UPPSALA UNIVERSITET

Advanced Probabilistic Machine Learning

Lecture 8 – Gaussian processes part II

UPPSALA UNIVERSITET

Andreas Lindholm
Division of Systems and Control
Department of Information Technology
Uppsala University

andreas.lindholm@it.uu.se
www.it.uu.se/katalog/andsv164

UPPSALA UNIVERSITET

Summary of lecture 7: The Gaussian process

Gaussian processes are a tool for regression, that is, describing the relationship between x and $y = f(x) + \epsilon$.

- Gaussian processes can be used for other problems than regression. Not in this course.
- In this presentation, x and y are always one-dimensional. Gaussian processes are not restricted to that case (only harder to illustrate).

Summary of lecture 7: Deriving the Gaussian process

For a finite vector $\mathbf{f}$, which we block in two parts $\mathbf{f} = \begin{bmatrix} \mathbf{f}_a \\ \mathbf{f}_b \end{bmatrix}$, we can assume a multivariate normal distribution

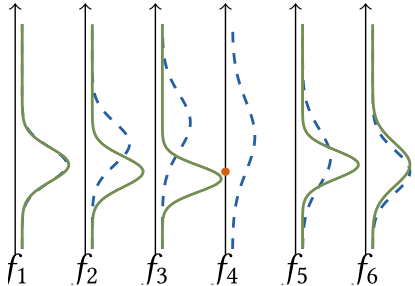
$$\begin{bmatrix} \mathbf{f}_a \\ \mathbf{f}_b \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix} \right)$$

and get

$$\mathbf{f}_a \,|\, \mathbf{f}_b \sim \mathcal{N}\left(\boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{f}_b - \boldsymbol{\mu}_b), \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \right)$$

If we observe $\mathbf{f}_b$, we get an updated prediction for $\mathbf{f}_a$ as $p(\mathbf{f}_a \,|\, \mathbf{f}_b)$

For example, let $\mathbf{f}_b = [f_1 \ f_2 \ f_3 \ f_5 \ f_6]^\mathsf{T}$ and $\mathbf{f}_b = f_4$.



Summary of lecture 7: Deriving the Gaussian process

The model for the finite vector $[f_1 \; f_2 \; \cdots \; f_n]^\mathsf{T}$

$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_n^2 \end{bmatrix} \right)$$

can be generalized to a model for $[f(x_1) \; f(x_2) \; \cdots \; f(x_n)]^\mathsf{T}$, with $x_1, x_2, \dots, x_n$ arbitrary, by using a *covariance function/kernel* $\kappa(x, x')$ such that

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \dots & \kappa(x_1, x_n) \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & \dots & \kappa(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(x_n, x_1) & \kappa(x_n, x_2) & \dots & \kappa(x_n, x_n) \end{bmatrix} \right)$$

Summary of lecture 7: The Gaussian process

With $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$, $f(\mathbf{X}) = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix}$, $K(x_\star, x_\star) = \kappa(x_\star, x_\star)$,

$K(\mathbf{X}, x_\star) = \begin{bmatrix} \kappa(x_1, x_\star) \\ \vdots \\ \kappa(x_N, x_\star) \end{bmatrix} = K(x_\star, \mathbf{X})^\mathsf{T}$ and $K(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} \kappa(x_1, x_1) & \dots & \kappa(x_1, x_N) \\ \vdots & & \vdots \\ \kappa(x_N, x_1) & \dots & \kappa(x_N, x_N) \end{bmatrix}$ and

$\mathbf{y} = f(\mathbf{X}) + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$ we have

$$\begin{bmatrix} \mathbf{y} \\ f(x_\star) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} & K(x_\star, \mathbf{X}) \\ K(\mathbf{X}, x_\star) & K(x_\star, x_\star) \end{bmatrix} \right)$$

