Gaussian process latent variable models and deep Gaussian processes

Mauricio A. Álvarez

Modelos probabilísticos profundos
AIR Institute

Contents

A comment on notation

- Let $\mathbf{X} \in \mathbb{R}^{N \times Q}$.
- We will use the notation $\mathbf{x}_{i,:}$ to refer to the row vector i in $\mathbf{X}$ and $\mathbf{x}_{:,j}$ to the column vector j in $\mathbf{X}$.

Contents

Contents

GPLVM

- The Gaussian process latent variable model (GPLVM) is an example of the use of GPs for unsupervised learning.
- The GPLVM is a probabilistic model for non-linear dimensionality reduction.
- It uses Gaussian processes as priors over the function that maps the latent space to the observed data.
- There are many variants of GPLVM.

PPCA (I)

- A way to motivate the GPLVM is by seeing it as a dual formulation for the probabilistic PCA (PPCA) model.

- In PPCA, the observed output vector $\mathbf{y}$ is represented as a linear transformation from a lower dimensional vector in a latent space, plus Gaussian noise

$$\mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \boldsymbol{\eta}_n,$$

where $\mathbf{y}_n \in \mathbb{R}^{D\times 1}$ is part of $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_N]^\top \in \mathbb{R}^{N\times D}$, $\mathbf{W} \in \mathbb{R}^{D\times q}$ and $\mathbf{x}_n \in \mathbb{R}^{q\times 1}$.

- The matrix $\mathbf{Y}$ is assumed to be centered.

- The variables $\boldsymbol{\eta}_n$ are sampled from

$$p(\boldsymbol{\eta}_n) = \mathcal{N}\left(\boldsymbol{\eta}_n \mid \mathbf{0}, \beta^{-1}\mathbf{I}\right)$$

PPCA (II)

- The likelihood for a data point can then be written as

$$p\left(\mathbf{y}_n \mid \mathbf{x}_n, \mathbf{W}, \beta\right) = \mathcal{N}\left(\mathbf{y}_n \mid \mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I}\right).$$

- For PPCA, it is assumed that $p\left(\mathbf{x}_n\right) = \mathcal{N}\left(\mathbf{x}_n \mid \mathbf{0}, \mathbf{I}\right)$.
- To obtain the marginal likelihood we integrate over the latent variables

$$p\left(\mathbf{y}_n \mid \mathbf{W}, \beta\right) = \int p\left(\mathbf{y}_n \mid \mathbf{x}_n, \mathbf{W}, \beta\right) p\left(\mathbf{x}_n\right) d\mathbf{x}_n = \mathcal{N}\left(\mathbf{y}_n \mid \mathbf{0}, \mathbf{W}\mathbf{W}^{\top} + \beta^{-1}\mathbf{I}\right).$$

- The likelihood of the full dataset is given as

$$p(\mathbf{Y} \mid \mathbf{W}, \beta) = \prod_{n=1}^{N} p\left(\mathbf{y}_n \mid \mathbf{W}, \beta\right).$$

- The parameters $\mathbf{W}, \beta$ are found by maximising the marginal likelihood.

Dual PPCA

- Instead of marginalising $\mathbf{x}_n$, dual PPCA marginalises $\mathbf{w}_{i,:}$ using the prior

$$p(\mathbf{W}) = \prod_{i=1}^{D} \mathcal{N}\left(\mathbf{w}_{i,:} \mid \mathbf{0}, \mathbf{I}\right).$$

- The resulting marginalised likelihood takes the form

$$p(\mathbf{Y} \mid \mathbf{X}, \beta) = \prod_{d=1}^{D} p\left(\mathbf{y}_{:,d} \mid \mathbf{X}, \beta\right) = \prod_{d=1}^{D} \mathcal{N}\left(\mathbf{y}_{:,d} \mid \mathbf{0}, \mathbf{X}\mathbf{X}^{\top} + \beta^{-1}\mathbf{I}\right),$$

  where $\mathbf{y}_{:,d}$ is a column of $\mathbf{Y}$.

- Here it is where the GP prior comes into play.
- One can assume that the matrix $\mathbf{X}\mathbf{X}^{\top} + \beta^{-1}\mathbf{I}$ is computed using the kernel function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^{\top}\mathbf{x}_j + \beta^{-1}\mathbf{I}.$$

GPLVM

- If we use a general kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$, we can write

$$p(\mathbf{Y} \mid \mathbf{X}, \beta) = \prod_{d=1}^{D} p\left(\mathbf{y}_{:,d} \mid \mathbf{X}, \beta\right) = \prod_{d=1}^{D} \mathcal{N}\left(\mathbf{y}_{:,d} \mid \mathbf{0}, \mathbf{K} + \beta^{-1}\mathbf{I}\right).$$

- We can get to the same expression above by assuming the following likelihood function

$$p(\mathbf{Y} \mid \mathbf{F}, \beta) = \prod_{d=1}^{D} p\left(\mathbf{y}_{:,d} \mid \mathbf{f}_{:,d}, \beta\right) = \prod_{d=1}^{D} \mathcal{N}\left(\mathbf{y}_{:,d} \mid \mathbf{f}_{:,d}, \beta^{-1}\mathbf{I}\right),$$

and a GP prior for $\mathbf{f}_{:,d} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$.

Latent space $\mathbf{X}$

- The latent space can be found by optimising the marginal log-likelihood with respect to $\mathbf{X}$,

$$L = -\frac{DN}{2}\ln 2\pi - \frac{D}{2}\ln|\mathbf{K}| - \frac{1}{2}\operatorname{tr}\left[(\mathbf{K}+\beta^{-1}\mathbf{I})^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right].$$

- We compute the derivative $\frac{\partial L}{\partial \mathbf{K}}$ and combine it with $\frac{\partial \mathbf{K}}{\partial x_{n,j}}$ using the chain rule.

- The parameter Q is set to 2 for visualisation purposes or it is found using the reconstruction error on a validation set.

- Many different techniques can be used to scale the GPLVM, including methods based on inducing inputs.

Contents

A prior over the latent space $\mathbf{X}$

- An alternative way to make inference over $\mathbf{X}$ is by assigning a prior to it, $p(\mathbf{X})$ and then finding the MAP estimate

$$\mathbf{X}_{\text{MAP}} = \arg\max_{\mathbf{X}} p(\mathbf{Y} \mid \mathbf{X}) p(\mathbf{X}),$$

where

$$p(\mathbf{Y} \mid \mathbf{X}, \beta) = \prod_{d=1}^{D} \mathcal{N}\left(\mathbf{y}_{:,d} \mid \mathbf{0}, \mathbf{K} + \beta^{-1}\mathbf{I}\right).$$

- Both the ML and MAP approaches to estimation have several drawbacks:
  - Since it does not marginalise out the latent inputs, it could be sensitive to overfitting.
  - These point objective functions cannot provide any insight for selecting the optimal number of latent dimensions.

Bayesian inference for the GPLVM

- The joint model for the GPLVM with a prior for $\mathbf{X}$ follows as

$$p\left(\mathbf{Y}, \mathbf{F}, \mathbf{X} \mid \boldsymbol{\theta}_f, \boldsymbol{\theta}_x, \sigma^2\right) = p\left(\mathbf{Y} \mid \mathbf{F}, \sigma^2\right) p\left(\mathbf{F} \mid \mathbf{X}, \boldsymbol{\theta}_f\right) p\left(\mathbf{X} \mid \boldsymbol{\theta}_x\right)$$
$$= \prod_{d=1}^{D} p\left(\mathbf{y}_{:,d} \mid \mathbf{f}_{:,d}, \sigma^2\right) p\left(\mathbf{f}_{:,d} \mid \mathbf{X}, \boldsymbol{\theta}_f\right) p\left(\mathbf{X} \mid \boldsymbol{\theta}_x\right),$$

  where $\sigma^2 = \beta^{-1}$, $\boldsymbol{\theta}_f$ are the hyperparameters of the kernel function, and $\boldsymbol{\theta}_x$ are the hyperparameters of the prior $p(\mathbf{X})$.

- Besides marginalising $\mathbf{F}$ by using GP priors over $\mathbf{f}_{:,d}$, in a Bayesian framework, we would like to marginalise the input space $\mathbf{X}$.

- We can use variational inference to approximately marginalise $\mathbf{X}$.

A tractable lower bound for the marginal likelihood (I)

- Titsias and Lawrence (2010) introduced inducing variables to write a tractable lower bound for the marginal likelihood in the GPLVM.
- The joint model is augmented with a new set of variables $\mathbf{U}$ and we write

$$p\left(\mathbf{Y}, \mathbf{F}, \mathbf{U}, \mathbf{X} \mid \mathbf{X}_u\right) = p(\mathbf{Y} \mid \mathbf{F}) p\left(\mathbf{F} \mid \mathbf{U}, \mathbf{X}, \mathbf{X}_u\right) p\left(\mathbf{U} \mid \mathbf{X}_u\right) p(\mathbf{X})$$
$$= \left(\prod_{d=1}^{D} p\left(\mathbf{y}_{:,d} \mid \mathbf{f}_{:,d}\right) p\left(\mathbf{f}_{:,d} \mid \mathbf{u}_{:,d}, \mathbf{X}, \mathbf{X}_u\right) p\left(\mathbf{u}_{:,d} \mid \mathbf{X}_u\right)\right) p(\mathbf{X}),$$

