Добрый день, уважаемые коллеги! Меня зовут Игорь, тема моей работы **Прогнозирование временных рядов с помощью рекуррентных нейронных сетей.** Научный руководитель – Санжар Досов.

В начале поговорим об актуальности этой темы.

Если оглянуться по сторонам, то легко заметить, что временные ряды окружают нас буквально повсюду. Людям необходимо делать прогнозы, чтобы предугадывать тренды, продажи (чего-угодно), потребление ресурсов и многое другое, можно долго продолжать этот список. При этом где-то требуется высокое качество прогнозирования, а где-то нужно минимизировать время обработки запросов.

При этом, в отличие, например, от области обработки естественного языка, где лидируют трансформерные алгоритмы, в прогнозировании временных рядов пока нет универсального решения, а значит есть потенциал для развития.

Тогда назревает вопрос, почему именно рекуррентные сети? Во-первых, рекуррентные сети по своей природе хорошо подходят для временных рядов, и позволяют находить сложные зависимости в данных, в отличие от классических методов. Во-вторых, RNN – это намного более простая архитектура, чем Transformer, а значит обучаются и работают такие модели намного быстрее. То есть, с позиции нашей работы, RNN – это некий компромисс.

Задачи, которые мы перед собой ставили, представлены на слайде.

Если вкратце, то сначала нужно было провести обзор литературы. Выбрать разные, при этом популярные, алгоритмы прогнозирования временных рядов.

Затем выбрать датасеты и метрики, которые мы будем использовать для сравнения, а также зафиксировать все условия экспериментов.

После этого реализовать несколько рекуррентных нейронных сетей, попробовать их модифицировать, чтобы улучшить качество.

И в конце провести общее сравнение, собственных реализаций и моделей из статей, и сделать выводы.

В качестве данных был выбран единственный датасет под названием M4.

Это датасет с крупного соревнования Makridakis Competition. Он чаще других фигурирует в научной литературе и популярен у исследователей.

Состоит из 100 тысяч одномерных временных рядов, собранных в разное время, из разных сфер деятельности человека и разной частоты. На слайде есть таблица с количеством датасетов каждого вида.

Соревнованием также даны несколько простейших baseline-решений (под названием naïve), формулы которых есть на слайде. В конце мы можем к ним вернуться.

В качестве основной метрики качества была выбрана Overall Weighted Average, которая предложена и используется в M4. Она состоит из усредненной и нормированной суммы двух других метрик. Их формулы есть на слайде.

Помимо этого, измерялось среднее время тренировки моделей и среднее время прогнозирования.

В качестве классических подходов мы взяли сразу 4 модели: это ARMA (авторегрессия и скользящее среднее).

Затем ARIMA – та же ARMA, но с возможностью использования для нестационарных временных рядов.

SARIMA – модель, учитывающая также сезонность в данных.

И TBATS – улучшенная версия, которая сама подбирает параметры, позволяет учесть несколько сезонностей и предобрабатывает данные.

В качестве нейросетевых методов (без использования attention) мы отобрали 2 модели.

N-BEATS – созданная авторами с нуля глубокая архитектура с остаточными связями. Каждый блок состоит из линейного слоя с разделением на предсказывающую и резервную части. Блоки объединяются в длинные структуры, формирующие в конце общее решение.

LTSF-Linear – напротив, простая линейная модель, обернутая в (или даже спрятанная) нейросеть. С отсутствием функций активации. По заявлениям авторов, модель показывает хорошие результаты, но на самом деле это оказалось не так, об этом дальше.

Мы не могли обойти стороной трансформеры. Выбрали 2 модели.

Начнём с iTransformer. Тут авторы решили попробовать инвертировать оси в данных и брать за данные не сами измерения, а каждый отдельный признак. В остальном использована архитектура Transformer. Тут нужно упомянуть, что данная модель рассчитана под многомерный временные ряды, а в нашем эксперименте ряды одномерные, поэтому смысл такой модели тут уменьшается, но мы решили попробовать.

Вторая модель – PI-Transformer. Это также трансформер, но с некоторыми модификациями, которые перечислены. Главной модификацией является Persistence Initialization – это добавление остаточной связи от входного слоя к выходному и специальная инициализация весов. Это позволяет модели предсказывать предыдущее значение ряда ещё до начала обучения как некоторый baseline.

Теперь собственные реализации.

Слева на слайде блок-схема использованной модели. В ней параметры подбирались экспериментально, а RNN-блок (второй сверху) является одним из правых.

Справа 3 типа архитектур, которые мы попробовали. Это обычная RNN, LSTM и более продвинутая GRU. Думаю, что нет смысла на них останавливаться, это известные модели.

Также мы попробовали модификации.

Во-первых, 2 способа нормализации данных, чтобы не подавать сети большие значния.

Затем пробовали заменять функции активации на гиперболический тангенс (могу в конце объяснить, для чего).

И также попробовали метод Persistence Initialization из PI-Transformer, добавив в модель остаточную связь как на схеме справа.

Перейдём, наконец, к результатам. На всех таблицах далее зелёным фоном выделены собственные реализации, серым – baseline алгоритм naïve2 (из соревнования). Чем выше строчка – тем лучше результат.

Левая таблица – результаты по доменам. Правая – по периодам сбора данных. Порядок строчек в них одинаковый.

На этом слайде сравнивается метрика качества прогнозирования (OWA). Видим, что лучшие версии алгоритмов LSTM и GRU занимают верхние строчки рейтинга, и немного остают от N-BEATS и PI-Transformer.

LTSF-Linear и iTransformer в свою очередь показали неудовлетворительные результаты, недалёкие от бейзлайна.

Классические модели неплохи, но нам удалось их победить.

Теперь те же таблицы, но со сравнением по времени тренировки.

Тут всё более наглядно. Оба трансформера, а также N-BEATS имеют существенные отставания от наших реализаций, которые в свою очередь ещё сильнее отстают от классических методов.

При сравнении среднего времени предсказания порядок моделей тот же.

По итогу мы можем сделать вывод, что реализованные нами модели показали весьма неплохие результаты. С одной стороны их качество лучше, чем качество классических моделей. А с другой, наши реализации работают намного быстрее тяжеловесных архитектур. Таким образом, мы получили некоторый компромисс между качеством и скоростью, что может быть полезно в некоторых ситуациях.

На этом у меня всё, спасибо!  
Готов ответить на ваши вопросы.