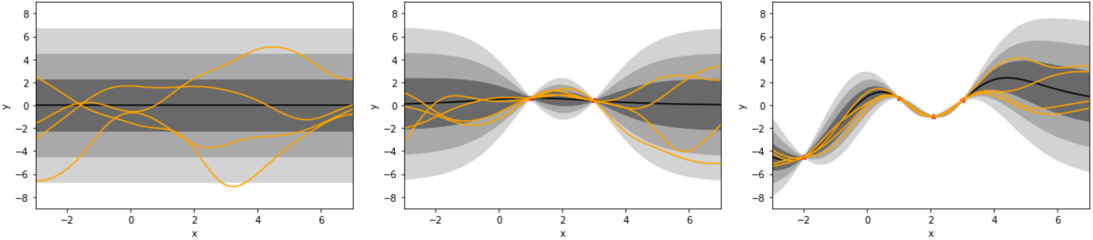
and most important

$$\boxed{f(x_\star) \,|\, \mathbf{y} \sim \mathcal{N}\left(\mathbf{K}(x_\star, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, K(x_\star, x_\star) - K(x_\star, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I})^{-1} K(\mathbf{X}, x_\star)\right)}$$

Summary of lecture 7: The Gaussian process

This gives a distribution over $f(x_\star)$, which we can condition on observations $\mathbf{y}$

$$f(x_\star)\,|\,\mathbf{y} \sim \mathcal{N}\left(\mathbf{K}(x_\star,\mathbf{X})(K(\mathbf{X},\mathbf{X})+\sigma_n^2\mathbf{I})^{-1}\mathbf{y}, K(x_\star,x_\star) - K(x_\star,\mathbf{X})(K(\mathbf{X},\mathbf{X})+\sigma_n^2\mathbf{I})^{-1}K(\mathbf{X},x_\star)\right)$$



(Here, $x_\star$ is a vector with one element for each pixel on the screen $\rightarrow$ the samples look continuous!)

Summary of lecture 7: Derivation from BLR

- It is also possible to derive Gaussian processes from Bayesian linear regression
- If we introduce nonlinear input/feature transformations $\phi(x)$, the covariance function/kernel becomes $\kappa(x, x') = \phi(x)^\mathsf{T}\phi(x')$
- We can use "the kernel trick" and choose $\kappa(x, x')$ directly without bothering about what $\phi(x)$ it corresponds to $\rightarrow$ Gaussian process regression

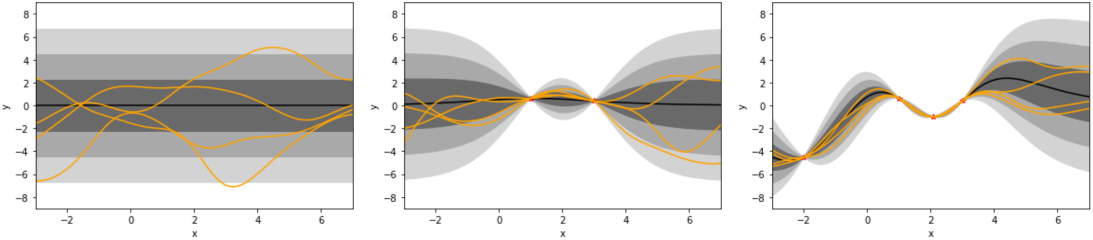
Today

How to choose kernel and its hyperparameters?

The Gaussian process

$$f(x_\star)\,|\,\mathbf{y} \sim \mathcal{N}\left(\mathbf{K}(x_\star,\mathbf{X})(K(\mathbf{X},\mathbf{X}){+}\sigma_n^2\mathbf{I})^{-1}\mathbf{y},\, K(x_\star,x_\star) - K(x_\star,\mathbf{X})(K(\mathbf{X},\mathbf{X}){+}\sigma_n^2\mathbf{I})^{-1}K(\mathbf{X},x_\star)\right)$$

$$\kappa(x,x') = \sigma^2\left(1+\frac{|x-x'|^2}{2\alpha\ell}\right)^{-\alpha}, \quad \sigma^2=5, \alpha=2, \ell=3$$



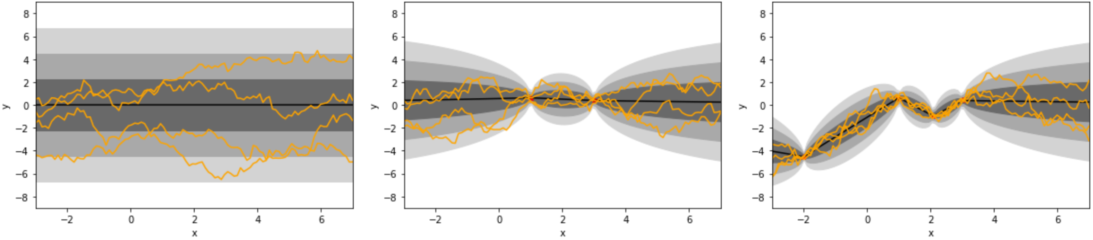
More examples at http://www.it.uu.se/edu/course/homepage/apml/GP/

The choice of kernel and hyperparameter is crucial!

