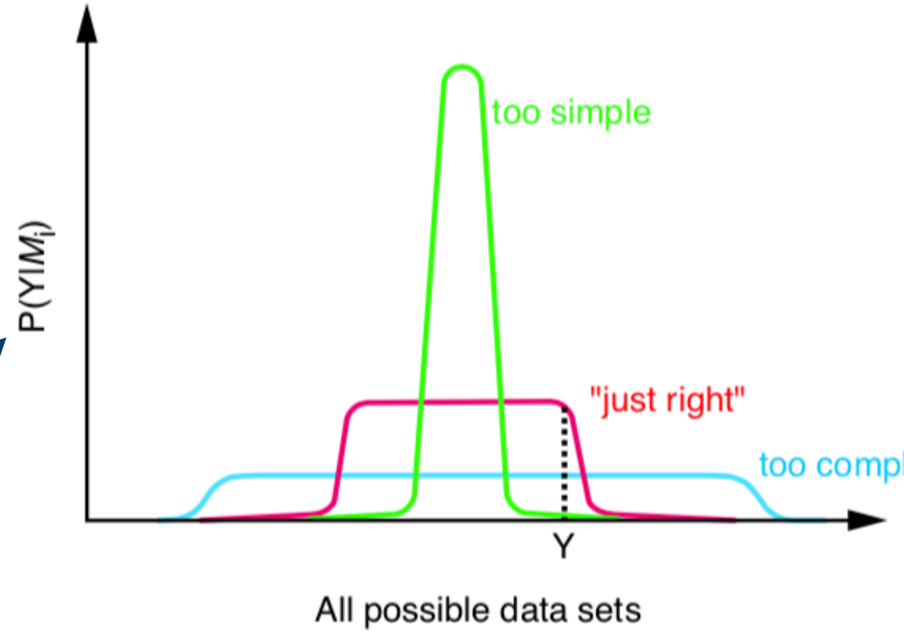




Protection Against Overfitting

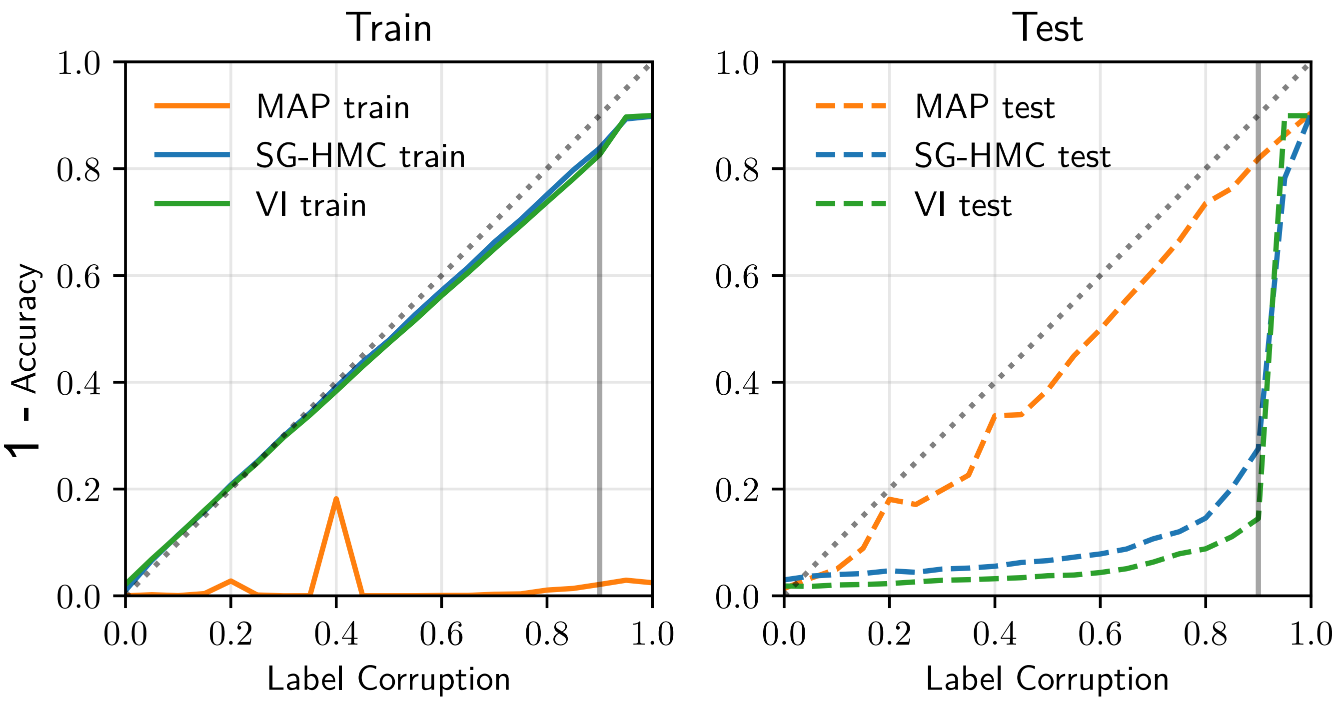
- Model uncertainty is transformed into predictive uncertainty
- With uninformative prior, automatic tradeoff between goodness of fit and model complexity: Automatic Ockham's Razor. (Related to MDL)

