

# Основы глубинного обучения

## Лекция 8

Задача идентификации. Работа с текстами.

Евгений Соколов

[esokolov@hse.ru](mailto:esokolov@hse.ru)

НИУ ВШЭ, 2024

Идентификация объектов

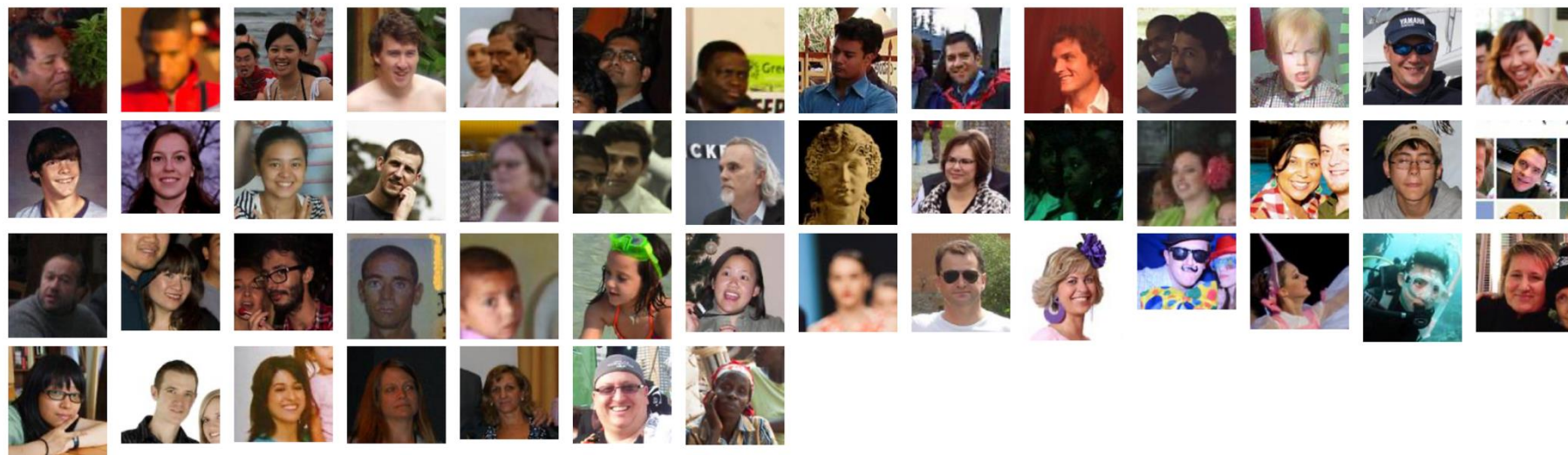
# Labeled Faces in the Wild

- Около 13 тысяч фотографий
- Около 6 тысяч человек

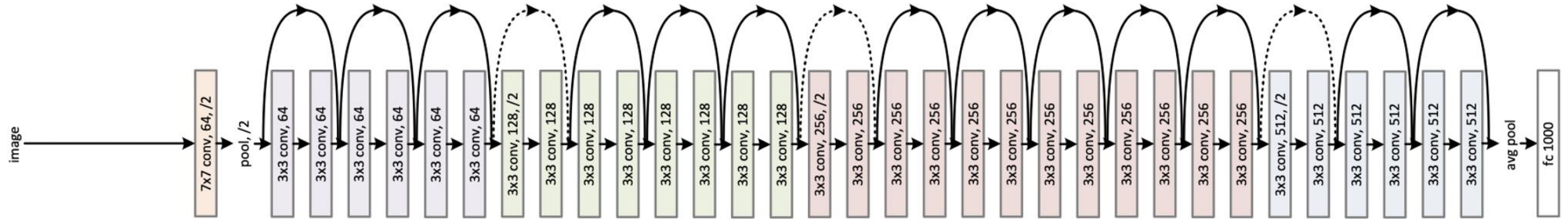


# MegaFace

- 4.7 миллионов фотографий
- Около 700 тысяч человек
- В среднем 7 фото на человека



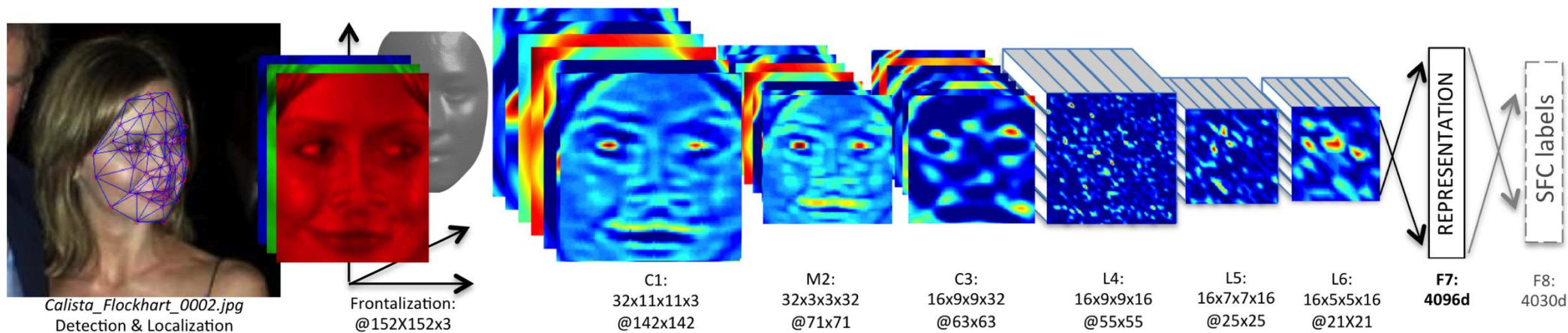
# Дообучение



Если данных совсем мало:

- Берём модель из другой задачи
- Заменяем последний слой на слой с нужным числом выходов
- Обучаем только его
- По сути, это обучение линейной модели

# DeepFace



# DeepFace

- Обучаем некоторую архитектуру для классификации (число классов = число людей в данных)
- Используем выходы предпоследнего слоя как признаковое описание изображения
- Признаки нормализуются (чтобы норма была единичной)
- Считаем близость векторов по какой-нибудь метрике



