Deep Learning: A Statistical Perspective

Pietro Lesci

Motivation and Contributions

The aim of my thesis is to provide a fresh and aware statistical view of a subset of modern Machine Learning algorithms named *Deep Learning*.

Main contributions:

- Provide a statistical view of a new hot topic in the field of AI: *Meta-Learning*
- Analyse a recently proposed class of models employed in meta-learning: *Neural Processes*[1]
- Contribute a Python package, `NeuralProcesses`, written in `PyTorch`

[1] Garnelo et al. (2018); Kim et al. (2019).

Agenda

1. Introduction
2. Meta-Learning
3. Neural Processes
4. Experiments

Introduction

Problems and Solutions

We focus on prediction problems, i.e. we want to reason about $p(y|x)$.

In general, we seek a function f that underlies the predictive relationship between inputs and outputs, i.e. that emulates the mechanics of nature

$$x \longrightarrow \boxed{\text{Nature}} \longrightarrow y$$

Usually f is impossible to obtain, therefore the goal is to find a useful approximation $\hat{f}$:

- Linear models (linear regression, ridge regression, lasso, etc.)
- Basis expansions (splines, wavelets, neural networks, etc.)
- Kernel methods and Local regression (SVM, PCA, etc.)
- Nearest-Neighbour methods

Deep Learning

We observe the input-output pairs $\{(y_i, x_i)\}$, $i = 1, \ldots, N$.

Linear regression: We assume a linear input-output relation

$$y_i = x_i\beta + \varepsilon_i$$

Linear basis function regression: Inputs fed through a set of fixed scalar-valued nonlinear transformations, $\Phi_i = [\phi_1(x_i), \ldots, \phi_K(x_i)]$, often assumed to be *fixed* and *mutually orthogonal*

$$y_i = \Phi_i\beta + \varepsilon_i$$

Neural Networks: Basis functions are *parametrized*, i.e. adaptive, not fixed. A scalar-valued nonlinear function is applied to the affine transformation of the inputs, that is, $\phi_k^{w,b}(x_i) = \varphi(w_k x_i + b_k)$

Deep Neural Nets: Hierarchy of parametrised basis functions. For the l-th *layer* we can write $\Phi^{(l)} = W^{(l)}\Phi^{(l-1)} + b^{(l)}$, $l = 1, \ldots, L$

$$y_i = W^{(L)}\Phi^{(L)} + b^{(L)} + \varepsilon_i$$

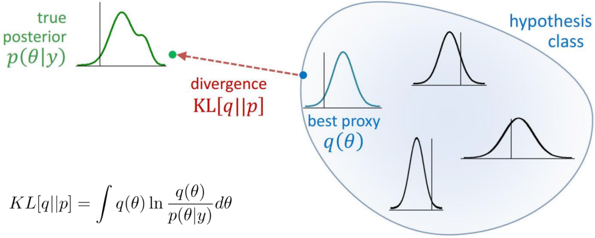
Bayesian Deep Learning

Let θ denote all the parameters in the deep neural net. We want to obtain the posterior distribution

$$p(\theta|x, y) = \frac{p(y|x, \theta)\ p(\theta)}{p(y|x)}$$

Challenges

- Marginalization at the denominator (usually intractable)
- Big data



Variational Inference

VI methods are a family of techniques for approximating intractable integrals arising in Bayesian inference and machine learning. The problem of integration is transformed into optimization.

How it works

- Define a family of approximate densities $\mathcal{Q}$
- Find the member of the family that minimizes

$$
\begin{aligned}
\mathrm{KL}\big[q(\theta)\|p(\theta|y)\big] &= \mathrm{E}\big[\log q(\theta)\big] - \mathrm{E}\big[\log p(\theta|y)\big] \\
&= \mathrm{E}\big[\log q(\theta)\big] - \mathrm{E}\big[\log p(\theta, y)\big] + \log p(y)
\end{aligned}
$$