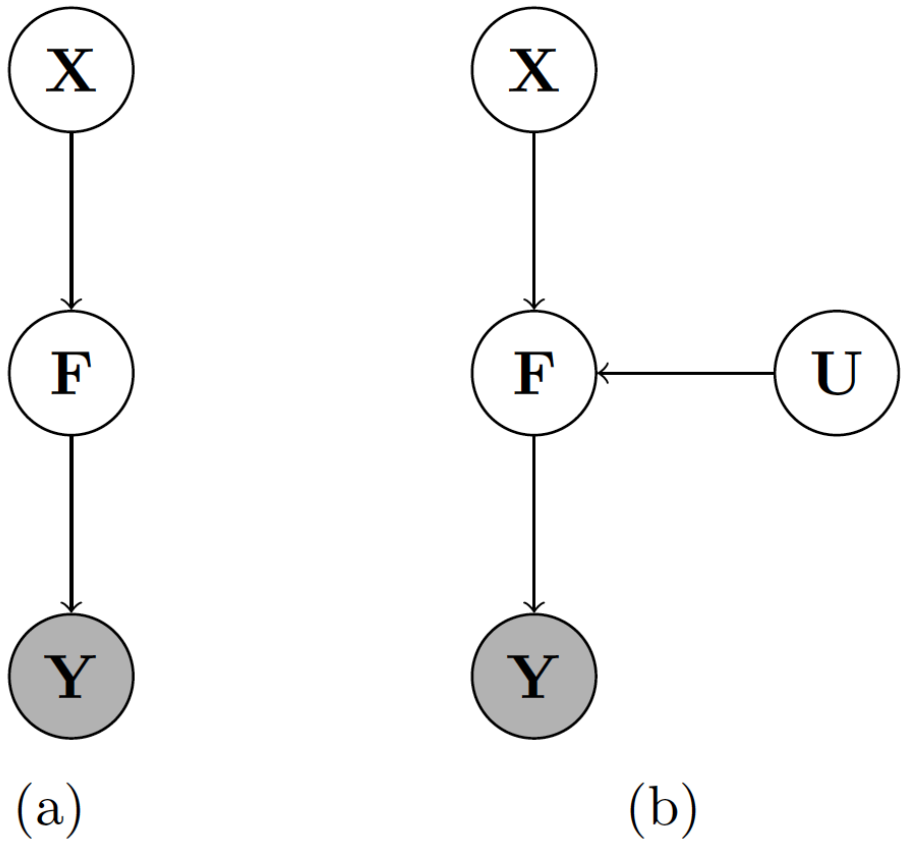
  where $\mathbf{X}_u$ are the inducing inputs.
- The conditional distributions $p\left(\mathbf{f}_{:,d} \mid \mathbf{u}_{:,d}, \mathbf{X}, \mathbf{X}_u\right)$ follow

$$p\left(\mathbf{f}_{:,d} \mid \mathbf{u}_{:,d}, \mathbf{X}, \mathbf{X}_u\right) = \mathcal{N}\left(\mathbf{f}_{:,d} \mid \mathbf{a}_d, \Sigma_f\right)$$

  with

$$\mathbf{a}_d = \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u}_{:,d}$$
$$\Sigma_f = \mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}.$$

A tractable lower bound for the marginal likelihood (II)



In (a), a graphical model for the GPLVM. In (b), a graphical model for the augmented GPLVM.

A tractable lower bound for the marginal likelihood (III)

- In what follows, we omit $\mathbf{X}_u$, which are treated as variational parameters from now on.

- The true posterior factorizes as

$$p(\mathbf{F}, \mathbf{U}, \mathbf{X} \mid \mathbf{Y}) = p(\mathbf{F} \mid \mathbf{U}, \mathbf{Y}, \mathbf{X}) p(\mathbf{U} \mid \mathbf{Y}, \mathbf{X}) p(\mathbf{X} \mid \mathbf{Y}).$$

- The posterior is approximated with the distribution

$$q(\mathbf{F}, \mathbf{U}, \mathbf{X}) = p(\mathbf{F} \mid \mathbf{U}, \mathbf{X}) q(\mathbf{U}) q(\mathbf{X}) = \left(\prod_{d=1}^{D} p\left(\mathbf{f}_{:,d} \mid \mathbf{u}_{:,d}, \mathbf{X}\right) q\left(\mathbf{u}_{:,d}\right) \right) q(\mathbf{X})$$

- It is assumed that $q(\mathbf{X})$ follows a Gaussian distribution and $q(\mathbf{U})$ is an unrestricted variational distribution.

A tractable lower bound for the marginal likelihood (IV)

- Following the variation inference framework, the log-marginal likelihood can be expressed as

$$\log p(\mathbf{Y}) \geq \mathcal{L}(q(\mathbf{X}), q(\mathbf{U})) = \int q(\mathbf{F}, \mathbf{U}, \mathbf{X}) \log \frac{p(\mathbf{Y}, \mathbf{F}, \mathbf{U}, \mathbf{X})}{q(\mathbf{F}, \mathbf{U}, \mathbf{X})} \mathrm{d}\mathbf{X}\mathrm{d}\mathbf{F}\mathrm{d}\mathbf{U}$$

- Replacing the expression for the joint likelihood and the approximated posterior in the bound, we get

$$\begin{aligned}
\mathcal{L}(q(\mathbf{X}), q(\mathbf{U})) &= \int \prod_{d=1}^{D} p\left(\mathbf{f}_{:,d} | \mathbf{u}_{:,d}, \mathbf{X}\right) q\left(\mathbf{u}_{:,d}\right) q(\mathbf{X}) \log \frac{\prod_{d=1}^{D} p\left(\mathbf{y}_{:,d} | \mathbf{f}_{:,d}\right) p\left(\mathbf{u}_{:,d}\right)}{\prod_{d=1}^{D} q\left(\mathbf{u}_{:,d}\right)} \mathrm{d}\mathbf{X}\mathrm{d}\mathbf{F}\mathrm{d}\mathbf{U} \\
&\quad - \int q(\mathbf{X}) \log \frac{q(\mathbf{X})}{p(\mathbf{X})} \mathrm{d}\mathbf{X} \\
&= \hat{\mathcal{L}}(q(\mathbf{X}), q(\mathbf{U})) - \mathrm{KL}(q(\mathbf{X}) \| p(\mathbf{X})),
\end{aligned}$$

where

$$\hat{\mathcal{L}}(q(\mathbf{X}), q(\mathbf{U})) = \int \prod_{d=1}^{D} p\left(\mathbf{f}_{:,d} | \mathbf{u}_{:,d}, \mathbf{X}\right) q\left(\mathbf{u}_{:,d}\right) q(\mathbf{X}) \log \frac{\prod_{d=1}^{D} p\left(\mathbf{y}_{:,d} | \mathbf{f}_{:,d}\right) p\left(\mathbf{u}_{:,d}\right)}{\prod_{d=1}^{D} q\left(\mathbf{u}_{:,d}\right)} \mathrm{d}\mathbf{X}\mathrm{d}\mathbf{F}\mathrm{d}\mathbf{U}$$

$\hat{\mathcal{L}}(q(\mathbf{X}), q(\mathbf{U}))$ in $\mathcal{L}(q(\mathbf{X}), q(\mathbf{U}))$

- The term $\hat{\mathcal{L}}(q(\mathbf{X}), q(\mathbf{U}))$ in $\mathcal{L}(q(\mathbf{X}), q(\mathbf{U}))$ can be written as

$$\begin{aligned}
\hat{\mathcal{L}}(q(\mathbf{X}), q(\mathbf{U})) &= \int \prod_{d=1}^{D} p\left(\mathbf{f}_{:,d} | \mathbf{u}_{:,d}, \mathbf{X}\right) q\left(\mathbf{u}_{:,d}\right) q(\mathbf{X}) \log \frac{\prod_{d=1}^{D} p\left(\mathbf{y}_{:,d} | \mathbf{f}_{:,d}\right) p\left(\mathbf{u}_{:,d}\right)}{\prod_{d=1}^{D} q\left(\mathbf{u}_{:,d}\right)} \mathrm{d}\mathbf{X}\mathrm{d}\mathbf{F}\mathrm{d}\mathbf{U} \\
&= \int \prod_{j=1}^{D} p\left(\mathbf{f}_{:,j} | \mathbf{u}_{:,j}, \mathbf{X}\right) q\left(\mathbf{u}_{:,j}\right) q(\mathbf{X}) \sum_{d=1}^{D} \log \frac{p\left(\mathbf{y}_{:,d} | \mathbf{f}_{:,d}\right) p\left(\mathbf{u}_{:,d}\right)}{q\left(\mathbf{u}_{:,d}\right)} \mathrm{d}\mathbf{X}\mathrm{d}\mathbf{F}\mathrm{d}\mathbf{U} \\
&= \sum_{d=1}^{D} \int p\left(\mathbf{f}_{:,d} | \mathbf{u}_{:,d}, \mathbf{X}\right) q\left(\mathbf{u}_{:,d}\right) q(\mathbf{X}) \log \frac{p\left(\mathbf{y}_{:,d} | \mathbf{f}_{:,d}\right) p\left(\mathbf{u}_{:,d}\right)}{q\left(\mathbf{u}_{:,d}\right)} \mathrm{d}\mathbf{X}\mathrm{d}\mathbf{f}_{:,d}\mathrm{d}\mathbf{u}_{:,d} \\
&= \sum_{d=1}^{D} \hat{\mathcal{L}}_d(q(\mathbf{X}), q(\mathbf{u}_{:,d})).
\end{aligned}$$

- In turn, each term $\hat{\mathcal{L}}_d(q(\mathbf{X}), q(\mathbf{U}))$ can be written as

$$\begin{aligned}
\hat{\mathcal{L}}_d(q(\mathbf{X}), q(\mathbf{u}_{:,d})) &= \int p\left(\mathbf{f}_{:,d} | \mathbf{u}_{:,d}, \mathbf{X}\right) q\left(\mathbf{u}_{:,d}\right) q(\mathbf{X}) \log \frac{p\left(\mathbf{y}_{:,d} | \mathbf{f}_{:,d}\right) p\left(\mathbf{u}_{:,d}\right)}{q\left(\mathbf{u}_{:,d}\right)} \mathrm{d}\mathbf{X}\mathrm{d}\mathbf{f}_{:,d}\mathrm{d}\mathbf{u}_{:,d} \\
&= \int q\left(\mathbf{u}_{:,d}\right) \left(\left\langle \log p\left(\mathbf{y}_{:,d} \mid \mathbf{f}_{:,d}\right) \right\rangle_{p\left(\mathbf{f}_{:,d} | \mathbf{u}_{:,d}, \mathbf{x}\right) q(\mathbf{x})} + \log \frac{p\left(\mathbf{u}_{:,d}\right)}{q\left(\mathbf{u}_{:,d}\right)} \right) \mathrm{d}\mathbf{u}_{:,d}
\end{aligned}$$

$\left\langle \log p\left(\mathbf{y}_{:,d} \mid \mathbf{f}_{:,d}\right)\right\rangle_{p(\mathbf{f}_{:,d}|\mathbf{u}_{:,d},\mathbf{X})q(\mathbf{X})}$ in $\hat{\mathcal{L}}(q(\mathbf{X}), q(\mathbf{u}_{:,d}))$

- The term $\left\langle \log p\left(\mathbf{y}_{:,d} \mid \mathbf{f}_{:,d}\right)\right\rangle_{p\left(\mathbf{f}_{:,d}|\mathbf{u}_{:,d},\mathbf{X}\right)}$ can be written as

$$\log \mathcal{N}\left(\mathbf{y}_{:,d} \mid \mathbf{a}_d, \sigma^2\mathbf{I}\right) - \frac{1}{2\sigma^2}\operatorname{tr}\left(\mathbf{K}_{ff}\right) + \frac{1}{2\sigma^2}\operatorname{tr}\left(\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\mathbf{K}_{fu}\right)$$

