Анализ временных рядов методами DL

Введение в эффективные системы ML
Спецкурс ММП ВМК МГУ
Алексеев Илья, @voorhs

18 окт 2024

1 Введение

§1.1 Что такое временные ряды

Одномерным временным рядом (univariate time series) мы будем называть числовую последовательность $x_1, \dots, x_T$. Число $x_t \in \mathbb{R}$ мы называем наблюдением, или замером, (observation) в момент времени t (timestamp). Эти названия отражают, в каких областях обычно возникают временные ряды: замер температуры воздуха, концентрации вещества и проч.

Многомерным временным рядом (multivariate time series) мы будем называть последовательность векторов $x_1, \dots, x_T$. В данном случае наблюдением будет целый вектор $x_t \in \mathbb{R}^C$. Он представляет собой замеры сразу нескольких величин. Например, вместе с температурой можно мерить давление и влажность воздуха. Отличие C-мерного временного ряда от набора из C одномерных временных рядов в том, что в многомерном ряде обычно предполагают, что компоненты вектора x_t относятся к одному моменту времени.

§1.2 Задачи анализа временных рядов

Прогнозирование (forecasting) временного ряда заключается в предсказании неизвестных будущих значений временного ряда на основе известных исторических значений. Обычно разделяют на два подтипа: краткосрочное (short-term) и долгосрочное прогнозирование (long-term forecasting). Эти задачи отличаются не временными промежутками (часы, дни, месяцы, годы), а количеством наблюдений (десятки, сотни, тысячи), которое требуется предсказать. Такое разделение объясняется двумя причинами. В первую очередь — тем, что модель, дающая на выходе тысячи векторов должна быть вычислительно эффективной. Во вторую очередь — тем, что не все методы способны промоделировать зависимости между тысячами векторов.

Классификация временных рядов заключается в присвоении метки всему ряду. Самая популярный пример задачи — это классификация ЭКГ на нормальные и патологические.

Обнаружение аномалий (anomaly detection) состоит в выявлении атипичных значений или сегментов временного ряда. Чаще всего встречается для выявления сбоев в работе оборудования по замерам с датчиков.

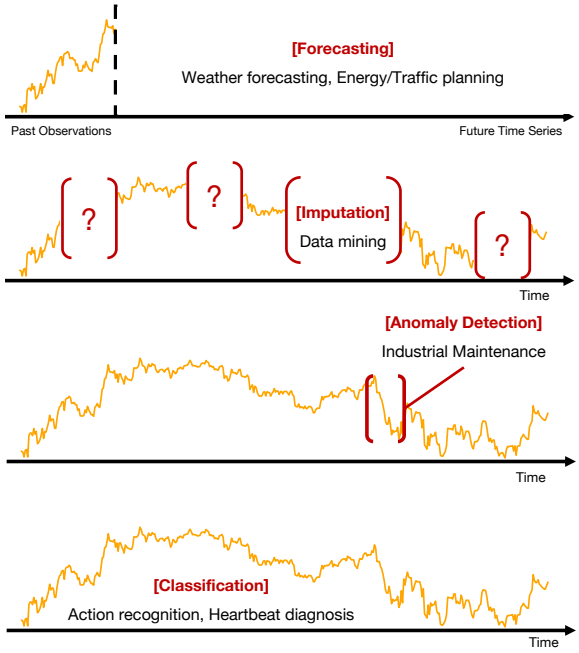

Рис. 1. Задачи анализа временных рядов.

Заполнение пропусков (imputation) — это восстановление пропущенных значений.

2 Нейросетевые методы для работы с временными рядами

Для анализа временных рядов можно применять любые нейросети, работающие с последовательностями: CNN, RNN, Transformer. Если длина временного ряда никогда не меняется, можно применять даже MLP.

