DEEP LEARNING V: РЕКУРРЕНТНЫЕ СЕТИ

Сергей Николенко

НИУ ВШЭ — Санкт-Петербург
8 декабря 2017 г.

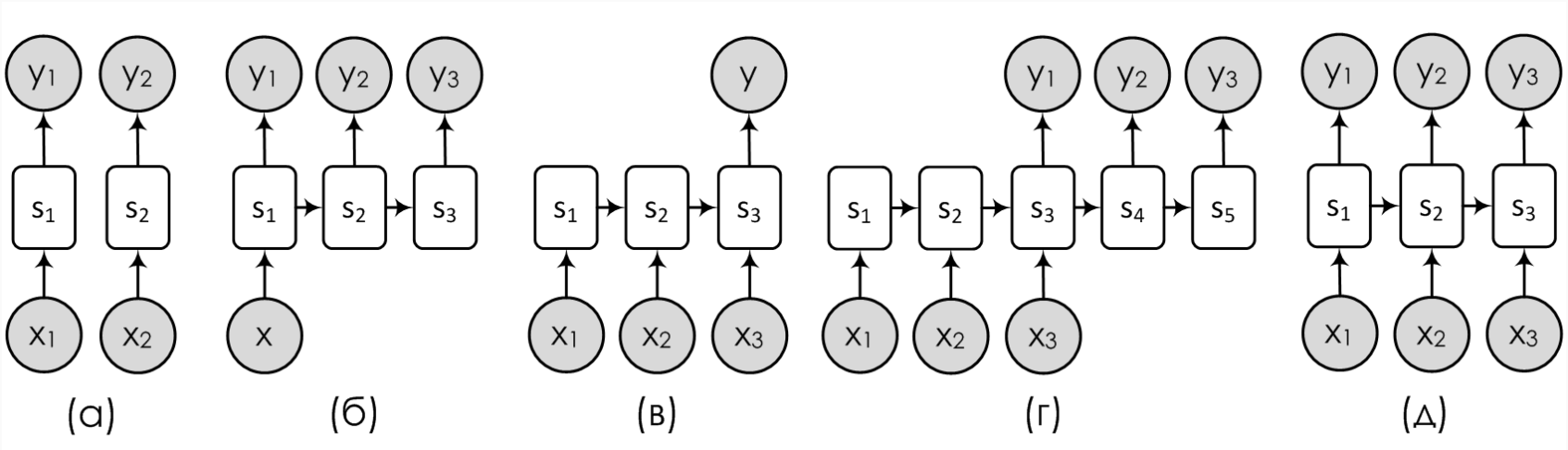
Random facts:

- 8 декабря 877 г., после смерти Карла II Лысого, короновался его сын Людовик II Косноязычный
- 8 декабря 1660 г. на английской сцене впервые появилась женщина; играла Дездемону
- 8 декабря 1854 г. папа Пий IX издал буллу Ineffabilis Deus о непорочном зачатии, в результате чего 8 декабря стало одним из центральных католических праздников
- 8 декабря 1991 г. Ельцин, Кравчук и Шушкевич подписали в Беловежской пуще соглашение о создании СНГ
- 8 декабря --- День студента в Болгарии и Македонии

РЕКУРРЕНТНЫЕ НЕЙРОННЫЕ СЕТИ

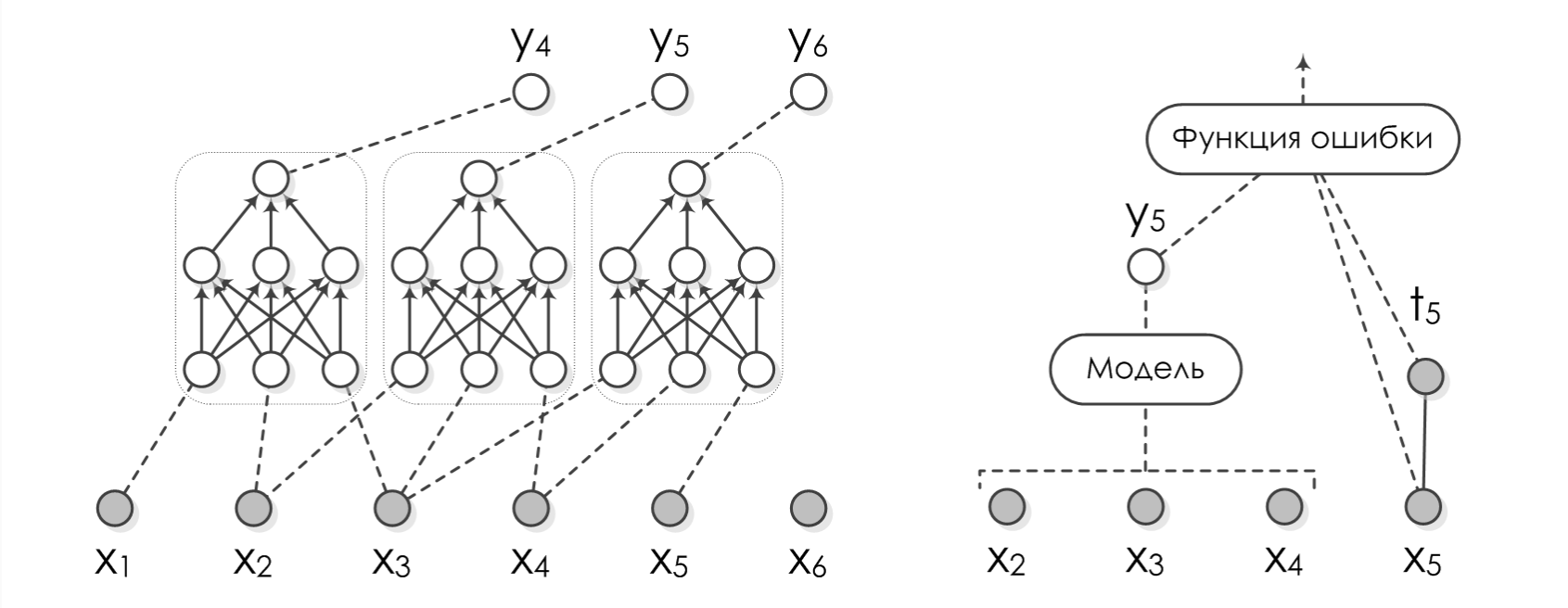
ПОСЛЕДОВАТЕЛЬНОСТИ

- Последовательности: текст, временные ряды, речь, музыка...
- Есть разные виды задач, основанных на последовательностях:



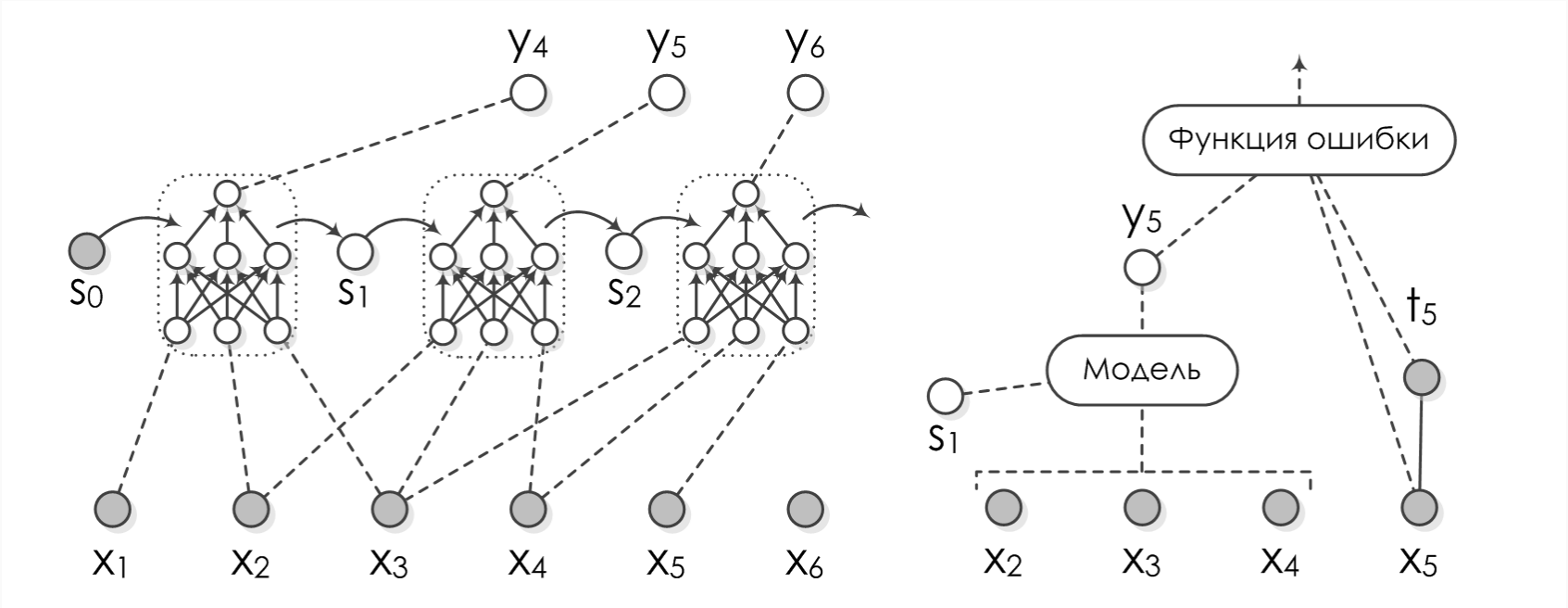
ПОСЛЕДОВАТЕЛЬНОСТИ

- Как применить к последовательности нейронную сеть?
- Можно использовать скользящее окно:



ПОСЛЕДОВАТЕЛЬНОСТИ

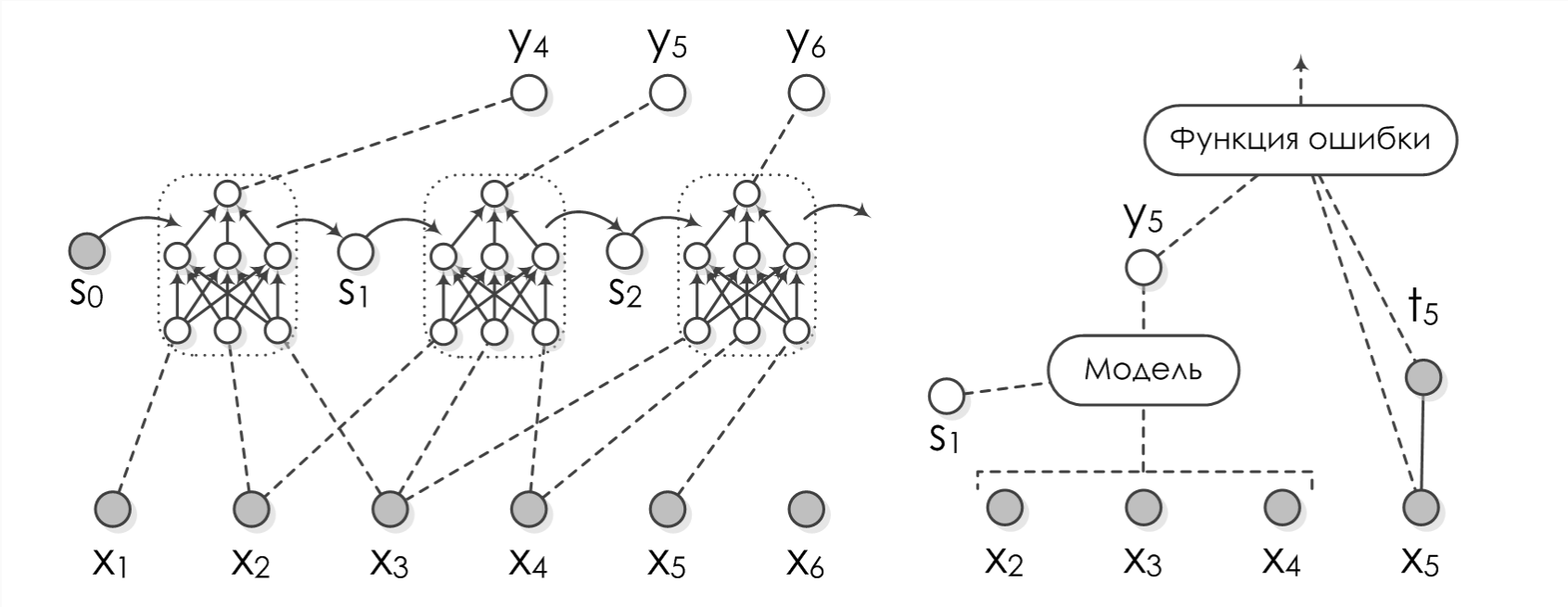
- ...но ещё лучше будет сохранять какое-нибудь скрытое состояние и обновлять его каждый раз.
- Это в точности идея *рекуррентных нейронных сетей* (recurrent neural networks, RNN).



ПОСЛЕДОВАТЕЛЬНОСТИ

- Но как теперь делать backpropagation? Получается, что в графе вычислений теперь циклы:

$$s_i = h(x_i, x_{i+1}, x_{i+2}, s_{i-1}).$$



- Это же ужасно, и всё сломалось?..

ПОСЛЕДОВАТЕЛЬНОСТИ

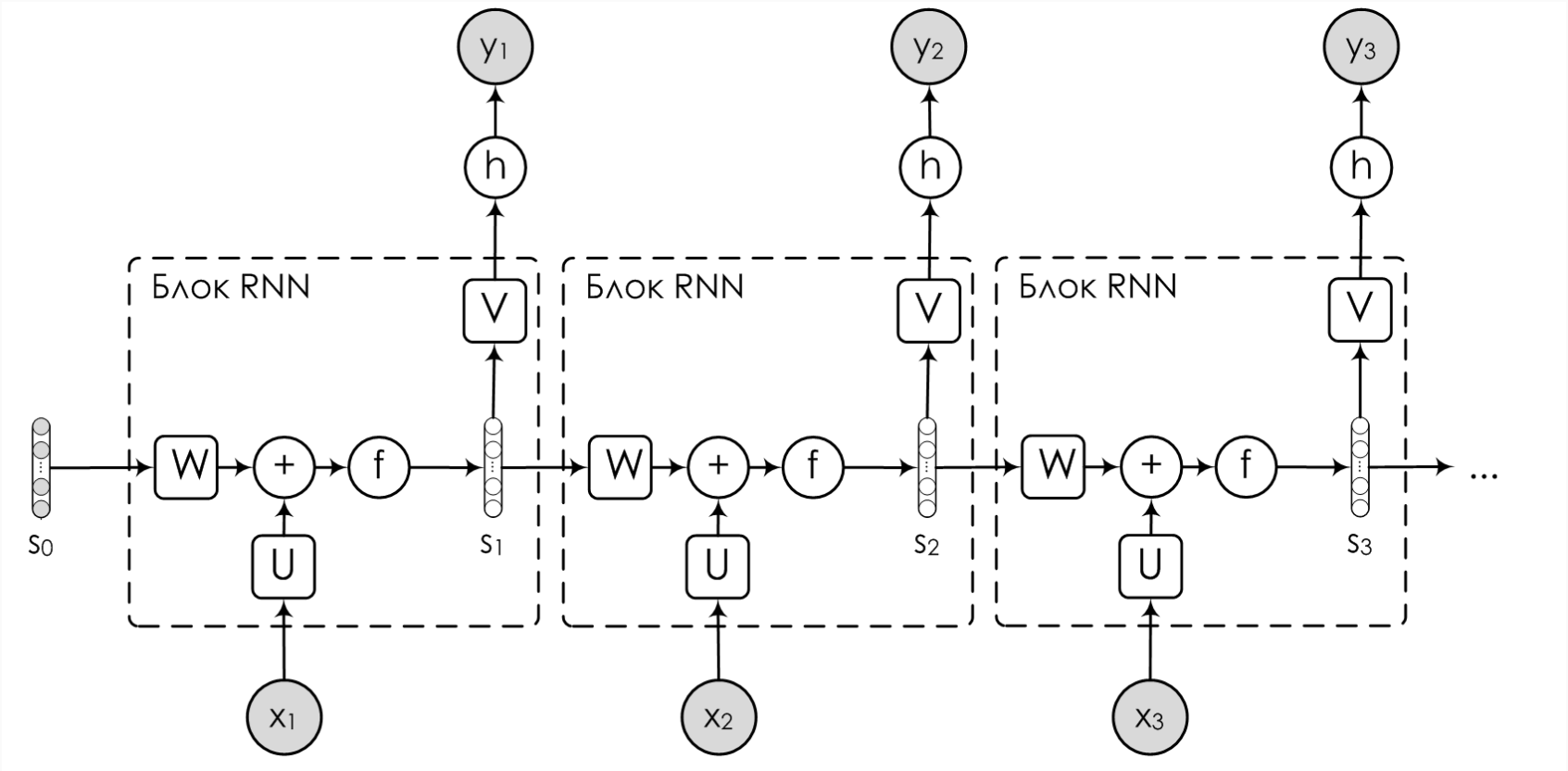
- ...да нет, конечно. Можно “развернуть” циклы обратно:

$$y_6 = f(x_3, x_4, x_5, s_2) = f(x_3, x_4, x_5, h(x_2, x_3, x_4, s_1)) = \\ = f(x_3, x_4, x_5, h(x_2, x_3, x_4, h(x_1, x_2, x_3, s_0))).$$

- Так что формально проблемы нет.
- Но масса проблем в реальности: получается, что рекуррентная сеть – это такая *очень* глубокая сеть с кучей общих весов...

ПРОСТАЯ RNN

- “Простая” RNN:



ПРОСТАЯ RNN

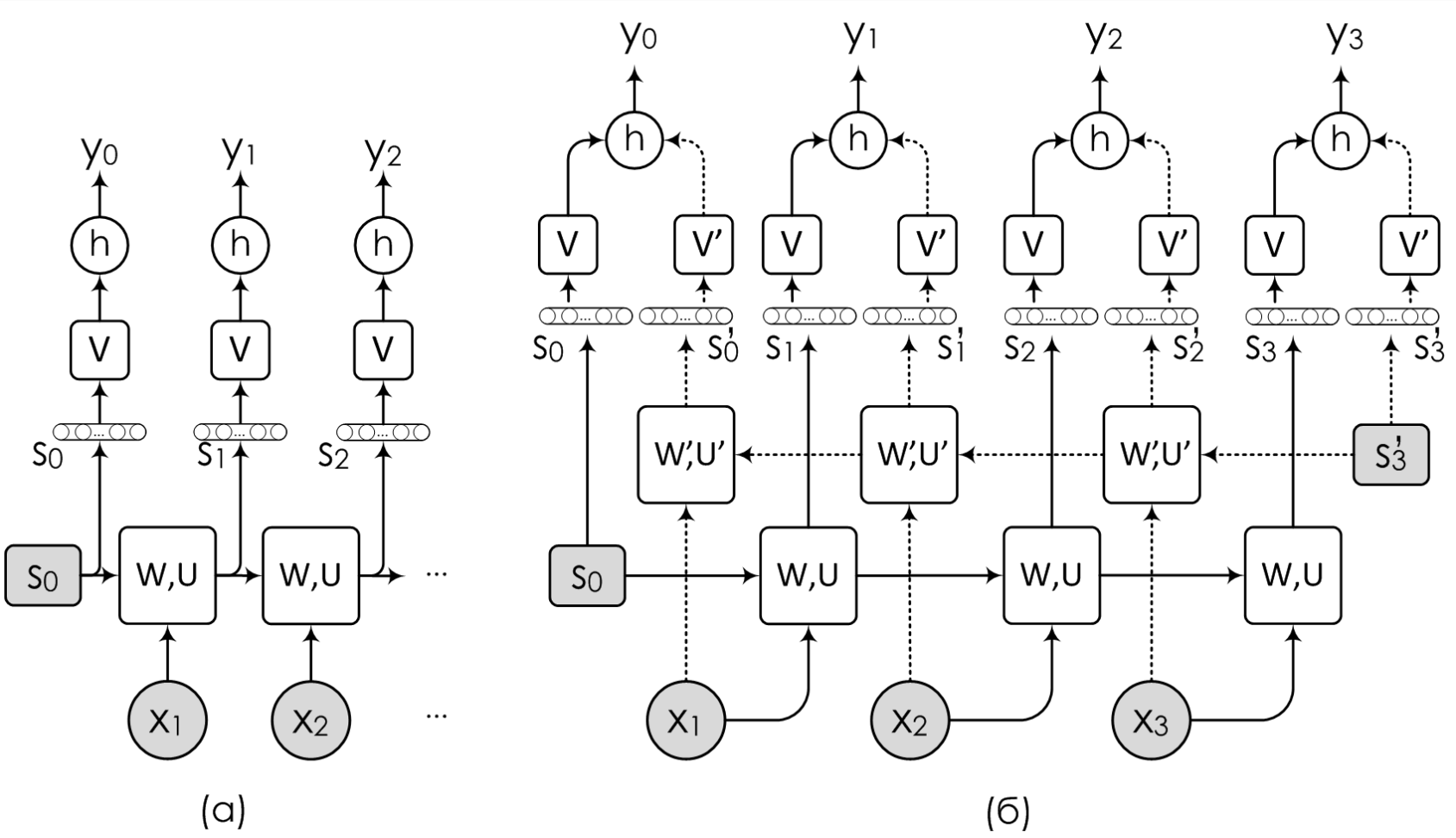
- Формально:

$$\begin{aligned}
\mathbf{a}_t &= \mathbf{b} + W\mathbf{s}_{t-1} + U\mathbf{x}_t, \\
\mathbf{s}_t &= f(\mathbf{a}_t), \\
o_t &= c + V\mathbf{s}_t, \\
\mathbf{y}_t &= h(o_t),
\end{aligned}$$

где f – рекуррентная нелинейность, h – функция выхода.