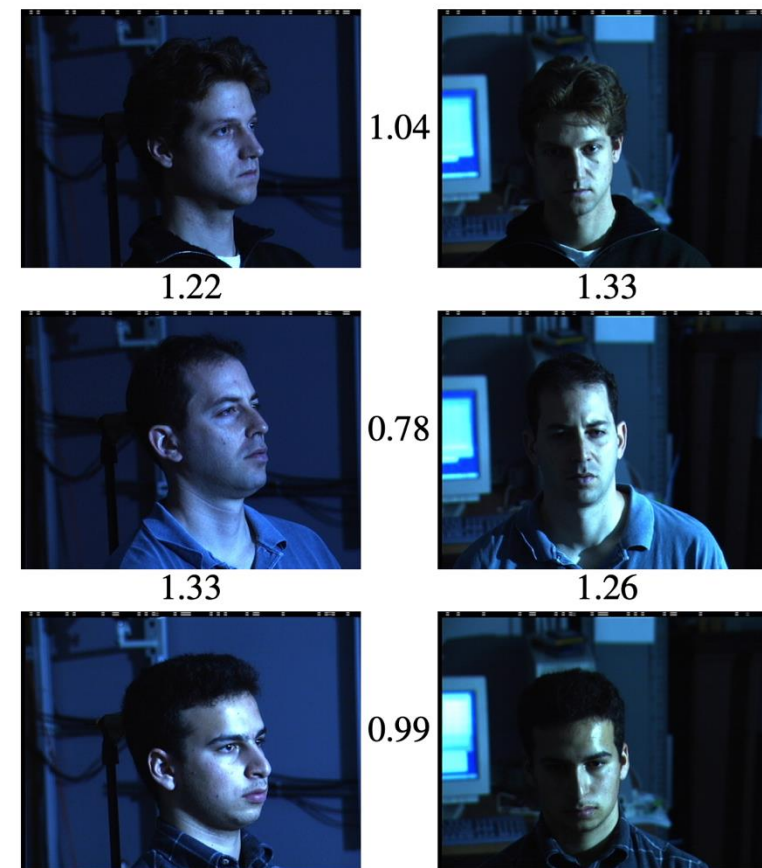
# DeepFace

- Можно сравнить расстояние с порогом, чтобы идентифицировать человека

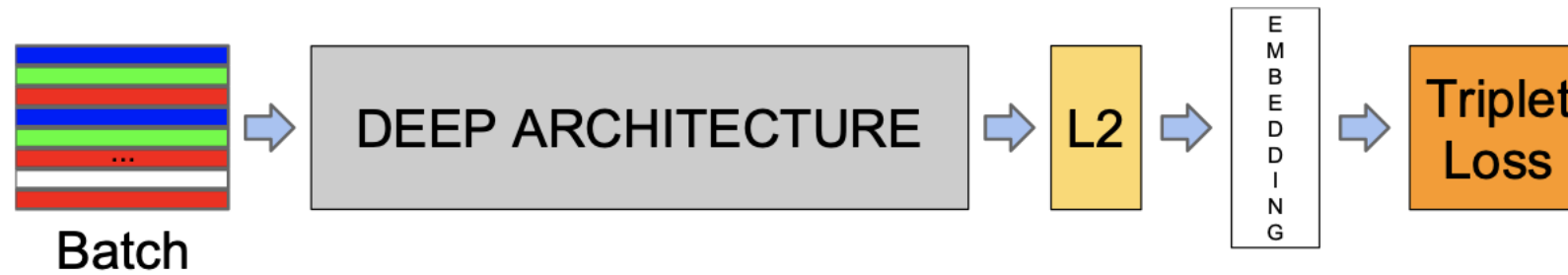


# FaceNet

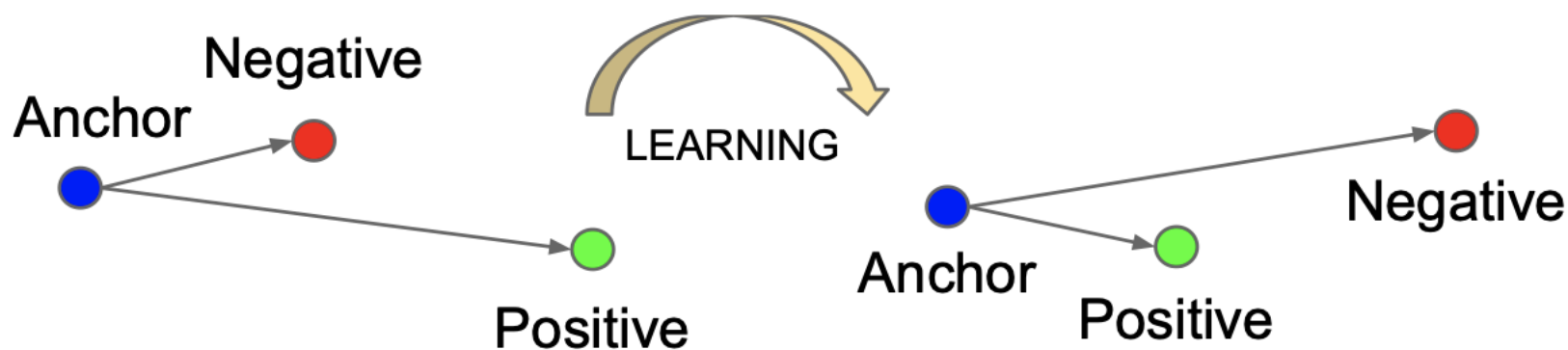
- Почему бы в явном виде не обучать представления изображений так, чтобы фотографии одного человека имели близкие представления?



# FaceNet



# FaceNet



$$\sum_i^N \left[ \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

# FaceNet

- Важно правильно выбирать триплеты
- Обычно: выбираем positive и ищем semi-hard negatives

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2$$

# Триплетная и попарная ошибки

- Попарная ошибка:

