Scalable GP-LSTMs with Semi-Stochastic Gradients

Maruan Al-Shedivat
Carnegie Mellon University
alshedivat@cs.cmu.edu

Andrew Gordon Wilson
Cornell University
andrew@cornell.edu

Yunus Saatchi
saatchi@cantab.net

Zhiting Hu
Carnegie Mellon University
zhitingh@cs.cmu.edu

Eric P. Xing
Carnegie Mellon University
epxing@cs.cmu.edu

1 Introduction

There exists a vast array of machine learning applications where the underlying datasets are sequential. Applications range from the entirety of robotics, to speech, audio and video processing. While neural-network-based approaches have dealt with the issue of *representation learning* for sequential data, the important question of modeling and propagating uncertainty across time has rarely been addressed by these models. For a robotics application such as a self-driving car, however, it is not just desirable, but essential to have complete predictive densities for variables of interest. When trying to stay in lane and keep a safe following distance from the vehicle front, knowing the uncertainty associated with lanes and lead vehicles is as important as the point estimates.

Recurrent models with long short-term memory (LSTM) [4] have recently emerged as the leading approach to modeling sequential structure. LSTM cells use a gating mechanism that stabilizes the flow of the back-propagated errors and hence improve the learning process of the model. While the LSTM already provides state-of-the-art results on speech and text data [3, 6], quantifying uncertainty or extracting full predictive distributions from deep models models is an area of active research [2].

In this paper, we quantify the predictive uncertainty of deep models by following a Bayesian nonparametric approach. In particular, we propose kernel functions which fully encapsulate the structural properties of LSTMs, for use with Gaussian processes. The resulting model enables Gaussian processes to achieve state-of-the-art performance on sequential regression tasks, while also allowing for a principled representation of uncertainty, and non-parametric flexibility. For scalability, we use semi-stochastic optimization and exploit the algebraic structure of these kernels, decomposing the relevant covariance matrices into Kronecker products of circulant matrices, for $\mathcal{O}(n)$ training time and $\mathcal{O}(1)$ test predictions [8]. Our model not only can be interpreted as a Gaussian process with a recurrent kernel, but also as a deep recurrent network with probabilistic outputs, infinitely many hidden units, and a utility function robust to overfitting.

More details can be found in the full version of this paper available on arXiv [1]. Our code is available at http://github.com/alshedivat/kgp/.

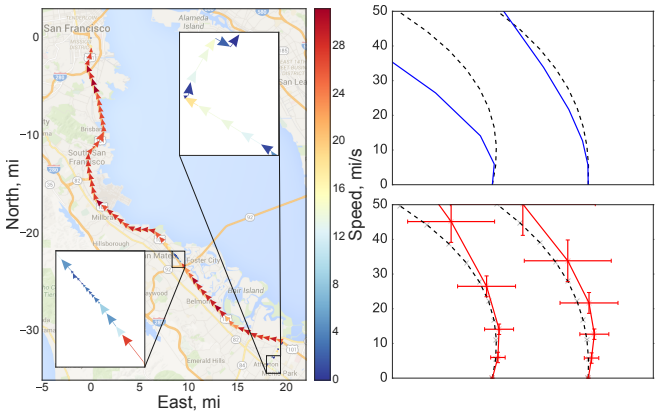
2 Methods

We consider the problem of learning a regression function that maps sequences to real-valued targets. Formally, let $\bar{\boldsymbol{X}} = \{\bar{\boldsymbol{x}}_i\}_{i=1}^n$ be a collection of sequences, $\bar{\boldsymbol{x}}_i = [\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \cdots, \boldsymbol{x}^{l_i}]$, $\boldsymbol{x}_i^j \in \mathcal{X}$. Let $\boldsymbol{y} = \{y_i\}_{i=1}^n$, $y_i \in \mathbb{R}$, be a collection of corresponding real-valued targets. Assuming that the maximum length of a sequence is L, the goal is to learn a function, $f : \mathcal{X}^L \mapsto \mathbb{R}$, from some family, $\mathcal{F}$, based on the available data.

We propose to use Gaussian processes (GPs) with LSTM-structured kernel functions:

$$\tilde{k}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}') = k(\phi(\bar{\boldsymbol{x}}), \phi(\bar{\boldsymbol{x}}')), \text{ where } \bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}' \in \mathcal{X}^L, \text{ and } \tilde{k} : (\mathcal{X}^L)^2 \mapsto \mathbb{R}.$$

Workshop on Bayesian Deep Learning, NIPS 2016, Barcelona, Spain.



