



# ТЕХНОСФЕРА

## Лекция 8 Рекуррентные нейронные сети

Нестеров Павел. Храбров Кузьма

3 ноября 2021 г.

# План лекции

Предпосылки

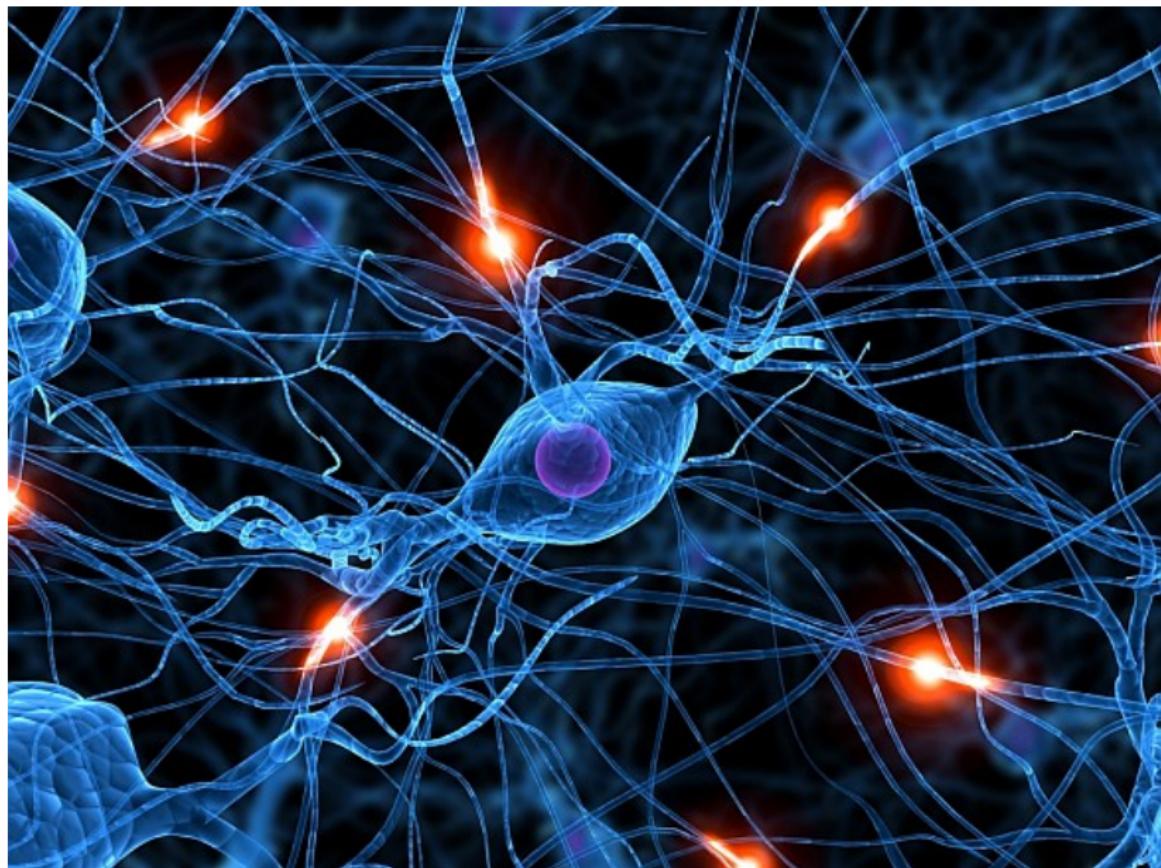
Развитие RNN

RNN для последовательностей

Simple RNN

Backpropagation through time

## Биологическая нейронная сеть



# Искусственная сеть прямого распространения

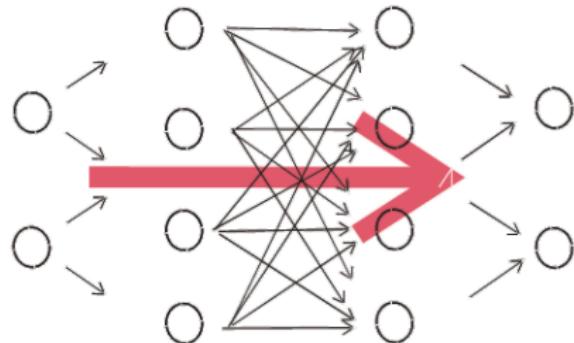


Рис.: Модель многослойной сети прямого распространения

- ▶ аппроксимирует любую функцию
- ▶ Основная часть публикаций до сих пор именно об этой модели

# Искусственная рекуррентная нейронная сеть

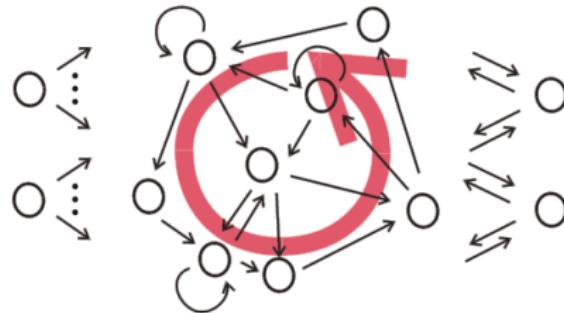


Рис.: Модель рекуррентной сети<sup>1</sup>

- ▶ все биологические сети - рекуррентные
- ▶ RNN моделирует динамическую систему
- ▶ существует несколько алгоритмов обучения без явного лидера

---

<sup>1</sup><http://minds.jacobs-university.de/sites/default/files/uploads/papers/ESNTutorialRev.pdf>

# Универсальная теорема аппроксимации

- ▶ MLP: аппроксимирует любую функцию;
- ▶ RNN: аппроксимирует поведение любой динамической системы<sup>2</sup>;
- ▶ RNN: все машины Тьюринга могут быть смоделированы полносвязной рекуррентной нейронной сетью с сигмоидальной функцией активации<sup>3</sup>.

Таким образом: тренировка многослойного персептрона - это оптимизация функций, а тренировка рекуррентной сети - это оптимизация программ.

---

<sup>2</sup><http://minds.jacobs-university.de/sites/default/files/uploads/papers/ESNTutorialRev.pdf>

<sup>3</sup>Siegelmann & Sontag, 1991, Applied Mathematics Letters, vol 4, pp 77-80.

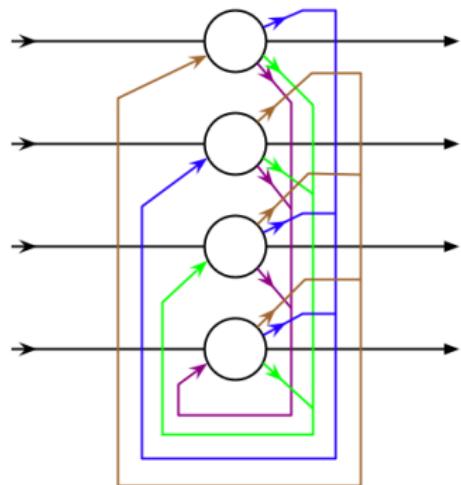
# Направления исследований RNN

- ▶ 1958, Розенблatt: персепtron с обратной связью, но после статьи Минского про него как то забыли;
- ▶ 1978, Хопфилд: энергетическая интерпретация сетей с обратной связью;
- ▶ 1986, Майкл Джордан: рекуррентная нейросеть с единичной задержкой;
- ▶ 1990, Джеф Элман: апгрейдит сеть Джордана и внедряет на практике;
- ▶ 1997, Ёрген Шмидтхубер: long short term memory<sup>4</sup>
- ▶ 2005+, все: глубокие сети и мультимодальное обучение
- ▶ 2017 Ashish Vaswani et al. : Transformer

---

<sup>4</sup>[http://en.wikipedia.org/wiki/Long\\_short\\_term\\_memory](http://en.wikipedia.org/wiki/Long_short_term_memory)

# Нейросеть Хопфилда



- ▶ энергия :  $E = -\sum_i s_i b_i - \sum_{i < j} s_i s_j w_{ij}$
- ▶ ассоциативная память
- ▶ обратная связь
- ▶ пороговая функция активации

Такая сеть (рекуррентная нейронная сеть) может находиться как в стабильном состоянии, осциллировать, или даже проявлять признаки детерминированного хаоса.

Хопфилд показал, что при симметричной матрице весов, существует такая функция энергии бинарных состояний системы, что при симуляции система эволюционирует в одно из низко-энергетических состояний.

