
Neural SDE: phase trajectories of SDE in the action

A Preprint

Papay Ivan
papai.id@phystech.edu

Vladimirov Eduard
vladimirov.ea@phystech.edu

Strijov Vadim
strijov@phystech.edu

2024 год

Данная статья предлагает развить математический аппарат, на котором строится модель Neural SDE. В ней рассмотрено, как вычисление фазовых траекторий стохастических дифференциальных уравнений обеспечивает качественный прогноз аномалий во временном ряду. Таким образом, это предоставит как возможность эффективнее бороться с шумами, так и, в частности, полезный инструмент для упреждения появления так называемых ‘черных лебедей’ - резких перемен поведения временных рядов, как случайных процессов. Подобного рода аномалии способны нарушить корректную работу Neural SDE в виду высокой корреляции элементов анализируемой выборки между собой.

Keywords SDE · Stratonovich integral · Breakdown Points · More

1 Введение

Сбор данных и подготовка их к последующей обработке являются одной из важнейших задач машинного обучения. Не всегда исследователь может гарантировать их целостность и корректность, ведь для тренировки модели чаще всего требуются выборки из тысяч, а то и десятков тысяч элементов — не удивительно, что в данных допускается наличие шума, влияющего на работу обученной модели. Эта задача остаётся актуальной и для временных рядов. Нужно, имея данные для начала временного ряда, проверить: возможно ли предсказать его на некотором горизонте так, что это предсказание будет точным, и не сломает ли это природу текущего временного ряда в вероятностном смысле?

В данной работе предполагается, что природа данных стохастическая. Входная выборка представляет из себя данные о некоем дискретном случайном процессе. От дискретного процесса корректно будет перейти к непрерывному в силу того факта, что он порождается сигма-алгеброй из конечномерных распределений, которые реально аппроксимировать с помощью данных, предоставленных для обучения модели.

Под данными имеются в виду значения n -мерного случайного процесса в фиксированном конечном множестве точек, которые необходимо будет сжать посредством CCA + CCM (Convergent-Cross Mapping)[6]. В таком случае в качестве варианта аппроксимации полученного одномерного временного ряда предлагается взять метод приближения обыкновенными дифференциальными уравнениями, от которых затем уже перейти к стохастическим.

Сама идея использования обыкновенных дифференциальных уравнений (‘ОДУ’) не нова[1][8]. Так, примерно с 2017-го года она была использована[2] для создания и теоретического обоснования корректности работы модели Neural ODE. Тем не менее, такой метод был всё ещё слаб в робастном смысле[2]: был уязвим к состязательным атакам. Модель Neural SDE[2-4] уже строилась на использовании стохастических дифференциальных уравнений (‘СДУ’) и была в этом плане эффективнее своего предшественника.

Главной целью данного исследования является построение decision-rejection (принятие-отрицание) критерия корректности той или иной гипотезы о вероятностном распределении входных данных, как некоторого непрерывного случайного процесса. Для проверки фрагмента временного ряда на наличие аномалий достаточно применить этот критерий для проверки гипотезы о тождественности распределений его и всего остального ряда — разумно будет заключить, что в ряду происходят аномалии, если природа данных в стохастическом смысле резко поменялась.

Таким образом, полагая, что временной ряд порождается определенными конечномерными распределениями, мы сможем приблизить его с помощью стохастических дифференциальных уравнений. То же применимо и к анализируемому диапазону, который требуется проверить на наличие аномалий. Если фазовые траектории полученных дифференциальных уравнений различаются, то есть происходит резкое их возмущение, то очевидно, что в ряду произошла аномалия. Такого рода аномалии отсюда и далее мы будем называть точками разладки.

В прошлых работах, связанных с Neural SDE[2-4], СДУ использовались только для построения доверительных интервалов для элементов временного ряда. Этот подход в статье предлагается развить посредством использования фазовых траекторий полученных СДУ. Таким образом, можно будет проверять большие массивы данных на корреляцию между собой. В работе рассматривается задача — найти точки разладки для фиксированного временного ряда. То есть для каждой из точек временного ряда проверяется decision-rejection критерий о согласованности поведения ряда в этой конкретной точке и всего остального ряда.

