Conditional Neural Processes for Semi-Supervised Learning

Yann Dubois, Richard E. Turner
Department of Engineering, University of Cambridge

UNIVERSITY OF CAMBRIDGE

Introduction

What : Semi-supervised learning (SSL) tackles problems with labelled $(\mathrm{X}_\mathcal{L}, \mathrm{Y}_\mathcal{L}) \coloneqq \{(\mathbf{x}^{(l)}, \mathbf{t}^{(l)})\}_{l=1}^{L}$ and unlabelled points $\mathrm{X}_\mathcal{U} \coloneqq \{\mathbf{x}^{(u)}\}_{u=1}^{U}$. Typically $L \lll U$. The assumption is that decision boundaries pass through low density regions.

Why Important real-life scenarios, as labels are usually expensive and cumbersome to obtain, but state of the art algorithms require large amount of data.

Problem SSL applications are not benefiting from the deep learning revolution as:

- Most research has been restricted to simple computer vision tasks.
- Many deep SSL methods are based on latent-variable models, which tend to under-perform compared to transfer learning due to approximate inference.
- Real-world data often has missing features or comprises of unevenly spaced time series, making them hard to use by deep learning.

Aim : Implement state of the art deep SSL methods in an easy to use framework. As well as, investigate the use of Conditional Neural Processes (CNP) [1] for SSL, which are latent-free, applicable to non uniformly discretized time series, and robust to missing features.

Semi Supervised Learning Framework

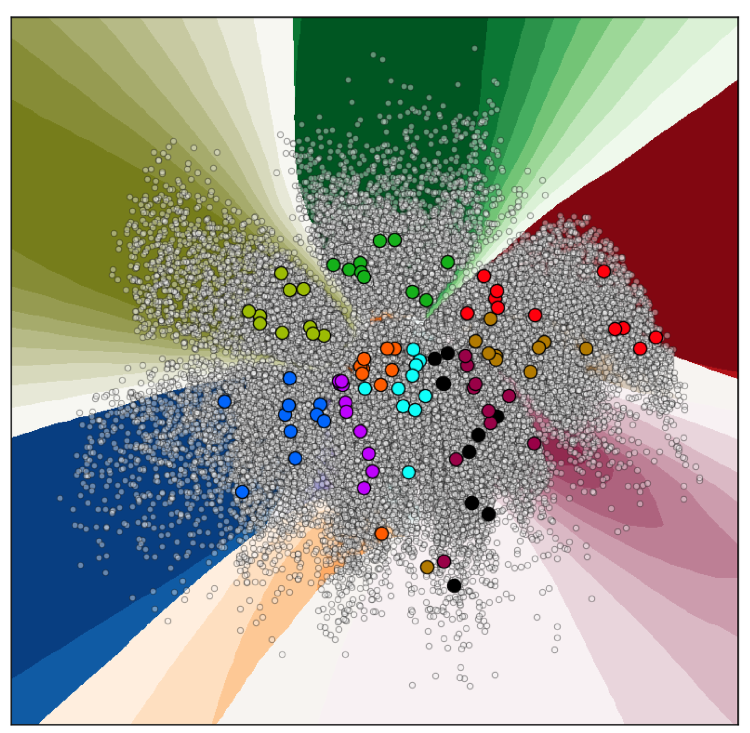


Fig. 1: Label Spreading [2] decision boundaries and confidence, when predicting in a 2 dimensional latent space of a variational auto encoder for 100 labels of MNIST.

In order to help and encourage future work in SSL we have implemented a general framework with many recent deep SSL methods: `github.com/YannDubs/Semi-Supervised-Neural-Processes`. Fig. 1 shows how easy it is to combine different methods and quickly get good results.

References

[1] Marta Garnelo et al. "Conditional neural processes". In: *arXiv preprint arXiv:1807.01613* (2018).

[2] Dengyong Zhou et al. "Learning with local and global consistency". In: *Advances in neural information processing systems*. 2004, pp. 321–328.

[3] Marta Garnelo et al. "Neural processes". In: *arXiv preprint arXiv:1807.01622* (2018).

[4] Hyunjik Kim et al. "Attentive Neural Processes". In: *arXiv preprint arXiv:1901.05761* (2019).

