Deep Learning: Classics and Trends

Imperial College London

Learning Invariances
using the **Marginal Likelihood**

Mark van der Wilk

Department of Computing
Imperial College London

@markvanderwilk
`m.vdwilk@imperial.ac.uk`

June 8, 2020

This presentation contains animations, which unfortunately do not work with all pdf viewers. Use e.g. Adobe Acrobat Reader.

Overview

- How to find the appropriate **inductive bias**?
  - E.g. type of layers, filter size, data augmentation...

Overview

- How to find the appropriate **inductive bias**?
  - E.g. type of layers, filter size, data augmentation...
- Based on our NeurIPS 2018 paper

Learning Invariances using the Marginal Likelihood

Mark van der Wilk
PROWLER.io
Cambridge, UK
mark@prowler.io

Matthias Bauer
MPI for Intelligent Systems
University of Cambridge
msb55@cam.ac.uk

ST John
PROWLER.io
Cambridge, UK
st@prowler.io

James Hensman
PROWLER.io
Cambridge, UK
james@prowler.io

Overview

- How to find the appropriate **inductive bias**?
  - E.g. type of layers, filter size, data augmentation...
- Based on our NeurIPS 2018 paper

Learning Invariances using the Marginal Likelihood

Mark van der Wilk
PROWLER.io
Cambridge, UK
mark@prowler.io

Matthias Bauer
MPI for Intelligent Systems
University of Cambridge
msb55@cam.ac.uk

ST John
PROWLER.io
Cambridge, UK
st@prowler.io

James Hensman
PROWLER.io
Cambridge, UK
james@prowler.io

Deep Learning: Classics and Trends

Overview

- How to find the appropriate **inductive bias**?
  - E.g. type of layers, filter size, data augmentation...
- Based on our NeurIPS 2018 paper

Learning Invariances using the Marginal Likelihood

Mark van der Wilk	Matthias Bauer	ST John	James Hensman
PROWLER.io	MPI for Intelligent Systems	PROWLER.io	PROWLER.io
Cambridge, UK	University of Cambridge	Cambridge, UK	Cambridge, UK
mark@prowler.io	msb55@cam.ac.uk	st@prowler.io	james@prowler.io

Deep Learning: Classics and Trends

✓ Classic: Bayesian model selection

Overview

- How to find the appropriate **inductive bias**?
  - E.g. type of layers, filter size, data augmentation...
- Based on our NeurIPS 2018 paper

Learning Invariances using the Marginal Likelihood

Mark van der Wilk
PROWLER.io
Cambridge, UK
mark@prowler.io

Matthias Bauer
MPI for Intelligent Systems
University of Cambridge
msb55@cam.ac.uk

ST John
PROWLER.io
Cambridge, UK
st@prowler.io

James Hensman
PROWLER.io
Cambridge, UK
james@prowler.io

Deep Learning: Classics and Trends

✓ Classic: Bayesian model selection

✓ Trend: Invariances

Overview

- How to find the appropriate **inductive bias**?
  - E.g. type of layers, filter size, data augmentation...
- Based on our NeurIPS 2018 paper

Learning Invariances using the Marginal Likelihood

Mark van der Wilk
PROWLER.io
Cambridge, UK
mark@prowler.io

Matthias Bauer
MPI for Intelligent Systems
University of Cambridge
msb55@cam.ac.uk

ST John
PROWLER.io
Cambridge, UK
st@prowler.io

James Hensman
PROWLER.io
Cambridge, UK
james@prowler.io

Deep Learning: Classics and Trends

✓ Classic: Bayesian model selection

✓ Trend: Invariances

? Deep

Overview

- How to find the appropriate **inductive bias**?
  - E.g. type of layers, filter size, data augmentation...
- Based on our NeurIPS 2018 paper

Learning Invariances using the Marginal Likelihood

Mark van der Wilk, PROWLER.io, Cambridge, UK, mark@prowler.io

Matthias Bauer, MPI for Intelligent Systems, University of Cambridge, msb55@cam.ac.uk

ST John, PROWLER.io, Cambridge, UK, st@prowler.io

James Hensman, PROWLER.io, Cambridge, UK, james@prowler.io

~~Deep~~ Learning: Classics and Trends

✓ Classic: Bayesian model selection

✓ Trend: Invariances

× Deep: Our method actually uses a Gaussian process (shallow)

