# Deep Gaussian Processes

Neil D. Lawrence
Sheffield Institute of Translational Neuroscience and Department of Computer Science, University of Sheffield, U.K.

University College London

4th September 2014

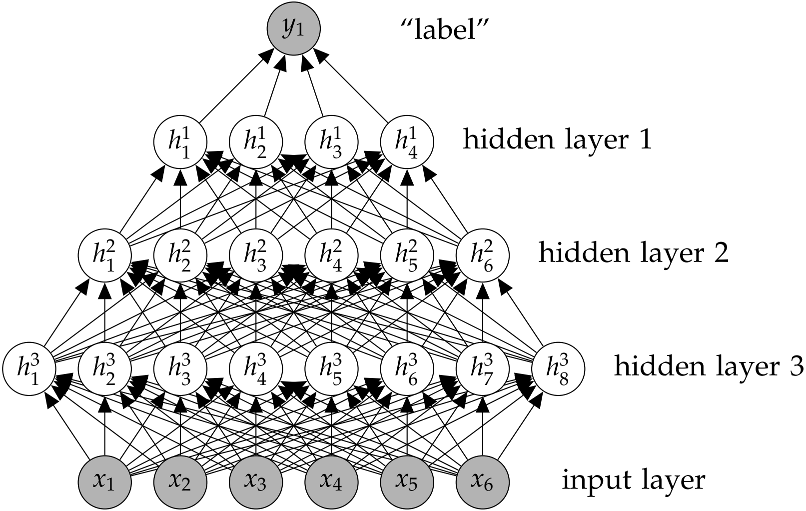
# Outline

Introduction

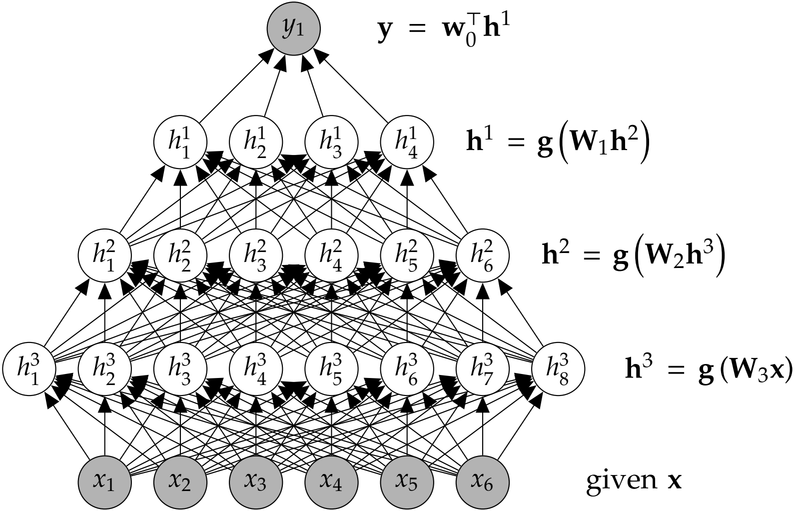
Deep Gaussian Process Models

Conclusions

# Deep Neural Network



# Deep Neural Network



# Mathematically

$$\mathbf{h}^3 = \mathbf{g}\left(\mathbf{W}_3\mathbf{x}\right)$$
$$\mathbf{h}^2 = \mathbf{g}\left(\mathbf{W}_2\mathbf{h}^3\right)$$
$$\mathbf{h}^1 = \mathbf{g}\left(\mathbf{W}_1\mathbf{h}^2\right)$$
$$\mathbf{y} = \mathbf{w}_0^\top\mathbf{h}^1$$

# Overfitting

- Potential problem: if number of nodes in two adjacent layers is big, corresponding **W** is also very big and there is the potential to overfit.
- Proposed solution: "dropout".
- Alternative solution: parameterize **W** as it's SVD.