The Gaussian process

$$f(x_\star)\,|\,\mathbf{y} \sim \mathcal{N}\left(\mathbf{K}(x_\star,\mathbf{X})(K(\mathbf{X},\mathbf{X})+\sigma_n^2\mathbf{I})^{-1}\mathbf{y},\, K(x_\star,x_\star) - K(x_\star,\mathbf{X})(K(\mathbf{X},\mathbf{X})+\sigma_n^2\mathbf{I})^{-1}K(\mathbf{X},x_\star)\right)$$

$$\kappa(x,x') = \sigma^2 \exp\left(1 + \frac{|x-x'|}{\ell^2}\right), \quad \sigma^2 = 5, \ell = 3$$



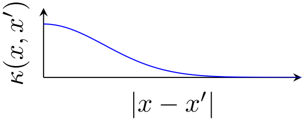
More examples at http://www.it.uu.se/edu/course/homepage/apml/GP/

The choice of kernel and hyperparameter is crucial!

Some kernels

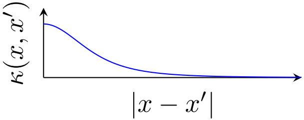
Squared exponential/RBF

$$\kappa(x, x') = \sigma^2 \exp(-\tfrac{1}{2\ell^2}(x - x')^2)$$



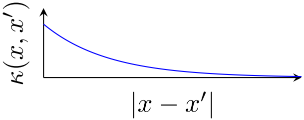
Rational quadratic

$$\kappa(x, x') = \sigma^2 \left(1 + \frac{|x-x'|^2}{2\alpha\ell}\right)^{-\alpha}$$



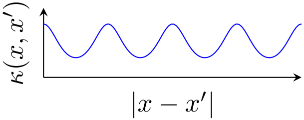
Matérn 1

$$\kappa(x, x') = \sigma^2 \exp(-\tfrac{1}{\ell^2}|x - x'|)$$



Periodic kernel

$$\kappa(x, x') = \sigma^2 \exp(-\tfrac{2}{\ell^2}\sin^2(\pi\tfrac{|x-x'|}{p}))$$



Importance of kernel choice

- The kernel $\kappa(x, x')$ encodes assumptions on how much correlation there is between $f(x)$ and $f(x')$
- The kernel tells how the model should generalize the training data

---

Even with prior mean 0, the predictive posterior does not have mean 0 thanks to the kernel.

Constructing new kernels

For a kernel valid for Gaussian processes, the matrix

$$K(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} \kappa(x_1, x_1) & \dots & \kappa(x_1, x_N) \\ \vdots & & \vdots \\ \kappa(x_N, x_1) & \dots & \kappa(x_N, x_N) \end{bmatrix}$$

must be positive semidefinite for all possible $\mathbf{X}$.

- You can invent completely new kernels, as long as they fulfill this criterion.
- You can create composite kernels by multiplying or adding existing ones

$$\kappa_\times(x, x') = \kappa_1(x, x')\kappa_2(x, x')$$
$$\kappa_+(x, x') = \kappa_1(x, x') + \kappa_2(x, x')$$

Measurement noise as part of the kernel

$$f(x_\star)\,|\,\mathbf{y} \sim \mathcal{N}\Big(\mathbf{K}(x_\star,\mathbf{X})(\underbrace{K(\mathbf{X},\mathbf{X})+\sigma_n^2\mathbf{I}})^{-1}\mathbf{y}, K(x_\star,x_\star) - K(x_\star,\mathbf{X})(\underbrace{K(\mathbf{X},\mathbf{X})+\sigma_n^2\mathbf{I}})^{-1}K(\mathbf{X},x_\star)\Big)$$

Sometimes σ_n^2 is seen as a part of the kernel, by defining

$$\tilde{\kappa}(x,x') = \kappa(x,x') + \sigma_n^2\mathbb{I}_{\{x=x'\}}$$

where $\mathbb{I}_{\{x=x'\}}$ is the identity function $\begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases}.$

The formulation is mathematically equivalent to what we have done previously, it is just a matter of book-keeping.

The function $\sigma_n^2\mathbb{I}_{\{x=x'\}}$ is itself a kernel, sometimes referred to as the “white noise kernel”.

