Special course on Gaussian processes
Session #8: Deep GPs

William Wilkinson

Aalto University

william.wilkinson@aalto.fi

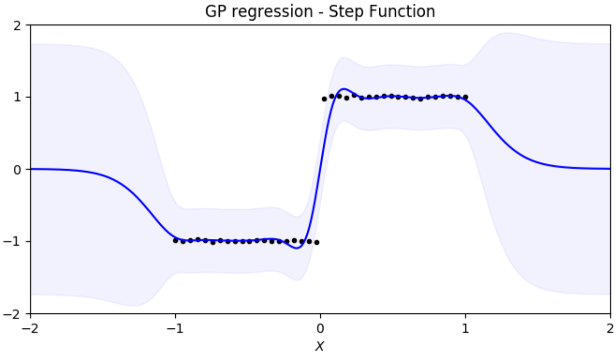
04/02/21

Roadmap for today

1. Introductions to Deep GPs
   - Limitations of standard GPs
   - Function Composition and Deep Learning
2. The Deep GP Model
   - Combining Layers of GPs
   - Deep GP Covariance
   - The Deep GP Posterior
3. Inference in Deep GPs
   - Stochastic Variational Inference
   - Alternative Approaches
   - Performance and Issues

Limitations of Standard GPs

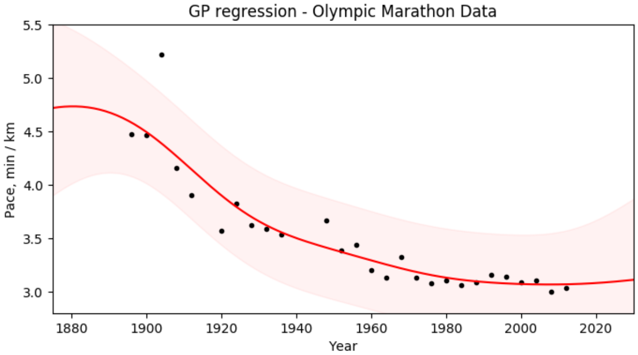
- **Discontinuities / jumps**



- A stationary GP fails to capture the sharp jump, and the variance is too large everywhere.

Limitations of Standard GPs

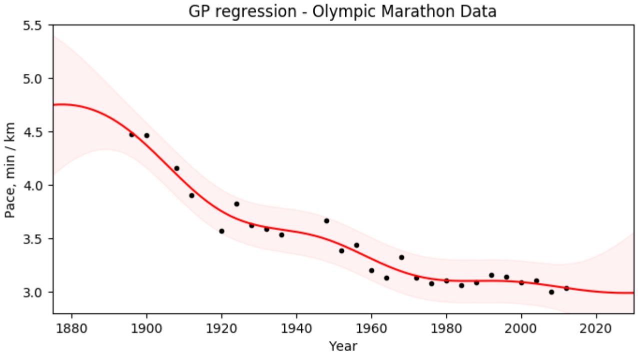
- **Discontinuities / jumps**
- **Outliers**



- The outlier has a *very* low probability under the model.
- To account for this, the model learns a likelihood variance that is too high for all other data points.

Limitations of Standard GPs

- **Discontinuities / jumps**
- **Outliers**

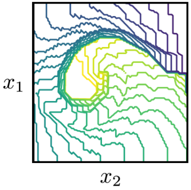


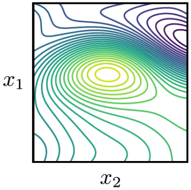
- Removing the outlier vastly improves the result. But we'd rather avoid such a manual intervention.

Limitations of Standard GPs

- **Discontinuities / jumps**
- **Outliers**
- **Non-stationarity**

The previous two problems can be seen as issues arising due to a *stationary* model being applied to non-stationary data.

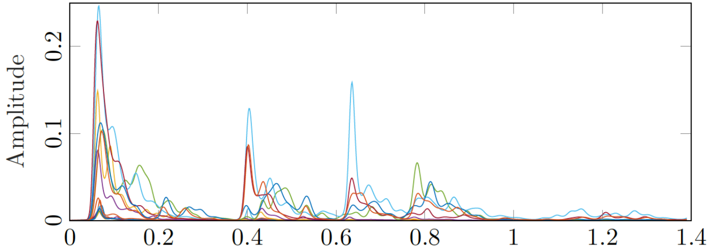




- Many real-world data sets do not have constant smoothness across the entire input space.

Limitations of Standard GPs

- **Discontinuities / jumps**
- **Outliers**
- **Non-stationarity**
- **Misalignment**



- Multiple misaligned data streams cannot be modelling with a standard (multi-output) GP.
- The data must be aligned via a pre-processing step.
- Ideally this step should be incorporated into the probabilistic model, so that its uncertainty can be incorporated.

Function Composition

- Function composition is at the heart of modern-day machine learning. Deep neural networks are made up of compositions of neural networks.
- Deep Gaussian processes work in an analogous way, whilst incorporating uncertainty and prior knowledge.

Function Composition

- Function composition is at the heart of modern-day machine learning. Deep neural networks are made up of compositions of neural networks.
- Deep Gaussian processes work in an analogous way, whilst incorporating uncertainty and prior knowledge.

Single GPs can model simple, stationary functions. The composition of multiple GPs,

$$f_3(f_2(f_1(\cdot)))$$

can model more complex, nonstationary functions.

Function Composition

- Function composition is at the heart of modern-day machine learning. Deep neural networks are made up of compositions of neural networks.
- Deep Gaussian processes work in an analogous way, whilst incorporating uncertainty and prior knowledge.

Single GPs can model simple, stationary functions. The composition of multiple GPs,

$$f_3(f_2(f_1(\cdot)))$$

can model more complex, nonstationary functions.

- We can view each "layer" as a warping of the inputs before feeding to the next layer.
- Function composition can be used to incorporate multiple layers of prior knowledge.

Deep GP Intuition

Before writing down the model, let's gain some intuition about hierarchies of Gaussian processes.

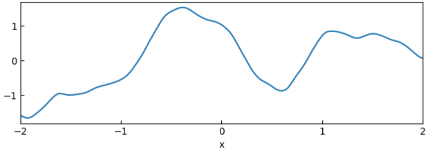
Deep GP Intuition

Before writing down the model, let's gain some intuition about hierarchies of Gaussian processes.