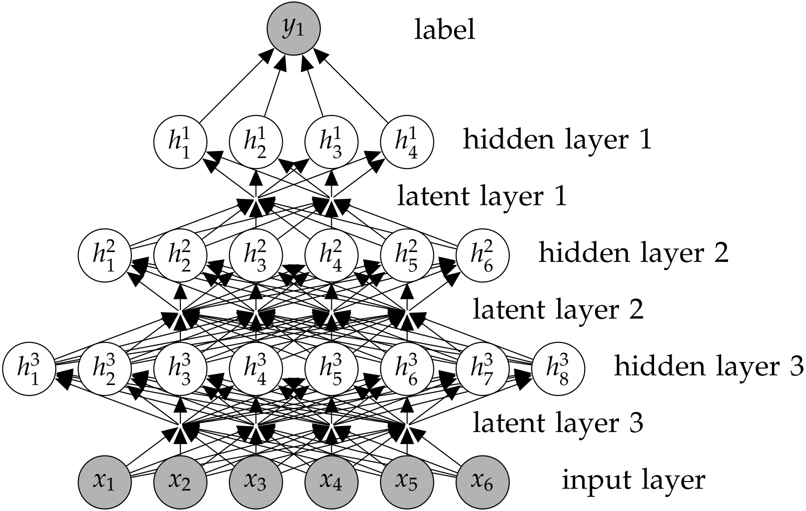
$$\mathbf{W} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top$$

or

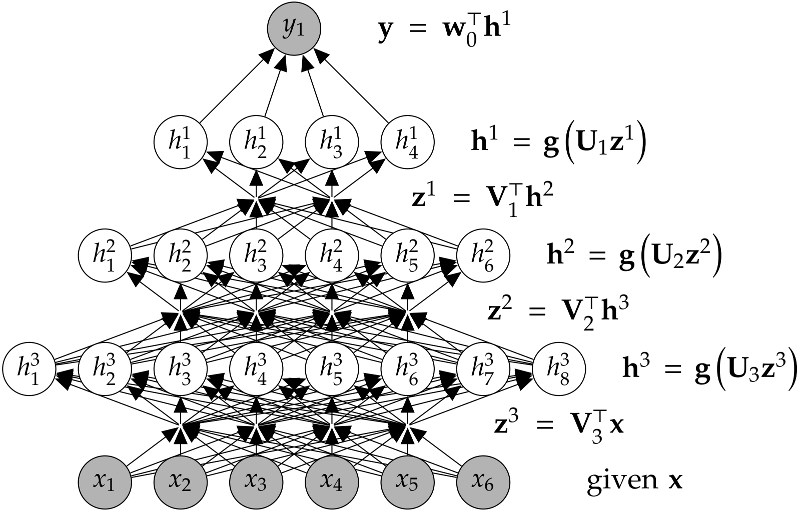
$$\mathbf{W} = \mathbf{U}\mathbf{V}^\top$$

where if $\mathbf{W} \in \Re^{k_1 \times k_2}$ then $\mathbf{U} \in \Re^{k_1 \times q}$ and $\mathbf{V} \in \Re^{k_2 \times q}$, i.e. we have a low rank matrix factorization for the weights.

# Deep Neural Network



# Deep Neural Network



# Mathematically

$$\begin{aligned}
\mathbf{z}^3 &= \mathbf{V}_3^\top \mathbf{x} \\
\mathbf{h}^3 &= \mathbf{g}\left(\mathbf{U}_3 \mathbf{z}^3\right) \\
\mathbf{z}^2 &= \mathbf{V}_2^\top \mathbf{h}^3 \\
\mathbf{h}^2 &= \mathbf{g}\left(\mathbf{U}_2 \mathbf{z}^2\right) \\
\mathbf{z}^1 &= \mathbf{V}_1^\top \mathbf{h}^2 \\
\mathbf{h}^1 &= \mathbf{g}\left(\mathbf{U}_1 \mathbf{z}^1\right) \\
\mathbf{y} &= \mathbf{w}_0^\top \mathbf{h}^1
\end{aligned}$$

# A Cascade of Neural Networks

$$
\begin{aligned}
\mathbf{z}^3 &= \mathbf{V}_3^\top \mathbf{x} \\
\mathbf{z}^2 &= \mathbf{V}_2^\top \mathbf{g}\left(\mathbf{U}_3 \mathbf{z}^3\right) \\
\mathbf{z}^1 &= \mathbf{V}_1^\top \mathbf{g}\left(\mathbf{U}_2 \mathbf{z}^2\right) \\
\mathbf{y} &= \mathbf{w}_0^\top \mathbf{U}_1 \mathbf{z}^1
\end{aligned}
$$