$$\sum_{(i,j) \in R} [a(x_i) - a(x_j) < 0] \rightarrow \min$$

- Не совсем про обучение расстояния

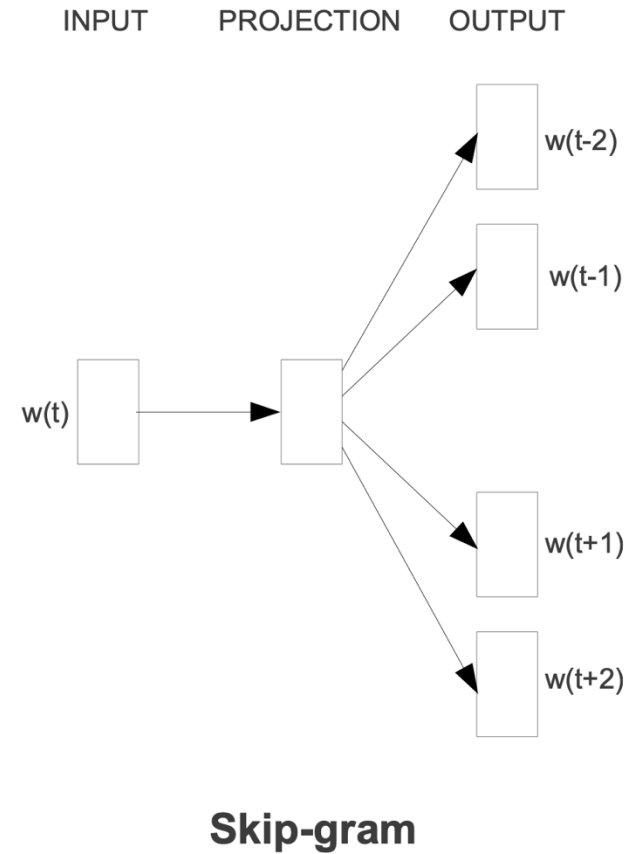
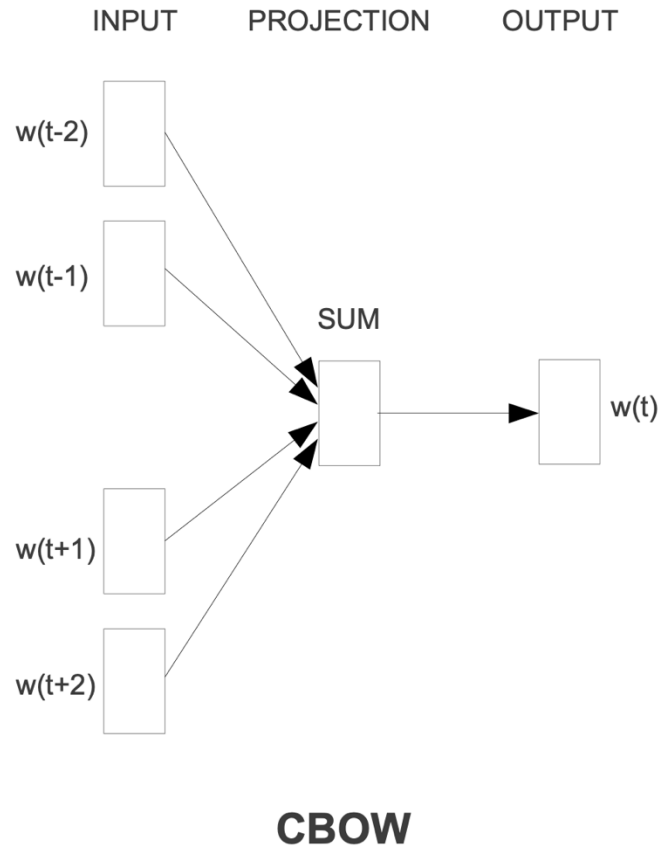
# FaceNet

- Точность на LFW: 99.63%

Векторные представления слов



# word2vec

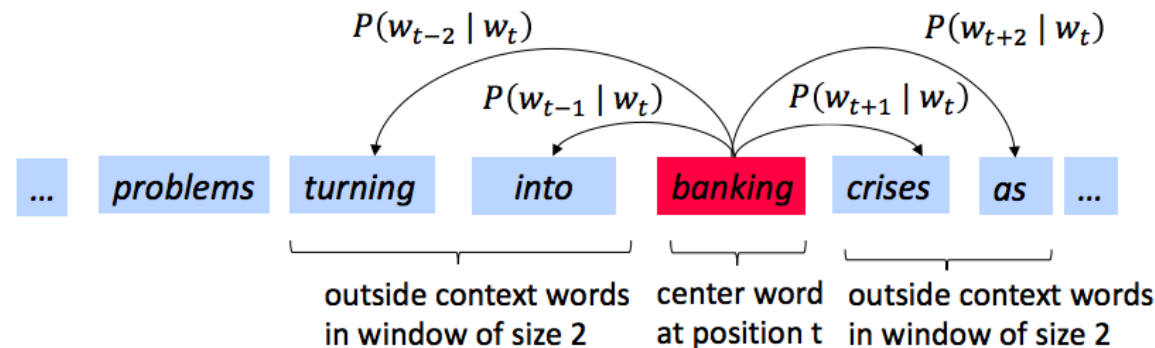


# Skip-gram model

- Вероятность встретить слово  $w_O$  рядом со словом  $w_I$ :

$$p(w_O | w_I) = \frac{\exp(\langle v'_{w_O}, v_{w_I} \rangle)}{\sum_{w \in W} \exp(\langle v'_w, v_{w_I} \rangle)}$$

- $W$  — словарь
- $v_w$  — «центральное» представление слова
- $v'_w$  — «контекстное» представление слова



# Skip-gram model

- Вероятность встретить слово  $w_O$  рядом со словом  $w_I$ :

$$p(w_O|w_I) = \frac{\exp(\langle v'_{w_O}, v_{w_I} \rangle)}{\sum_{w \in W} \exp(\langle v'_w, v_{w_I} \rangle)}$$

- Функционал для текста  $T = (w_1 w_2 \dots w_n)$ :

$$\sum_{i=1}^n \sum_{\substack{-c \leq j \leq c \\ j \neq 0}} \log p(w_{i+j}|w_i) \rightarrow \max$$

# Skip-gram model

- Вероятность встретить слово  $w_O$  рядом со словом  $w_I$ :

$$p(w_O|w_I) = \frac{\exp(\langle v'_{w_O}, v_{w_I} \rangle)}{\sum_{w \in W} \exp(\langle v'_w, v_{w_I} \rangle)}$$

- Считать знаменатель ОЧЕНЬ затратно
- Значит, и производные считать тоже долго

# Negative sampling

$$p(w_O|w_I) = \log \sigma(\langle v'_{w_O}, v_{w_I} \rangle) + \sum_{i=1}^k \log \sigma(-\langle v'_{w_i}, v_{w_I} \rangle)$$

- $w_i$  — случайно выбранные слова
- Слово  $w$  генерируется с вероятностью  $P(w)$  — шумовое распределение
- $P(w) = \frac{U(w)^{\frac{3}{4}}}{\sum_{v \in W} U(v)^{\frac{3}{4}}}$ ,  $U(v)$  — частота слова  $v$  в корпусе текстов

# word2vec: особенности обучения

- Положительные примеры — слова, стоящие рядом
- Отрицательные примеры: подбираем к слову «шум», то есть другое слово, которое не находится рядом
- Важно семплировать в SGD слова с учётом их популярности — иначе будем обучаться только на самые частые слова

