Scalable GP-LSTMs with Semi-Stochastic Gradients

Maruan Al-Shedivat Andrew Gordon Wilson Yunus Saatchi Zhiting Hu Eric P. Xing

SAILING LAB · ML MACHINE LEARNING DEPARTMENT · Carnegie Mellon University

Summary

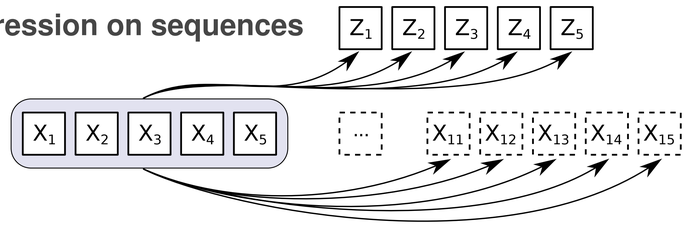
- **GP-LSTM:** a Gaussian process model which fully encapsulate the structural properties of LSTMs.
- **Semi-stochastic gradient descent:** new provably convergent optimization procedure for learning recurrent kernels.
- **State-of-the-art** results on sequence-to-reals regression.
- **Predictive uncertainty** for autonomous driving and other tasks.

Full paper: https://arxiv.org/abs/1610.08936

Code: https://github.com/alshedivat/kgp

1. Background & Motivation

Regression on sequences



- **Input sequences:** $\mathbf{X} = \{\bar{\mathbf{x}}_i\}_{i=1}^n$, $\bar{\mathbf{x}}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2, \cdots, \mathbf{x}_i^{l_i}]$, $\mathbf{x}_i^j \in \mathcal{X}$.
- **Real-valued targets:** $\mathbf{y} = \{y_i\}_{i=1}^n$, $y_i \in \mathbb{R}$.

Applications:

- Predictions of the expensive sensory data in **robotic systems**.
- Temporal modeling of medical biomarkers for predictive healthcare.
- Energy / power forecasting.
- Financial markets.

Recurrent neural networks

Advantages:
- A flexible framework for composing complex models.
- Accounts for long-range nonlinear correlations.
- Trainable by backprop.

Drawbacks:
- Easily overfits to noise.

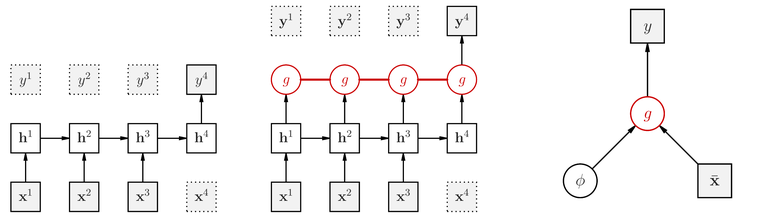
Gaussian processes

Advantages:
- Enjoys Bayesian nonparametric flexibility of the GPs.
- Robust to overfitting.

Drawbacks:
- Accounts only for pairwise linear correlations between the time points.

Our proposal: *Combine GPs with LSTMs to get flexible recurrent Bayesian nonparameteric models that are robust to noise and are able to provide predictive uncertainty estimates.*

2. Gaussian Processes with Recurrent Kernels



(a) LSTM (b) GP-LSTM (c) Graphical model view

LSTM-structured kernel

Let $\phi_{\mathbf{w}} : \mathcal{X}^L \mapsto \mathcal{H}$ be a recurrent transformation. Define the kernel as:

$$\tilde{k}_\theta(\bar{\mathbf{x}}, \bar{\mathbf{x}}') = k_\gamma(\phi_{\mathbf{w}}(\bar{\mathbf{x}}), \phi_{\mathbf{w}}(\bar{\mathbf{x}}')),$$

where $k_\gamma : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is the base kernel, $\bar{\mathbf{x}}, \bar{\mathbf{x}}' \in \mathcal{X}^L$ are L-length sequences, $\tilde{k}_\theta : \mathcal{X}^L \times \mathcal{X}^L \mapsto \mathbb{R}$, and $\theta = \{\gamma, \mathbf{w}\}$.

Negative log marginal likelihood objective

$$\mathcal{L}(K_\theta) = \overbrace{-\mathbf{y}^\top (K_\theta + \sigma^2 I)^{-1}\mathbf{y}}^{\text{model fit}} - \overbrace{\log\det(K_\theta + \sigma^2 I)}^{\text{complexity penalty}} + \text{const}$$

$$\nabla_\theta \mathcal{L} = \frac{1}{2}\operatorname{tr}\left[\left(K_\theta^{-1}\mathbf{y}\mathbf{y}^\top K_\theta^{-1} - K_\theta^{-1}\right)\nabla_\theta K_\theta\right]$$

Advantages:
- Enjoys Bayesian nonparametric flexibility of the GPs.
- Robust to overfitting.