## Нейросеть Хопфилда, эволюция системы

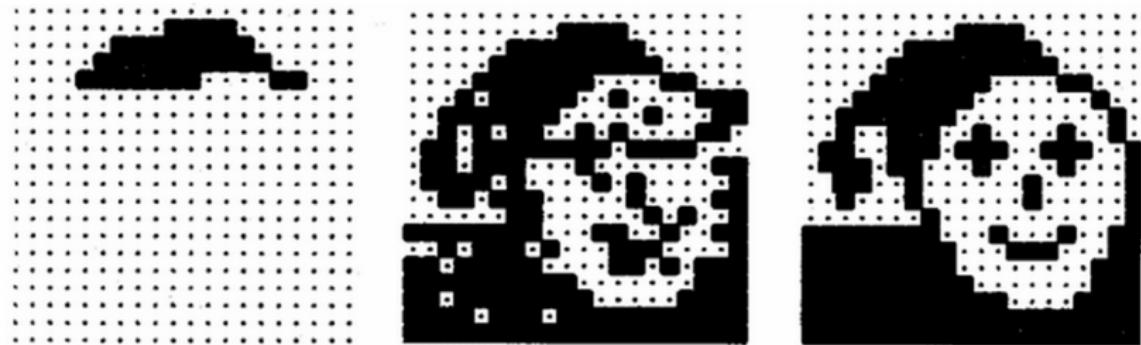
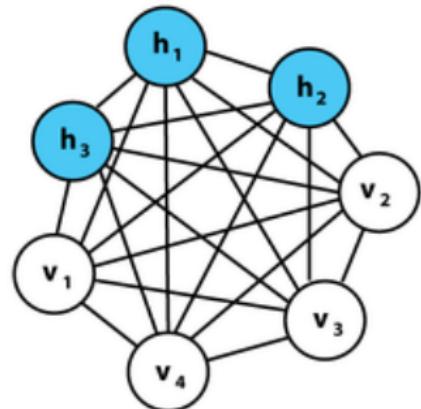


Рис.: Восстановление запомненного образа

# Машина Больцмана - стохастический генеративный вариант сети Хопфилда



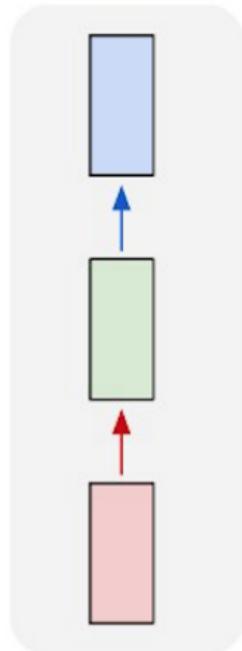
- ▶ энергия не изменилась:  
$$E = - \sum_i s_i b_i - \sum_{i < j} s_i s_j w_{ij}$$
- ▶ симметричная матрица весов  
 $w_{ij} = w_{ji}$ , но нет обратных связей:  
 $w_{ii} = 0$
- ▶ появились скрытые состояния  
(система ищет такую конфигурацию скрытых состояний которая лучшим образом описывает видимые состояния)
- ▶  $\forall i : s_i \in \{0, 1\}$
- ▶ стохастический нейрон
- ▶ цель: научиться описывать видимые переменные  $\vec{v}$  с помощью скрытых  $\vec{h}$  (напоминает автоэнкодер?)

# Моделирование последовательностей

- ▶ преобразование последовательности одной природы в последовательность другой природы
  - ▶ последовательность звуков в последовательность слов
- ▶ предсказание следующего члена последовательности (граница между обучением с учителем и без учителя становится все тоньше, вспомним хотя бы автоенкодеры)
  - ▶ временные ряды
  - ▶ изображения: прогнозирование следующего пикселя на основе окружения (смотрим house generate.gif)
  - ▶ видео: следующий кадр на основе предыдущих
  - ▶ текст: генерация следующего слова

# Обычный MLP

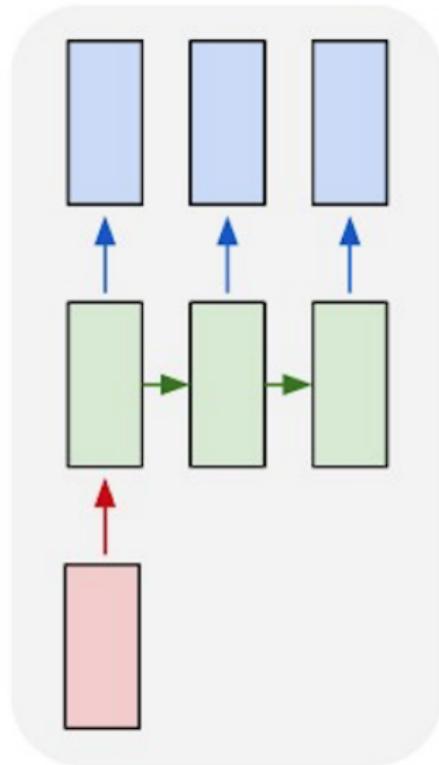
one to one



используется для отображения одного вектора/примера в другой, например для классификации

RNN один ко многим, #1

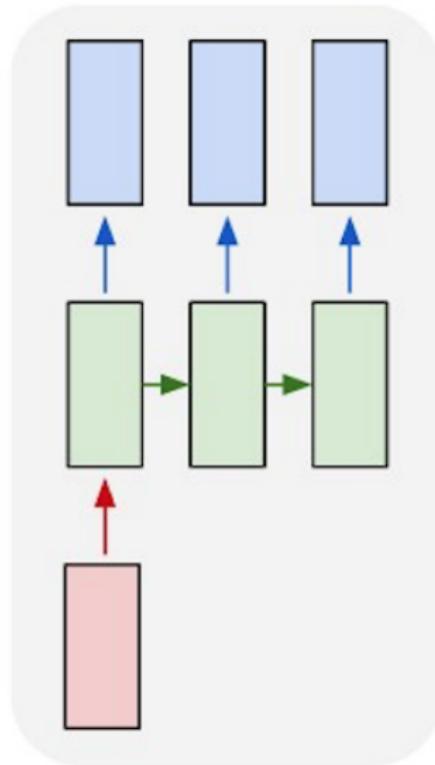
one to many



для чего?

## RNN один ко многим, #2

### one to many

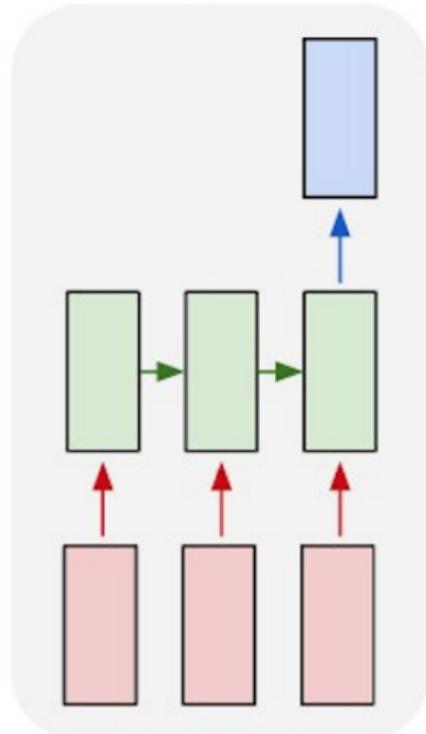


отображение одного примера в  
последовательность

- ▶ описание картинки естественным языком
- ▶ генерация музыки по стилю

RNN многие к одному, #1

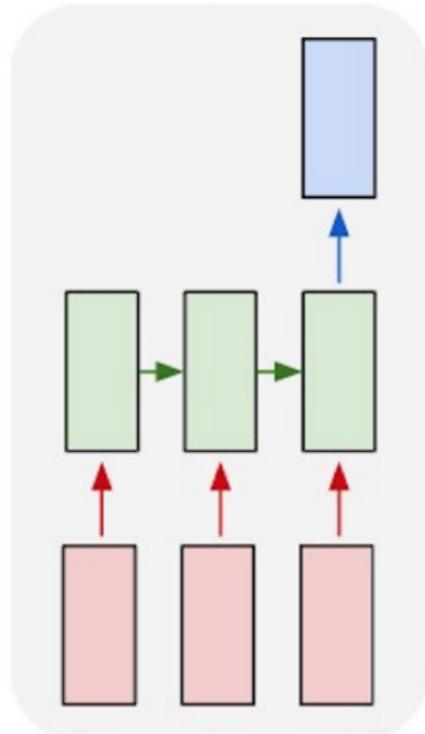
many to one



для чего?

## RNN многие к одному, #2

### many to one

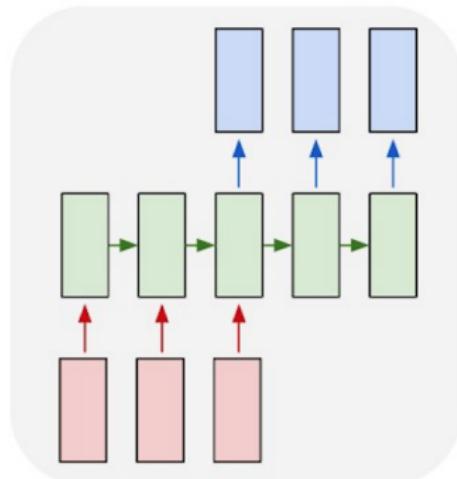


последовательность в пример

- ▶ определение тональности текста
- ▶ определение стиля изображения

# RNN многие ко многим асинхронно, #1

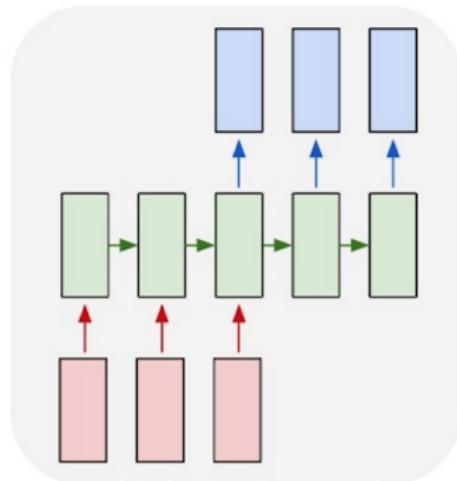
many to many



для чего?