Neural Processes Family

A CNP is a neural model $Q_{\boldsymbol\theta}$, inspired by conditional stochastic processes, which predict the target posterior at $\mathrm{X}_\mathcal{T} \coloneqq \{\mathbf{x}^{(t)}\}_{t=1}^{T}$ conditioned on context points $(\mathrm{X}_\mathcal{C}, \mathrm{Y}_\mathcal{C}) \coloneqq \{(\mathbf{x}^{(c)}, \mathbf{y}^{(c)})\}_{c=1}^{C}$:

$$p(\mathrm{Y}_\mathcal{T}|\mathrm{X}_\mathcal{T}, \mathrm{X}_\mathcal{C}, \mathrm{Y}_\mathcal{C}) \overset{(a)}{\approx} p(\mathrm{Y}_\mathcal{T}|\mathrm{X}_\mathcal{T}, \mathbf{r}_\mathcal{C}) \overset{(b)}{\approx} \prod_{t=1}^{T} Q_{\boldsymbol\theta}(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, \mathbf{r}_\mathcal{C}) \qquad (1)$$

$$\mathbf{r}_\mathcal{C} \coloneqq \bigoplus_{c=1}^{C} h_{\boldsymbol\theta}(\mathbf{x}^{(c)}, \mathbf{y}^{(c)})$$

$$\boldsymbol\theta^* = \arg\min_{\boldsymbol\theta} -\mathbb{E}_f\left[\mathbb{E}_{\mathrm{X}_\mathcal{T}}\left[\mathbb{E}_{\mathrm{X}_\mathcal{C},\mathrm{Y}_\mathcal{C}}\left[\log Q_{\boldsymbol\theta}\left(\mathrm{Y}_\mathcal{T}|\mathrm{X}_\mathcal{T}, \mathrm{X}_\mathcal{C}, \mathrm{Y}_\mathcal{C}\right)\right]\right]\right]$$

- $\bigoplus$ denotes any commutative operation, thereby enforcing permutation invariance in the context $(\mathrm{X}_\mathcal{C}, \mathrm{Y}_\mathcal{C})$.
- The fix dimensional representation $\mathbf{r}_\mathcal{C}$ assumption (Eq. 1a) gains in scalability: $O(C+T)$ but makes the model non consistent. I.e. auto-regressive prediction give different results than predicting at once.
- The factorisation assumption (Eq. 1b) enforces permutation invariance in $\mathrm{X}_\mathcal{T}$.
- CNP require training on a distribution over functions f.
- This framework has been extended to have latent variable [3] and use target dependent attention of the context [4]. Fig. 2 shows how these model compare.

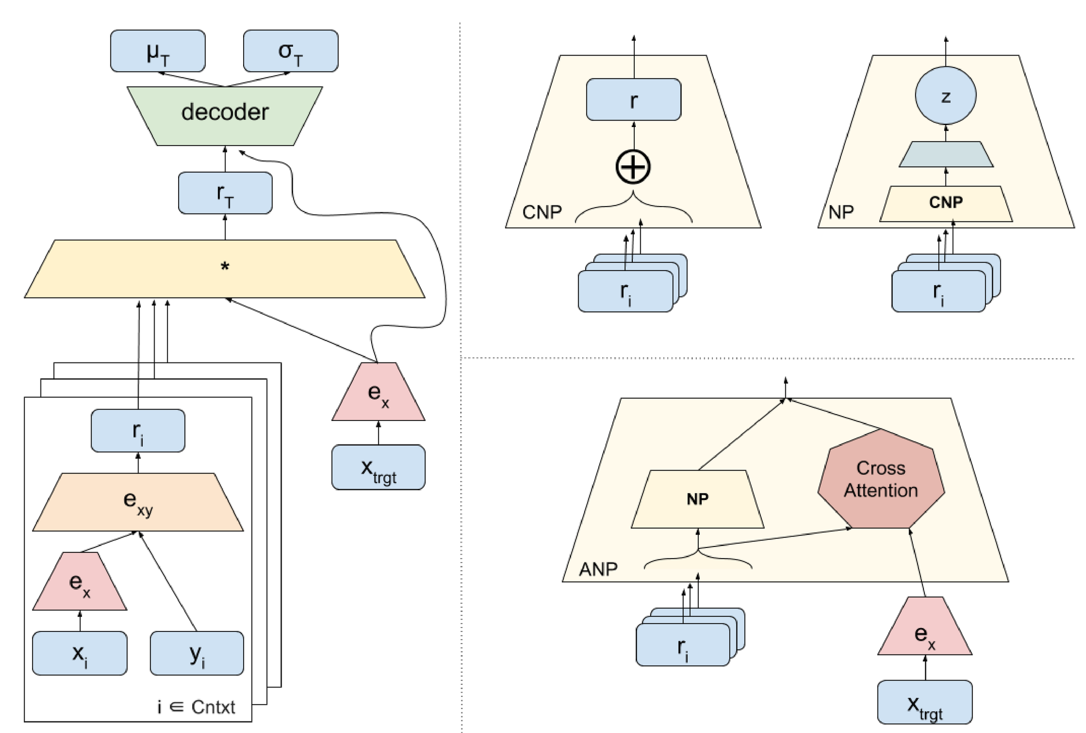
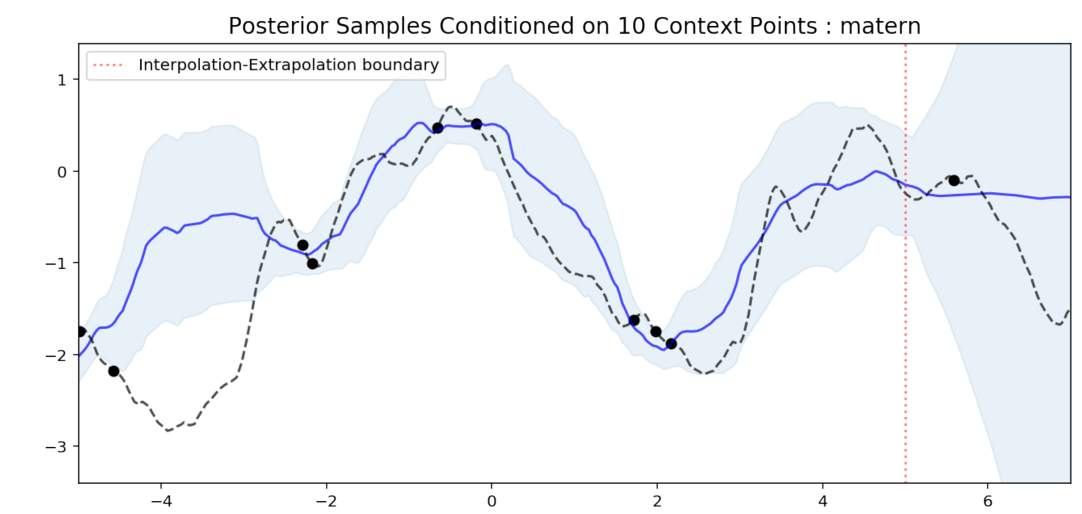


