Применение Recurrent Sigmoid Piecewise нейрона для прогнозирования временных рядов

Введение

Постановка задачи

Имеется временной ряд $x_1, ..., x_N$, сгенерированный некоторым вероятностным процессом с неизвестными совместными распределениями:

$$p(x_{t+k}, x_t, x_{t-1}, ..., x_{t-n}).$$

Необходимо использовать сгенерированный временной ряд для нахождения прогнозирующей модели вида:

$$\hat{x}_{t+k} = f^*(x_t, ..., x_{t-n}),$$

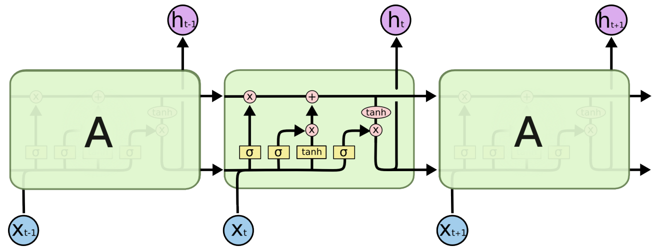
которая минимизирует математическое ожидание ошибки:

$$f^* = argmin_f \{E_{p(x_{t+k}, x_t, x_{t-1}, ..., x_{t-n})}[L(f(x_t, ..., x_{t-n}), x_{t+k})]\},$$

где $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ - функция ошибки, чаще всего используется либо квадратическая ошибка $L(x, y) = (x - y)^2$ либо абсолютная ошибка $L(x, y) = |x - y|$.
ДОБАВИТЬ: Обзор основных методов прогнозирования.

Recurrent Sigmoid Piecewise (RSP) нейрон

В задачах обработки естественного языка, таких как построение языковых моделей, автоматический перевод текста и пр. хорошо себя "зарекомендовали"рекуррентные нейронные сети на основе Long Short Term Memory (LSTM) и/или Gated Recurrent Unit (GRU) нейронов (**ДОБАВИТЬ: ссылки**). Данные нейроны имеют схожую структуру, которая позволяет уменьшить влияние проблемы затухающего/взрывающегося градиента при обучении рекуррентных моделей с использованием Backpropagation Through time (BPTT) алгоритма на длинных последовательностях. За счет этого, на практике, сети с этими нейронами более стабильны в обучении и имеют большую "точность"при работе на длинных последовательностях. LSTM-нейрон имеет следующую структуру:



Полное математическое описание классического LSTM нейрона:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

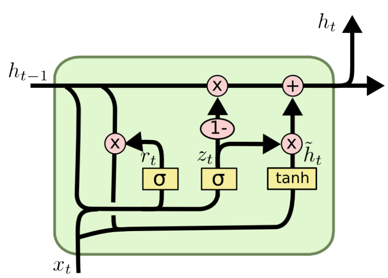
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * tanh(c_t)$$

GRU нейрон это, по сути, упрощенная версия LSTM нейрона:



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

В данном нейроне вектор выходов h_t так же "выполняет" роль вектора контекста, и используются следующие блоки:

- Блок обновления $z_t(x_t, h_{t-1}; W_z)$, рассчитывающий веса в диапазоне $(0, 1)$, которые применяются для рассчета нового вектора выходов (и, одновременно, контекста) h_t исходя из вектора-кандидата $\tilde{h}_t$ и предыдущего вектора h_{t-1}

- Блок "релевантности" $r_t(x_t, h_{t-1}; W_r)$, рассчитывающий веса в диапазоне $(0, 1)$, которые определяют "релевантность"/"важность" значений предыдущего выходного вектора h_{t-1} при рассчете вектора-кандидата для нового выходного вектора $\tilde{h}_t$

- Блок рассчета вектора-кандидата новых выходов $\tilde{h}_t(x_t, h_{t-1}, r_t; W)$
- Блок рассчета нового вектора выходов $h_t(h_{t-1}, \tilde{h}_t, z_t)$ как взвешенной суммы соответствующих значений из предыдущего вектора h_{t-1} и нового вектора-кандидата $\tilde{h}_t$, где веса для значений под индексом i выбираются как $1 - z_t[i]$ и $z_t[i]$ соответственно.