Marginal Likelihood



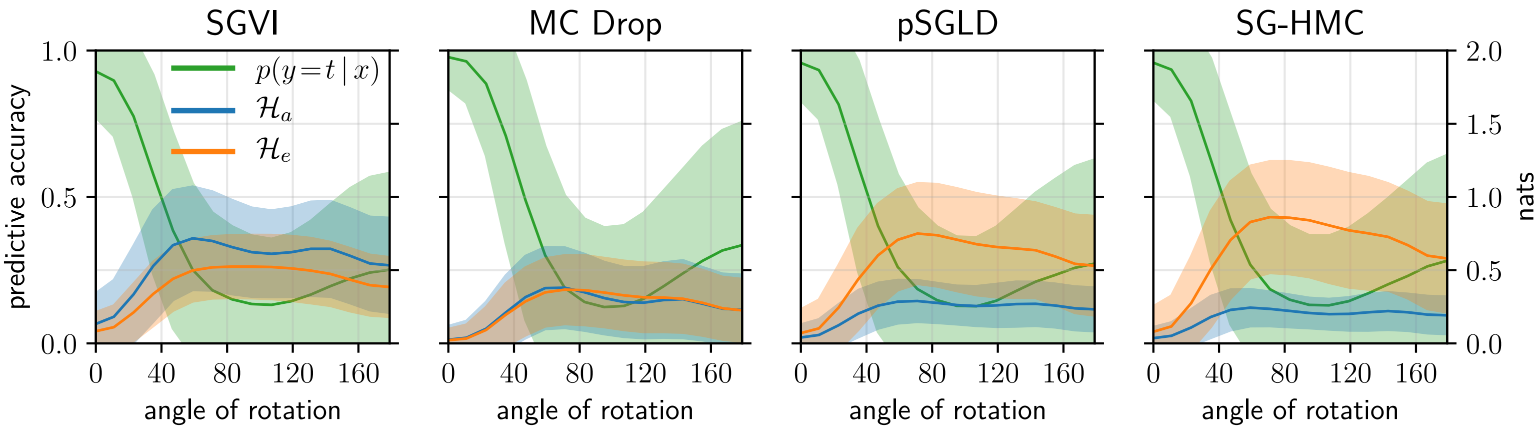
Protection Against Overfitting (cont)

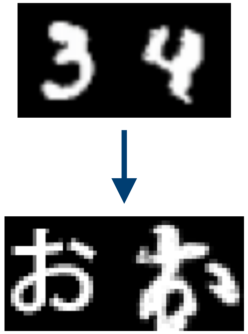
- Plot of train error and test error when training different models with increasing number of permuted samples

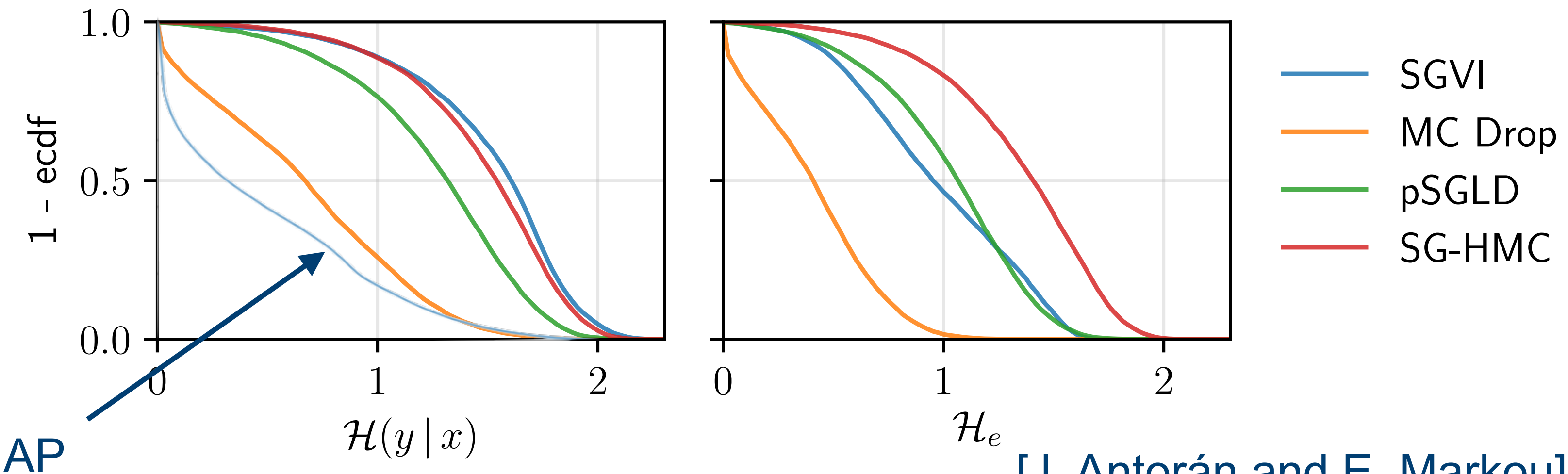


[Unpublished results]

