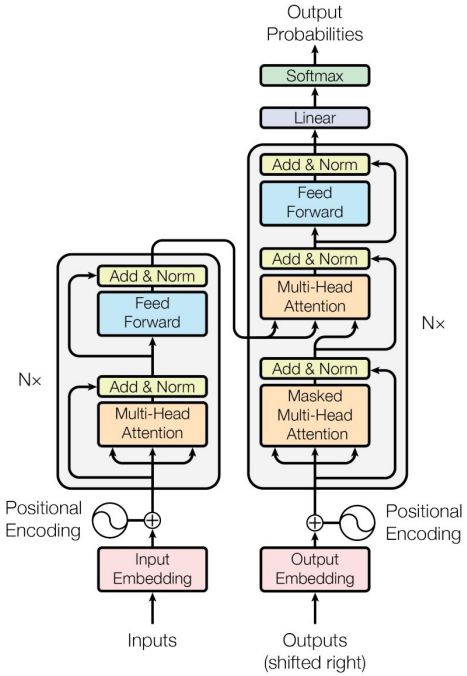
Анализ временных рядов

Азиз Темирханов

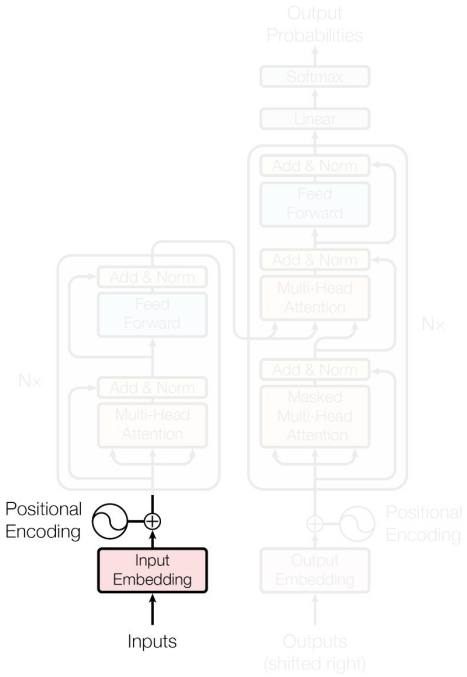
План лекции

- Вспомним механизм внимания
- Узнаем про архитектуры с трансформером
- Поговорим о фунд моделях

Transformer



Transformer



Входной текст может быть символами, словами, «токенами»:

"The detective investigated" -> [The_] [detective_] [invest] [igat] [ed_]

Токены — это индексы в «словаре»:
[The_] [detective_] [invest] [igat] [ed_] -> [3 721 68 1337 42]

Каждая запись словаря соответствует выученному вектору размерности d_model.

[3 721 68 1337 42] -> [[0.123, -5.234, …], […], […], […], […]]

Позиционный экнодинг

Помните: attention инвариантен к перестановкам, а язык — нет!

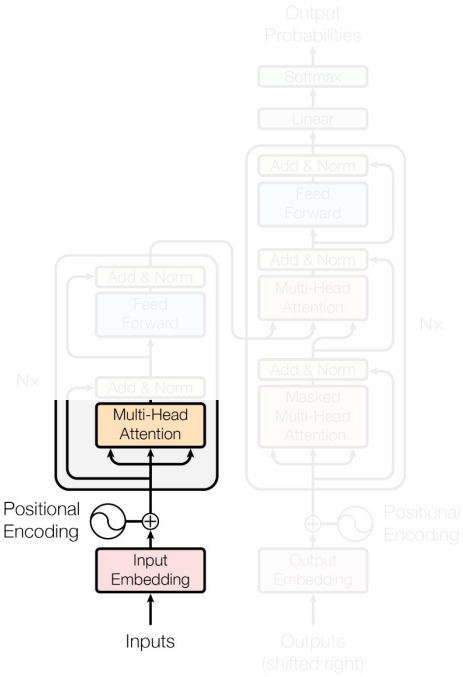
(“The mouse ate the cat” vs “The cat ate the mouse”)

Нужно закодировать позицию каждого слова; просто добавьте что-нибудь.

Думайте так: [The_] + 10 [detective_] + 20 [invest] + 30 … но умнее.

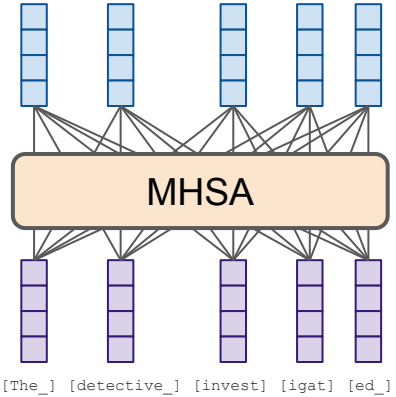
Thanks to Basil Mustafa for slide inspiration (and Lucas Beyer https://www.youtube.com/watch?v=Eixl6t5oif0)

Transformer

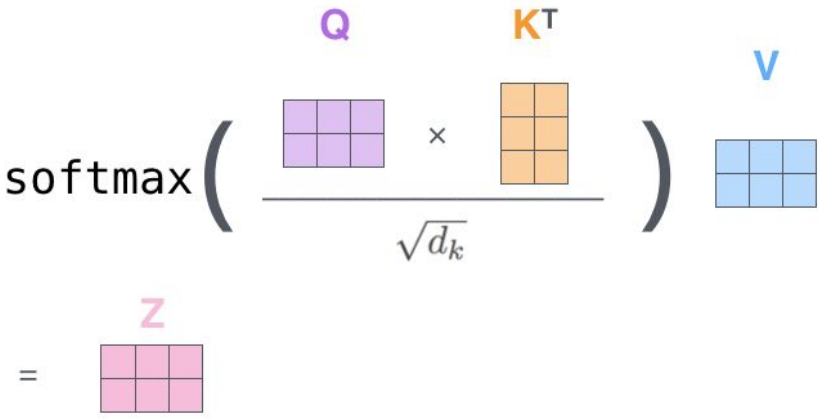


То есть входная последовательность используется для построения **queries**, **keys** и **values**!