Take inputs $\boldsymbol{x}$, and evaluate a GP, $f_1(.) \sim \mathcal{GP}(\mu_1(.), \kappa_1(., .))$:

$$f_1(\boldsymbol{x}) \sim \mathcal{N}(\mu_1(\boldsymbol{x}), \kappa_1(\boldsymbol{x}, \boldsymbol{x}))$$

Draw a sample, $\tilde{\boldsymbol{y}}_1$, from this multivariate Gaussian:



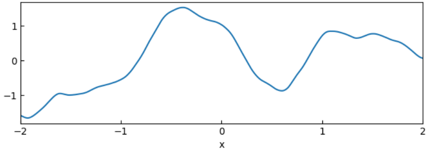
Deep GP Intuition

Before writing down the model, let's gain some intuition about hierarchies of Gaussian processes.

Take inputs $\boldsymbol{x}$, and evaluate a GP, $f_1(.) \sim \mathcal{GP}(\mu_1(.), \kappa_1(.,.))$:

$$f_1(\boldsymbol{x}) \sim \mathcal{N}(\mu_1(\boldsymbol{x}), \kappa_1(\boldsymbol{x}, \boldsymbol{x}))$$

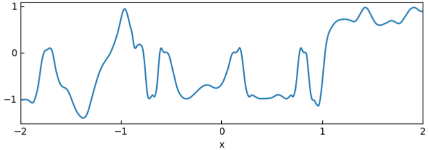
Draw a sample, $\tilde{\boldsymbol{y}}_1$, from this multivariate Gaussian:



Treat this sample as the input to *another* GP, $f_2(.) \sim \mathcal{GP}(\mu_2(.), \kappa_2(.,.))$:

$$f_2(\tilde{\boldsymbol{y}}_1) \sim \mathcal{N}(\mu_2(\tilde{\boldsymbol{y}}_1), \kappa_2(\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_1))$$

and draw a sample, $\tilde{\boldsymbol{y}}_2$.



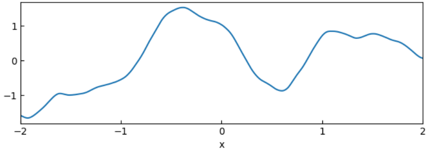
Deep GP Intuition

Before writing down the model, let's gain some intuition about hierarchies of Gaussian processes.

Take inputs $\boldsymbol{x}$, and evaluate a GP, $f_1(.) \sim \mathcal{GP}(\mu_1(.), \kappa_1(.,.))$:

$$f_1(\boldsymbol{x}) \sim \mathcal{N}(\mu_1(\boldsymbol{x}), \kappa_1(\boldsymbol{x}, \boldsymbol{x}))$$

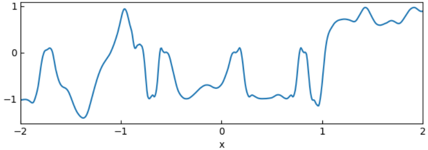
Draw a sample, $\tilde{\boldsymbol{y}}_1$, from this multivariate Gaussian:



Treat this sample as the input to *another* GP, $f_2(.) \sim \mathcal{GP}(\mu_2(.), \kappa_2(.,.))$:

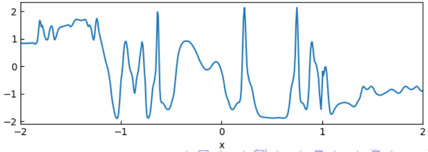
$$f_2(\tilde{\boldsymbol{y}}_1) \sim \mathcal{N}(\mu_2(\tilde{\boldsymbol{y}}_1), \kappa_2(\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_1))$$

and draw a sample, $\tilde{\boldsymbol{y}}_2$.

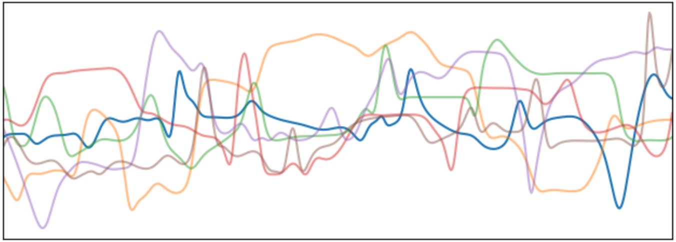


Repeat a third time for $f_3(.) \sim \mathcal{GP}(\mu_3(.), \kappa_3(.,.))$:

$$f_3(\tilde{\boldsymbol{y}}_2) \sim \mathcal{N}(\mu_3(\tilde{\boldsymbol{y}}_2), \kappa_3(\tilde{\boldsymbol{y}}_2, \tilde{\boldsymbol{y}}_2))$$

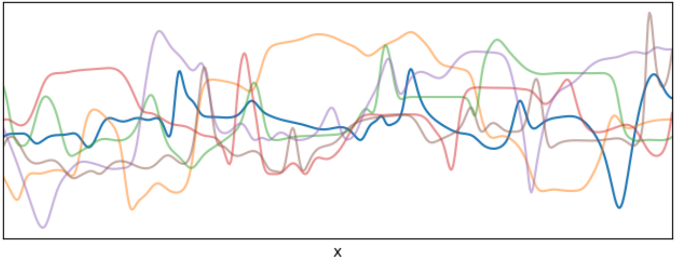


Deep GP Intuition



These are samples from a 3-layer deep GP.

Deep GP Intuition

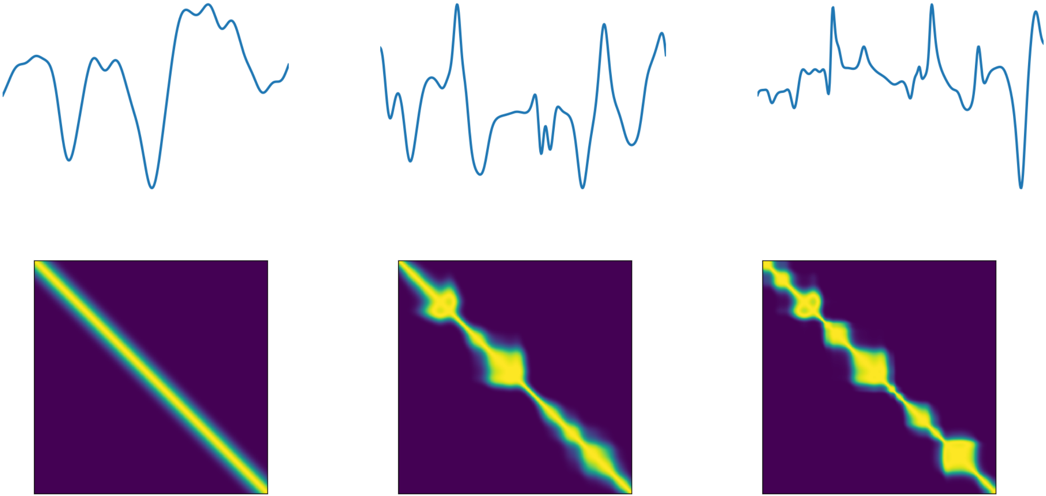


These are samples from a 3-layer deep GP.