# RNN многие ко многим асинхронно, #2

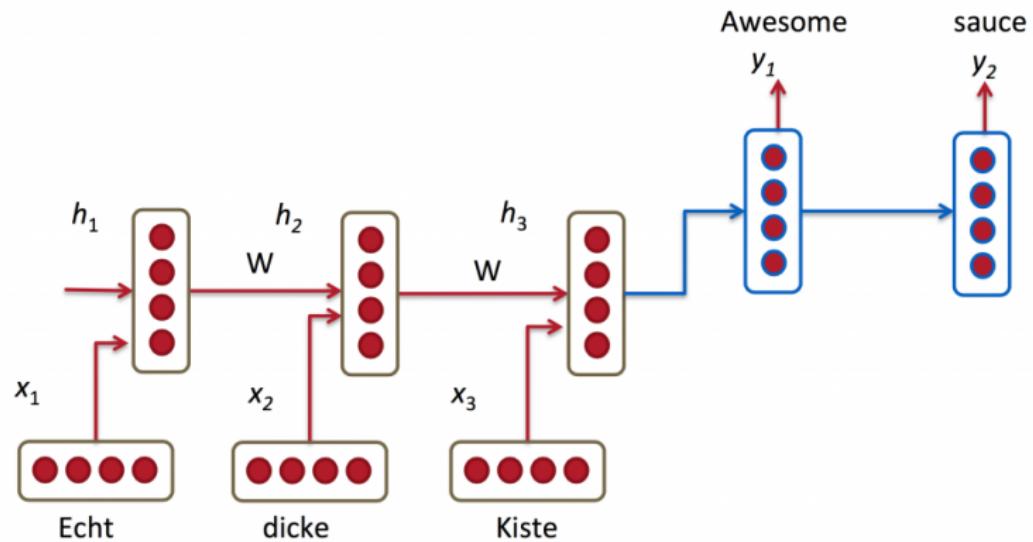
many to many



последовательность в  
последовательность

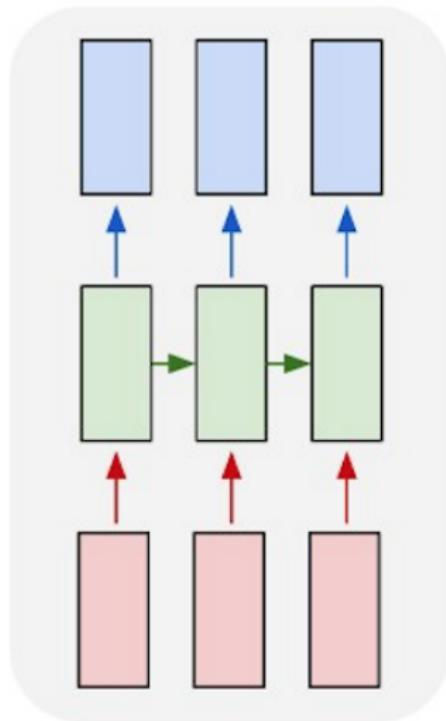
- ▶ перевод текста (зачем асинхронность?)

## RNN многие ко многим асинхронно, #3



RNN многие ко многим синхронно, #1

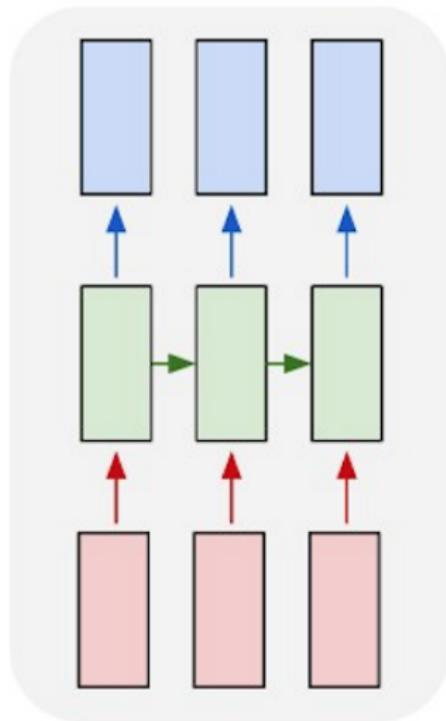
many to many



для чего?

RNN многие ко многим синхронно, #2

## many to many



последовательность в  
последовательность

- ▶ описать каждый кадр видео

## Способы: авторегрессионная модель

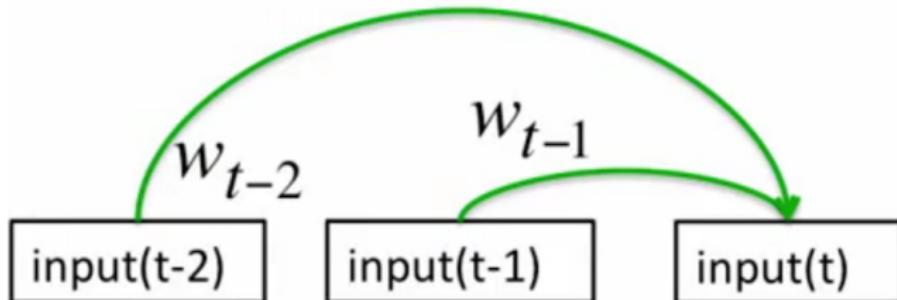
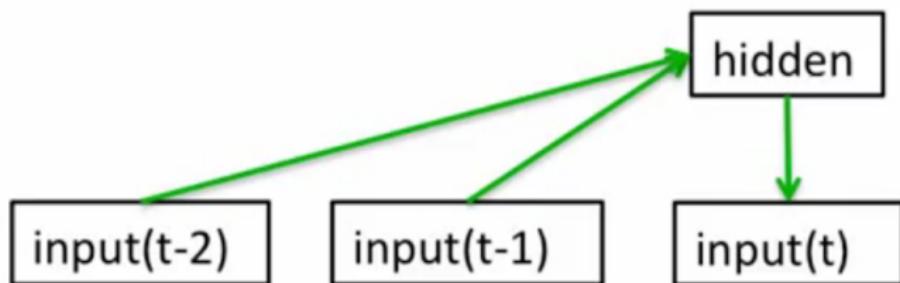
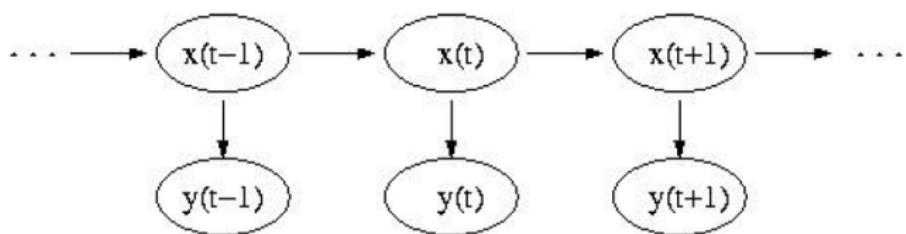


Рис.: Модель авторегрессии

## Способы: MLP



## Способы: скрытые модели Маркова



## Simple RNN, #1

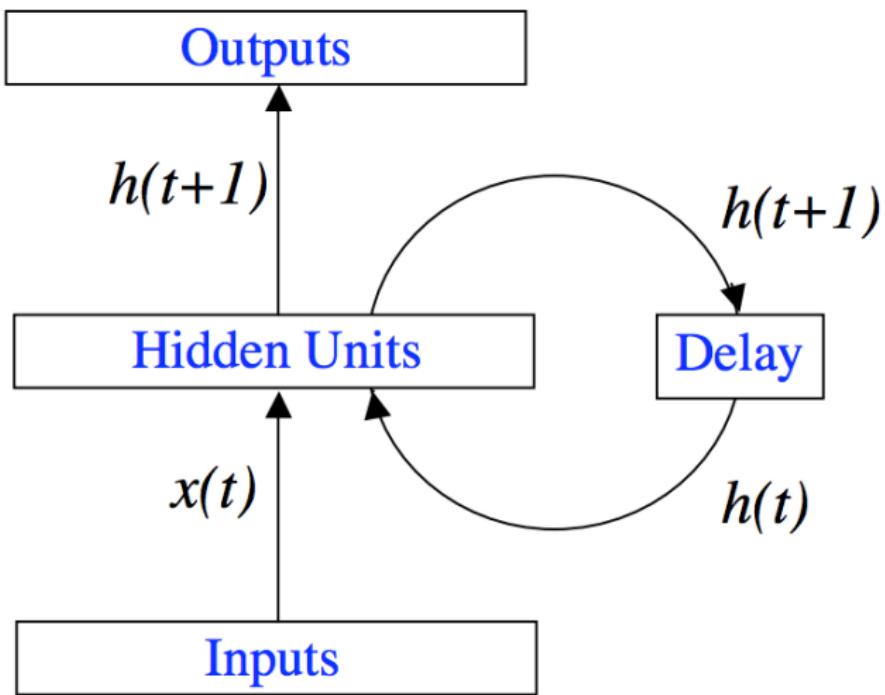
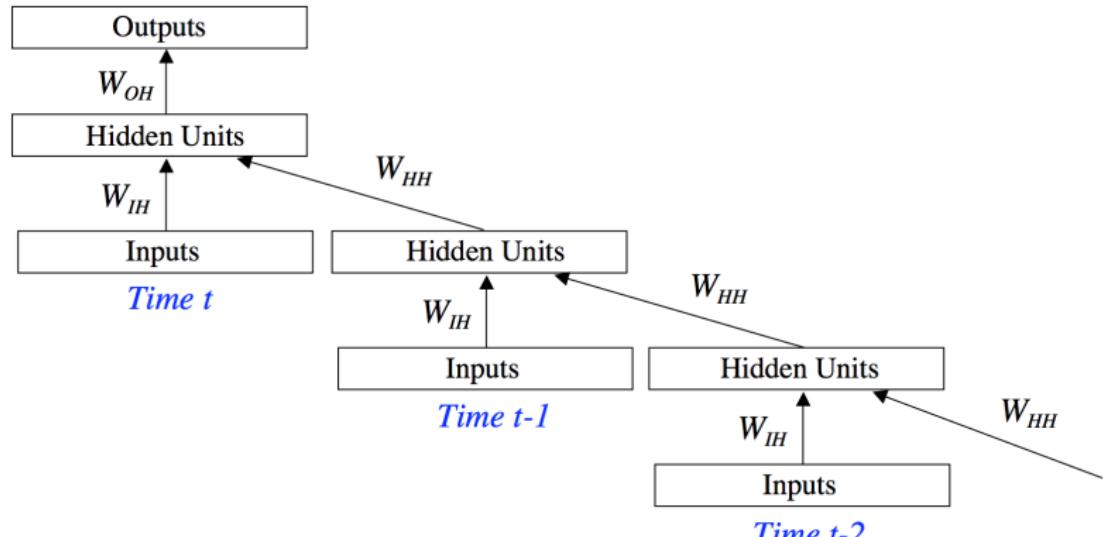