Out of Distribution Sample Detection

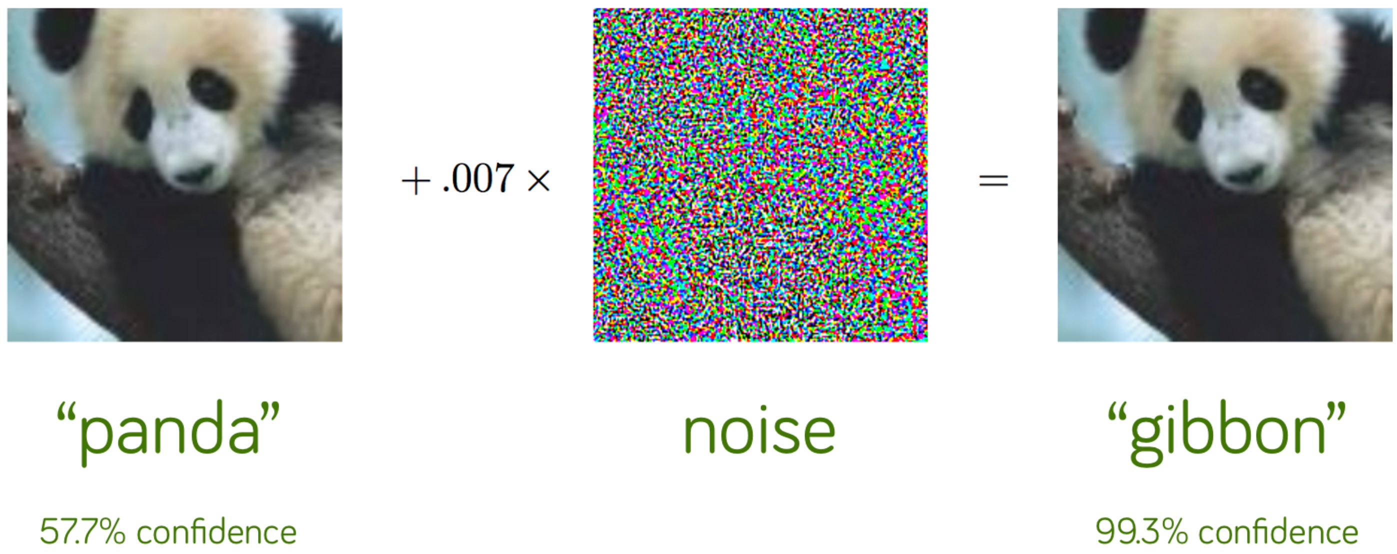




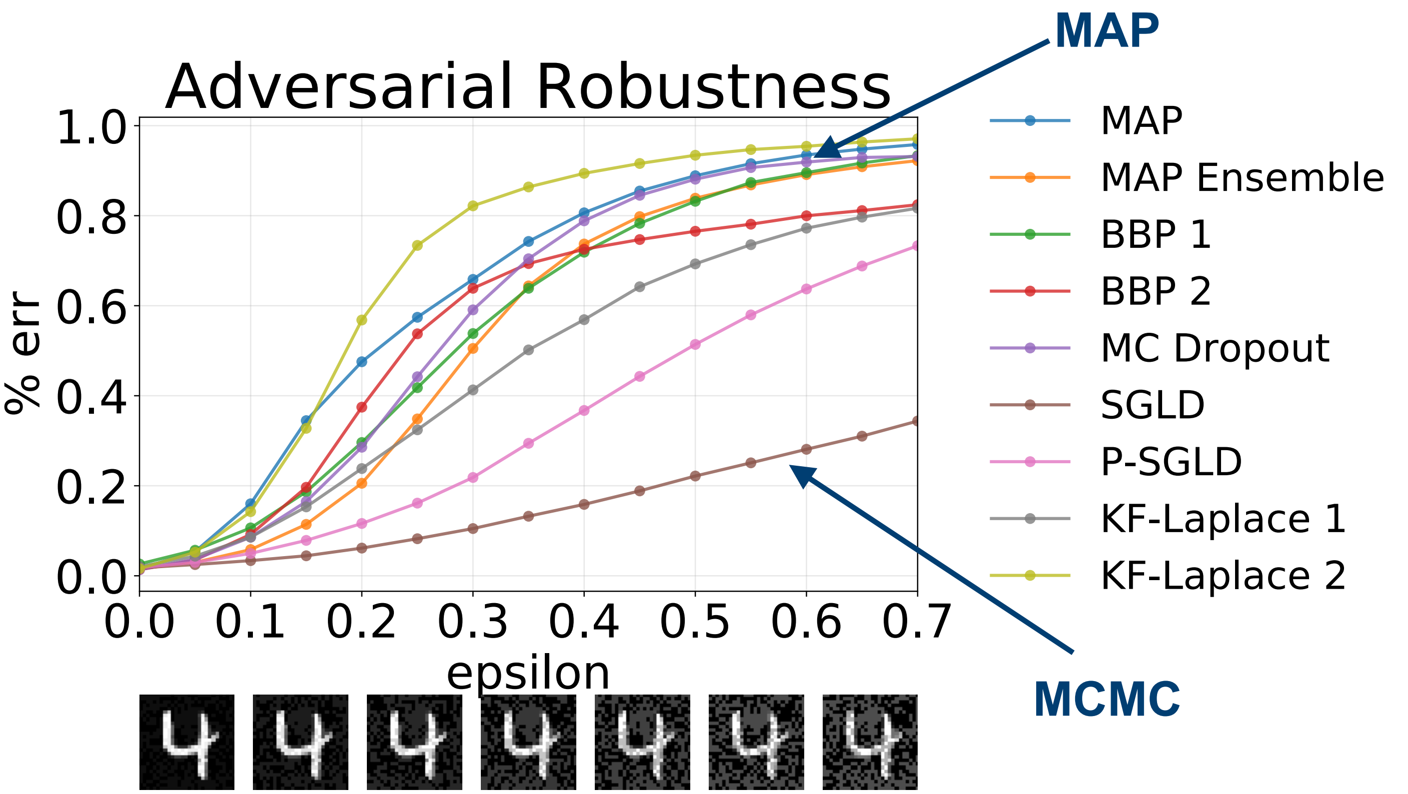


[J. Antorán and E. Markou]