# Как это использовать?

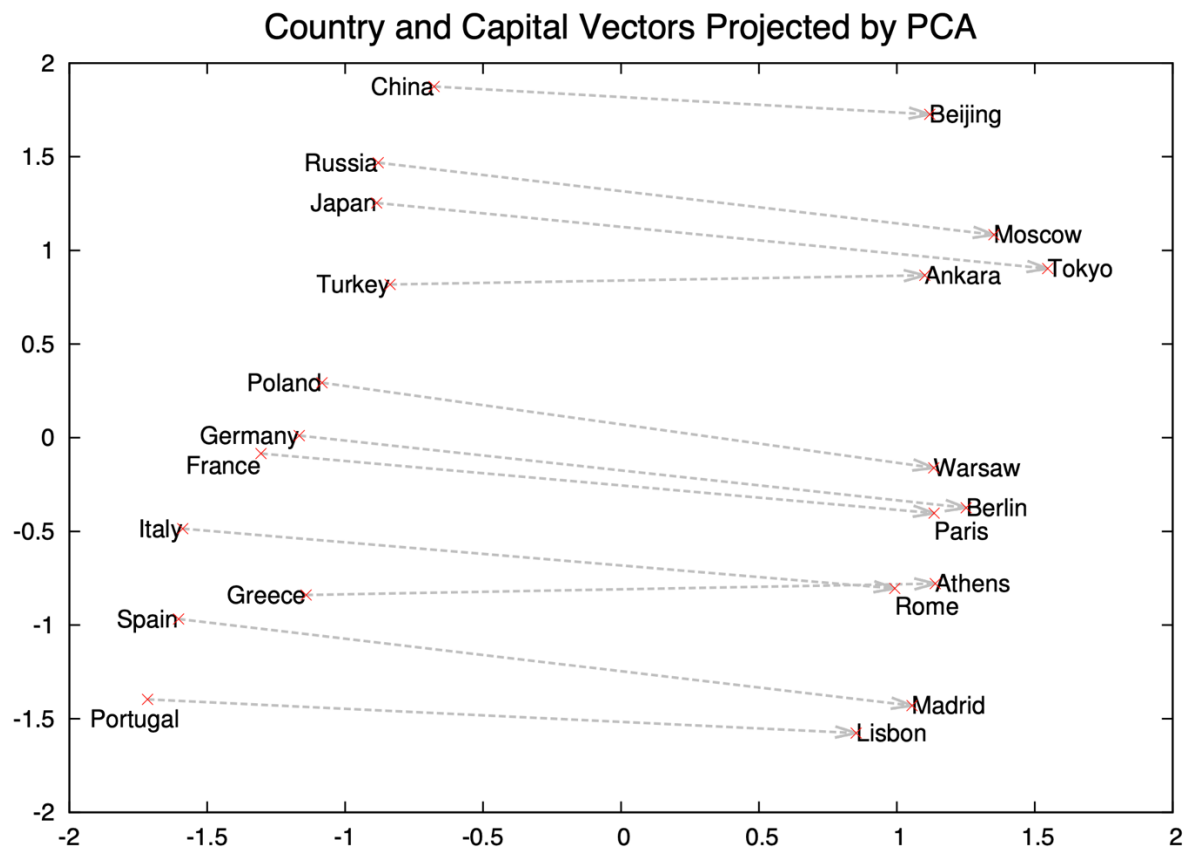
- Можно искать похожие слова
- Можно менять формы слов
- Можно искать определённые отношения
- Можно использовать как признаки для моделей



# word2vec

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna Check crown Polish zolty CTK	Hanoi Ho Chi Minh City Viet Nam Vietnamese	airline Lufthansa carrier Lufthansa flag carrier Lufthansa Lufthansa	Moscow Volga River upriver Russia	Juliette Binoche Vanessa Paradis Charlotte Gainsbourg Cecile De

# word2vec



# word2vec

- Яркий пример self-supervision
- Сейчас находит применения для изображений и даже для табличных данных
- Оказывается, в данных очень много информации даже без разметки

# Проблемы word2vec

- Не учитываем структуру слов
- Не закладываем никакой априорной информации о разных формах одного слова
- Не умеем обрабатывать опечатки

# FastText

- Заменяем каждое слово на «мешок»
- «руслан» -> (<руслан>, <ру, рус, усл, сла, лан, ан>)
- Слово  $w$  заменяется на набор токенов  $t_1, \dots, t_n$
- Мы обучаем векторы токенов:  $v_{t_1}, \dots, v_{t_n}$  (на самом деле есть «центральные» и «контекстные» версии всех векторов)
- $z_w = \sum_{i=1}^n v_{t_i}$  — вектор слова
- Все остальные детали — как в word2vec

# Что бывает ещё?

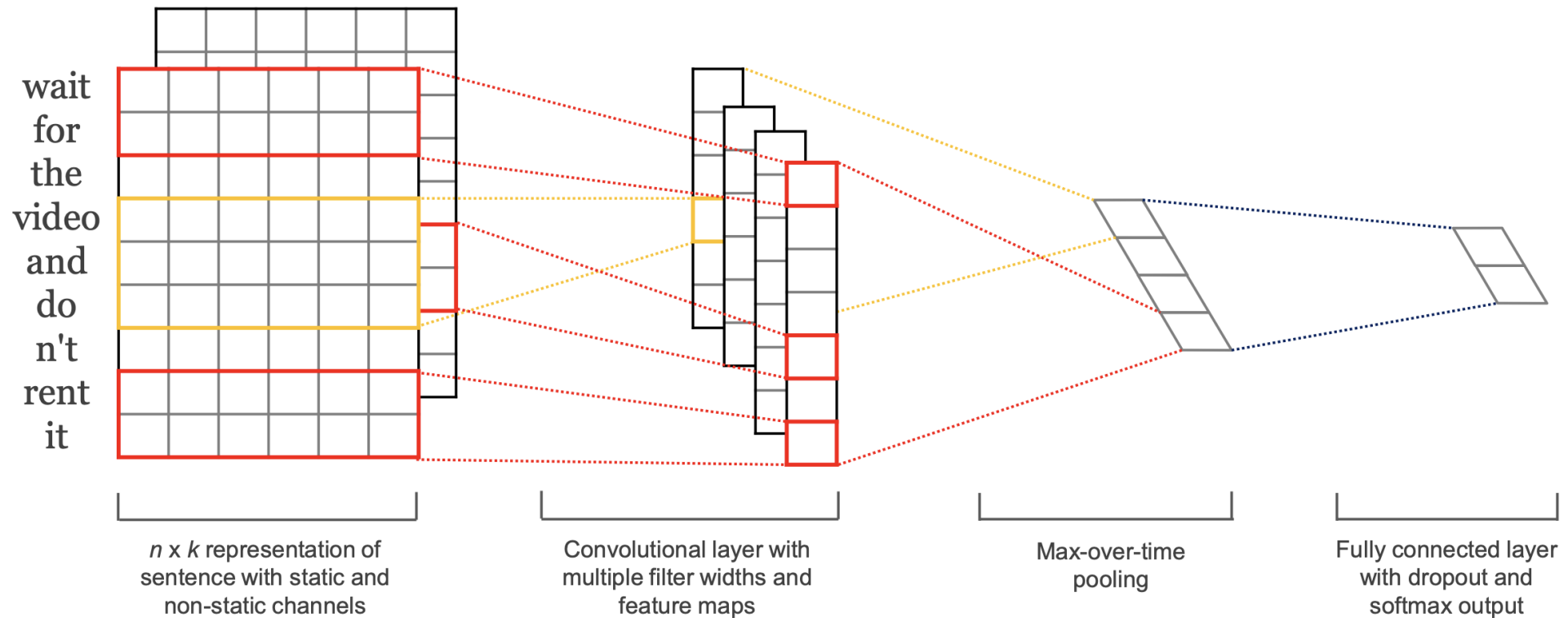
- GloVe
- ELMo/BERT (в следующих лекциях)

# Работа с текстом

- Векторные представления строятся для слов
- Можно просто усреднить по всем словам — получим признаки для текста
- Можно усреднять с весами
- Можно ли умнее?