Choosing kernels

In the end, the choice of kernel is a design choice left to the machine learning engineer.

Meaning of hyperparameters

http://www.it.uu.se/edu/course/homepage/apml/GP/

Choosing hyperparameters

How to choose the hyperparameters $\xi \triangleq \{\sigma_n^2, \ell, \dots\}$?

- The go-to solution for machine learning: **(k-fold) cross validation**
- A Bayesian alternative: **Maximizing the marginal likelihood**

Both approaches can be used in practice, we will have a closer look at the marginal likelihood.

Marginal likelihood

How to choose the hyperparameters $\xi \triangleq \{\sigma_n^2, \ell, \dots\}$?

The kernel function depends on ξ, hence we write $\kappa_\xi(x, x')$, $K_\xi(\mathbf{X}, \mathbf{X})$, etc.

The Gaussian process model says

$$p(f(\mathbf{X})) = \mathcal{N}(f(\mathbf{x}); \mathbf{0}, K_\xi(\mathbf{X}, \mathbf{X}))$$

and since $y = f(\mathbf{x}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_n^2)$,

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \mathbf{0}, K_\xi(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I})$$

$$\Rightarrow \log p(\mathbf{y}) = -\frac{1}{2}\mathbf{y}^\mathsf{T}(K_\xi(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log\det(K_\xi(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}) - \frac{N}{2}\log 2\pi$$

UPPSALA UNIVERSITET

Likelihood vs marginal likelihood

In parametric models (with θ, such as linear regression) we have

- Likelihood: $p(\mathbf{y} \mid \theta)$

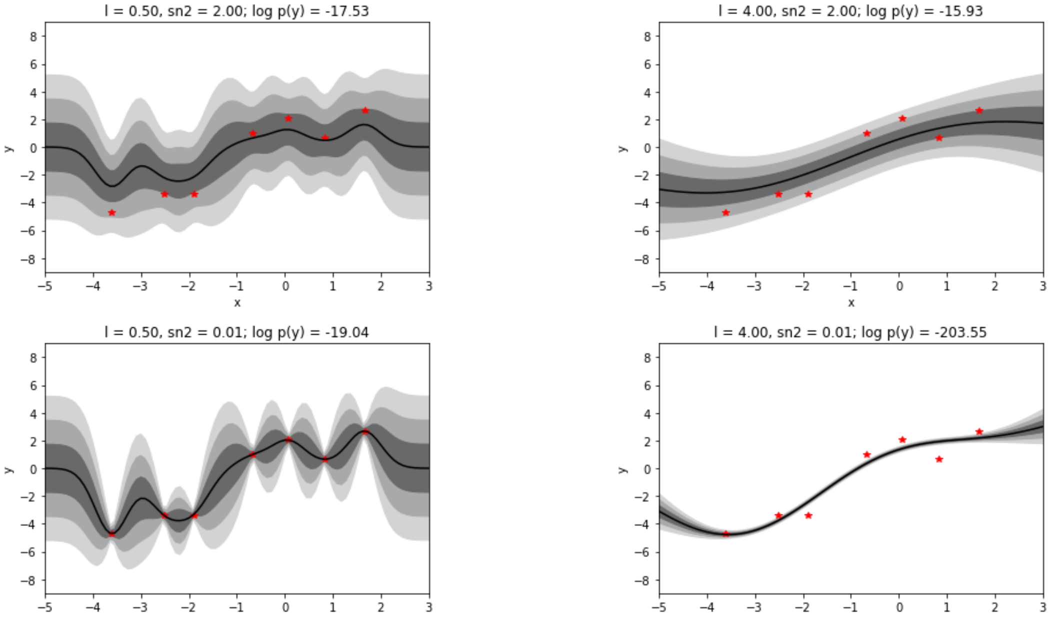
  Selecting θ by maximizing the likelihood often leads to overfit.

- Marginal likelihood: $p(\mathbf{y}) = \int p(\mathbf{y} \mid \theta) p(\theta) d\theta$

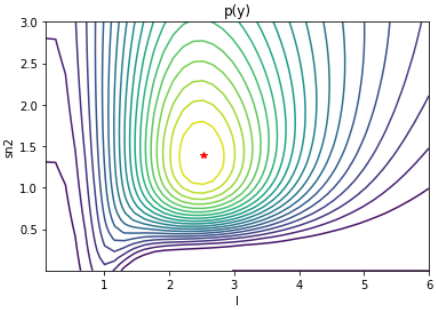
  Selecting hyperparameters by maximizing the marginal likelihood do (most of the time) not lead to overfit.