- sharp jumps / discontinuities.
- highly nonstationary smoothness.

Deep GP Covariance

As well as sampling, we can also plot the covariance matrix in each layer.



The Deep GP Model

Now let's write down the deep GP model and look at its properties. Inference will come later.

$$
\begin{aligned}
f_i(\cdot) &\sim \mathcal{GP}\left(\mu_i(\cdot), \kappa_i(\cdot,\cdot)\right), \qquad && i = 1, \ldots, L \\
p(\tilde{\boldsymbol{y}}_i \mid f_i, \tilde{\boldsymbol{y}}_{i-1}) &= \prod_n \mathcal{N}(\tilde{y}_{i,n} \mid f_i(\tilde{y}_{i-1,n}), \sigma_i^2), \qquad && \tilde{\boldsymbol{y}}_1 = \boldsymbol{x} \\
p(\boldsymbol{y} \mid f_L, \tilde{\boldsymbol{y}}_{L-1}) &= \prod_n p(y_n \mid f_L(\tilde{y}_{L-1,n}))
\end{aligned}
$$

The Deep GP Model

Now let's write down the deep GP model and look at its properties. Inference will come later.

$$
\begin{aligned}
f_i(\cdot) &\sim \mathcal{GP}\left(\mu_i(\cdot), \kappa_i(\cdot,\cdot)\right), \qquad i = 1, \dots, L \\
p(\tilde{\boldsymbol{y}}_i \mid f_i, \tilde{\boldsymbol{y}}_{i-1}) &= \prod_n \mathcal{N}(\tilde{y}_{i,n} \mid f_i(\tilde{y}_{i-1,n}), \sigma_i^2), \qquad \tilde{\boldsymbol{y}}_1 = \boldsymbol{x} \\
p(\boldsymbol{y} \mid f_L, \tilde{\boldsymbol{y}}_{L-1}) &= \prod_n p(y_n \mid f_L(\tilde{y}_{L-1,n}))
\end{aligned}
$$

- L layers of Gaussian process priors.
- $\tilde{\boldsymbol{y}}_i$ are latent variables - treated as input to layer $i+1$.
- Typically include Gaussian noise between layers.

The Deep GP Model

Now let's write down the deep GP model and look at its properties. Inference will come later.

$$f_i(\cdot) \sim \mathcal{GP}\left(\mu_i(\cdot), \kappa_i(\cdot,\cdot)\right), \qquad i = 1, \ldots, L$$
$$p(\tilde{\boldsymbol{y}}_i \mid f_i,\, \tilde{\boldsymbol{y}}_{i-1}) = \prod_n \mathcal{N}(\tilde{y}_{i,n} \mid f_i(\tilde{y}_{i-1,n}),\, \sigma_i^2), \qquad \tilde{\boldsymbol{y}}_1 = \boldsymbol{x}$$
$$p(\boldsymbol{y} \mid f_L,\, \tilde{\boldsymbol{y}}_{L-1}) = \prod_n p(y_n \mid f_L(\tilde{y}_{L-1,n}))$$

- L layers of Gaussian process priors.
- $\tilde{\boldsymbol{y}}_i$ are latent variables - treated as input to layer $i+1$.
- Typically include Gaussian noise between layers.
- For notational convenience, we can drop the explicit Gaussian noise between layers by moving the noise into the kernel, $\kappa_i(\cdot,\cdot)$.

The Deep GP Model

Now let's write down the deep GP model and look at its properties. Inference will come later.

$$p(f_i \mid f_{i-1}) = \mathcal{GP}(\cdot,\cdot)\,, \qquad i = 1,\ldots,L$$
$$p(\boldsymbol{y} \mid f_L) = \prod_n p(y_n \mid f_{L,n})$$

where $f_0 = \boldsymbol{x}$ and $f_{L,n} = f_L(f_{L-1}(\ldots(x_n)))$.

- L layers of Gaussian process priors.
- $\tilde{\boldsymbol{y}}_i$ are latent variables - treated as input to layer $i+1$.
- Typically include Gaussian noise between layers.
- For notational convenience, we can drop the explicit Gaussian noise between layers by moving the noise into the kernel, $\kappa_i(\cdot,\cdot)$.

The Deep GP Model

Now let's write down the deep GP model and look at its properties. Inference will come later.

$$p(f_i \mid f_{i-1}) = \mathcal{GP}(\cdot, \cdot)\,, \qquad i = 1, \ldots, L$$
$$p(\boldsymbol{y} \mid f_L) = \prod_n p(y_n \mid f_{L,n})$$

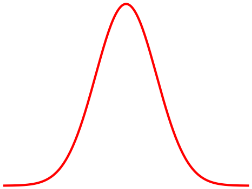
where $f_0 = \boldsymbol{x}$ and $f_{L,n} = f_L(f_{L-1}(\ldots(x_n)))$.