Adversarial Robustness

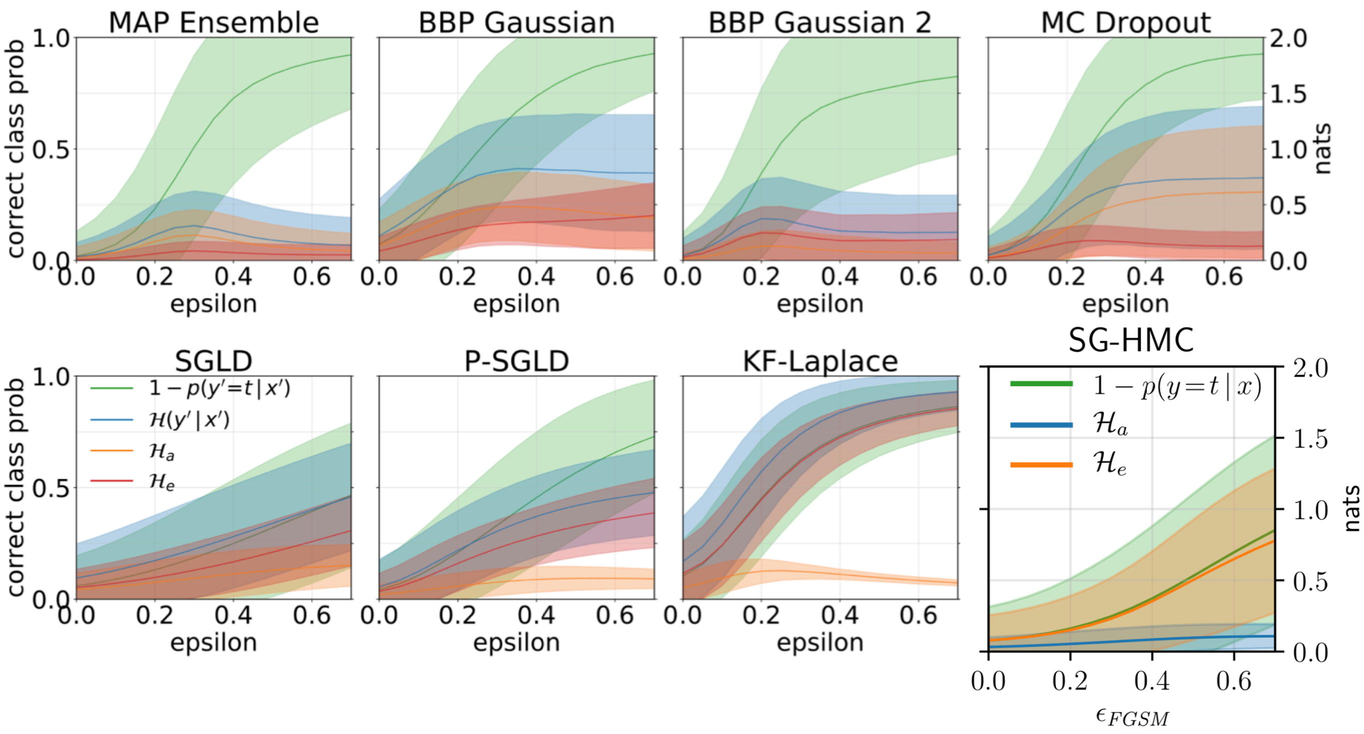


Adversarial Robustness



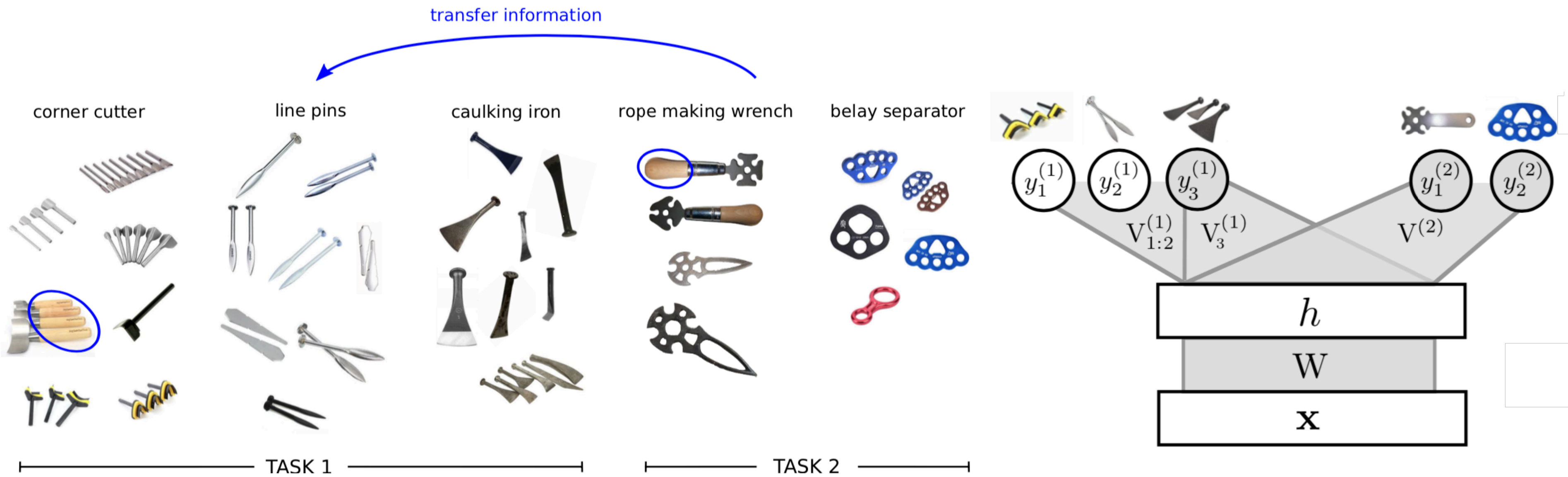
[J. Antorán and E. Markou]