Overview

- How to find the appropriate **inductive bias**?
  - E.g. type of layers, filter size, data augmentation...
- Based on our NeurIPS 2018 paper

Learning Invariances using the Marginal Likelihood

Mark van der Wilk
PROWLER.io
Cambridge, UK
mark@prowler.io

Matthias Bauer
MPI for Intelligent Systems
University of Cambridge
msb55@cam.ac.uk

ST John
PROWLER.io
Cambridge, UK
st@prowler.io

James Hensman
PROWLER.io
Cambridge, UK
james@prowler.io

~~Deep~~ Learning: Classics and Trends

✓ Classic: Bayesian model selection

✓ Trend: Invariances

✓ Deep: Our method actually uses a Gaussian process (shallow)
General principles: We will discuss implications on deep learning

Overview

Learning inductive biases

Bayesian model selection

Invariances & model selection

Conclusions & Implications

Bonus slides

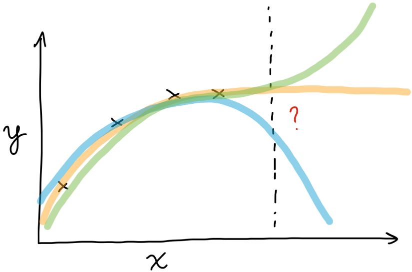
Supervised learning

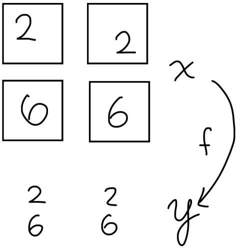
We observe training examples from some relationship $f^*(\cdot)$:

$$f^*(\mathbf{x}_1) = y_1\,, \quad f^*(\mathbf{x}_2) = y_2\,, \quad f^*(\mathbf{x}_3) = y_3\,, \quad \ldots \quad , f^*(\mathbf{x}_n) = y_n\,.$$

The goal is to make good predictions at **new** points

$$f^*(\mathbf{x}_{n+1}) = ?$$

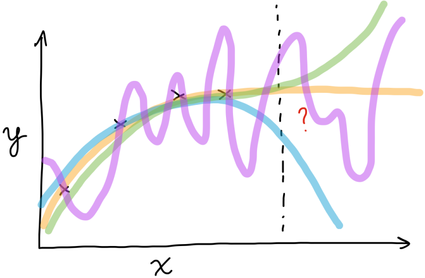




Inductive bias

Which prediction to choose?

- Many predictions fit the training data
- Which one you choose is determined by your **inductive bias**
- Inductive bias is a **constraint** on plausible predictions
  You choose a prediction because you rule out others



We want our inductive bias to **generalise**, i.e.

$$f(\mathbf{x}_{n+1}) \approx f^*(\mathbf{x}_{n+1})$$

Inductive biases

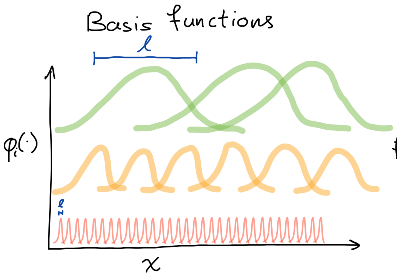
Inductive bias is specified by

- Regularisation
- Non-linearity
- Neural network architecture
- Data augmentation
- ...

A simple problem

Single layer neural network:

$$f(x) = \sum_{i=1}^{B} \phi_i(\mathbf{x}; \ell) w_i\,, \quad y_n = f(\mathbf{x}_n) + \epsilon\,, \qquad \epsilon_n \sim \mathcal{N}(0, \sigma^2)$$



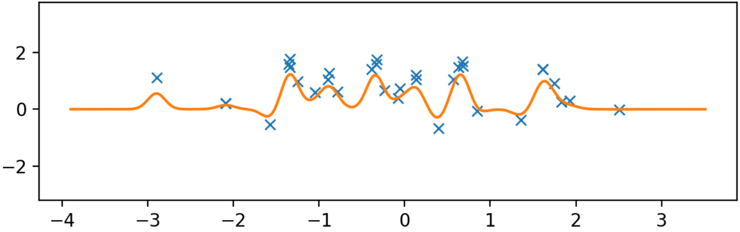
Hyperparameters ℓ, σ determine inductive bias ("wigglyness", deviation from observations)

How to set these parameters?

