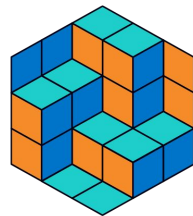


Предсказание временных рядов

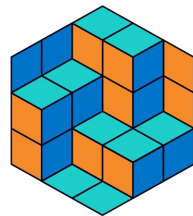
Лекция 4



Глава 1

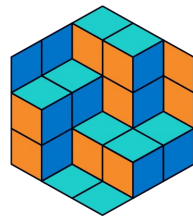
ETS Model

Вспомним



Давайте посмотрим на еще одну вариацию авторегрессионного алгоритма, только теперь мы будем:

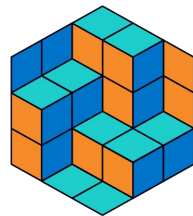
- Брать лаг



Вспомним

Давайте посмотрим на еще одну вариацию авторегрессионного алгоритма, только теперь мы будем:

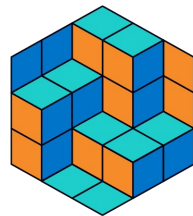
- Брать лаг
- Учитывать последние n значений



Вспомним

Давайте посмотрим на еще одну вариацию авторегрессионного алгоритма, только теперь мы будем:

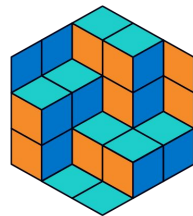
- Брать лаг
- Учитывать последние n значений
- Учитывать все значения



Вспомним

Давайте посмотрим на еще одну вариацию авторегрессионного алгоритма, только теперь мы будем:

- Брать лаг
- Учитывать последние n значений
- Учитывать все значения
- Учитывать все значения, но по разному



Вспомним

Давайте посмотрим на еще одну вариацию авторегрессионного алгоритма, только теперь мы будем:

- Брать лаг
- Учитывать последние n значений
- Учитывать все значения
- Учитывать все значения, но по разному

Как же это сделать?

Как, а?



Как же это сделать?

Можно ведь просто давать более старым переменным меньшие веса!



А как?

Тут нам поможет **экспоненциальное сглаживание**.

Выглядит это так:

$$\hat{y}_{T+1|T} = \alpha \sum_{i=0}^{\infty} (1 - \alpha)^i y_{t-i}$$

А как?

Тут нам поможет **экспоненциальное сглаживание**.

Выглядит это так:

$$\hat{y}_{T+1|T} = \alpha \sum_{i=0}^{\infty} (1 - \alpha)^i y_{t-i}$$

То есть с каждым новым шагом мы как раз и будем давать все меньший вес нашим старым значениям.

А как?

И задача в простом понимании сводится к формулам:

Прогноз: $\hat{y}_{t+h|t} = l_t$

Сглаживание: $l_t = \alpha y_t + (1 - \alpha)l_{t-1}$

Но...

Но как же тренд? А сезонность? А какой-то шум?

Но...

Но как же тренд? А сезонность? А какой-то шум?

Оно тут тоже есть - **Error**, **Seasonal**, **Trend** вместе и дают название модели.

ETS. Error

Если ошибка аддитивная, то функция будет такой:

$$y_{T+1} = l_t + b_t + \varepsilon_{T+1}$$

А если мультипликативной, то:

$$y_{T+1} = (l_t + b_t)(1 + \varepsilon_{T+1}).$$

ETS. Trend

Аналогично - если тренда нет, то ничего не меняется. Если он постоянен:

$$\hat{y}_{T+h} = l_t + hb_t$$

А если сходит на нет со временем, то:

$$y_t = l_t + \phi_h b_t, \phi_h = \sum_{i=1}^h \phi^i$$

ETS. Seasonal

И сезонность - если аддитивная:

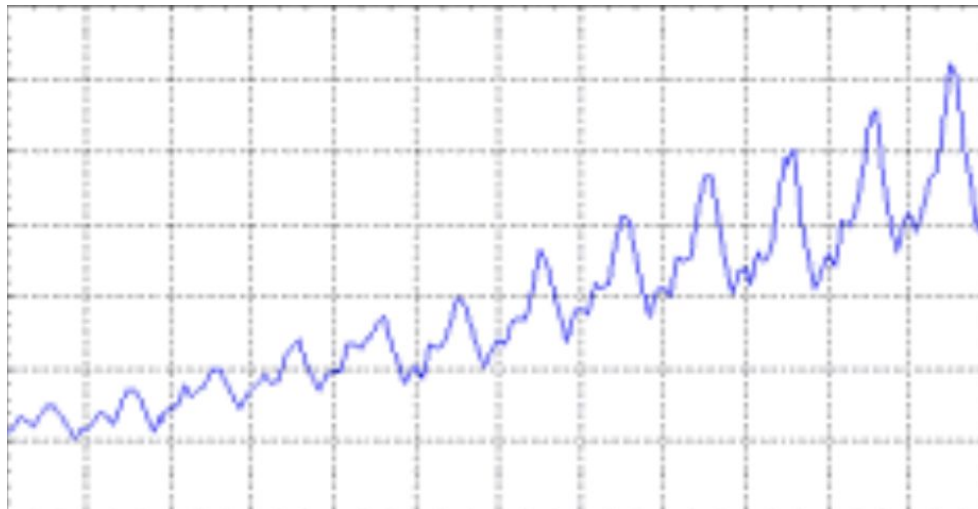
$$\hat{y}_{T+h} = l_t + s_{t+h-m(k+1)}, \text{ где } m - \text{период, } k = [(h-1)/m].$$

А если мультипликативная:

$$\hat{y}_{T+h} = l_t s_{t+h-m(k+1)}$$

Отступление

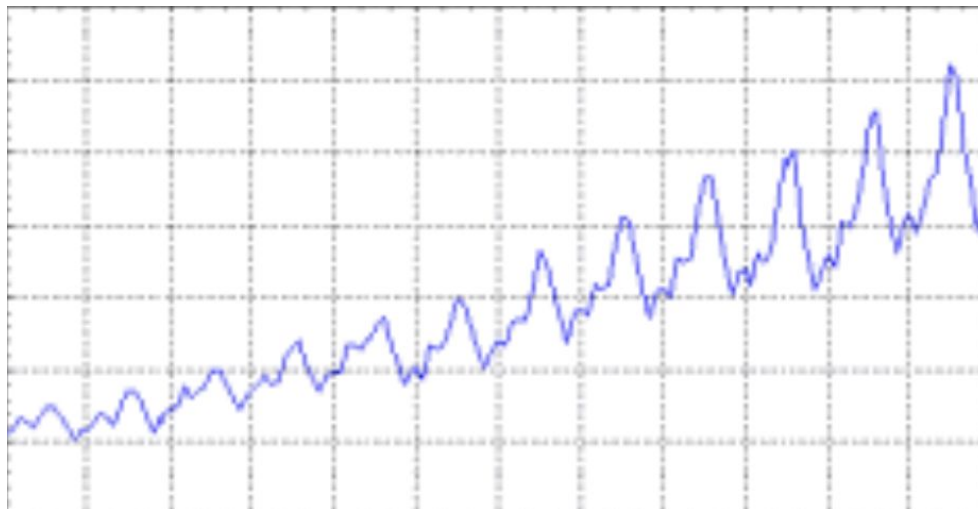
Кстати, а если у нас сезонность сохраняется, но дисперсия в каждом сезоне мультипликативно растет, что можно сделать?



Отступление

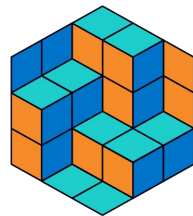
Кстати, а если у нас сезонность сохраняется, но дисперсия в каждом сезоне мультипликативно растет, что можно сделать?