В данной работе предлагается новая модель рекуррентного нейрона Recurrent Sigmoid Piecewise (RSP), в основе которой лежит Sigmoid Piecewise (SP) нейрон со следующей математической моделью:

$$SP(x; w_+, w_-, s, k) = \frac{w_+ \cdot x}{1 + e^{-k(s \cdot x)}} + \frac{w_- \cdot x}{1 + e^{k(s \cdot x)}}$$

Используя обозначение сигмоидального нейрона:

$$\sigma(x; s) = \frac{1}{1 + e^{s \cdot x}}$$

и $k = 1$ получаем:

$$SP(x; w_+, w_-, s) = \sigma(x; s)(w_+ \cdot x) + \sigma(x; -s)(w_- \cdot x)$$

Используя равенство $\sigma(x; -s) = 1 - \sigma(x; s)$:

$$SP(x; w_+, w_-, s) = (1 - \sigma(x; s))(w_- \cdot x) + \sigma(x; s)(w_+ \cdot x)$$

Если вместо одного SP нейрона описывается слой из N нейронов, то вместо векторов w_+, w_-, s будут использоваться матрицы W_+, W_-, S:

$$SP(x; W_+, W_-, S) = (1 - \sigma(x; S)) * (W_- \cdot x) + \sigma(x; S) * (W_+ \cdot x)$$

Введя обозначения $z = \sigma(x; S)$, $a = W_- \cdot x$ и $b = W_+ \cdot x$ получаем:

$$SP(x) = (1 - z) * a + z * b$$

Что очень похоже на блок рассчета нового вектора выходов в нейроне GRU:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Таким образом, слегка изменив SP нейрон, можно получить его рекуррентную версию, Recurrent Sigmoid Piecewise (RSP) нейрон, который принимает на вход вектор $p_t = [h_{t-1}, x_t]$ и выдает h_t:

$$h_t = RSP(p_t; W_+, W_-, S) = (1 - \sigma(p_t; S)) * (W_- \cdot p_t) + \sigma(p_t; S) * (W_+ \cdot p_t)$$

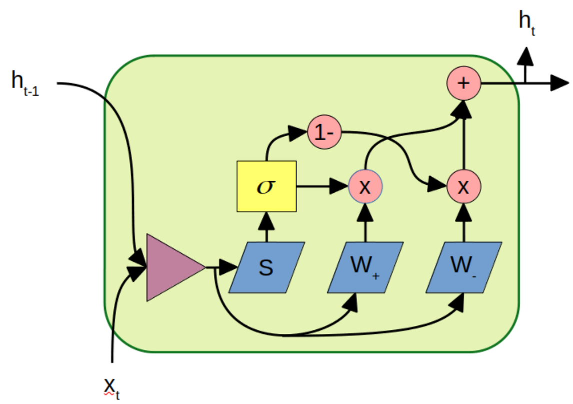
Либо же, по аналогии с LSTM/GRU нейронами, мат. модель RSP нейрона можно записать в несколько этапов/блоков:

$$z_t = \sigma(S \cdot [h_{t-1}, x_t])$$

$$q_t = W_- \cdot [h_{t-1}, x_t]$$

$$\tilde{h}_t = W_+ \cdot [h_{t-1}, x_t]$$
$$h_t = (1 - z_t) * q_t + z_t * \tilde{h}_t$$

И представить их в виде структурной схемы:

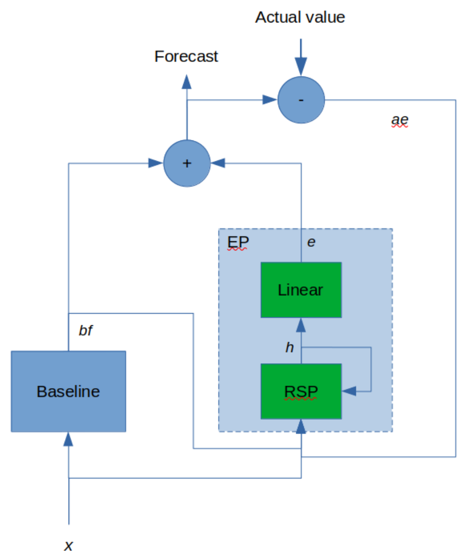


Поверхностно сравнив RSP нейрон с LSTM и GRU нейронами можно сделать следующие наблюдения:

- Математическая модель RSP нейрона проще (используется лишь один нелинейный сигмоидальный блок) чем модели LSTM и GRU нейронов. В задаче прогнозирования временных рядов более простые модели часто предпочтительны на практике.
- TODO

Применение рекуррентных сетей на основе RSP нейронов для прогнозирования временных рядов

Предлагается следующая схема прогнозирования временных рядов с использованием сетей на основе RSP нейрона:



где:

- x - входной вектор, предыдущие значения часового ряда (либо нескольких рядов)
- **Baseline** - "базисный"метод прогнозирования, рассчитывающий начальную оценку прогноза, например линейная регрессия; соответственно bf - базисное значение прогноза
- **EP** - error prediction блок, рассчитывающий оценку ошибки прогноза e базисного метода на основе: входного вектора, самого значения базисного прогноза и ошибку прогноза с предыдущего шага
- Базисный прогноз bf и оценка ошибки e складываются для получения финального прогноза: $f = bf + e$
- На следующем шаге прогнозирования также рассчитывается настоящая ошибка прогноза с предыдущего шага $ae = av - f$ и передается в блок рассчета ошибки прогноза текущего шага
- Блок рассчета ошибки состоит из RSP нейрона и простого линейного слоя. По своей математической модели RSP нейрон может "естественным"способом рассчитывать новое значение коррекции как взвешенную сумму предыдущей ошибки и нового значения контекста.

Преимущества данной схемы:

- В качестве базисного метода можно брать любой существующий метод прогнозирования, и таким образом в процессе обучения EP блок будет пытаться только улучшать прогноз базисного метода.

- Рассчет настоящего значения ошибки прогноза с предыдущего шага (шагов) в теории дает возможность ЕР блоку динамически "реагировать"на изменения в качестве прогноза. Также такая схема фактически реализует нелинейную error moving average составляющую ARMA модели.

- Мат. модель RSP нейрона естественным образом подходит для рассчета некоторой ошибки прогноза.

- В теории возможно поэтапное обучение ЕР и базисного блоков - на первом этапе обучаем параметры ЕР блока, на втором - фиксируем их и обучаем параметры базисного блока и т.д.

ДОБАВИТЬ:

- Использование нескольких слоев из RSP нейронов.

- Использование модифицированного positional encoding.

  В качестве дополнительной информации, которую можно передавать в нейронную сеть, предлагается использовать слегка модифицированный positional encoding вектор вида:

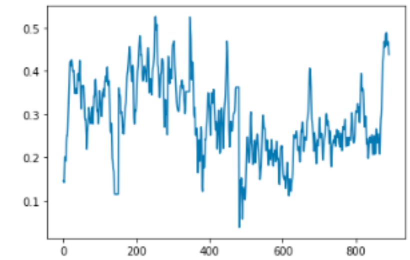
  $$p(t,k) = sin(w_k * t), k = 1, 2, ..., K,$$

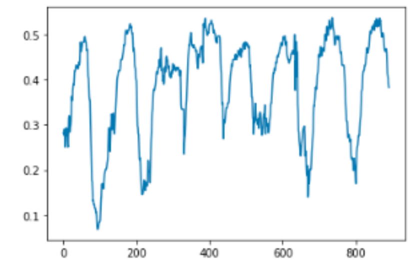
  где t - индекс текущего значения временного ряда, $w_1, ..., w_k$ - частоты, использующиеся для построения вектора модифицированного positional encoding. Добавление этой информации позволит модели проще предсказывать значения сезонных временных рядов.