Algorithm 1 Semi-stochastic gradient descent with delayed kernel updates.

input Data – $(\boldsymbol{X}, \boldsymbol{y})$, kernel – $k_{\boldsymbol{\theta}}(\cdot,\cdot)$, recurrent transformation – $\phi_{\boldsymbol{w}}(\cdot)$.
1: Initialize $\boldsymbol{\theta}$ and $\boldsymbol{w}$; compute initial K.
2: **repeat**
3: $\quad\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \texttt{upd}_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{w}, K)$.
4: $\quad$**for all** mini-batches $\boldsymbol{X}_b$ in $\boldsymbol{X}$ **do**
5: $\quad\quad\boldsymbol{w} \leftarrow \boldsymbol{w} + \texttt{upd}_{\boldsymbol{w}}(\boldsymbol{X}_b, \boldsymbol{w}, K^{\text{stale}})$.
6: $\quad$**end for**
7: $\quad$Update the kernel matrix, K.
8: **until** Convergence
output Optimal $\boldsymbol{\theta}^*$ and $\boldsymbol{w}^*$

Figure 1: *Left:* Autonomous car route. *Middle:* Point-wise estimation of the lanes. Dashed – ground truth, blue – LSTM predictions, red – GP-LSTM predictions. *Right:* The algorithm used to train GPs with recurrent kernels.

where $\phi : \mathcal{X}^L \mapsto \mathcal{H}$ denotes the recurrent transformation. We train the model, GP-LSTM, by optimizing the probability of the training data, w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{w}$, having performed Bayesian marginalization over the induced distribution over functions given by the Gaussian process. The gradient of the objective w.r.t. the parameters of the recurrent transformation can be written as follows:

$$\frac{\partial \mathcal{L}}{\partial w_l} = \frac{1}{2}\sum_{i,j}\left(K^{-1}\boldsymbol{y}\boldsymbol{y}^\top K^{-1} - K^{-1}\right)_{ij}\left\{\left(\frac{\partial k(\boldsymbol{h}_i,\boldsymbol{h}_j)}{\partial \boldsymbol{h}_i}\right)^\top \frac{\partial \boldsymbol{h}_i}{\partial w_l} + \left(\frac{\partial k(\boldsymbol{h}_i,\boldsymbol{h}_j)}{\partial \boldsymbol{h}_j}\right)^\top \frac{\partial \boldsymbol{h}_j}{\partial w_l}\right\}.$$

Generally, neither the GP objective nor its gradients decompose over the data. However, the gradient does factorize when the the kernel matrix is *fixed*. This observation motivates us to propose a semi-stochastic optimization procedure that uses delayed kernel updates for efficiency (see Algorithm 1 in Figure 1). Convergence of the algorithm is given by Theorem 1 (further details and proofs can be found in the full version of the paper [1]).

Theorem 1. *Semi-stochastic gradient descent with τ-delayed kernel updates converges to a fixed point when the learning rate, λ_t, decays as $\Theta(1/\tau t^{\frac{1+\delta}{2}})$ for any $\delta \in (0, 1]$.*

3 Results

Our model, GP-LSTM, outperforms a number of classical (GPs, NARX, recurrent nets) and recent (recurrent GPs [5]) baselines on sequence regression tasks (a subset of results is given in Table 1). Additionally, the model is able to quantify predictive uncertainties (see Figure 1 for an example of lane prediction).

Our semi-stochastic asynchronous gradient descent speeds up convergence of the training and, when used in conjunction with KISS-GP [7, 8], scales as $\mathcal{O}(n)$ with the number of training data and scales linearly with the number of inducing points. Prediction requires constant time per testing point[1].

Table 1: Performance of the models in terms of RMSE on system identification & autonomous driving data.

Data	Task	NARX	RNN	LSTM	RGP	GP-NARX	GP-RNN	GP-LSTM
Drives	system ident.	0.423	0.408	0.382	0.249	0.403	0.332	**0.225**
Actuator		0.482	0.771	0.381	0.368	0.891	0.492	**0.347**
Car	speed	0.114	0.152	0.027	—	0.125	0.088	**0.019**
	gyro yaw	0.189	0.223	0.121	—	0.242	0.238	**0.076**
	lane seq.	0.128	0.331	0.078	—	0.101	0.472	**0.055**
	lead vehicle pos.	0.410	0.452	0.400	—	0.341	0.412	**0.312**

[1]These results are omitted due to space constraints, but are available in [1].

References

[1] M. Al-Shedivat, A. G. Wilson, Y. Saatchi, Z. Hu, and E. P. Xing. Learning scalable deep kernels with recurrent structure. *arXiv preprint arXiv:1610.08936*, 2016.

[2] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

[3] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] C. L. C. Mattos, Z. Dai, A. Damianou, J. Forth, G. A. Barreto, and N. D. Lawrence. Recurrent gaussian processes. *arXiv preprint arXiv:1511.06644*, 2015.

[6] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[7] A. G. Wilson and H. Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1775–1784, 2015.

[8] A. G. Wilson, C. Dann, and H. Nickisch. Thoughts on massively scalable gaussian processes. *arXiv preprint arXiv:1511.01870*, 2015.