§2.1 Применение полносвязных сетей

- для классификации можно построить сеть, сужающуюся до слоя шириной в число классов
- для прогнозирования можно
  - построить сеть, сужающуюся до слоя шириной в число прогнозируемых значений [2]
  - а можно использовать авто-регрессионное прогнозирование
- можно применять современные MLP-based архитектуры вроде MLP-mixer, FNet, gMLP [1]

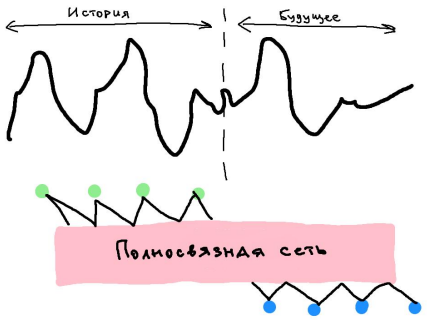

Рис. 2. MLP forecasting

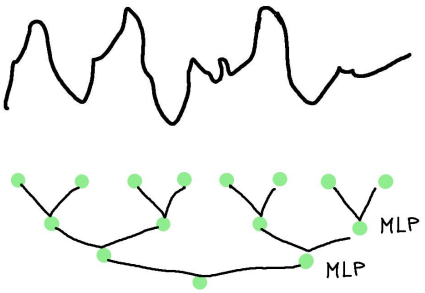

Рис. 3. MLP classification

§2.2 Применение свёрточных сетей

- для классификации можно построить сеть, сужающуюся до слоя шириной в число классов [5]
- для прогнозирования можно
  - построить сеть, сужающуюся до слоя шириной в число прогнозируемых значений
  - либо использовать казуальные свёртки в авторегрессионном режиме [3,6,7]

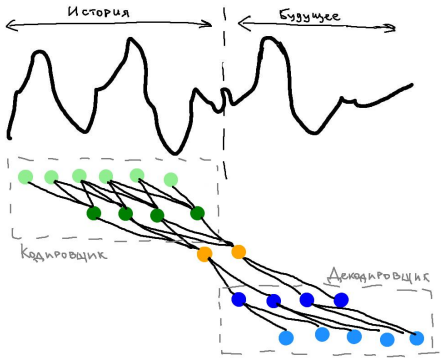

Рис. 4. CNN forecasting

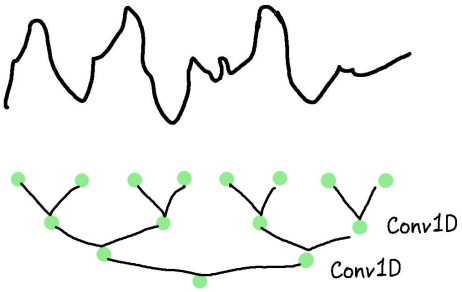

Рис. 5. CNN classification

§2.3 Применение рекуррентных сетей

- для классификации можно
  - построить обычную сеть (даже двунаправленную) и взять выходное представление с последнего момента времени,
  - либо построить сужающуюся сеть (что в целом для RNN экзотика, но возможно)

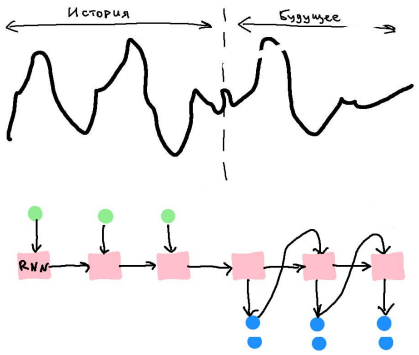

Рис. 6. RNN forecasting

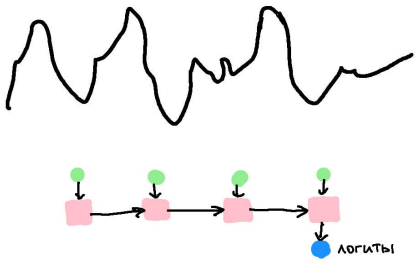

Рис. 7. RNN classification

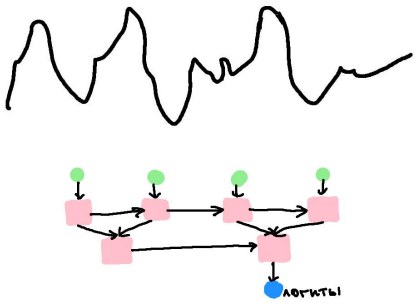

Рис. 8. RNN classification

- для предсказания можно использовать обычную казуальную архитектуру [4]

Есть еще экзотические и передовые технологии SSM [10]. Если вкратце, то это RNN вычисляющая коэффициенты оптимальной аппроксимации исторических данных с помощью линейной комбинации системы полиномов Лагранжа или Лежандра. Этот материал не является предметом настоящего занятия.