Каждый токен может «оглядываться» на весь вход и решать, как обновить своё представление, исходя из того, что он видит.

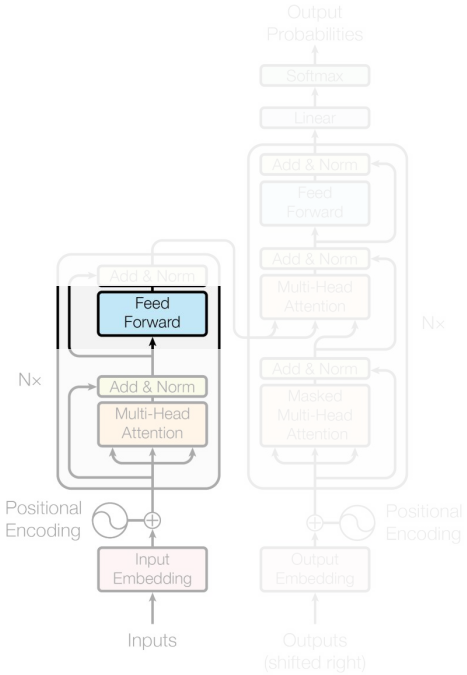


Attention



http://jalammar.github.io/illustrated-transformer

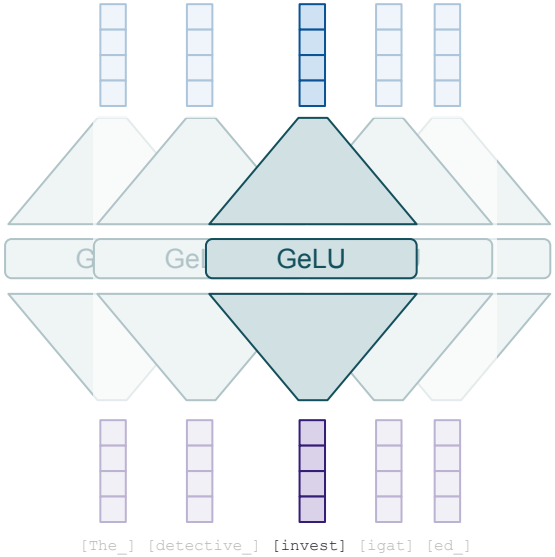
MLP



Простой MLP применяется к каждому токену:

$$z_i = W_2\,\mathrm{GeLU}(W_1 x + b_1) + b_2$$

Представьте, что каждый токен самостоятельно размышляет над тем, что он наблюдал ранее.
Есть также некоторые слабые доказательства того, что именно здесь хранится «знание о мире».
В нём содержится основная часть параметров.
Когда люди создают гигантские модели и разреженные/MOE-модели, именно это и становится гигантским.
Некоторые люди любят называть это 1x1 свертками



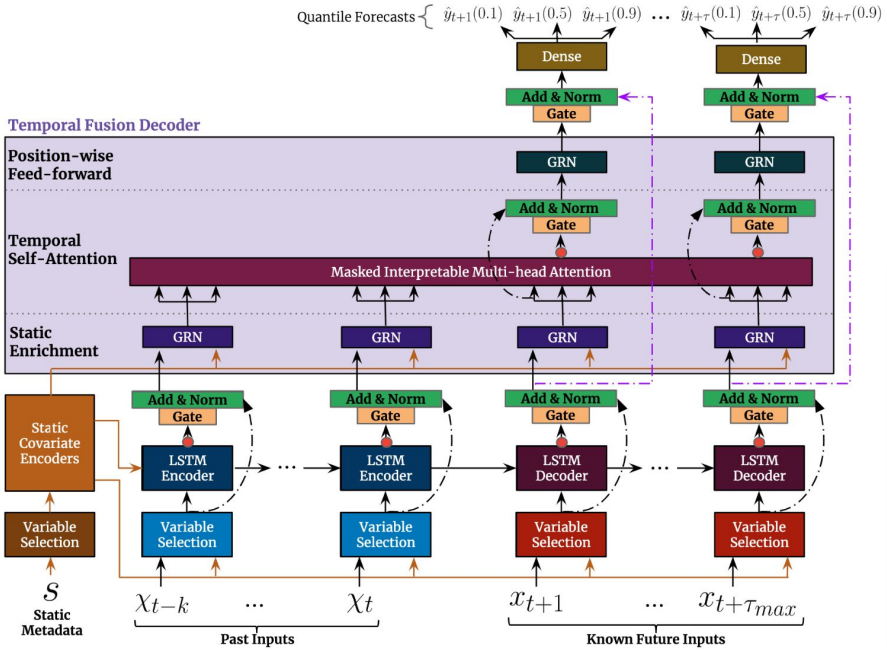
Transformer-модели

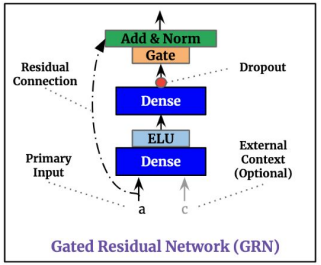
Temporal Fusion Transformer (TFT, 2019, 2,148 citations)

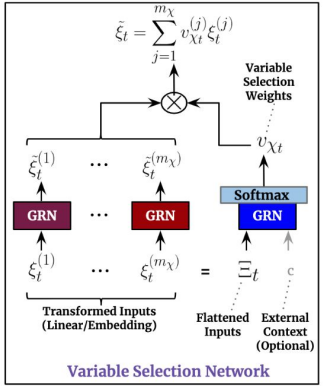
Основная идея: многогоризонтное прогнозирование со смешанными ковариатами (статическими, известными на будущее, наблюдаемыми из истории), объединяющее рекуррентные слои (локальные зависимости) + интерпретируемый self-attention + отбор признаков + гейтинг.

Результаты: сообщается о существенных улучшениях по сравнению с бенчмарками на нескольких реальных наборах данных, а также приводятся примеры интерпретируемости и сценарии ее использования.