- This means that the bound $\hat{\mathcal{L}}_d(q(\mathbf{X}), q(\mathbf{u}_{:,d}))$ can be written as

$$\hat{\mathcal{L}}_d\left(q(\mathbf{X}), q\left(\mathbf{u}_{:,d}\right)\right) = \int q\left(\mathbf{u}_{:,d}\right) \log \frac{e^{\left\langle \log \mathcal{N}\left(\mathbf{y}_{:,d}|\mathbf{a}_d,\sigma^2\mathbf{I}\right)\right\rangle_{q(\mathbf{X})}} p\left(\mathbf{u}_{:,d}\right)}{q\left(\mathbf{u}_{:,d}\right)} \mathrm{d}\mathbf{u}_{:,d} - \mathcal{A}$$

where

$$\mathcal{A} = \frac{1}{2\sigma^2}\operatorname{tr}\left(\left\langle \mathbf{K}_{ff}\right\rangle_{q(\mathbf{X})}\right) - \frac{1}{2\sigma^2}\operatorname{tr}\left(\mathbf{K}_{uu}^{-1}\left\langle \mathbf{K}_{uf}\mathbf{K}_{fu}\right\rangle_{q(\mathbf{X})}\right).$$

Reversing Jensen's inequality (I)

- Using a similar idea to Titsias (2009), we can reverse Jensen's inequality to obtain

$$\hat{\mathcal{L}}_d\left(q(\mathbf{X})\right) = \log \int e^{\left\langle \log \mathcal{N}\left(\mathbf{y}_{:,d} | \mathbf{a}_d, \sigma^2 \mathbf{I}\right)\right\rangle_{q(\mathbf{X})}} p\left(\mathbf{u}_{:,d}\right) \mathrm{d}\mathbf{u}_{:,d} - \mathcal{A}.$$

- Furthermore, assuming that $q(\mathbf{X})$ factorizes across the instances, $q(\mathbf{X}) = \prod q(\mathbf{x}_{n,:})$, $\hat{\mathcal{L}}_d\left(q(\mathbf{X})\right)$ follows as

$$\log \int \prod_{n=1}^{N} e^{\left\langle \log \mathcal{N}\left(y_{n,d} | a_{n,d}, \sigma^2\right) - \frac{1}{2\sigma^2}\left(k_f(\mathbf{x}_{n,:}, \mathbf{x}_{n,:}) - k_f(\mathbf{x}_{n,:}, \mathbf{X}_u) \mathbf{K}_{uu}^{-1} k_f(\mathbf{X}_u, \mathbf{x}_{n,:})\right)\right\rangle_{q(\mathbf{x}_{n,:})}} p\left(\mathbf{u}_{:,d}\right) \mathrm{d}\mathbf{u}_{:,d}$$

Reversing Jensen's inequality (II)

- It can further be simplified as

$$\hat{\mathcal{L}}_d(q(\mathbf{X})) = \log\left[\frac{\sigma^{-n}\left|\mathbf{K}_{uu}\right|^{\frac{1}{2}}}{(2\pi)^{\frac{n}{2}}\left|\sigma^{-2}\boldsymbol{\Psi}_2 + \mathbf{K}_{uu}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{y}_{:,d}^\top \mathbf{W}\mathbf{y}_{:,d}}\right] - \frac{\psi_0}{2\sigma^2} + \frac{1}{2\sigma^2}\operatorname{tr}\left(\mathbf{K}_{uu}^{-1}\boldsymbol{\Psi}_2\right)$$

  where

$$\psi_0 = \operatorname{tr}\left(\left\langle \mathbf{K}_{ff}\right\rangle_{q(\mathbf{X})}\right), \boldsymbol{\Psi}_1 = \left\langle \mathbf{K}_{fu}\right\rangle_{q(\mathbf{X})}, \quad \boldsymbol{\Psi}_2 = \left\langle \mathbf{K}_{uf}\mathbf{K}_{fu}\right\rangle_{q(\mathbf{X})}$$

  are referred to as the Ψ statistics and

$$\mathbf{W} = \sigma^{-2}\mathbf{I}_n - \sigma^{-4}\boldsymbol{\Psi}_1\left(\sigma^{-2}\boldsymbol{\Psi}_2 + \mathbf{K}_{uu}\right)^{-1}\boldsymbol{\Psi}_1^\top.$$

- The Ψ statistics can be computed in closed form for a particular class of kernel functions.

Inference for the posterior and the hyperparameters

- The posterior over the latent space, $q(\mathbf{X})$, can have different forms.
- One that is easy to implement, is to assume $q(\mathbf{x}_{n,:}) = \mathcal{N}(\mathbf{x}_{n,:}|\boldsymbol{\mu}_{n,:}, \mathbf{S}_n)$.
- We use the lower bound

$$\sum_{d=1}^{D} \hat{\mathcal{L}}_d(q(\mathbf{X})) - \mathrm{KL}(q(\mathbf{X})\|p(\mathbf{X}))$$

  to compute the parameters of the posterior distributions $q(\mathbf{x}_{n,:})$ and the hyperparameters $\boldsymbol{\theta}_x, \boldsymbol{\theta}_f, \sigma^2$ using numerical optimisation.

- Similarly to the "collapsed" variational inference approach to sparse GPs (Titsias, 2009), we can compute the posterior distribution for $q(\mathbf{u}_{:,d})$ in closed form.

Prediction

- The Bayesian GPLVM can be used for different types of predictions.
- It can be used to compute the probability density $p(\mathbf{Y}_* \mid \mathbf{Y})$.
- Let $\mathbf{Y}_* = (\mathbf{Y}_*^u, \mathbf{Y}_*^o)$, being $\mathbf{Y}_*^u$ the unobserved part of $\mathbf{Y}_*$ and $\mathbf{Y}_*^o$ the observed part.
- It can also be used to compute $p(\mathbf{Y}_*^u \mid \mathbf{Y}_*^o, \mathbf{Y})$, where $\mathbf{Y}_*^u$.

Example: Oil flow data

- The multiphase oil flow data (Bishop and James, 1993) consists of 1000, 12 dimensional observations belonging to three known classes corresponding to different phases of oil flow.

- Titsias and Lawrence (2010) shows the result of applying GPLVM and Bayesian GPLVM for dimensionality reduction in this dataset.

Example: Oil flow data results

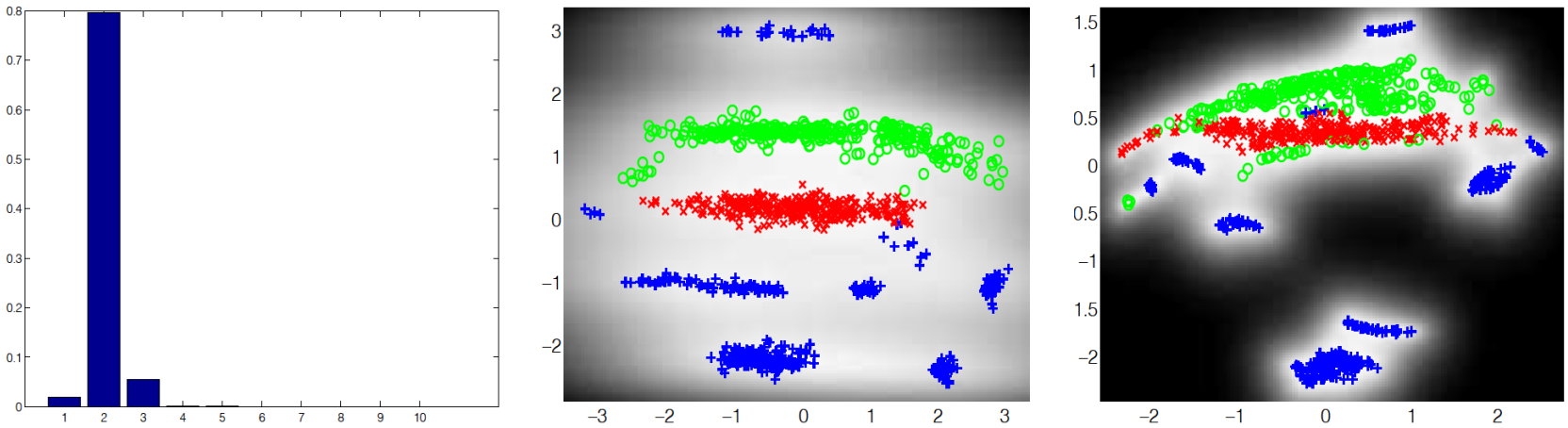


Figure 1: Panel (a) shows the inverse lengthscales found by applying the Bayesian GP-LVM with ARD SE kernel on the oil flow data. Panel (b) shows the visualization achieved by keeping the most dominant latent dimensions (2 and 3) which have the largest inverse lengthscale value. Dimension 2 is plotted on the y-axis and 3 and on the x-axis. Plot (c) shows the visualization found by standard sparse GP-LVM.

Example: Frey faces data

- Titsias and Lawrence (2010) also considers an example of a dataset of faces (Roweis et al., 2002) taken from a video sequence that consists of 1965 images of size 28×20.

- The model is trained using a random selection of 1000 images and then considering the remaining 965 images as test data.

- For each test image, it is assumed that only half of the image pixels are observed. The missing pixels were chosen randomly for each test image.