# Replace Each Neural Network with a Gaussian Process

$$\mathbf{z}^3 = \mathbf{f}(\mathbf{x})$$

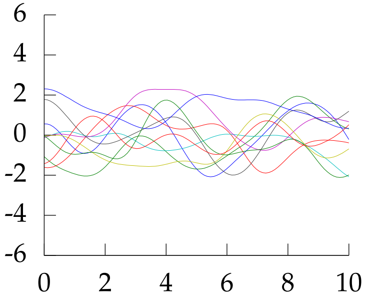
$$\mathbf{z}^2 = \mathbf{f}\left(\mathbf{z}^3\right)$$

$$\mathbf{z}^1 = \mathbf{f}\left(\mathbf{z}^2\right)$$

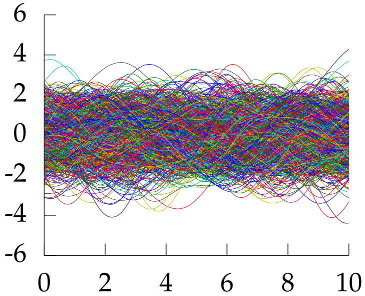
$$\mathbf{y} = \mathbf{f}\left(\mathbf{z}^1\right)$$

This is equivalent to Gaussian prior over weights and integrating out all parameters and taking width of each layer to infinity.

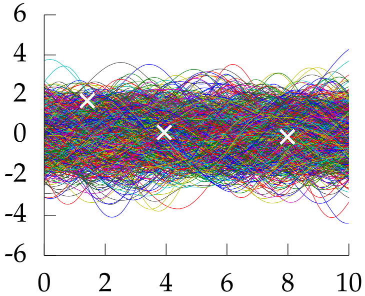
# Gaussian Processes: Extremely Short Overview



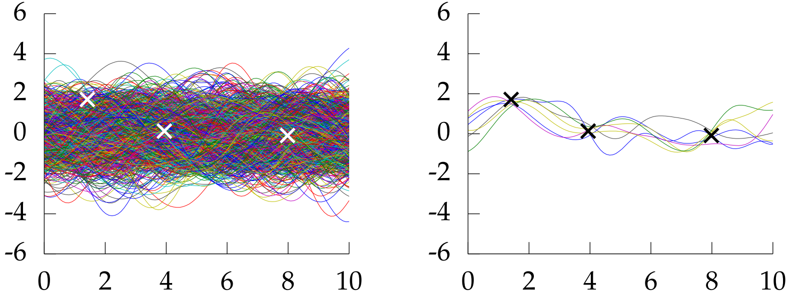
# Gaussian Processes: Extremely Short Overview



# Gaussian Processes: Extremely Short Overview



# Gaussian Processes: Extremely Short Overview



# Outline

# Mathematically

- Composite *multivariate* function

$$\mathbf{f}(\mathbf{x}) = \mathbf{g}_5(\mathbf{g}_4(\mathbf{g}_3(\mathbf{g}_2(\mathbf{g}_1(\mathbf{x})))))$$

# Why Deep?

- Gaussian processes give priors over functions.
- Elegant properties:
  - e.g. *Derivatives* of process are also Gaussian distributed (if they exist).
- For particular covariance functions they are 'universal approxiamtors', i.e. all functions can have support under the prior.
- Gaussian derivatives might ring alarm bells.
- E.g. a priori they don't believe in function 'jumps'.

# Difficulty for Probabilistic Approaches

- Propagate a probability distribution through a non-linear mapping.
- Normalisation of distribution becomes intractable.

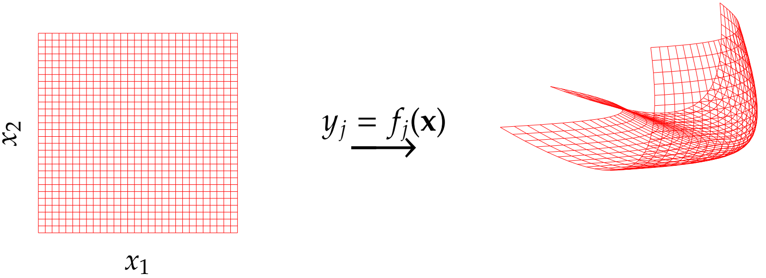



Figure : A three dimensional manifold formed by mapping from a two dimensional space to a three dimensional space.

# Difficulty for Probabilistic Approaches

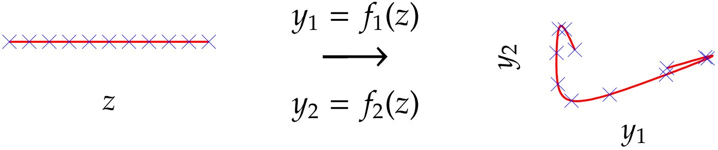



Figure : A string in two dimensions, formed by mapping from one dimension, $z$, line to a two dimensional space, $[y_1,\ y_2]$ using nonlinear functions $f_1(\cdot)$ and $f_2(\cdot)$.