Minimising training loss — A strawman method

We use the training loss to learn the weights? Why not the hyperparameters too?

$$\mathcal{L}(\mathbf{w}, \ell, \sigma) = -N \log 2\pi\sigma^2 - \left[\sum_{n=1}^{N} \frac{1}{2\sigma^2} (\boldsymbol{\phi}(\mathbf{x}_n; \ell)^\mathsf{T} \mathbf{w} - y_n)^2 \right] - \|\mathbf{w}\|_{\mathcal{H}_\ell}$$



- Training loss prefers **least constraints** to fitting the training data
- Inductive bias is **always a constraint**

Cross-validation

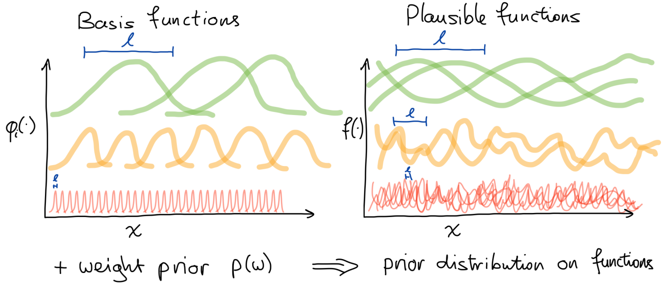
Goal: Find inductive bias which **generalises**
Standard procedure: Cross-validation

- Find weights given regularisation on training set
- Estimate generalisation error on a validation set
- Try multiple regularisation parameters
- Pick the one with the best validation loss

Disadvantages

- Need to train a whole model for each hyperparameter setting
- Can't use gradients
- Need a separate validation set from training set

Inductive biases and priors



- Express inductive bias as distribution on functions $p(f(\cdot)|\theta)$
- Make predictions with **posterior**

$$p(f(\cdot) \mid X, \mathbf{y}, \theta) = \frac{\prod_{n=1}^{N} p(y_n \mid f(\mathbf{x}_n))p(f(\cdot)|\theta)}{p(\mathbf{y}|X, \theta)}$$

Bayes for hyperparameters

To find the inductive bias (hyperparameters θ), just apply Bayes rule!

$$p(f, \theta \mid \mathbf{y}, X) = \frac{p(\mathbf{y}, f, \theta \mid X)}{p(\mathbf{y}|X)} = \frac{p(\mathbf{y} \mid f, X, \theta)p(f|\theta)p(\theta)}{p(\mathbf{y}|X)} \tag{1}$$

$$= \underbrace{\frac{p(\mathbf{y} \mid f, X, \theta)p(f|\theta)}{p(\mathbf{y} \mid X, \theta)}}_{p(f|\mathbf{y},X)} \underbrace{\frac{p(\mathbf{y} \mid X, \theta)p(\theta)}{p(\mathbf{y}|X)}}_{p(\theta \mid \mathbf{y},X)} \tag{2}$$

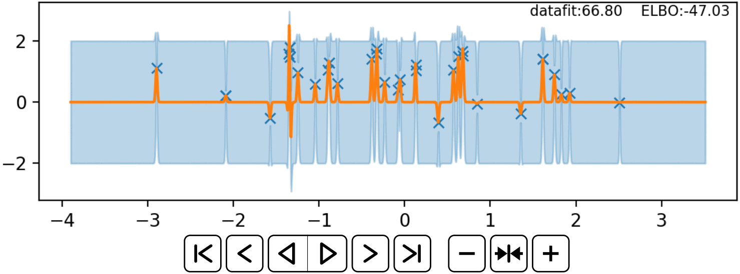
Posterior over f and θ consists of two parts

1. The original posterior over f,
2. A posterior over θ using the **marginal likelihood**:

$$p(\mathbf{y}|X, \theta) = \int p(\mathbf{y}|f, X, \theta)p(f|\theta)\mathrm{d}\theta \tag{3}$$

Model selection procedure

1. Compute marginal likelihood (this is often difficult)
2. Choose model with maximum marginal likelihood (simplification) $\theta^* = \text{argmax}_\theta \log p(\mathbf{y} \mid X, \theta)$
3. Predict with posterior $p(f \mid \mathbf{y}, X, \theta^*) = \frac{\prod_{n=1}^{N} p(y_n \mid f(\mathbf{x}_n)) p(f(\cdot)|\theta^*)}{p(\mathbf{y}|X,\theta^*)}$



- More sensible fit as the marginal likelihood rises
- Datafit gets worse!

