Neural Controlled Differential Equations for Irregular Time Series

Patrick Kidger

Mathematical Institute
University of Oxford

Market Generators 2020

University of Oxford
Mathematical Institute

Oxford Mathematics

Joint work with...

UNIVERSITY OF OXFORD
Mathematical Institute

James Morrill, James Foster, Terry Lyons

Links

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

Summary

Neural Controlled Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

Summary

Neural Controlled Differential Equations

Mathematical Institute

— New tool for time series

Summary

Neural Controlled Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

— New tool for time series

— Acts directly on irregularly sampled partially observed multivariate time series.

— New tool for time series

— Acts directly on irregularly sampled partially observed multivariate time series.

— Can be trained with memory-efficient adjoint backpropagation, even across observations

— New tool for time series

— Acts directly on irregularly sampled partially observed multivariate time series.

— Can be trained with memory-efficient adjoint backpropagation, even across observations

— Straightforward to implement with existing tools.

University of Oxford
Mathematical Institute

— New tool for time series

— Acts directly on irregularly sampled partially observed multivariate time series.

— Can be trained with memory-efficient adjoint backpropagation, even across observations

— Straightforward to implement with existing tools.

— Demonstrates state-of-the-art performance.

Recap

Controlled Differential Equations

Recap

~~Controlled~~ Ordinary Differential Equations

Mathematical Institute

(vector field) $\quad f : \mathbb{R}^w \to \mathbb{R}^w$

Recap

~~Controlled~~ Ordinary Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

$$\text{(vector field)} \qquad f\colon \mathbb{R}^w \to \mathbb{R}^w$$

$$\text{(solution)} \qquad z\colon [0, T] \to \mathbb{R}^w$$

Recap

~~Controlled~~ Ordinary Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

(vector field) $f\colon \mathbb{R}^w \to \mathbb{R}^w$

(solution) $z\colon [0, T] \to \mathbb{R}^w$

(ODE)
$$\frac{\mathrm{d}z}{\mathrm{d}t}(t) = f(z(t))$$
$$z(0) = z_0$$

Recap

~~Controlled~~ Ordinary ~~Ordinary~~ Controlled Differential Equations

$$
\begin{aligned}
&\text{(vector field)} && f\colon \mathbb{R}^w \to \mathbb{R}^w \\
&\text{(solution)} && z\colon [0, T] \to \mathbb{R}^w \\
&\text{(ODE)} && \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f(z(t)) \\
& && z(0) = z_0
\end{aligned}
$$

Recap

~~Controlled~~ → ~~Ordinary~~ → *Controlled* Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

(control)	$X \colon [0, T] \to \mathbb{R}^v$
(vector field)	$f \colon \mathbb{R}^w \to \mathbb{R}^w$
(solution)	$z \colon [0, T] \to \mathbb{R}^w$
(ODE)	$\dfrac{\mathrm{d}z}{\mathrm{d}t}(t) = f(z(t))$
	$z(0) = z_0$

Recap

~~Controlled~~ Ordinary Controlled Differential Equations

(control)	$X\colon [0, T] \to \mathbb{R}^v$
(vector field)	$f\colon \mathbb{R}^w \to \mathbb{R}^{w \times v}$
(solution)	$z\colon [0, T] \to \mathbb{R}^w$
(ODE)	$\frac{\mathrm{d}z}{\mathrm{d}t}(t) = f(z(t))$
	$z(0) = z_0$

Recap

~~Controlled~~ ~~Ordinary~~ Controlled Differential Equations

$$
\begin{aligned}
&\text{(control)} && X \colon [0, T] \to \mathbb{R}^v \\
&\text{(vector field)} && f \colon \mathbb{R}^w \to \mathbb{R}^{w \times v} \\
&\text{(solution)} && z \colon [0, T] \to \mathbb{R}^w \\
&\text{(ODE)} && \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f(z(t)) \frac{\mathrm{d}X}{\mathrm{d}t}(t) \\
& && z(0) = z_0
\end{aligned}
$$

