Neural Controlled Differential Equations for Irregular Time Series

PATRICK KIDGER, JAMES MORRILL, JAMES FOSTER, TERRY LYONS

Mathematical Institute
University of Oxford

Neural Information Processing Systems 2020

UNIVERSITY OF OXFORD
Mathematical Institute

Oxford Mathematics

Summary

Mathematical Institute

- Neural ODEs are famously the continuous-time limit of ResNets:

$$z_{n+1} = z_n + f_\theta(z_n) \quad \longrightarrow \quad \frac{\mathrm{d}z}{\mathrm{d}t} = f_\theta(z(t))$$

- We introduce Neural Controlled Differential Equations as the continuous-time limit of RNNS.

- Operate on messy time series; memory efficient; state-of-the-art performance; easy to implement.

Neural ODEs

(Chen at al. 2018)

UNIVERSITY OF OXFORD
Mathematical Institute

Use an ODE

$$z(t) = z(0) + \int_0^t f_\theta(z(s))\,\mathrm{d}s \qquad \text{for } t \in [0, T]$$

as a learnt component of a differentiable computation graph.

- The integral over $s \in [0, T]$ is introduced and then integrated over, and is just an internal detail of the model.
- But given some ordered data $(x_0, \ldots, x_n)$, we would like to align $s \in [0, T]$ with the ordering of the data.
- Problem: The solution to an ODE is determined by its initial condition.

Controlled Differential Equations

(Lyons 1998, Lyons et al. 2004)

UNIVERSITY OF OXFORD
Mathematical Institute

Have the local dynamics depend upon some time-varying $X\colon [0, T] \to \mathbb{R}^v$. This gives

$$z(t) = z(0) + \int_0^t f(z(s))\,\mathrm{d}X(s)\;\underbrace{\int_0^t f(z(s))\,\mathrm{d}X(s)}_{=\int_0^t f(z(s))\frac{\mathrm{d}X}{\mathrm{d}s}(s)\,\mathrm{d}s}$$

which is a Riemann–Stieltjes integral, and "$f(z(s))\,\mathrm{d}X(s)$" is a matrix-vector product.

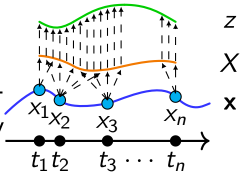
Bringing them together: Neural CDEs

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
Let $X \colon [t_0, t_n] \to \mathbb{R}^{v+1}$ interpolate this data, so $X(t_i) = (x_i, t_i)$.

Learn functions ζ_θ, f_θ and a linear map ℓ_θ such that

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

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



ζ_θ and f_θ are arbitrary neural networks (feed-forward, ...), and z is hidden state. Directly analogous to an RNN "$z_{n+1} = f_\theta(z_n, x_n)$".

Advantages

Mathematical Institute

- Using a continuous-time theory pushes the problem of messy data into the interpolation, which is better suited for handling it.
- The equation $z(t) = z(0) + \int_0^t f_\theta(z(s)) \frac{\mathrm{d}X}{\mathrm{d}s} \, \mathrm{d}s$ may be interpreted as a (neural) ODE, so:
  - We can solve it using existing software.
  - The adjoint method may be applied, even over observations.
- Neural CDEs demonstrate state-of-the-art performance.
- Drawing on the existing theory of CDEs gives strong theoretical guarantees. For example, neural CDEs are universal approximators.

Results - Character Trajectories

UNIVERSITY OF OXFORD
Mathematical Institute

Model	Test Accuracy (mean ± std, computed across five runs)			Memory usage
	30% dropped	50% dropped	70% dropped	
GRU-ODE	92.6% ± 1.6%	86.7% ± 3.9%	89.9% ± 3.7%	1.5
GRU-Δt	93.6% ± 2.0%	91.3% ± 2.1%	90.4% ± 0.8%	15.8
GRU-D	94.2% ± 2.1%	90.2% ± 4.8%	91.9% ± 1.7%	17.0
ODE-RNN	95.4% ± 0.6%	96.0% ± 0.3%	95.3% ± 0.6%	14.8
Neural CDE (ours)	**98.7% ± 0.8%**	**98.8% ± 0.2%**	**98.6% ± 0.4%**	**1.3**

Gotchas

UNIVERSITY OF OXFORD
Mathematical Institute

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

- ζ_θ is a function of the initial point for translation sensitivity.
  - Can be a function of any static features as well.
- Include time as a channel in X. (Recall $X(t_i) = (x_i, t_i)$.)
- f_θ should have a final tanh nonlinearity.

Discussion

Comparison to control theory problems (speaking *very* broadly):

- Control theory: System f is fixed; try to find optimal X producing a desired response z.
- (Neural) CDEs: Input X is fixed; try to find optimal f_θ producing a desired response z.

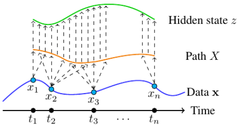
Exciting future applications: can train Neural SDEs as a GAN, using a Neural CDE as the discriminator! (Kidger et al. 2020)

Summary

UNIVERSITY OF OXFORD
Mathematical Institute

- New tool for time series, uniting the well-understood mathematics of CDEs with the well-understood machine learning of Neural ODEs.
- Acts on messy (irregularly sampled, partially observed, variable length) time series.
- Memory efficient to train.
- Strong theoretical connections.
- Demonstrates state-of-the-art performance.
- Straightforward to implement with existing tools.

Links



https://arxiv.org/abs/2005.08926
https://github.com/patrick-kidger/NeuralCDE
https://youtu.be/sbcIKugElZ4

Library: https://github.com/patrick-kidger/torchcde

These slides: https://kidger.site/links/NeurIPS-2020-Neural-CDEs

References

Mathematical
Institute

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural Ordinary Differential Equations", in *Advances in Neural Information Processing Systems 31*, pp. 6571–6583, Curran Associates, Inc., 2018

T. Lyons, *Differential equations driven by rough signals*, Revista Matemática Iberoamericana, 14(2) pp. 215–310, 1998

T. Lyons, M. Caruana, and T. Levy, *Differential equations driven by rough paths*. Springer, 2004. École d'Été de Probabilités de Saint-Flour XXXIV - 2004

P. Kidger, J. Foster, X. Li, H. Oberhauser, T. Lyons, "Neural SDEs Made Easy: SDEs are Infinite-Dimensional GANs", *OpenReview*, 2020.