Можно взять логарифм))



Как итог

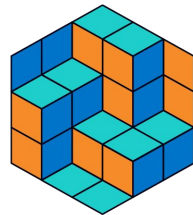
Мы получаем большую и сложную модель, которая в разных случаях выглядит по-разному. Но работает она здорово!



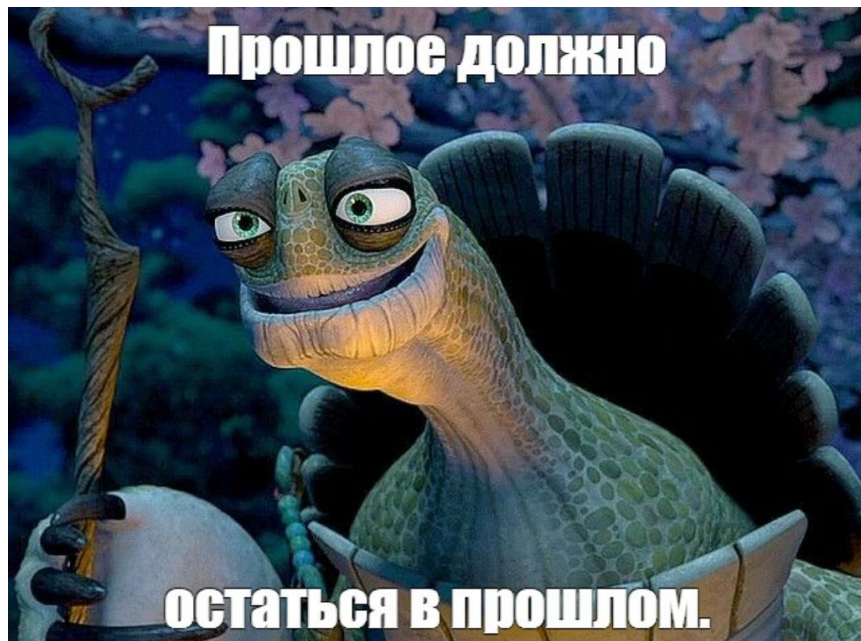
Глава 2

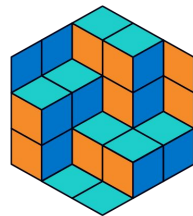
Prophet

Наши дни



Чтож, базовые методы мы разобрали, но что же происходит в наши дни?

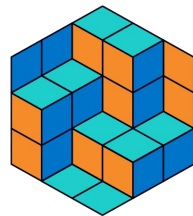




Наши дни

В 2017 году произошли следующие события:

- “Зенит” вылетел из Лиги Европы
- В 99-й раз отметили День Защитника Отечества
- Facebook выпустил **Prophet**



Наши дни

Что это?

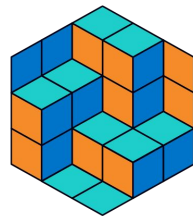
- *open-source* проект

Зачем?

- user-friendly система для предсказания внутренних процессов

Как?

- Сейчас разберемся



Prophet

Выглядит он просто:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

Где:

- $g(t)$ - тренд
- $s(t)$ - сезонность
- $h(t)$ - аномальные дни
- ϵ_t - шум

Prophet. Тренд

- Кусочно-линейная или логистическая функция
- Можно подать на вход точки изменения тренда...
- ... или Prophet найдет их сам :)



Prophet. Сезонность

- Недельная сезонность:
 - 6 экзогенных параметров: **пн**, **вт**, **ср**, **чт**, **пт**, **сб**
 - А куда делось **вс**?
- Годовая сезонность - ряды Фурье.