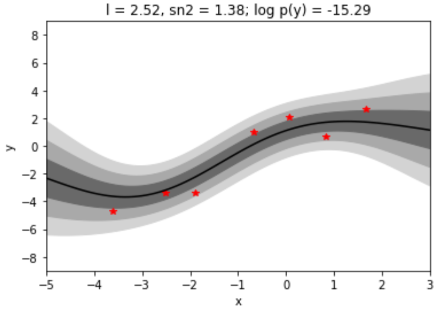
In non-parametric models like the Gaussian process (no finite-dimensional θ), we cannot really talk about $p(\mathbf{y} \mid \theta)$. The marginal likelihood $p(\mathbf{y})$, however, still exists.

The marginal likelihood landscape



The marginal likelihood landscape





We have here chosen $\xi = \{\sigma_n^2, \ell\}$ by maximizing the marginal likelihood.

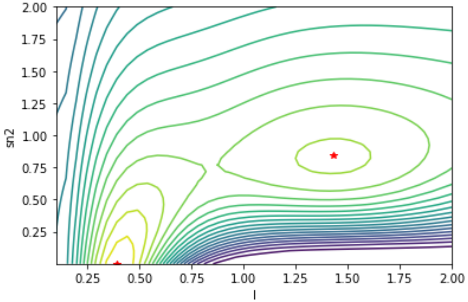
(In this example, $\kappa(x, x') = \exp(-\frac{1}{2\ell^2}(x - x')^2)$)

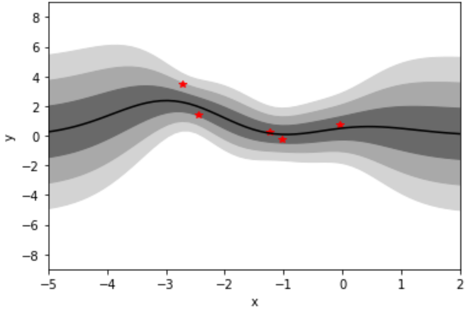
Note: maximizing the marginal likelihood does **not** necessarily mean choosing ξ such that the predictive posterior mean (black line) goes exactly through all training data!

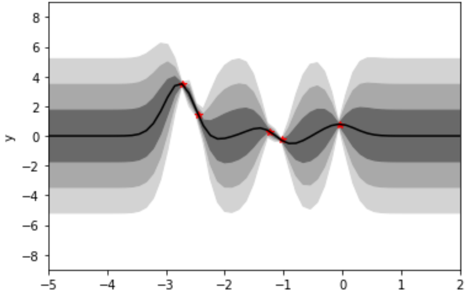
The marginal likelihood may have multiple minima

$$\kappa(x, x') = \sigma^2 \exp(-\frac{1}{2\ell^2}(x - x')^2)$$

$\sigma^2 = 3$; optimize σ_n^2 and ℓ



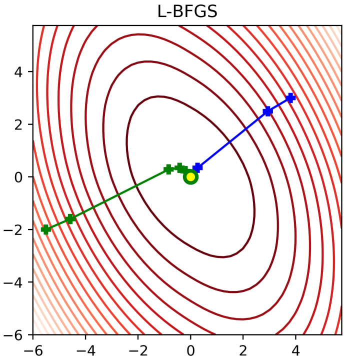


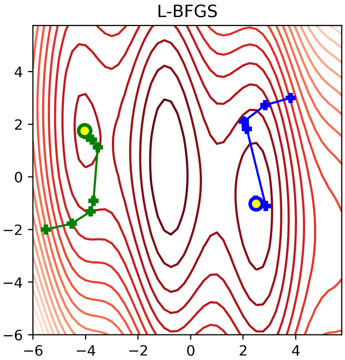


Optimizing the marginal likelihood

We have to use numerical optimization, such as BGFS or similar.

Idea: Compute the gradient $\nabla_{\xi} p(\mathbf{y})$ and numerically estimate the Hessian $\nabla^2_{\xi} p(\mathbf{y})$.
Take a "Newton step" $\xi^{t+1} \leftarrow \xi^{t} - [\nabla^2_{\xi} p(\mathbf{y})]^{-1} [\nabla_{\xi} p(\mathbf{y})]$.