# Difficulty for Probabilistic Approaches

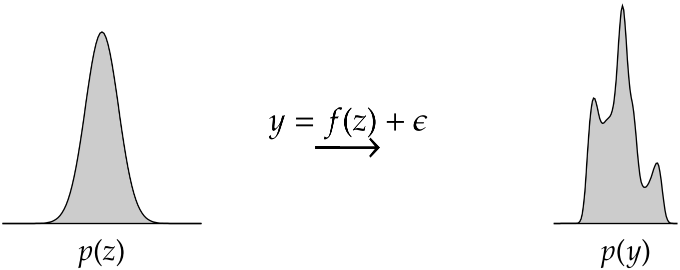



Figure : A Gaussian distribution propagated through a non-linear mapping. $y_i = f(z_i) + \epsilon_i$. $\epsilon \sim \mathcal{N}\left(0, 0.2^2\right)$ and $f(\cdot)$ uses RBF basis, 100 centres between -4 and 4 and $\ell = 0.1$. New distribution over $y$ (right) is multimodal and difficult to normalize.

# Analysis of Deep GPs

- Duvenaud et al. (2014) Duvenaud et al show that the derivative distribution of the process becomes more *heavy tailed* as number of layers increase.

# Variational Compression

(Lawrence, 2007; Titsias, 2009)

- Complexity of standard GP:
  - $\mathcal{O}(n^3)$ in computation.
  - $\mathcal{O}(n^2)$ in storage.

# Variational Compression

(Lawrence, 2007; Titsias, 2009)

- Complexity of standard GP:
  - $\mathcal{O}(n^3)$ in computation.
  - $\mathcal{O}(n^2)$ in storage.
- Via low rank representations of covariance:
  - $\mathcal{O}(nm^2)$ in computation.
  - $\mathcal{O}(nm)$ in storage.
- Where $m$ is user chosen number of *inducing* variables. They give the rank of the resulting covariance.

# Variational Compression

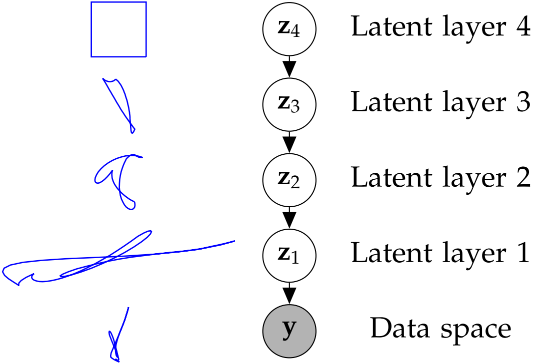
(Lawrence, 2007; Titsias, 2009)

- Complexity of standard GP:
  - $\mathcal{O}(n^3)$ in computation.
  - $\mathcal{O}(n^2)$ in storage.
- Via low rank representations of covariance:
  - $\mathcal{O}(nm^2)$ in computation.
  - $\mathcal{O}(nm)$ in storage.
- Where $m$ is user chosen number of *inducing* variables. They give the rank of the resulting covariance.

# Variational Compression

- Inducing variables are a compression of the real observations.
- They can live in space of $\mathbf{f}$ or a space that is related through a linear operator (Álvarez et al., 2010) — could be gradient or convolution.
- There are inducing variables associated with each set of hidden variables, $\mathbf{z}^i$.
- **Importantly** conditioning on inducing variables renders the likelihood independent across the data.
  - It turns out that this allows us to variationally handle uncertainty on the kernel (including the inputs to the kernel).
  - It also allows standard scaling approaches: stochastic variational inference Hensman et al. (2013), parallelization Gal et al. (2014) and work by Zhenwen Dai on GPUs to be applied: an *engineering* challenge?

# Structures for Extracting Information from Data



# Damianou and Lawrence (2013)

## Deep Gaussian Processes

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

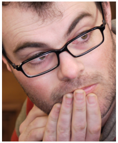
### Abstract

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 GP. The inputs to that Gaussian process are then governed by another GP. A single layer model is equivalent to a standard GP or the GP latent variable model (GP-LVM). We perform inference in

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.