# CNN для последовательностей



# CNN для последовательностей

- Можно обучать представления слов с нуля
- А можно инициализировать с помощью w2v — это сильно лучше!

# CNN для последовательностей

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	<b>89.6</b>
CNN-non-static	<b>81.5</b>	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	<b>88.1</b>	93.2	92.2	<b>85.0</b>	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	—	—	—	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	—	—	—	—
RNTN (Socher et al., 2013)	—	45.7	85.4	—	—	—	—
DCNN (Kalchbrenner et al., 2014)	—	48.5	86.8	—	93.0	—	—
Paragraph-Vec (Le and Mikolov, 2014)	—	<b>48.7</b>	87.8	—	—	—	—
CCAE (Hermann and Blunsom, 2013)	77.8	—	—	—	—	—	87.2
Sent-Parser (Dong et al., 2014)	79.5	—	—	—	—	—	86.3
NBSVM (Wang and Manning, 2012)	79.4	—	—	93.2	—	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	—	—	<b>93.6</b>	—	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	—	—	93.4	—	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	—	—	<b>93.6</b>	—	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	—	—	—	—	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	—	—	—	—	—	82.7	—
SVM <sub>S</sub> (Silva et al., 2011)	—	—	—	—	<b>95.0</b>	—	—

# Минусы

- Ищем выразительные «локальные» комбинации
- Не пытаемся понять смысл текста в целом

# Рекуррентные модели

# Марковские модели

- Предположение: наличие конкретного слова в тексте объясняется только  $k$  словами перед ним
- $p(w_1, \dots, w_n) = p(w_1)p(w_2|w_1) \dots p(w_n|w_{n-1}, \dots, w_{n-k})$
- $p(w_n|w_{n-1}, \dots, w_{n-k})$  — можно оценить
- Как часто встречается слово  $w_n$  после последовательности из слов  $w_{n-1}, \dots, w_{n-k}$ ?
- Обычно делают со сглаживанием

# Марковские модели

'I am a master armorer , lords of Westeros , sawing out each bay and peninsula  
'Jon Snow is with the Night's Watch . I did not survive a broken hip , a leath  
'Jon Snow is with the Hound in the woods . He won't do it . " Please don't'  
'Where are the chains , and the Knight of Flowers to treat with you , Imp . "'  
'Those were the same . Arianne demurred . " So the fishwives say , " It was Ty  
'He thought that would be good or bad for their escape . If they can truly giv  
'I thought that she was like to remember a young crow he'd met briefly years b



# Идея

- Мы читаем текст последовательно
- И постепенно всё лучше понимаем, о чём он

# Рекуррентные сети (RNN)

- Последовательность:  $x_1, x_2, \dots, x_n, \dots$
- Читаем слева направо
- $h_t$  — накопленная информация после чтения  $t$  элементов (вектор)

# Рекуррентные сети (RNN)

- Последовательность:  $x_1, x_2, \dots, x_n, \dots$
- $x_i$  — либо one-hot вектор, либо векторное представление (word2vec, fasttext, ...)

# Рекуррентные сети (RNN)

- Последовательность:  $x_1, x_2, \dots, x_n, \dots$
- Читаем слева направо
- $h_t$  — накопленная информация после чтения  $t$  элементов (вектор)
- $h_t = f(W_{xh}x_t + W_{hh}h_{t-1})$
- Если хотим что-то выдавать на каждом шаге:  $o_t = f_o(W_{ho}h_t)$

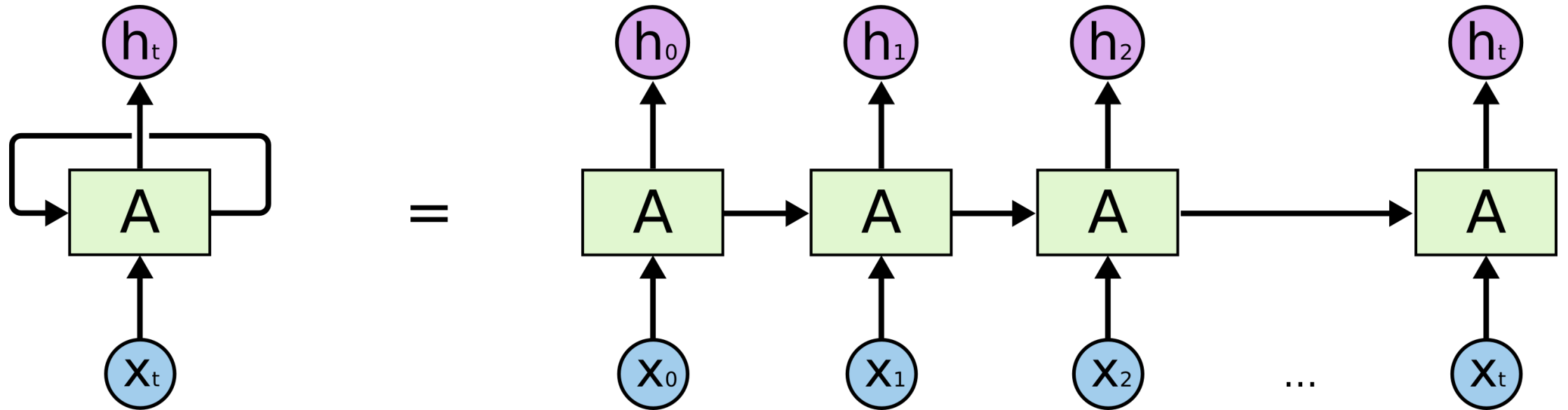
# Рекуррентные сети (RNN)

- Типичный случай:  $o_t \in \mathbb{R}^N$
- $N$  — размер словаря
- То есть предсказываем вероятность того, что здесь стоит конкретное слово
- Предсказываем следующее слово