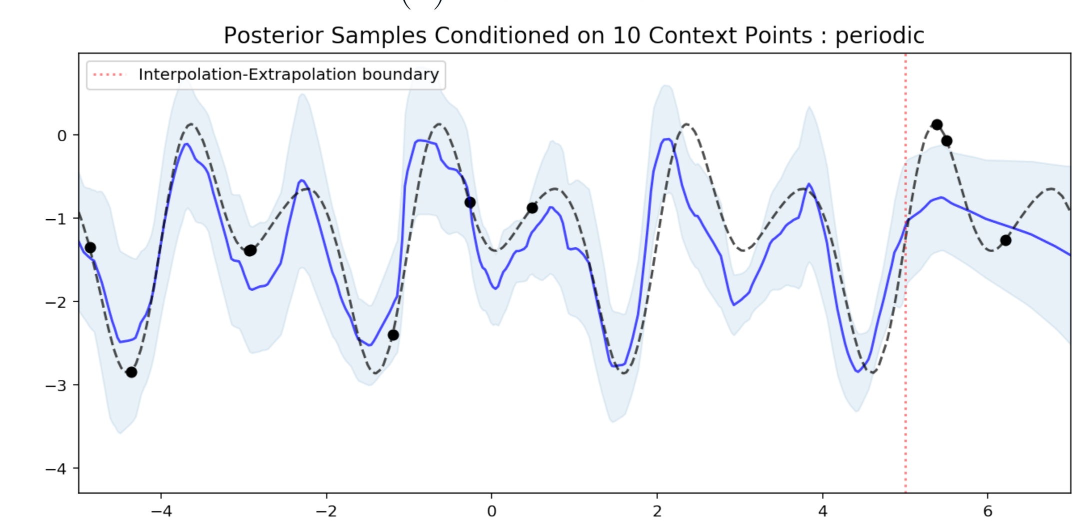
Fig. 2: Left) Computational graph of the general Neural Process framework. Right) Zoom in on the * module, incorporating the Conditional Neural Process[1] (top center), the latent Neural Process [3] (top right) and the Attentive Neural Process [4] (bottom).

Initial Experiments

Fig. 3 shows that the target function is nearly completely contained in the region of uncertainty of the model. Despite this, the model seems to underfit as it does not pass through all context points and always breaks in the extrapolation setting.