Ограничения / область применимости: supervised модель





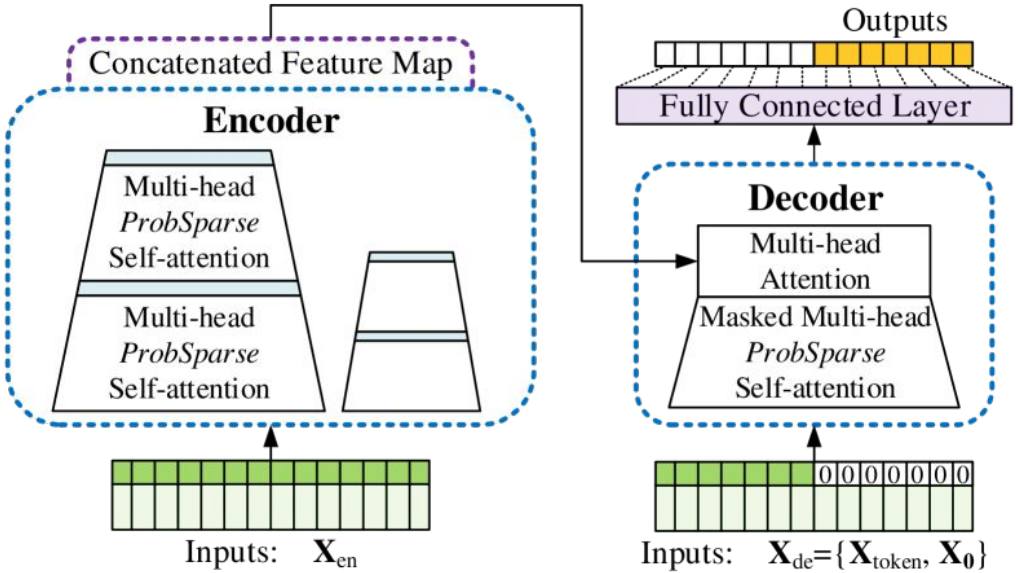


Informer (2020, 6368 citations)

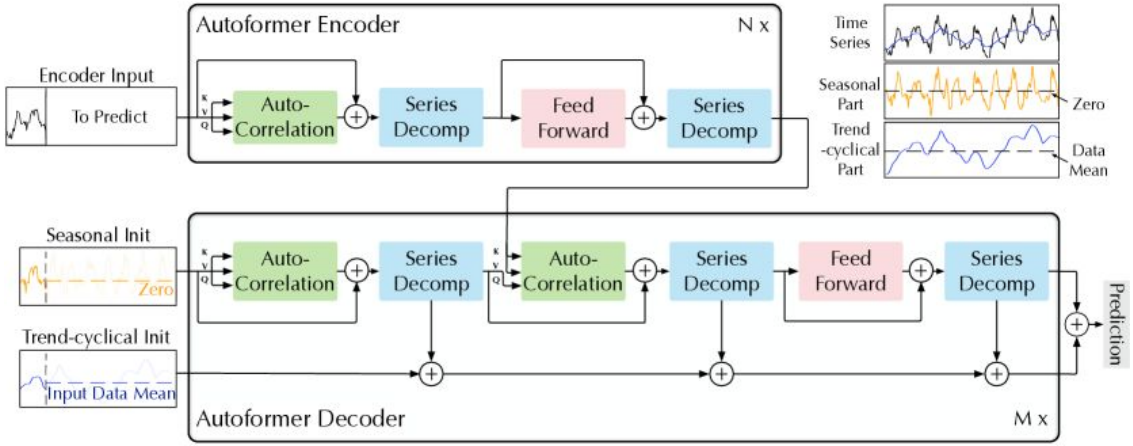
Основная идея: сделать трансформеры практичными для прогнозирования временных рядов по длинным последовательностям (LSTF) с помощью внимания ProbSparse (приблизительно O(LlogL), а также дистилляции внимания, чтобы справляться с чрезвычайно длинными входами.

Результаты: демонстрирует сильные результаты в LSTF при снижении затрат по времени и памяти по сравнению со стандартным вниманием.

Ограничения, отмеченные позже: критика в работе LTSF-Linear / DLinear утверждает, что self-attention, инвариантное к перестановкам, может терять информацию о временном порядке, ставя под сомнение многие варианты трансформеров, используемые в LSTF.



Autoformer (2021, 3726 citations)