Using notation $\boldsymbol{f}_i = f(\boldsymbol{f}_{i-1})$, the full process has joint density

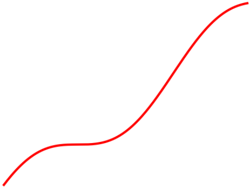
$$p(\boldsymbol{y}, \{\boldsymbol{f}_i\}_{i=1}^{L}) = \underbrace{\prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n})}_{\text{Likelihood}} \underbrace{\prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1})}_{\text{Deep GP Prior}}$$

The Deep GP Posterior

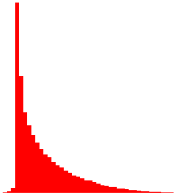
A Gaussian propagated through a nonlinearity is no longer Gaussian:



$x \sim \mathcal{N}(x \mid \cdot, \cdot)$



$f(\cdot)$

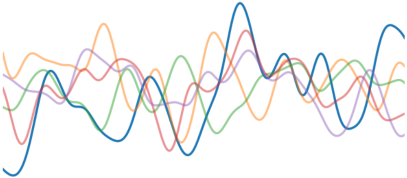


$f(x) \sim$ **???**

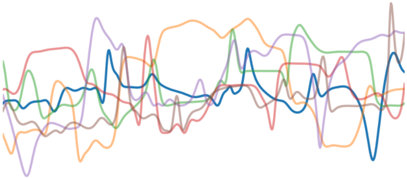
The Deep GP Posterior

Similarly, a Gaussian process propagated through a nonlinearity (e.g., another GP) is no longer a Gaussian process (in the original inputs $\boldsymbol{x}$).

$$f_1(\cdot) \sim \mathcal{GP}\left(\mu_1(\cdot), \kappa_1(\cdot,\cdot)\right)$$
$$f_2(\cdot) \sim \mathcal{GP}\left(\mu_2(\cdot), \kappa_2(\cdot,\cdot)\right)$$



$f_1(\boldsymbol{x}) \sim \mathcal{GP}(\cdot,\cdot)$



$f_2(f_1(\boldsymbol{x})) \sim$ **???**

Inference in Deep GPs

Since the posterior is not Gaussian, it is clear that we must resort to approximate inference.

- Various schemes have been proposed: Variational Inference, Expectation Propagation, Hamiltonian Monte Carlo.
- We will focus on **sparse, stochastic variational inference**.

Inference in Deep GPs

Since the posterior is not Gaussian, it is clear that we must resort to approximate inference.

- Various schemes have been proposed: Variational Inference, Expectation Propagation, Hamiltonian Monte Carlo.
- We will focus on **sparse, stochastic variational inference**.
- Recall our (joint) model:

$$p(\boldsymbol{y}, \{\boldsymbol{f}_i\}_{i=1}^{L}) = \prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1})$$

where $\boldsymbol{f}_0 = \boldsymbol{x}$.

Inference in Deep GPs

Since the posterior is not Gaussian, it is clear that we must resort to approximate inference.

- Various schemes have been proposed: Variational Inference, Expectation Propagation, Hamiltonian Monte Carlo.
- We will focus on **sparse, stochastic variational inference**.
- Recall our (joint) model:

$$p(\boldsymbol{y}, \{\boldsymbol{f}_i\}_{i=1}^L) = \prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1})$$

  where $\boldsymbol{f}_0 = \boldsymbol{x}$.

- We introduce inducing points $\boldsymbol{z}_i$ in each layer:

$$p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) = \prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)$$

  where $\boldsymbol{u}_i = f_i(\boldsymbol{z}_i)$. $p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i)$ is a standard Gaussian conditional.

Stochastic VI for Sparse Deep GPs

$$p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) = \prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)$$

Stochastic VI for Sparse Deep GPs

$$p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) = \prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)$$

- To construct a variational lower bound for the deep GP, we must first define an **approximate posterior**:

$$q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) = \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) q(\boldsymbol{u}_i)$$

where $q(\boldsymbol{u}_i) = \mathcal{N}(\boldsymbol{u}_i \mid \boldsymbol{m}_i, \boldsymbol{S}_i)$ are free-form Gaussians whose parameters are to be optimised.

Stochastic VI for Sparse Deep GPs

- Recall the sparse variational bound for a single GP derived in previous lectures:

$$
\begin{aligned}
\ln p(\boldsymbol{y}) \geq \mathcal{L}_3 &\equiv \sum_{n=1}^{N} \int q(f_n) \ln p(y_n|f_n)\, \mathrm{d}f_n - \mathbb{D}\left[q(\boldsymbol{u})||p(\boldsymbol{u})\right] \\
&= \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}\left[\ln p(\boldsymbol{y}|\boldsymbol{f})\right] + \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}\left[\ln p(\boldsymbol{f},\boldsymbol{u})\right] - \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}\left[\ln q(\boldsymbol{f},\boldsymbol{u})\right] \\
&= \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}\left[\ln \frac{p(\boldsymbol{y},\boldsymbol{f},\boldsymbol{u})}{q(\boldsymbol{f},\boldsymbol{u})}\right]
\end{aligned}
$$

- We will now derive a similar bound for the deep GP.

Stochastic VI for Sparse Deep GPs

$$\textbf{joint:} \quad p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) = \prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)$$

$$\textbf{approx. posterior:} \quad q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) = \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) q(\boldsymbol{u}_i)$$

Stochastic VI for Sparse Deep GPs

$$\textbf{joint:} \quad p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) = \prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)$$

$$\textbf{approx. posterior:} \quad q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) = \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) q(\boldsymbol{u}_i)$$

- The variational bound is

$$\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} = \mathbb{E}_{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L})} \left[\ln \frac{p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L})}{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L})} \right]$$

Stochastic VI for Sparse Deep GPs

$$\textbf{joint:} \quad p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) = \prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)$$

$$\textbf{approx. posterior:} \quad q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) = \prod_{i=1}^{L} p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) q(\boldsymbol{u}_i)$$

- The variational bound is

$$\begin{aligned}
\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} &= \mathbb{E}_{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L})} \left[\ln \frac{p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L})}{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L})} \right] \\
&= \int \int q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}) \ln \left(\frac{p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L})}{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L})} \right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}
\end{aligned}$$

Stochastic VI for Sparse Deep GPs

$$\textbf{joint:} \quad p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) = \prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)$$

$$\textbf{approx. posterior:} \quad q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) = \prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) q(\boldsymbol{u}_i)$$

- The variational bound is

$$\begin{aligned}
\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} &= \mathbb{E}_{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)} \left[\ln \frac{p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)}{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)} \right] \\
&= \int \int q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) \ln \left(\frac{p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)}{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)} \right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L \\
&= \int \int q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) \ln \left(\frac{\prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)}{\prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) q(\boldsymbol{u}_i)} \right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L
\end{aligned}$$

Stochastic VI for Sparse Deep GPs

$$\textbf{joint:} \quad p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) = \prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)$$

$$\textbf{approx. posterior:} \quad q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) = \prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) q(\boldsymbol{u}_i)$$