Будет уместным сказать, что ход решения задачи будет схож с методами генеративных моделей[12], а не дискриминативными[9][11]. Требуется предъявить читателю не столько новый способ отделить точки разладки, сколько метод оценки вероятности того, что та или иная точка таковой является. Для корректного решения проблемы следует взять лучшее из обоих подходов аналогично тому, как некоторые исследователи уже делали раньше[10].

Ниже детально разобрано, как по СДУ, построенным по временным рядам, получить фазовые траектории путём зануления диффузии СДУ и его семплирования, как случайной величины. В качестве decision-rejection критерия используется сравнение полученных траекторий в плане их главных компонент.

2 Связанные работы

2.1 Точки разладки

Следует тщательно разобраться в том, что означает такое понятие, как чёрный лебедь, заявленное в аннотации статьи. Пуск имеет некоторый временной ряд. Стохастическая природа его известна, но в определенный момент она меняется. Этот момент называется моментом разладки.

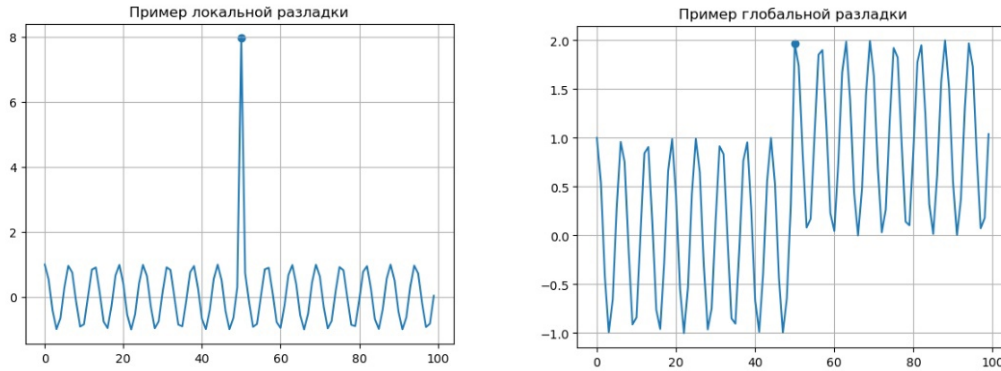


Рис. 1: примеры разладок

Точка разладки является моментом локальной разладки, если при игнорировании её природа всего временного ряда в стохастическом плане останется однородной.

Глобальной же, если по достижению оной всё поведение временного ряда резко меняется.

2.2 Neural ODE

Изначально Neural ODE был разработан как альтернатива методу остаточных нейронных сетей, состоящих из последовательности скрытых слоёв, значения на каждом из которых подчинялись следующей формуле:

$$h_{k+1} = h_k + f(h_k, w_k) \quad (1)$$

— где h_k - вход k -го слоя для $k \in [1, K]$, K -число слоёв и $f(h_k, w_k)$ - нелинейная функция, параметризованная по w_k - динамический параметр, задающийся непосредственно перед началом обучения модели.

Было предложено[5] представление (1) в виде:

$$h_t = h_s + \int_s^t f(h_l, l; w) dl, \quad (2)$$

— вычисление такого дифференциального уравнения является задачей для Neural ODE. В данной работе в качестве числа слоев берется число элементов во временном ряду.

Algorithm 1 Neural ODE-solver

Require: динамические параметры w , начальное/конечное время t_0, t_1 , конечное значение $z(t_1)$, градиент функции потерь в конечной точке $\frac{\delta L}{\delta z(t_1)}$

$s_0 = [z(t_1), \frac{\delta L}{\delta z(t_1)}, 0_{[w]}]$ ▷ Начальное состояние

$[z(t_0), \frac{\delta L}{\delta z(t_1)}, \frac{\delta L}{w}] = \text{ODESolve}(s_0, [f(z(t), t, w), -a(t)^T \frac{\delta f}{\delta z}, -a(t)^T \frac{\delta f}{\delta w}], t_1, t_0, w)$

return $\frac{\delta L}{\delta z(t_0)}, \frac{\delta L}{w}$ ▷ Возвращаем градиенты

Здесь ODEsolve - это черный ящик, возвращающий решение ODE — в качестве него предлагается использовать метод Рунге-Кутты, а L — MSE (mean squared error — среднее квадратов отклонения элементов выборки от их оценок).