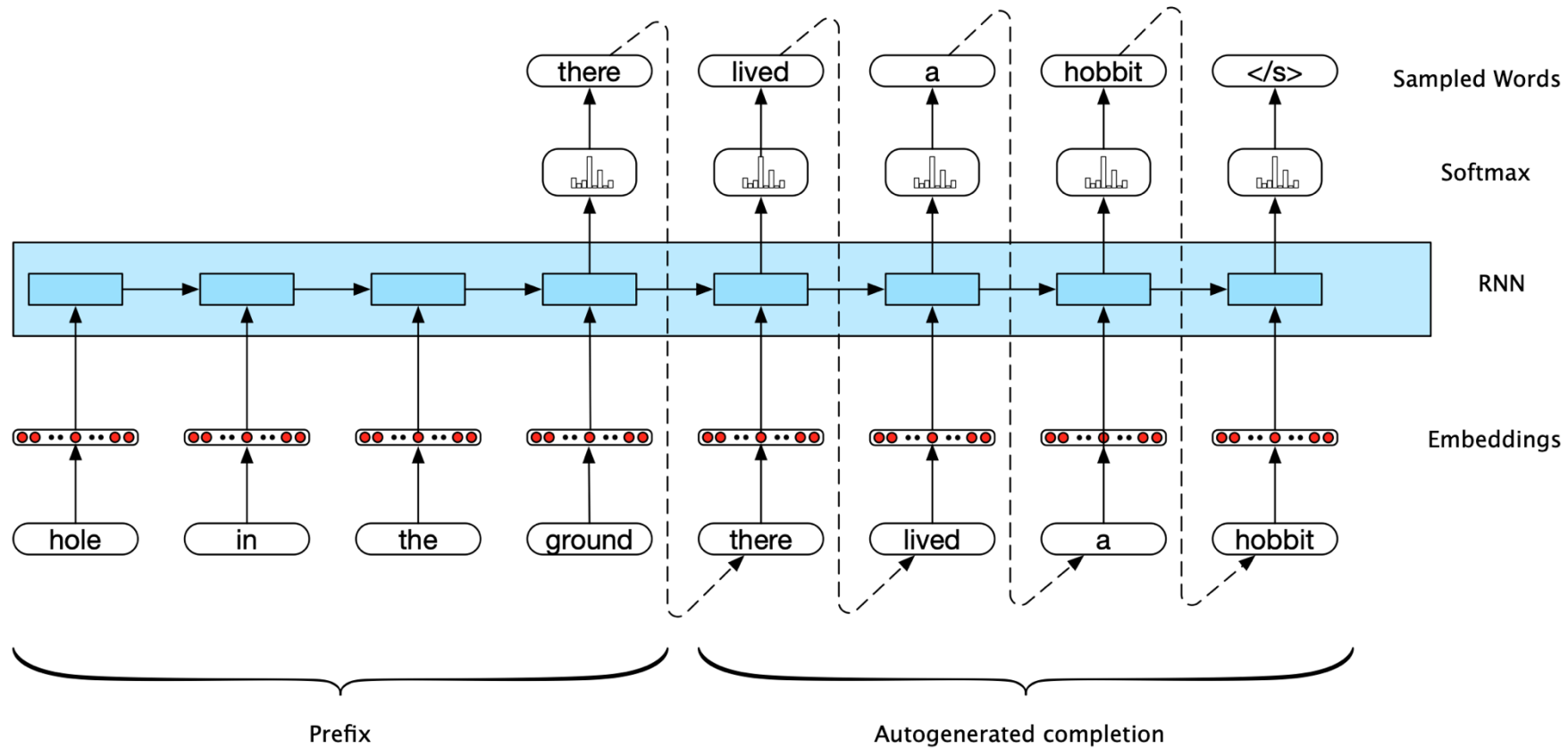
# Рекуррентные сети (RNN)

- Типичный случай:  $o_t \in \mathbb{R}^N$
- $N$  — количество частей речи
- POS tagging

# Рекуррентные сети (RNN)

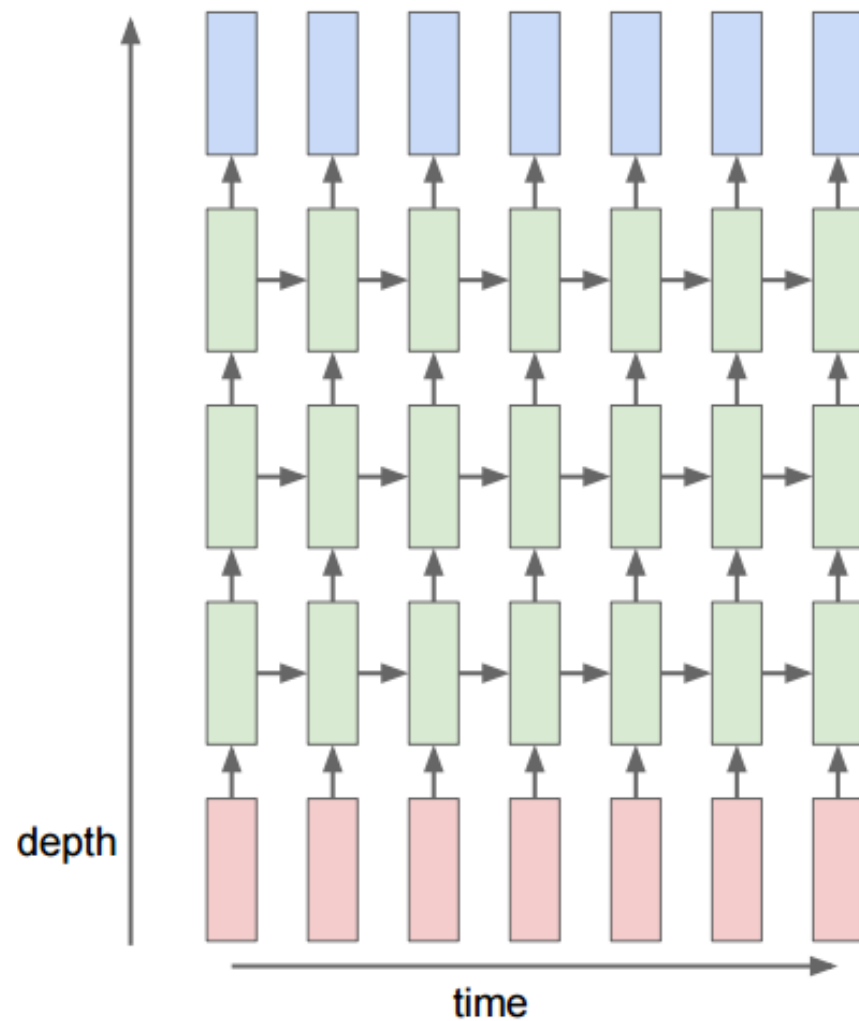


# Рекуррентные сети (RNN)





# Можно делать многослойные RNN



# Примеры

PANDARUS:

Alas, I think he shall be come approached and the day  
When little strain 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 nudes 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.

