Supplementary Material: Deep Kernel Learning

Andrew Gordon Wilson[*]
CMU

Zhiting Hu[*]
CMU

Ruslan Salakhutdinov
University of Toronto

Eric P. Xing
CMU

1 Experimental Results

1.1 Recovering step function

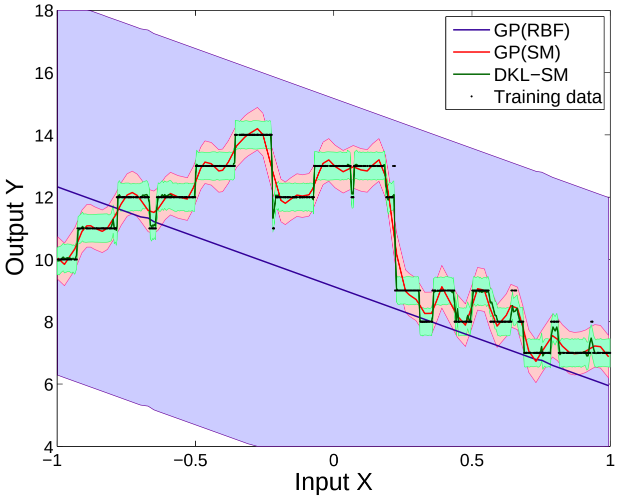

Figure 1: Recovering step function. We show the predictive mean and 95% of the predictive probability mass for regular GPs with RBF and SM kernels, and DKL with SM base kernel. We set $Q = 4$ for SM kernels.

We test the ability of DKL to recover step functions, a challenging benchmark problem for kernel learning methods due to the underlying smoothness assumptions. We consider a particularly complicated step function with many discontinuities. The training data consists of 60,000 points (shown as black dots in Figure 1) and the test data contains 2,000 points, both uniformly distributed over $(-1, 1)$. The DKL-SM model uses a DNN with an 1-1000-1000-500-50-2 architecture, which is the same as what used in the UCI regression tasks (section 5.1).

Because of the number of discontinuities in this step function, and the strong smoothness assumptions in the RBF kernel, most of the structure in the data is discounted as noise. The GP with SM kernel performs much better, but is unable to capture the sharpness of the discontinuities. By contrast, the DKL-SM model accurately encodes the discontinuities of the function, and has reasonable uncertainty on the whole domain. Indeed, the ability for DKL to easily output predictive uncertainties, or even a whole posterior predictive distribution, through the Gaussian process is an additional advantage over stand-alone deep learning architectures. For this reason, DKL could naturally be implemented in reinforcement learning problems, or in Bayesian optimisation, where predictive distributions (rather than point predictions) have particular practical value.

1.2 Convolutional network architecture

Table 1 lists the architecture of the convolutional networks used in the tasks of face orientation extraction (section 5.2) and digit magnitude extraction (section 5.3). The CNN architecture is original from the LeNet LeCun et al. (1998) (for digit classification) and adapted to the above tasks with one or two more fully-connected layers for feature transformation.

References

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Layer	conv1	pool1	conv2	pool2	full3	full4	full5	full6
kernel size	5×5	2×2	5×5	2×2	-	-	-	-
stride	1	2	1	2	-	-	-	-
channel	20	20	50	50	1000	500	50	2

Table 1: The architecture of the convolutional network used in face orientation extraction. The CNN used in the MNIST digit magnitude regression has a similar architecture except that the *full3* layer is omitted. Both *pool1* and *pool2* are max pooling layers. ReLU layer is placed after *full3* and *full4*.