GPflux: A Library for Deep Gaussian Processes

Vincent Dutordoir[1,2], Hugh Salimbeni[2], Eric Hambro[2], John McLeod[2], Felix Leibfried[2], Artem Artemev[2,3], Mark van der Wilk[3], James Hensman[2], Marc P. Deisenroth[4], ST John[2]

[1]University of Cambridge, [2]Secondmind.ai, [3]Imperial College London, [4]University College London

Correspondence to V. Dutordoir <vd309@cam.ac.uk>

Summary

GPflux is a toolbox for Deep Gaussian processes. Combining the mathematical building blocks from GPflow with the tools from TensorFlow /Keras leads to a framework that can be used for:

- researching new (deep) Gaussian process models,
- and building, training, evaluating and deploying (deep) Gaussian processes in a modern way — making use of the tools developed by the deep learning community.

Deep Gaussian Processes

The hierarchical extension of Gaussian processes (GP):

$$y = f_L(f_{L-1}(\ldots, f_2(f_1(x)))) + \varepsilon, \quad \text{where} \quad f_\ell \sim \mathcal{GP}(0, k_\ell)$$

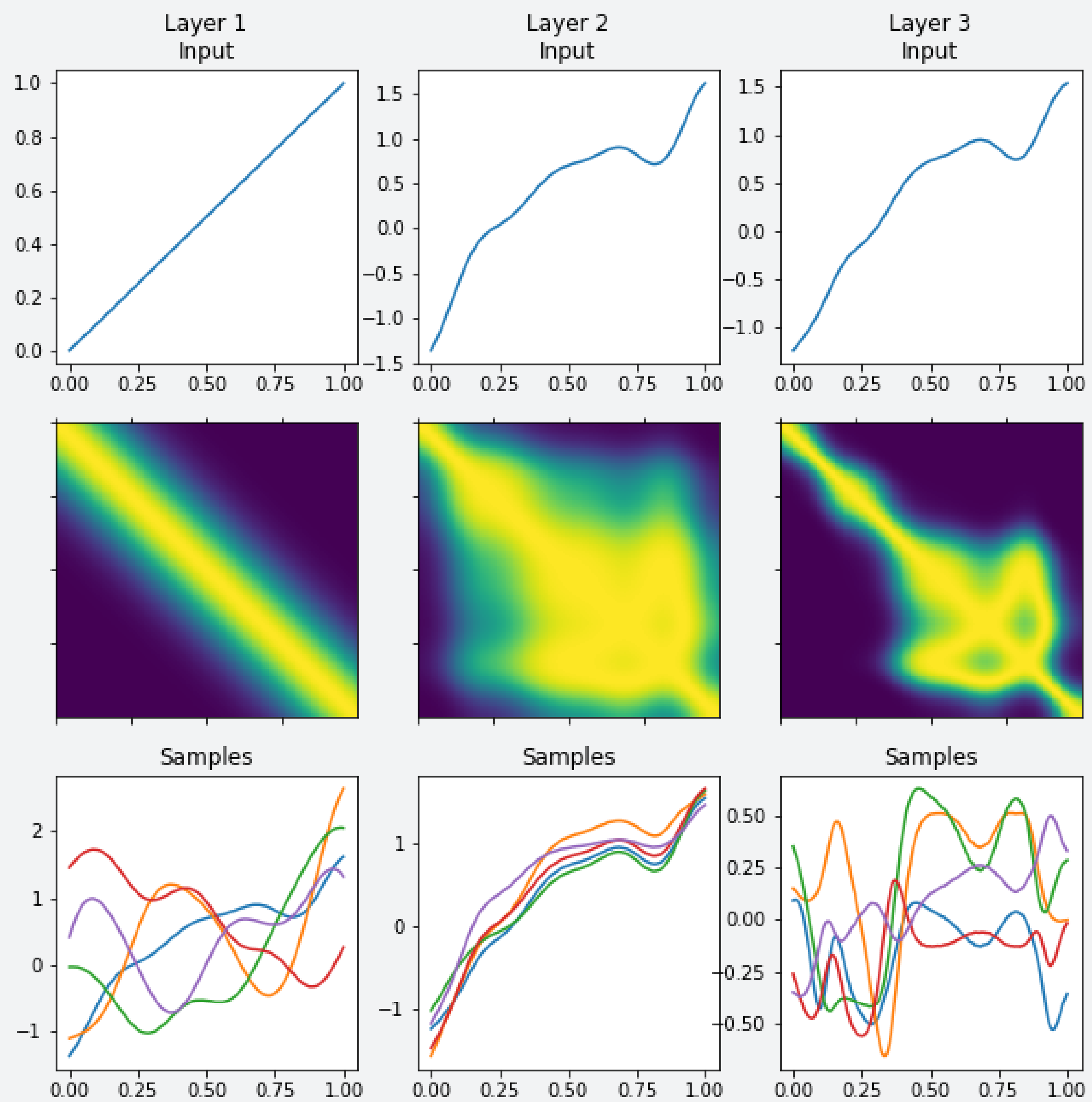

