Языковое моделирование; Работа с последовательностями

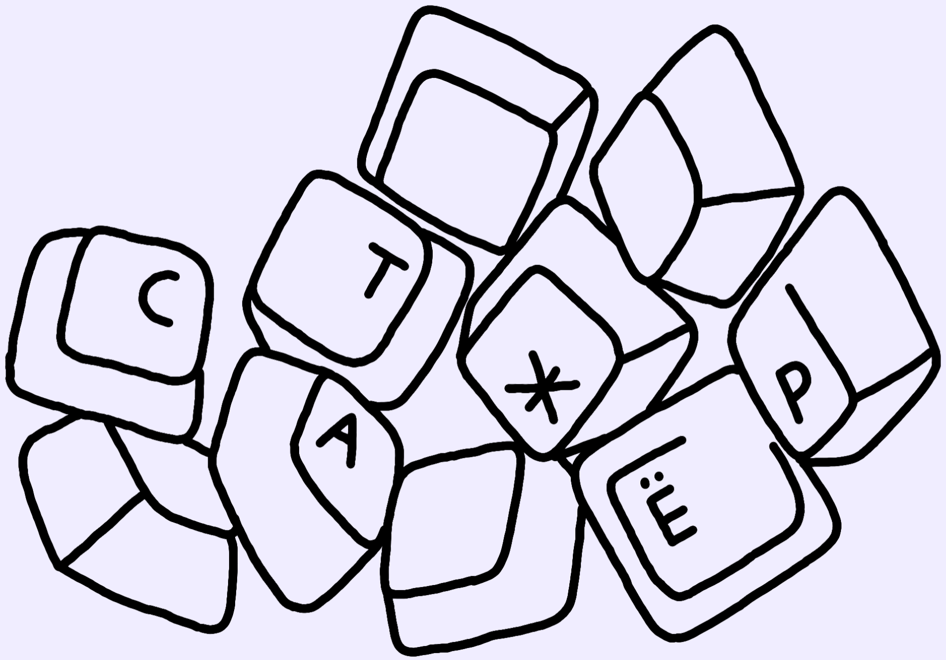
YOUNG&&YANDEX

Радослав Нейчев
Выпускник и преподаватель ШАД и МФТИ,
руководитель группы ML-разработки в Яндексе,
основатель girafe-ai

Содержание

Генеративные модели до ChatGPT

01



Давным-давно, в далёком-далёком 2012

Shakespeare

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Algebraic Geometry (Latex)

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

Linux kernel (source code)

```c
/*
 * If this error is set, we will need anything right after that BSD.
 */
static void action_new_function(struct s_stat_info *wb)
{
  unsigned long flags;
  int lel_idx_bit = e->edd, *sys & ~((unsigned long) *FIRST_COMPAT);
  buf[0] = 0xFFFFFFFF & (bit << 4);
  min(inc, slist->bytes);
  printk(KERN_WARNING "Memory allocated %02x/%02x, "
    "original MLL instead\n"),
    min(min(multi_run - s->len, max) * num_data_in),
    frame_pos, sz + first_seg);
  div_u64_w(val, inb_p);
  spin_unlock(&disk->queue_lock);
  mutex_unlock(&s->sock->mutex);
  mutex_unlock(&func->mutex);
  return disassemble(info->pending_bh);
}
```


source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Давным-давно, в далёком-далёком 2012

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

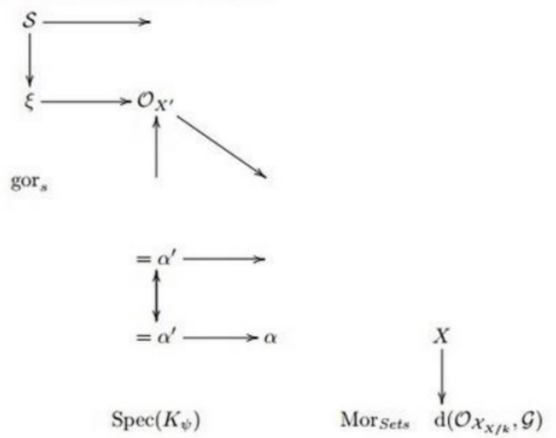
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \ \ \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Давным-давно, в далёком-далёком 2012

```
#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setew.h>
#include <asm/pgproto.h>

#define REG_PG    vesa_slot_addr_pack
#define PFM_NOCOMP  AFSR(0, load)
#define STACK_DDR(type)     (func)

#define SWAP_ALLOCATE(nr)     (e)
#define emulate_sigs()  arch_get_unaligned_child()
#define access_rw(TST)  asm volatile("movd %%esp, %0, %3" : : "r" (0));   \
  if (__type & DO_READ)

static void stat_PC_SEC __read_mostly offsetof(struct seq_argsqueue, \
          pC>[1]);

static void
os_prefix(unsigned long sys)
{
#ifdef CONFIG_PREEMPT
  PUT_PARAM_RAID(2, sel) = get_state_state();
  set_pid_sum((unsigned long)state, current_state_str(),
           (unsigned long)-1->lr_full; low;
}
```


source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Обработка последовательностей

02

Последовательности формально

Последовательность объектов/событий: $x_1, x_2, \ldots, x_t$

Каждый объект – вектор: $x_i \in \mathbb{R}$

Задача: научиться обрабатывать последовательности переменной длины;

Последовательности всюду: тексты, видео, действия пользователей, действия агентов и пр.

Марковское свойство

Последовательность $x_1, x_2, \ldots, x_t$ обладает марковским свойством:

$$P(x_{t+1}|x_1, x_2, \ldots, x_t) = P(x_{t+1}|x_t)$$

Здесь мы предполагаем, что для каждого элемента задано вероятностное распределение (формально говоря, мы рассматриваем случайный процесс).

Т.е. в марковском процессе играет роль лишь фиксированная часть истории.

Как работать с последовательностью переменной длины?

