# Deep Gaussian Processes for Regression using Approximate Expectation Propagation

Thang D. Bui[1], José Miguel Hernández-Lobato[2], Daniel Hernández-Lobato[3], Yingzhen Li[1], Richard E. Turner[1]

University of Cambridge[1], Harvard University[2], Universidad Autónoma de Madrid[3]

## 1 - Contributions

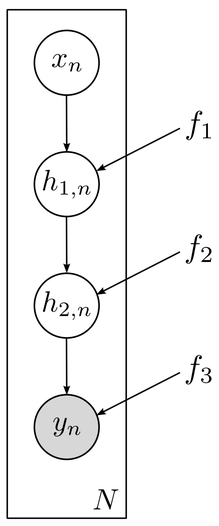
- **Deep GPs for large scale regression**: **scalable** + **flexible** + **calibrated**, largest dataset: **500k** datapoints, **90**-dimensional inputs
- **Novel approx. inference technique**: direct optimisation of the EP energy
- **Rigorous experiments**: an extensive comparison to GPs and Bayesian neural networks, demonstrating state-of-the-art performance of DGPs on many datasets
- **Code**: http://github.com/thangbui/deepGP_approxEP

## 3 - Deep Gaussian processes for regression

Deep GPs are

- multi-layer generalisation of Gaussian processes, hence,
- equivalent to deep neural networks with infinitely wide hidden layers

$$f_l \sim \mathcal{GP}(0, k(.,.))$$
$$h_{l,n} := f_l(f_{l-1}(\cdots f_1(\mathbf{x}_n)))$$
$$y_n = g(\mathbf{x}_n) = f_L(f_{L-1}(\cdots f_2(f_1(\mathbf{x}_n)))) + \epsilon_n$$




**Advantages** of Deep GPs:

- discover useful input warping or dimensionality compression/expansion, i.e. automatic, nonparametric Bayesian kernel design,
- give a non-Gaussian functional mapping $g$,
- repair the damage done by using sparse approximations to GPs,
- support approximate Bayesian inference, and
- give better uncertainty estimates.

## 4 - Direct optimisation of EP energy

Approximate marginal likelihood given by EP:

$$\log p(\mathbf{Y}) \approx \mathcal{F}(\theta) = \phi(\theta) - \phi(\theta_{\text{prior}}) + \sum_{n=1}^{N} \left[\log \mathcal{Z}_n + \phi(\theta^{\backslash n}) - \phi(\theta)\right],$$

where $\log \mathcal{Z}_n = \log \int \mathrm{d}f\ q^{\backslash n}(f)\, p(y_n|f, \mathbf{x}_n)$; $\phi(\theta) = \int \mathrm{d}f\ p(f_{\neq \mathbf{u}}|\mathbf{u}) \exp[\Phi^\intercal(\mathbf{u})\theta]$

## 7 - Comparison with Gaussian processes and Bayesian neural networks

**Task**: regression on 10 datasets from the UCI repository (8 have 20 train/test splits)
The largest dataset has **500k** datapoints and **90** dimensional inputs.

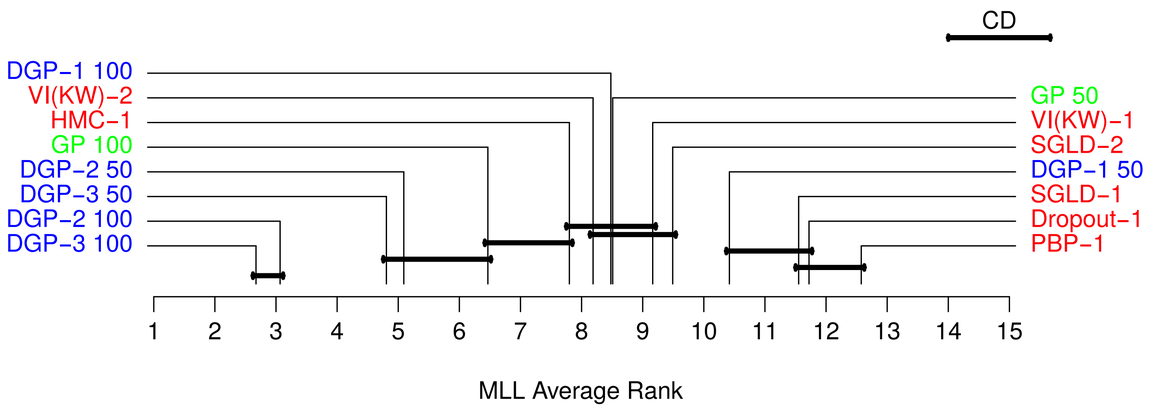



Figure : Rankings of all methods (lower is better).