§2.4 Применение трансформеров

- для классификации можно использовать двунаправленный трансформер и агрегировать представления с последнего слоя
- для предсказания можно
  - использовать трансформер в казуальном режиме,
  - но чаще всего берут энкодер-декодер [8, 9]. В декодере не используют авторегрессионное предсказание, вместо этого (поскольку известна длина выходной последовательности) используют плейсхолдеры, которые после прохода через каждый слой как бы заполняются новой информацией и формируют итоговое предсказание

Чаще всего признаки временного ряда не подаются в сыром виде в нейросеть. Помимо прочих препроцессингов, всегда используется проецирование с помощью од-

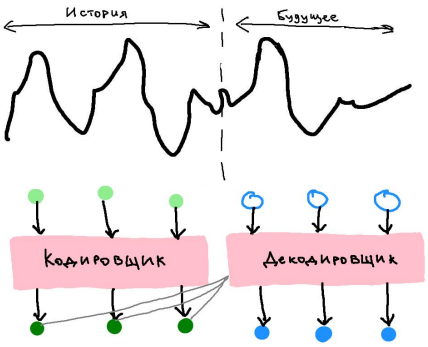

Рис. 9. Transformer forecasting

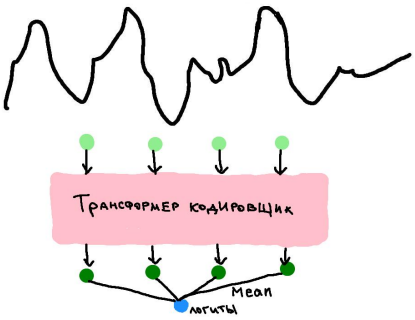

Рис. 10. Transformer classification

ного или нескольких полносвязных слоев в большую размерность вроде 96 или 128. Такие представления заменяют эмбеддинги.

3 Архитектуры

§3.1 Работа с периодичными рядами. TimesNet

Большинство временных рядов, встречающихся на практике, имеют периоды. Поэтому очень полезно уметь их выделять. Проще всего это сделать с помощью быстрого преобразования Фурье. Оно переводит функцию времени $f(t)$ в функцию частоты $F(\nu)$. Если пики функции $f(t)$ соответствуют мощности замера t, то пики функции $F(\nu)$ — степени присутствия частоты ν.

Допустим, мы выделили одну из пиковых частот ν. Мы можем порезать весь временной ряд на отрезки длины $T\nu$ и сопоставить их друг с другом. Тогда мы получим двумерное представление временного ряда в виде матрицы, где в столбцах находятся замеры из одного кусочка, а в строках точки — точки из разных кусочков, но с одинаковой фазой относительно периода $1/\nu$. Это представление обладает важным свойством: оно интерпретируемо как изображение. Действительно, "пиксели соседние данному, должны быть очень похожи.

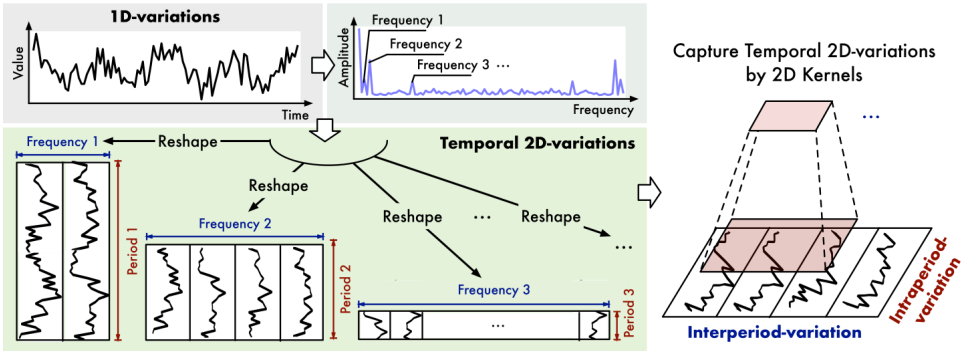

Рис. 11. Фурье-анализ и получение двумерного представления временного ряда

Архитектура TimesNet [11] состоит из последовательности TimesBlock'ов, каждый из которых выделяет k пиковых частот, формирует двумерные представле-

ния для каждой частоты и применяет к ним двумерную свёртку, как будто это изображения. Результирующие преобразования агрегируются и формируют выход TimesBlock'а. Такое разделение на k представлений играет роль, схожую той, что головы играют в мультиголовочном аттеншене.