Problem: It requires the log evidence; optimize an equivalent objective function up to an added constant

$$
\begin{aligned}
\mathrm{ELBO}(q) &= \mathrm{E}\big[\log p(\theta, y)\big] - \mathrm{E}\big[\log q(\theta)\big] \\
&= \mathrm{E}\big[\log p(y|\theta)\big] - \mathrm{KL}\big[q(\theta)\|p(\theta)\big]
\end{aligned}
$$

Meta-Learning

Motivation

Meta-Learning refers to the extraction of domain-general information that can act as an *inductive bias* to improve learning efficiency in novel tasks. It attempts to endow machine learning models with the ability to learn from small data leveraging past experience

Challenges

- Computational costs: Avoid re-training a model from scratch
- Insufficient data: Borrow strength among similar tasks
- Flexibility: Fast adaptation

Goal

Model the relation between inputs and outputs in each condition in a way that satisfies the following two requirements

- Maximize predictive performance on each task
- Leverage shared statistical structure among tasks

Statistical View (1)

Data from multiple experiments $\{(y_{it}, x_{it})\}$, $i = 1, \ldots, N_t$, $t = 1, \ldots, T$

Case 1 (pool)	$y_t = m(x_t) + \varepsilon_t$	$m \sim \mathcal{F}$
Case 2 (indep.)	$y_t = m_t(x_t) + \varepsilon_t$	$m_t \overset{ind}{\sim} \mathcal{F}$
Case 3 (hierar.)	$y_t = m_t(x_t) + \varepsilon_t$	$m_t \sim \mathcal{F}$

Challenge: Specify a BNP prior distribution $\mathcal{F}$

Solution: Assume the existence of a *fixed* function, g_θ, *shared* across tasks, and a set of dependent *latent variables*, $\{z_t\}_{t=1}^T$, encoding the task-specific information

Meta-Learning $\quad y_t = g(x_t, z_t; \theta) + \varepsilon_t$

$$g = \mathrm{NN}(\cdot; \theta)$$
$$\theta \sim p(\theta)$$
$$z_t | \eta \overset{iid}{\sim} p(\cdot|\eta)$$

The two channels through which statistical strength is shared: η and θ.

Statistical View (2)

The fully-Bayesian definition of the model is

$$
\begin{tikzcd}
& \eta \arrow[dl] \arrow[dr] & \\
z_1 \arrow[d] & \cdots & z_T \arrow[d] \\
g(\cdot, z_1; \theta) \arrow[d] & \cdots & g(\cdot, z_T; \theta) \arrow[d] \\
y_{11}, \dots, y_{N_1 1} & \cdots & y_{1T}, \dots, y_{N_T T}
\end{tikzcd}
$$

$$
\begin{aligned}
&\eta \sim p(\eta) \\
&z_t | \eta \overset{iid}{\sim} p(\cdot | \eta) \\
&g = \mathrm{NN}_\theta, \ \theta \sim p(\theta) \\
&y_{it} | x_{it}, z_t, \theta \overset{ind}{\sim} p(\cdot | x_{it}, z_t, \theta)
\end{aligned}
$$

In practice both η and θ are estimated using Empirical Bayes methods

$$
\begin{aligned}
y_{it} | x_{it}, z_t &\overset{ind}{\sim} p(\cdot | x_{it}, z_t; \hat{\theta}) \\
z_t &\overset{iid}{\sim} p(\cdot; \hat{\eta}) \\
g &= \mathrm{NN}(\cdot; \hat{\theta})
\end{aligned}
$$

Note that the randomness in g depends on the randomness in z.

Neural Processes

The Model

Garnelo et al. (2018) introduced an instance of the class of model described above called **Neural Processes**

$$
\begin{aligned}
y_{it} | x_{it}, z_t &\overset{ind}{\sim} \mathcal{N}\Big(g_\theta(x_{it}, z_t), \sigma^2\Big), \qquad g_\theta = \mathrm{NN}(\cdot; \theta) \\
z_t &\overset{iid}{\sim} \mathcal{N}(0, 1)
\end{aligned}
$$

Goal: Learning a distribution over random functions, i.e. capture the variability of the estimated regression function