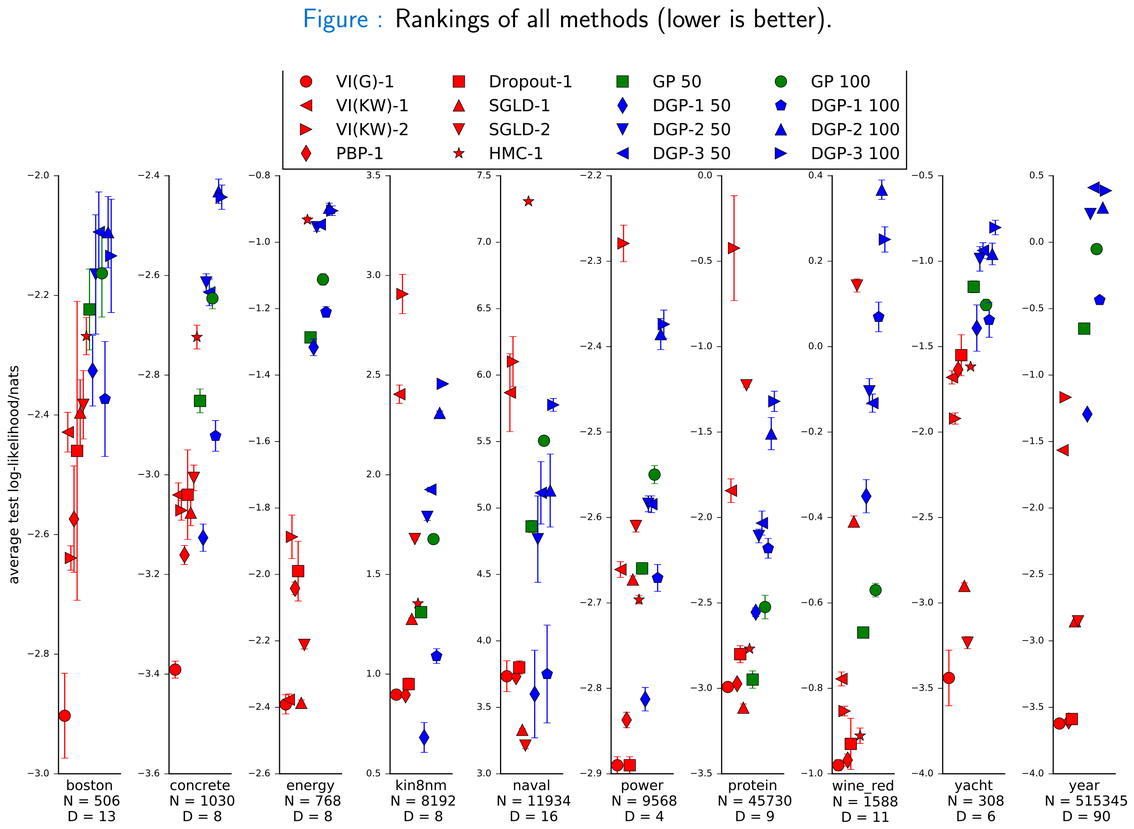



Figure : Average test log-likelihood (higher is better).

## 2 - A motivating example

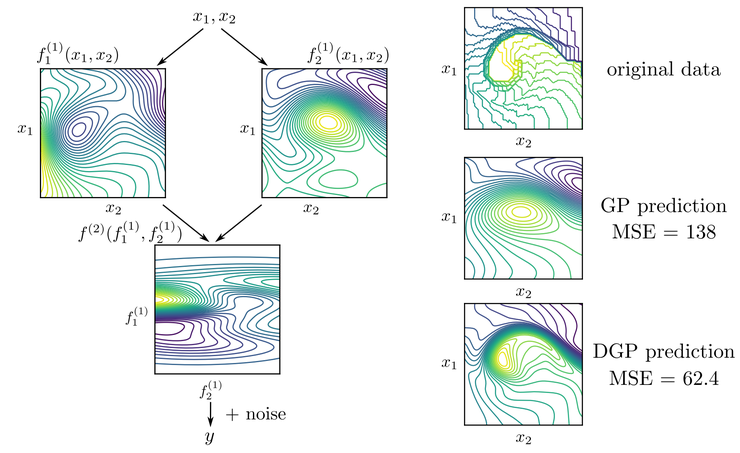
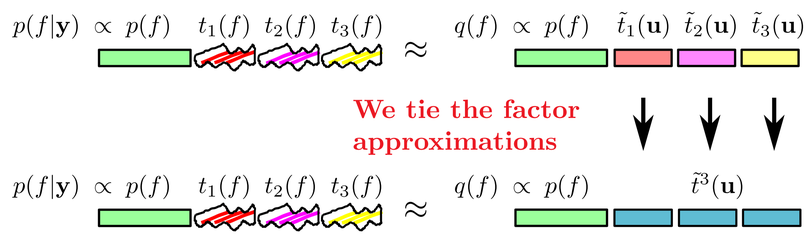



Figure : Fitting the value function of the 'mountain car' problem

## 5 - Approximate EP with tied approximate factors

The approximate posterior is parameterised by **pseudo-points**, $\mathbf{u}$ and following [Li et al. 2015], we tie the factor aproximations:




## 6 - Nested Gaussian projections

Computing $\log \mathcal{Z}_n$ is challenging due to the uncertainty in the inputs of the GP predictive distribution. We approximate this by a recursive Gaussian projection and the projection for each layer is demonstrated in the figure below.