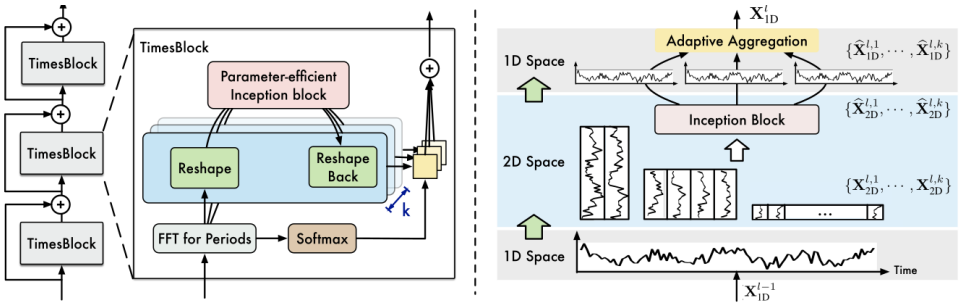

Рис. 12. Архитектура TimesNet

§3.2 Работа с нестационарными временными рядами

Временной ряд называют стационарным, если со временем среднее и дисперсия его значений не меняются. Нестационарные ряды наиболее тяжелы для анализа методами DL. Это связано с известным фактом, что нейросети тяжело учатся, когда на вход подаются величины с постоянно меняющейся магнитудой. Особо осведомленные читатели знают, что причиной тому стохастическая оптимизация.

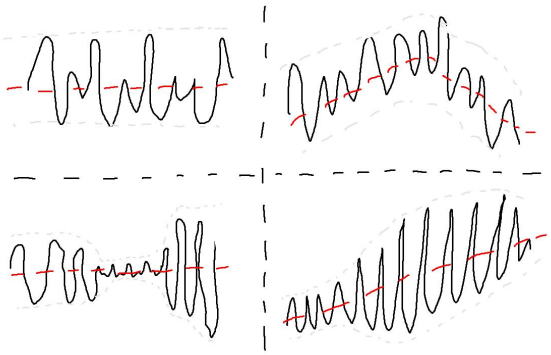

Рис. 13. Примеры нестационарных временных рядов

Трендом временного ряда называют долгосрочное изменение. Другими словами, это наиболее общее направление, которое можно получить сглаживанием или усреднением с помощью скользящего окна. Тренд связан с понятием среднего. Если произвести преобразование временного ряда таким образом, чтобы тренд и дисперсия стали константными, то в данных останутся только мелкомасштабные изменения, которые гораздо проще моделировать. Такое преобразование называют стационаризацией временного ряда. На идее стационаризации основано большинство методов анализа временных рядов.

3.2.1 Разложение на компоненты

Трендово-циклической компонентной называют не наиболее общее направление временного ряда. Будем воспринимать это как элемент иерархии, находящийся на ступень ниже. Обычно цикличностью ряда называют его свойство повторять предыдущие значения. Это не совсем периодичность, поскольку эти повторения могут происходить с разной регулярностью.

Если убрать из данных трендово-циклическую компоненту, оставшуюся компоненту называют сезонной.

Autoformer

Введем операцию разложения на две компоненты, трендовую и сезонную:

$$x_{\text{trend}} = \text{MovAvg}(\text{Padding}(x)),$$
$$x_{\text{season}} = x - x_{\text{trend}}.$$

Напомним, что операция moving average заключается во взятии бегущего среднего k последовательных замеров. Нулевой падинг добавляется с краев, чтобы длина последовательности не менялась (стандартный прием).

Введенную операцию разложения можно применить внутри трансформера архитектуры энкодер-декодер. На вход кодировщику подается ненормированный ряд, внутри каждого трансформерного блока выделяется трендовая часть и отбрасывается. Это сделано, чтобы обрабатывать только сезонную часть. То есть с проходом через блоки кодировщика модель выделяет все более мелкомасштабные изменения.

Это отражает то, что основную информацию несут локальные зависимости, отделенные от общего тренда. Если энкодер обрабатывает сезонную компоненту прошлого, то свою очередь декодер будет предсказывать сезонную компоненту для будущего.

В autoformer [12] есть сразу два набора плейсхолдеров: для сезонной и трендовой компоненты. По мере прохождения через блоки кодировщика, эти компоненты как бы уточняются и наполняются информацией. Это отражает важность нестационарной информации для предсказания.

