Log-signatures and Neural Rough Differential Equations

James Foster

(joint with James Morrill, Cristopher Salvi, Patrick Kidger and Terry Lyons)

DataSıg and Oxford

DataSıg

A rough path between mathematics and data science

University of Oxford · The Alan Turing Institute · Imperial College London · UCL · UKRI Engineering and Physical Sciences Research Council

Outline

Signatures and Log-signatures

As we've already seen, the *signature* is a collection of features $\mathrm{Sig}_{a,b}(X)$ that we define from a continuous path $X : [0, T] \to \mathbb{R}^d$ (of finite length).

Definition (Depth-N Signature)

The depth-N signature transform of X over the interval $[a, b]$ is given by

$$\mathrm{Sig}_{a,b}^N(X) = \Big(\{S_{a,b}^i(X)\}_{i=1}^d, \{S_{a,b}^{i,j}(X)\}_{i,j=1}^d, \cdots, \{S_{a,b}^{i_1,\cdots,i_N}(X)\}_{i_1,\cdots,i_N=1}^d\Big) \tag{1}$$

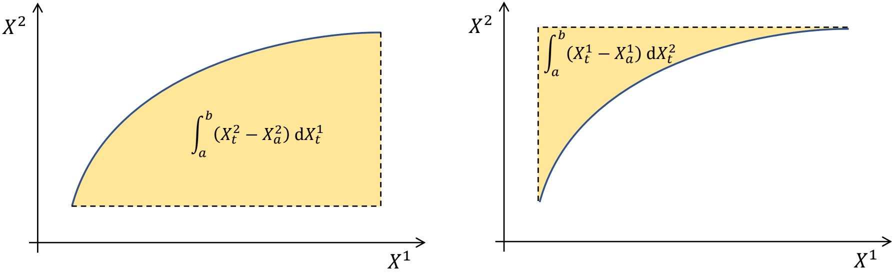
where

$$S_{a,b}^{i_1,\cdots,i_k}(X) = \int\cdots\int_{a<t_1<t_2<\cdots<t_k<b} \mathrm{d}X_{t_1}^{i_1}\,\mathrm{d}X_{t_2}^{i_2}\cdots\mathrm{d}X_{t_k}^{i_k}.$$

Is there redundancy in this feature set? If so, how can it be compressed?

Signatures and Log-signatures

Already with $d = 2$ and $N = 2$, we can see there is some redundancy



as a simple application of integration by parts yields

$$\iint_{a<t_1<t_2<b} \mathrm{d}X_{t_1}^1\,\mathrm{d}X_{t_2}^2 + \iint_{a<t_1<t_2<b} \mathrm{d}X_{t_1}^2\,\mathrm{d}X_{t_2}^1 = \left(X_b^1 - X_a^1\right)\left(X_b^2 - X_a^2\right).$$

Signatures and Log-signatures

Definition (Depth-2 Log-signature)

The depth-2 log-signature transform of X is given by

$$\mathrm{LogSig}^2_{a,b}(X) = \Bigg(\big\{X^i_b - X^i_a\big\}_{1\leq i\leq d}\,, \qquad (2)$$
$$\bigg\{\frac{1}{2}\bigg(\iint_{a<t_1<t_2<b} \mathrm{d}X^i_{t_1}\,\mathrm{d}X^j_{t_2} - \iint_{a<t_1<t_2<b} \mathrm{d}X^j_{t_1}\,\mathrm{d}X^i_{t_2}\bigg)\bigg\}_{1\leq i<j\leq d}\Bigg)$$

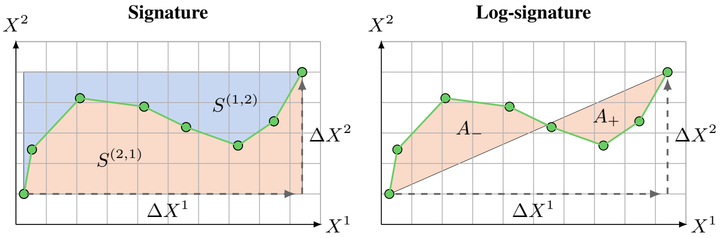


Figure: Illustration of depth-2 signature and log-signature (taken from [1]).

Signatures and Log-signatures

Removing redundancy at depth-3 gives log-signature terms of the form:

$$\frac{1}{6}\iiint_{a<t_1<t_2<t_3<b}\Big(dX_{t_1}^i\big(dX_{t_2}^j dX_{t_3}^k - dX_{t_2}^k dX_{t_3}^j\big) - \big(dX_{t_1}^j dX_{t_2}^k - dX_{t_1}^k dX_{t_2}^j\big)dX_{t_3}^i\Big).$$

Log-signatures have an interesting computational and algebraic story!