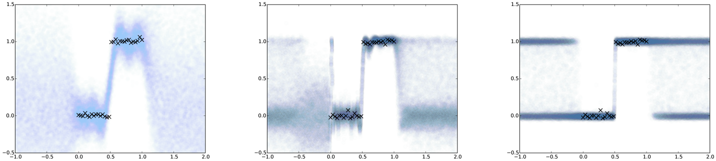
The traditional approach to deep learning is based around binary latent variables and the restricted Boltzmann machine (RBM) [Hinton, 2010]. Deep hierarchies are constructed by stacking these models and various approximate inference techniques (such as contrastive divergence)

# Collapsed Deep GPs

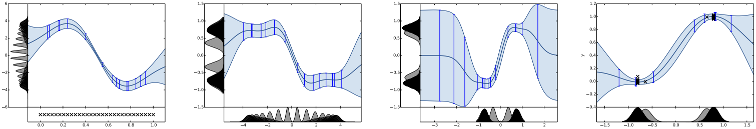


- By sustaining explicity distributions over inducing variables James Hensman has developed a collapsed GP.
- Exciting thing: it mathematically looks like a deep neural network, but with inducing variables in the place of basis functions.
- Additional complexity control term in the objective function.

# Derivative Tails Increase with Layers: Step Function

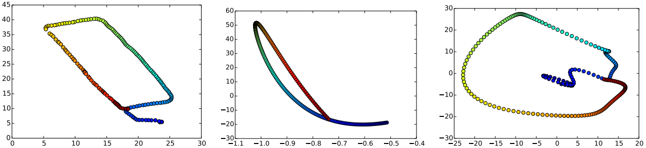


(a) GP (b) 2 layers (c) 4 layers



(d) Hidden spaces for 4 layer model

# Loop Detection in Robotics




(e) True path (f) Hidden layer 1 (g) Hidden layer 2

- Dynamically constrained model
- Correctly detects the loop
- Learns temporal continuity and corner-like features in different layers

# Data fit for Loop Closure

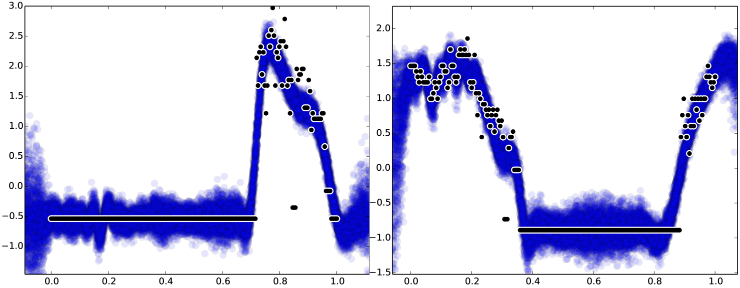
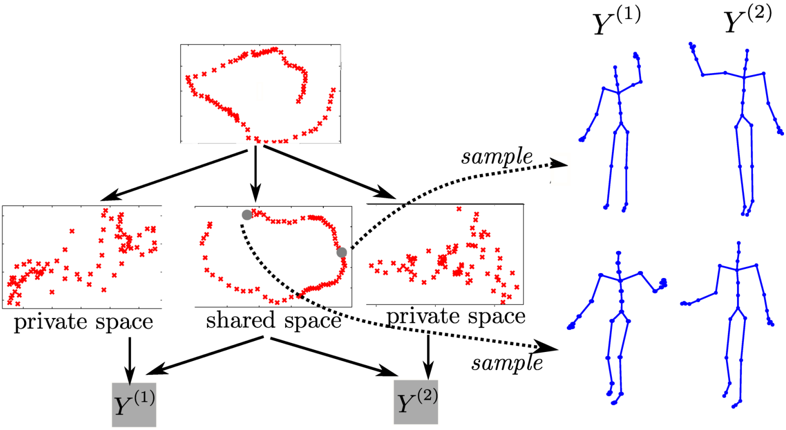



Figure : Example data fits for 2 of the 30 output dimensions

# Motion Capture

- ‘High five’ data.
- Model learns structure between two interacting subjects.