Глобальные статистики

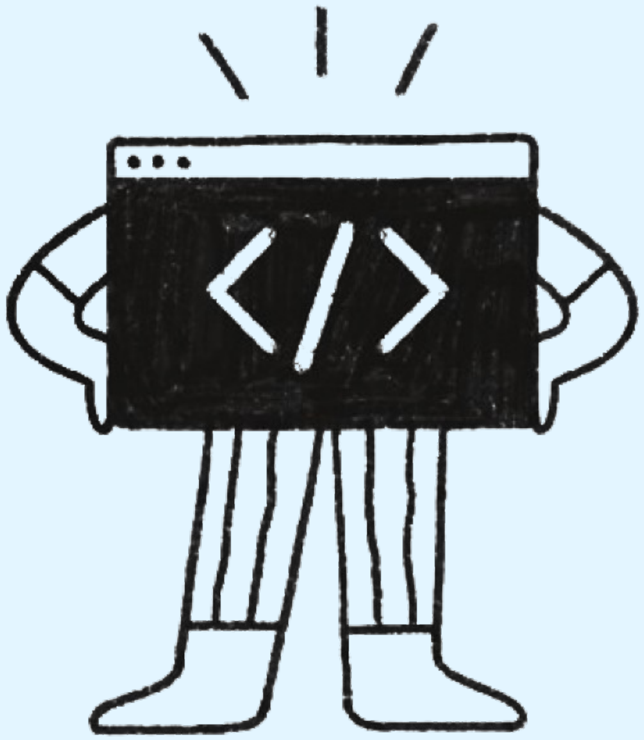
$$z = \frac{1}{T}\sum_{i=1}^{T} x_i$$

Пошаговая обработка

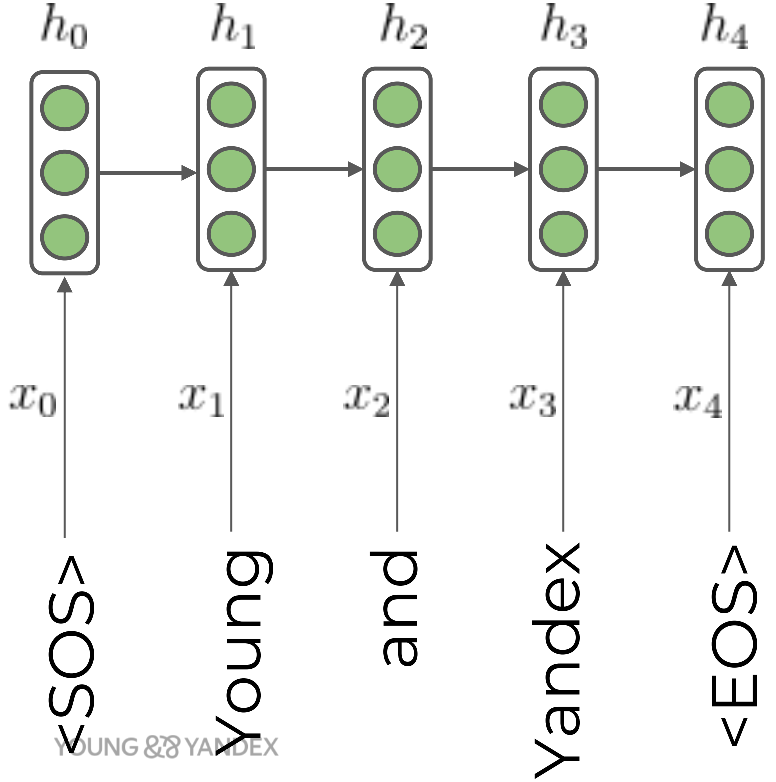
$$\boxed{h_{t+1} = f_\theta(h_t, x_t)}$$