Adversarial Robustness

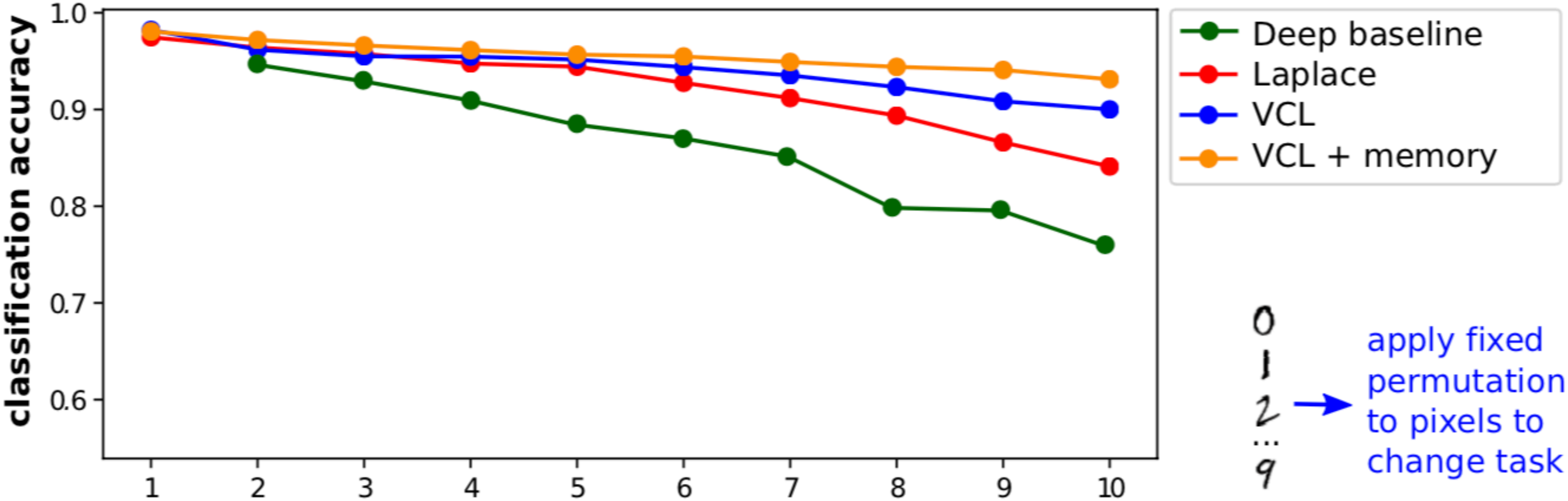


[J. Antorán and E. Markou]

Continual Learning



Continual Learning

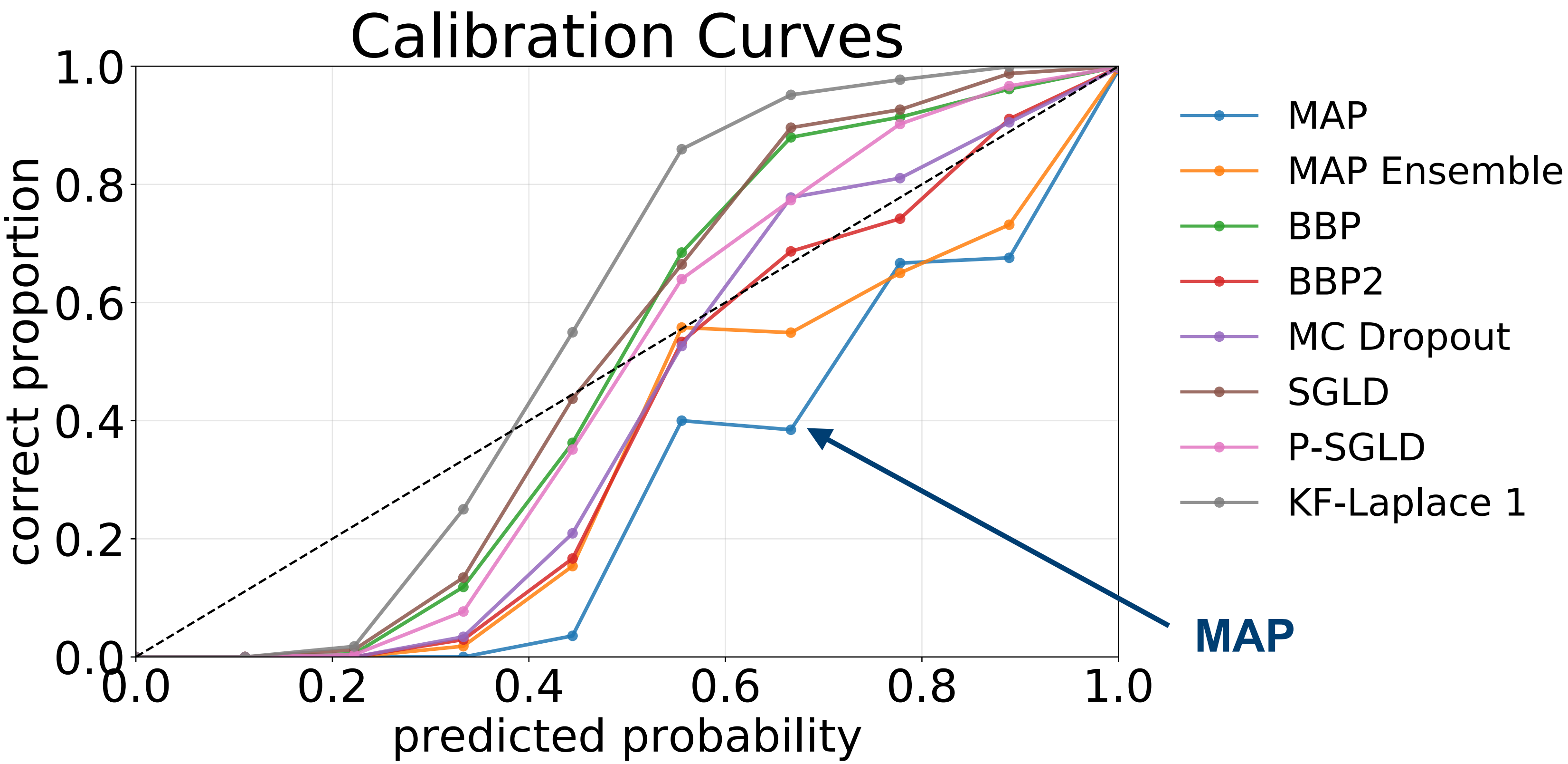


apply fixed permutation to pixels to change task

$$p(\mathbf{w}|\mathcal{D}_1,\mathcal{D}_2) = \frac{p(\mathcal{D}_2|\mathbf{w})q(\mathbf{w}|\mathcal{D}_1)}{p(\mathcal{D}_2)}$$

[C. Nguyen et al.]