Рис.: RNN с задержкой на скрытом слое<sup>5</sup>

<sup>5</sup><http://www.cs.bham.ac.uk/~jxb/INC/112.pdf>

# Simple RNN, #2



Pic.: RNN unfolding<sup>6</sup>

<sup>6</sup><http://www.cs.bham.ac.uk/~jxb/INC/112.pdf>

# Simple RNN, #3

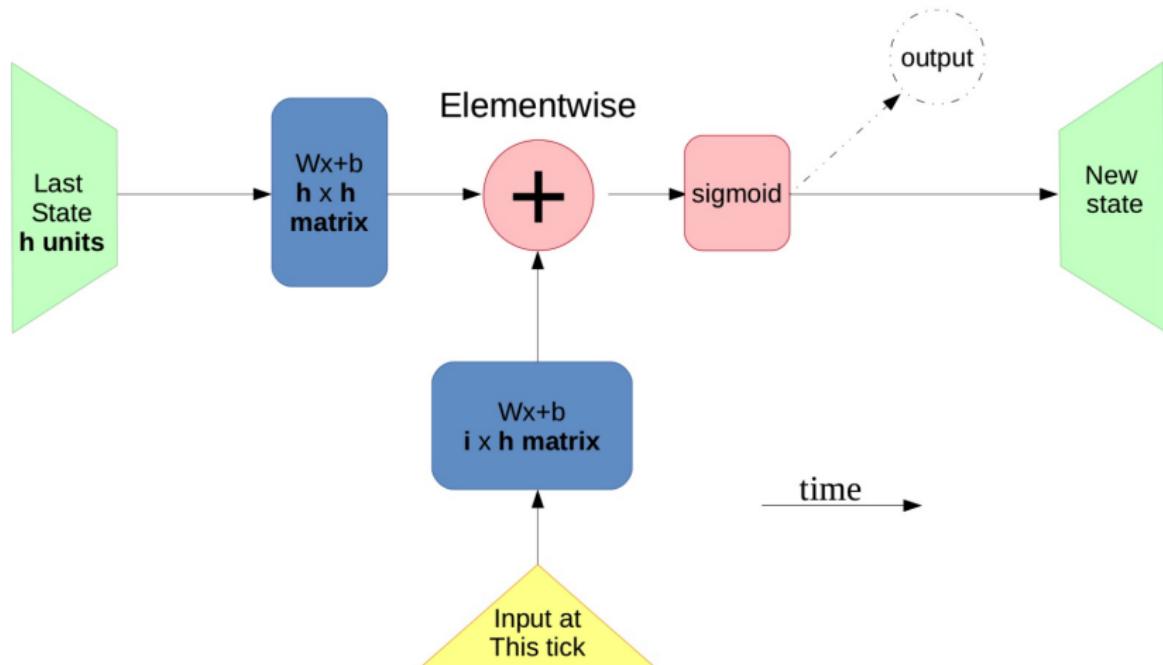


Рис.: RNN cell<sup>7</sup>

<sup>7</sup>[https://github.com/yandexdataschool/Practical\\_DL](https://github.com/yandexdataschool/Practical_DL)

## Simple RNN, #4

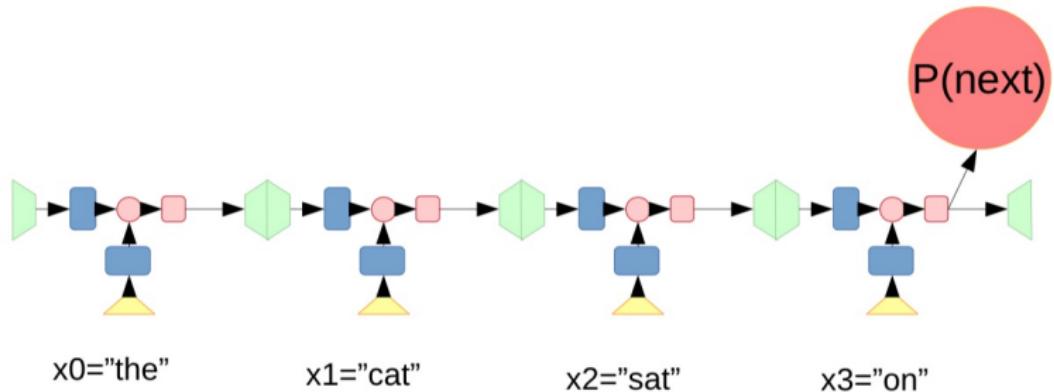


Рис.: RNN<sup>8</sup>

<sup>8</sup>[https://github.com/yandexdataschool/Practical\\_DL](https://github.com/yandexdataschool/Practical_DL)