- The variational bound is

$$\begin{aligned}
\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} &= \mathbb{E}_{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)} \left[\ln \frac{p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)}{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)} \right] \\
&= \int \int q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) \ln \left(\frac{p(\boldsymbol{y}, \{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)}{q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L)} \right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L \\
&= \int \int q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) \ln \left(\frac{\prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) p(\boldsymbol{u}_i)}{\prod_{i=1}^L p(\boldsymbol{f}_i \mid \boldsymbol{f}_{i-1}, \boldsymbol{u}_i) q(\boldsymbol{u}_i)} \right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L \\
&= \int \int q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) \ln \left(\frac{\prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^L p(\boldsymbol{u}_i)}{\prod_{i=1}^L q(\boldsymbol{u}_i)} \right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L
\end{aligned}$$

Stochastic VI for Sparse Deep GPs

- Simplifying further:

$$\mathcal{L}_{DGP} = \int \int q(\{\boldsymbol{f}_i,\, \boldsymbol{u}_i\}_{i=1}^{L}) \ln \left(\frac{\prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^{L} p(\boldsymbol{u}_i)}{\prod_{i=1}^{L} q(\boldsymbol{u}_i)} \right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}$$

Stochastic VI for Sparse Deep GPs

- Simplifying further:

$$\mathcal{L}_{DGP} = \int\int q(\{\boldsymbol{f}_i,\, \boldsymbol{u}_i\}_{i=1}^{L}) \ln\left(\frac{\prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^{L} p(\boldsymbol{u}_i)}{\prod_{i=1}^{L} q(\boldsymbol{u}_i)}\right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}$$

$$= \int\int q(\{\boldsymbol{f}_i,\, \boldsymbol{u}_i\}_{i=1}^{L}) \ln\left(\prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n})\right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}$$

$$+ \int\int q(\{\boldsymbol{f}_i,\, \boldsymbol{u}_i\}_{i=1}^{L}) \ln\left(\frac{\prod_{i=1}^{L} p(\boldsymbol{u}_i)}{\prod_{i=1}^{L} q(\boldsymbol{u}_i)}\right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}$$

Stochastic VI for Sparse Deep GPs

- Simplifying further:

$$\begin{aligned}
\mathcal{L}_{DGP} &= \int\int q(\{\boldsymbol{f}_i,\, \boldsymbol{u}_i\}_{i=1}^{L}) \ln\left(\frac{\prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^{L} p(\boldsymbol{u}_i)}{\prod_{i=1}^{L} q(\boldsymbol{u}_i)}\right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L} \\
&= \int\int q(\{\boldsymbol{f}_i,\, \boldsymbol{u}_i\}_{i=1}^{L}) \ln\left(\prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n})\right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L} \\
&\quad + \int\int q(\{\boldsymbol{f}_i,\, \boldsymbol{u}_i\}_{i=1}^{L}) \ln\left(\frac{\prod_{i=1}^{L} p(\boldsymbol{u}_i)}{\prod_{i=1}^{L} q(\boldsymbol{u}_i)}\right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{L}
\end{aligned}$$

- The likelihood (first term) only depends on $\boldsymbol{f}_L$, and the second term does not depend on $\boldsymbol{f}_i$. So finally, the bound reduces to:

$$\mathcal{L}_{DGP} = \int q(\boldsymbol{f}_L) \ln\left(\prod_{n=1}^{N} p(y_n \mid \boldsymbol{f}_{L,n})\right) \mathrm{d}\boldsymbol{f}_L + \int q(\{\boldsymbol{u}_i\}_{i=1}^{L}) \ln\left(\frac{\prod_{i=1}^{L} p(\boldsymbol{u}_i)}{\prod_{i=1}^{L} q(\boldsymbol{u}_i)}\right) \mathrm{d}\{\boldsymbol{u}_i\}_{i=1}^{L}$$

Stochastic VI for Sparse Deep GPs

- Simplifying further:

$$\mathcal{L}_{DGP} = \int\int q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) \ln\left(\frac{\prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n}) \prod_{i=1}^L p(\boldsymbol{u}_i)}{\prod_{i=1}^L q(\boldsymbol{u}_i)}\right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L$$

$$= \int\int q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) \ln\left(\prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n})\right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L$$

$$+ \int\int q(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L) \ln\left(\frac{\prod_{i=1}^L p(\boldsymbol{u}_i)}{\prod_{i=1}^L q(\boldsymbol{u}_i)}\right) \mathrm{d}\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^L$$

- The likelihood (first term) only depends on $\boldsymbol{f}_L$, and the second term does not depend on $\boldsymbol{f}_i$. So finally, the bound reduces to:

$$\mathcal{L}_{DGP} = \int q(\boldsymbol{f}_L) \ln\left(\prod_{n=1}^N p(y_n \mid \boldsymbol{f}_{L,n})\right) \mathrm{d}\boldsymbol{f}_L - \sum_{i=1}^L \mathbb{D}\left[q(\boldsymbol{u}_i) || p(\boldsymbol{u}_i)\right]$$

Stochastic VI for Sparse Deep GPs

- The single GP bound:

$$\ln p(\boldsymbol{y}) \geq \mathcal{L}_3 = \sum_{n=1}^{N} \int q(f_n) \ln p(y_n | f_n) \mathrm{d} f_n - \mathbb{D}\left[q(\boldsymbol{u}) || p(\boldsymbol{u})\right]$$

- The deep GP bound:

$$\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} = \sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d} f_{L,n} - \sum_{i=1}^{L} \mathbb{D}\left[q(\boldsymbol{u}_i) || p(\boldsymbol{u}_i)\right]$$

Stochastic VI for Sparse Deep GPs

- The single GP bound:

$$\ln p(\mathbf{y}) \geq \mathcal{L}_3 = \sum_{n=1}^{N} \int q(f_n) \ln p(y_n|f_n) \mathrm{d}f_n - \mathbb{D}\left[q(\boldsymbol{u})||p(\boldsymbol{u})\right]$$

- The deep GP bound:

$$\ln p(\mathbf{y}) \geq \mathcal{L}_{DGP} = \sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n|f_{L,n})\, \mathrm{d}f_{L,n} - \sum_{i=1}^{L} \mathbb{D}\left[q(\boldsymbol{u}_i)||p(\boldsymbol{u}_i)\right]$$

- Notice that the first term still decomposes across the data points, **and only depends on the posterior marginal at the last layer**. Therefore this bound is also amenable to stochastic optimisation (*i.e.*, mini-batching).

