Neural Controlled Differential Equations

James Foster

DataSıg and the University of Bath

DataSıg and Bath

DataSıg

A rough path between mathematics and data science

Outline

Signatures and Controlled Differential Equations

As we have already seen, the *signature* is a collection of features that we can 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 $[0, t]$ is given by

$$\mathrm{Sig}^N_{0,t}(X) = \Big(\big\{S^{i}_{0,t}(X)\big\}_{i=1}^{d}, \big\{S^{i,j}_{0,t}(X)\big\}_{i,j=1}^{d}, \cdots, \big\{S^{i_1,\cdots,i_N}_{0,t}(X)\big\}_{i_1,\cdots,i_N=1}^{d}\Big),$$

where

$$S^{i_1,\cdots,i_k}_{0,t}(X) = \int\cdots\int_{0<s_1<s_2<\cdots<s_k<t} \mathrm{d}X^{i_1}_{s_1}\,\mathrm{d}X^{i_2}_{s_2}\cdots\mathrm{d}X^{i_k}_{s_k}\,.$$

We can extend the above to define the full path signature (i.e. $N = \infty$).

Signatures and Controlled Differential Equations

Thus, it follows that different entries in the signature can be related as

$$S_{0,t}^{i_1,\cdots,i_k}(X) = \int_0^t S_{0,s}^{i_1,\cdots,i_{k-1}}(X)\,\mathrm{d}X_s^{i_k}.$$

Thus, it follows that different entries in the signature can be related as

$$S_{0,t}^{i_1,\cdots,i_k}(X) = \int_0^t S_{0,s}^{i_1,\cdots,i_{k-1}}(X)\,\mathrm{d}X_s^{i_k}.$$

Definition (Controlled differential equation)

We say $Y : [0,T] \to \mathbb{R}^n$ solves a Controlled Differential Equation (CDE) if

$$Y_t = Y_0 + \int_0^t f(Y_s)\,\mathrm{d}X_s,$$

where $f : \mathbb{R}^n \to \mathbb{R}^{n\times d}$ and $X : [0,T] \to \mathbb{R}^d$.

Signatures and Controlled Differential Equations

Thus, it follows that different entries in the signature can be related as

$$S_{0,t}^{i_1,\cdots,i_k}(X) = \int_0^t S_{0,s}^{i_1,\cdots,i_{k-1}}(X)\,\mathrm{d}X_s^{i_k}.$$

Definition (Controlled differential equation)

We say $Y : [0,T] \to \mathbb{R}^n$ solves a Controlled Differential Equation (CDE) if

$$Y_t = Y_0 + \int_0^t f(Y_s)\,\mathrm{d}X_s\,, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^{n\times d}$ and $X : [0,T] \to \mathbb{R}^d$. We often write (1) in the form:

$$\mathrm{d}Y_t = f(Y_t)\,\mathrm{d}X_t. \tag{2}$$

Informal Theorem

"Path Signature + Linear Regression = Linear CDE"

Outline

Neural Controlled Differential Equations (NCDEs)

Whilst CDEs encompass path signatures, they also extend ODEs since

$$Y_t = Y_0 + \int_0^t f(Y_s)\, dX_s = Y_0 + \int_0^t f(Y_s)\, \frac{\mathrm{d}X_s}{\mathrm{d}s}\, \mathrm{d}s.$$

That is, when X is continuously differentiable, a CDE can be written as

$$\frac{\mathrm{d}Y_t}{\mathrm{d}t} = g(t, Y_t)\,, \tag{3}$$

where $g(t,y) = f(y)\,\frac{\mathrm{d}X}{\mathrm{d}t}$.

Neural Controlled Differential Equations (NCDEs)

Whilst CDEs encompass path signatures, they also extend ODEs since

$$Y_t = Y_0 + \int_0^t f(Y_s)\,dX_s = Y_0 + \int_0^t f(Y_s)\,\frac{\mathrm{d}X_s}{\mathrm{d}s}\,\mathrm{d}s.$$

That is, when X is continuously differentiable, a CDE can be written as

$$\frac{\mathrm{d}Y_t}{\mathrm{d}t} = g(t, Y_t)\,, \tag{4}$$

where $g(t, y) = f(y)\,\frac{\mathrm{d}X}{\mathrm{d}t}$. Hence, we can learn f using the methodology in

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt and David Duvenaud. *Neural Ordinary Differential Equations*. NeurIPS 2018.

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 $x_i \in \mathbb{R}^d$.

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 [2] and piecewise linear/rectilinear [3])

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 [2] and piecewise linear/rectilinear [3])

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

$$z(0) = \zeta_\theta(t_0, x_0), \quad z(t) = z(0) + \int_0^t f_\theta(z(s))\, \mathrm{d}X(s), \tag{5}$$

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

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 [2] and piecewise linear/rectilinear [3])

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

$$z(0) = \zeta_\theta(t_0, x_0), \quad z(t) = z(0) + \int_0^t f_\theta(z(s))\, \mathrm{d}X(s), \tag{5}$$

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

The CDE model (5) is discretized, the output is fed into a loss function (L^2, cross entropy, etc) and trained using stochastic gradient descent.

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 [2] and piecewise linear/rectilinear [3])

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

$$z(0) = \zeta_\theta(t_0, x_0), \quad z(t) = z(0) + \int_0^t f_\theta(z(s))\, \mathrm{d}X(s), \tag{5}$$

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

The CDE model (5) 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)

CDEs are reparameterization invariant and well suited to tasks involving (partially-observed and/or irregularly sampled) multivariate time series.