2.3 Neural CDE

В случае анализа временных рядов для выражения(2) было предложено[7] рассматривать не классический интеграл Римана-Стилтьеса, а стохастический интеграл Стратоновича:

$$h_t = h_s + \int_s^t f(h_l, l; w) dB_l, \quad (3)$$

— где $B_t = [B_t^1 \dots B_t^K]$ - Винеровский процесс той же размерности, что и X_t .

Таким образом получается контролируемое дифференциальное уравнение, подходящее для данных типа Time-Series, особенно когда временные метки ряда не являются регулярными в плане тождественности их расстояний между собой.

2.4 Neural SDE

Обобщением предыдущих двух методов является модель Neural SDE.

Для учёта шума и максимально возможной аппроксимации прогноза в нашем дифференциальное уравнение следует учесть и недетерминированную компоненту, и диффузию. Получится следующее выражение, являющееся полным стохастическим дифференциалом:

$$dX_t^w = h(t, X_t^w; w)dt + \sigma(X_t^w; w)dB_t \quad (4)$$

— где $B_t = [B_t^1 \dots B_t^K]$ - Винеровский процесс той же размерности, что и X_t , а $\sigma(X_t^w; w)$ - его матрица ковариаций в t -й момент времени

Обобщим это выражение для модели ResNet:

$$dh_t = f(h_t, t, w), \quad (5)$$

— таким образом выражение (1) для $(k+1)$ -го слоя изменится:

$$h_{k+1} = h_k + f(h_k, w_k) + \sigma(X_k^w; w)B_k, \quad (6)$$

— соответственно алгоритм остаётся тем же, что и для ODE с поправкой на вычисление матрицы ковариации путём семплирования входной выборки.

3 Постановка задачи

В общем виде для решения поставленной задачи требуется осуществить следующие шаги - применить метод Neural ODE к временному ряду, учесть гауссовский шум и тем самым свести задачу к модели Neural SDE, вычислить фазовые траектории для временного ряда, предварительно свернув исходный многомерный временной ряд к минимально возможному размеру и, наконец, сравнить полученные фазовые траектории временных рядов по поведению с отдельными точками, проверяемыми на элемент разладки. Далее предлагается вкратце их разобрать.

Исходный временной ряд соответственно:

$$\{X_{t_i}\}_{i=1}^n \quad (7)$$

— где t_i - временные метки, упорядоченные по возрастанию. Предполагается, что траектория ряда подчиняется некоторому СДУ (см. 2.2)

Решив СДУ с помощью SDE-solver-a мы получим точные значения $f(h_n, w_n) = C_n - \text{const}$ (см. (4)) для $\forall n$. Тогда в той же формуле (4) приняв во внимание факт того, что $EB_t = 0$, занулим коэффициент диффузии и получим следующее выражение:

$$h_{n+1} = h_n + C_n \quad (8)$$

— это оценки приращений элементов временного ряда. По ним строится матрица Ганкеля, описывающая фазовые траектории процесса, размера $n/2$ на $n/2$ вида:

$$\begin{bmatrix} C_1 & C_2 & \cdots & C_{n/2} \\ C_2 & C_3 & \cdots & C_{n/2+1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n/2} & C_{n/2+1} & \cdots & C_n \end{bmatrix} \quad (9)$$

Таким образом аналогично методу SSA получим оценки на значения исходного временного ряда. Среднее каждого из элементов оценивается выборочным средним значений на соответствующей антидиагонали матрицы Ганкеля, а его разброс выборочной дисперсией.

В таком случае - хоть и в силу неточности приближения мы неспособны прогнозировать дальнейшее поведение временного ряда, но момент разладки найти возможно. Если сэмплировать получить оценки 95-й и 5-й квантили, зная средние и дисперсии каждого из элементов ряда, то тем самым возможно получить доверительный интервал для его значений. Если в тот или иной момент ряда он резко вышел из этого диапазона, то можно сказать, что произошла разладка.