Takeaway: Neural Processes are a data-driven alternative to BNP prior distributions like Gaussian Processes. Each dataset, corresponding to a specific task, is interpreted as a sample from one trajectory of the true underlying stochastic process

Implicit Processes

Following Diggle and Gratton (1984)

- **Prescribed** statistical model: parametric specification of the distribution of a random vector
- **Implicit** statistical model: generating stochastic mechanism

Gaussian Processes: $f \sim \mathcal{GP}\big(m(\cdot), k(\cdot)\big)$ if each finite collection $\mathbf{f} = \big(f(x_1), \ldots, f(x_N)\big)$ is defined by the sampling process $\mathbf{f} = Bz + m$, where $z \sim \mathcal{N}(0, 1)$ and $K = BB'$ (Cholesky)

Implicit Stochastic Processes: $f \sim \mathcal{IP}\big(g_\theta(\cdot, z), p_z\big)$ if each finite collection $\mathbf{f} = \big(f(x_1), \ldots, f(x_N)\big)$ is defined by the sampling process $\mathbf{f} = g_\theta(\cdot, z)$, where $z \sim p(z)$ – Ma et al. (2018)

Defining $g_\theta = \mathrm{NN}_\theta$ and $p(z) = \mathcal{N}(0, 1)$ we retrieve the definition of Neural Processes

Inference

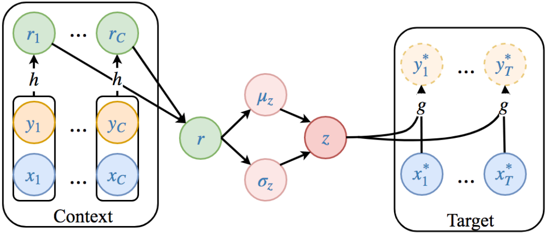
We want to compute the posterior distribution $p(z|x, y)$ that is intractable since g is nonlinear. Inference is performed via amortized VI

The basic idea is to use a powerful predictor to predict the variational parameters, thus replacing them with a function of the data whose parameters are shared across all observations

Procedure

- Define the variational distribution: $q(z) = \mathcal{N}\left(\mu_z, \sigma_z^2 I\right)$
- Parametrize $\mu_z = \mu(x, y)$ and $\sigma_z^2 = \sigma^2(x, y)$
- Minimize ELBO via stochastic gradient descent

Implementation

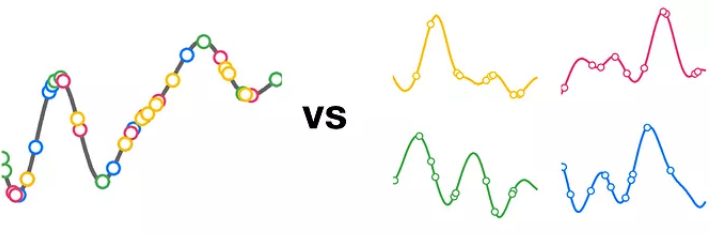


Source: `https://kasparmartens.rbind.io/post/np/`

The Generative model

- The context points $\{(x_i, y_i)\}_{i \in C}$ are mapped through a NN, h, to obtain a latent representation r_i
- The vectors $\{r_i\}_{i \in C}$ are aggregated (in practice: averaged) to obtain a single value r
- The aggregated representation r is used to parametrise the variational distribution of z, i.e. $q(z|x_\mathcal{C}, y_\mathcal{C}) = \mathcal{N}(\mu(r), \sigma^2(r))$
- To obtain a prediction at x_i^*, sample z, and feed them both to g

Training Procedure



Source: https://github.com/deepmind/neural-processes

At each iteration of the learning process

- Select randomly a task, $D_t = \{(x_{it}, y_{it})\}_{i=1}^{N_t}$, also called target set, $\mathcal{T}_t$
- Select randomly the number of context points
- Select randomly a context set, $\mathcal{C}_t = \{(x_{it}, y_{it})\}_{i \in C}$
- Compute the parameters of $q(z|\text{target})$ and $q(z|\text{context})$
- Minimize

