Способы учёта шума данных в модели нейронных дифференциальных уравнений

Владимиров Эдуард
vladimirov.ea@phystech.edu

Стрижов Вадим
strijov@phystech.edu

13 декабря 2022 г.

Аннотация

TODO

Ключевые слова: временной ряд · нейронные обыкновенные дифференциальные уравнения · нейронные стохастические дифференциальные уравнения · шум

1 Введение

Извлечение шума является важной частью предобработки данных, так как он ограничивает процесс извлечения информации. Для того, чтобы нивелировать эффект зашумлённости данных на качество предсказательной модели, необходимо либо заранее предобработать временной ряд, либо внедрить в неё механизм очистки данных, например аугментации, либо, наоборот, добавить шум к чистым данным.

Во многих случаях временные ряды создаются низкоразмерной динамической системой. Тогда загрязнение шумом временных рядов может привести к затруднению поиска оптимальной размерности эмбеддинга [1] и ограничению точности прогнозирования [2]. Поэтому снижение уровня шума при сохранении базовой динамики, генерируемой на основе временных рядов, имеет первостепенное значение.

Таблица 1: Классификация методов фильтрации временных рядов и их аналогов в фреймворке Neural ODE

Механизм предобработки	**Фильтры**	**Аналог в фреймворке NODE**
Разложение	вейвлет-преобразование, преобразование Фурье, метод главных компонент	—
Итеративный со скрытой динамикой	фильтр Калмана	непрерывный процесс потока времени (Continuous Time Flow Process, CTFP) [3], GRU-ODE-Bayes [4]
Итеративный	экспоненциальное сглаживание, модель Хольта-Брауна, пакетная нормализация	—
Введение случайности	dropout, добавление гауссовского шума	NSDE с разными компонентами перед дифференциалом от стандартного броуновского движения [5]

Существует несколько основных методов предобработки сигналов. Фильтр Калмана [6] использует информацию о физике самого явления. Вейвлет преобразование [7] раскладывает временной ряд на множество других с разным разрешением. Экспоненциальное сглаживание [8] выравнивает временной ряд с помощью экспоненциальной оконной функции. В работе [9] фильтрация одномерного временного ряда выполняется в несколько этапов. Вначале строится матрица Ханкеля на основе теоремы Такенса [10]. Затем выполняется сингулярное разложение этой матрицы с сохранением некоторого числа первых компонент. После этого исходный ряд восстанавливается как среднее значение элементов на побочных диагоналях.

Модель нейронных дифференциальных уравнений (Neural Ordinary Differential Equation, NODE) [11] объединяет в себе нейронные сети и дифференциальные уравнения. Помимо использования в моделировании динамических системах, она также применяется в задачах интерполяции, экс-

траполяции и генерации временных рядов, в частности в финансах, робототехнике [12, 13, 14]. Данный механизм является непрерывным аналогом сети ResNet [15], в которой впервые использовались остаточные соединения для решения проблемы затухающих градиентов. NODE объединяет в себе плюсы из обоих миров. Структура нейронной сети даёт высокую обобщающую способность, а структура дифференциального уравнения даёт твёрдый теоретический фундамент. Однако у этого фреймворка есть и минусы. Существует класс функций, которые не представимы этой моделью [16], а также она плохо устойчива к шуму в данных [5].

Для решения вышеизложенных проблем было разработано новое семейство нейронных сетей — стохастические дифференциальные уравнения (Neural Stochastic Differential Equation, NSDE). Она внедряет шум в модель NODE с помощью стохастического дифференциального уравнения (SDE). Данный фреймворк позволяет эксплуатировать такие регуляризаторы, как Dropout и слой гауссовского шума [17, 18]. Однако остаётся вопрос: можно ли внедрить другие регуляризаторы (например, BatchNorm) и вышеупомянутые методы фильтрации в фреймворк NODE? В данной работе планируется ответить на этот вопрос.

[TODO: Краткое описание вычислительного эксперимента]

2 Сопутствующие работы

Наша работа вдохновлена успехом модели Neural ODE и её многочисленных модификаций: ANODE, Neural SDE, Neural CDE. Мы стремимся перенести некоторые методы фильтрации временных рядов в модель нейронных дифференциальных уравнений. Однако такие регуляризаторы, как пакетная нормализация и вейвлет преобразование, тяжело внедрить в Neural ODE ввиду отсутствия скрытой динамики.