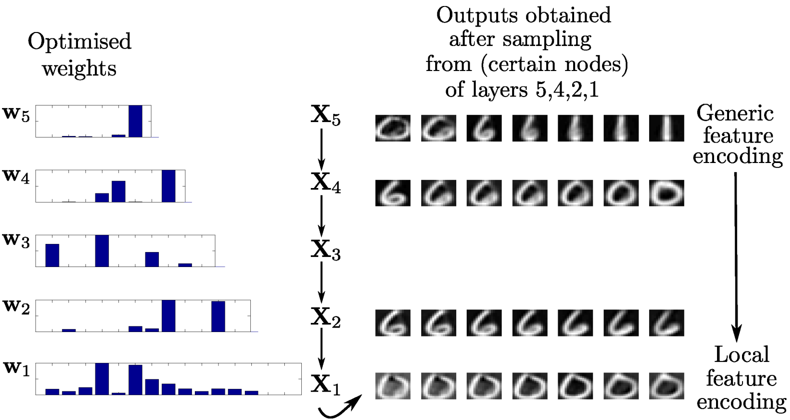
# Deep hierarchies – motion capture



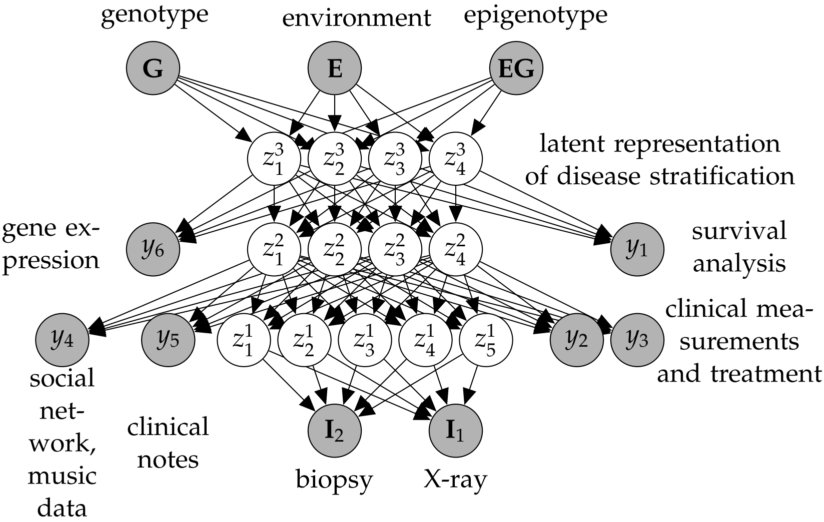
# Digits Data Set

- Are deep hierarchies justified for small data sets?
- We can lower bound the evidence for different depths.
- For 150 6s, 0s and 1s from MNIST we found at least 5 layers are required.

# Deep hierarchies – MNIST



# Deep Health


genotype
environment
epigenotype
G
E
EG
latent representation of disease stratification
gene expression
survival analysis
clinical measurements and treatment
social network, music data
clinical notes
biopsy
X-ray

# Summary

- Deep Gaussian Processes allow unsupervised and supervised deep learning.
- They can be easily adapted to handle multitask learning.
- Data dimensionality turns out to not be a computational bottleneck.
- Variational compression algorithms show promise for scaling these models to *massive* data sets.

# References I

M. A. Álvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In Y. W. Teh and D. M. Titterington, editors, *Proceedings of the Thirteenth International Workshop on Artificial Intelligence and Statistics*, volume 9, pages 25–32, Chia Laguna Resort, Sardinia, Italy, 13-16 May 2010. JMLR W&CP 9. [PDF].

A. Damianou and N. D. Lawrence. Deep Gaussian processes. In C. Carvalho and P. Ravikumar, editors, *Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics*, volume 31, AZ, USA, 2013. JMLR W&CP 31. [PDF].

D. Duvenaud, O. Rippel, R. Adams, and Z. Ghahramani. Avoiding pathologies in very deep networks. In S. Kaski and J. Corander, editors, *Proceedings of the Seventeenth International Workshop on Artificial Intelligence and Statistics*, volume 33, Iceland, 2014. JMLR W&CP 33.

Y. Gal, M. van der Wilk, and C. E. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. *arXiv:1402.1389*, 2014.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In A. Nicholson and P. Smyth, editors, *Uncertainty in Artificial Intelligence*, volume 29. AUAI Press, 2013. [PDF].

N. D. Lawrence. Learning for larger datasets with the Gaussian process latent variable model. In M. Meila and X. Shen, editors, *Proceedings of the Eleventh International Workshop on Artificial Intelligence and Statistics*, pages 243–250, San Juan, Puerto Rico, 21-24 March 2007. Omnipress. [PDF].

M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelfth International Workshop on Artificial Intelligence and Statistics*, volume 5, pages 567–574, Clearwater Beach, FL, 16-18 April 2009. JMLR W&CP 5.