# Примеры

For  $\bigoplus_{n=1,\dots,m} \mathcal{L}_{m,n} = 0$ , hence we can find a closed subset  $\mathcal{H}$  in  $\mathcal{H}$  and any sets  $\mathcal{F}$  on  $X$ ,  $U$  is a closed immersion of  $S$ , then  $U \rightarrow T$  is a separated algebraic space.

*Proof.* Proof of (1). It also start we get

$$S = \text{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by  $\coprod Z \times_U U \rightarrow V$ . Consider the maps  $M$  along the set of points  $\text{Sch}_{fppf}$  and  $U \rightarrow U$  is the fibre category of  $S$  in  $U$  in Section, ?? and the fact that any  $U$  affine, see Morphisms, Lemma ?? . Hence we obtain a scheme  $S$  and any open subset  $W \subset U$  in  $\text{Sh}(G)$  such that  $\text{Spec}(R') \rightarrow S$  is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that  $f_i$  is of finite presentation over  $S$ . We claim that  $\mathcal{O}_{X,x}$  is a scheme where  $x, x', s'' \in S'$  such that  $\mathcal{O}_{X,x'} \rightarrow \mathcal{O}'_{X',x'}$  is separated. By Algebra, Lemma ?? we can define a map of complexes  $\text{GL}_{S'}(x'/S'')$  and we win.  $\square$

To prove study we see that  $\mathcal{F}|_U$  is a covering of  $\mathcal{X}'$ , and  $\mathcal{T}_i$  is an object of  $\mathcal{F}_{X/S}$  for  $i > 0$  and  $\mathcal{F}_p$  exists and let  $\mathcal{F}_i$  be a presheaf of  $\mathcal{O}_X$ -modules on  $\mathcal{C}$  as a  $\mathcal{F}$ -module. In particular  $\mathcal{F} = U/\mathcal{F}$  we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\text{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1} \mathcal{F}$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = (\text{Sch}/S)_{fppf}^{opp}, (\text{Sch}/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \mapsto (U, \text{Spec}(A))$$

is an open subset of  $X$ . Thus  $U$  is affine. This is a continuous map of  $X$  is the inverse, the groupoid scheme  $S$ .

*Proof.* See discussion of sheaves of sets.  $\square$

The result for prove any open covering follows from the less of Example ?? . It may replace  $S$  by  $X_{spaces, \acute{e}tale}$  which gives an open subspace of  $X$  and  $T$  equal to  $S_{Zar}$ , see Descent, Lemma ?? . Namely, by Lemma ?? we see that  $R$  is geometrically regular over  $S$ .

**Lemma 0.1.** Assume (3) and (3) by the construction in the description.

Suppose  $X = \lim |X|$  (by the formal open covering  $X$  and a single map  $\text{Proj}_X(\mathcal{A}) = \text{Spec}(B)$  over  $U$  compatible with the complex

$$\text{Set}(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X, \mathcal{O}_X}).$$

When in this case of to show that  $\mathcal{Q} \rightarrow \mathcal{C}_{Z/X}$  is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If  $T$  is surjective we may assume that  $T$  is connected with residue fields of  $S$ . Moreover there exists a closed subspace  $Z \subset X$  of  $X$  where  $U$  in  $X'$  is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem

(1)  $f$  is locally of finite type. Since  $S = \text{Spec}(R)$  and  $Y = \text{Spec}(R)$ .

*Proof.* This is form all sheaves of sheaves on  $X$ . But given a scheme  $U$  and a surjective étale morphism  $U \rightarrow X$ . Let  $U \cap U = \coprod_{i=1,\dots,n} U_i$  be the scheme  $X$  over  $S$  at the schemes  $X_i \rightarrow X$  and  $U = \lim_i X_i$ .  $\square$

The following lemma surjective restrocomposes of this implies that  $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{X,\dots,0}$ .

**Lemma 0.2.** Let  $X$  be a locally Noetherian scheme over  $S$ ,  $E = \mathcal{F}_{X/S}$ . Set  $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$ . Since  $\mathcal{I}^n \subset \mathcal{I}^n$  are nonzero over  $i_0 \leq \mathfrak{p}$  is a subset of  $\mathcal{J}_{n,0} \circ \mathcal{A}_2$  works.

**Lemma 0.3.** In Situation ?? . Hence we may assume  $\mathfrak{q}' = 0$ .

*Proof.* We will use the property we see that  $\mathfrak{p}$  is the next functor (??). On the other hand, by Lemma ?? we see that

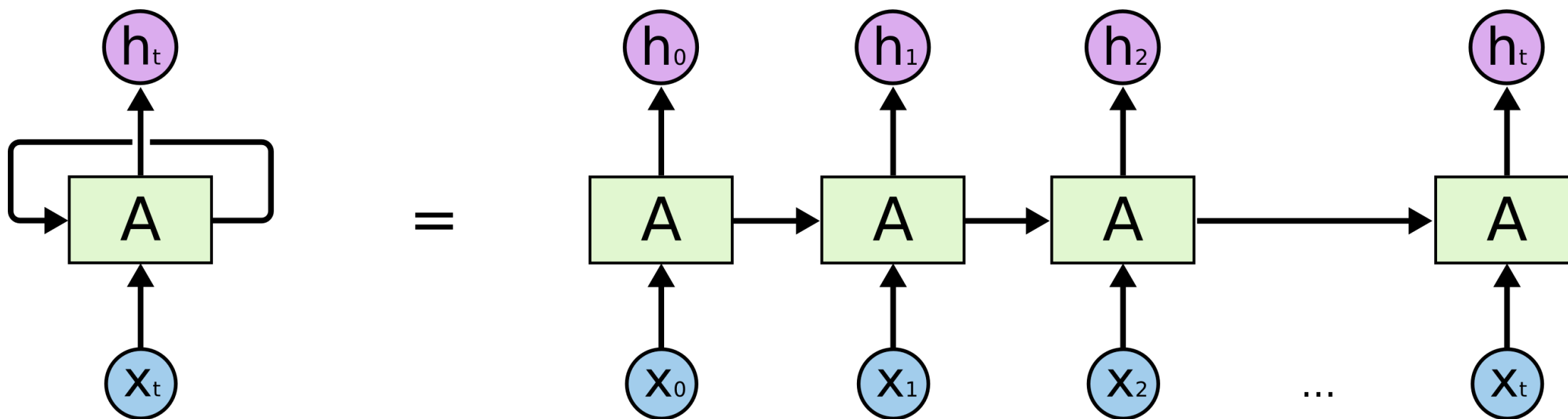
$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where  $K$  is an  $F$ -algebra where  $\delta_{n+1}$  is a scheme over  $S$ .  $\square$