It is important how you initialize your hyperparameter search!

Gaussian processes in machine learning

- The idea dates back to the 60's ('kriging'); model the presence of gold in South Africa based on information from boreholes (a regression problem!)
- Big interest within the machine learning research, because of its Bayesian and non-parametric nature
- Has not (yet?) become as popular among practitioners as, e.g., random forests and neural networks
- Many interesting research directions!

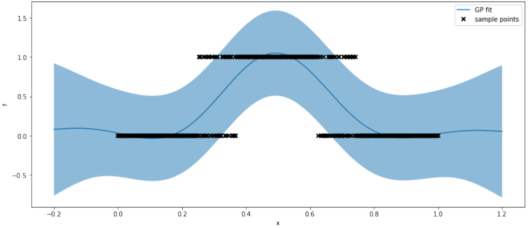
Outlook: Deep Gaussian processes

A deep neural network (“deep learning”) is a concatenation of multiple 1-layer neural networks.

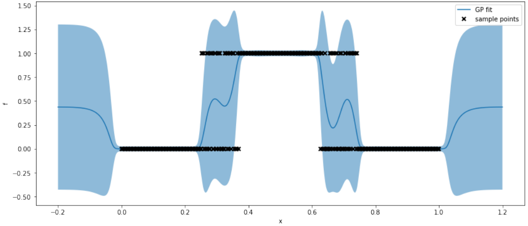
Can we construct a Deep Gaussian process by concatenating several Gaussian processes?

Yes $\rightarrow$ deep Gaussian process

Standard GP regression



Deep GP regression



Deep Gaussian Processes, A. Damianou and N. Lawrence, *AISTATS 2013*.
Deep Gaussian Processes for Regression using Approximate Expectation Propagation, T. Bui, J. M. Hernández-Lobato, D. Hernández-Lobato, Y. Li, R. Turner, *ICML 2016*.
How Deep Are Deep Gaussian Processes?, M. Dunlop, M. Girolami, A. Stuart, A. Teckentrup, *JMLR 19, 2018*.
`https://github.com/SheffieldML/PyDeepGP`
Images from `https://nbviewer.jupyter.org/github/gpschool/labs/blob/2019/2019/.answers/lab_2_extra.ipynb`

UPPSALA UNIVERSITET

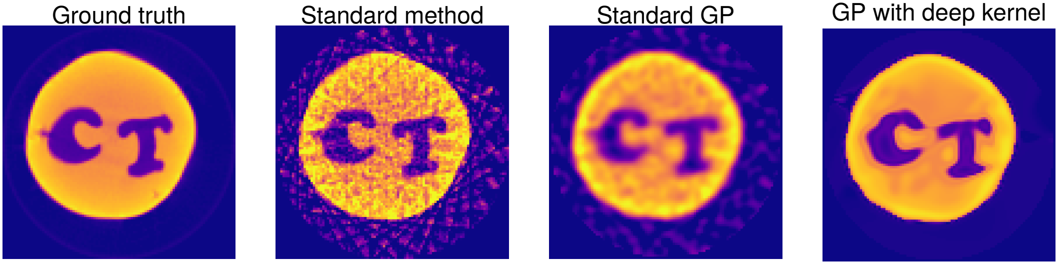
Outlook: Deep kernel learning

Can we use a (deep?) neural network to *learn* a kernel gor Gaussian process regression?

Yes $\rightarrow$ deep kernel learning

$\tilde{\kappa}(x, x') = \kappa(u(x), u(x'))$, $\kappa(\cdot, \cdot)$ is some standard kernel, and $u(\cdot)$ is a (deep) neural network to be learned from training data.

Research application of deep kernel learning: Reconstruct images from CT scans



Deep Kernel Learning, A. Wilson, Z. Hu, R. Salakhutdinov, E. Xing, *AISTATS 2016*.
Manifold Gaussian processes for regression, R. Calendra, J. Peters, C. E. Rasmussen, M. P. Deisenroth, *IJCNN 2016*.
Deep kernel learning for integral measurements, C. Jidling, J. Hendricks, T. B: Schön, A. wills, *ArXiv:1909:01844*.

A few concepts to summarize lecture 8

- Choosing and constructing kernels for Gaussian process regression
- Learning/estimating hyperparameters from data by maximizing the marginal likelihood