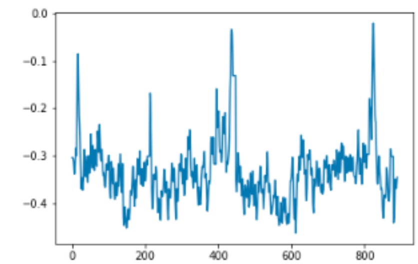
- Модификация функции ошибки для уменьшение влияния проблемы затухающего градиента. Эта идея пока не полностью формализована, но мне кажется, что можно добавить определенные составляющие в функцию ошибки, которые будут влиять на параметры сети таким образом, чтобы уменьшать затухание градиентов.

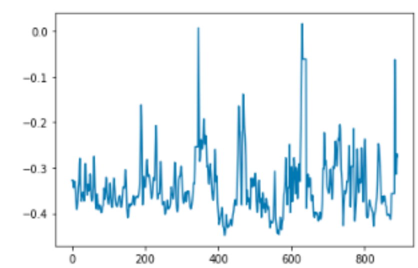
Практические примеры использования рекуррентных RSP сетей

Для тестирования использовались показатели сердечного ритма 4 разных пациентов в разных состояниях:







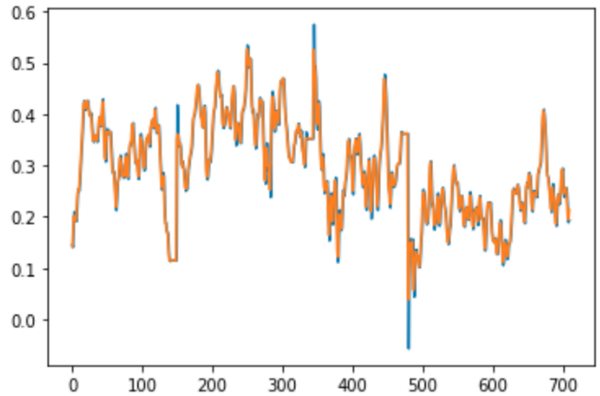


Построив "базисную"модель - оптимальный линейный предиктор получаем следующие среднеквадратические ошибки прогноза на обучающей и тестовой выборке (усредненный по всем 4 пациентам):

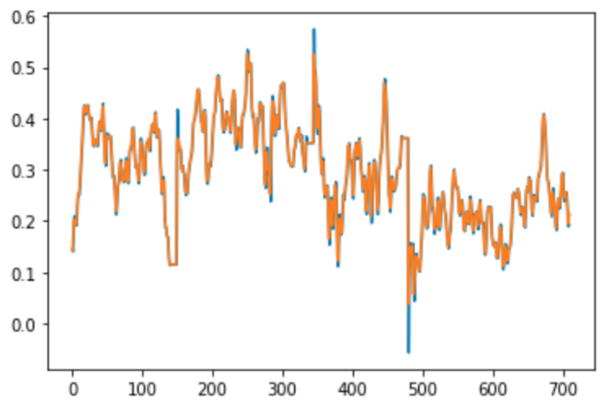
$$L_{train}^{baseline} = 0.0004731$$

$$L_{test}^{baseline} = 0.00048$$

Пример прогнозов линейного предиктора для одного пациента на обучающей:



и тестовой:



выборках.

После "фиксации"базисного предиктора, добавления блока корекции на основе RSP нейрона и обучения его параметров получаем следующие значения среднеквадратических ошибок:

$$L_{train}^{new} = 0.0004442$$

$$L_{test}^{new} = 0.00046$$

что соотвествует приблизительно 5% уменьшению среднеквадратической ошибки прогноза.

РАСШИРИТЬ ЭТУ ЧАСТЬ

Выводы и дальнейшие направления работы