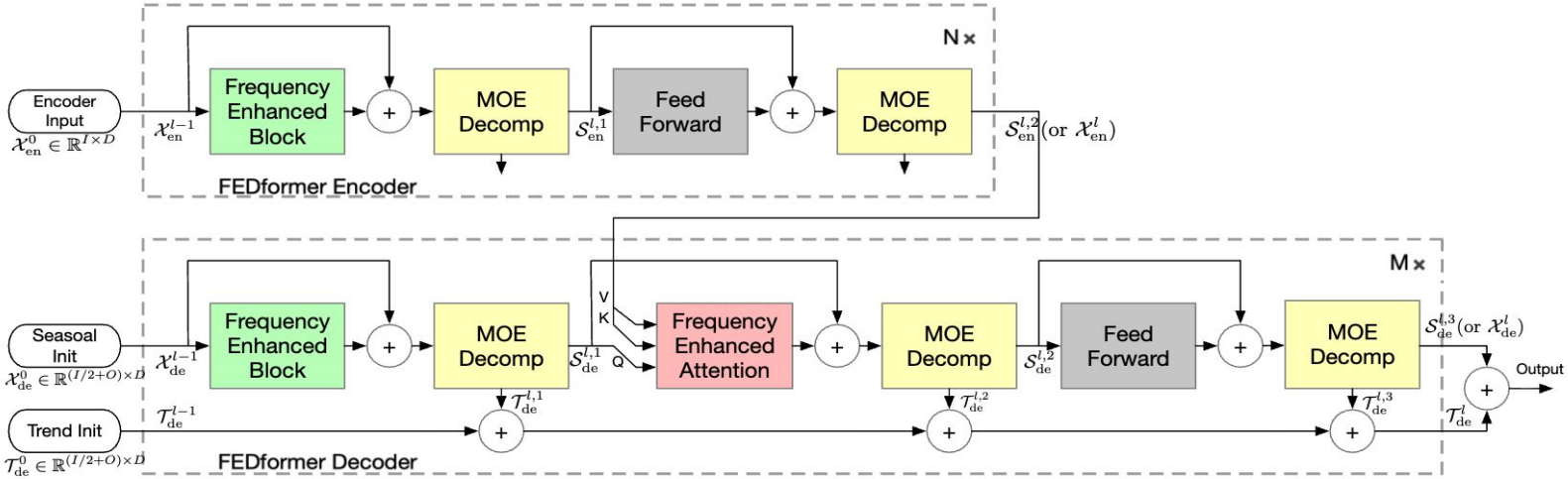


Основная идея: добавить явную декомпозицию сезонность–тренд и заменить self-attention механизмом авто-корреляции, который агрегирует зависимости на уровне подрядов (используя периодичность).

Результаты: заявляет SOTA в долгосрочном прогнозировании с ~38% относительным улучшением на шести бенчмарках.

Ограничения / предположения: ключевой механизм построен вокруг периодичности / структуры автокорреляции ряда — это сильное индуктивное смещение, которое может подходить не всем доменам одинаково хорошо.

FEDformer (2022, 2542 citations)



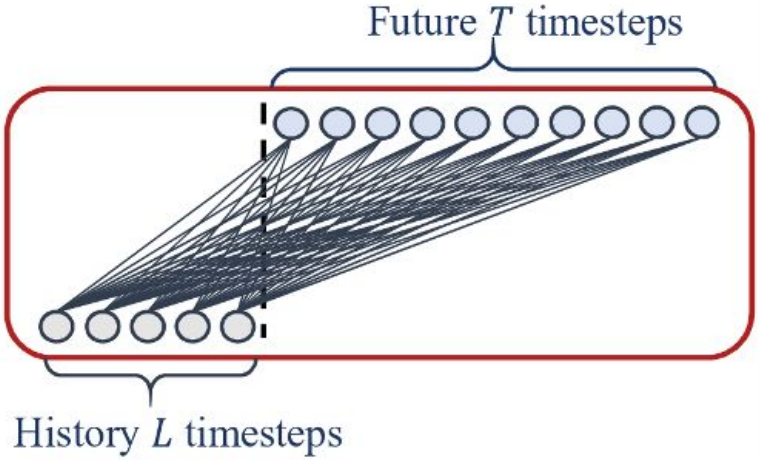
Основная идея: объединить декомпозицию сезонность–тренд с «частотно-усиленным» трансформером, исходя из того, что многие ряды имеют разреженное представление в частотной области (Фурье / вейвлеты), и тем самым добиться линейной сложности по длине последовательности.

Результаты: сообщает о снижении ошибки примерно на ~14.8% (многомерные ряды) и ~22.6% (одномерные ряды) по сравнению с предыдущим SOTA на шести бенчмарках.

Ограничения: явно опирается на предпосылку «разреженности в частотном базисе» — качество может зависеть от того, насколько это соответствует данным.

DLinear

DLinear (2023, 4319 citations)



Идея: использовать простые линейные модели

$$\hat{X}_i = W X_i \qquad W \in \mathbb{R}^{T \times L}$$

Предобработка:

- Декомпозиционный модуль из Autoformer
- Извлечение тренда скользящим средним

$$\mathcal{X}_{\mathrm{t}} = \mathrm{AvgPool}(\mathrm{Padding}(\mathcal{X}))$$
$$\mathcal{X}_{\mathrm{s}} = \mathcal{X} - \mathcal{X}_{\mathrm{t}},$$

отдельные модели

Код: https://github.com/vivva/DLinear
Статья: https://arxiv.org/abs/2205.13504

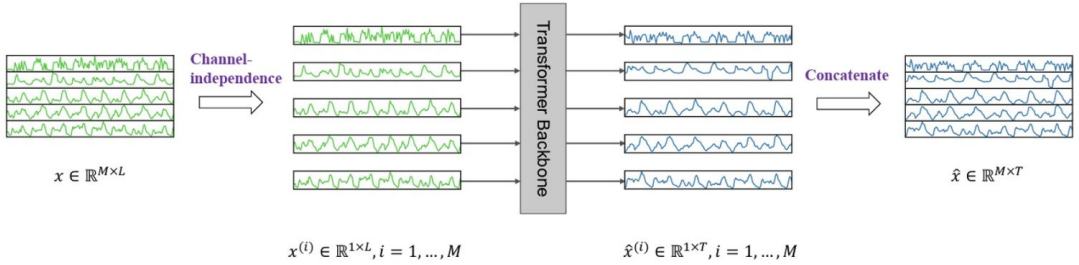
PatchTST (2022, 2673 citations)

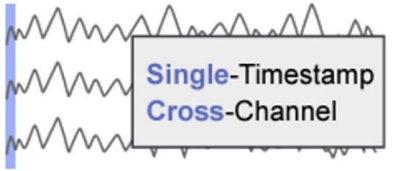
Идея: одно наблюдение в конкретный момент времени несет мало информации
(как один пиксель в картинке) - лучше рассматривать “патчи”

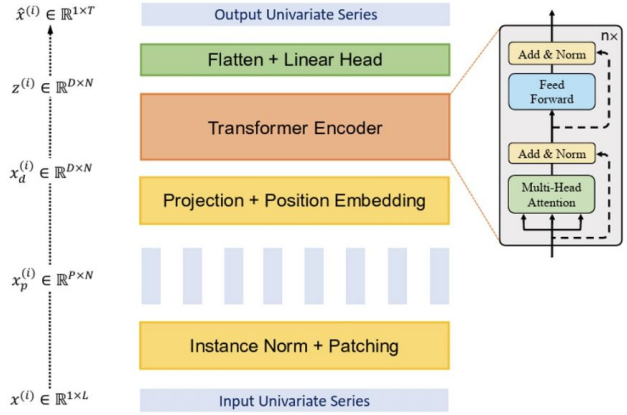
Нормализуем и патчим входную последовательность