Example: Frey faces data results

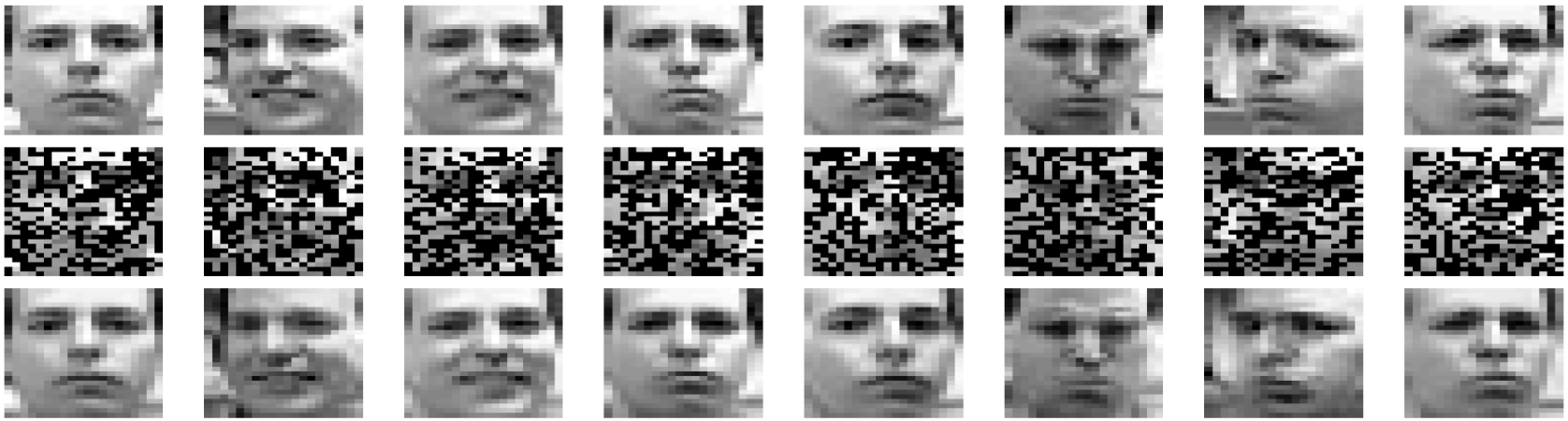
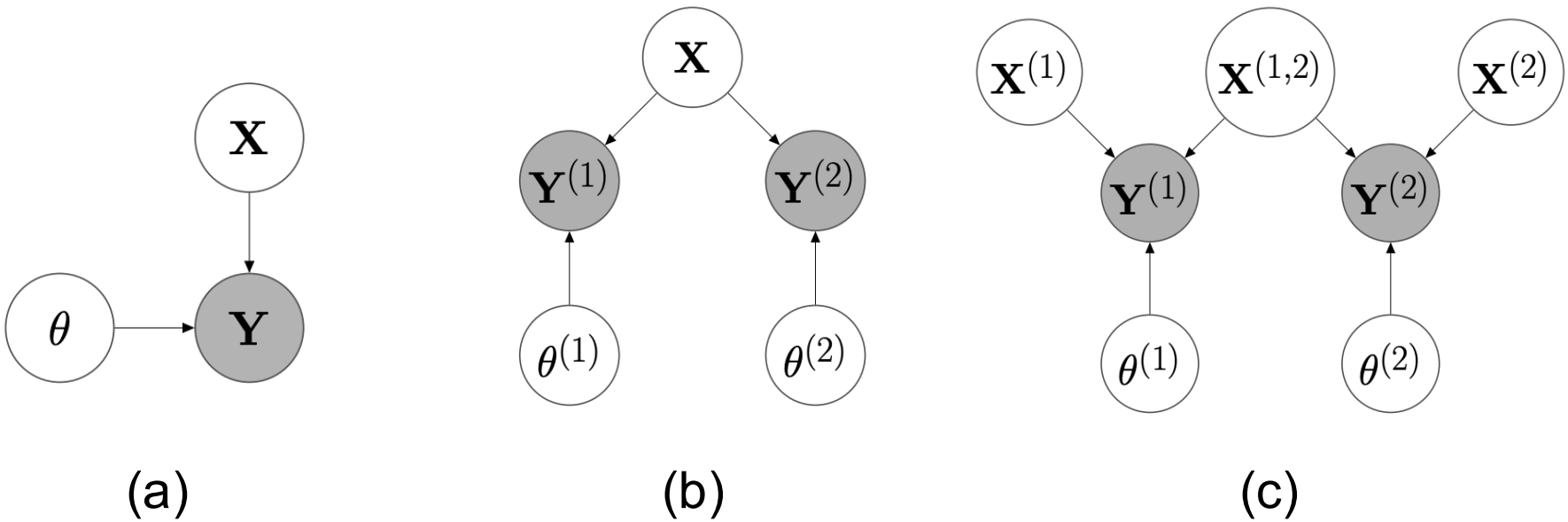


Figure 2: Examples of reconstruction of partially observed test images in Frey faces by applying the Bayesian GP-LVM. Each column corresponds to a test image. In every column, the top panel shows the true test image, the middle panel the partially observed image (where missing pixels are shown in black) and the bottom image is the reconstructed image.

Contents

Extensions (I)



(a) (b) (c)

In (a), the GPLVM. In (b), a version for multi-view learning proposed by Shon et al. (2006). In (c), a version with a factorised latent space, proposed by Ek et al. (2008).

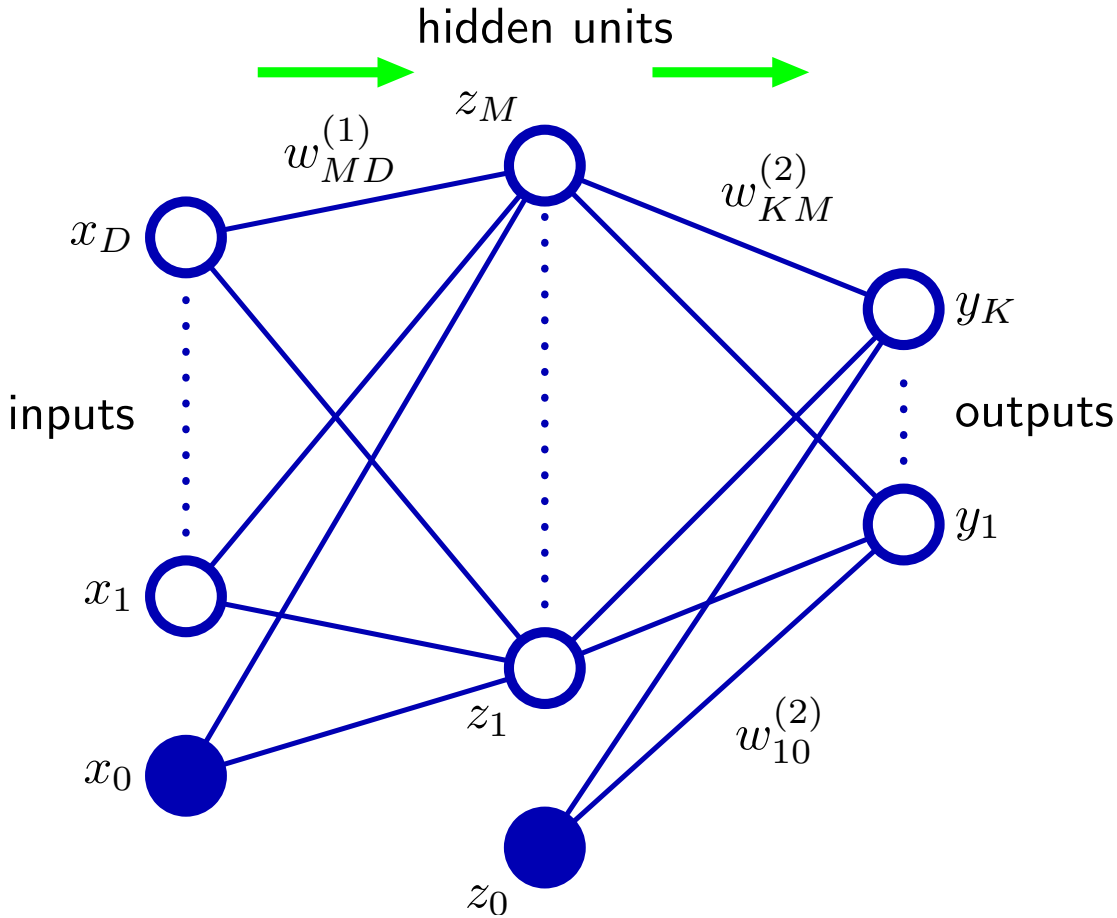
Extensions (II)

- Supervised versions of GPLVM for doing dimensionality reduction while doing supervised learning.

- Several extensions are reviewed in Li and Chen (2016).

Contents

Typical architecture of a neural network



Model (I)

- The basic model of a neural network (NN) can be described by a series of functional transformations.

- We first construct M linear combinations of the input variables $x_1, \ldots, x_D$ in the form

$$a_j = \sum_{i=1}^{D} w_{j,i}^{(1)} x_i + w_{j,0}^{(1)},$$

where $j = 1, \ldots, M$, and the superindex (1) indicates the parameters corresponding to the first "layer" of the network.

Model (II)

- The quantities a_j are known as *activations*.
- The a_j are transformed using a non-linear activation function to give

$$z_j = h(a_j).$$

- In this context, these functions are known as *hidden nodes*.

Model (III)

- The z_j are linearly combined again to give *output activations*

$$a_k = \sum_{j=1}^{M} w_{k,j}^{(2)} z_j + w_{k,0}^{(2)},$$

  where $k = 1, \ldots, K$, and K is the total number of outputs.

- The super-index (2) refers to the parameters of the second "layer" of the network.

- The output activations are transformed or not depending on the problem to address
  - Regression $\rightarrow y_k = a_k$.
  - Classification $\rightarrow y_k = \sigma(a_k)$.

Model (IV)

- Combining both stages, we get

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{j=1}^{M} w_{k,j}^{(2)} h\left(\sum_{i=1}^{D} w_{j,i}^{(1)} x_i + w_{j,0}^{(1)}\right) + w_{k,0}^{(2)}\right),$$

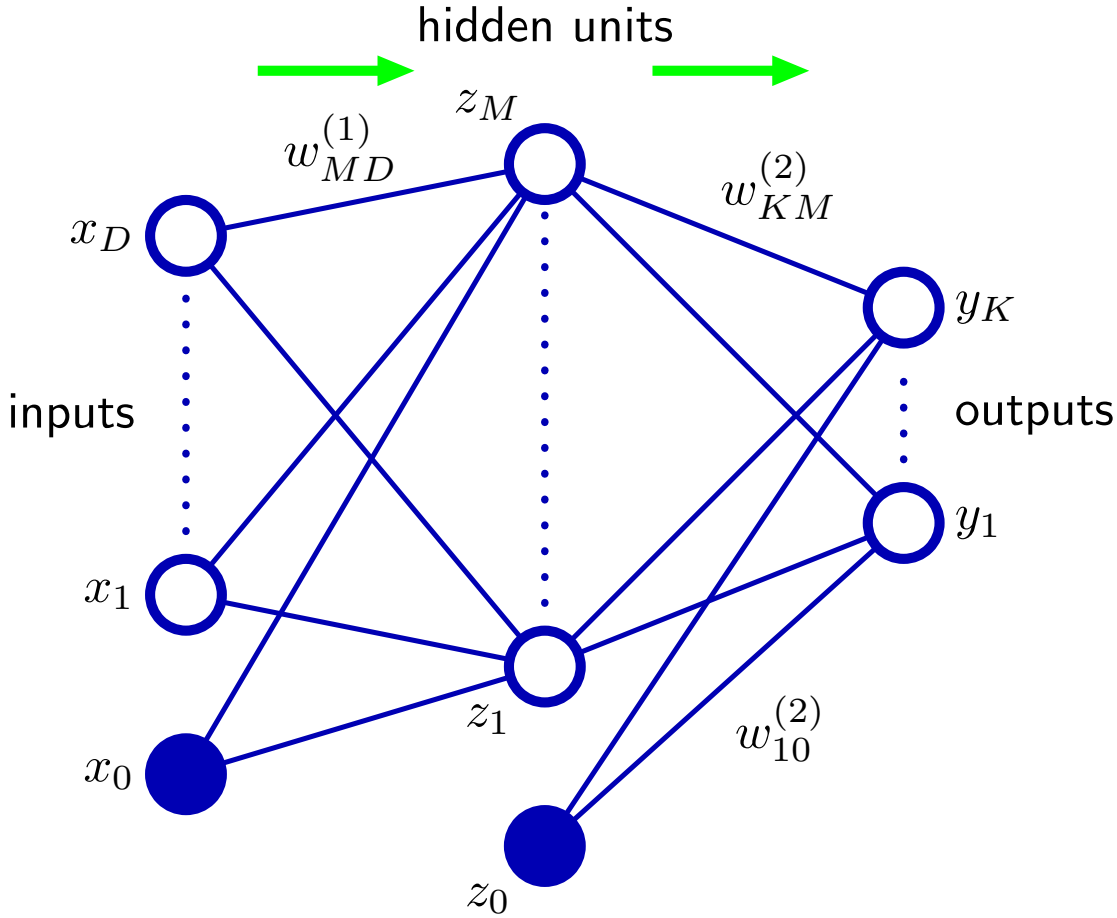
$$= \sigma\left(\sum_{j=0}^{M} w_{k,j}^{(2)} h\left(\sum_{i=0}^{D} w_{j,i}^{(1)} x_i\right)\right),$$

  with $x_0 = 1$.