Stochastic VI for Sparse Deep GPs

- The single GP bound:

$$\ln p(\mathbf{y}) \geq \mathcal{L}_3 = \sum_{n=1}^{N} \int q(f_n) \ln p(y_n|f_n) \mathrm{d}f_n - \mathbb{D}\left[q(\mathbf{u})||p(\mathbf{u})\right]$$

- The deep GP bound:

$$\ln p(\mathbf{y}) \geq \mathcal{L}_{DGP} = \sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n|f_{L,n}) \, \mathrm{d}f_{L,n} - \sum_{i=1}^{L} \mathbb{D}\left[q(\mathbf{u}_i)||p(\mathbf{u}_i)\right]$$

- Notice that the first term still decomposes across the data points, **and only depends on the posterior marginal at the last layer**. Therefore this bound is also amenable to stochastic optimisation (*i.e.*, mini-batching).
- However, computing the marginal $q(f_{L,n})$ is hard.

Computing the Deep GP Marginal

- Computing the marginal $q(f_{L,n})$ is the final step in performing inference.

- Fortunately, our model choices make *sampling* from this marginal efficient.

Computing the Deep GP Marginal

- Computing the marginal $q(f_{L,n})$ is the final step in performing inference.
- Fortunately, our model choices make *sampling* from this marginal efficient.
- To see this, consider the marginal distribution for a single layer, i. We obtain the marginal for a single point by integrating out the inducing variables from the approximate posterior:

$$q(\boldsymbol{f}_{i,n}) = \int q(\boldsymbol{f}_{i,n} \mid \boldsymbol{u}_i) q(\boldsymbol{u}_i) \, \mathrm{d}\boldsymbol{u}_i = \mathcal{N}(\boldsymbol{f}_i \mid \cdot, \cdot)$$

Computing the Deep GP Marginal

- Computing the marginal $q(f_{L,n})$ is the final step in performing inference.
- Fortunately, our model choices make *sampling* from this marginal efficient.
- To see this, consider the marginal distribution for a single layer, i. We obtain the marginal for a single point by integrating out the inducing variables from the approximate posterior:

$$q(\boldsymbol{f}_{i,n}) = \int q(\boldsymbol{f}_{i,n} \mid \boldsymbol{u}_i) q(\boldsymbol{u}_i) \, \mathrm{d}\boldsymbol{u}_i = \mathcal{N}(\boldsymbol{f}_i \mid \cdot, \cdot)$$

- It follows that, given $q(\boldsymbol{u}_i)$, computing $q(\boldsymbol{f}_{i,n})$ only requires knowledge of the marginal inputs $\boldsymbol{f}_{i-1,n}$.

Computing the Deep GP Marginal

- Computing the marginal $q(f_{L,n})$ is the final step in performing inference.
- Fortunately, our model choices make *sampling* from this marginal efficient.
- To see this, consider the marginal distribution for a single layer, i. We obtain the marginal for a single point by integrating out the inducing variables from the approximate posterior:

$$q(\boldsymbol{f}_{i,n}) = \int q(\boldsymbol{f}_{i,n} \mid \boldsymbol{u}_i) q(\boldsymbol{u}_i) \, \mathrm{d}\boldsymbol{u}_i = \mathcal{N}(\boldsymbol{f}_i \mid \cdot, \cdot)$$

- It follows that, given $q(\boldsymbol{u}_i)$, computing $q(\boldsymbol{f}_{i,n})$ only requires knowledge of the marginal inputs $\boldsymbol{f}_{i-1,n}$.
- This mean that sampling from $q(\boldsymbol{f}_{i,n})$ is cheap, and does not involve sampling from the full GP at each layer (in fact, it only requires sampling from univariate Gaussians).

Optimising the Deep GP Bound

$$\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} = \sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d}f_{L,n} - \sum_{i=1}^{L} \mathbb{D}\left[q(\boldsymbol{u}_i) || p(\boldsymbol{u}_i)\right]$$

Inference amounts to evaluating the above bound (and applying gradient ascent), as follows:

Optimising the Deep GP Bound

$$\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} = \sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d}f_{L,n} - \sum_{i=1}^{L} \mathbb{D}\left[q(\boldsymbol{u}_i) || p(\boldsymbol{u}_i)\right]$$

Inference amounts to evaluating the above bound (and applying gradient ascent), as follows:

- Recursively draw S samples, $\tilde{f}_{i,n,s}$, from each layer, treating samples from the previous layer as deterministic inputs. Do this for all $n = 1, \ldots, N_*$ in the mini-batch.

Optimising the Deep GP Bound

$$\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} = \sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d}f_{L,n} - \sum_{i=1}^{L} \mathbb{D}\left[q(\boldsymbol{u}_i) || p(\boldsymbol{u}_i)\right]$$

Inference amounts to evaluating the above bound (and applying gradient ascent), as follows:

- Recursively draw S samples, $\tilde{f}_{i,n,s}$, from each layer, treating samples from the previous layer as deterministic inputs. Do this for all $n = 1, \ldots, N_*$ in the mini-batch.
- At the final layer, predict the GP mean, $m_{L,n,s}$, and covariance, $C_{L,n,s}$, using $\tilde{f}_{L-1,n,s}$ as inputs.

Optimising the Deep GP Bound

$$\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} = \sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d}f_{L,n} - \sum_{i=1}^{L} \mathbb{D}\left[q(\boldsymbol{u}_i) || p(\boldsymbol{u}_i)\right]$$

Inference amounts to evaluating the above bound (and applying gradient ascent), as follows:

- Recursively draw S samples, $\tilde{f}_{i,n,s}$, from each layer, treating samples from the previous layer as deterministic inputs. Do this for all $n = 1, \ldots, N_*$ in the mini-batch.
- At the final layer, predict the GP mean, $m_{L,n,s}$, and covariance, $C_{L,n,s}$, using $\tilde{f}_{L-1,n,s}$ as inputs.
- Approximate the first term in the ELBO by averaging across the samples, *i.e.*,:

$$\sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d}f_{L,n} \approx \frac{1}{S} \frac{N}{N_*} \sum_{s=1}^{S} \sum_{n=1}^{N_*} \int \mathcal{N}(f_{L,n} \mid m_{L,n,s}, \, C_{L,n,s}) \ln p(y_n | f_{L,n}) \, \mathrm{d}f_{L,n}$$

Optimising the Deep GP Bound

$$\ln p(\mathbf{y}) \geq \mathcal{L}_{DGP} = \sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d} f_{L,n} - \sum_{i=1}^{L} \mathbb{D} \left[q(\boldsymbol{u}_i) || p(\boldsymbol{u}_i) \right]$$