Recap

~~Controlled~~ → Ordinary → *Controlled*

Controlled Differential Equations

(control) $X \colon [0, T] \to \mathbb{R}^v$

(vector field) $f \colon \mathbb{R}^w \to \mathbb{R}^{w \times v}$

(response) $z \colon [0, T] \to \mathbb{R}^w$

(CDE) $$\frac{\mathrm{d}z}{\mathrm{d}t}(t) = f(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t)$$

$$z(0) = z_0$$

Recap

Neural Ordinary Differential Equations

Goal: learn a map $x \mapsto y$

UNIVERSITY OF OXFORD
Mathematical Institute

Goal: learn a map $x \mapsto y$ by learning a function f_θ and linear maps ℓ_θ^1, ℓ_θ^2 such that

Goal: learn a map $x \mapsto y$ by learning a function f_θ and linear maps ℓ^1_θ, ℓ^2_θ such that

$$z(0) = \ell^1_\theta(x)$$

Recap

Neural Ordinary Differential Equations

Goal: learn a map $x \mapsto y$ by learning a function f_θ and linear maps ℓ^1_θ, ℓ^2_θ such that

$$z(0) = \ell^1_\theta(x) \quad \text{and} \quad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))$$

Recap

Neural Ordinary Differential Equations

Goal: learn a map $x \mapsto y$ by learning a function f_θ and linear maps ℓ^1_θ, ℓ^2_θ such that

$$z(0) = \ell^1_\theta(x) \quad \text{and} \quad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t)) \quad \text{and} \quad y \approx \ell^2_\theta(z(T)).$$

Recap

Neural Ordinary Differential Equations

University of Oxford
Mathematical Institute

Goal: learn a map $x \mapsto y$ by learning a function f_θ and linear maps ℓ_θ^1, ℓ_θ^2 such that

$$z(0) = \ell_\theta^1(x) \quad \text{and} \quad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t)) \quad \text{and} \quad y \approx \ell_\theta^2(z(T)).$$

z is "hidden state".

Goal: learn a map $x \mapsto y$ by learning a function f_θ and linear maps ℓ^1_θ, ℓ^2_θ such that

$$z(0) = \ell^1_\theta(x) \quad \text{and} \quad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t)) \quad \text{and} \quad y \approx \ell^2_\theta(z(T)).$$

z is "hidden state".

Have an efficient training algorithm (adjoint backpropagation) that uses $\mathcal{O}(1)$ memory in the time horizon T.

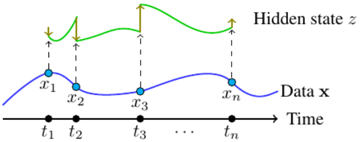
Recap

Neural Ordinary Differential Equations for time series

Recap

Neural Ordinary Differential Equations for time series

UNIVERSITY OF OXFORD
Mathematical Institute



Splines

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.

Neural Controlled Differential Equations

Splines

Observe $\mathbf{x} = ((t_0, x_0), \dots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0,\ t_n = T$)

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0,\ t_n = T$)
Let $X : [t_0, t_n] = [0, T] \to \mathbb{R}^v$ be the natural cubic spline interpolation of this data, so $X(t_i) = (x_i, t_i)$.

Neural Controlled Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0,\ t_n = T$)
Let $X \colon [t_0, t_n] = [0, T] \to \mathbb{R}^v$ be the natural cubic spline interpolation of this data, so $X(t_i) = (x_i, t_i)$.

Neural Controlled Differential Equations

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0$, $t_n = T$)
Let $X : [t_0, t_n] = [0, T] \to \mathbb{R}^v$ be the natural cubic spline interpolation of this data, so $X(t_i) = (x_i, t_i)$.

Goal: learn a map $\mathbf{x} \mapsto y$

Neural Controlled Differential Equations

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0$, $t_n = T$)
Let $X \colon [t_0, t_n] = [0, T] \to \mathbb{R}^v$ be the natural cubic spline interpolation of this data, so $X(t_i) = (x_i, t_i)$.

Goal: learn a map $\mathbf{x} \mapsto y$ by learning functions ζ_θ, f_θ and a linear map ℓ_θ such that

Neural Controlled Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0$, $t_n = T$)
Let $X\colon [t_0, t_n] = [0, T] \to \mathbb{R}^v$ be the natural cubic spline interpolation of this data, so $X(t_i) = (x_i, t_i)$.