4 Вычислительный эксперимент

Продemonстрируем ход алгоритма, описанного выше, на практике и явно покажем, как осуществляется процесс поиска точек разладки временного ряда.

Эксперимент проводится над выборкой из $N = 150$ элементов. В качестве тестовой выборки берётся искусственно зашумленный ряд синусов.

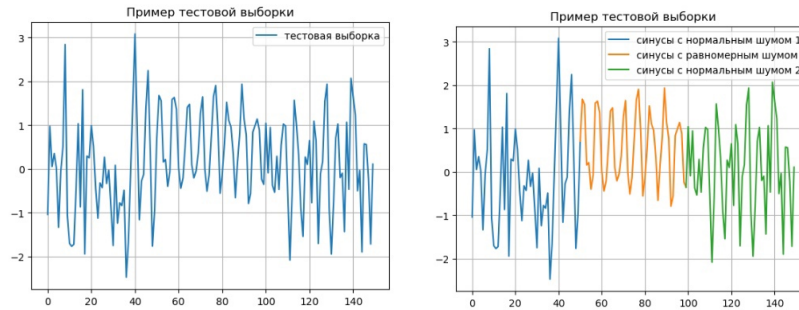


Рис. 2: тестовая выборка

Таким образом явное выражение n -ого члена временного ряда выражается по формуле:

$$X_n = \sin(n) + \xi_n \quad (10)$$

— где ξ_n случайная величина, распределение которой задаётся следующим образом:

$$\xi_n \sim \begin{cases} \mathcal{N}(0, 1) : n < 50 \text{ or } n > 100 \\ \mathcal{U}(0, 1) : 50 \leq n \leq 100 \end{cases}$$

С помощью модели Neural SDE, реализованной библиотекой torchsde, строится прогноз продолжения полученного ряда.

По приращениям элементов прогноза получим матрицу фазовых траекторий процесса и сравнивая её с целевой матрицей Ганкеля на фиксированном окне запустим процесс обучения модели. Цель: подобрать параметры drift и diffusion такие, чтобы отклонение от target-а было минимальным.

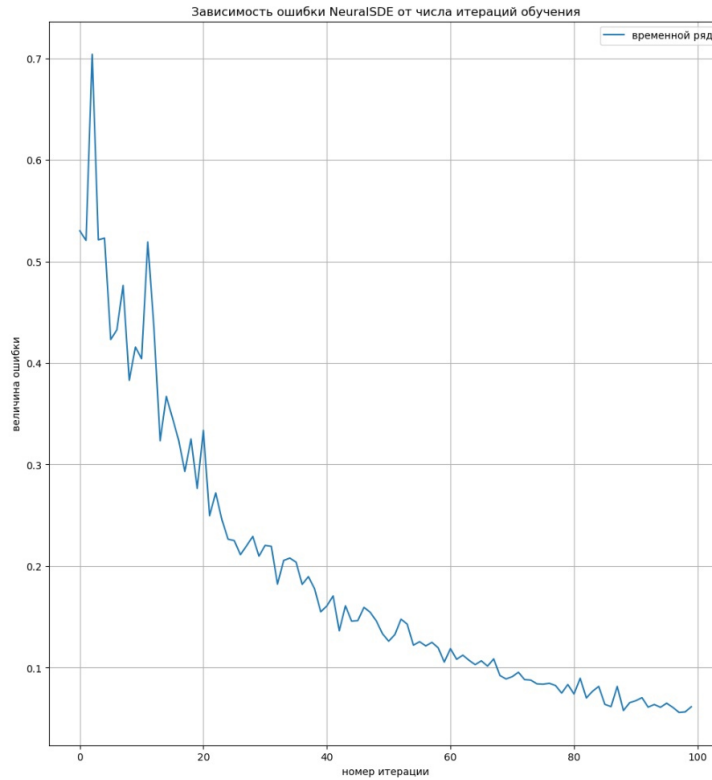
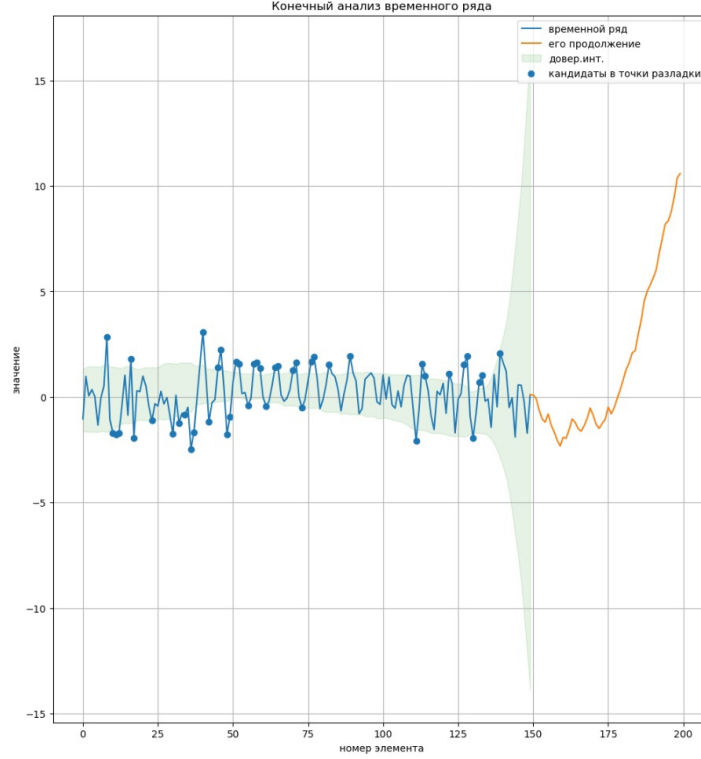