ДВУНАПРАВЛЕННАЯ RNN

- Иногда нужен контекст с обеих сторон:



ДВУНАПРАВЛЕННАЯ RNN

- Формально:

$$\begin{aligned}
\mathbf{s}_t &= \sigma\left(\mathbf{b} + W\mathbf{s}_{t-1} + U\mathbf{x}_t\right), \\
\mathbf{s}'_t &= \sigma\left(\mathbf{b}' + W'\mathbf{s}'_{t+1} + U'\mathbf{x}_t\right), \\
o_t &= c + V\mathbf{s}_t + V'\mathbf{s}'_t, \\
\mathbf{y}_t &= h\left(o_t\right).
\end{aligned}$$

- И это, конечно, обобщается на любой другой тип конструкций.

ВЗРЫВАЮЩИЕСЯ ГРАДИЕНТЫ

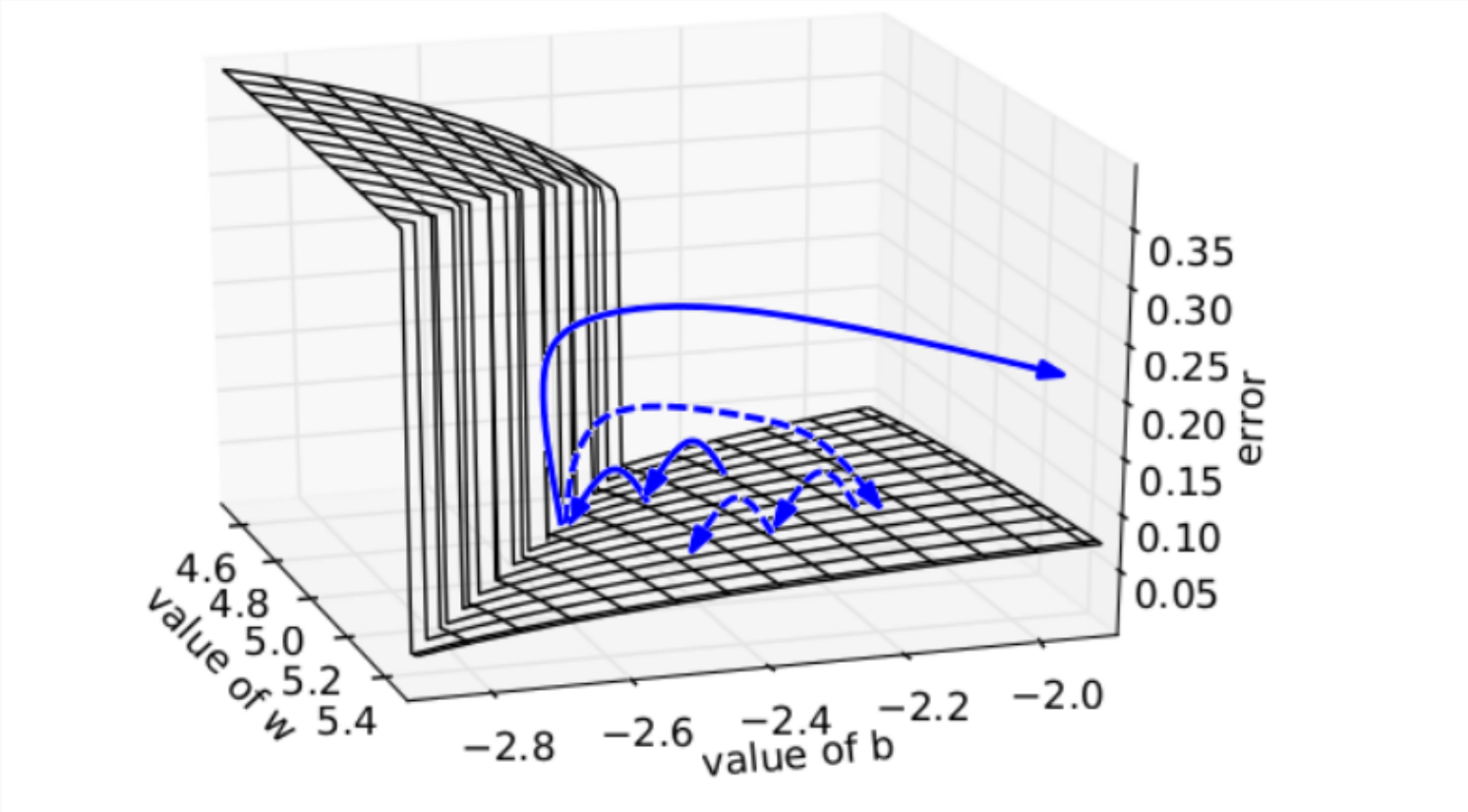
- Две проблемы:
  - взрывающиеся градиенты (exploding gradients);
  - затухающие градиенты (vanishing gradients).
- Надо каждый раз умножать на одну и ту же W, и норма градиента может расти или убывать экспоненциально.
- Взрывающиеся градиенты: надо каждый раз умножать на W, и норма градиента может расти экспоненциально.
- Что делать?

ВЗРЫВАЮЩИЕСЯ ГРАДИЕНТЫ

- Да просто обрезать градиенты, ограничить сверху, чтобы не росли.
- Два варианта – ограничить общую норму или каждое значение:
  - `sgd = optimizers.SGD(lr=0.01, clipnorm=1.)`
  - `sgd = optimizers.SGD(lr=0.01, clipvalue=.05)`

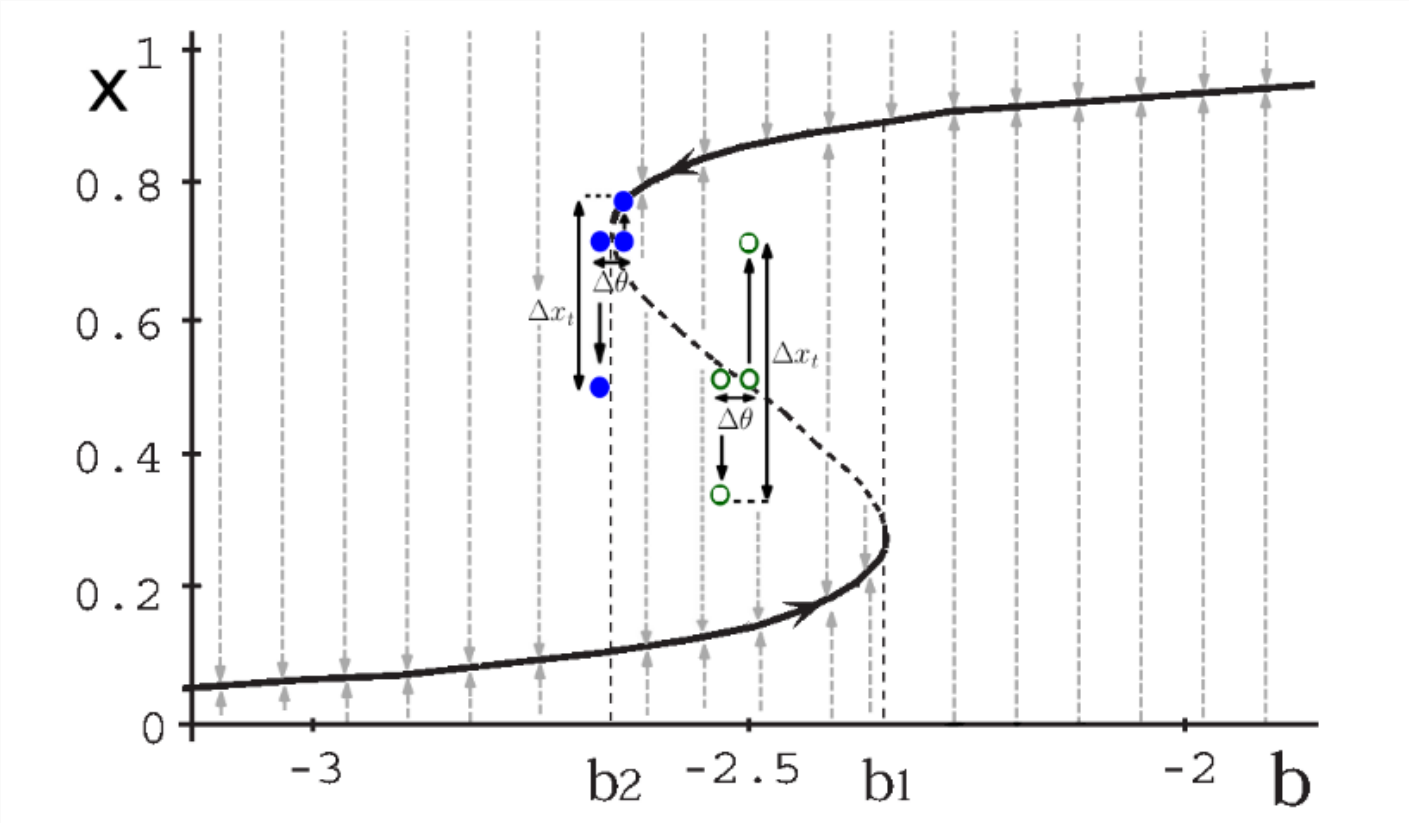
ВЗРЫВАЮЩИЕСЯ ГРАДИЕНТЫ

- (Pascanu et al., 2013) – вот что будет происходить:



ВЗРЫВАЮЩИЕСЯ ГРАДИЕНТЫ

- Там же объясняется, откуда возьмутся такие перепады: есть точки бифуркации у RNN.



Карусель константной ошибки: LSTM и GRU

LSTM

- Затухающие градиенты: надо каждый раз умножать на W.
- Поэтому не получается долгосрочную память реализовать.

- А хочется. Что делать?..