2.1 Neural ODE

Как было сказано ранее, модель нейронных дифференциальных уравнений является непрерывным расширением модели ResNet, в которой параметризуется производная скрытого состояния.

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f_{\theta_t}(\mathbf{h}_t).$$

Разность $\mathbf{h}_{t+1} - \mathbf{h}_t$ можно интерпретировать как дискретизацию производной $\mathbf{h}'(t)$ с разницей во времени $\Delta t = 1$. Устремив $\Delta t \to 0$, получим:

$$\lim_{\Delta t \to 0} \frac{\mathbf{h}_{t+\Delta t} - \mathbf{h}_t}{\Delta t} = \frac{d\mathbf{h}(t)}{dt} = f_\theta(\mathbf{h}(t), t)$$

Модель нейронных дифференциальных уравнений аппроксимирует отображение $\mathbf{x} \to \mathbf{y}$ путём обучения нейронной сети f_θ и линейных отображений l_θ^1, l_θ^2.

$$\mathbf{y} \approx l_\theta^2(\mathbf{h}_T), \text{ где } \mathbf{h}_T = \mathbf{h}_0 + \int_0^T f_\theta(\mathbf{h}_t)dt \text{ и } \mathbf{h}_0 = l_\theta^1(\mathbf{x})$$

2.2 Neural SDE

Конструкция выглядит следующим образом. Зафиксируем момент времени $T > 0$. Обозначим за $(\mathbf{Y}_t, t \in [0, T]$ искомый y-мерный случайный процесс, где y — размерность данных.

Пусть $(\mathbf{W}_t, t \in [0, T])$ — w-мерный винеровский процесс и $V \sim \mathcal{N}(0, I_v)$ — v-мерный стандартный гауссовский вектор. Пусть

$$\zeta_\theta : \mathbb{R}^v \to \mathbb{R}^d,\ \mu_\theta : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d,\ \sigma_\theta : [0, T] \times \mathbb{R}^d \to \mathbb{R}^{d \times w},\ l_\theta : \mathbb{R}^d \to \mathbb{R}^y,$$

где d — размерность скрытого состояния, $\zeta_\theta, \mu_\theta, \sigma_\theta$ — нейронные сети, l_θ — линейное преобразование. Тогда модель Neural SDE имеет следующий вид:

$$\mathbf{h}_0 = \zeta_\theta(V), \quad d\mathbf{h}_t = \mu_\theta(t, \mathbf{h}_t)dt + \sigma_\theta(t, \mathbf{h}_t) \circ d\mathbf{W}_t, \quad \widehat{\mathbf{Y}_t} = l_\theta(\mathbf{h}_t).$$

Решением SDE служит случайный d-мерный процесс $(\mathbf{h}_t, t \in [0, T])$. Модель обучается таким образом, что $\widehat{\mathbf{Y}} \approx \mathbf{Y}$

В работе [5] показано, что добавление гауссовского шума выражается следующим образом: $\sigma(t, \mathbf{h}_t) = \Sigma(t)$, где $\Sigma(t)$ — диагональная матрица с элементами, отвечающими за дисперсию шума, добавляемого к вектору скрытого состояния. А слой Dropout представляется так: $\sigma(t, \mathbf{h}_t) = \sqrt{\frac{1-p}{p}}\mu_\theta(t, \mathbf{h}_t)$, где p — вероятность успеха в схеме испытаний Бернулли.

2.3 ResNet

Нетрудно заметить, что блок 2.1 модели ResNet соответствует одному шагу метода Эйлера численного интегрирования уравнения 2.1. Учитывая этот

факт и то, что существует множество регуляризаторов для данного семейства моделей: DropBlock, Shake-Shake [TODO: ссылки]; хочется найти их аналоги для фреймворка Neural ODE.