- P. Kidger and T. Lyons. *Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU*. In International Conference on Learning Representations, 2021. `https://github.com/patrick-kidger/signatory`

- J. Diehl, T. Lyons, R. Preiß and J. Reizenstein. *Areas of areas generate the shuffle algebra*. arXiv:2002.02338, 2020.

- J. Reizenstein. *Calculation of Iterated-Integral Signatures and Log Signatures*. arXiv:1712.02757, 2017.

Log-signatures as a method for lossy compression

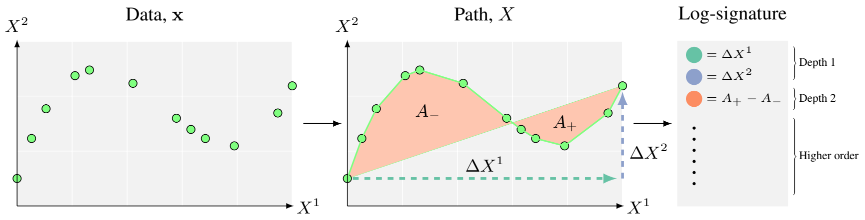


Figure: Illustration of depth-2 log-signature (taken from [6]).

We can reduce the length of a time series by computing log-signatures (locally) over different intervals. This gives a length/channel trade-off.

Using log-signatures as a preprocessing step can improve performance for ML models such as RNNs [1, 5] and neural differential equations [6].

We focus on the latter, which we call *Neural Rough Differential Equations*

Outline

Neural Controlled Differential Equations (NCDEs)

We observe $\boldsymbol{x} = ((t_0, x_0), (t_1, x_1), \cdots, (t_n, x_n))$, with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^d$.

Let $X : [0, n] \to \mathbb{R}^{d+1}$ be a continuous path that interpolates this data, so $X(i) = (t_i, x_i)$. (e.g. cubic splines [7] and piecewise linear/rectilinear [8])

The NCDE model involves learnt functions ζ_θ, f_θ and a linear map ℓ_θ with

$$z(0) = \zeta_\theta(t_0, x_0), \quad z(t) = z(0) + \int_0^t f_\theta(z(s))\, \mathrm{d}X(s), \tag{3}$$

and the output is either $\ell_\theta(z(T))$ or $\{\ell_\theta(z(t))\}$.

The CDE model (3) is discretized, the output is fed into a loss function (L^2, cross entropy, etc) and trained using stochastic gradient descent.

Here ζ_θ and f_θ are neural nets, z is hidden state: Continuous Time RNN

Neural Controlled Differential Equations (NCDEs)

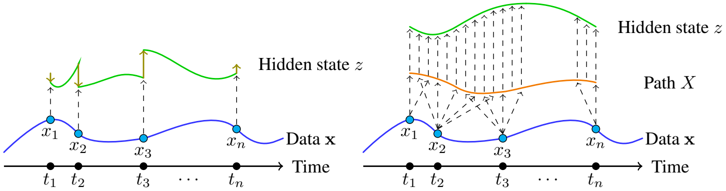


Figure: Illustration of the RNN and NCDE models (taken from [7]).

What does rough path theory tell us about log-signatures / CDEs?

$$
\begin{aligned}
\text{CDE solution} &\approx \text{Truncated Taylor expansion of CDE solution}\\
&= \text{Linear functional applied to } \mathrm{Sig}^N(X)\\
&= \text{Linear functional applied to exponential of } \mathrm{LogSig}^N(X)\\
&\approx \text{Exponential of linear functional applied to } \mathrm{LogSig}^N(X)\\
&= \text{Solving an ODE obtained from } f_\theta \text{ and } \mathrm{LogSig}^N(X)
\end{aligned}
$$

Conclusion

Log-signatures and Neural CDEs are a natural fit!