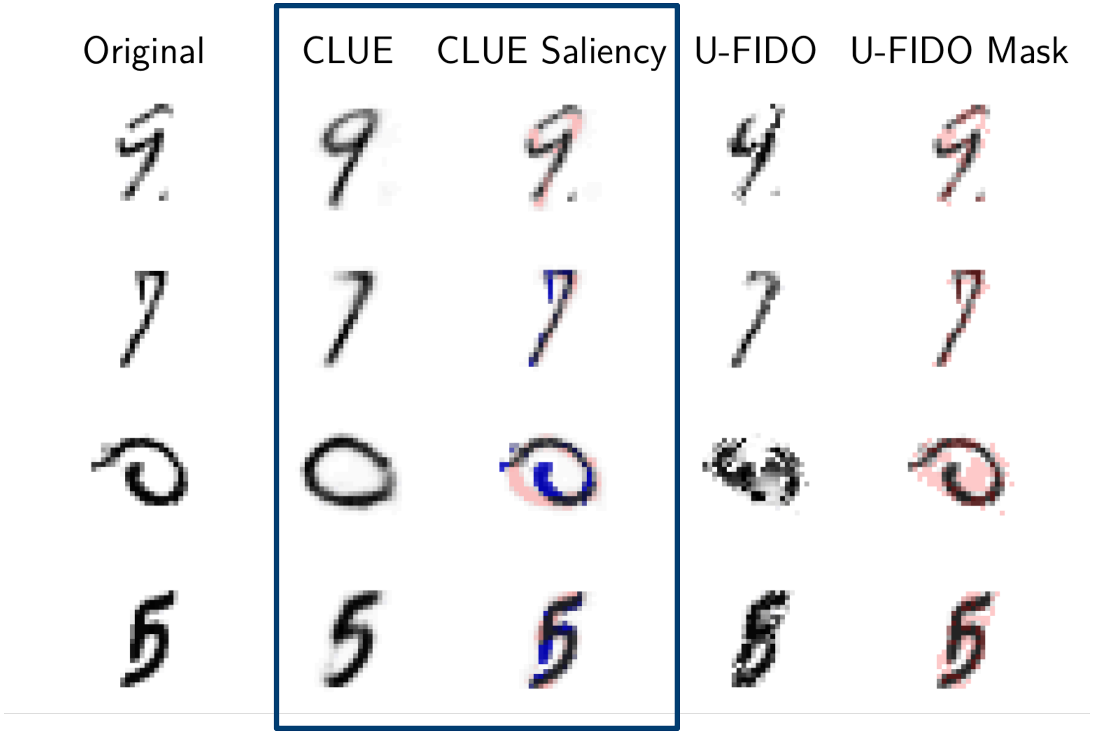
Calibration



[J. Antorán and E. Markou]

Uncertainty Interpretability

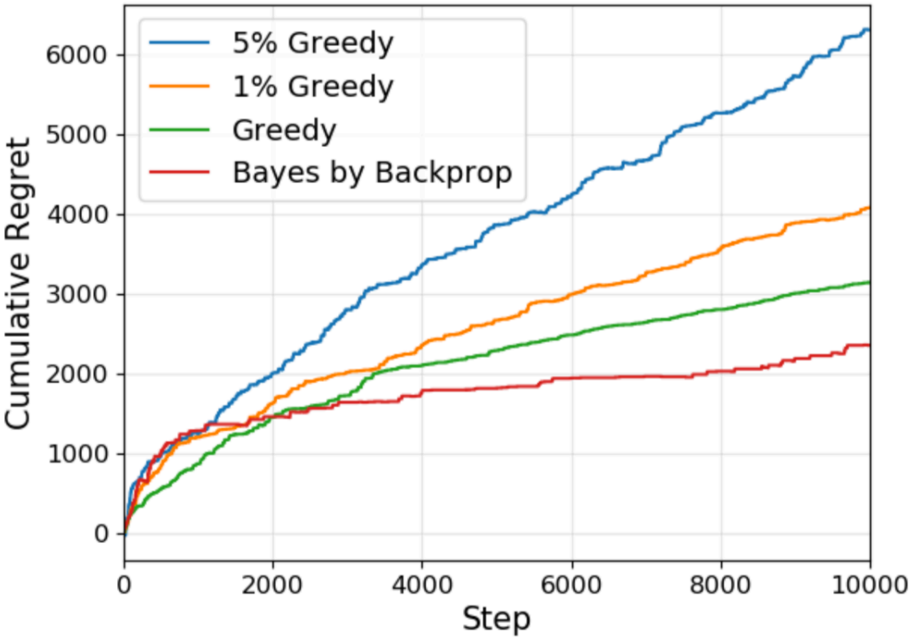
- Why is our model uncertain?