Inference amounts to evaluating the above bound (and applying gradient ascent), as follows:

- Recursively draw S samples, $\tilde{f}_{i,n,s}$, from each layer, treating samples from the previous layer as deterministic inputs. Do this for all $n = 1, \ldots, N_*$ in the mini-batch.
- At the final layer, predict the GP mean, $m_{L,n,s}$, and covariance, $C_{L,n,s}$, using $\tilde{f}_{L-1,n,s}$ as inputs.
- Approximate the first term in the ELBO by averaging across the samples, *i.e.*,:

$$\sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d} f_{L,n} \approx \frac{1}{S} \frac{N}{N_*} \sum_{s=1}^{S} \sum_{n=1}^{N_*} \int \mathcal{N}(f_{L,n} \mid m_{L,n,s}, \, C_{L,n,s}) \ln p(y_n | f_{L,n}) \, \mathrm{d} f_{L,n}$$

- For the second term, compute the KL divergence between $q(\boldsymbol{u}_i)$ and $p(\boldsymbol{u}_i)$ in each layer separately (this is available in closed form since both terms are Gaussian).

Optimising the Deep GP Bound

$$\ln p(\boldsymbol{y}) \geq \mathcal{L}_{DGP} = \sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d} f_{L,n} - \sum_{i=1}^{L} \mathbb{D}\left[q(\boldsymbol{u}_i) || p(\boldsymbol{u}_i)\right]$$

Inference amounts to evaluating the above bound (and applying gradient ascent), as follows:

- Recursively draw S samples, $\tilde{f}_{i,n,s}$, from each layer, treating samples from the previous layer as deterministic inputs. Do this for all $n = 1, \ldots, N_*$ in the mini-batch.
- At the final layer, predict the GP mean, $m_{L,n,s}$, and covariance, $C_{L,n,s}$, using $\tilde{f}_{L-1,n,s}$ as inputs.
- Approximate the first term in the ELBO by averaging across the samples, *i.e.*,:

$$\sum_{n=1}^{N} \int q(f_{L,n}) \ln p(y_n | f_{L,n}) \, \mathrm{d} f_{L,n} \approx \frac{1}{S} \frac{N}{N_*} \sum_{s=1}^{S} \sum_{n=1}^{N_*} \int \mathcal{N}(f_{L,n} \mid m_{L,n,s}, \, C_{L,n,s}) \ln p(y_n | f_{L,n}) \, \mathrm{d} f_{L,n}$$

- For the second term, compute the KL divergence between $q(\boldsymbol{u}_i)$ and $p(\boldsymbol{u}_i)$ in each layer separately (this is available in closed form since both terms are Gaussian).
- This inference technique is called **doubly stochastic VI**, due to the two sources of stochasticity.

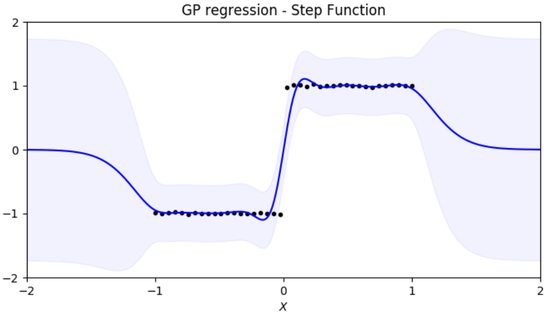
Alternative Approaches

Other approaches to deep GP inference exist, but we won't go over them here:

- **Deep GP Expectation Propagation** - similar to the above, but using EP for inference, and replacing the sampling procedure with Gaussian projections to approximate the marginals.
- **Importance-weighted VI with latent variables** - introduces additional latent variables which allow the model to represent non-Gaussian posteriors.
- **Hamiltonian Monte Carlo** - uses a sophisticated sampling approach to represent non-Gaussian posteriors.

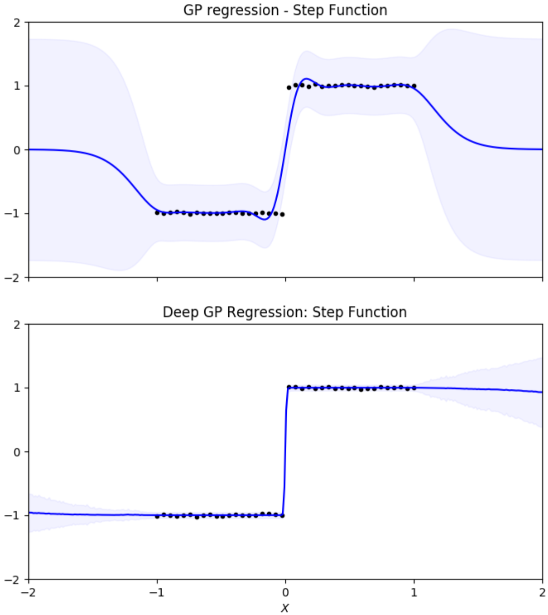
Deep GP Performance

- **Discontinuities / jumps:**



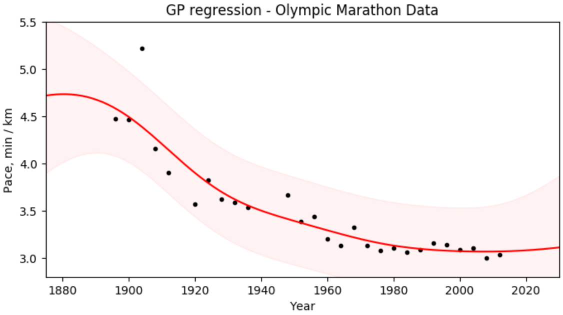
Deep GP Performance

- **Discontinuities / jumps:**
- The deep GP captures the jump, whilst the variance elsewhere remains low.
- However, we would prefer that the variances increases in the region of the discontinuity.
- In the exercises, you will examine what happens in each layer.