Рис. 3: процесс обучения

По итогу обучения получаем следующий результат работы алгоритма на исходном временном ряду.



Доверительные интервалы пускай и не точно, но позволяют оценить примерное число точек разладки и место их наибольшего сосредоточения: как раз тот самый отрезок ряда от 50-го до 100-го элементов, который был ранее зашумлен нормальным шумом, а не равномерным в отличие от остальных.

Итоговое сравнение с другими методами поисками точек разладки показывает то, что NeuralSDE выигрывает в плане точности, но уступает в плане скорости.

$$\begin{bmatrix}
 \textit{Model} & \textit{Speed(sec)} & \textit{Error} \\
 \textit{NeuralSDE} & 47.2 & 0.044 \\
 \textit{SSA} & 6.48 & 0.12 \\
 \textit{CuSum} & 3.56 & 0.08 \\
 \textit{CuSumSqr} & 20.03 & 0.061
 \end{bmatrix} \quad (11)$$

5 Заключение

Результатом проведенной работы является обоснование корректности нового способа поиска точек разладки на практике. Плюсы и минусы метода ясны, и зная их, исследователь способен использовать его как новый и полезный инструмент.

Список литературы

- [1] “Neural Ordinary Differential Equations Ricky” T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud
- [2] “Neural SDE: Stabilizing Neural ODE Networks with Stochastic Noise” Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, Cho-Jui Hsieh
- [3] “Riemannian Neural SDE: Learning Stochastic Representations on Manifolds” Sung Woo Park , Hyomin Kim , Kyungjae Lee , Junseok Kwon
- [4] “Riemannian Diffusion Models” Chin-Wei Huang, Milad Aghajohari, Avishek Joey Bose, Prakash Panangaden, Aaron Courville
- [5] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In Advances in Neural Information Processing Systems, pages 6572–6583, 2018.

-
- [6] F.Yu. Yaushev, R. V. Isachenko, V. V. Strijov. Concordant models for latent space projections in forecasting, 2020.
 - [7] Neural Controlled Differential Equations for Irregular Time Series Patrick Kidger, James Morrill, James Foster, Terry Lyons
 - [8] Neural Ordinary Differential Equations Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud
 - [9] Discriminative models for multi-instance problems with tree-structure Tomas Pevny, Petr Somol
 - [10] Training Discriminative Models to Evaluate Generative Ones Timothée Lesort, Andrei Stoain, Jean-François Goudou, David Filliat
 - [11] Discriminative Neural Topic Models Gaurav Pandey, Ambedkar Dukkipati
 - [12] Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models Sam Bond-Taylor, Adam Leach, Yang Long, Chris G. Willcocks