# Detailed example 1

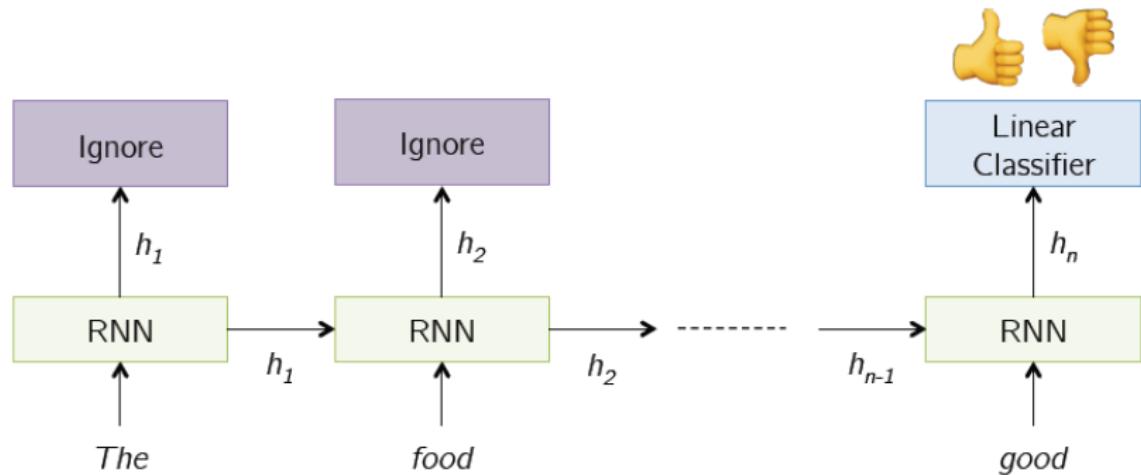


Рис.: Первый вариант для предсказания тональности<sup>9</sup>

<sup>9</sup><http://slazebni.cs.illinois.edu/spring17/>

## Detailed example 2

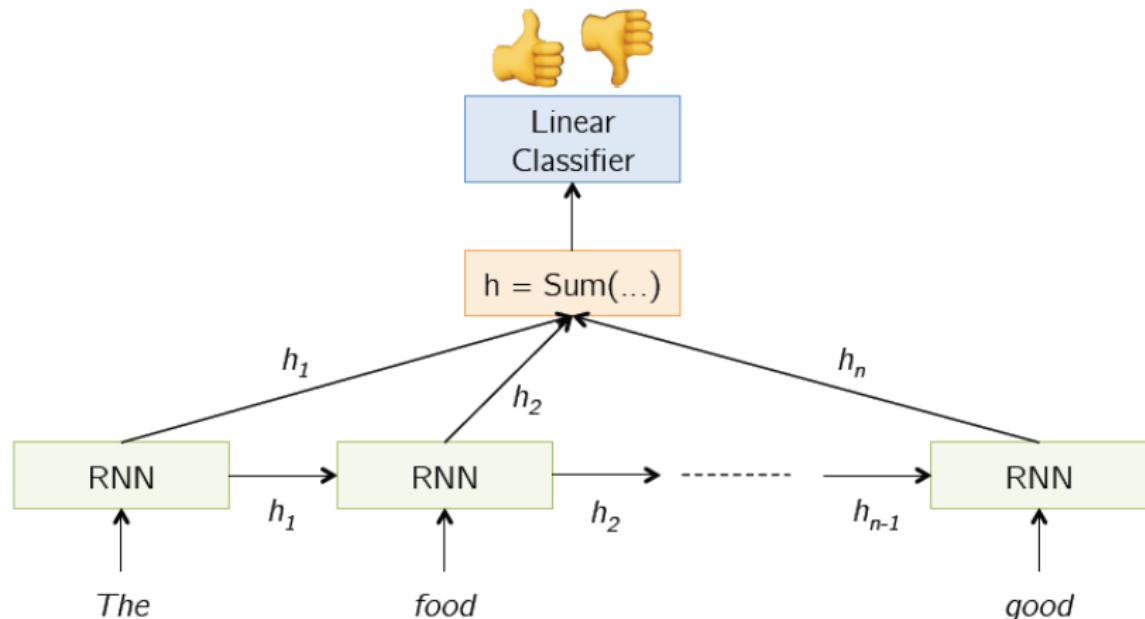


Рис.: Второй вариант для предсказания тональности<sup>10</sup>

<sup>10</sup><http://slazebni.cs.illinois.edu/spring17/>

# Deep RNN

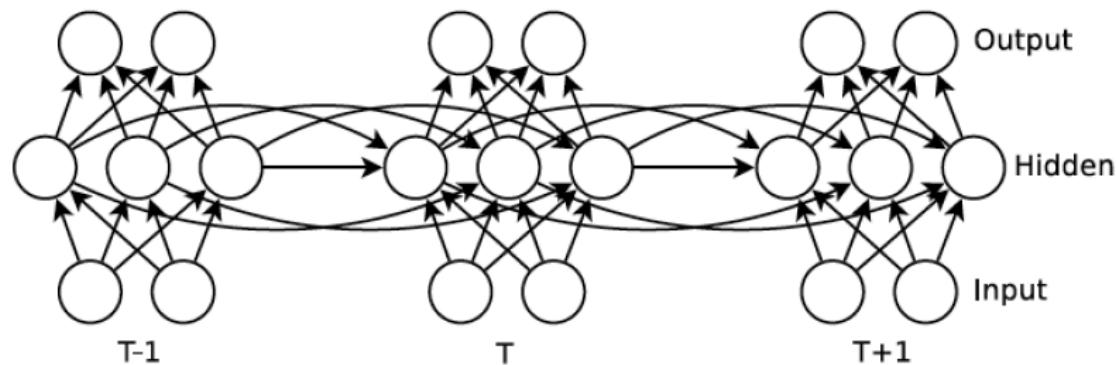


Рис.: Generating Text with Recurrent Neural Networks<sup>11</sup>

<sup>11</sup><http://www.cs.toronto.edu/~ilya/pubs/2011/LANG-RNN.pdf>

# Backpropagation through time. Problems with gradients

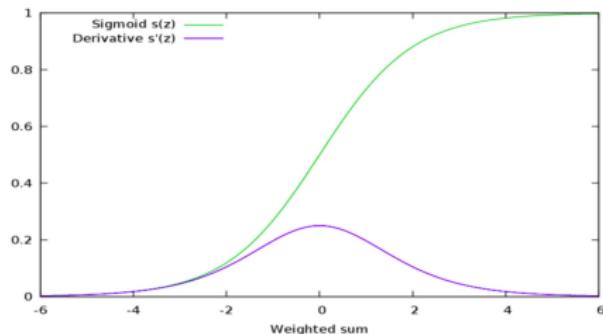
$$L = \sum_t L_t$$

$$\frac{\partial L_t}{\partial w} = \frac{\partial L_t}{\partial y_t} \frac{\partial y_t}{\partial w}$$

$$\frac{\partial y_t}{\partial w} = \frac{\partial y_t}{\partial h_t} \left( \frac{\partial h_t}{\partial w} + \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial w} + \dots \right) =$$

$$= \frac{\partial y_t}{\partial h_t} \sum_{k=0}^t \left( \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial w}$$

$$\left| \left| \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} \right| \right| = ?$$



# Методы лечения

## Решение проблемы «Exploding gradients»

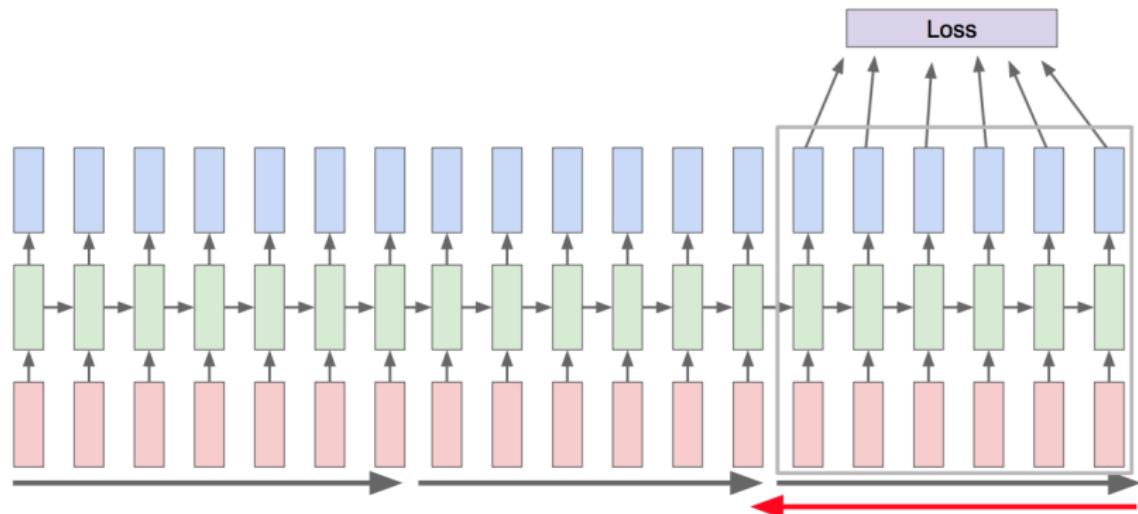
- ▶ Регуляризация
- ▶ Обрезка градиентов (Clipping gradients)
- ▶ Метод форсирования учителя (Teacher Forcing)
- ▶ Ограничение шагов обратного распространения (Truncated Backpropagation Through Time)

# Методы лечения

## Решение проблемы «Vanishing gradients»

- ▶ Специальные блоки (Gated self-loops: LSTM, GRU)
- ▶ Использование методов оптимизации с Гессианом
- ▶ Leaky Integration Units  $h_t = \alpha h_{t-1} + (1 - \alpha) h_t^{RNN}$
- ▶ Специальная регуляризация (Vanishing Gradient Regularization / Gradient propagation regularizer)
- ▶ Инициализация (например, ортогональная)

# Truncated Backpropagation through time



# Backpropagation through time. Residual connections

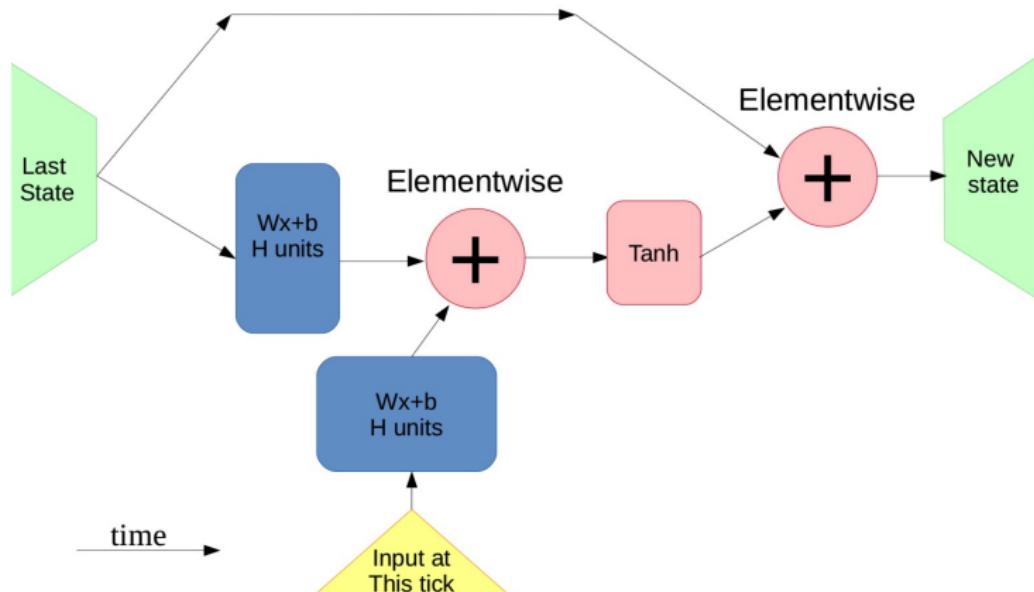


Рис.: RNN cell with residual<sup>12</sup>

<sup>12</sup>[https://github.com/yandexdataschool/Practical\\_DL](https://github.com/yandexdataschool/Practical_DL)

# LSTM, #1

Память в RNN:

- ▶ long-term memory: параметры/веса сети в процессе обучения медленно изменяются, кодируя общие знания о предметной области
- ▶ short-term memory: проявляется в процессе прохода сигнала по рекуррентным слоям
- ▶ Long Short-Term Memory<sup>13</sup> промежуточный способ памяти, проявляется в специальной конструкции нейронов с памятью

Так же, LSTM - это способ борьбы с ростом и затуханием градиентов.