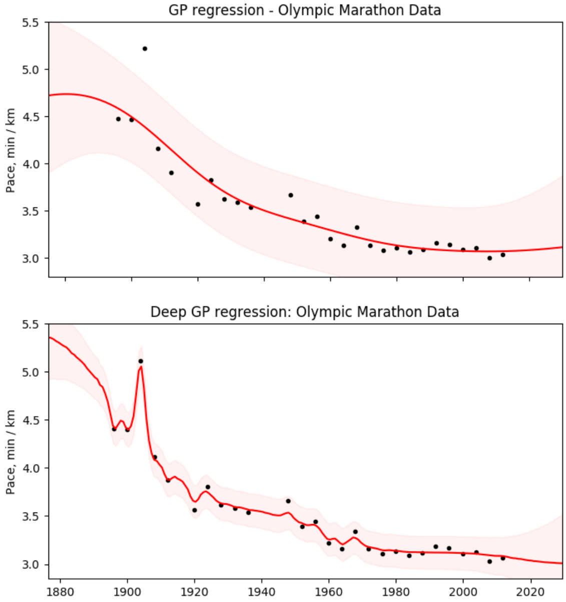
Deep GP Performance

- **Outliers:**



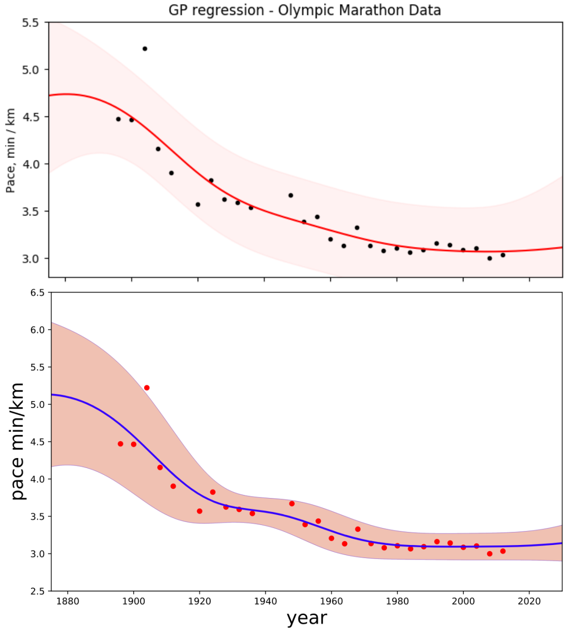
Deep GP Performance

- **Outliers:**
- The deep GP seems to overfit the outlier.

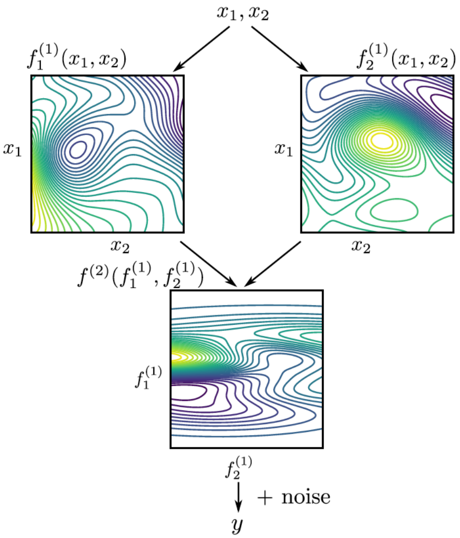


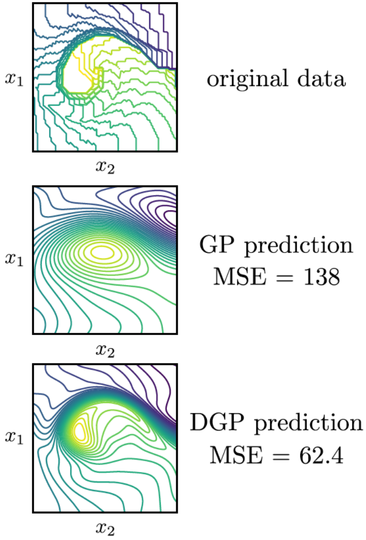
Deep GP Performance

- **Outliers:**
- The deep GP seems to overfit the outlier.
- Whereas the originally proposed deep GP methods claim to solve these tasks well.
- But doubly stochastic VI reports superior performance on many machine learning tasks, potentially because it scales to large data.



Deep GP Performance





Issues with Deep GPs

- As with many deep learning approaches, things become less stable as the depth is increased:

Issues with Deep GPs

- As with many deep learning approaches, things become less stable as the depth is increased:
- **Deep GPs are much more sensitive to initialisation** than standard GPs (in both the hyperparameters and the inducing point locations).

Issues with Deep GPs

- As with many deep learning approaches, things become less stable as the depth is increased:
- **Deep GPs are much more sensitive to initialisation** than standard GPs (in both the hyperparameters and the inducing point locations).
- **Training can be slow**: we trade off the number of samples with accuracy.

Issues with Deep GPs

- As with many deep learning approaches, things become less stable as the depth is increased:
- **Deep GPs are much more sensitive to initialisation** than standard GPs (in both the hyperparameters and the inducing point locations).
- **Training can be slow**: we trade off the number of samples with accuracy.
- **Training is more prone to getting stuck in local minima** since there are many more parameters to optimise.

Issues with Deep GPs

- As with many deep learning approaches, things become less stable as the depth is increased:
- **Deep GPs are much more sensitive to initialisation** than standard GPs (in both the hyperparameters and the inducing point locations).
- **Training can be slow**: we trade off the number of samples with accuracy.
- **Training is more prone to getting stuck in local minima** since there are many more parameters to optimise.
- **Current approaches to VI tend to "turn off" layers**, or reduce their variance to near-zero (such that they behave like deterministic mappings).

Deep GP Performance

- Deep GPs have been shown to have excellent performance on many medium-large machine learning tasks.

Deep GP Performance

- Deep GPs have been shown to have excellent performance on many medium-large machine learning tasks.
- They have been combined with convolutional kernels (as presented in the previous lecture) to produce state-of-the-art results on image classification.
- Performance matches *e.g.*, deep CNNs, but improves uncertainty quantification in predictions, *i.e.*, **the model is more aware when it is wrong.**

Deep GP Performance

- Deep GPs have been shown to have excellent performance on many medium-large machine learning tasks.
- They have been combined with convolutional kernels (as presented in the previous lecture) to produce state-of-the-art results on image classification.
- Performance matches *e.g.*, deep CNNs, but improves uncertainty quantification in predictions, *i.e.*, **the model is more aware when it is wrong.**
- So deep GPs have great potential. But, as we have seen, there is still much work to be done.

End of Today's Lecture

- Next time: Aki Vehtari will give a lecture about model selection
- Now time for questions. Next week's assignment (#5) will include sampling from and training a deep GP.