Получаем э ↓ динги, которые отправляем в трансформерный encoder

Используем ↓ ten + линейную голову для получе предсказаний





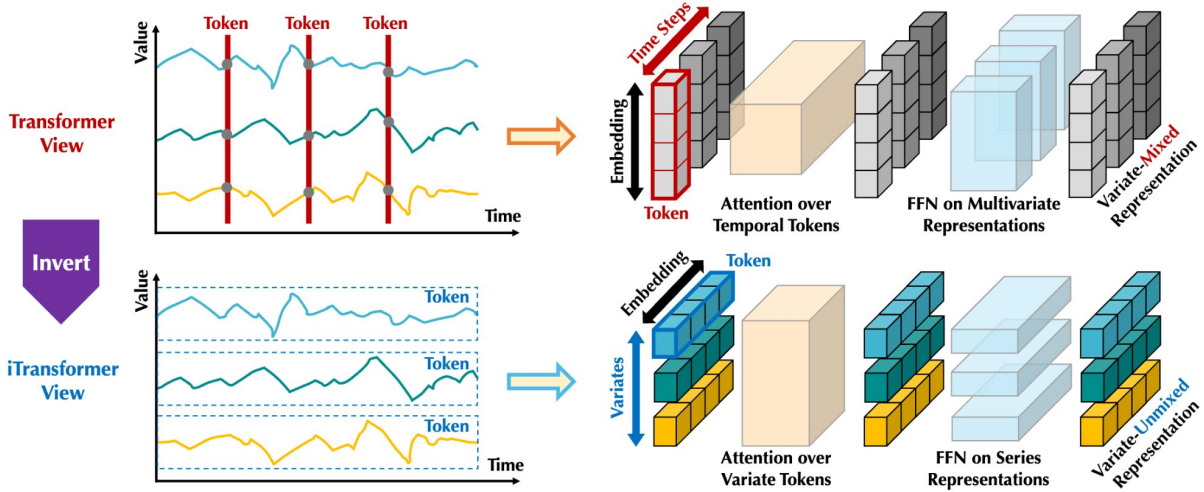


iTransformer (2023, 1241 citations)

Основная идея: инвертировать токенизацию: рассматривать переменные (каналы) как токены и применять внимание *между* каналами, решая проблемы подходов, где временные токены/эмбеддинги смешивают несколько каналов и плохо масштабируются при длинном окне истории.

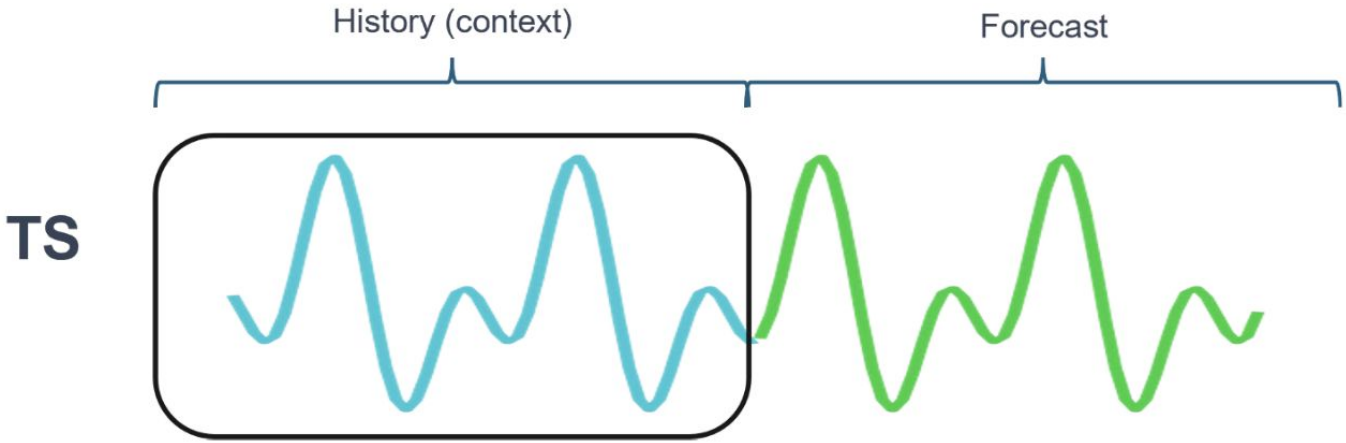
Результаты: заявляет SOTA на реальных датасетах, лучшую обобщаемость между каналами и более хорошую работу с произвольной длиной окна наблюдений.

Зачем: прямой ответ эпохе после DLinear: сохранить стандартные блоки Transformer, но изменить то, *над чем* работает внимание (по каналам, а не по времени).