---

<sup>13</sup>LONG SHORT-TERM MEMORY, Hochreiter & Schmidhuber  
[http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97\\_lstm.pdf](http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf)

## LSTM, #2

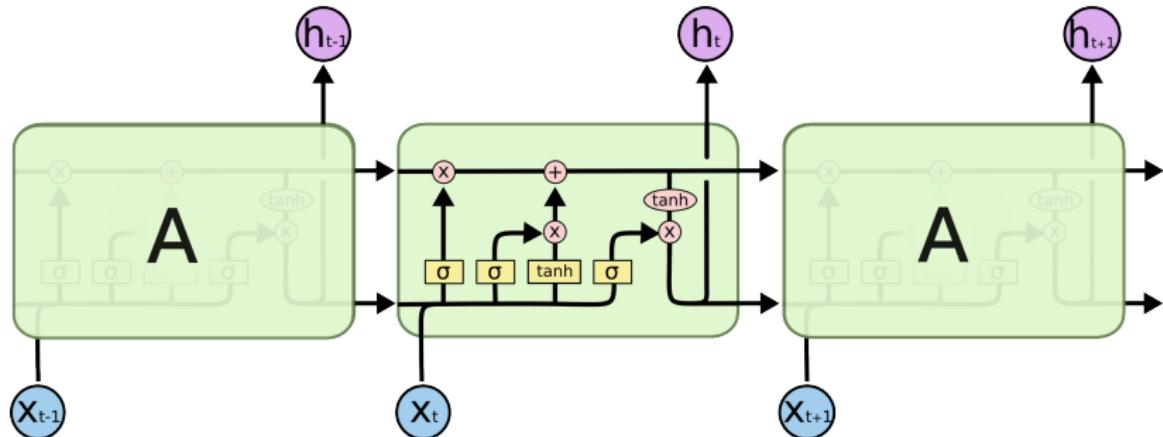
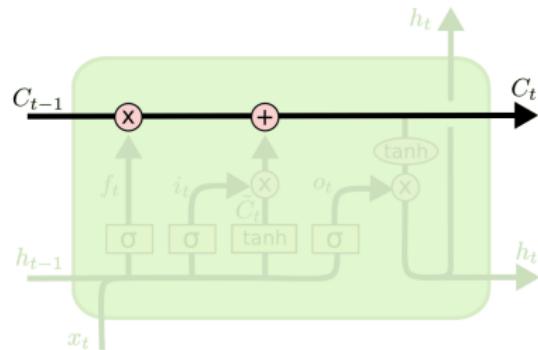


Рис.: Оригинальная модель memory unit версии 1997 года<sup>14</sup>; Так же как и в простых рекуррентных сетях, у нас есть скрытый слой с циклической связью.

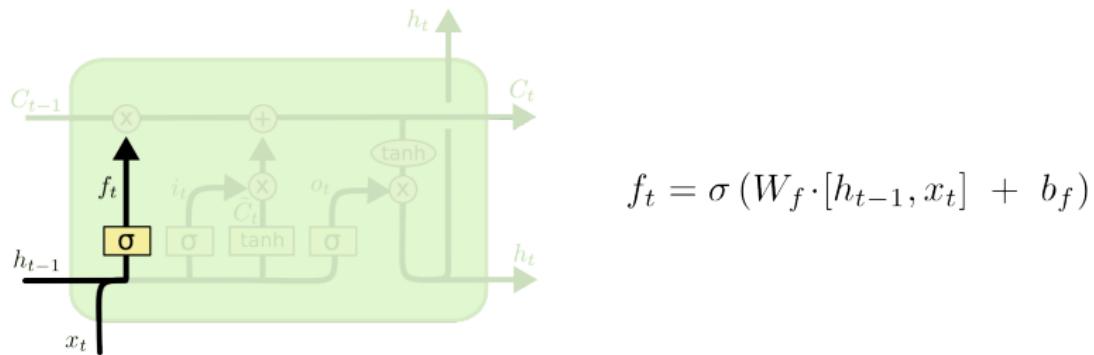
<sup>14</sup><http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

## LSTM, #3. "Конвейер" состояний



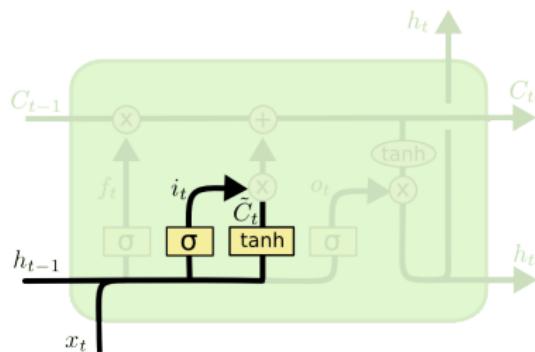
- ▶ Важная составляющая LSTM - слой состояния сети  $C_t$ . LSTM может как добавлять новую информацию к состояниям, так и стирать старую.

## LSTM, #4. Забывающий слой



- ▶ Forget gate layer  $f_t$  с помощью сигмоидальной функции смотрит на значения  $x_t$  и  $h_t$  и выдает для каждого числа в  $C_{t-1}$  число от 0 (полностью забыть) до 1 (полностью сохранить).

# LSTM, #5

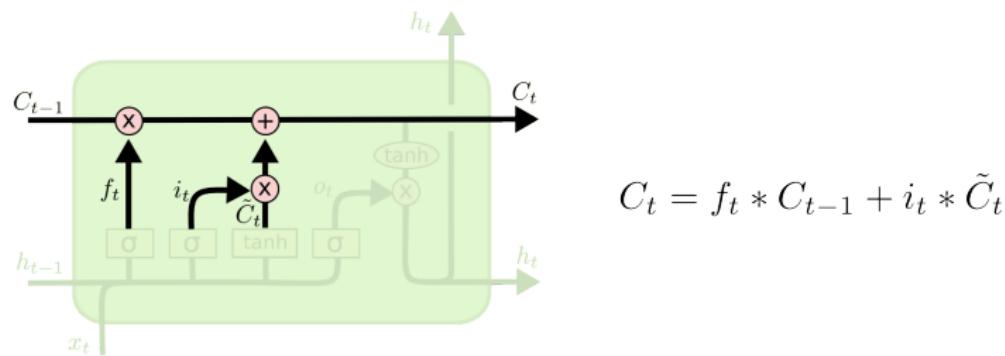


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

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

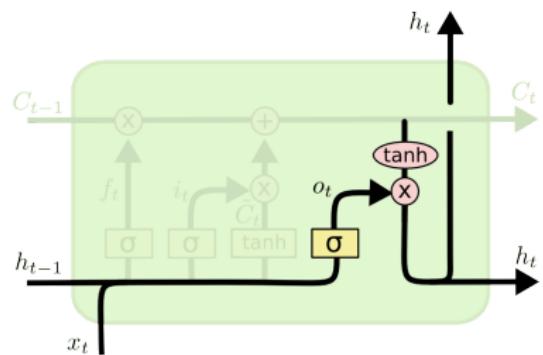
- ▶ Операция  $i_t$  играет роль входного слоя ("input gate layer"), решая какие значения обновлять. Как и в забывающем слое при  $i_t = 0$  ничего не передается, при  $i_t = 1$  передается все.
- ▶ Далее с помощью  $\tanh$  вычисляются значения-"**кандидаты**" новых состояний.

# LSTM, #6



- Обновляем вектор состояний

## LSTM, #7. Слой выхода

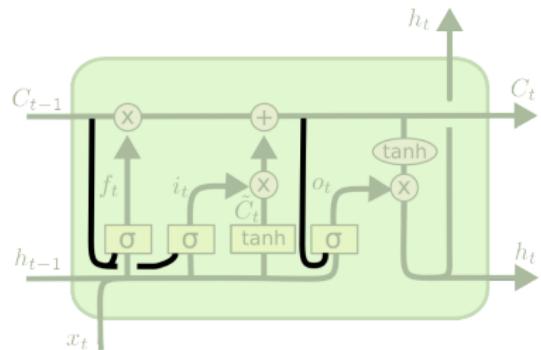


$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$

- ▶ Наконец, решаем, что нам нужно вывести на данном шаге.

# LSTM, #8. Варианты



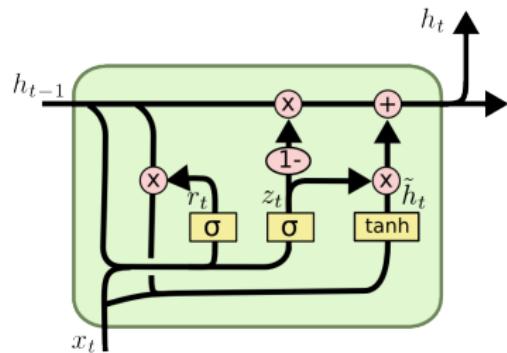
$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i)$$
$$o_t = \sigma(W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$

Рис.: “Peephole connections”<sup>15</sup>

- Даем возможность всем слоям смотреть на вектор состояний.

<sup>15</sup><ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf>

# LSTM, #9. Варианты



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

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

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

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

Рис.: Gated Recurrent Unit<sup>16</sup>

- Объединяем забывающий и входной слои в "слой обновления".  
Объединяем состояния и скрытый слой.