- Notice that in the second equality, we have made use of $z_0 = h\left(\sum_{i=0}^{D} w_{0,i}^{(1)} x_i\right)$.
- Parameters $\{w_{j,i}\}_{j=1,i=0}^{M,D}$ y $\{w_{k,j}\}_{k=1,j=0}^{K,M}$ are joinly denoted as $\mathbf{w}$.
- Neural network: non-linear function of $\{x_i\}_{i=1}^{D}$ to $\{y_k\}_{k=1}^{K}$ controlled by $\mathbf{w}$.

Model (V)

The network in the figure is a two-layer NN due to two is the number of layers with adaptive weights.



How to train a neural network?

- The backpropagation algorithm.
- The core ideas behind modern feed-forward networks have not changed substantially since the 1980s.
- Improvements today are mainly due to: large datasets and, powerful computers and software infrastructure.
- Algorithmic changes:
  - Cross-entropy error functions instead of mean-squared error.
  - Replacement of sigmoid hidden units with piecewise linear units, such as rectified linear units.

Contents

Deep Gaussian Processes

Deep Gaussian Processes

Andreas C. Damianou **Neil D. Lawrence**
Dept. of Computer Science & Sheffield Institute for Translational Neuroscience,
University of Sheffield, UK

Abstract

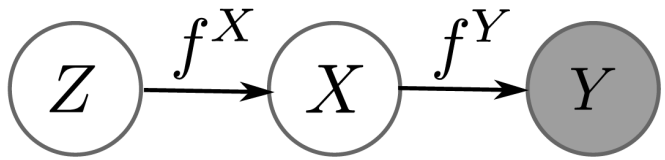
In this paper we introduce deep Gaussian process (GP) models. Deep GPs are a deep belief network based on Gaussian process mappings. The data is modeled as the output of a multivariate

the question as to whether deep structures and the learning of abstract structure can be undertaken in *smaller* data sets. For smaller data sets, questions of generalization arise: to demonstrate such structures are justified it is useful to have an objective measure of the model's applicability.

The traditional approach to deep learning is based around

Published at AISTATS, 2013.

Deep GPs: model



- The leaf nodes $\mathbf{Y} \in \mathbb{R}^{N \times D}$ (observations)
- Intermediate latent spaces $\mathbf{X}_h \in \mathbb{R}^{N \times Q_h}$, with $h = 1, \ldots, H-1$, H number of hidden layers.
- Parent latent node $\mathbf{Z} = \mathbf{X}_H \in \mathbb{R}^{N \times Q_z}$ (unobserved or inputs).
- Example generative process with two layers:

$$x_{nq} = f_q^X(\mathbf{z}_n) + \epsilon_{nq}^X, \quad q = 1, \ldots, Q \quad \mathbf{z}_n \in \mathbb{R}^{Q_z}$$
$$y_{nd} = f_d^Y(\mathbf{x}_n) + \epsilon_{nd}^Y, \quad d = 1, \ldots, D \quad \mathbf{x}_n \in \mathbb{R}^{Q}$$

- The functions $f^X \sim \mathcal{GP}(0, k^X(\mathbf{Z}, \mathbf{Z}))$ and $f^Y \sim \mathcal{GP}(0, k^Y(\mathbf{X}, \mathbf{X}))$.

Deep GPs: variational inference

- Variational inference requires the optimisation of

$$\log p(\mathbf{Y}) = \log \int_{\mathbf{X},\mathbf{Z}} p(\mathbf{Y}|\mathbf{X}) p(\mathbf{X}|\mathbf{Z}) p(\mathbf{Z}) d\mathbf{X} d\mathbf{Z}$$

- Intractability since $\mathbf{X}$ and $\mathbf{Z}$ appear inside the kernel functions.
- A variational lower bound can be found using

$$\mathcal{L} = \int_{\mathbf{X},\mathbf{Z},\mathbf{F}^Y,\mathbf{F}^X} \mathcal{Q} \log \frac{p(\mathbf{Y}, \mathbf{F}^Y, \mathbf{X}, \mathbf{F}^X, \mathbf{Z})}{\mathcal{Q}} d\mathbf{X} d\mathbf{Z} d\mathbf{F}^Y d\mathbf{F}^X,$$

  where $\mathcal{Q}$ is an approximated posterior to be defined.
- The complete likelihood follows as

$$p(\mathbf{Y}, \mathbf{F}^Y, \mathbf{X}, \mathbf{F}^X, \mathbf{Z}) = p(\mathbf{Y}|\mathbf{F}^Y) p(\mathbf{F}^Y|\mathbf{X}) p(\mathbf{X}|\mathbf{F}^X) p(\mathbf{F}^X|\mathbf{Z}) p(\mathbf{Z}).$$

Deep GPs: inducing variables

- The key trick for inference: augment the probability space above with K auxiliary pseudo-inputs $\widetilde{\mathbf{X}} \in \mathbb{R}^{K \times Q}$ and $\widetilde{\mathbf{Z}} \in \mathbb{R}^{K \times Q_z}$.
- These auxiliary pseudo-inputs correspond to function values $\mathbf{U}^Y \in \mathbb{R}^{K \times D}$ and $\mathbf{U}^X \in \mathbb{R}^{K \times Q}$.
- The augmented probability space is given as

$$p(\mathbf{Y}, \mathbf{F}^Y, \mathbf{X}, \mathbf{F}^X, \mathbf{Z}, \mathbf{U}^Y, \mathbf{U}^X, \widetilde{\mathbf{X}}, \widetilde{\mathbf{Z}}) = p(\mathbf{Y}|\mathbf{F}^Y)p(\mathbf{F}^Y|\mathbf{U}^Y, \mathbf{X})p(\mathbf{U}^Y|\widetilde{\mathbf{X}})\times$$
$$p(\mathbf{X}|\mathbf{F}^X)p(\mathbf{F}^X|\mathbf{U}^X, \mathbf{Z})p(\mathbf{U}^X|\widetilde{\mathbf{Z}})p(\mathbf{Z}).$$

Deep GPs: choice of the posterior $\mathcal{Q}$

- The posterior distribution is chosen as

$$\mathcal{Q} = p(\mathbf{F}^Y|\mathbf{U}^Y, \mathbf{X})q(\mathbf{U}^Y)q(\mathbf{X})p(\mathbf{F}^X|\mathbf{U}^X, \mathbf{Z})q(\mathbf{U}^X)q(\mathbf{Z}),$$

where $q(\mathbf{U}^Y)$ and $q(\mathbf{U}^X)$ are allowed a free-form and

$$q(\mathbf{X}) = \prod_{q=1}^{Q} \mathcal{N}(\mathbf{x}_{:,q}|\boldsymbol{\mu}_q^X, \mathbf{S}_q^X), \qquad q(\mathbf{Z}) = \prod_{q=1}^{Q_Z} \mathcal{N}(\mathbf{z}_{:,q}|\boldsymbol{\mu}_q^Z, \mathbf{S}_q^Z).$$

- With these choices, the new lower bound is

$$\mathcal{L} = \int \mathcal{Q} \log \frac{p(\mathbf{Y}|\mathbf{F}^Y)p(\mathbf{U}^Y)p(\mathbf{X}|\mathbf{F}^X)p(\mathbf{U}^X)p(\mathbf{Z})}{\mathcal{Q}'} d\mathbf{X}d\mathbf{Z}d\mathbf{F}^Y d\mathbf{F}^X \mathbf{U}^Y d\mathbf{U}^X,$$

where $Q' = q(\mathbf{U}^Y)q(\mathbf{X})q(\mathbf{U}^X)q(\mathbf{Z})$.

Deep GPs: factorised bound

- The bound can be written as

$$\mathcal{L} = \mathbf{g}_Y + \mathbf{r}_X + \mathcal{H}_{q(\mathbf{X})} - \mathrm{KL}(q(\mathbf{Z}) \| p(\mathbf{Z})),$$

where

$$\begin{aligned}
\mathbf{g}_Y &= g\left(\mathbf{Y}, \mathbf{F}^Y, \mathbf{U}^Y, \mathbf{X}\right) \\
&= \left\langle \log p\left(\mathbf{Y} \mid \mathbf{F}^Y\right) + \log \frac{p\left(\mathbf{U}^Y\right)}{q\left(\mathbf{U}^Y\right)} \right\rangle_{p(\mathbf{F}^Y|\mathbf{U}^Y,\mathbf{X})q(\mathbf{U}^Y)q(\mathbf{X})} \\
\mathbf{r}_X &= r\left(\mathbf{X}, \mathbf{F}^X, \mathbf{U}^X, \mathbf{Z}\right) \\
&= \left\langle \log p\left(\mathbf{X} \mid \mathbf{F}^X\right) + \log \frac{p\left(\mathbf{U}^X\right)}{q\left(\mathbf{U}^X\right)} \right\rangle_{p(\mathbf{F}^X|\mathbf{U}^X,\mathbf{Z})q(\mathbf{U}^X)q(\mathbf{X})q(\mathbf{Z})}
\end{aligned}$$

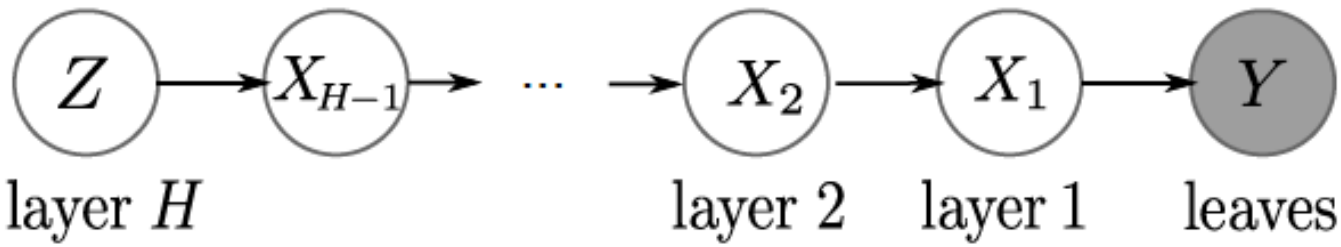
- $\mathcal{H}_{q(\mathbf{X})}$ is the entropy for $q(\mathbf{X})$ and KL denotes the Kullback – Leibler divergence.