Marginal likelihood as incremental prediction

We can split the marginal likelihood up using the **product rule**:

$$p(\mathbf{y} \mid \theta, X) = p(y_1|\theta, \mathbf{x}_1)p(y_2|\theta, \mathbf{x}_1, y_1, \mathbf{x}_2)p(y_3|\theta, \{\mathbf{x}_i, y_i\}_{i=1}^2, \mathbf{x}_3)\dots \quad (4)$$

$$= \prod_{n=1}^{N} p(y_n|\theta, \{\mathbf{x}_i, y_i\}_{i=1}^{n-1}, \mathbf{x}_n) \quad (5)$$

Remember

$$p(y_n|\theta, \{\mathbf{x}_i, y_i\}_{i=1}^{n-1}, \mathbf{x}_n) = \int p(y_n|f(\mathbf{x}_n))p(f(\mathbf{x}_n) \mid \{\mathbf{x}_i, y_i\}_{i=1}^{n-1}, \mathbf{x}_n)\mathrm{d}f(\mathbf{x}_n)$$

i.e. the predictive distribution of y_n based on the posterior given all points up to $n-1$.

Marginal likelihood as incremental prediction

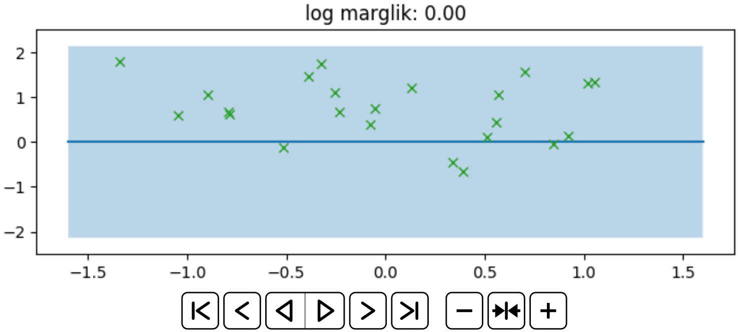
We can split the marginal likelihood up using the **product rule**:

$$p(\mathbf{y} \mid \theta, X) = p(y_1|\theta, \mathbf{x}_1)p(y_2|\theta, \mathbf{x}_1, y_1, \mathbf{x}_2)p(y_3|\theta, \{\mathbf{x}_i, y_i\}_{i=1}^2, \mathbf{x}_3)\dots \tag{6}$$

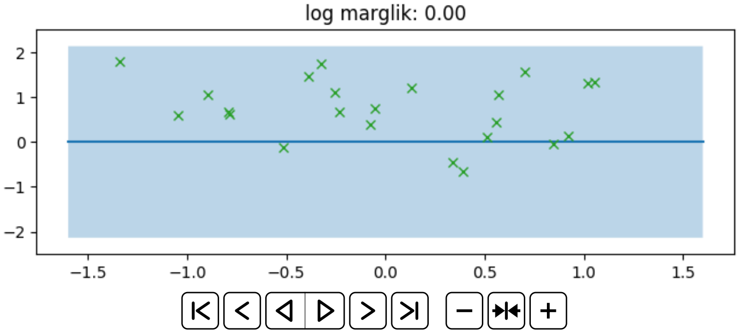
$$= \prod_{n=1}^{N} p(y_n|\theta, \{\mathbf{x}_i, y_i\}_{i=1}^{n-1}, \mathbf{x}_n) \tag{7}$$

- The marginal likelihood measures how well previous training points predict the next one
- If it continuously predicted well on all N points previously, it probably will do well next time

Marginal likelihood computation in action



Marginal likelihood computation in action



Invariance: A strong inductive bias

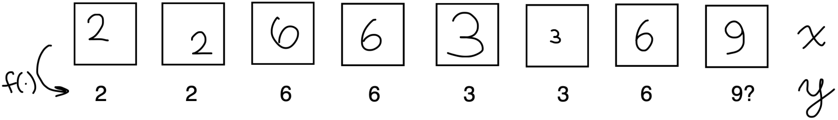
- Transformation on input, leaves output unchanged
- Single training point can generalise to many different new inputs!

Strict invariance:

$$f(t(\mathbf{x})) = f(\mathbf{x}) \qquad \forall \mathbf{x} \in \mathcal{X} \qquad \forall t \in \mathcal{T}$$

Insensitivity (weak invariance):

$$P\left([f(t(\mathbf{x})) - f(\mathbf{x})]^2 > L\right) < \delta \qquad \forall \mathbf{x} \in \mathcal{X} \qquad t \sim p(t)$$



Convolutions, data augmentation, group convolutions, ...