Рекуррентный блок
RNN

03



Кодирование последовательностей

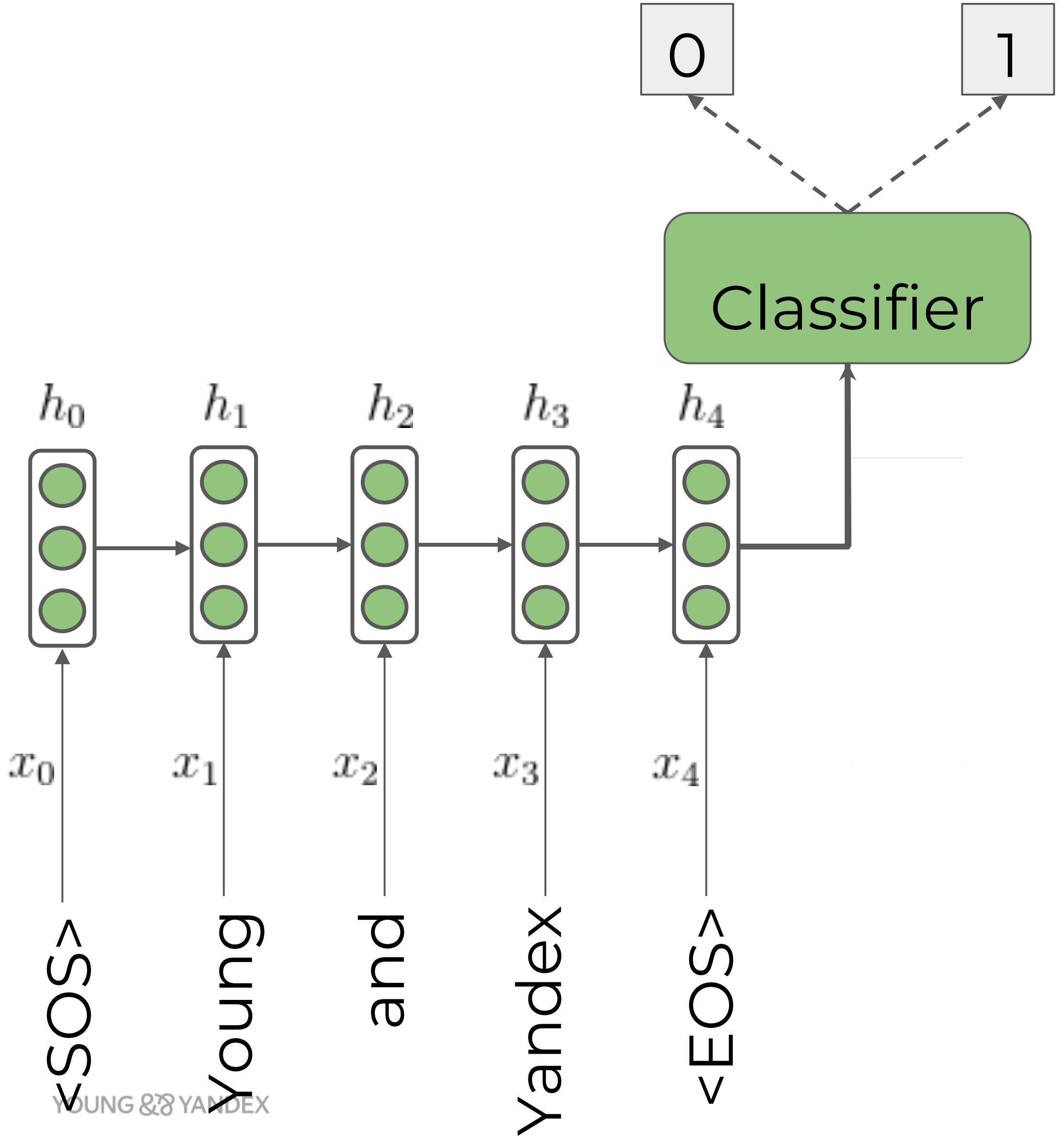


Если бы могли закодировать текущий контекст в виде вектора h_t и обогащать его новой информацией итеративным образом…

Что же за функция ниже?

$$f_\theta(h_t, x_t)$$

Классификация последовательностей

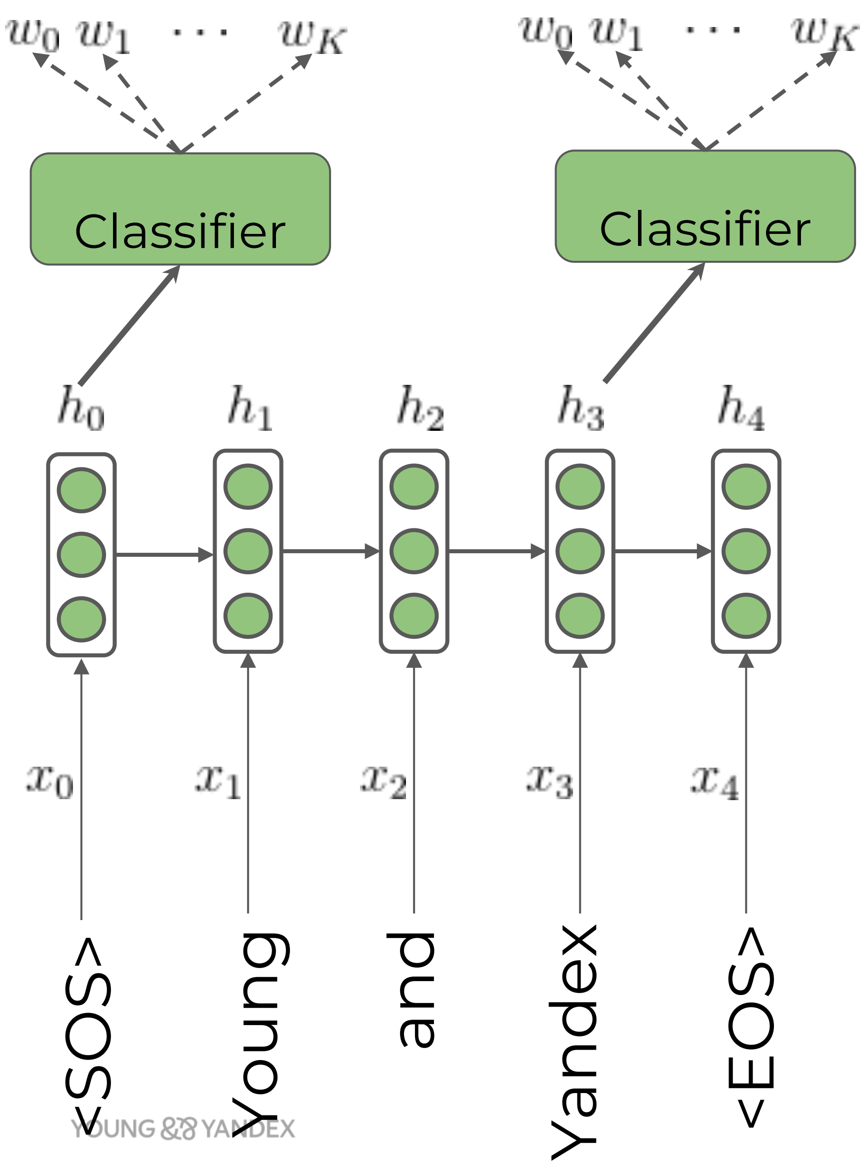


Тогда вектор h_t выступал бы эмбеддингом всего левого контекста.

Мы могли бы решать задачу классификации последовательностей.