Foundational Models

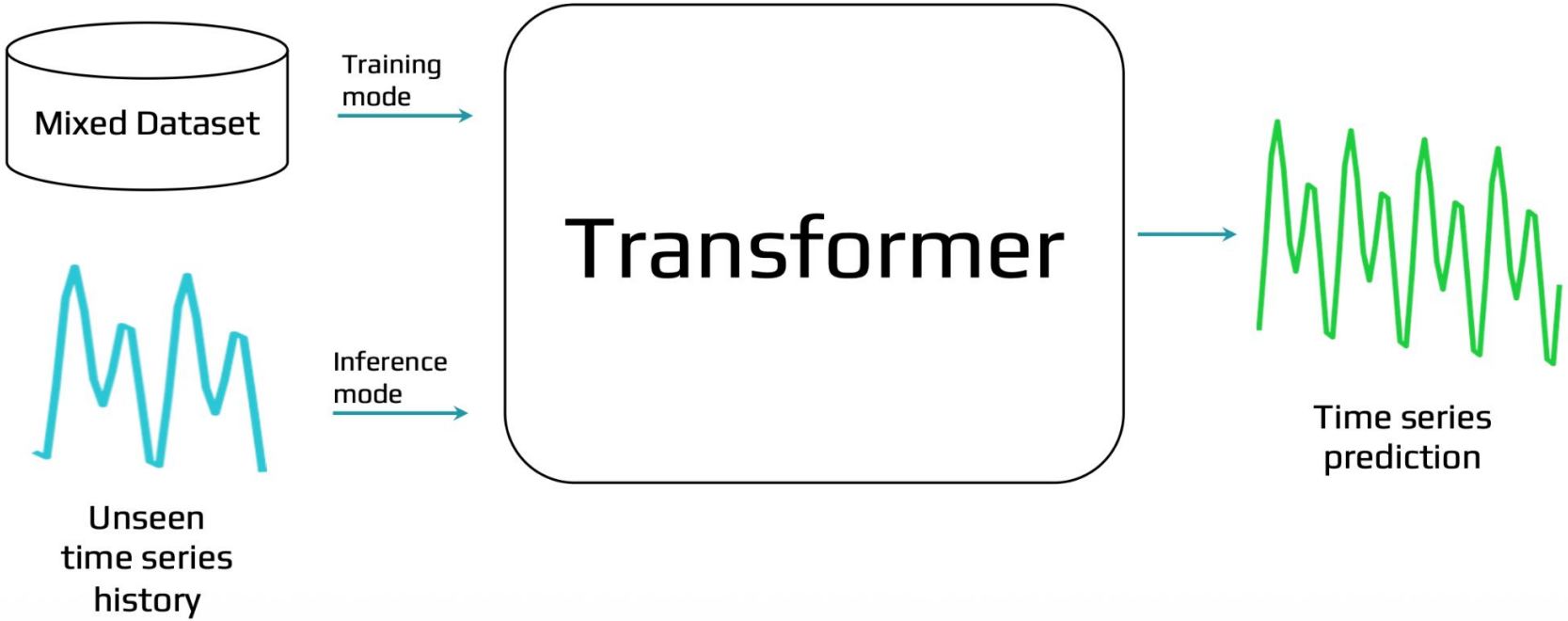
Что такое Zero-Shot?



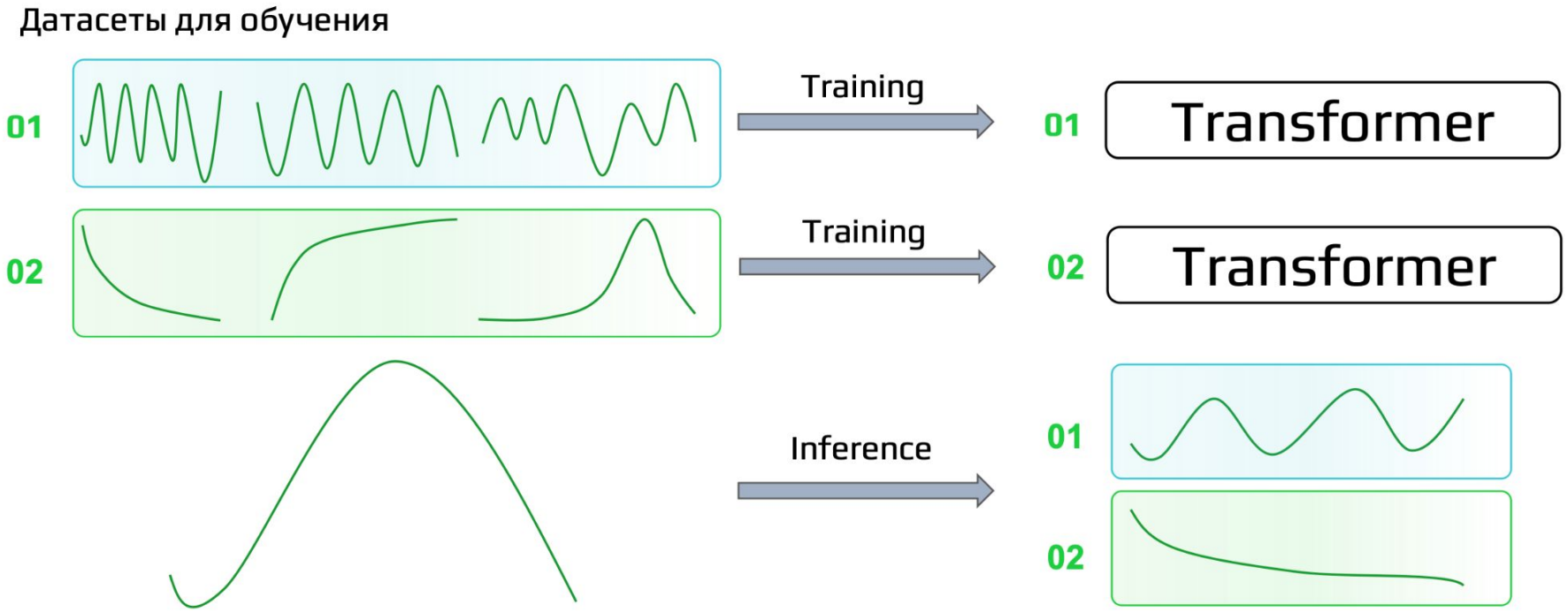
Zero-shot: Train the model to predict labels for new data **without training on the target dataset**, based on patterns identified in the unrelated data.

Что такое Zero-Shot?

Zero-shot модели для временных рядов – это в основном трансформеры.
Также они требуют большой разнообразный датасет для предобучения.



Что такое Zero-Shot?



LLMTime (2023, 886 citations)

Идея и Архитектура: кодируем последовательность как текст и используем любую из доступных LLM.

Как кодировать временной ряд?