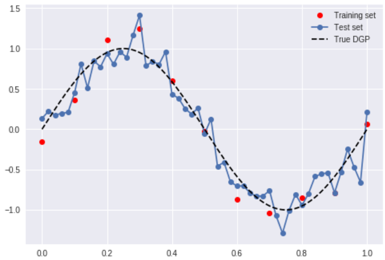
$$\mathrm{ELBO}_{[\mathcal{T}|\mathcal{C}]}(\phi) = \mathrm{E}\big[\log p(y_{\mathcal{T}}|z, y_{\mathcal{C}}, x_{\mathcal{C}}, x_{\mathcal{T}})\big] - \mathrm{KL}\big[q_\phi(z|y_{\mathcal{T}}, x_{\mathcal{T}}) \| q_\phi(z|y_{\mathcal{C}}, x_{\mathcal{C}})\big]$$

Experiments

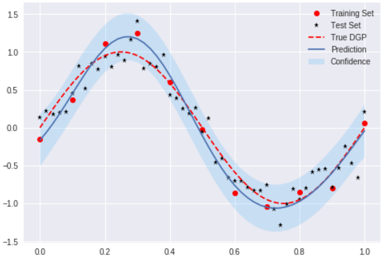
Experiment 1 – Single 1D regression task

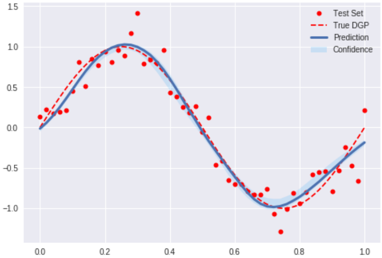
A GP with RBF kernel is used as baseline. The inputs consist of the points in the linear space $[0,1]$, the outputs are generated according to

$$y_i = \sin(2\pi x_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 0.2)$$



Results predicting the same task





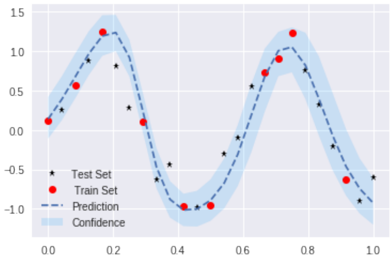
Experiment 2 – Thirty 1D regression tasks

30 different functions generated according to the following equation

$$y_i = \sin(a\pi x_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 0.2)$$

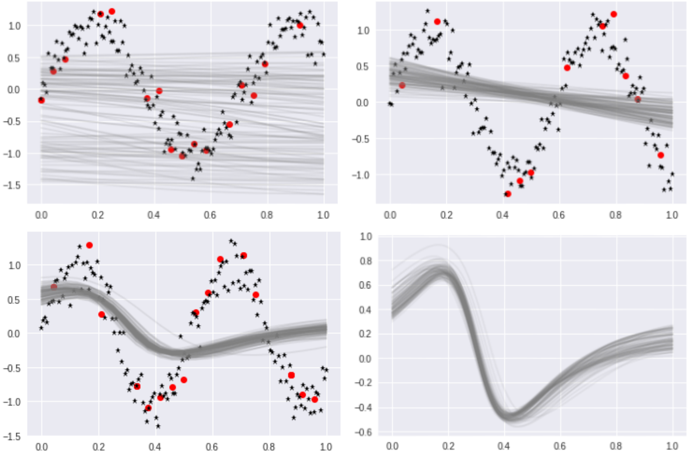
$$a \sim Unif(2, 4)$$

Results predicting unseen task based on 10 input-output pairs



Experiment 2 (2)

Learned-prior distribution after 1, 2000, 4000, 8000 training iterations.



References

Diggle, P. J. and Gratton, R. J. (1984). Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2):193–227.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M. A., and Teh, Y. W. (2018). Neural processes. *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*.

Kim, H., Mnih, A., Schwarz, J., Garnelo, M., Eslami, S. M. A., Rosenbaum, D., Vinyals, O., and Teh, Y. W. (2019). Attentive neural processes. *International Conference on Learning Representations*.

Ma, C., Li, Y., and Hernández-Lobato, J. M. (2018). Variational implicit processes. *ICML Workshop on Bayesian Deep Learning*.