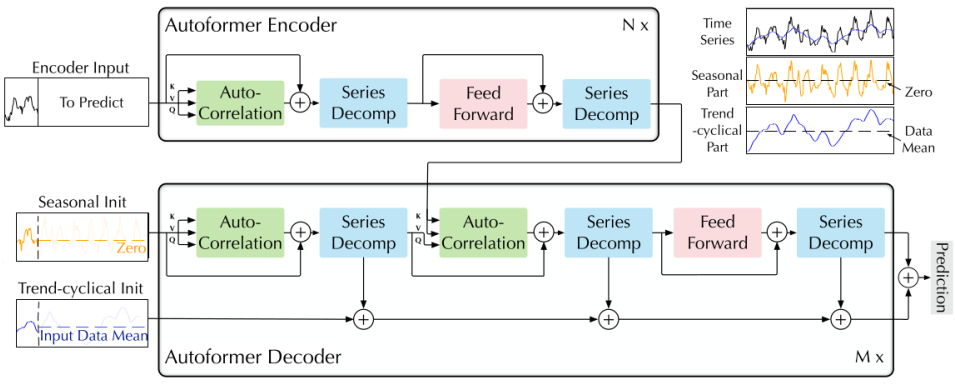

Рис. 14. Как трендовая и сезонная компоненты используются в Autoformer

AutoCorrelation

Дополнительно в autoformer учитываются периоды, но более хитрым способом, нежели это было в TimesNet. Для случайного процесса можно определить функционал автокорреляции:

$$\mathcal{R}_{xx}(\tau) = \lim_{L\to\infty} \frac{1}{L}\sum_{t=1}^{L} x_t x_{t-\tau}.$$

Этот функционал характеризует схожесть процесса $\{x_t\}$ с процессом $\{x_{t-\tau}\}$. Это может служить мерой уверенности, что процесс периодичен с длиной периода τ. Для временных рядов (как реализаций случайного процесса) этот функционал тоже можно определить. Посчитать его можно с помощью FFT.

Если до этого периоды были головами в ансамбле, то теперь периоды служат токенами в механизме внимания:

$$\tau_1, \ldots, \tau_k = \arg\operatorname{topk}\{\mathcal{R}_{Q,K}(\tau)\} \tag{3.1}$$

$$\alpha_1, \ldots, \alpha_k = \operatorname{Softmax}(\mathcal{R}_{Q,K}(\tau_1), \ldots, \mathcal{R}_{Q,K}(\tau_k)) \tag{3.2}$$

$$\operatorname{AutoCorrelation}(Q, K, V) = \sum_{i=1}^{k} \alpha_i \operatorname{Roll}(V, \tau_i). \tag{3.3}$$

Такая операция напрямую вытягивает похожие моменты из прошлого из формирует предсказание как взвешенное среднее между этими моментами. Операция $\operatorname{Roll}(X, \tau)$ совершает циклический сдвиг во временном ряде так, чтобы сопоставить точки находящиеся в одной фазе относительно периода τ.

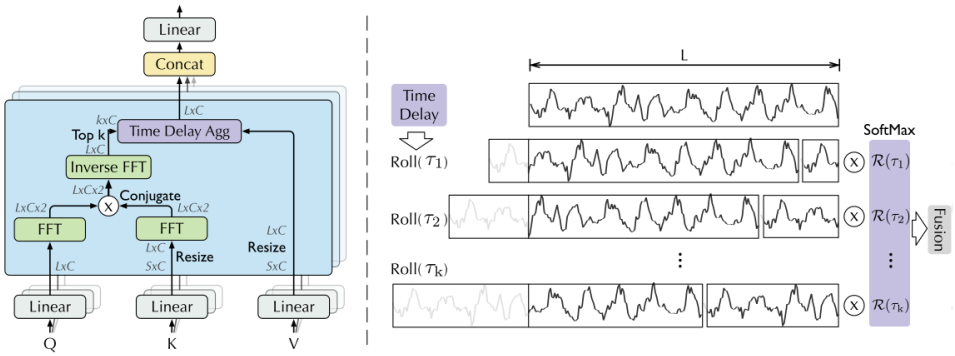

Рис. 15. Механизм автокорреляции в Autoformer

3.2.2 Нормализация. RevIN

Архитектура autoformer достаточно тяжеловесна для восприятия. Существует метод, который тоже борется с проблемой анализа нестационарных рядов, однако делает это куда более простым и изящным способом.