Invariances and Bayesian model selection

How can we learn the invariance using Bayesian model selection?

1. Specify a collection of priors on functions which satisfies a particular invariance.
2. Compute marginal likelihood for each invariant prior
3. Choose the invariance with the highest marginal likelihood

So how do we specify a prior on invariant functions?

Invariant priors

We can construct an invariant function $f(\cdot)$ from a non-invariant function $g(\cdot)$:

$$f(\mathbf{x}) = \sum_{\mathbf{x}_a \in \mathcal{O}(\mathbf{x})} g(\mathbf{x}_a) \qquad \text{or} \qquad f(\mathbf{x}) = \int g(t(\mathbf{x}))p(t)\mathrm{d}t$$

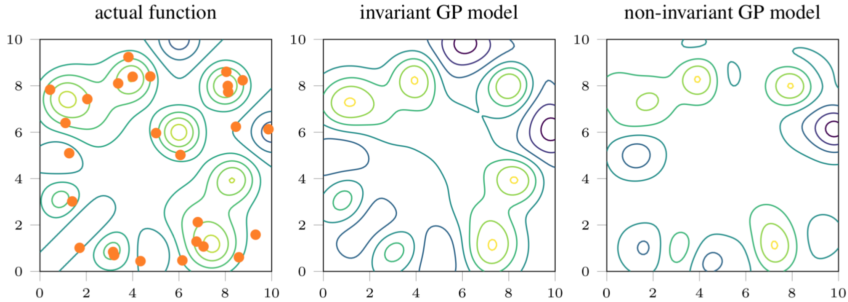
$$= \int g(\mathbf{x}_a)p(\mathbf{x}_a \,|\, \mathbf{x})\mathrm{d}\mathbf{x}_a$$

Average either over:

- (strict invariance) the **orbit** $\mathcal{O}(\mathbf{x})$ of a point $\mathbf{x}$. Set of points obtained from applying all transformations to $\mathbf{x}$.
- (weak invariance) the distribution on transformations, or equivalently the distribution of transformed images.

A prior on $g(\cdot)$ now implies a prior on $f(\cdot)$!

Example: Learning with reflective symmetry



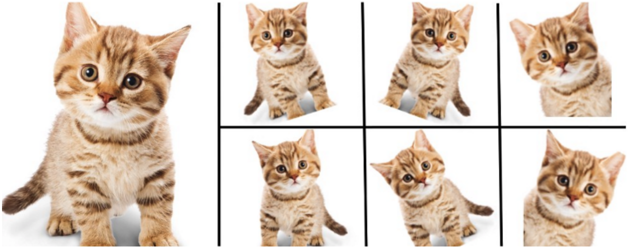
log marginal lik. = −34.906
training set RMSE = 0.007
test set RMSE = 0.43

log marginal lik. = −43.768
training set RMSE = 0.001
test set RMSE = 0.87

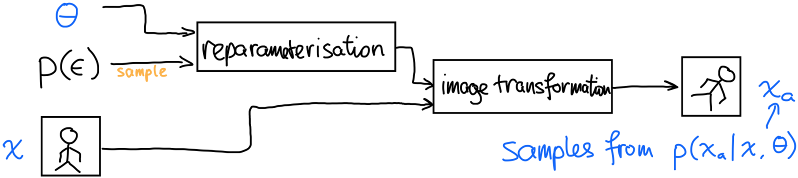
- Gaussian process prior on $g(\cdot)$.
- Orbit: $\mathcal{O}(\mathbf{x}) = \{(x_1, x_2), (x_2, x_1)\}$.
- Marginal likelihood correctly identifies the invariant model as the best

Learning Data Augmentation

Data augmentation expresses an invariance through a distribution of transformations on an input image.



[Image source: kdnuggets.com]



Learning Data Augmentation: Our method

- Take a **differentiable** transformation of the input image $\mathbf{x} \implies$ can sample from $p(\mathbf{x}_a \,|\, \mathbf{x}, \theta)$ using the reparameterisation trick.
- The parameters θ control the data augmentation.
- The transformation could be as flexible as a general neural network! We use
- Using a Gaussian process prior on $g(\cdot)$, define our "data augmentation invariant" function as

$$f(\mathbf{x}) = \int g(\mathbf{x}_a) p(\mathbf{x}_a \,|\, \mathbf{x}, \theta) \mathrm{d}\mathbf{x}_a = F(g(\cdot), \theta)\,.$$

Problems:

- $p(\mathbf{x}_a | \mathbf{x}, \theta)$ can be very complex, so integrals are intractable
- Our prior over $p(f|\theta)$ is therefore intractable

How do we compute the marginal likelihood?