Deep GPs: factorised bound

- Because $\mathbf{g}_Y$ and $\mathbf{r}_X$ only involve Gaussian densities, these terms are tractable.

- The $\mathbf{g}_Y$ is associated to the leaves (the observed data) and follows a similar form to the bound found in the Bayesian GPLVM.

Extending the hierarchy



Extending the bound

- The model can be extended vertically (adding more hidden layers) or horizontally (considering conditional independencies of the latent variables belonging to the same layer).

- Vertical extension:
  - Add more $\mathbf{r}_X$ functions to the bound
  - Instead of a single term $\mathbf{r}_X$, there will be a sum

$$\sum_{h=1}^{H-1} \mathbf{r}_{X_h},$$

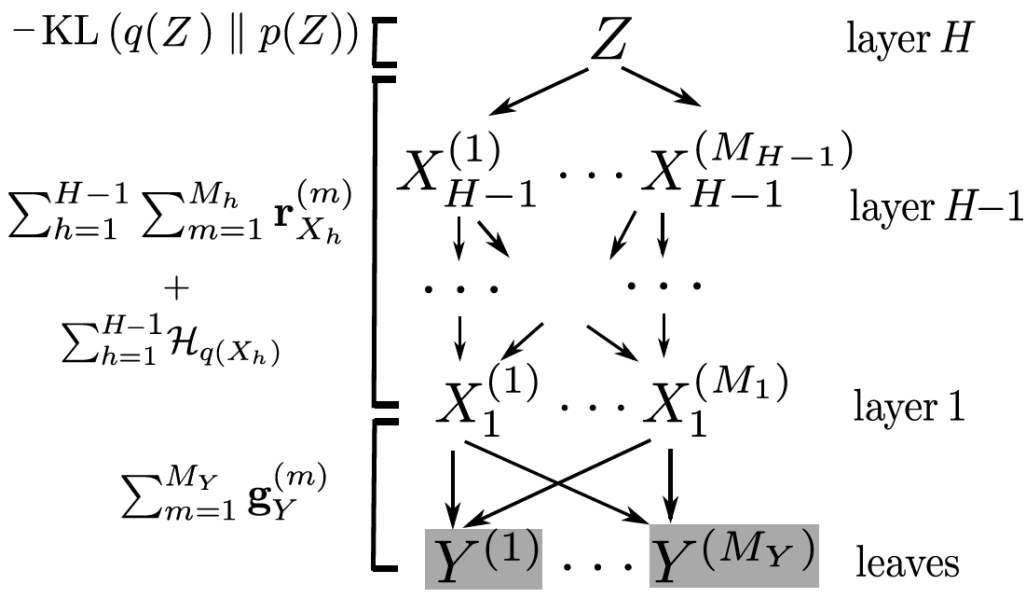
where $\mathbf{r}_{X_h} = r\left(\mathbf{X}_h, \mathbf{F}^{X_h}, \mathbf{U}^{X_h}, \mathbf{X}_{h+1}\right), \mathbf{X}_H = \mathbf{Z}$.

- Horizontal extension:
  - Break the single latent space $\mathbf{X}_h$, of layer h, to M_h conditionally independent subsets.
  - This can be achieved by breaking the original $\mathbf{r}_{X_h}$ into the sum $\sum_{m=1}^{M_h} \mathbf{r}_{X_h}^{(m)}$.

General bound

If there additionally are M_Y outputs, the extended bound follows as

$$\mathcal{L} = \sum_{m=1}^{M_Y} \mathbf{g}_Y^{(m)} + \sum_{h=1}^{H-1} \sum_{m=1}^{M_h} \mathbf{r}_{X_h}^{(m)} + \sum_{h=1}^{H-1} \mathcal{H}_{q(\mathbf{x}_h)} - \mathrm{KL}(q(\mathbf{Z}) \| p(\mathbf{Z}))$$



Deep GPs: step function

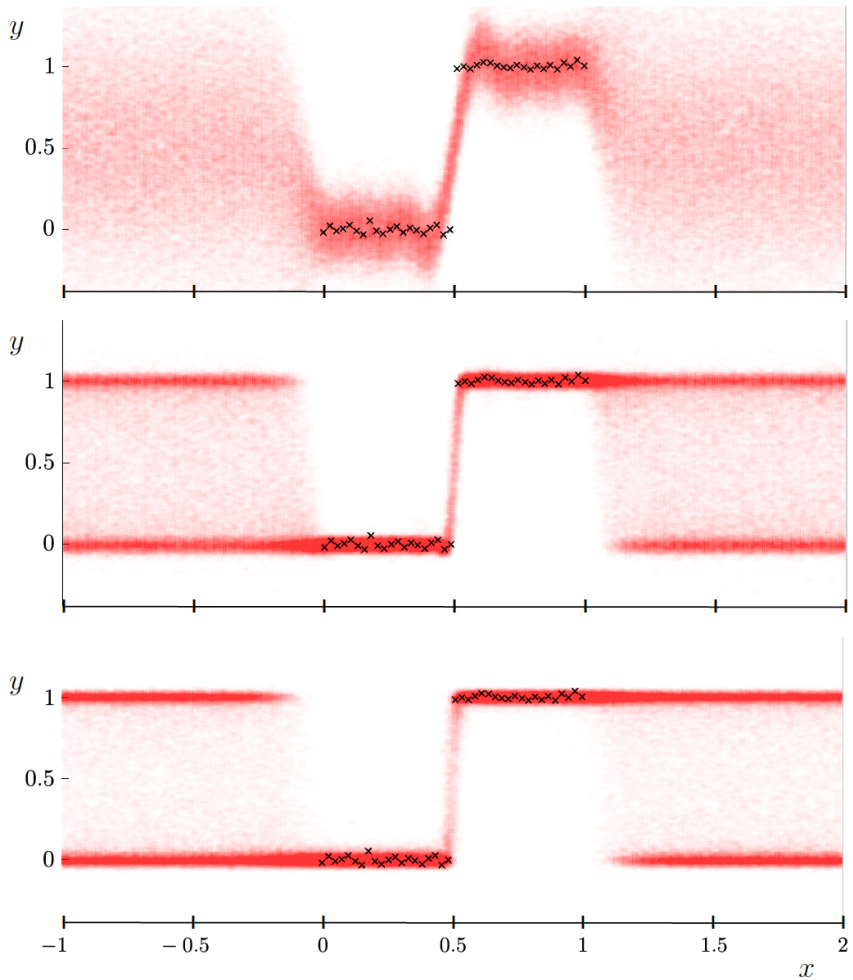
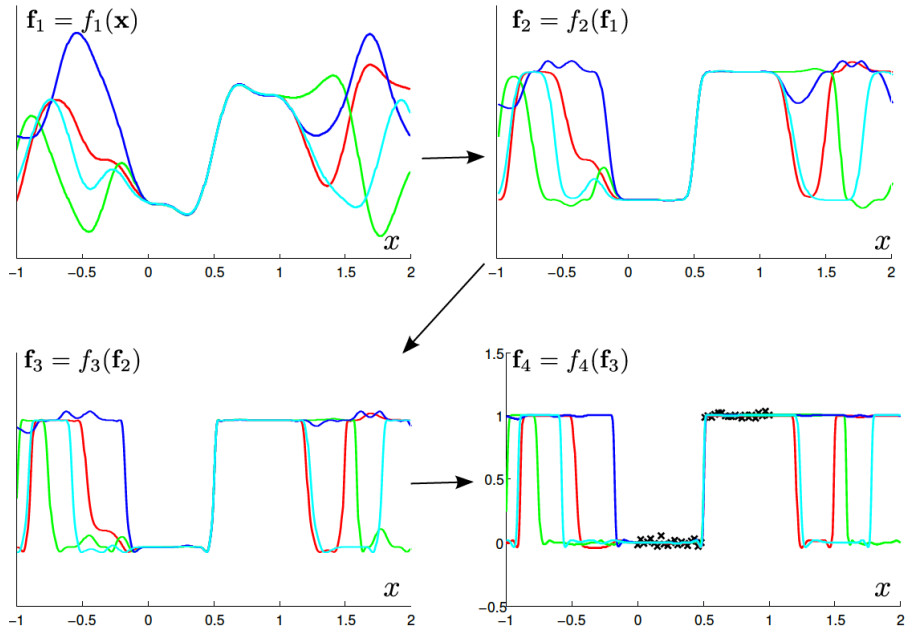


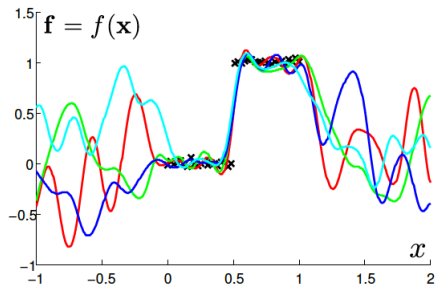
Fig. 6.5: Samples drawn from the posterior of a standard GP (top), a deep GP with 1 hidden layer (middle) and a deep GP with three hidden layers (bottom). As in [Hensman and Lawrence, 2014], for better visualisation the sampled points are drawn as small, red semi-transparent squares, whereas the training points are plotted as black opaque x's.

Damianou (2015)

Deep GPs: another view of the step function



(a) Deep GP with three hidden plus one warping layer.



(b) Standard GP

Fig. 6.6: (a) Successive warping of sample paths drawn from a deep GP, gradually "pushing" them towards a bimodal regime. (b) Sample paths drawn from a standard GP. In this figure, each sample is drawn versus the initial input $\mathbf{x}$. Instead, figure 6.7 shows a plot of each layer's output signal versus its input signal.