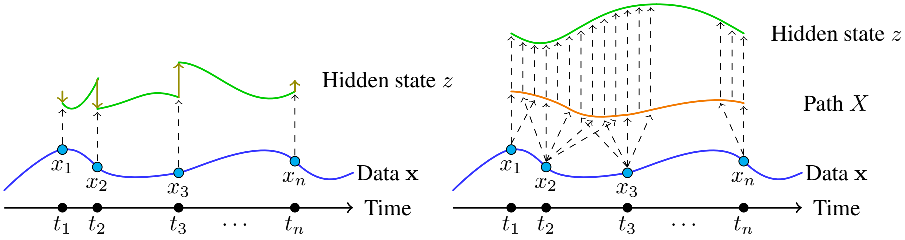
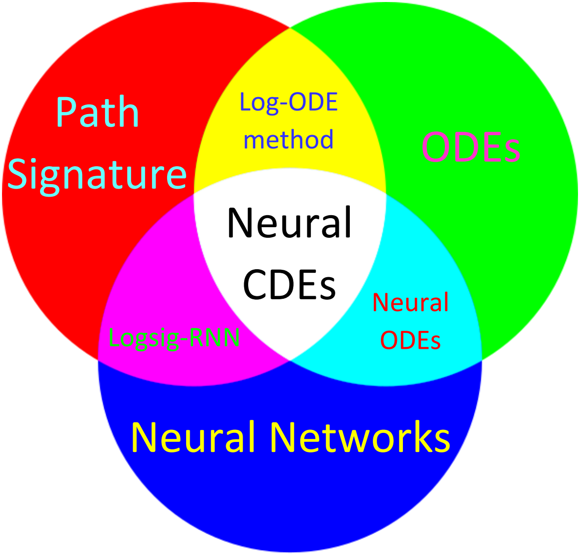


Figure: Illustration of the RNN and NCDE models (taken from [2]).

Neural Controlled Differential Equations (NCDEs)



NCDEs work! (and allow for memory-efficient training)

Model	Test Accuracy	Memory usage (Mb)
GRU-ODE	$47.9\% \pm 2.9\%$	**0.164**
GRU-Δt	$43.3\% \pm 33.9\%$	1.54
GRU-D	$32.4\% \pm 34.8\%$	1.64
ODE-RNN	$65.9\% \pm 35.6\%$	1.40
Neural CDE	$\mathbf{89.8\%} \pm \mathbf{2.5\%}$	0.167

Table: Speech Commands classification (regularly spaced, fully observed)

Model	Test AUC	Memory usage (Mb)
GRU-ODE	0.852 ± 0.010	454
GRU-Δt	0.878 ± 0.006	837
GRU-D	0.871 ± 0.022	889
ODE-RNN	0.874 ± 0.016	696
Neural CDE	$\mathbf{0.880} \pm \mathbf{0.006}$	**244**

Table: PhysioNet Sepsis prediction (irregularly sampled, partially observed)

Outline

Conclusion and related work

- Neural CDEs are a model for continuous paths (and time series), at the intersection of Path Signatures, ODEs and Neural Networks (enjoying the benefits of all three!)

- "Neural Rough Differential Equations for Long Time Series" [4]

- Subsequent applications:
  - Reinforcement learning for healthcare [5]
  - Continuous-time multiscale control in robotics [6]
  - Modelling of counterfactual outcomes for healthcare [7]
  - Signature-based autoencoder for feature extraction in NRDEs [8]
  - CDE discriminator in GANs: Neural SDEs [9] and ECG Synthesis [10]

- Software available:
  - `https://github.com/patrick-kidger/torchcde`
  - `https://github.com/patrick-kidger/diffrax`

Thank you
for your attention!

Outline

References I

R. T. Q. Chen, Y. Rubanova, J. Bettencourt and D. Duvenaud. *Neural Ordinary Differential Equations*. In Advances in Neural Information Processing Systems, 2018.

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. *On the Choice of Interpolation Scheme for Neural CDEs*. Transactions on Machine Learning Research, 2022.

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.

References II

T. W. Killian, H. Zhang, J. Subramanian, M. Fatemi and M. Ghassemi. *An Empirical Study of Representation Learning for Reinforcement Learning in Healthcare*. Machine Learning for Health NeurIPS Workshop, 2020.

S. Singh, F. M. Ramirez, J. Varley, A. Zeng and V. Sindhwani. *Multiscale Sensor Fusion and Continuous Control with Neural CDEs*. Robotics: Science and Systems Workshop on "Overlooked Aspects of Imitation Learning: Systems, Data, Tasks, and Beyond", 2022. (preprint available at arxiv:2203.08715).

N. Seedat, F. Imrie, A. Bellot, Z. Qian and M. van der Schaar. *Continuous-Time Modeling of Counterfactual Outcomes Using Neural Controlled Differential Equations*. Proceedings of the 39th International Conference on Machine Learning (ICML), 2022.

References III

J. Lee, J. Jeon, S. Jhin, J. Hyeong, J. Kim, M. Jo, K. Seungji and N. Park. *LORD: Lower-Dimensional Embedding of Log-Signature in Neural Rough Differential Equations*. Proceedings of the 10th International Conference on Learning Representations, 2022.

P. Kidger, J. Foster, X. Li, H. Oberhauser and T. Lyons. *Neural SDEs as Infinite-Dimensional GANs*. Proceedings of the 38th International Conference on Machine Learning (ICML), 2021.

M. Habiba, E. Borphy, B. A. Pearlmutter and T. Ward . *ECG Synthesis with Neural ODE and GAN Model*, International Conference on Electrical, Computer and Energy Technologies (ICECET), 2021.