# Примеры

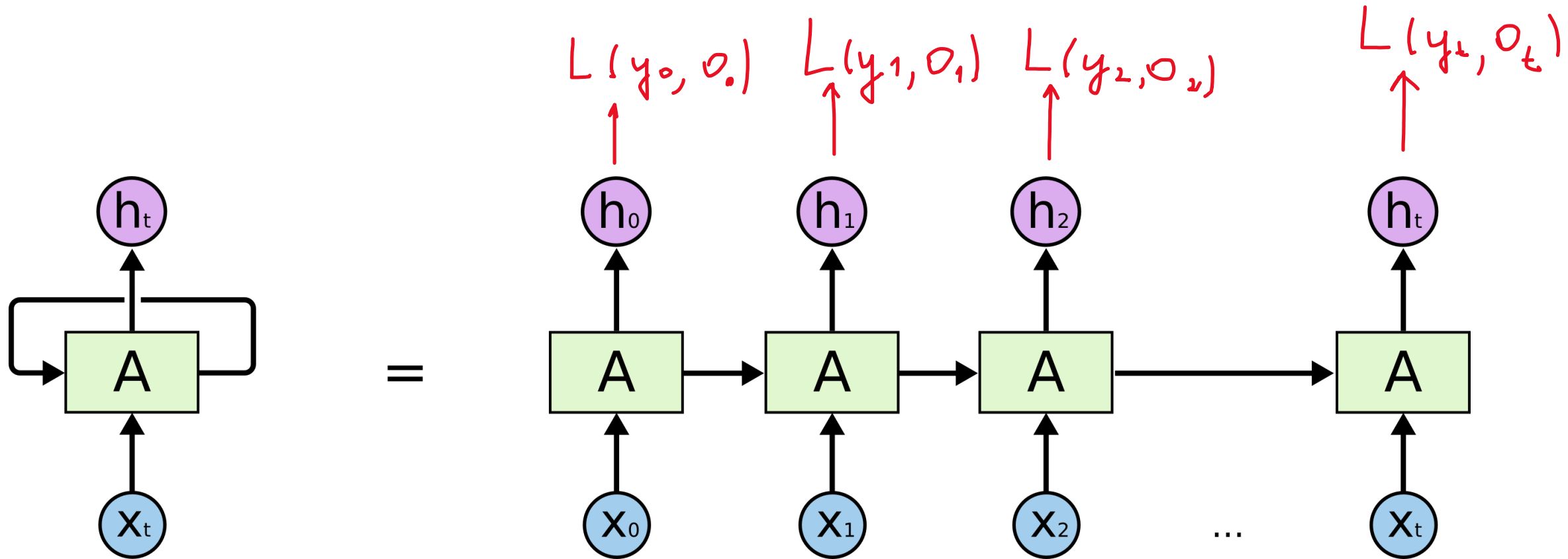
```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
    int error;
    if (fd == MARN_EPT) {
        /*
         * The kernel blank will coeld it to userspace.
         */
        if (ss->segment < mem_total)
            unblock_graph_and_set_blocked();
        else
            ret = 1;
        goto bail;
    }
    segaddr = in_SB(in.addr);
    selector = seg / 16;
    setup_works = true;
    for (i = 0; i < blocks; i++) {
        seq = buf[i++];
        bpf = bd->bd.next + i * search;
        if (fd) {
            current = blocked;
        }
    }
    rw->name = "Getjbbregs";
    bprm_self_clearl(&iv->version);
    regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
    return segtable;
}
```

# Развёртка RNN



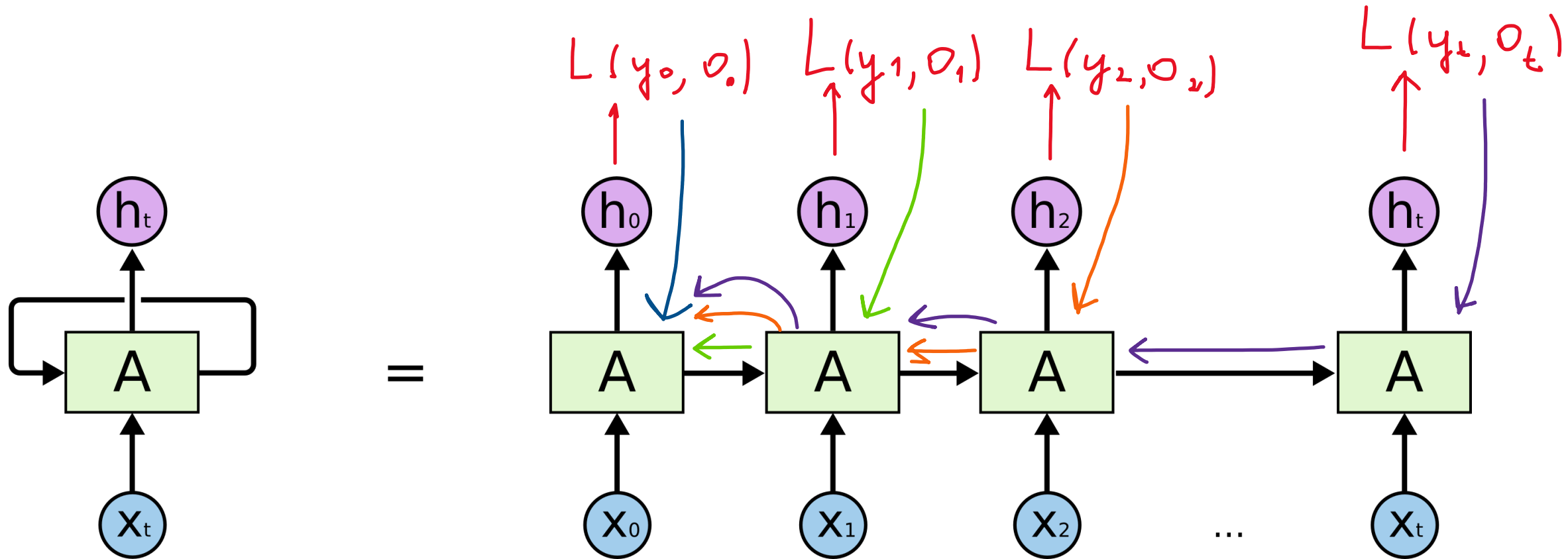
- $h_t = f(W_{xh}x_t + W_{hh}h_{t-1})$
- $o_t = f_o(W_{ho}h_t)$

# Backpropagation Through Time (BPTT)



- $h_t = f(W_{xh}x_t + W_{hh}h_{t-1})$
- $o_t = f_o(W_{ho}h_t)$

# Backpropagation Through Time (BPTT)



- $h_t = f(W_{xh}x_t + W_{hh}h_{t-1})$
- $o_t = f_o(W_{ho}h_t)$

# Backpropagation Through Time (BPTT)