Drawbacks:
- Does not factorize over the data.
- Involves kernel matrix inverses.

Algorithm: semi-stochastic asynchronous gradient descent

Input: Data: $(\mathbf{X}, \mathbf{y})$, kernel: $k_\gamma(\cdot,\cdot)$, recurrent transformation: $\phi_{\mathbf{w}}(\cdot)$

1. Initialize γ and $\mathbf{w}$; compute initial K
2. **repeat**
3. $\quad \gamma \leftarrow \gamma + \texttt{update}_\gamma(\mathbf{X}, \mathbf{w}, K)$.
4. $\quad$ **for all** mini-batches $\mathbf{X}_b$ in $\mathbf{X}$
5. $\quad\quad \mathbf{w} \leftarrow \mathbf{w} + \texttt{update}_{\mathbf{w}}(\mathbf{X}_b, \mathbf{w}, K^{\text{stale}})$
6. $\quad$ **endfor**
7. **Until:** convergence

Output: Optimal γ^* and $\mathbf{w}^*$

Theorem [Al-Shedivat et al., 2016]. *Semi-stochastic asynchronous 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. Experiments

Qualitative: Lane prediction for autonomous driving

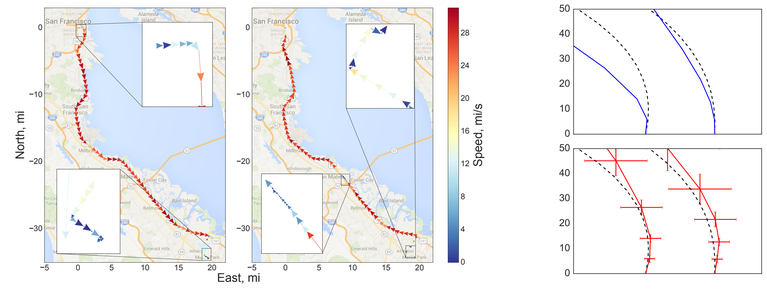


Figure 2: **Left:** Train and test routes of the autonomous car. **Right:** Point-wise estimation of the lanes. Dashed – ground truth, blue – LSTM, red – GP-LSTM.

Quantitative performance evaluation

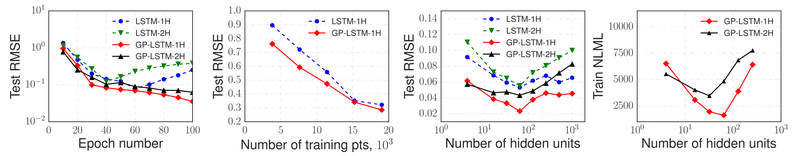


Figure 3: RMSE/NLML vs. the number points/parameters per layer.

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**
GEF	power pred.	0.529	0.622	0.465	—	0.780	0.242	**0.158**
	wind est.	0.837	0.818	0.781	—	0.835	0.792	**0.764**
Car	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**

Table 1: Performance of the models in terms of RMSE.

Convergence of the optimization

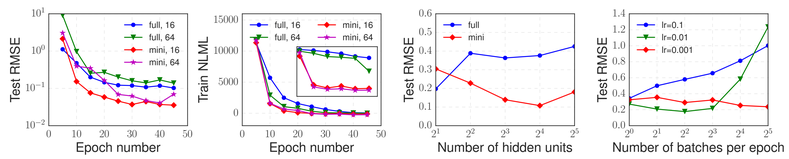


Figure 4: Convergence of the optimization with full-/mini-batches.

Scalability

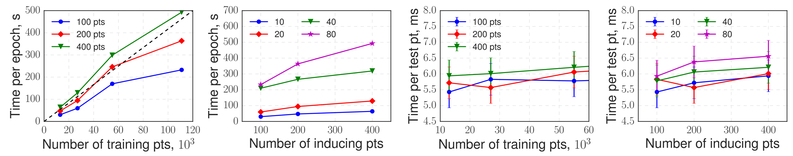


Figure 5: Scalability of learning and inference of GP-LSTM.

Correspondence: {alshedivat@cs.cmu.edu, andrew@cornell.edu}