Известным примером является метод RevIN для прогнозирования [13]. Он заключается в следующем: что если нормализовать данные перед подачей в нейросеть, а на выходе произвести обратное преобразование и вернуть среднее и дисперсию. На входе используется instance нормализация с обучаемыми параметрами γ, δ:

$$\mu = \frac{1}{T}\sum_{t=1}^{T} x_t, \quad \sigma^2 = \frac{1}{T}\sum_{t=1}^{T}(x_t - \mu)^2, \quad \widehat{x}_t = \gamma \circ \frac{x_t - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \delta.$$

Ряд $\widehat{x} \in \mathbb{R}^{T \times C}$ подается на вход нейросети, которая выдает предсказания $\widetilde{y} \in \mathbb{R}^{N \times C}$ для будущих N замеров. К ним применяется обратное преобразование:

$$y_t = \sqrt{\sigma^2 + \varepsilon} \circ \frac{\tilde{y}_t - \delta}{\gamma} + \mu.$$

Название метода расшифровывается как reversible instance normalization. Во-первых, это не совсем стационаризация, скорее просто нормализация. Но нормализация для нейросетей играет большую роль. Во-вторых, это не нормализация всей обучающей выборки по каждому признаку, а это нормализация только внутри одного временного ряда. То есть instance нормализация может преобразовать два ряда с совершенно разными средними и дисперсиями в одинаковый. И в этом состоит предположение о том, что намного важнее предсказывать какие-то локальные динамики, чем учитывать, что «этот ряд в диапазоне [5,10], а этот ряд в диапазоне [100,300]». Если между этими рядами есть что-то общее, то нейросеть не увидит этого из-за огромной разницы в масштабах.

3.2.3 Нормализация. Non-Stationary Transformer

На самом деле информация о среднем и дисперсии тоже ценна. Она может сообщать о природе ряда. Идея Non-Stationary Transformer (будем называть NST [9]) в том, чтобы учесть эту информацию не только на входе и выходе модели, но и внутри.

Итак, NST вооружен обратимой нормализацией, как и RevIN. Правда авторы NST обошлись без обучаемых параметров:

$$\widehat{x}_t = \frac{x_t - \mu}{\sqrt{\sigma^2 + \varepsilon}}, \quad y_t = \sqrt{\sigma^2 + \varepsilon} \circ \tilde{y}_t + \mu.$$

Нормализованный ряд $\widehat{x} \in \mathbb{R}^{T \times C}$ не содержит информации о среднем и дисперсии. Он подается на вход сначала полносвязным слоям, играющим роль эмбедера, а затем скармливается трансформеру. Как использовать информацию об изначальном ненормализованном ряде внутри трансформера? Самое важное место в трансформере — это блоки с аттеншеном. Попытаемся проанализировать, как отличается аттеншен с пре-нормализацией и без ней. Чтобы упростить анализ, сделаем предположение, что в нейросети нет функций активаций и прочих нелинейностей, кроме операции аттеншена. Результат анализа, полученный при таком предположении мы потом попытаемся применить к общему нелинейному случаю.

Линейный случай

Механизм внимания:

$$\mathrm{Attn}(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V.$$

Представления Q, K, V являются результатами линейного преобразования входа x (согласно нашему предположению), т.е. $Q = f_Q(x)$, $K = f_K(x)$, $V = f_V(x)$. Обозначим $Q' = f_Q(\widehat{x})$, $K' = f_K(\widehat{x})$, $V' = f_V(\widehat{x})$.

Как отличаются Q' от Q?

$$Q' = f_Q\left(\frac{x - \mathbf{1}\mu^T}{\sigma}\right) = \frac{f_Q(x) - \mathbf{1}f_Q(\mu)^T}{\sigma} = \frac{Q - \mathbf{1}\mu_Q^T}{\sigma}, \quad \mu_Q = \frac{1}{T}\sum_{t=1}^{T} q_t.$$

По аналогии будет

$$K' = \frac{K - \mathbf{1}\mu_K^T}{\sigma}.$$

Как отличаются $Q'K'^T$ от QK^T?

$$Q'K'^T = \frac{1}{\sigma^2}(QK^T - \mathbf{1}(\mu_Q^T K^T) - (Q\mu_K)\mathbf{1}^T + \mathbf{1}(\mu_Q^T \mu_K)\mathbf{1}^T).$$