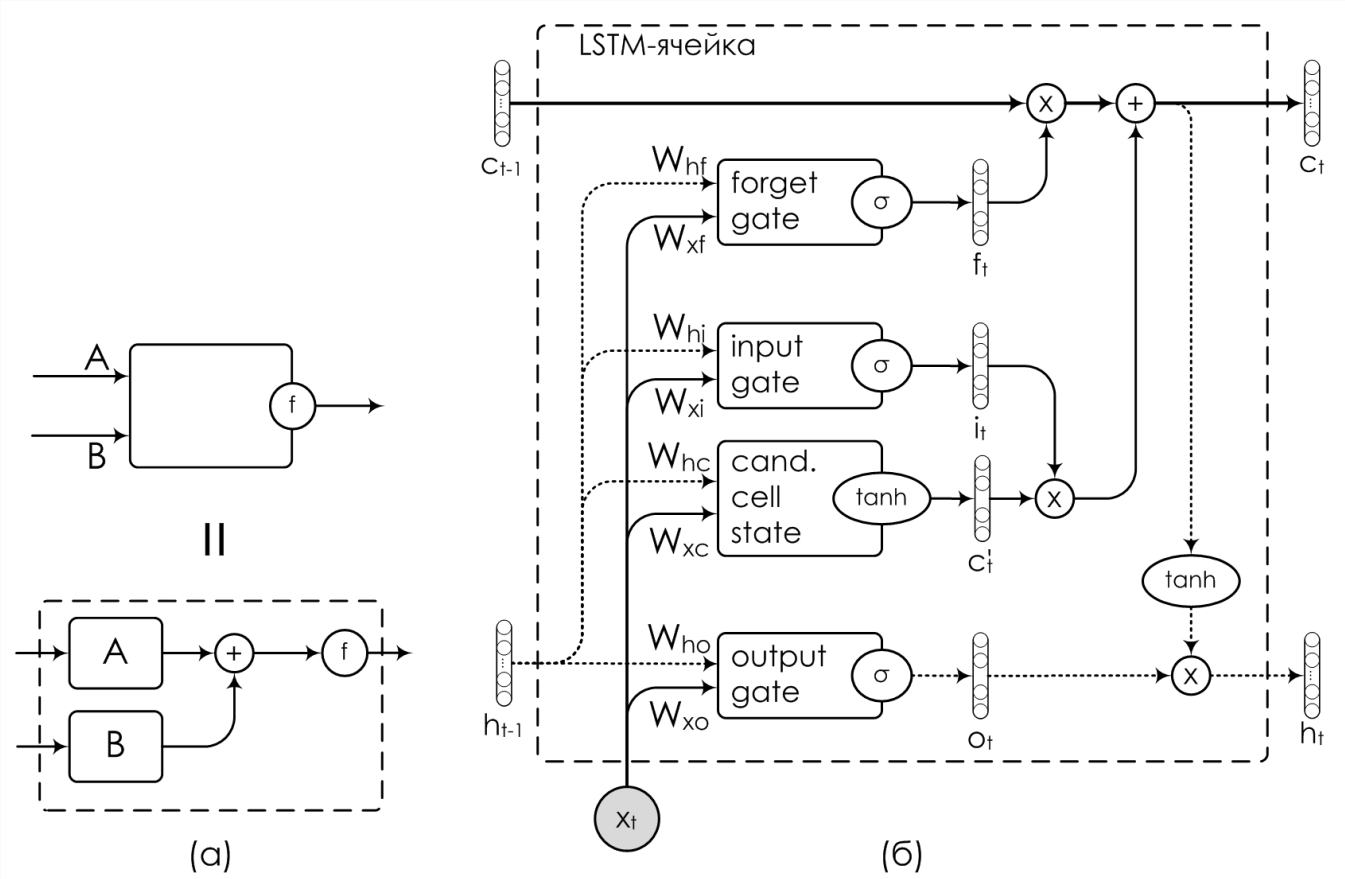
LSTM

- Базовую идею мы уже видели в ResNet: надо сделать так, чтобы градиент проходил.
- В RNN это называется «карусель константной ошибки» (constant error carousel).

- Идея из середины 1990-х (Шмидхубер): давайте составлять RNN из более сложных частей, в которых будет прямой путь для градиентов, и память будет контролироваться явно.

LSTM

- LSTM (long short-term memory). “Ванильный” LSTM: c_t – состояние ячейки памяти, h_t – скрытое состояние.
- Input gate и forget gate определяют, надо ли менять c_t на нового кандидата в состояния ячейки.



LSTM

- Формально:

$$
\begin{aligned}
c'_t &= \tanh\left(W_{xc}\mathbf{x}_t + W_{hc}h_{t-1} + \mathbf{b}_{c'}\right) && \textit{candidate cell state} \\
i_t &= \sigma\left(W_{xi}\mathbf{x}_t + W_{hi}h_{t-1} + \mathbf{b}_i\right) && \textit{input gate} \\
f_t &= \sigma\left(W_{xf}\mathbf{x}_t + W_{hf}h_{t-1} + \mathbf{b}_f\right) && \textit{forget gate} \\
o_t &= \sigma\left(W_{xo}\mathbf{x}_t + W_{ho}h_{t-1} + \mathbf{b}_o\right) && \textit{output gate} \\
c_t &= f_t \odot c_{t-1} + i_t \odot c'_t, && \textit{cell state} \\
h_t &= o_t \odot \tanh(c_t) && \textit{block output}
\end{aligned}
$$

- Так что LSTM может контролировать состояние ячейки при помощи скрытого состояния и весов.
- Например, если forget gate закрыт ($f_t = 1$), то получится карусель константной ошибки: $c_t = c_{t-1} + i_t \odot c'_t$, и $\frac{\partial c_t}{\partial c_{t-1}} = 1$.
- Важно инициализировать $\mathbf{b}_f$ большим, чтобы forget gate был закрыт поначалу.

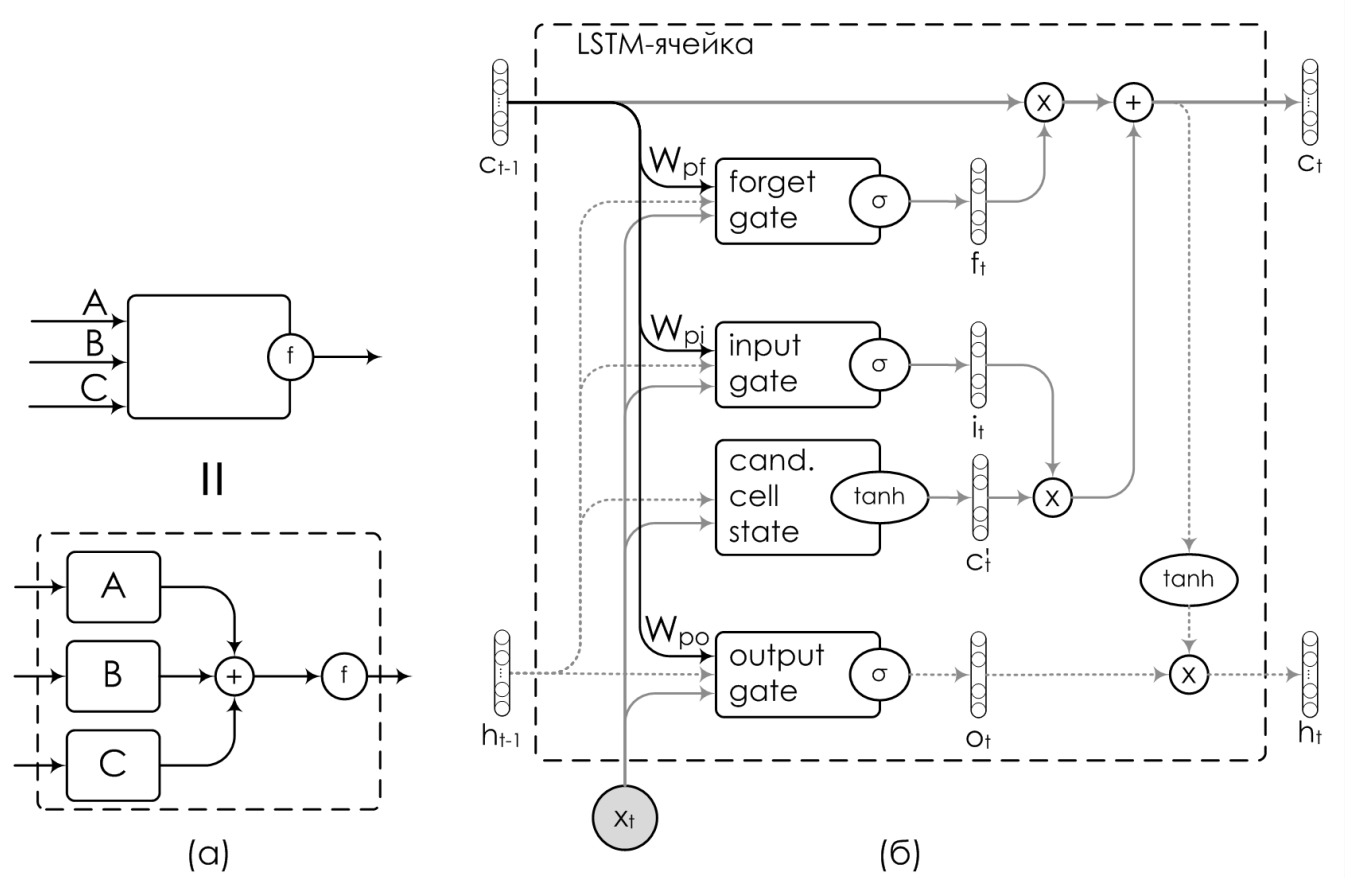
LSTM

- LSTM был создан в середине 1990-х (Hochreiter and Schmidhuber, 1995; 1997).
- В полностью современной форме в (Gers, Schmidhuber, 2000).
- Проблема: хотим управлять c, но гейты его не получают! Они видят только h_{t-1}, а это

$$h_{t-1} = o_{t-1} \odot \tanh(c_{t-1}).$$

- Так что если output gate закрыт, то поведение LSTM вообще от состояния ячейки не зависит.
- Нехорошо. Что делать?..