Damianou (2015)

Deep GPs: warping behavior

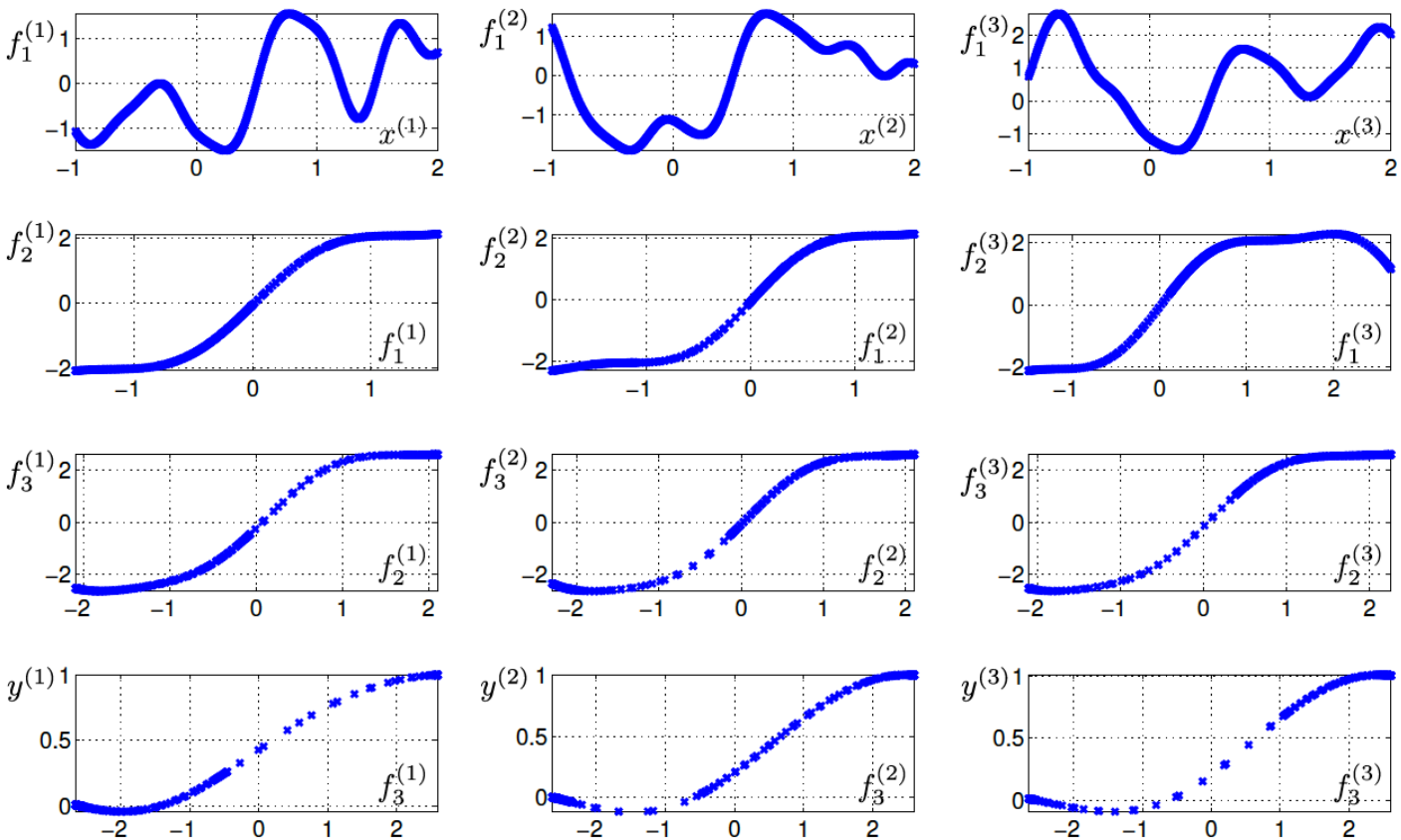


Fig. 6.7: Plotting the input versus the output sample signals of every layer in the deep GP. This reveals sigmoidal warping functions which successively "push" the top layer's samples into the bi-modal regime. In the axis labels, the subscript denotes the sample number and the subscript denotes the layer number.

Damianou (2015)

Deep GPs: toy regression problem

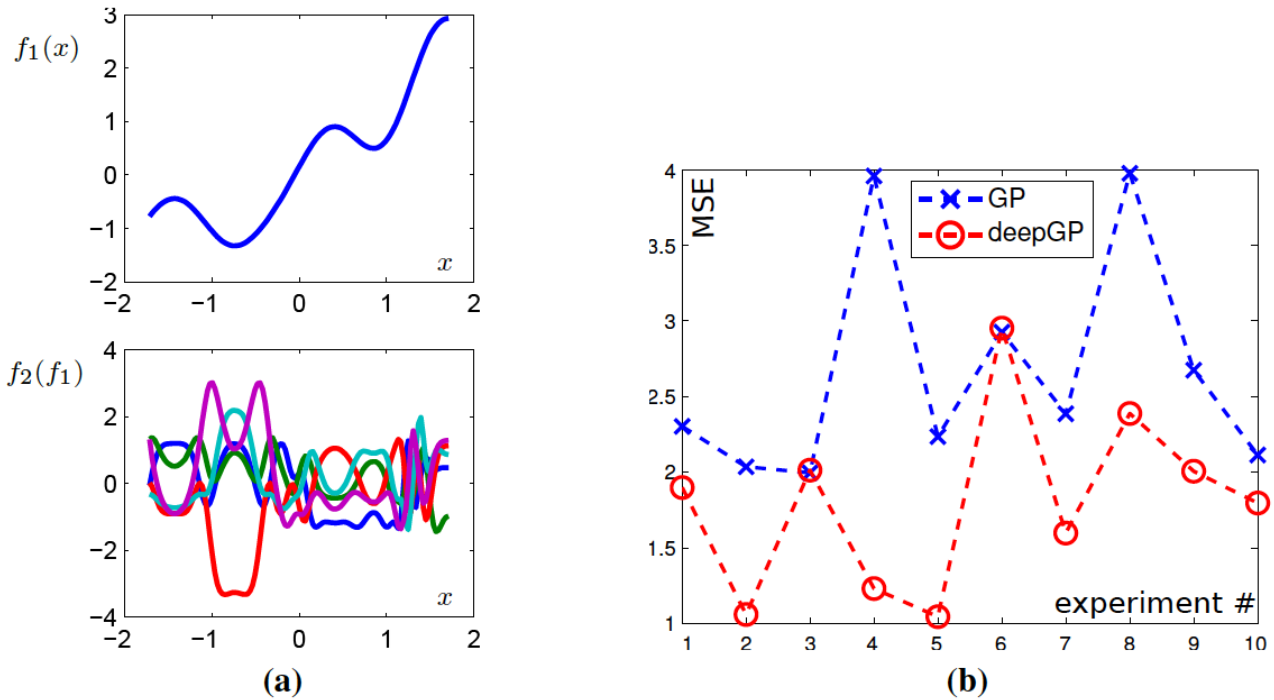
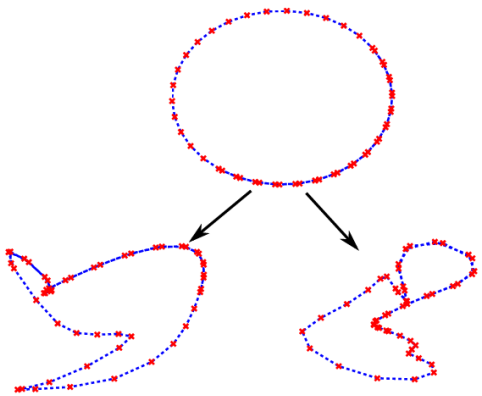


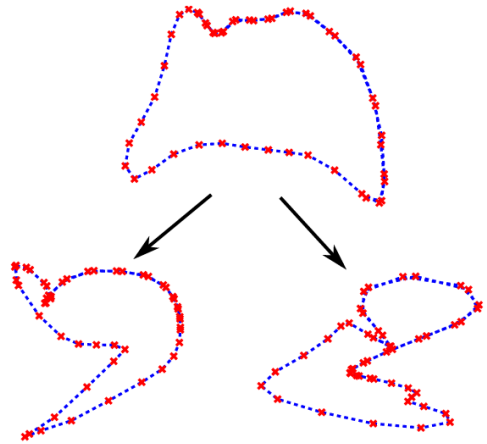
Fig. 6.8: Figure (a) shows the toy data created for the regression experiment. The top plot shows the (hidden) warping function and bottom plot shows the final (observed) output. Figure (b) shows the results obtained over each experiment repetition.

Damianou (2015)

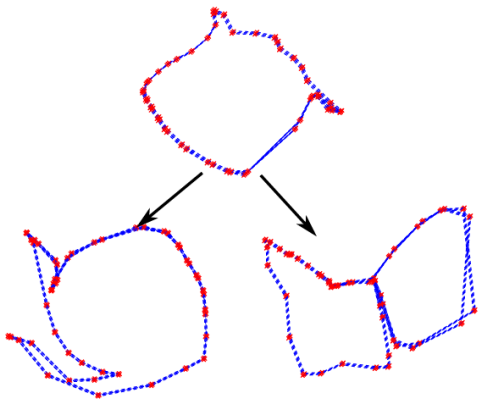
Deep GPs: manifold learning



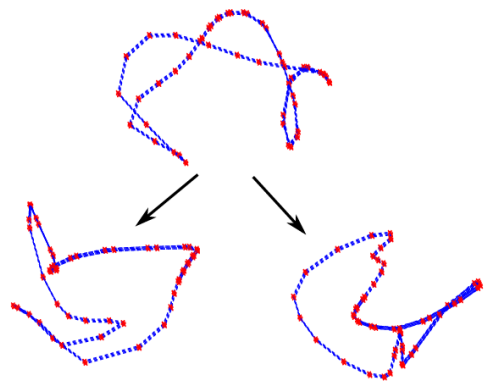
(a) Real data.



(b) Deep GP reconstruction.



(c) Stacked Isomap reconstruction.



(d) Stacked PCA reconstruction.

Fig. 6.9: Attempts to reconstruct the real data (fig. (a)) with our model (b), stacked Isomap (c) and stacked PCA (d). Our model can also find the correct dimensionalities automatically.

Damianou (2015)

Recent advances in Deep GPs

Random Feature Expansions for Deep Gaussian Processes

Kurt Cutajar [1] **Edwin V. Bonilla** [2] **Pietro Michiardi** [1] **Maurizio Filippone** [1]

Abstract

The composition of multiple Gaussian Processes as a Deep Gaussian Process (DGP) enables a deep probabilistic nonparametric approach to flexibly tackle complex machine learning problems with sound quantification of uncertainty. Existing inference approaches for DGP models have limited scalability and are notoriously cumbersome to construct. In this work we introduce a novel formulation of DGPs based on random feature expansions that we train using stochastic variational inference. This yields a practical learn-

2006) such that the output of each layer of GPs forms the input to the GPs at the next layer, effectively implementing a deep probabilistic nonparametric model for compositions of functions (Neal, 1996; Duvenaud et al., 2014).