3 Постановка задачи

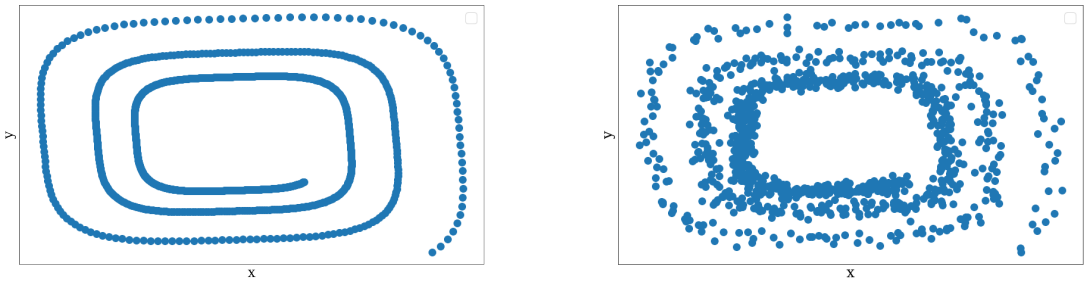
Общая схема данных в задаче удаления шума имеет вид:

$$\mathbf{x}_t = \mathbf{s}_t + \boldsymbol{\varepsilon_t},$$

где $\mathbf{x}_t$ — наблюдаемое значение временного ряда, $\boldsymbol{\varepsilon_t}$ — многомерный винеровский процесс, $\mathbf{s}_t$ — истинное значение временного ряда. Задача удаления шума заключается в подавлении шумовой составляющей временного ряда $\mathbf{x}_t$ и восстановлении $\mathbf{s}_t$.

4 Вычислительный эксперимент

Сначала покажем, что качество модели Neural ODE уменьшается при наличии шума в данных. Взят синтетический временной ряд "Спираль и его зашумлённый вариант с добавленным нормальным шумом $\mathcal{N}(0, 0.04^2)$ (график 1).



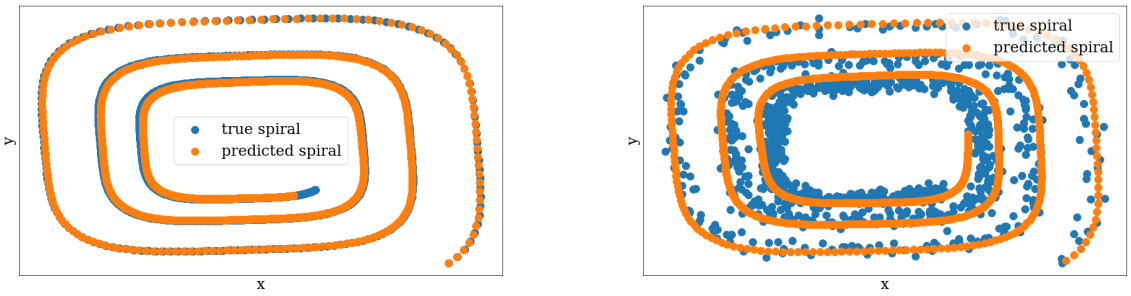
(a) Временной ряд "Спираль" (b) Зашумлённая спираль

Рис. 1: Синтетические данные

Результат применения модели Neural ODE и графики функции потерь при обучении приведены на графиках 2 и 3.