Сезон на севере



Лето Осень Зима Весна

Сезон на юге



Лето Осень Зима Весна

Prophet. Аномальные дни

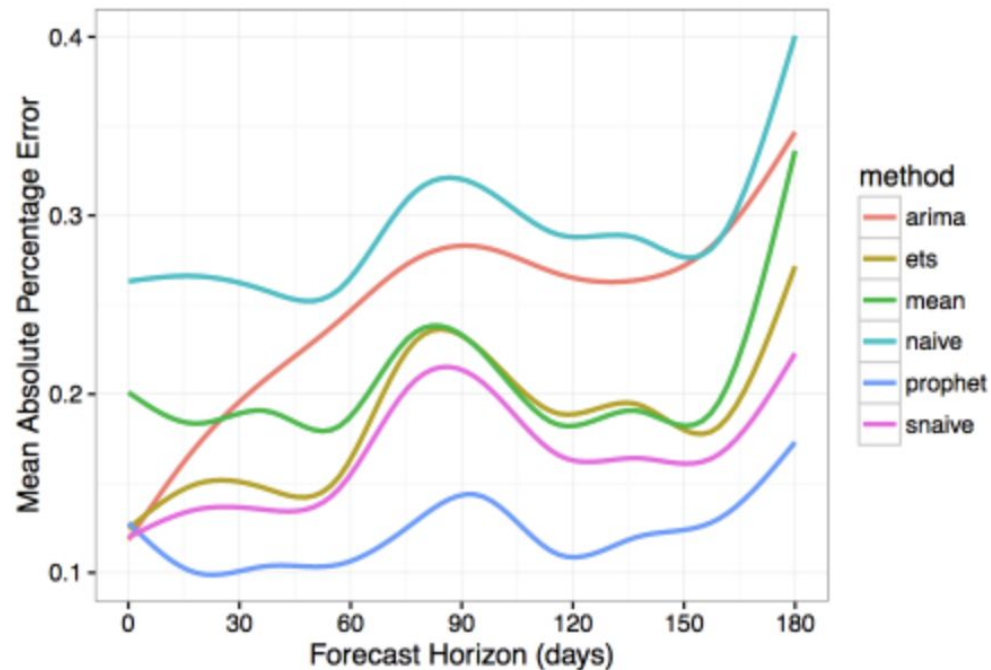
- Праздники
- Выходные
- Нерегулярные праздники - черная пятница, Хеллоуин.

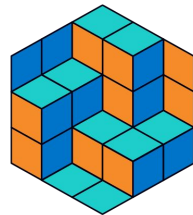


Prophet. Обучение

Как можно заметить, Prophet относится к **авторегрессионным моделям** - а значит обучается он так же, как **ARIMA**

Prophet. Профит





Глава 3

RNN & LSTM

Слишком просто

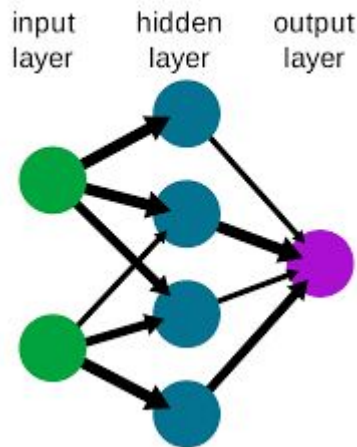
Давайте попробуем посложнее!

Neural Network

Работает просто (ну почти):

- На вход даем вектор - *input layer*
- Умножаем его на матрицу весов - *hidden layer*
- Умножаем еще раз, получаем ответ - *output layer*

A simple neural network



Neural Network

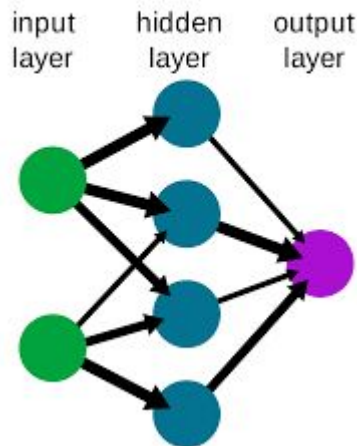
Работает просто (ну почти):

- На вход даем вектор - *input layer*
- Умножаем его на матрицу весов - *hidden layer*
- Умножаем еще раз, получаем ответ - *output layer*

Как правило, не все так просто, конечно:

- Между линейными слоями (где умножаем на матрицу) вставляют нелинейные преобразования
- Размер сети, конечно, больше чем на картинке)

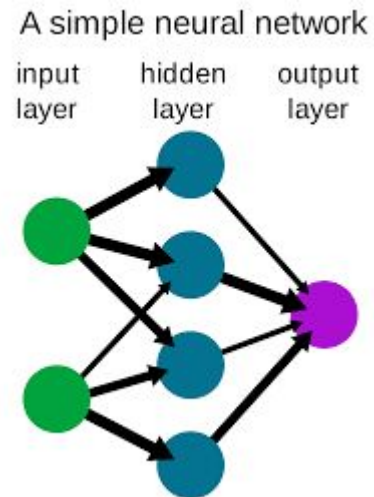
A simple neural network



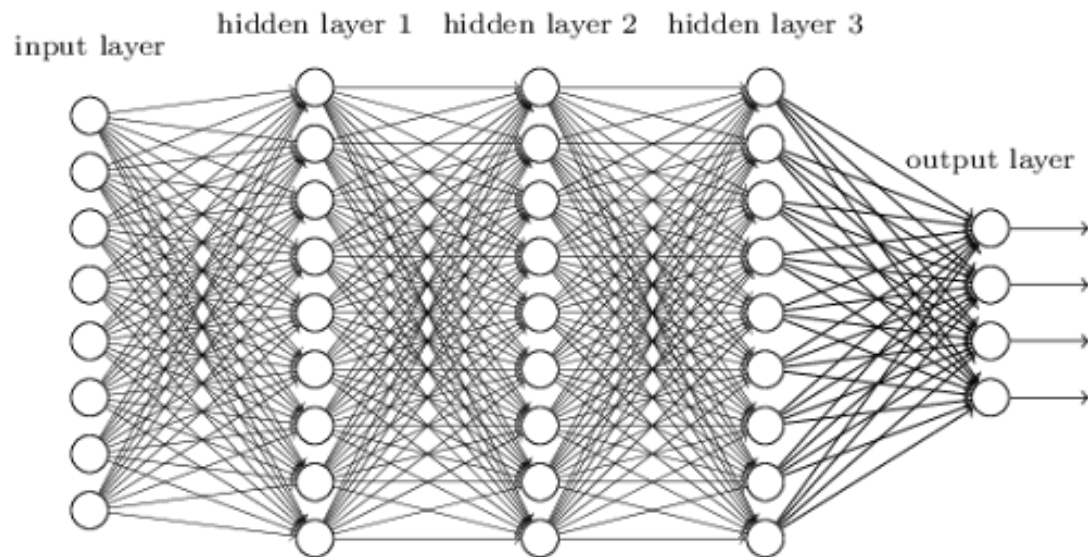
Neural Network

Обучение:

- Градиентный спуск + обратное распространение ошибки



Big Neural Network



И то это еще не большая...

Self Neural Network?

Мы видим что нейроны связаны со следующими нейронами.

А если мы хотим сделать связь каждого слоя с ним же?

Self Neural Network?