Как отличается $\text{Attn}(Q', K', V')$ от $\text{Attn}(Q, K, V)$? Выражение QK^T входит в аттеншен под софтмаксом. Софтмакс не меняется, если ко всем компонентам прибавить одно и то же число:

$$\begin{aligned}\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) &= \text{Softmax}\left(\frac{\sigma^2 Q'K'^T + \mathbf{1}(\mu_Q^T K^T) + (Q\mu_K)\mathbf{1}^T - \mathbf{1}(\mu_Q^T \mu_K)\mathbf{1}^T}{\sqrt{d_k}}\right) \\ &= \text{Softmax}\left(\frac{\sigma^2 Q'K'^T + \mathbf{1}(\mu_Q^T K^T)}{\sqrt{d_k}}\right)\end{aligned}$$

Получили, что подавая на вход трансформеру (полностью линейному) нормализованный временной ряд, мы имеем возможность прямо внутри трансформера посчитать аттеншен для ненормализованного временного ряда. Тем самым мы учтем среднее и дисперсию исходных данных.

Проанализируем полученное выражение. Что именно привносит информацию о распределении? Число σ^2 и $K^T\mu_Q$ — это статистики, посчитанные на исходном временном ряде.

Общий случай

Но в общем случае мы не можем использовать полностью линейный трансформер. А значит использовать σ^2 становится неосмысленно, а доступа к $K\mu_Q$ у нас по просту нет.

Основная идея de-stationary attention состоит в том, чтобы аппроксимировать $\tau \approx \sigma^2$ и $\Delta \approx K\mu_Q$ как некоторые функции от исходного ряда:

$$\log\tau = \text{MLP}(\sigma, x), \quad \Delta = \text{MLP}(\mu, x), \text{Attn}(Q', K', V') = \text{Softmax}\left(\frac{\tau Q'K'^T + \mathbf{1}\Delta^T}{\sqrt{d_k}}\right)V'.$$

Список литературы

[1] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. TSMixer: An All-MLP Architecture for Time Series Forecasting. 2023. https://arxiv.org/abs/2303.06053.

[2] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting? 2022. https://arxiv.org/abs/2205.13504.

[3] Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal Convolutional Networks for Action Segmentation and Detection. 2016. https://arxiv.org/abs/1611.05267.

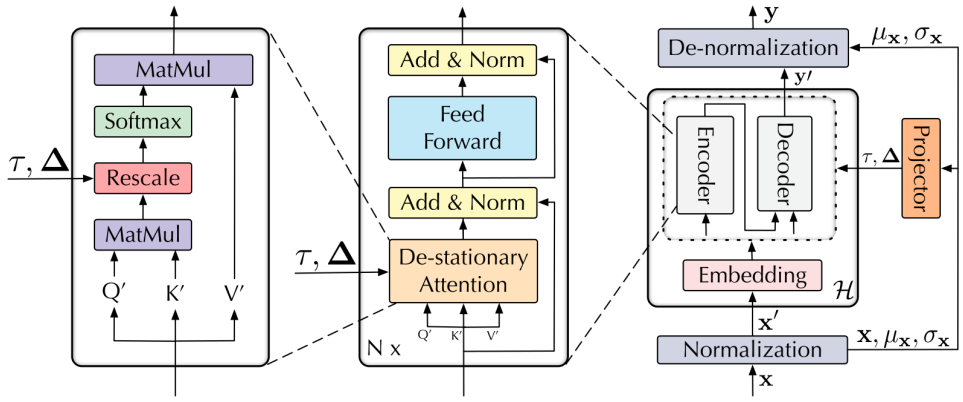

Рис. 16. Non-Stationary Transformer

[4] David Salinas, Valentin Flunkert, and Jan Gasthaus. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. 2019. https://arxiv.org/abs/1704.04110.

[5] Ismail Fawaz, Hassan, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, sep 2020. ISSN 1573-756X. http://dx.doi.org/10.1007/s10618-020-00710-y.

[6] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. 2016. https://arxiv.org/abs/1609.03499.

[7] Luo donghao and wang xue. ModernTCN: A Modern Pure Convolution Structure for General Time Series Analysis. In *The Twelfth International Conference on Learning Representations*, 2024. https://openreview.net/forum?id=vpJMJerXHU.

[8] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. 2021. https://arxiv.org/abs/2012.07436.

[9] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. 2023. https://arxiv.org/abs/2205.14415.

[10] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396, 2022. URL https://arxiv.org/abs/2111.00396.

[11] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186, 2023. URL https://arxiv.org/abs/2210.02186.

[12] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. arXiv preprint arXiv:2106.13008, 2022. URL https://arxiv.org/abs/2106.13008.

[13] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.