Figure 1: Building complex covariances by composing GPs

Code snippet

GP Layer:

```
gp_layer = gpflux.layers.GPLayer(
   kernel=gpflow.kernels.SquaredExponential(),
   inducing_variable=gpflow.inducing_variables.InducingPoints(X[:m]),
   num_data=len(X)
)
```

Initialising, Training and Evaluating a model:

```
# Initialise a 4-layer model consisting of NN layers and GP layers
model = Sequential([tf.keras.Dense(...), tf.keras.Convolution(...), gp_layer])
model.compile(loss=LikelihoodLoss(Gaussian()), optimizer="Adam")
# Fitting
callbacks = [ReduceLROnPlateau(), TensorBoard(), ModelCheckpoint()]
model.fit(X, Y, callbacks=callbacks)
# Evaluating
model.predict(X)
```

Example

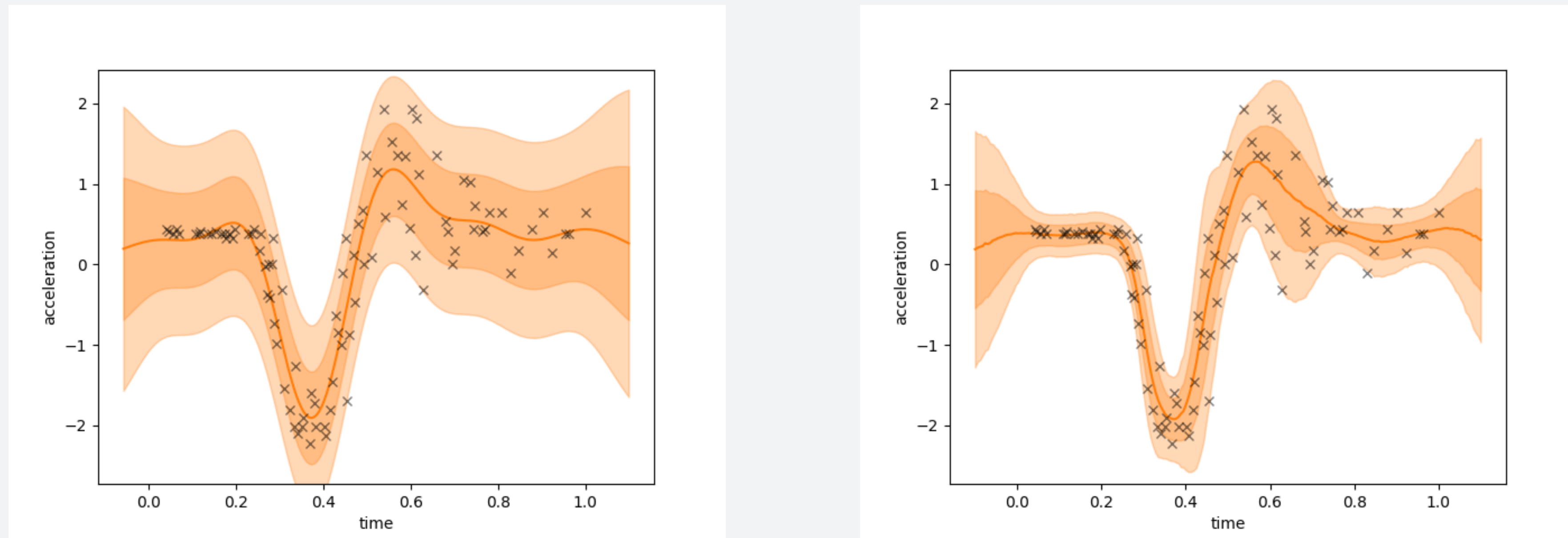

Figure 2: (left) A single layer GP model. **(right)** Deep Gaussian process latent variable model.

References

[1] A. Damianou and N. D. Lawrence. Deep Gaussian processes. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[2] H. Salimbeni and M. P. Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems 30 (NIPS)*.