[J. Antorán]

Balancing Exploration and Exploitation: Bandits

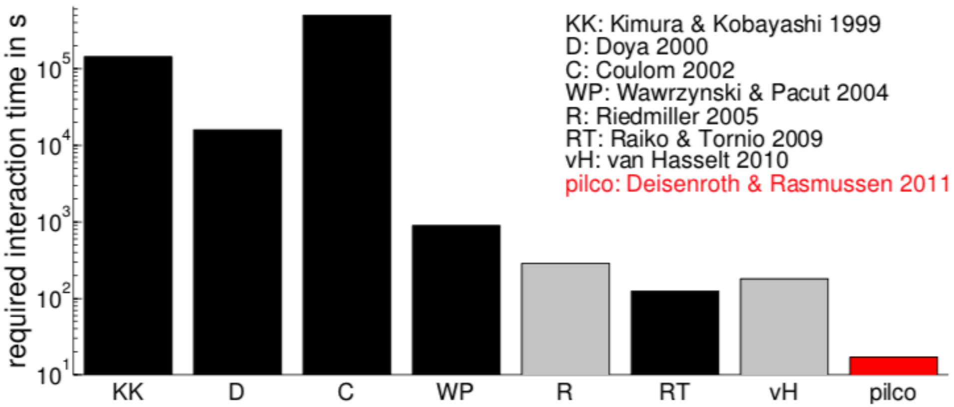
- Different mushrooms give different rewards with different probs.



- + RL / Active Learning / Bayesian Optimisation

Sample-Efficiency in Reinforcement Learning

- Probabilistic models help reduce model bias is RL
- PILCO uses a GP model and a free-form (linear, RBF, NN) controller.



PILCO learner

PILCO learner CS

[M. P. Deisenroth and C. E. Rasmussen]

Open Source BNNs

github.com/JavierAntoran/Bayesian-Neural-Networks

Bayesian-Neural-Networks

Pytorch implementations of Bayes By Backprop, MC Dropout, SGLD, the Local Reparametrization Trick, KF-Laplace and more

Jupyter Notebook ★ 140 19

Bayesian Neural Networks

License MIT | python 2.7+ | pytorch 1.0.1

Pytorch implementations for the following approximate inference methods:

- Bayes by Backprop
- Bayes by Backprop + Local Reparametrisation Trick
- MC dropout
- Stochastic Gradient Langevin Dynamics
- Preconditioned SGLD
- Kronecker-Factorised Laplace Approximation
- Stochastic Gradient Hamiltonian Monte Carlo (Coming soon)

We also provide code for:

- Bootstrap MAP Ensemble

References

- Some slides taken from Richard E. Turner

Uncertainty in Bayesian Neural Networks
J. Antorán and E. Markou.
Presented at: Workshop on The Mathematics of Deep Learning and Data Science, The Isaac Newton Institute for Mathematical Sciences, Cambridge. 2019.

Excellent textbook introductions to Bayesian machine learning:

D. J. C. MacKay Information Theory, Inference and Learning Algorithms, 2003

C. Bishop Pattern Recognition and Machine Learning, 2006

K. Murphy Machine Learning: A Probabilistic Perspective, 2012

Beautiful seminal work on Bayesian neural networks:

R. M. Neal Bayesian Learning for Neural Networks, Lecture Notes in Statistics No. 118. New York: Springer-Verlag. 1996

More References (VI and Laplace)

D. J. C. MacKay A practical Bayesian framework for backpropagation networks, Neural Computation 4(3) 448-472, 1992

D. J. C. MacKay Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks , Network: Computation in Neural Systems 6(3), 469-505, 1995

H. Ritter et al. A Scalable Laplace Approximation for Neural Networks, ICLR, 2018

G. Hinton and D. Van Camp Keeping the neural networks simple by minimizing the description length of the weights, COLT 5-13, ACM, 1993

D. Barber and C. Bishop Ensemble Learning in Bayesian Neural Networks, Neural Networks and Machine Learning 1998

A. Graves Practical Variational Inference for Neural Networks, NIPS 2011

C. Blundell et al. Weight Uncertainty in Neural Networks, ICML, 2015

More References (HMC)

Bayesian Learning via Stochastic Gradient Langevin Dynamics, Welling and Teh, ICML, 2011

Stochastic Gradient Hamiltonian Monte Carlo, Chen, Fox, Guestrin, NIPS, 2015