Because of their probabilistic formulation, it is natural to approach the learning of DGPs through Bayesian inference techniques; however, the application of such techniques to learn DGPs leads to various forms of intractability. A number of contributions have been proposed to recover tractability, extending or building upon the literature on approximate methods for GPs. Nevertheless, only few works leverage one of the key features that arguably make DNNs

Published at ICML, 2017.

RF expansions for Deep GPs

- This paper propose a practical approach to scale deep GPs using a random Fourier feature representations for the kernel functions,

$$k_{\text{rbf}}\left(\mathbf{x}_i, \mathbf{x}_j\right) \approx \frac{1}{N_{\text{RF}}} \sum_{r=1}^{N_{\text{RF}}} \mathbf{z}\left(\mathbf{x}_i \mid \tilde{\omega}_r\right)^\top \mathbf{z}\left(\mathbf{x}_j \mid \tilde{\omega}_r\right),$$

  where N_{RF} is the number of random features,
  $\mathbf{z}(\mathbf{x} \mid \omega) = \left[\cos\left(\mathbf{x}^\top \omega\right), \sin\left(\mathbf{x}^\top \omega\right)\right]^\top$ and $\tilde{\omega}_r \sim p(\omega)$.

- It uses SVI for Bayesian inference.

Recent advances in Deep GPs

Doubly Stochastic Variational Inference for Deep Gaussian Processes

Hugh Salimbeni
Imperial College London
hrs13@ic.ac.uk

Marc Deisenroth
Imperial College London
m.deisenroth@imperial.ac.uk

Abstract

Gaussian processes (GPs) are a good choice for function approximation as they are flexible, robust to over-fitting, and provide well-calibrated predictive uncertainty. Deep Gaussian processes (DGPs) are multi-layer generalisations of GPs, but inference in these models has proved challenging. Existing approaches to inference in DGP models assume approximate posteriors that force independence between the layers, and do not work well in practice. We present a doubly stochastic variational inference algorithm, which does not force independence between layers. With our method of inference we demonstrate that a DGP model can be used effectively on data ranging in size from hundreds to a billion points. We provide strong empirical evidence that our inference scheme for DGPs works well in practice in both classification and regression.

Published at NeurIPS, 2017.

Doubly SVI for Deep Gaussian Processes

- The variational approach used in the original deep GP make strong independence and Gaussianity assumptions.
- The true posterior is likely to exhibit high correlations between layers.
- This paper presents a variational algorithm for inference in DGP models that does not force independence or Gaussianity between the layers,

$$q\left(\left\{\mathbf{F}^{l}\right\}_{l=1}^{L}\right)=\prod_{l=1}^{L} q\left(\mathbf{F}^{l} \mid \mathbf{m}^{l}, \mathbf{S}^{l} ; \mathbf{F}^{l-1}, \mathbf{Z}^{l-1}\right).$$

- The paper also includes a linear mean function for all the inner layers.

Contents

Deep Kernel Learning

Deep Kernel Learning

Andrew Gordon Wilson* CMU
Zhiting Hu* CMU
Ruslan Salakhutdinov University of Toronto
Eric P. Xing CMU

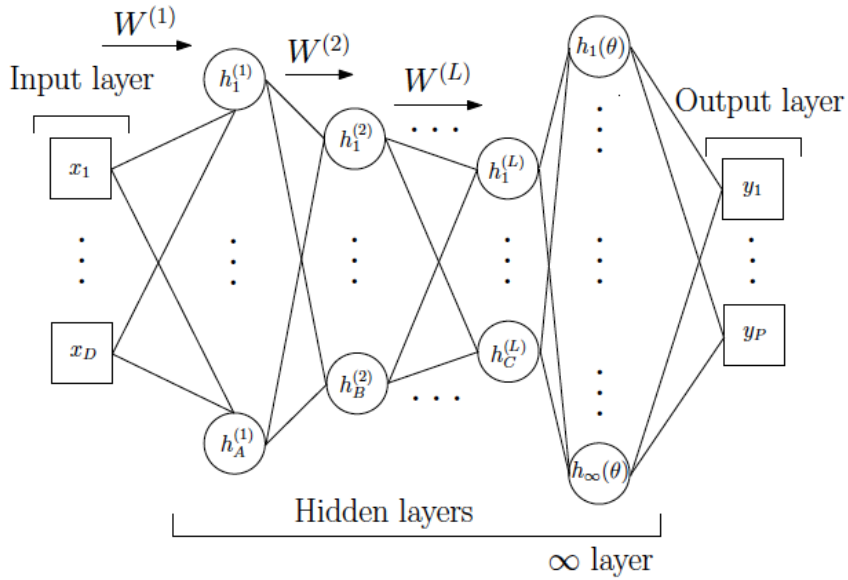
Abstract

We introduce scalable deep kernels, which combine the structural properties of deep learning architectures with the non-parametric flexibility of kernel methods. Specifically, we transform the inputs of a spectral mixture base kernel with a deep architecture, using local kernel interpolation,

(1996), who had shown that Bayesian neural networks with infinitely many hidden units converged to Gaussian processes with a particular kernel (covariance) function. Gaussian processes were subsequently viewed as flexible and interpretable alternatives to neural networks, with straightforward learning procedures. Where neural networks used finitely many highly adaptive basis functions, Gaussian processes typically used infinitely many fixed basis functions. As argued by MacKay (1998), Hinton et al.

Published at AISTATS, 2016.

Deep Kernel Learning



- Starting from a base kernel $k(\mathbf{x}_i, \mathbf{x}_j|\theta)$, the inputs are transformed as

$$k(\mathbf{x}_i, \mathbf{x}_j|\theta) \rightarrow k(g(\mathbf{x}_i, \mathbf{w}), g(\mathbf{x}_j, \mathbf{w})|\theta, \mathbf{w}),$$

  where $g(\mathbf{x}, \mathbf{w})$ is a non-linear mapping given by a deep architecture parametrized by weights $\mathbf{w}$.
- For scalability, they use KISS-GP (Kernel Interpolation for Scalable Structured), $K \approx MK_{U,U}^{\text{deep}}M^\top$

Stochastic Variational Deep Kernel Learning

Stochastic Variational Deep Kernel Learning

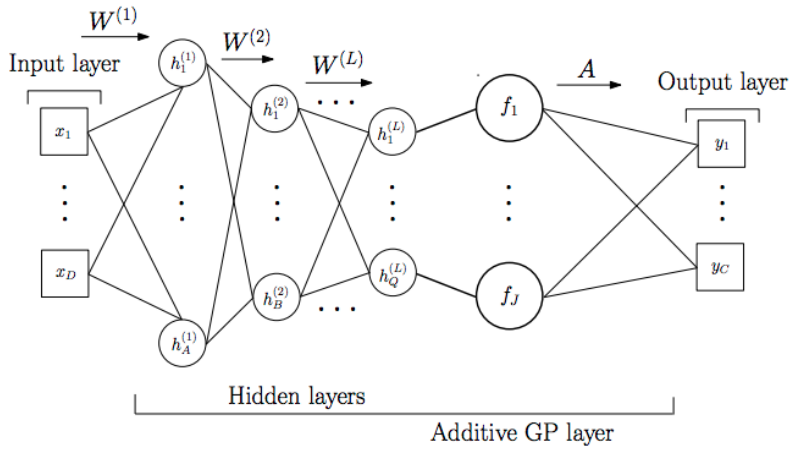
Andrew Gordon Wilson* Cornell University
Zhiting Hu* CMU
Ruslan Salakhutdinov CMU
Eric P. Xing CMU

Abstract

Deep kernel learning combines the non-parametric flexibility of kernel methods with the inductive biases of deep learning architectures. We propose a novel deep kernel learning model and stochastic variational inference procedure which generalizes deep kernel learning approaches to enable classification, multi-task learning, additive covariance structures, and stochastic gradient training. Specifically, we apply additive base kernels to subsets of output features from deep neural architectures, and jointly learn the parameters of the base kernels and deep network through a Gaussian process marginal likelihood objective. Within this framework, we derive an efficient form of stochastic variational inference which leverages local kernel interpolation, inducing points, and structure exploiting algebra. We show improved performance over stand alone deep networks, SVMs, and state of the art scalable Gaussian processes on several classification benchmarks, including an airline delay dataset containing 6 million training points, CIFAR, and ImageNet.

Published at NeurIPS, 2016.

Stochastic Variational Deep Kernel Learning



1. A deep non-linear transformation $\mathbf{h}(\mathbf{x}, \mathbf{w})$ parametrized by $\mathbf{w}$ is applied to the input vector $\mathbf{x}$ to produce Q features at the final layer L, $h_1^L, \ldots, h_Q^L$.
2. J Gaussian processes with base kernels $k_1, \ldots, k_J$, are applied to subsets of these features corresponding to an *additive GP*.
3. These GPs are linearly mixed by a matrix $A \in \mathbb{R}^{C \times J}$, and transformed by an observation model, to produce the output variables $y_1, \ldots, y_C$.

References I

C.M. Bishop and G.D. James. Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 327(2):580 – 593, 1993. ISSN 0168-9002. doi: https://doi.org/10.1016/0168-9002(93)90728-Z. URL http://www.sciencedirect.com/science/article/pii/016890029390728Z.

Andreas Damianou. *Deep Gaussian Processes and Variational Propagation of Uncertainty*. PhD thesis, Department of Neuroscience, University of Sheffield, 2015.

Carl Henrik Ek, Jon Rihan, Philip H. S. Torr, Grégory Rogez, and Neil D. Lawrence. Ambiguity modeling in latent spaces. In Andrei Popescu-Belis and Rainer Stiefelhagen, editors, *Machine Learning for Multimodal Interaction*, pages 62–73, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-85853-9.

Ping Li and Songcan Chen. A review on gaussian process latent variable models. *CAAI Transactions on Intelligence Technology*, 1(4):366 – 376, 2016. ISSN 2468-2322. doi: https://doi.org/10.1016/j.trit.2016.11.004. URL http://www.sciencedirect.com/science/article/pii/S2468232216300828.

Sam Roweis, Lawrence Saul, and Geoffrey E Hinton. Global coordination of local linear models. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 889–896. MIT Press, 2002. URL https://proceedings.neurips.cc/paper/2001/file/850af92f8d9903e7a4e0559a98ecc857-Paper.pdf.

Aaron Shon, Keith Grochow, Aaron Hertzmann, and Rajesh PN Rao. Learning shared latent structure for image synthesis and robotic imitation. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 1233–1240. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2005/file/030e65da2b1c944090548d36b244b28d-Paper.pdf.

References II

Michalis Titsias and Neil D. Lawrence. Bayesian gaussian process latent variable model. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 844–851, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Workshop and Conference Proceedings. URL `http://proceedings.mlr.press/v9/titsias10a.html`.

Michalis K. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *In Artificial Intelligence and Statistics 12*, pages 567–574, 2009.