$$\hat{y} \sim P_\phi(y|h_t)$$

Генерация последовательностей

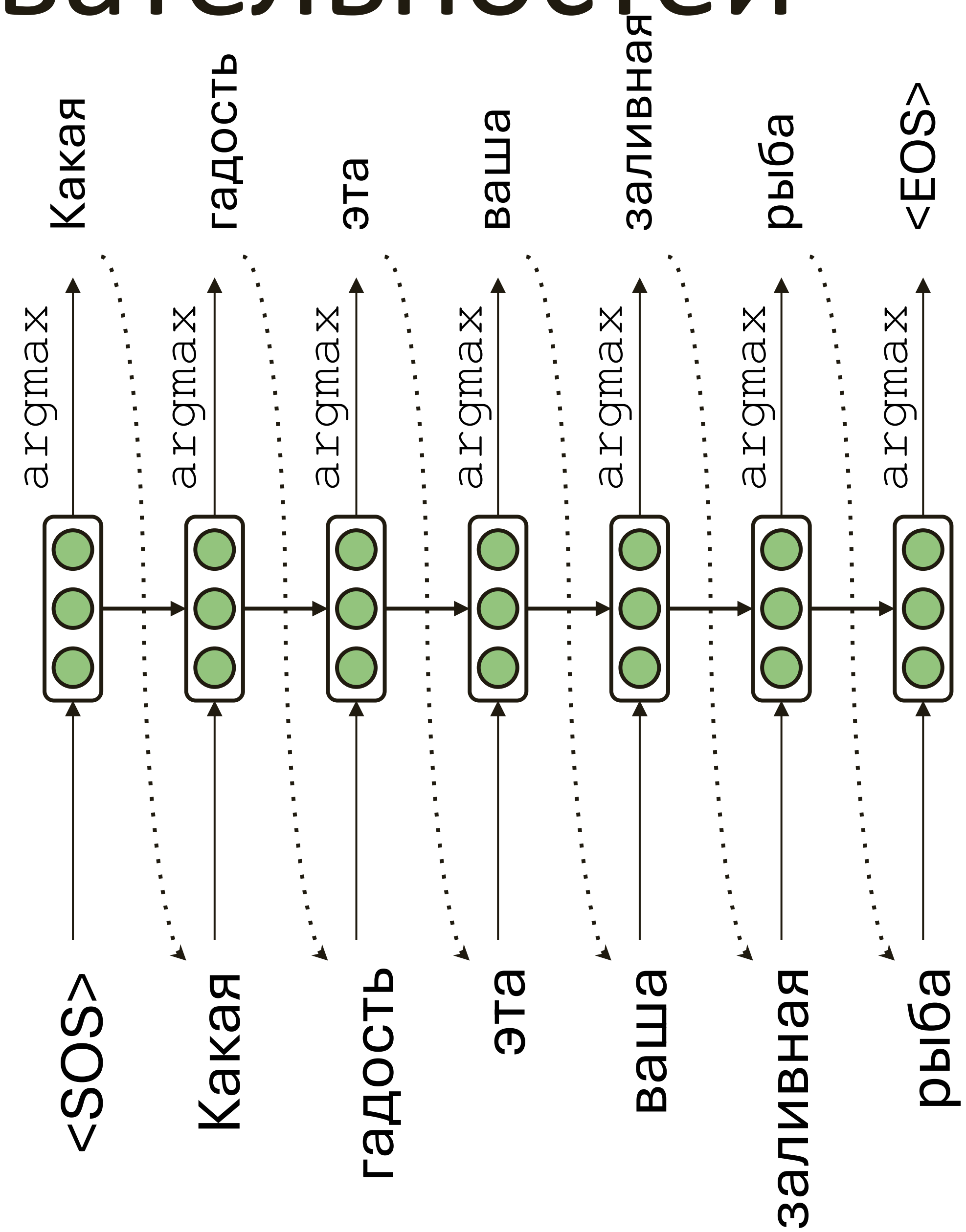


Тогда вектор h_t выступал бы эмбеддингом всего левого контекста.

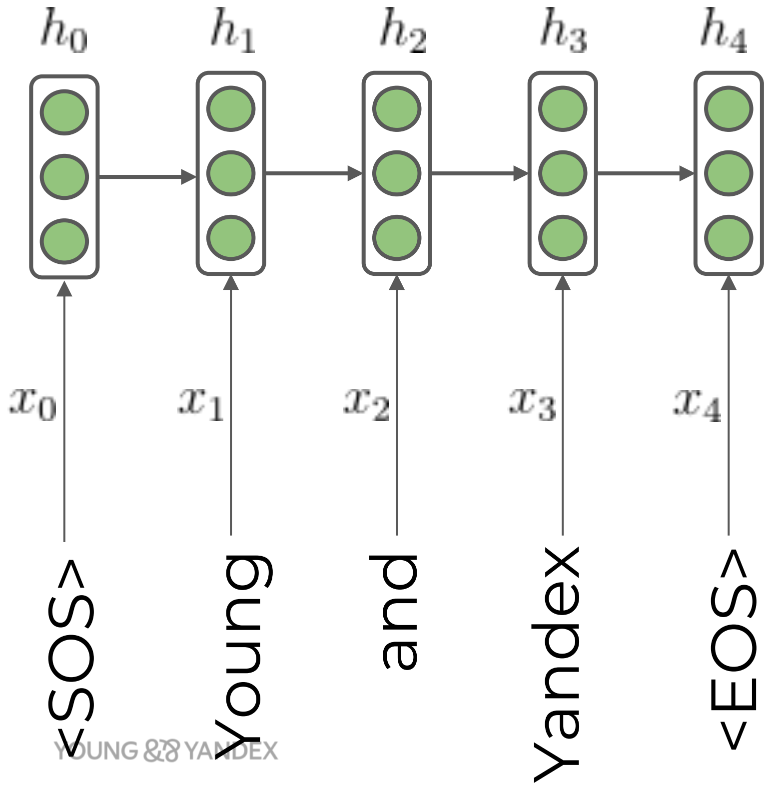
Мы могли бы решать задачу генерации последовательностей, предсказывая следующий токен на каждом шаге:

$$\hat{x}_{t+1} \sim P_\gamma(w|h_t)$$

Генерация последовательностей



Кодирование последовательностей

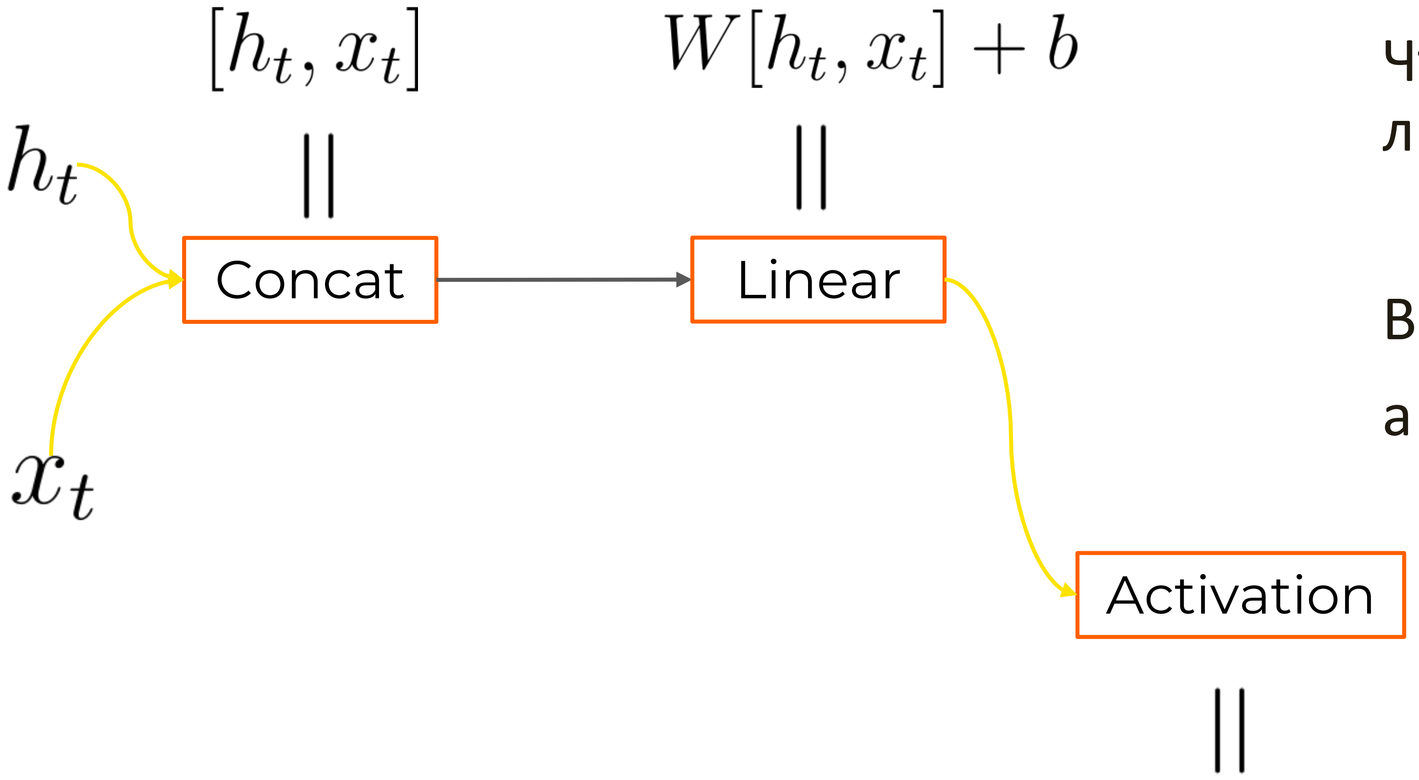


Если бы могли закодировать текущий контекст в виде вектора h_t и обогащать его новой информацией итеративным образом…

Что же за функция ниже?

$$f_\theta(h_t, x_t)$$

Рекурретный блок – основа RNN



Что может быть проще одного линейного слоя?

В данном случае $\theta = \{W, b\}$,

а [...] означает конкатенацию.

$$f_\theta(h_t, x_t) = \sigma(W[h_t, x_t] + b)$$

RNN в виде формул:

$$h_0 = \overline{0}$$

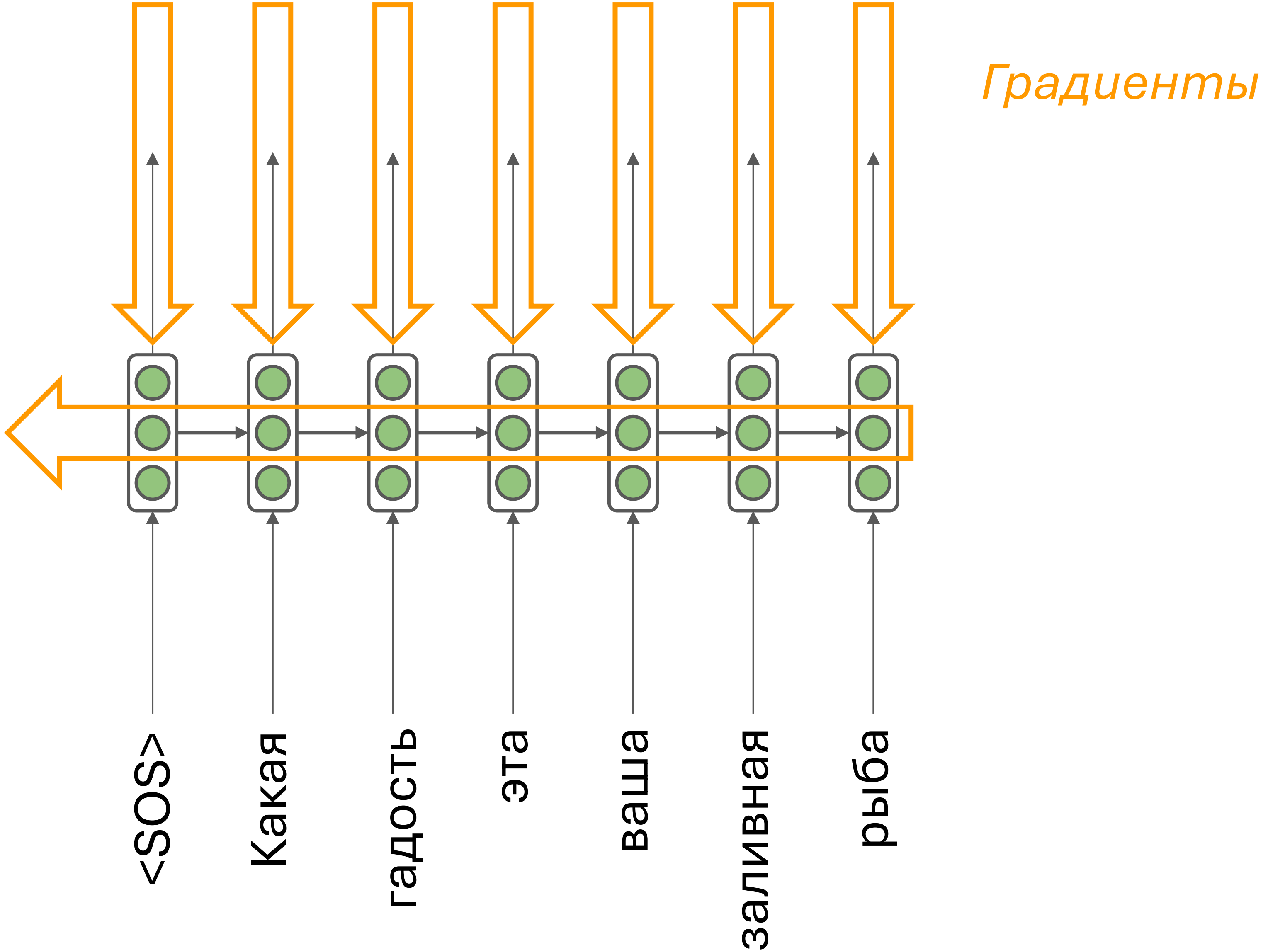
$$h_1 = \sigma\left(W[h_0, x_0] + b\right)$$

$$h_2 = \sigma\left(W[h_1, x_1] + b\right) = \sigma\left(W[\sigma\left(W_{\text{hid}}[h_0, x_0] + b, x_1\right)] + b\right)$$

$$h_{i+1} = \sigma\left(W_{\text{hid}}[h_i, x_i] + b\right)$$

$$x_{i+1} \sim P(w|h_i) = \text{softmax}\left(W_{\text{out}} h_i + b_{\text{out}}\right)$$

Как настраивать параметры?