- …конечно, добавить ещё несколько матриц! (peepholes)



LSTM

- Формально:

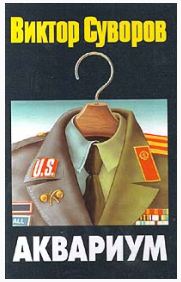
$$
\begin{aligned}
i_t &= \sigma\left(W_{xi}\mathbf{x}_t + W_{hi}h_{t-1} + W_{pi}c_{t-1} + \mathbf{b}_i\right) \\
f_t &= \sigma\left(W_{xf}\mathbf{x}_t + W_{hf}h_{t-1} + W_{pf}c_{t-1} + \mathbf{b}_f\right) \\
o_t &= \sigma\left(W_{xo}\mathbf{x}_t + W_{ho}h_{t-1} + W_{po}c_{t-1} + \mathbf{b}_o\right)
\end{aligned}
$$

- Видно, что тут есть огромное поле для вариантов LSTM: можно удалить любой гейт, любую замочную скважину, поменять функции активации...
- Как выбрать?

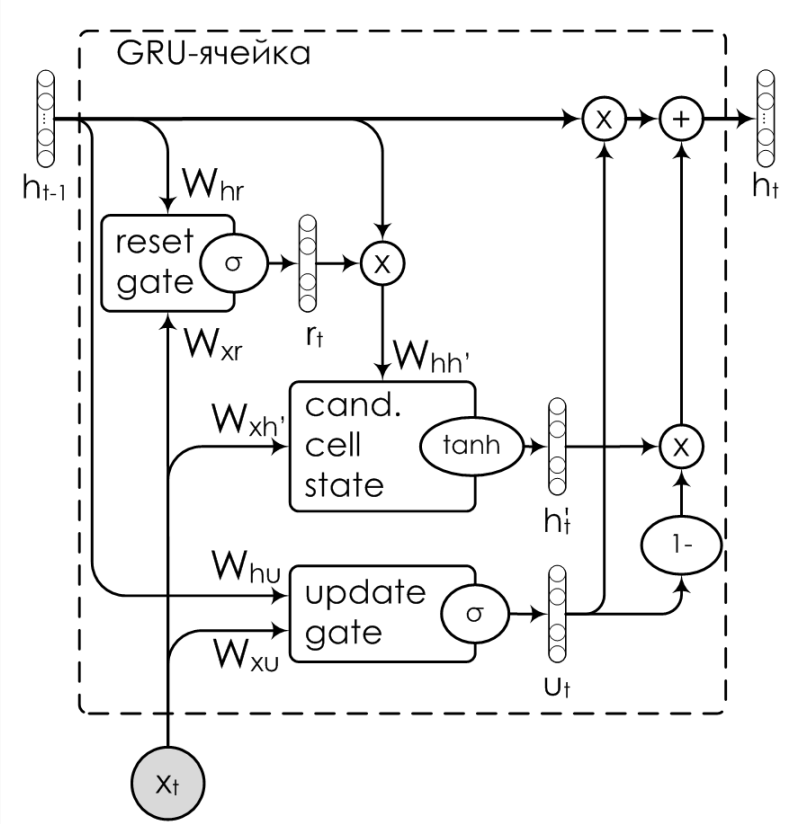
LSTM

- «LSTM: a Search Space Odyssey» (Greff et al., 2015).
- Большое экспериментальное сравнение.
- В честности, некоторые куда более простые архитектуры (без одного из гейтов!) не сильно проигрывали «ванильному» LSTM.
- И это приводит нас к...

GRU



- ...Gated Recurrent Units (GRU; Cho et al., 2014).
- В GRU тоже есть прямой путь для градиентов, но проще.



GRU

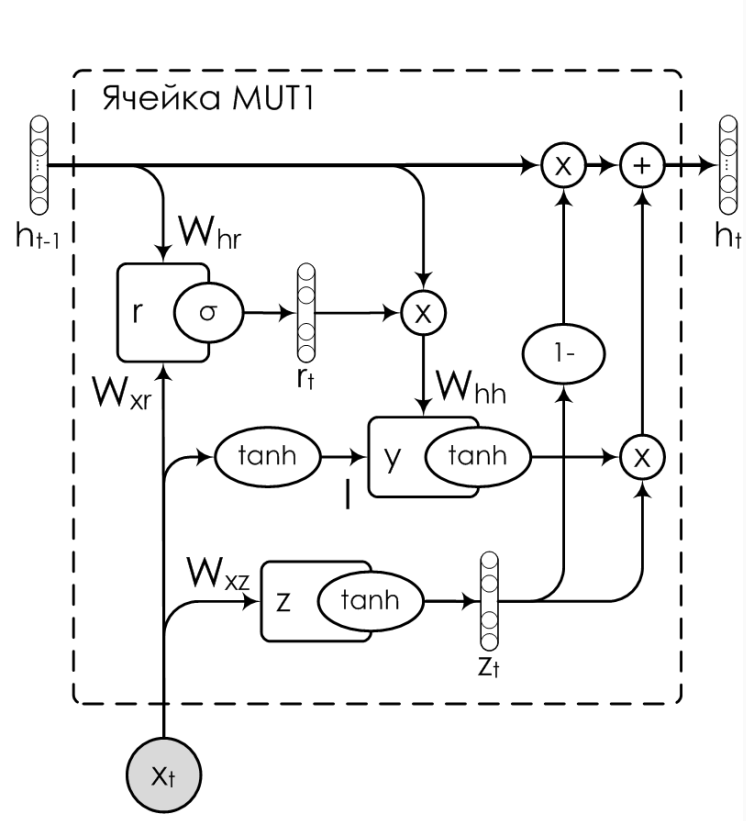
- Формально:

$$
\begin{aligned}
u_t &= \sigma(W_{xu}\mathbf{x}_t + W_{hu}h_{t-1} + \mathbf{b}_u) \\
r_t &= \sigma(W_{xr}\mathbf{x}_t + W_{hr}h_{t-1} + \mathbf{b}_r) \\
h'_t &= \tanh(W_{xh'}\mathbf{x}_t + W_{hh'}(r_t \odot h_{t-1})) \\
h_t &= (1 - u_t) \odot h'_t + u_t \odot h_{t-1}
\end{aligned}
$$

- Теперь есть update gate и reset gate, нет разницы между c_t и h_t.
- Меньше матриц (6, а не 8 или 11 с замочными скважинами), меньше весов, но только чуть хуже LSTM работает.
- Так что можно больше GRU поместить, и сеть станет лучше.

GRU

- Другие варианты тоже есть.
- (Józefowicz, Zaremba, Sutskever, 2015): огромное сравнение, выращивали архитектуры эволюционными методами.
- Три новых интересных архитектуры; например:



Долгосрочная память в базовых RNN

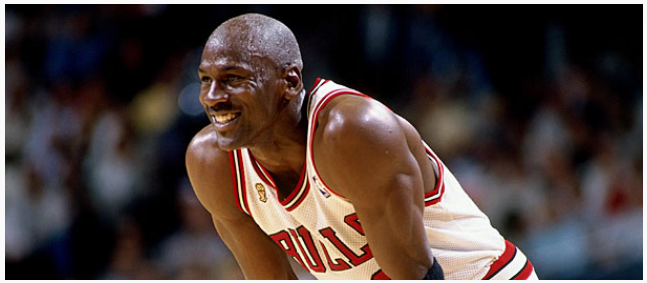
SCRN

- Следующая идея о том, как добавить долгосрочную память.
- Начнём опять с простой RNN:

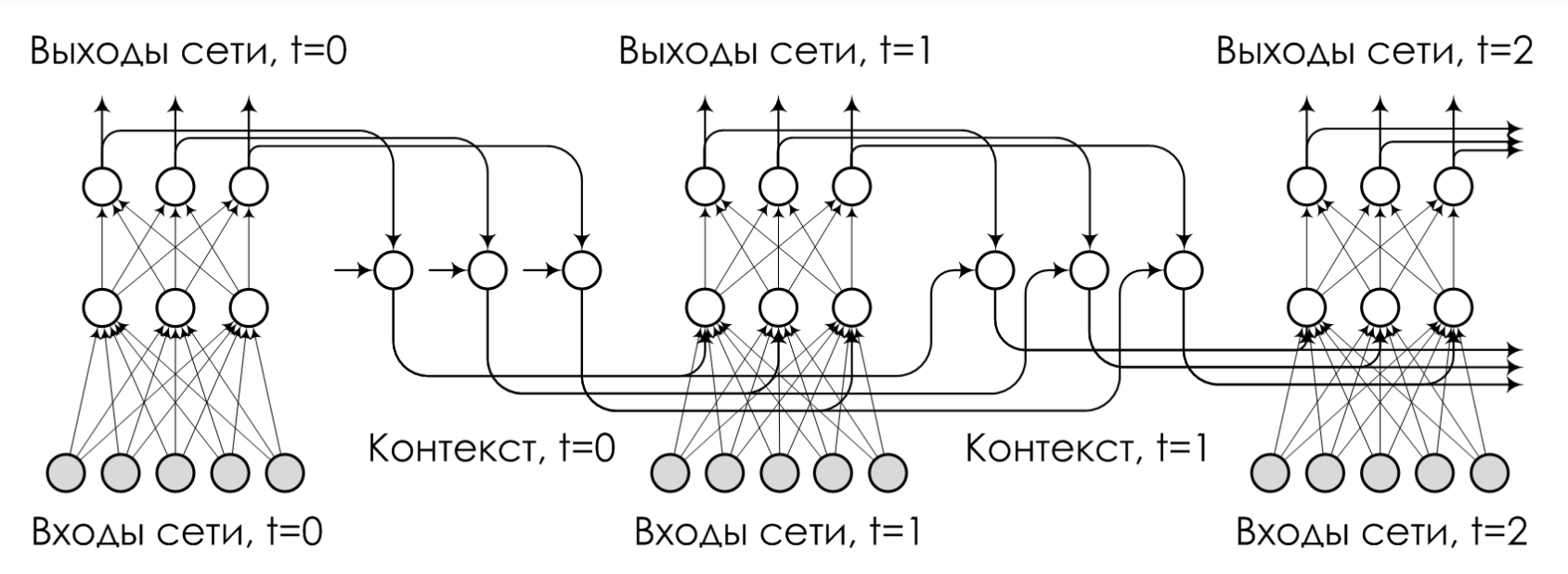
$$\mathbf{s}_t = f(U\mathbf{x}_t + W\mathbf{s}_{t-1} + \mathbf{b}), \quad \mathbf{y}_t = h(U\mathbf{s}_t + c).$$

- Проблема с градиентами в том, что мы умножаем на W, и градиенты либо взрываются, либо затухают.
- Давайте вернёмся к истории RNN...