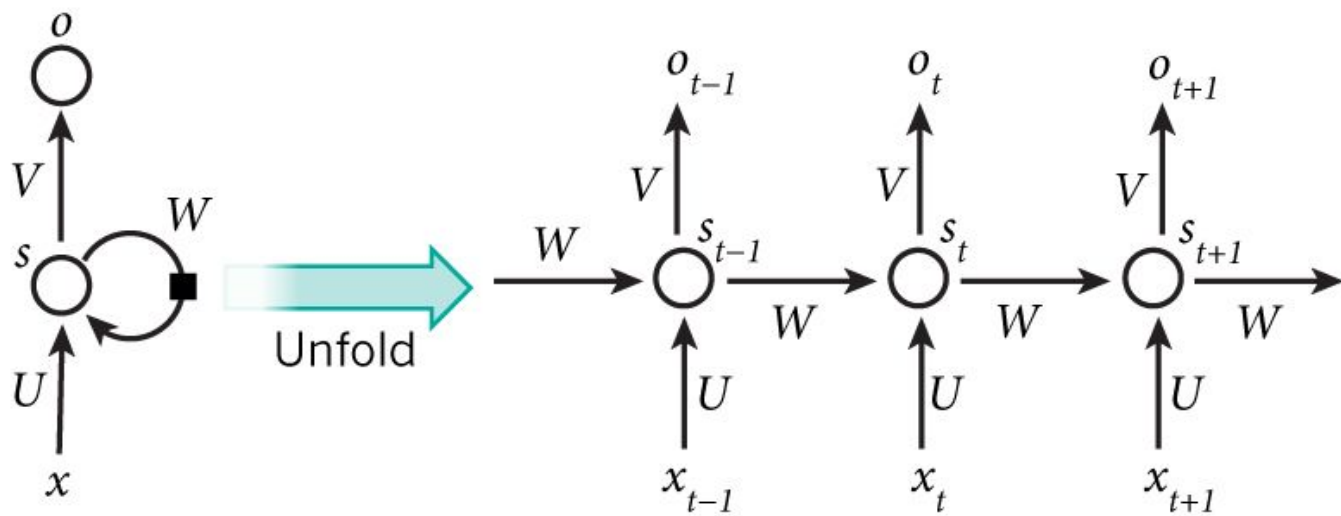
Мы видим что нейроны связаны со следующими нейронами.

А если мы хотим сделать связь каждого слоя с ним же?

Тут и появляются **Recurrent Neural Network**

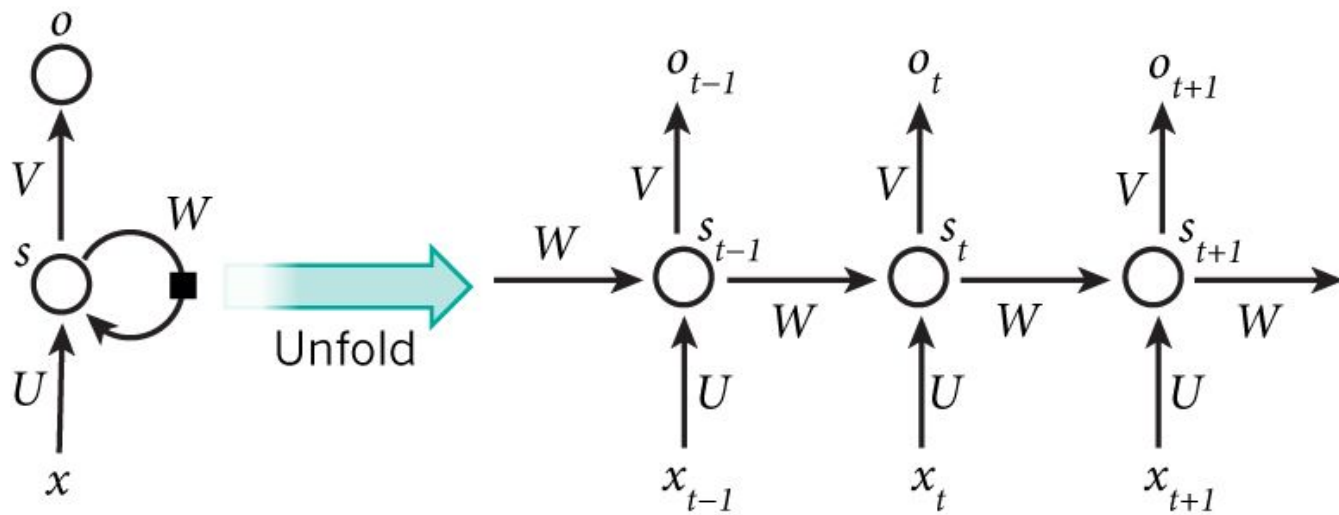
RNN

Механизм выглядит так:



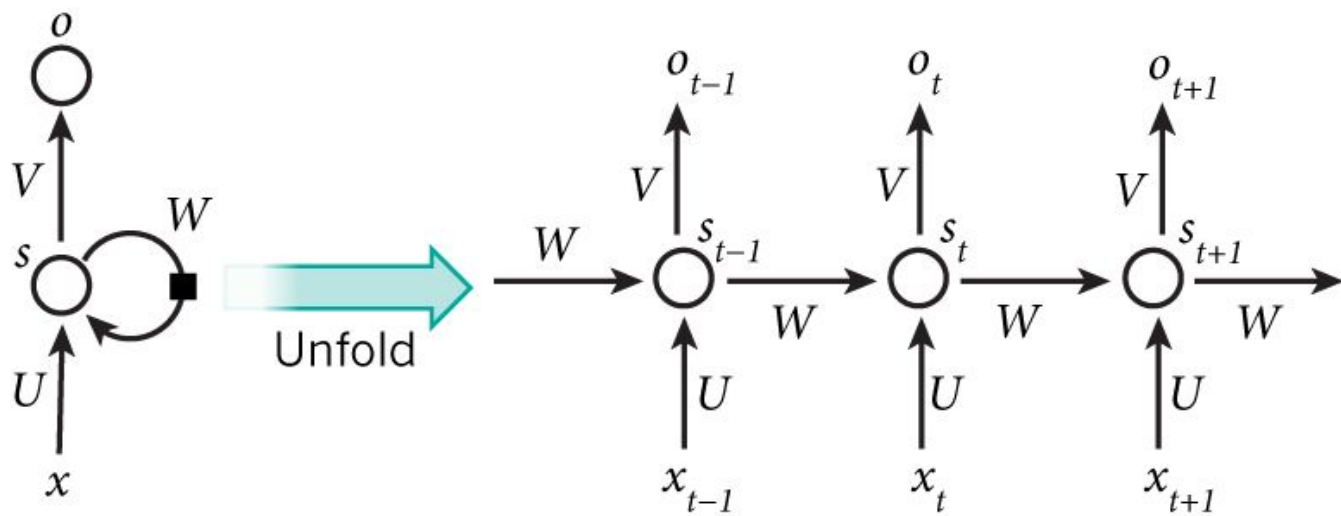
RNN

Таким образом, получаем сеть, которая может обрабатывать последовательности и выдавать ответы (1 или несколько)