Learning Data Augmentation: Variational inference

There is a **deterministic relationship** between $f(\cdot)$ and $g(\cdot)$.
$\implies$ it is equivalent to learn either!

- Approximate the posterior of $g(\cdot)$ instead of $f(\cdot)$ — this avoids needing to use the intractable prior on $f(\cdot)$
- Use **variational inference** to approximate the marginal likelihood.

$$\mathcal{L} = \sum_n \mathbb{E}_{q(g)}[\log p(y_n | F(g(\cdot), \theta))] - \mathrm{KL}[q(g)||p(g)]$$

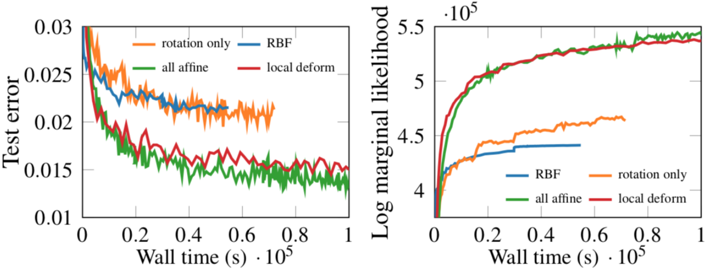
$$F(g(\cdot), \theta) = \int g(\mathbf{x}_a) p(\mathbf{x}_a \,|\, \mathbf{x}, \theta) \mathrm{d}\mathbf{x}_a$$

- Use Monte Carlo to estimate $F(g(\cdot), \theta)$ (see paper for details)

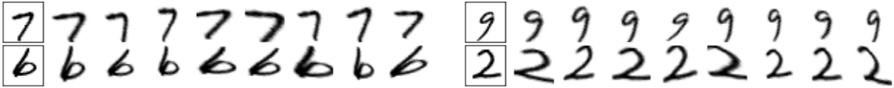
Result: Differentiable approximation to the marginal likelihood, with gradients to learn invariance parameters θ

Results: MNIST

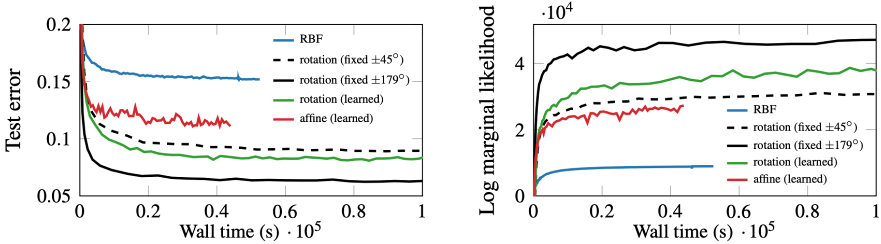
- Learns the invariance parameters through backprop
- Makes a weak Gaussian process classifier much stronger



Samples of $p(\mathbf{x}_a \mid \mathbf{x}, \theta)$, describing the **learned invariances**:



Results: MNIST-rot



- **Same** model on rotated MNIST dataset recovers **different** invariance
- No changes needed, whatsoever.
- Optimisation is difficult when using gradients, but the objective function correctly identifies the solution

Contributions

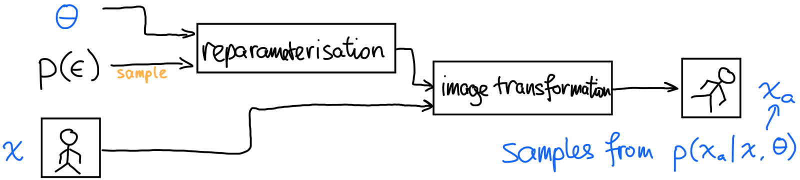
- We showed that invariances can (and from a Bayesian perspective, should) be expressed in the prior
- We developed an approximation to the marginal likelihood for invariant GP priors, that is **differentiable in the invariance parameters**
- We demonstrated the method on MNIST.

Limitation 1: Deep Neural Networks

Only works for Gaussian process models. Can we get it to work for deep models?