Rescaling

$$x_t \to \frac{x_t - b}{a} \begin{cases} b = \min_t x_t - \beta \cdot (\max_t x_t - \min_t x_t) \\ a = a\text{-percentile}(x_1 - b, x_2 - b, \ldots, x_T - b) \end{cases}$$

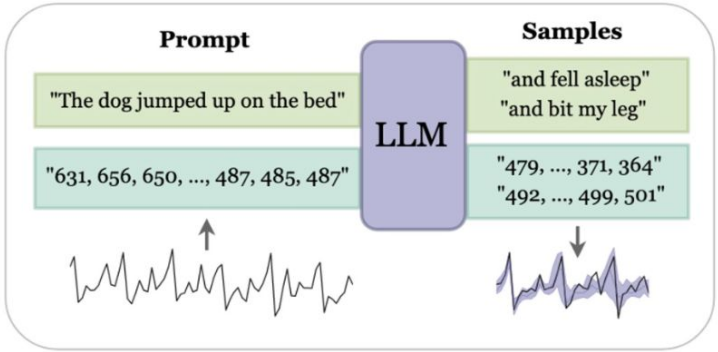
Type-changing 0.123, 1.23, 12.3, 123.0 → "12,123,1230,12300"

Tokenization

"151,167,...,267"

"151,167,...,267"

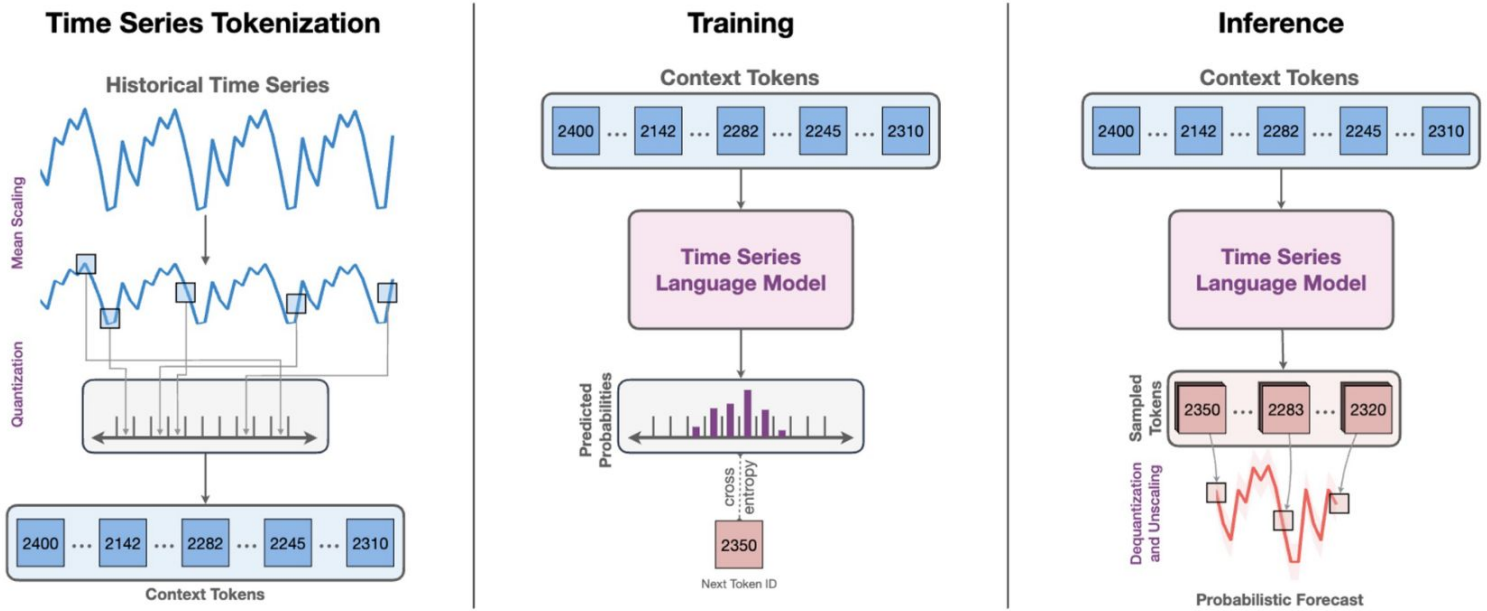
alpha=0.95, beta=0.3



Упрощенно, задача прогноза — это та же задача продолжения последовательности

CHRONOS (2024, 797)

Идея и Архитектура: кодируем последовательность как текст и используем любую из доступных LLM (T5). Только теперь с дообучением на синтетических и реальных данных с применением TSMixup аугментации (~84 млрд).