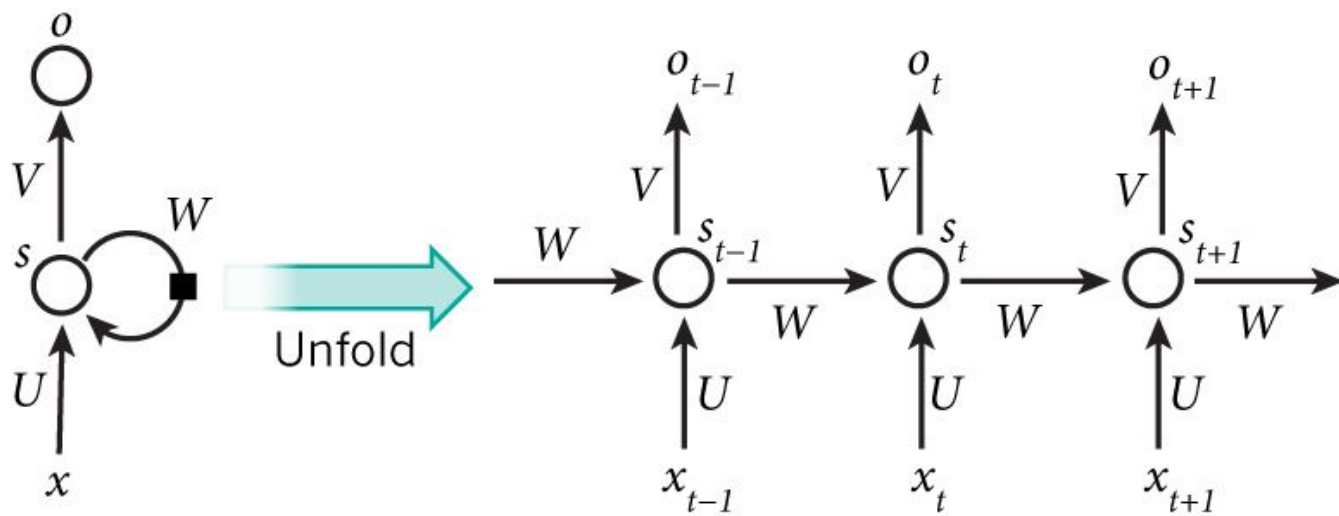
RNN

Это же ровно то что нам и нужно!!!



RNN

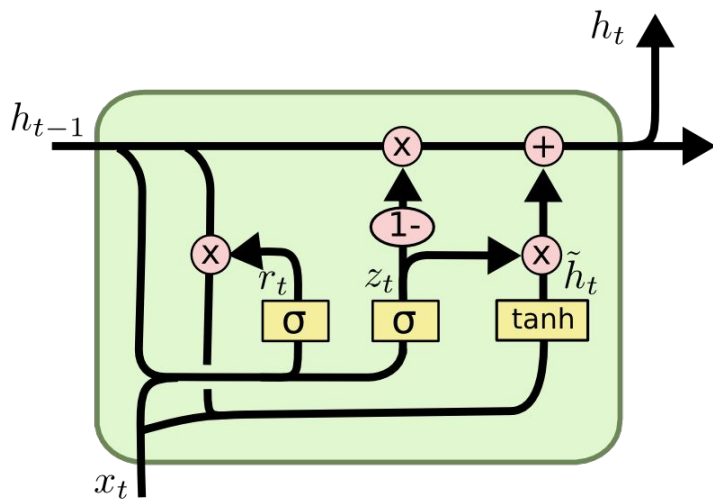
Но есть нюанс - по схеме видно, что сама по себе информация о старых значениях “забывается”. Что с этим делать?



LSTM

LSTM - Long short-term memory

Немного усложненный вид RNN:



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

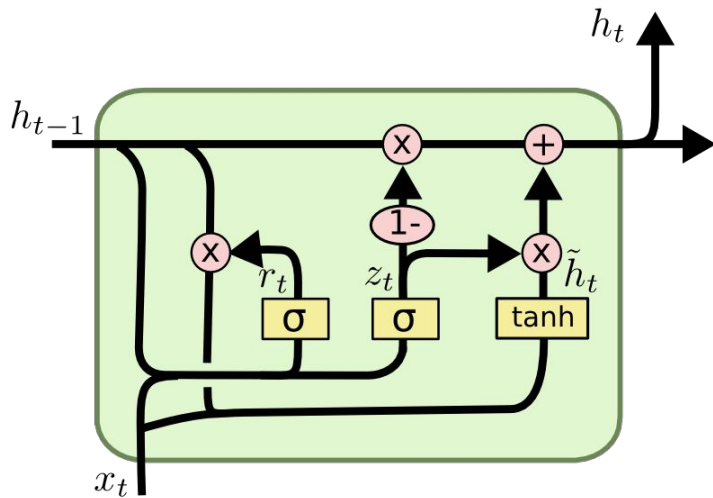
$$r_t = \sigma (W_r \cdot [h_{t-1}, x_t])$$

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

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

LSTM

Зачем? Видите вот эту h , которая на вход приходит? Ровно она и несет в себе память из предыдущих входов. Круто, правда?)



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

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

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

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

LSTM

Обучение:

- Все так же **градиентный спуск**

LSTM

Обучение:

- Все так же **градиентный спуск**

Плюсы:

- Зачастую очень мощный инструмент, который используется во многих задачах машинного обучения

Минусы:

- Реализовывать заметно сложнее(

Вопросы?