Проблемы с градиентами

04

Взрыв градиента (exploding gradient)

Что если градиент окажется очень большим?

$$\theta^{new} = \theta^{old} - \overbrace{\alpha}^{\text{learning rate}} \underbrace{\nabla_\theta J(\theta)}_{\text{gradient}}$$

Оптимизационный процесс может просто разойтись!

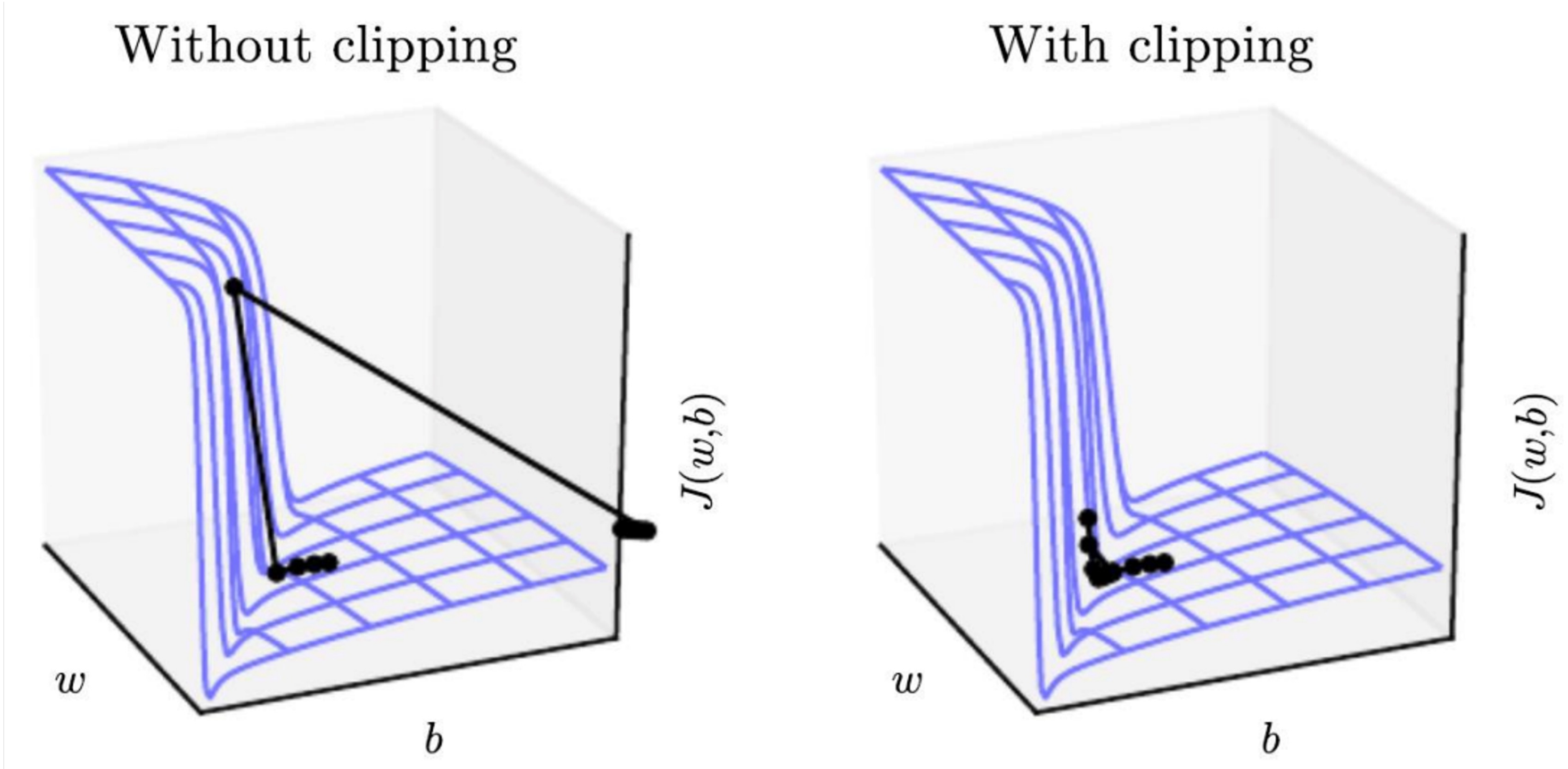
Взрыв градиента (exploding gradient)

Можно нормировать градиент, приводя к наибольшей допустимой норме:

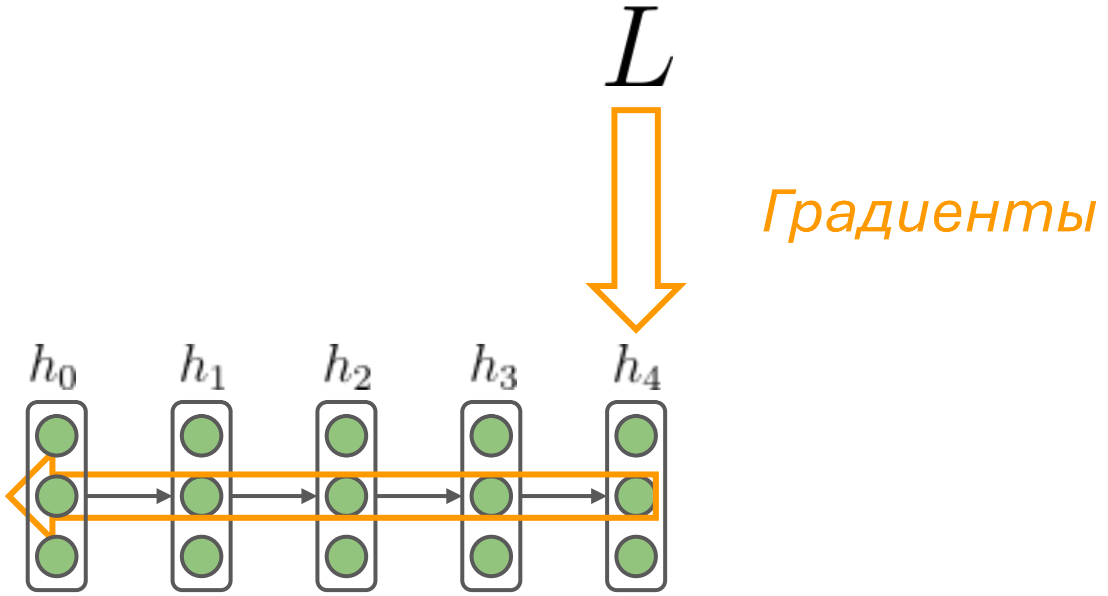
Algorithm 1 Pseudo-code for norm clipping