Outline

Neural Rough Differential Equations (NRDEs)

Definition (Neural RDE)

We pick a (uniform) partition $\{r_i\}$ of $[t_0, t_n]$ and depth hyperparameter N. The Neural RDE model follows the Neural CDE formulation, but with

$$z(t) = z(0) + \int_0^t g_{\theta,X}(z(s), s)\,\mathrm{d}s, \qquad (4)$$

where $$g_{\theta,X}(z,s) = \widehat{f}_\theta(z)\,\frac{\mathrm{LogSig}^N_{r_i,r_{i+1}}(X)}{r_{i+1} - r_i}, \quad \text{for } s \in [r_i, r_{i+1}).$$

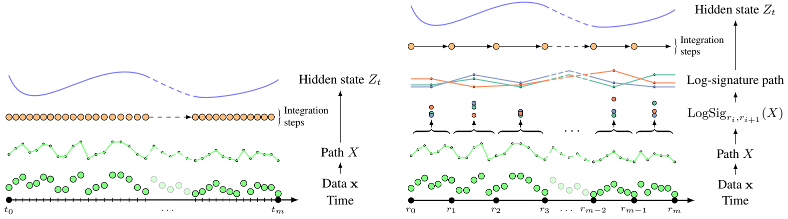


Figure: Illustration of the NCDE and NRDE models (taken from [6]).

Classification on time series with a length of $17K$

Model	Step	Accuracy (%)	Time (Hrs)	Mem (Mb)
ODE-RNN (folded)	1	–	–	–
	4	35.0 ± 1.5	0.8	3629.3
	32	32.5 ± 1.5	0.1	532.2
	128	47.9 ± 5.3	0.0	200.8
NCDE	1	62.4 ± 12.1	22.0	176.5
	4	66.7 ± 11.8	5.5	46.6
	32	64.1 ± 14.3	0.5	8.0
	128	48.7 ± 2.6	0.1	3.9
NRDE (depth 2)	4	$\mathbf{83.8} \pm \mathbf{3.0}$	2.4	180.0
	32	67.5 ± 12.1	0.7	28.1
	128	$\mathbf{76.1} \pm \mathbf{5.9}$	0.2	7.8
NRDE (depth 3)	4	76.9 ± 9.2	2.8	856.8
	32	$\mathbf{75.2} \pm \mathbf{3.0}$	0.6	134.7
	128	68.4 ± 8.2	0.1	53.3

Table: EigenWorms dataset: mean $\pm$ standard deviation of test set accuracy measured over three runs. Bold denotes the best score for a given step size.

Outline

Conclusion

- The log-signature is a compressed version of the signature
  - nice geometric interpretation
  - reduces length of a time series, but increases number of channels
  - can improve performance for RNNs [1, 5] and Neural CDEs [6]
- CDEs are naturally related to log-signatures and ODEs (“log-ODE”)
- Neural RDEs combine the strengths of NCDEs and log-signatures
  - memory efficient, continuous time and suitable for long time series
  - when step $= 1$ and depth $= 1$, reduces to usual Neural CDE model
- Code available
  - `https://github.com/patrick-kidger/signatory`
  - `https://github.com/patrick-kidger/torchcde`
  - `https://github.com/jambo6/neuralRDEs`

Thank you
for your attention!

and for more details, see

J. Morrill, C. Salvi, P. Kidger, J. Foster and T. Lyons. *Neural Rough Differential Equations for Long Time Series*. Proceedings of the 38th International Conference on Machine Learning (ICML), 2021.

Outline

References I

- J. Morrill, A. Fermanian, P. Kidger and T. Lyons. *A Generalised Signature Method for Multivariate Time Series Feature Extraction*. arXiv:2006.00873, 2020.

- P. Kidger and T. Lyons. *Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU*. In International Conference on Learning Representations, 2021. `https://github.com/patrick-kidger/signatory`

- J. Diehl, T. Lyons, R. Preiß and J. Reizenstein. *Areas of areas generate the shuffle algebra*. arXiv:2002.02338, 2020.

- J. Reizenstein. *Calculation of Iterated-Integral Signatures and Log Signatures*. arXiv:1712.02757, 2017.

References II

S. Liao, T. Lyons, W. Yang and H. Ni. *Learning stochastic differential equations using RNN with log signature features*. arXiv:1908.08286, 2019.

J. Morrill, C. Salvi, P. Kidger, J. Foster and T. Lyons. *Neural Rough Differential Equations for Long Time Series*. Proceedings of the 38th International Conference on Machine Learning (ICML), 2021.

P. Kidger, J. Morrill, J. Foster and T. Lyons. *Neural Controlled Differential Equations for Irregular Time Series*. In Advances in Neural Information Processing Systems, 2020.

J. Morrill, P. Kidger, L. Yang and T. Lyons. *Neural Controlled Differential Equations for Online Prediction Tasks*. arXiv:2106.11028, 2021.