Goal: learn a map $\mathbf{x} \mapsto y$ by learning functions ζ_θ, f_θ and a linear map ℓ_θ such that

$$z(0) = \zeta_\theta(t_0, x_0)$$

Neural Controlled Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0$, $t_n = T$)
Let $X \colon [t_0, t_n] = [0, T] \to \mathbb{R}^v$ be the natural cubic spline interpolation of this data, so $X(t_i) = (x_i, t_i)$.

Goal: learn a map $\mathbf{x} \mapsto y$ by learning functions ζ_θ, f_θ and a linear map ℓ_θ such that

$$z(0) = \zeta_\theta(t_0, x_0) \quad \text{and} \quad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t),$$

Neural Controlled Differential Equations

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0,\ t_n = T$)
Let $X \colon [t_0, t_n] = [0, T] \to \mathbb{R}^v$ be the natural cubic spline interpolation of this data, so $X(t_i) = (x_i, t_i)$.

Goal: learn a map $\mathbf{x} \mapsto y$ by learning functions ζ_θ, f_θ and a linear map ℓ_θ such that

$$z(0) = \zeta_\theta(t_0, x_0) \quad \text{and} \quad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t),$$

and $y(t) \approx \ell_\theta(z(t))$

Neural Controlled Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0$, $t_n = T$)
Let $X \colon [t_0, t_n] = [0, T] \to \mathbb{R}^v$ be the natural cubic spline interpolation of this data, so $X(t_i) = (x_i, t_i)$.

Goal: learn a map $\mathbf{x} \mapsto y$ by learning functions ζ_θ, f_θ and a linear map ℓ_θ such that

$$z(0) = \zeta_\theta(t_0, x_0) \quad \text{and} \quad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t),$$

and $y(t) \approx \ell_\theta(z(t))$ or $y \approx \ell_\theta(z(T))$.

Neural Controlled Differential Equations

Observe $\mathbf{x} = ((t_0, x_0), \ldots, (t_n, x_n))$ with $t_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^v$.
(WLOG $t_0 = 0$, $t_n = T$)
Let $X : [t_0, t_n] = [0, T] \to \mathbb{R}^v$ be the natural cubic spline interpolation of this data, so $X(t_i) = (x_i, t_i)$.

Goal: learn a map $\mathbf{x} \mapsto y$ by learning functions ζ_θ, f_θ and a linear map ℓ_θ such that

$$z(0) = \zeta_\theta(t_0, x_0) \quad \text{and} \quad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t),$$

and $y(t) \approx \ell_\theta(z(t))$ or $y \approx \ell_\theta(z(T))$.
(once again z is “hidden state”)

Neural Controlled Differential Equations

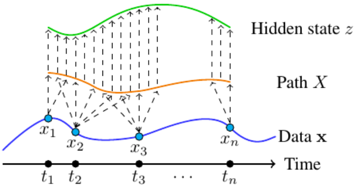
Mathematical Institute

$$z(0) = \zeta_\theta(t_0, x_0), \qquad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t), \qquad y \approx \ell_\theta(z(T))$$

Neural Controlled Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

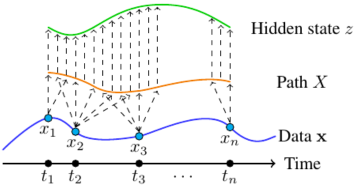
$$z(0) = \zeta_\theta(t_0, x_0), \qquad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t), \qquad y \approx \ell_\theta(z(T))$$

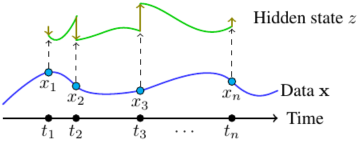


Neural Controlled Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

$$z(0) = \zeta_\theta(t_0, x_0), \qquad \frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t), \qquad y \approx \ell_\theta(z(T))$$





Neural Controlled Differential Equations

— Using a continuous-time theory pushes the problem of messy data into the spline interpolation, which is better suited for handling it. It doesn't need to affect the architecture of our model.

Neural Controlled Differential Equations

— Using a continuous-time theory pushes the problem of messy data into the spline interpolation, which is better suited for handling it. It doesn't need to affect the architecture of our model.