SCRN



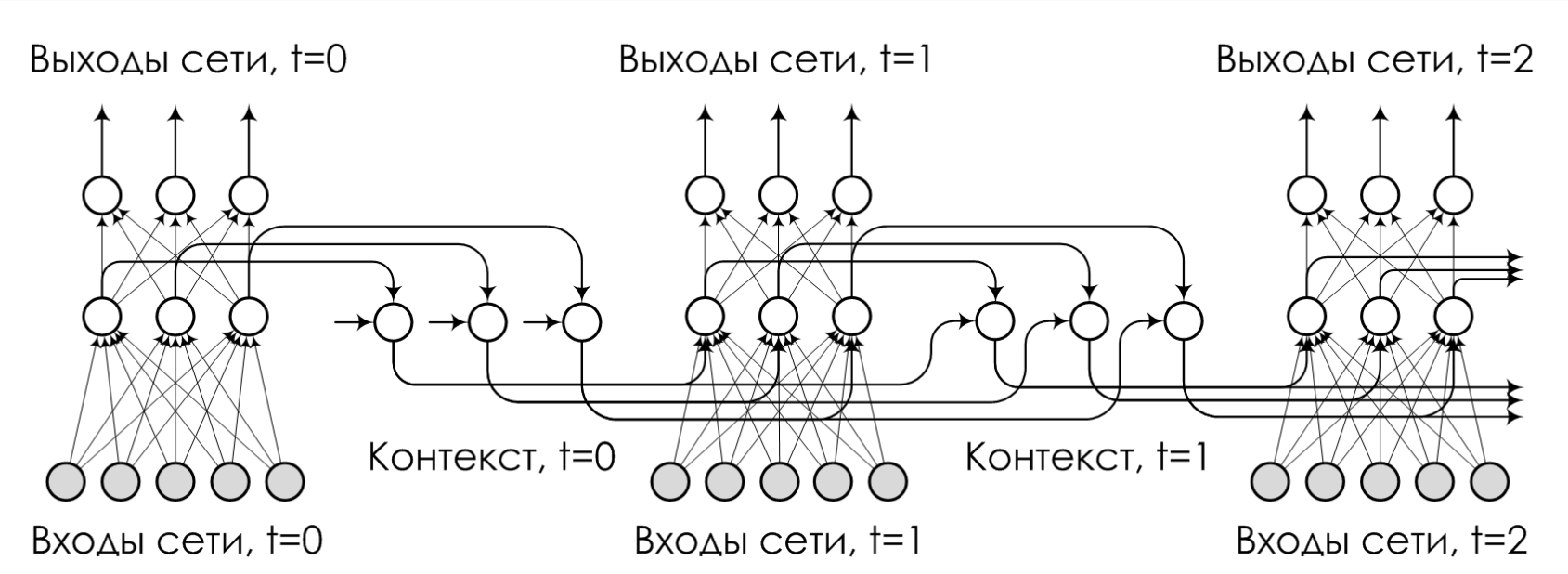
- Сеть Джордана (середина 1980-х):



- Считается первой успешной RNN.

SCRN

- Сеть Элмана (Elman; конец 1980-х):



- Разница в том, что нейроны контекста c_t получают входы со скрытого уровня, а не выходов.
- И нет никаких весов от предыдущих c_{t-1}! То есть веса фиксированы и равны 1.

SCRN

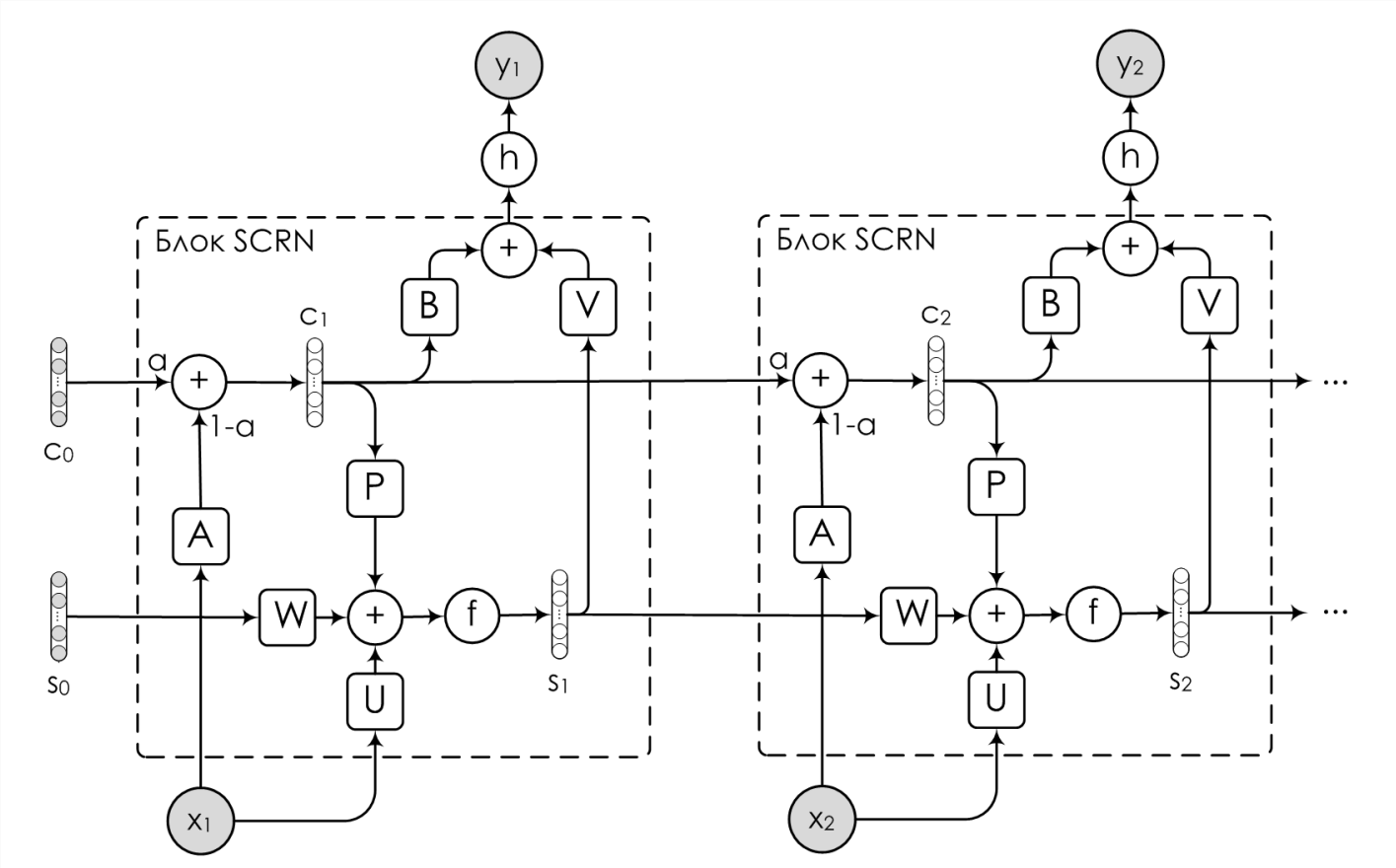
- Это приводит к хорошим долгосрочным эффектам, потому что нет нелинейности между последовательными шагами, и карусель константной ошибки получается по определению:

$$c_t = c_{t-1} + U\mathbf{x}_t.$$

- Идея: можно зафиксировать градиенты, использовав единичную матрицу весов вместо обучаемой W.
- Долгосрочная память тут есть... но обучать очень трудно, потому что градиенты надо возвращать к началу последовательности.

SCRN

- (Mikolov et al., 2014): Structurally Constrained Recurrent Network (SCRN).
- Сочетание двух идей – $\mathbf{s}_t$ с W и c_t с диагональной матрицей рекуррентных весов.



SCRN

- Формально:

$$
\begin{aligned}
c_t &= (1-\alpha)\, A\mathbf{x}_t + \alpha c_{t-1}, \\
\mathbf{s}_t &= f(Pc_t + U\mathbf{x}_t + W\mathbf{s}_{t-1}), \\
\mathbf{y}_t &= h(V\mathbf{s}_t + B\mathbf{s}_t).
\end{aligned}
$$

- SCRN – это просто обычный RNN, где $\mathbf{s}_t$ и c_t в одном векторе, и матрица рекуррентных весов имеет вид

$$
W = \begin{pmatrix} R & P \\ 0 & \alpha \mathbf{I} \end{pmatrix},
$$

РЕГУЛЯРИЗУЕМ w

- Альтернатива: давайте просто регуляризуем W так, чтобы $\det W = 1$.
- Мягкая регуляризация (Pascanu et al., 2013):

$$\Omega = \sum_k \Omega_k = \sum_k \left(\left\| \frac{\frac{\partial E}{\partial \mathbf{s}_{k+1}} \frac{\partial \mathbf{s}_{k+1}}{\partial \mathbf{s}_k}}{\frac{\partial E}{\partial \mathbf{s}_{k+1}}} \right\| - 1 \right)^2.$$

- Жёсткая регуляризация – сделаем W автоматически унитарной (Arjovsky et al., 2015):

$$W = D_3 R_2 F^{-1} D_2 \Pi R_1 F D_1,$$

где D – диагональные матрицы, F – преобразование Фурье, R – отражения, Π – перестановка.

- Кстати, и параметров меньше: теперь только $O(n)$ вместо $O(n^2)$.