- Marginal likelihood framework is very general. All we need is a good approximation to the marginal likelihood. This doesn't yet exist for DNNs.
- Deep GPs are quite unique in that they are deep, **and** have **usable marginal likelihood estimates**.

Limitation 2: Parameterisation of invariances



An occasional question:

Ok, so invariances are defined through transformations, and you define your transformation in your augmentation procedure. If they are defined, are you really learning the invariance?

A fair point, but I still think the answer is **yes, we learn the invariance**. The parameter θ determines **which** transformations get used, and by **how much**.

- All learning methods need to define their parameter space
- Current approaches use **fixed** non-adaptable invariances that are built into the model.

Conclusions

- Bayes gives you more than uncertainty!
- Bayesian model selection is elegant and powerful (when you can approximate the marginal likelihood)
- Invariances are inductive biases, and they should be expressed in the prior.
- By expressing invariances in the prior, they can be learned using the marginal likelihood.

But most of all...

There is a lot more work to be done!

Interested in working on this?

- I'm keen to collaborate.
  - Particularly to get these ideas useful and relevant to larger-scale deep learning solutions.
- Consider applying for a PhD at Imperial College! Oct/Nov is a good time to apply to start in Oct 2021.

Some literature

Bayesian model selection:
Rasmussen and Ghahramani (2001); MacKay (2002); Murray and Ghahramani (2005); Rasmussen and Williams (2006)

Constructing invariant functions:
Minsky (1961); Kondor (2008); Ginsbourger et al. (2012)

Variational inference in Gaussian processes:
Titsias (2009); Hensman et al. (2013); van der Wilk et al. (2017, 2018)

A tutorial / overview of VI in GPs:
van der Wilk et al. (2020)

References I

Ginsbourger, D., Bay, X., Roustant, O., and Carraro, L. (2012). Argumentwise invariant kernels for the approximation of invariant functions. Annales de la Facult de Sciences de Toulouse.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI), pages 282–290.

Kondor, R. (2008). Group theoretical methods in machine learning. PhD thesis, Columbia University.

MacKay, D. J. C. (2002). Information Theory, Inference & Learning Algorithms. Cambridge University Press, USA.

Minsky, M. (1961). Steps toward artificial intelligence. Proceedings of the IRE, 49(1):8–30.

Murray, I. and Ghahramani, Z. (2005). A note on the evidence and bayesian occams razor.

References II

Rasmussen, C. E. and Ghahramani, Z. (2001). Occam's razor. In Advances in Neural Information Processing Systems 13.

Rasmussen, C. E. and Williams, C. K. (2006). Gaussian Processes for Machine Learning. MIT Press.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, pages 567–574.

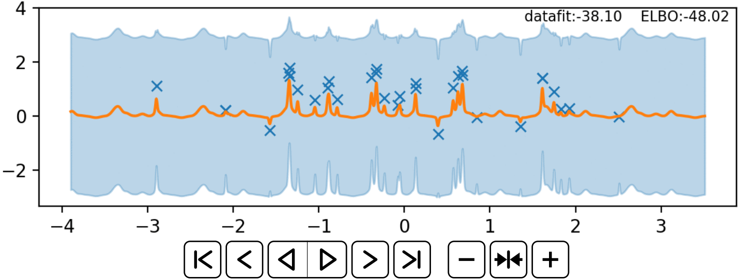
van der Wilk, M., Bauer, M., John, S., and Hensman, J. (2018). Learning invariances using the marginal likelihood. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, Advances in Neural Information Processing Systems 31, pages 9938–9948. Curran Associates, Inc.

van der Wilk, M., Dutordoir, V., John, S., Artemev, A., Adam, V., and Hensman, J. (2020). A framework for interdomain and multioutput gaussian processes.

References III

van der Wilk, M., Rasmussen, C. E., and Hensman, J. (2017). Convolutional gaussian processes. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, Advances in Neural Information Processing Systems 30, pages 2849–2858. Curran Associates, Inc.

Marginal likelihood in action



- Marginal likelihood learns **how** to generalise not just to fit the data.
- We chose the prior: $f(\mathbf{x}) = \theta_s f_{\text{smooth}}(\mathbf{x}) + \theta_p f_{\text{periodic}}(\mathbf{x})$, with smooth and periodic GP priors respectively.
- Amount of periodicity vs smoothness is automatically chosen by selecting hyperparameters θ_s, θ_p.

Marginal likelihood in action