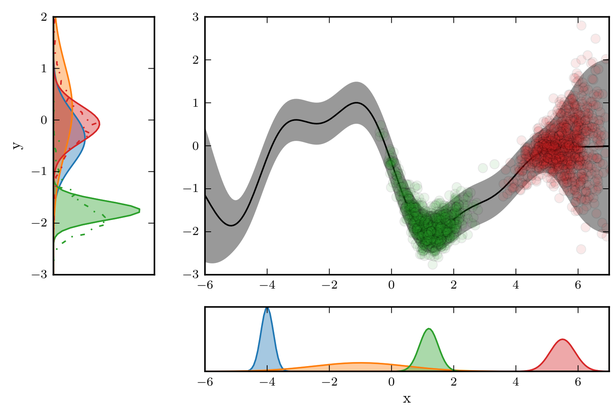
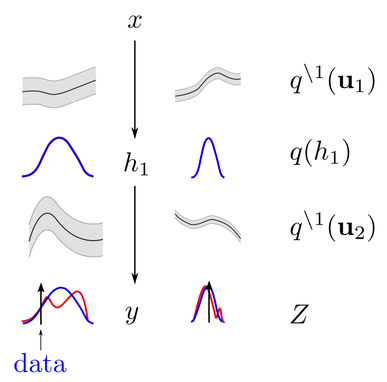


Figure : Gaussian projection: The bottom figure shows the distribution over the inputs of a layer, resulting in non-Gaussian output distributions (left, dash lines) which are then approximated by Gaussians.

The mean and variance of each approximate Gaussian can be computed analytically, for example, for an input distribution $q(h_1)$, the approximate output distribution is $q(h_2) = \mathcal{N}(h_2; m_2, v_2)$ where

$$m_2 = \mathrm{E}_{q(h_1)}[m_{2|h_1}] = \mathrm{E}_{q(h_1)}[\mathbf{K}_{h_2,\mathbf{u}_2}]\mathbf{A}$$
$$v_2 = \mathrm{E}_{q(h_1)}[v_{2|h_1}] + \mathrm{var}_{q(h_1)}[m_{2|h_1}]$$
$$= \sigma_2^2 + \mathrm{E}_{q(h_1)}[K_{h_2,h_2}] + \mathrm{tr}\left(\mathbf{B}\mathrm{E}_{q(h_1)}[\mathbf{K}_{\mathbf{u}_2,h_2}\mathbf{K}_{h_2,\mathbf{u}_2}]\right) - m_2^2$$

and $\mathbf{A} = \mathbf{K}_{\mathbf{u}_2,\mathbf{u}_2}^{-1}\mathbf{m}_2^{\backslash 1}$, $\mathbf{B} = \mathbf{K}_{\mathbf{u}_2,\mathbf{u}_2}^{-1}(\mathbf{V}_2^{\backslash 1} + \mathbf{m}_2^{\backslash 1}\mathbf{m}_2^{\backslash 1,\mathrm{T}})\mathbf{K}_{\mathbf{u}_2,\mathbf{u}_2}^{-1} - \mathbf{K}_{\mathbf{u}_2,\mathbf{u}_2}^{-1}$.
We repeat the above step for each layer, compute $\log \mathcal{Z}$ at the last layer and find its gradient using **back-propagation**.




## 8 - Predicting the efficiency of organic photovoltaic molecules

**Dataset:** 50k/10k training/test points, 512-dim. binary input features. Need calibrated error-bars for active learning or Bayesian optimisation

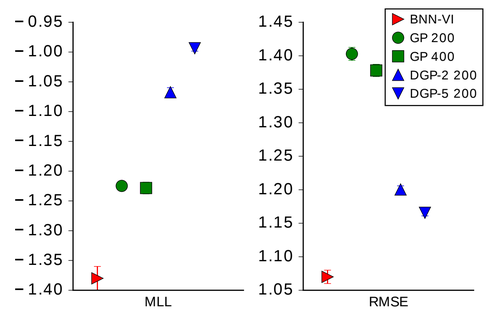



Figure : Average test log-likelihood [higher is better] and test error [lower is better].

## Summary

- **Deep GPs are state-of-the-art for regression**
- **Tying the factor approximations allows direct optimisation of the EP energy**