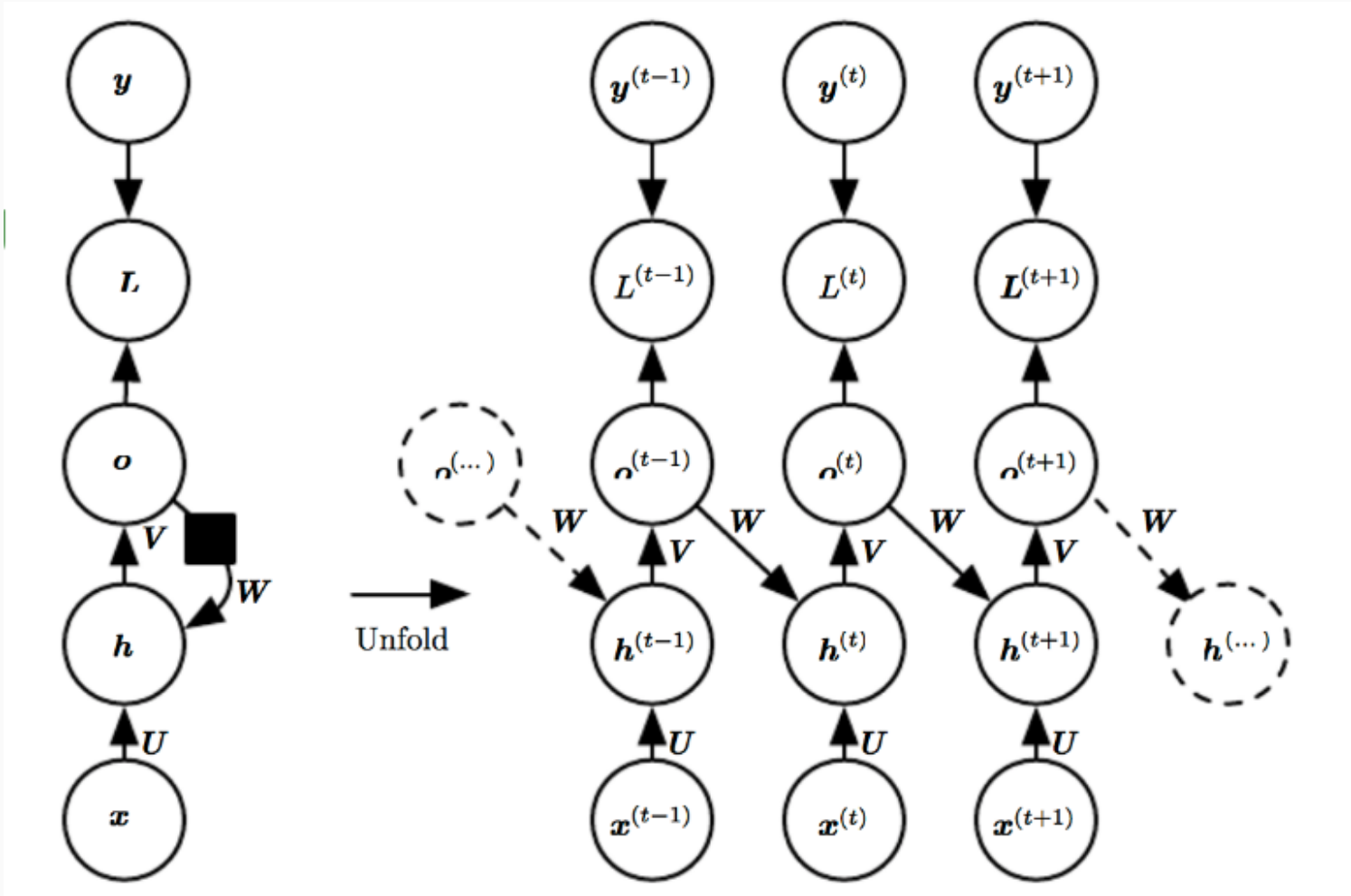
ИНИЦИАЛИЗИРУЕМ w

- И ещё более простой трюк: давайте правильно инициализируем W (Le, Jaitly, Hinton, 2015).
- Рассмотрим RNN с ReLU-активациями на рекуррентных весах (перед h).
- Тогда если W_{hh} – единичная матрица и $\mathbf{b}_h = 0$, скрытое состояние не изменится, градиент протечёт насквозь.
- Давайте так и инициализируем! Часто приводит к серьёзным улучшениям.

Варианты и трюки

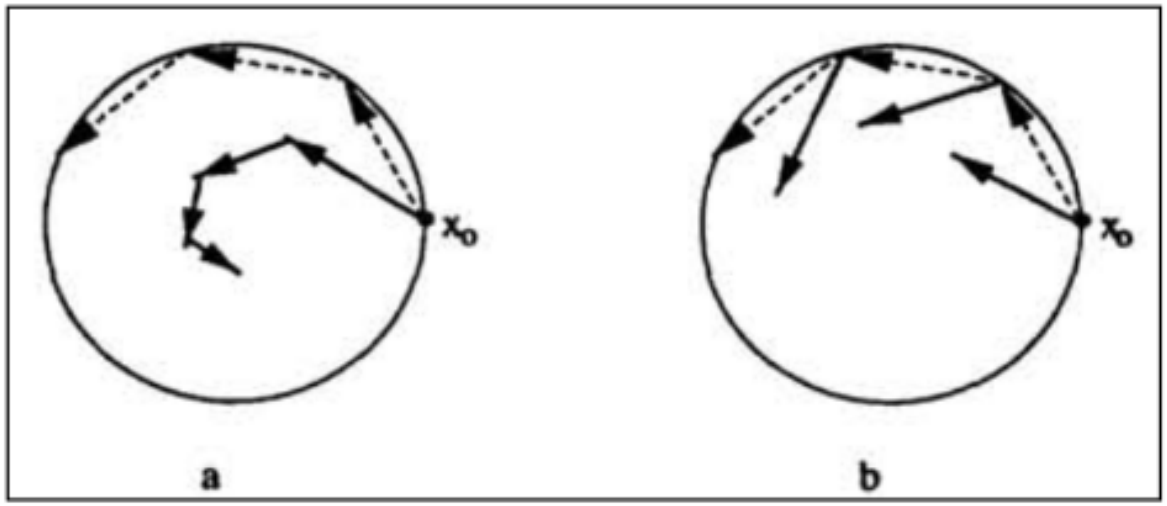
RNN со связями на выходе

- Другой вариант RNN – можно развить идею Джордана и сделать output-to-hidden связи:



RNN со связями на выходе

- Это хуже для моделирования долгосрочных зависимостей.
- Но зато обучение очень простое: подставим вместо выхода правильный ответ!
- Teacher forcing: будем подставлять y_t на каждом шаге, возвращая сеть на правильную траекторию.



RNN со связями на выходе

- А на самом деле это всего лишь максимизация правдоподобия.
- RNN моделирует последовательность как

$$p(y_1, y_2, \dots, y_T) = p(y_1)p(y_2 \mid y_1)p(y_3 \mid y_1, y_2) \dots p(y_T \mid y_{T-1}, \dots, y_1).$$

- И если это раскрыть:

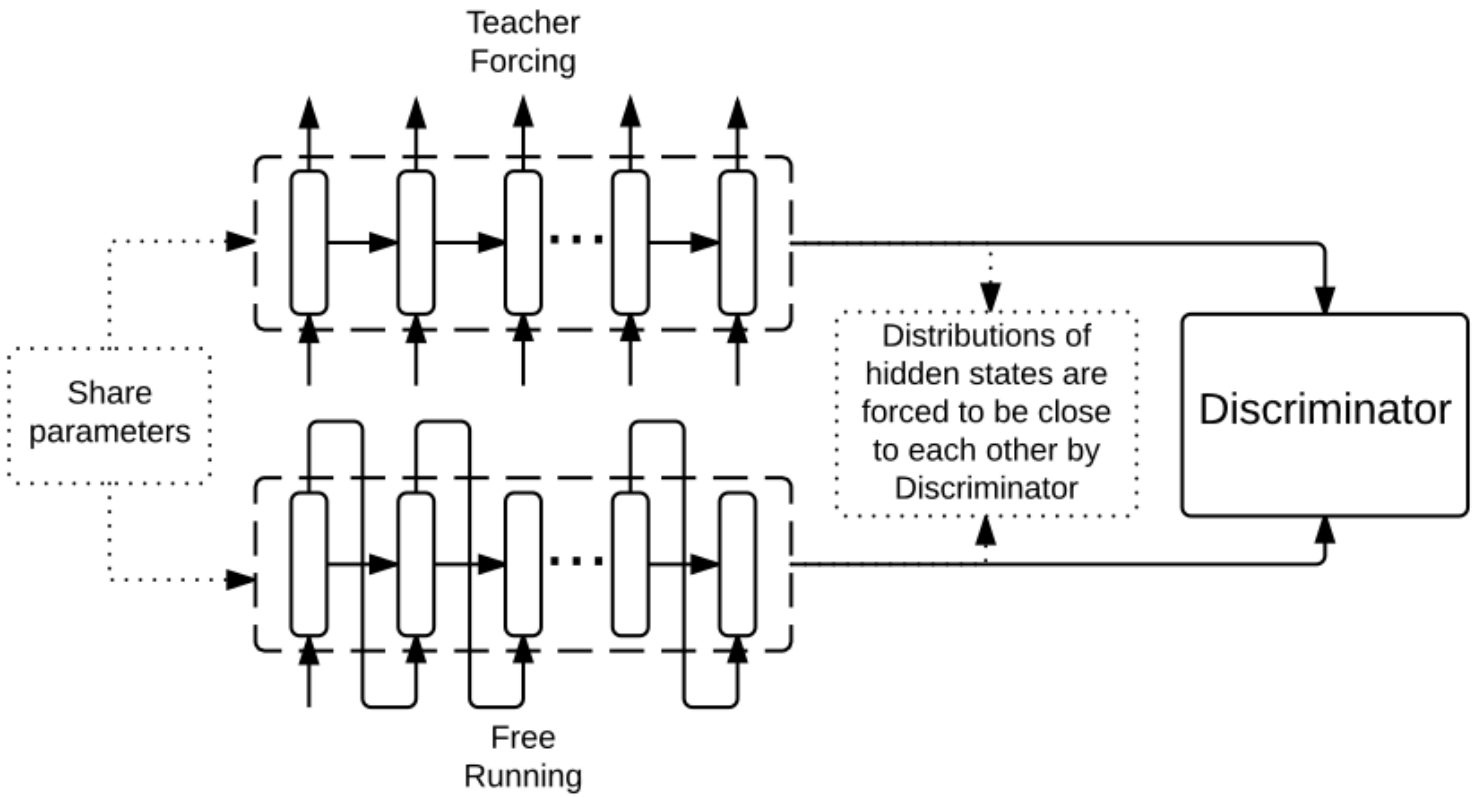
$$p(y_1, y_2 \mid \mathbf{x}_1, \mathbf{x}_2) = p(y_2 \mid y_1, \mathbf{x}_1, \mathbf{x}_2)p(y_1 \mid \mathbf{x}_1, \mathbf{x}_2),$$

мы видим, что y_1 надо подставить, чтобы перейти к y_2 – это и есть teacher forcing.

- Если есть и output-to-hidden, и hidden-to-hidden, то можно совместить teacher forcing с BPTT.

RNN со связями на выходе

- Проблема: стоит учителю отвернуться, и сеть будет плохо себя вести. Ошибки накапливаются.
- (Lamb, Goyal et al., 2016): Professor Forcing на основе GAN-подобных идей (подробности позже).



НОРМАЛИЗАЦИЯ ПО УРОВНЮ

- Вспомним batch normalization: очень хорошая штука, борется со сдвигом в переменных.
- Но применить batchnorm к RNN не получается:
  - теперь «уровень» – это шаг последовательности;
  - и получается, что веса общие, а статистики надо хранить по отдельности;
  - плохо, если последовательности разной длины;
  - совсем плохо, если при применении будет длиннее, чем в обучающей выборке.
- Что делать?