— Fixes a leaky abstraction.

Neural Controlled Differential Equations

— Using a continuous-time theory pushes the problem of messy data into the spline interpolation, which is better suited for handling it. It doesn't need to affect the architecture of our model.

- — Fixes a leaky abstraction.
- — Makes batching easy.

Neural Controlled Differential Equations

UNIVERSITY OF OXFORD
Mathematical Institute

— Using a continuous-time theory pushes the problem of messy data into the spline interpolation, which is better suited for handling it. It doesn't need to affect the architecture of our model.

— Fixes a leaky abstraction.

— Makes batching easy.

— The equation $\frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t)$ is still an ODE, so we can solve it with the same tools as for Neural ODEs.

UNIVERSITY OF OXFORD
Mathematical Institute

— Using a continuous-time theory pushes the problem of messy data into the spline interpolation, which is better suited for handling it. It doesn't need to affect the architecture of our model.

— Fixes a leaky abstraction.

— Makes batching easy.

— The equation $\frac{\mathrm{d}z}{\mathrm{d}t}(t) = f_\theta(z(t))\frac{\mathrm{d}X}{\mathrm{d}t}(t)$ is still an ODE, so we can solve it with the same tools as for Neural ODEs.

— In particular with the same software, hassle-free.

UNIVERSITY OF OXFORD
Mathematical Institute

— Because it's an ODE, we can use memory-efficient adjoint backpropagation.

— Because it's an ODE, we can use memory-efficient adjoint backpropagation.

— Let H be the cost of evaluating one 'step' of the model. Then alternatives (typically RNNs) use $\mathcal{O}(HT)$ memory. Here, we reduce it to just $\mathcal{O}(H + T)$.

Neural Controlled Differential Equations

— Because it's an ODE, we can use memory-efficient adjoint backpropagation.

— Let H be the cost of evaluating one 'step' of the model. Then alternatives (typically RNNs) use $\mathcal{O}(HT)$ memory. Here, we reduce it to just $\mathcal{O}(H + T)$.

— Neural CDEs demonstrate state-of-the-art performance.

Results!

CharacterTrajectories

Test accuracy (mean ± std, computed across five runs) and memory usage on CharacterTrajectories. Memory usage is independent of repeats and of amount of data dropped.

Model	Test Accuracy			Memory usage (MB)
	30% dropped	50% dropped	70% dropped	
GRU-ODE	89.9% ± 8.4%	89.6% ± 5.6%	86.6% ± 3.5%	1.5
GRU-Δt	94.4% ± 1.7%	92.0% ± 1.0%	91.1% ± 1.1%	15.6
GRU-D	93.2% ± 2.0%	92.7% ± 2.8%	90.8% ± 2.1%	16.9
ODE-RNN	97.9% ± 0.4%	97.5% ± 0.6%	96.7% ± 0.9%	14.7
Neural CDE (ours)	**99.2% ± 0.3%**	**99.3% ± 0.3%**	**99.4% ± 0.4%**	**1.3**

Results!

Speech Commands

UNIVERSITY OF OXFORD
Mathematical Institute

Test Accuracy (mean ± std, computed across five runs) and memory usage on Speech Commands. Memory usage is independent of repeats.

Model	Test Accuracy	Memory usage (GB)
GRU-ODE	47.9% ± 2.9%	**0.164**
GRU-Δt	43.3% ± 33.9%	1.54
GRU-D	32.4% ± 34.8%	1.64
ODE-RNN	65.9% ± 35.6%	1.40
Neural CDE (ours)	**89.8% ± 2.5%**	0.167

Summary

— New tool for time series

— Acts directly on irregularly sampled partially observed multivariate time series.

— Can be trained with memory-efficient adjoint backpropagation, even across observations

— Straightforward to implement with existing tools.

— Demonstrates state-of-the-art performance.

References

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

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.

Y. Rubanova, T. Q. Chen, and D. K. Duvenaud, "Latent Ordinary Differential Equations for Irregularly-Sampled Time Series," in *Advances in Neural Information Processing Systems 32*, pp. 5320–5330, Curran Associates, Inc., 2019