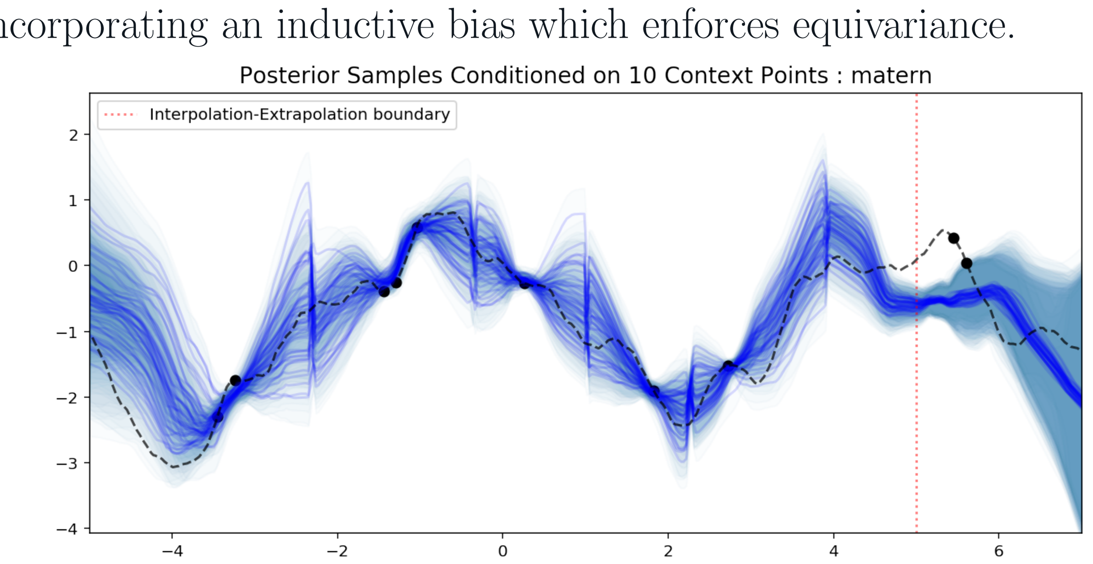
(a) Matern Kernel



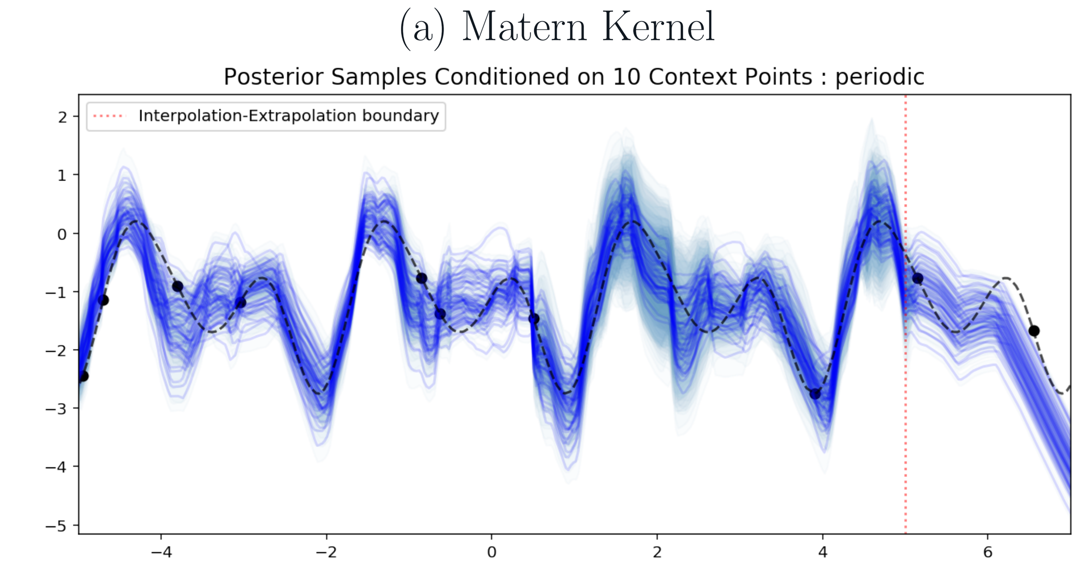
(b) Exp. Sin. Squared Kernel (Periodic)

Fig. 3: Approximations of Gaussian Processes (GP) using Conditional Neural Processes (CNP). During training, samples from a GP with a fixed kernel were generated and a random number of points were used as target and context. The dotted black line shows the target sample from the GP, while the black dots are context points. The blue region depicts the mean and standard deviation predicted by the CNP. The red dotted line separates the interpolation and extrapolation setting.

Fig. 4 shows how attention improves model compared to Fig. 3. Although the model has a much better fit, it is still unable to extrapolate, which is one of issues we want to solve by incorporating an inductive bias which enforces equivariance.



(a) Matern Kernel



(b) Exp. Sin. Squared Kernel (Periodic)

Fig. 4: Attentive Neural Processes (ANP) version of Fig. 3. Note that ANP have latent variables and can thus sample different mean functions at once (the various blues lines).