<sup>16</sup><http://arxiv.org/pdf/1406.1078v3.pdf>

# LSTM, #10

- ▶ визуализация работы LSTM - <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

# BRNN

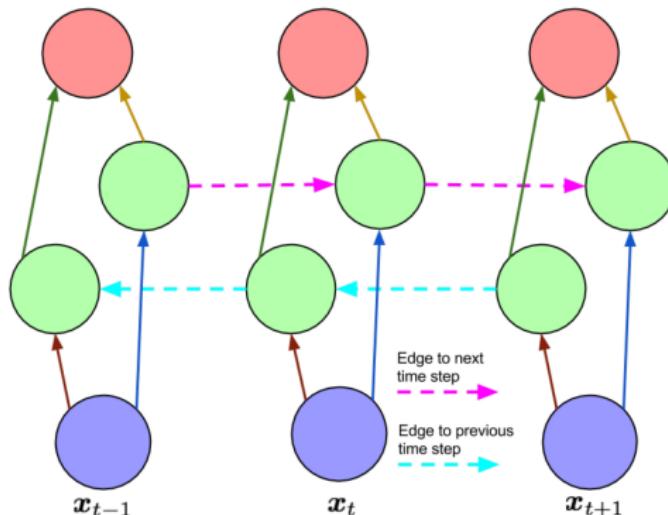


Рис.: Recurrent layer <sup>17</sup>

- ▶  $h^{(t)} = \sigma (W_{hx}x^{(t)} + W_{hh}h^{(t-1)} + b_h)$
- ▶  $z^{(t)} = \sigma (W_{zx}x^{(t)} + W_{zz}z^{(t-1)} + b_z)$
- ▶  $\hat{y}^{(t)} = \text{softmax} (W_{yh}h^{(t)} + W_{yz}z^{(t)} + b_y)$

<sup>17</sup> Bidirectional recurrent neural networks

<http://www.di.ufpe.br/~fnj/RNA/bibliografia/BRNN.pdf>

# Deep multimodal learning, #1

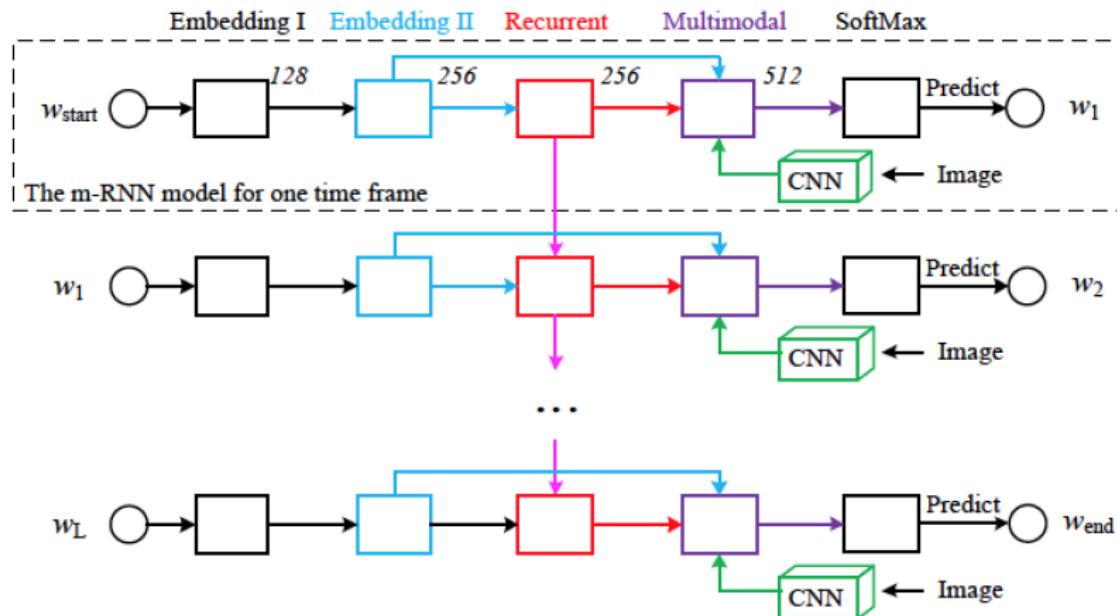


Рис.: DEEP CAPTIONING WITH MULTIMODAL RECURRENT NEURAL NETWORKS<sup>18</sup>

<sup>18</sup>[http://www.stat.ucla.edu/~yuille/Pubs14\\_15/MaoXuYangWangYuilleNIPS2014.pdf](http://www.stat.ucla.edu/~yuille/Pubs14_15/MaoXuYangWangYuilleNIPS2014.pdf)

## Deep multimodal learning, #2



1. Top view of the lights of a city at night, with a well-illuminated square in front of a church in the foreground;
  2. People on the stairs in front of an illuminated cathedral with two towers at night;
- 

A square with burning street lamps and a street in the foreground;

Рис.: Пример генерации описания<sup>19</sup>

---

<sup>19</sup>[http://www.cs.tufts.edu/~nville/Pubs14\\_15/MaoXuYangXiangXuvilleNIPS2014.pdf](http://www.cs.tufts.edu/~nville/Pubs14_15/MaoXuYangXiangXuvilleNIPS2014.pdf)

# Deep multimodal learning, #3



1. Tourists are sitting at a long table with beer bottles on it in a rather dark restaurant and are raising their bierglaeser;
  2. Tourists are sitting at a long table with a white table-cloth in a somewhat dark restaurant;
- 

Tourists are sitting at a long table with a white table cloth and are eating;

Рис.: Пример генерации описания<sup>20</sup>

---

<sup>20</sup>[http://www.stat.ucla.edu/~yuille/Pubs14\\_15/MaoXuYangWangYuilleNIPS2014.pdf](http://www.stat.ucla.edu/~yuille/Pubs14_15/MaoXuYangWangYuilleNIPS2014.pdf)

# Deep multimodal learning, #4



1. A dry landscape with light brown grass and green shrubs and trees in the foreground and large reddish-brown rocks and a blue sky in the background;
2. A few bushes at the bottom and a clear sky in the background;

---

A dry landscape with green trees and bushes and light brown grass in the foreground and reddish-brown round rock domes and a blue sky in the background;

Рис.: Пример генерации описания<sup>21</sup>

---

<sup>21</sup>[http://www.stat.ucla.edu/~yuille/Pubs14\\_15/MaoXuYangWangYuilleNIPS2014.pdf](http://www.stat.ucla.edu/~yuille/Pubs14_15/MaoXuYangWangYuilleNIPS2014.pdf)

# Deep multimodal learning, #5



1. Group picture of nine tourists and one local on a grey rock with a lake in the background;
  2. Five people are standing and four are squatting on a brown rock in the foreground;
- 

A blue sky in the background;

Рис.: Пример генерации описания<sup>22</sup>

---

<sup>22</sup>[http://www.stat.ucla.edu/~yuille/Pubs14\\_15/MaoXuYangWangYuilleNIPS2014.pdf](http://www.stat.ucla.edu/~yuille/Pubs14_15/MaoXuYangWangYuilleNIPS2014.pdf)

## RNN with memory

- ▶ Facebook: Memory Networks<sup>23</sup>
- ▶ Google: Neural Turing Machine<sup>24</sup>

Bilbo travelled to the cave. Gollum dropped the ring there. Bilbo took the ring. Bilbo went back to the Shire. Bilbo left the ring there. Frodo got the ring. Frodo journeyed to Mount-Doom. Frodo dropped the ring there. Sauron died. Frodo went back to the Shire. Bilbo travelled to the Grey-havens. The End.

- ▶ Where is the ring? A: Mount-Doom
- ▶ Where is Bilbo now? A: Grey-havens
- ▶ Where is Frodo now? A: Shire
- ▶ Google: Differentiable neural computers <sup>25</sup>