5 Заключение

TODO



(a) NODE на чистых данных

(b) NODE на зашумлённых данных

Рис. 2: Результаты предсказания

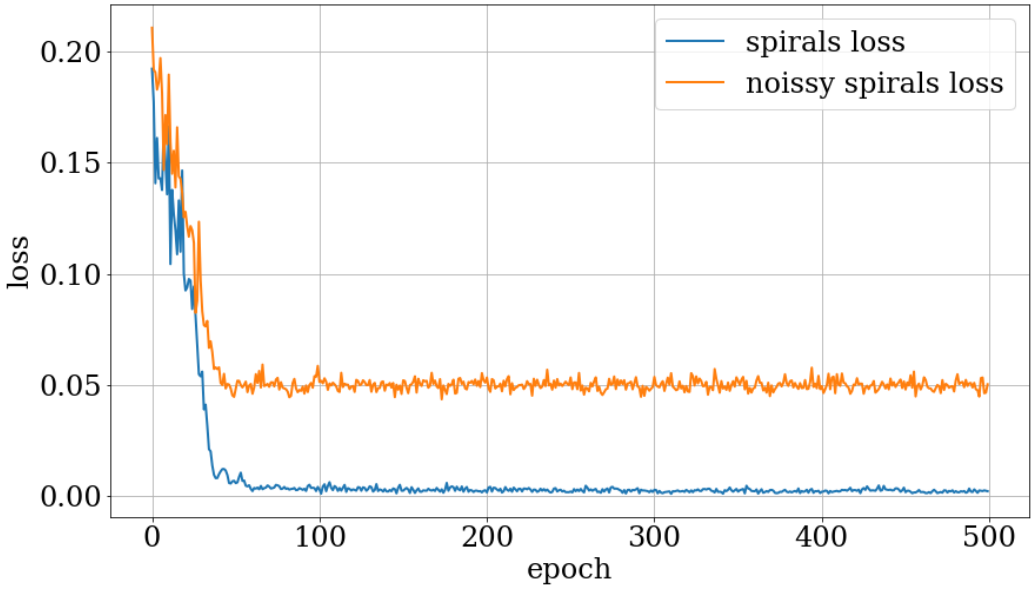


Рис. 3: Функция потерь при обучении Neural ODE на незашумлённых и зашумлённых данных

Список литературы

[1] Eric J Kostelich and James A Yorke. Noise reduction: Finding the simplest dynamical system consistent with the data. *Physica D: Nonlinear Phenomena*, 41(2):183–196, 1990.

[2] A Elshorbagy, SP Simonovic, and US Panu. Noise reduction in chaotic hydrologic time series: facts and doubts. *Journal of Hydrology*, 256(3-4): 147–165, 2002.

[3] Ruizhi Deng, Bo Chang, Marcus A. Brubaker, Greg Mori, and Andreas M. Lehrmann. Modeling continuous stochastic processes with dynamic normalizing flows. *Advances in Neural Information Processing Systems*, 2020-Decem(NeurIPS):1–11, 2020. ISSN 10495258.

[4] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. *Belgian/Netherlands Artificial Intelligence Conference*, (NeurIPS):364–366, 2020. ISSN 15687805.

[5] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural SDE: Stabilizing Neural ODE Networks with Stochastic Noise. (2), 2019. URL http://arxiv.org/abs/1906.02355.

[6] Kirti Dhwaj, Joshua M. Kovitz, Haozhan Tian, Li Jun Jiang, and Tatsuo Itoh. Half-mode cavity-based planar filtering antenna with controllable transmission zeroes. *IEEE Antennas and Wireless Propagation Letters*, 17 (5):833–836, 2018. ISSN 15361225. doi:10.1109/LAWP.2018.2818058.

[7] Rumaih M. Alrumaih and Mohammad A. Al-Fawzan. Time Series Forecasting Using Wavelet Denoising an Application to Saudi Stock Index. *Journal of King Saud University - Engineering Sciences*, 14(2):221–233, 2002. ISSN 10183639. doi:10.1016/S1018-3639(18)30755-4. URL http://dx.doi.org/10.1016/S1018-3639(18)30755-4.

[8] Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28, 1985.

[9] Andrey D. Ignatov and Vadim V. Strijov. Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. *Multimedia Tools and Applications*, 75(12):7257–7270, 2016. ISSN 15737721. doi:10.1007/s11042-015-2643-0. URL http://dx.doi.org/10.1007/s11042-015-2643-0.

[10] Lyle Noakes. The takens embedding theorem. *International Journal of Bifurcation and Chaos*, 1(04):867–872, 1991.

[11] David Duvenaud Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt. Ordinary Differential Equations. *UNITEXT - La Matematica per il 3 piu 2*, 109(NeurIPS):31–60, 2018. ISSN 20385757. doi:10.1007/978-3-662-55774-7_3.

[12] Xiang Xie, Ajith Kumar Parlikad, and Ramprakash Srinivasan Puri. A neural ordinary differential equations based approach for demand forecasting within power grid digital twins. In *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2019.

[13] Zoya Meleshkova, Sergei Evgenievich Ivanov, and Lubov Ivanova. Application of neural ode with embedded hybrid method for robotic manipulator control. *Procedia Computer Science*, 193:314–324, 2021.

[14] Jianzhun Du, Joseph Futoma, and Finale Doshi-Velez. Model-based reinforcement learning for semi-markov decision processes with neural odes. *Advances in Neural Information Processing Systems*, 33:19805–19816, 2020.

[15] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90: 119–133, 2019.

[16] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural ODEs. *Advances in Neural Information Processing Systems*, 32(NeurIPS): 1–11, 2019. ISSN 10495258.

[17] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018.

[18] Pierre Baldi and Peter J Sadowski. Understanding dropout. *Advances in neural information processing systems*, 26, 2013.