$\hat{\mathbf{g}} \leftarrow \frac{\partial \mathcal{E}}{\partial \theta}$
if $\|\hat{\mathbf{g}}\| \geq threshold$ **then**
  $\hat{\mathbf{g}} \leftarrow \frac{threshold}{\|\hat{\mathbf{g}}\|}\hat{\mathbf{g}}$
end if

Взрыв градиента (exploding gradient)

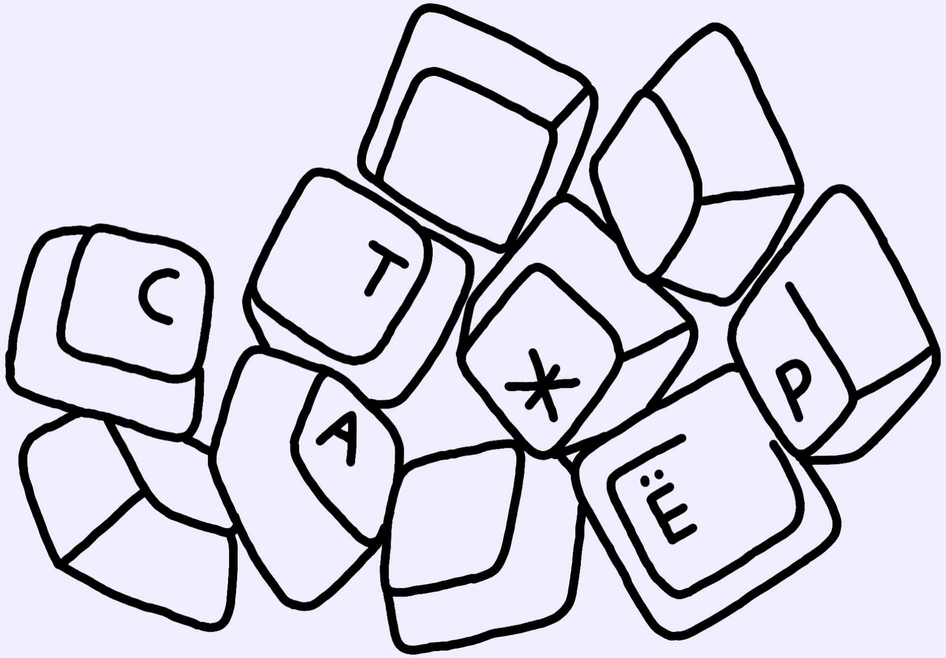


Затухающий градиент (vanishing gradient)



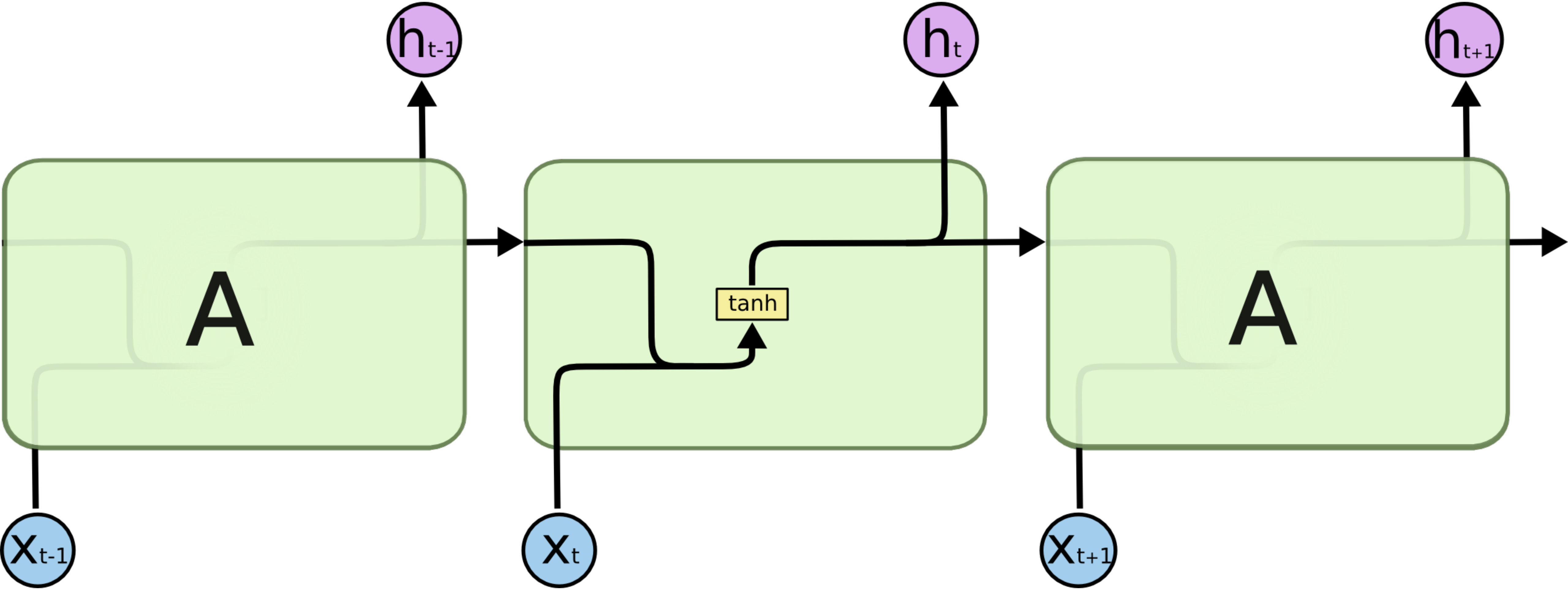
$$\frac{\partial L}{\partial h_0} = \frac{\partial L}{\partial h_4}\frac{\partial h_4}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial h_0}$$

Усложняя RNN – LSTM (как пример)