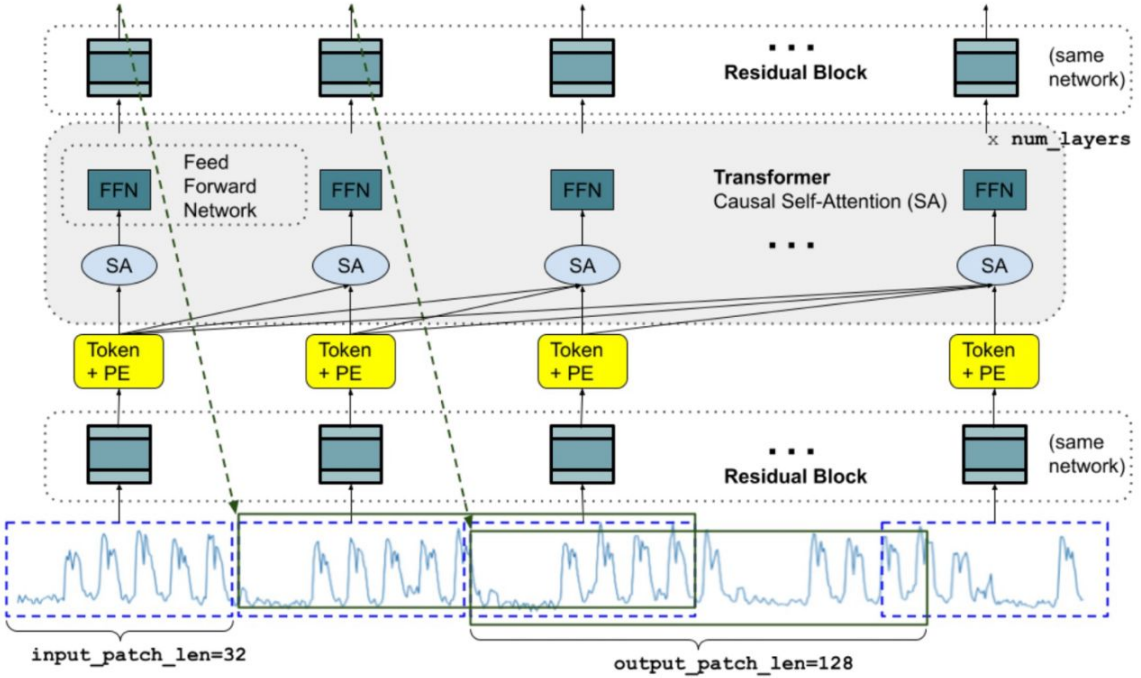


TimesFM (2024, 709)

Идея: обучаем трансформерный decoder на большом количестве синтетических и реальных рядов (~370 млрд).

Задачи: forecasting (+ quantile)

Постановка: univariate
(+ exogenous features)

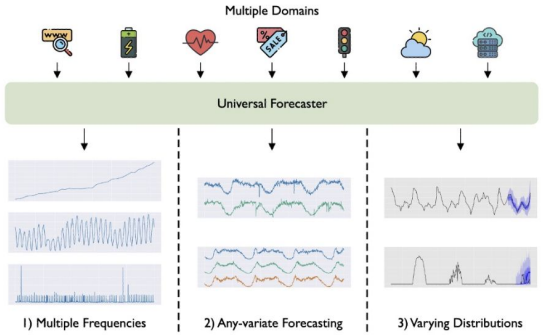


Moirai (2024, 467)

Идея: маскированная encoder-only модель, предобученная на рядах с различной грануляцией (~27 млрд наблюдений).

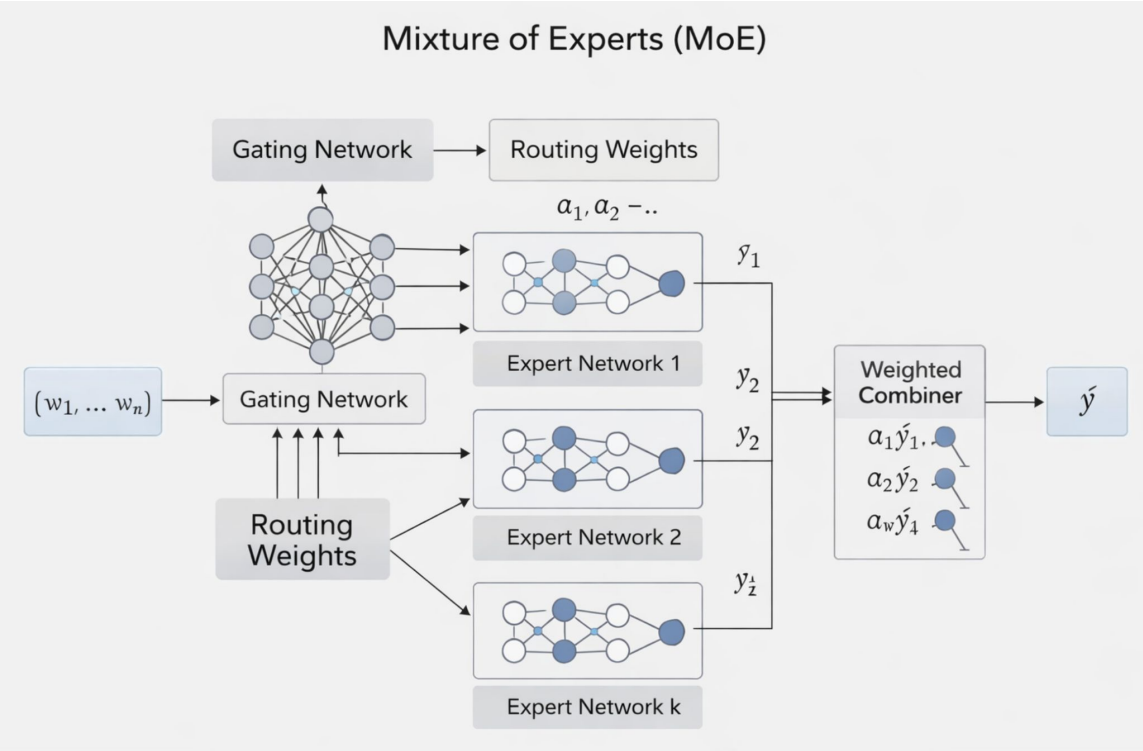
Задача: probabilistic forecasting

Архетектура: RoPE эмбеддинг, flatten для multivariate данных, благодаря чему может обрабатывать данные с произвольной размерностью.



9 доменов. Разная гранулярность — от года до секунды

MoE


Mixture of Experts (MoE)
Gating Network
Routing Weights
$\alpha_1, \alpha_2 -..$
y_1
Expert Network 1
y_2
y_2
Expert Network 2
y_z
Expert Network k
$(w_1, \ldots w_n)$
Gating Network
Routing Weights
Weighted Combiner
$\alpha_1 \hat{y}_1 .$
$\alpha_2 \hat{y}_2$
$\alpha_w \hat{y}_1$
$\hat{y}$

Moirai-MOE

Идея: частота — плохой прокси типа паттерна; вместо этого нужна **автоматическая специализация** внутри модели на уровне токенов.
Метод: единая проекция I/O + **sparse Mixture-of-Experts** внутри Transformer, чтобы разные эксперты ловили разные режимы/паттерны (token-level specialization).
Результат: в экспериментах на десятках датасетов — улучшения над Moirai и конкурентность/превосходство над другими TSFM при меньшем числе активных параметров.
Проблемы: MoE сложнее обучать/дебажить (балансировка экспертов, маршрутизация); возможны “мертвые” эксперты и нестабильность при сдвиге домена.