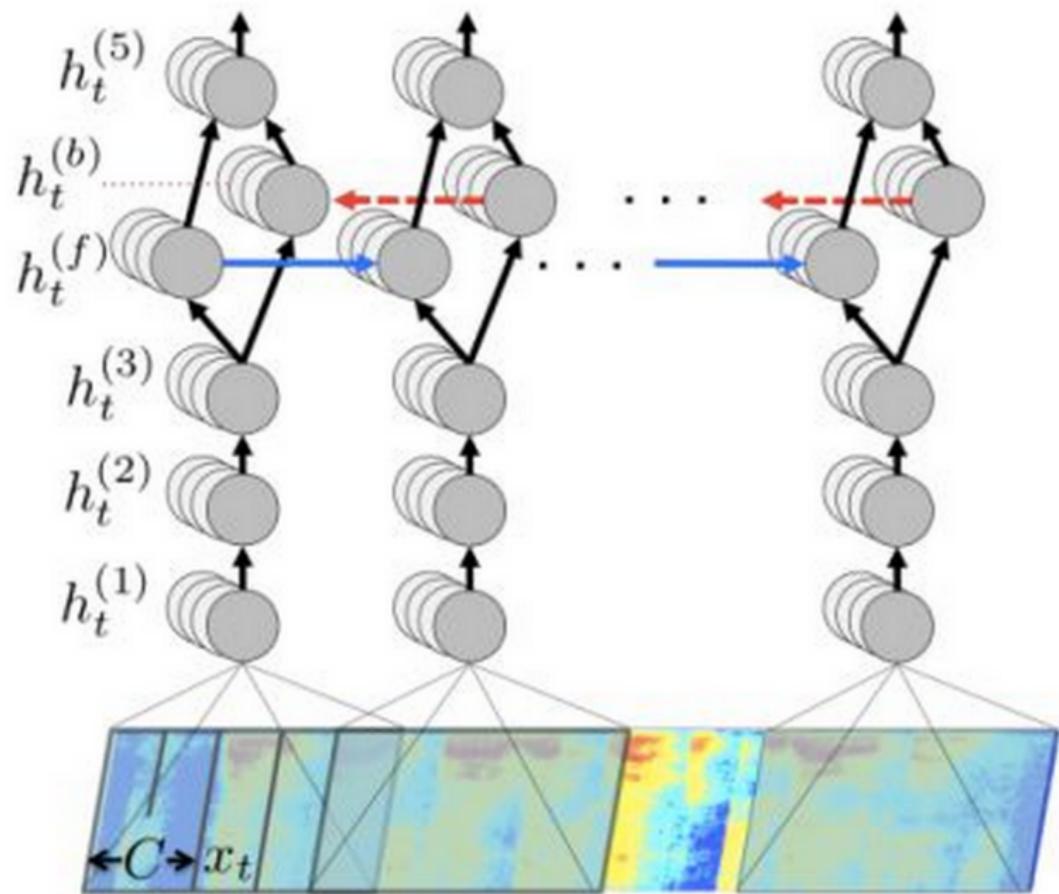
---

<sup>23</sup><http://arxiv.org/pdf/1410.3916v10.pdf>

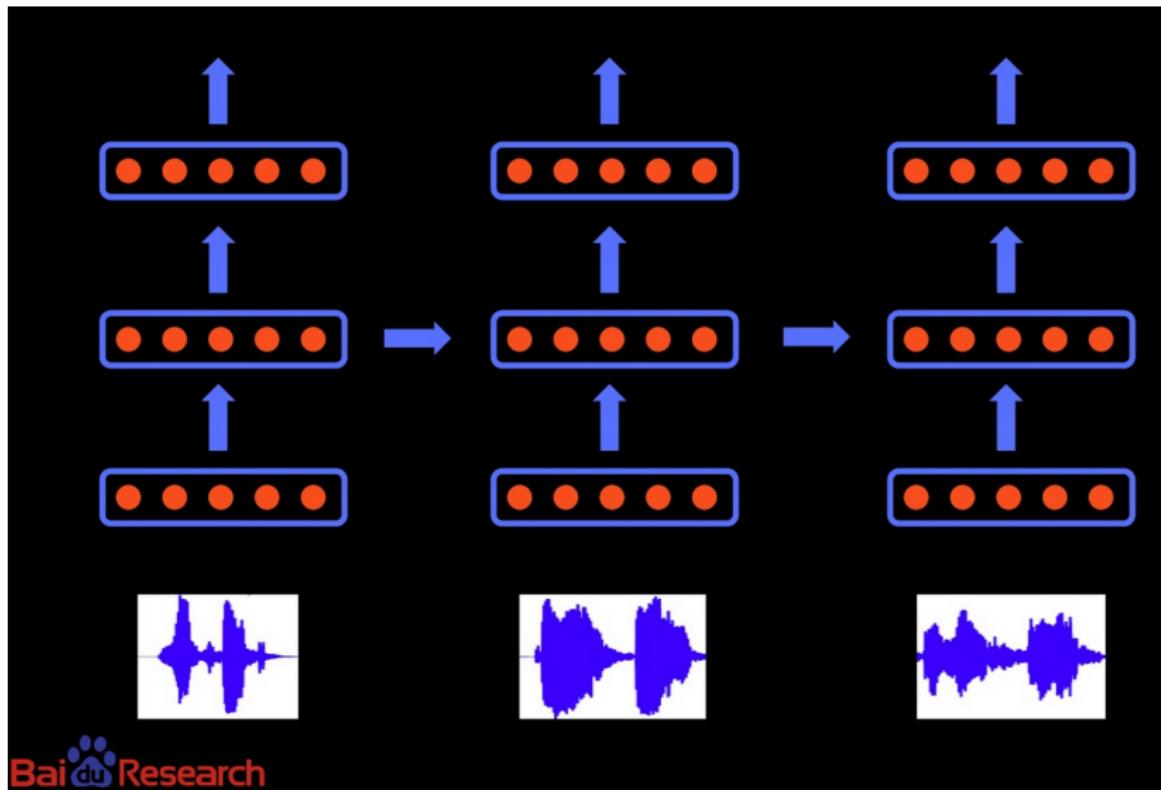
<sup>24</sup><http://arxiv.org/pdf/1410.5401v2.pdf>

<sup>25</sup><https://deepmind.com/blog/differentiable-neural-computers/>

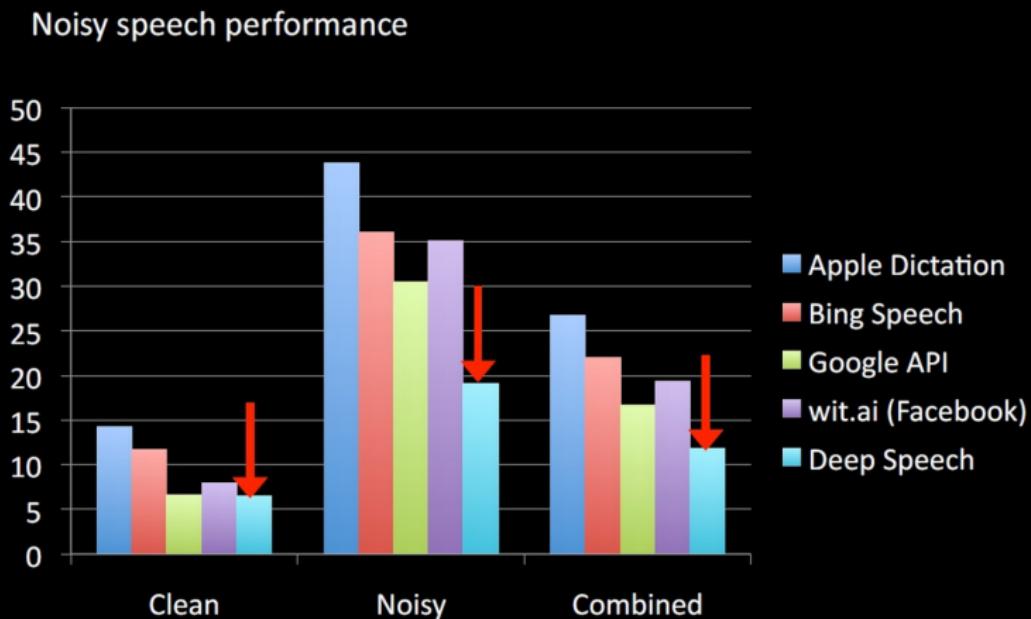
# Baidu Deep Speech, #1



# Baidu Deep Speech, #2



# Baidu Deep Speech, #3



# Google WaveNet, #1

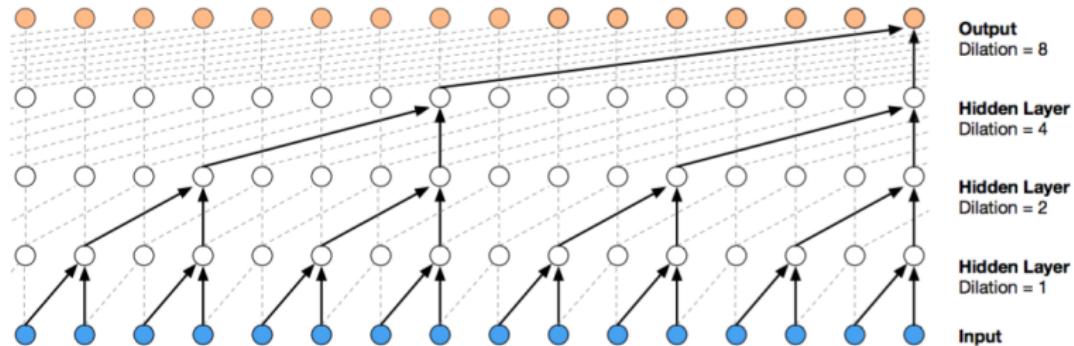


Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

Рис.: WaveNet<sup>26</sup>

---

<sup>26</sup><https://arxiv.org/pdf/1609.03499.pdf>

# Google WaveNet, #2

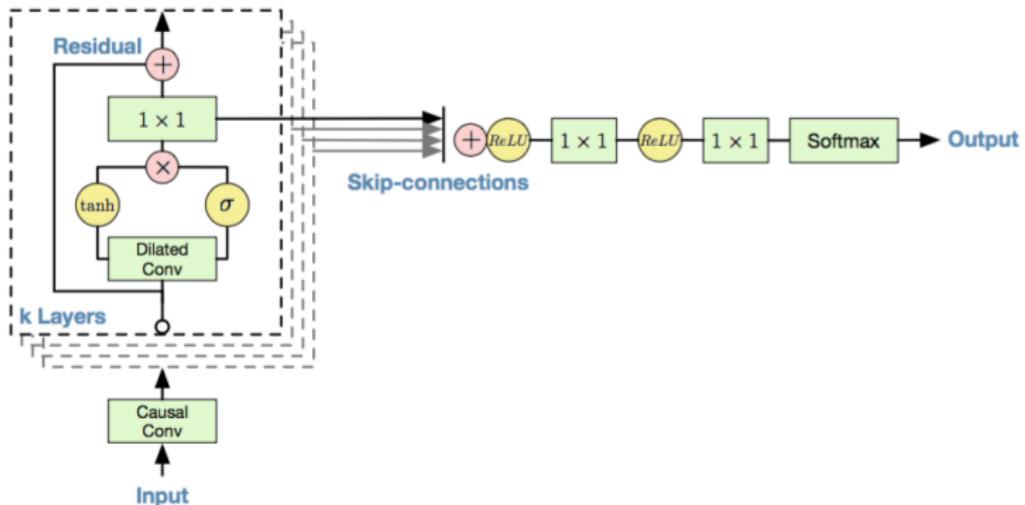


Figure 4: Overview of the residual block and the entire architecture.

## Вопросы