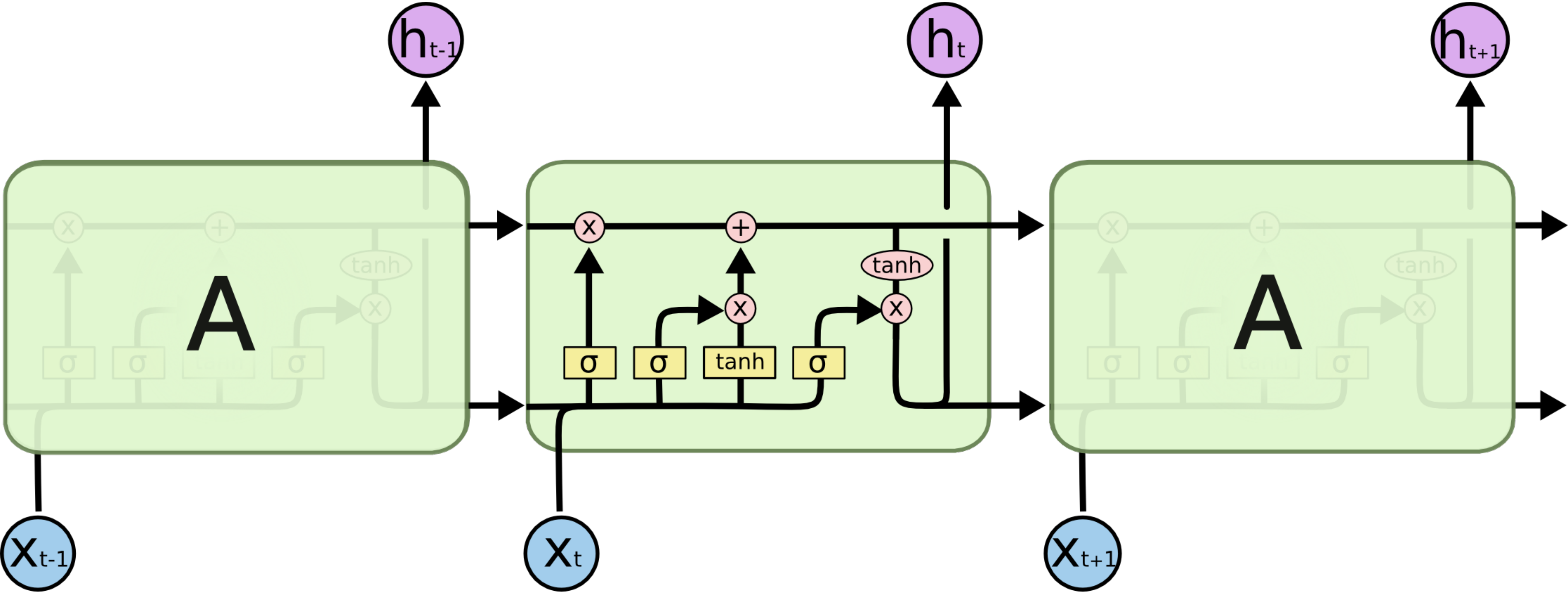
05

Vanilla RNN –> LSTM



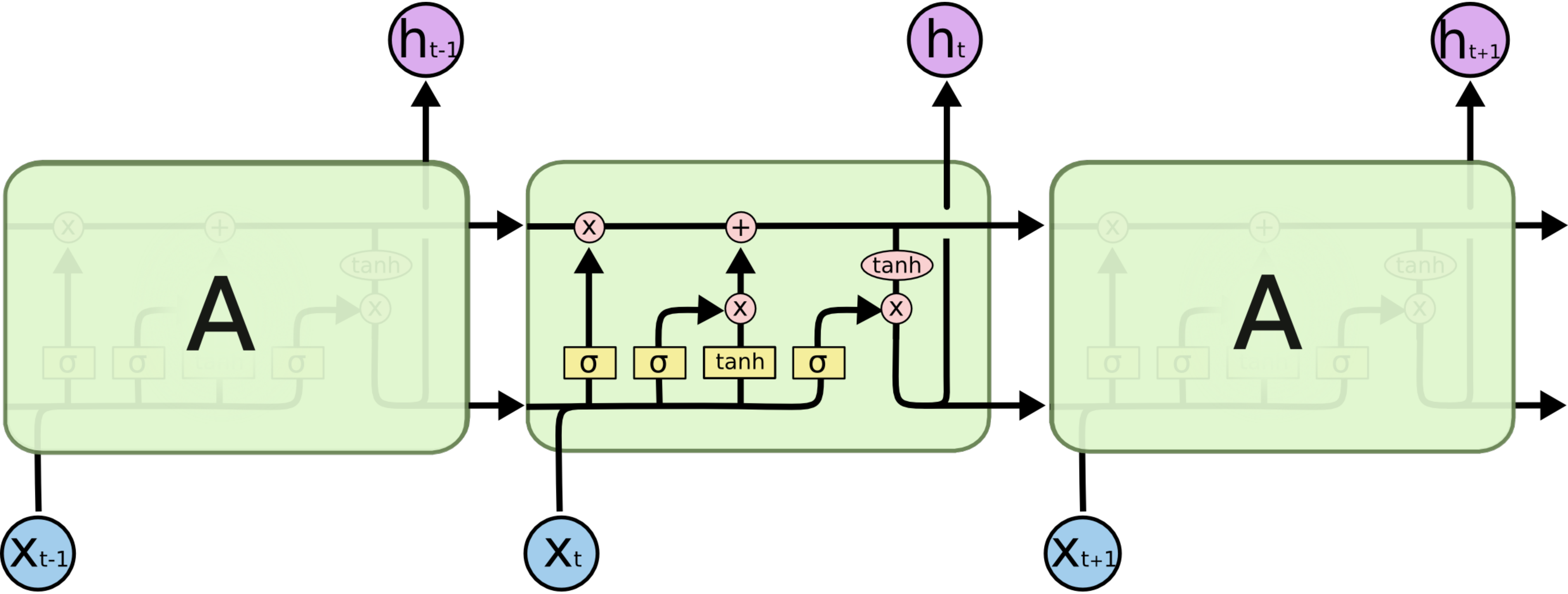
source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Vanilla RNN –> LSTM



source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Vanilla RNN –> LSTM



source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Спасибо за внимание