НОРМАЛИЗАЦИЯ ПО УРОВНЮ

- Нормализация по уровню (layer normalization; Ba, Kiros, Hinton, 2016): будем просто усреднять по одному уровню.
- Раньше было $\mathbf{a}_t = W_{hh} h_{t-1} + W_{xh} \mathbf{x}_t$.
- Теперь будет

$$h_t = f\left[\frac{\mathbf{g}}{\sigma_t} \odot (\mathbf{a}_t - \mu_t) + \mathbf{b}\right],$$

$$\text{где } \mu_t = \frac{1}{H}\sum_{i=1}^{H} \sigma_{it}, \quad \sigma_t = \sqrt{\frac{1}{H}\sum_{i=1}^{H} (a_{it} - \mu_t)^2},$$

  а $\mathbf{g}$ и $\mathbf{b}$ — параметры слоя нормализации, как и раньше.
- Это может заметно улучшить результаты рекуррентной сети.

ЧТО ДЕЛАТЬ С RNN НА ПРАКТИКЕ

ПРИНЦИПЫ

- RNN имеют довольно простую общую структуру: уровни LSTM или GRU.
- Все они выдают последовательность выходов, кроме, возможно, верхнего.
- Дропаут и batchnorm между слоями, а на рекуррентных связях надо аккуратно (потом поговорим).
- Слоёв немного; больше 3-4 трудно, 7-8 сейчас максимум.

ПРИНЦИПЫ

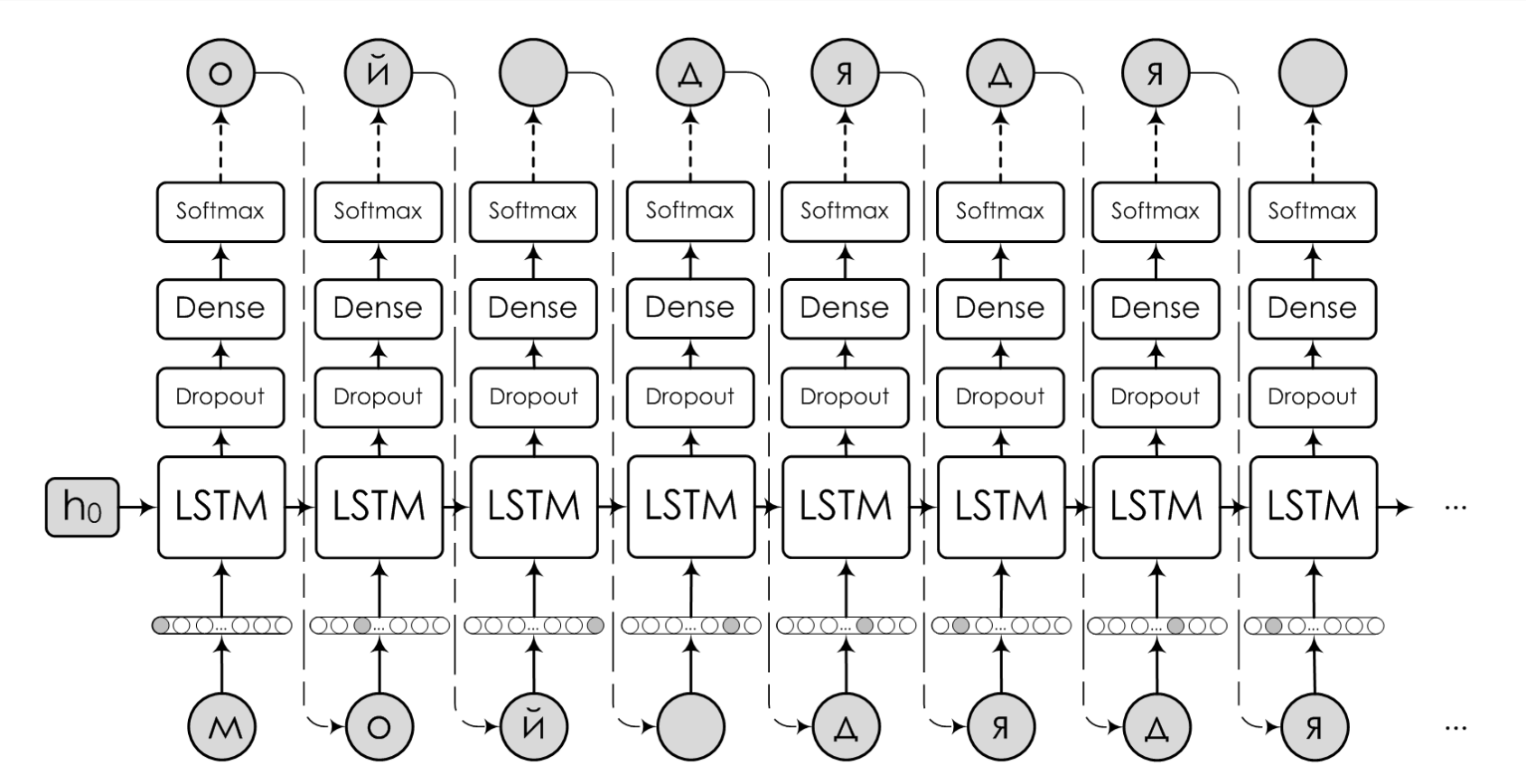
- Важный трюк: *skip-layer connections*, как residual, только проще. Добавляем выходы предыдущих слоёв «через один» или «через два», просто конкатенацией.
- RNN, сохраняющие состояние: состояния с одного мини-батча переиспользуются как начальные для следующего. Градиенты в BPTT останавливаются, но состояния остаются. В *Keras* легко: `stateful=True`.
- Начнём с простого примера анализа временных рядов...

ПРИМЕР: ПОРОЖДЕНИЕ ТЕКСТА С RNN

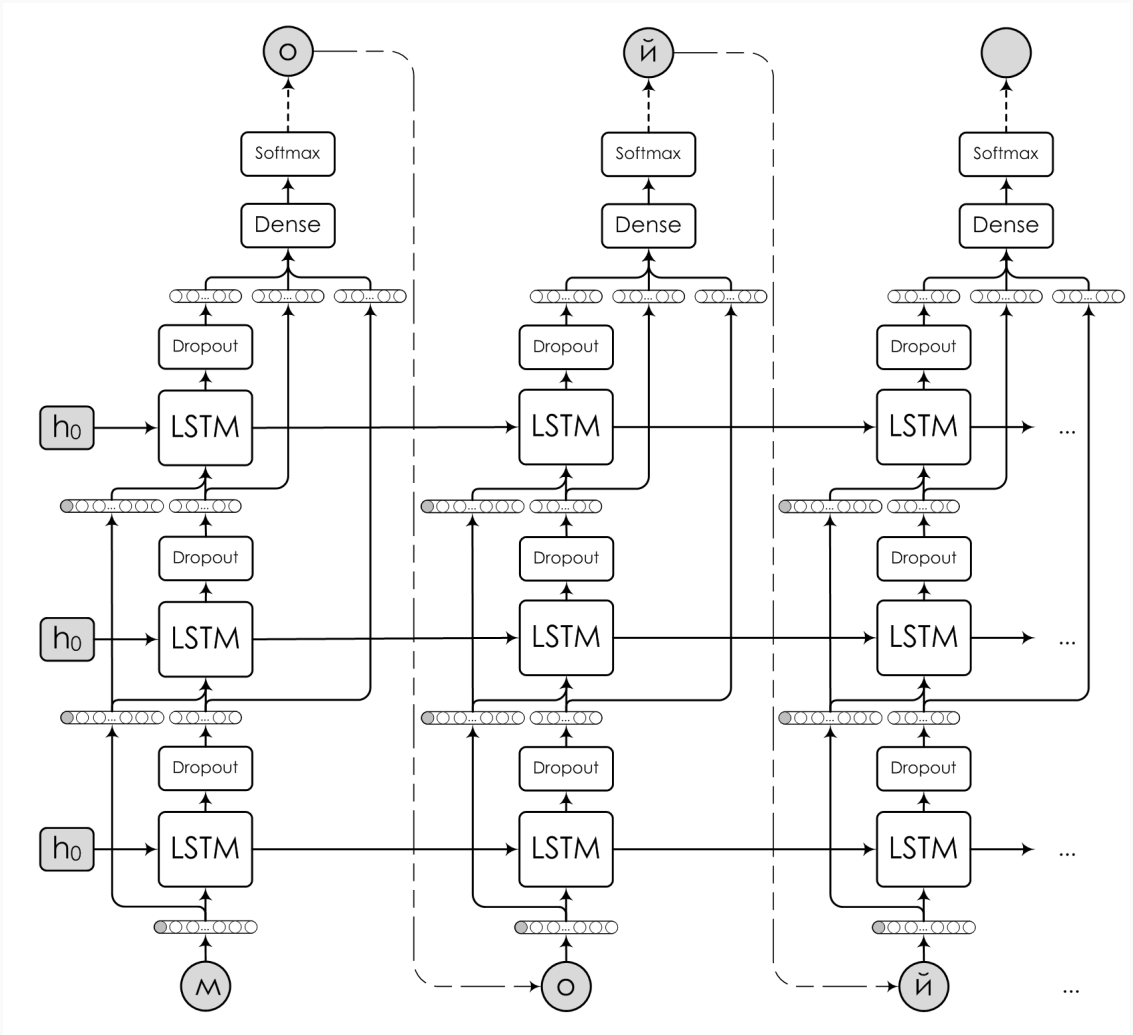
- Языковые модели – это естественное прямое приложение к NLP.
- Первая идея – давайте просто обучим последовательность слов через RNN/LSTM.
- О языке будем говорить позже, а пока любопытно, что можно обучить RNN порождать интересные последовательности даже просто символ за символом.
- Karpathy, «The Unreasonable Effectiveness of Neural Networks»; знаменитый пример из (Sutskever et al. 2011):

  The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger...
- Это, конечно, всего лишь эффекты краткосрочной памяти, никакого «понимания».

SIMPLE LSTM-BASED ARCHITECTURE



SLIGHTLY LESS SIMPLE LSTM-BASED ARCHITECTURE



ПОРОЖДЕНИЕ ТЕКСТА С RNN

- Можно менять diversity (температуру сэмплирования), получая разные абсурдистские тексты. Случайный пример с seed «*обещал, на рождество, но спустя семь лет. имя глав*».
- Низкий diversity – монолог Лаки:

обещал, на рождество, но спустя семь лет. имя главного командования в составе советского союза с 1976 года. после проведения в 1992 году в составе советского союза (1977). в 1967 году в составе советского союза состоялся в 1952 году в составе советско-финской войны 1877 - 1877 годов. в 1966 году в составе советского союза с 1965 года по 1977 год...

- Средний diversity – что-то более разумное:

обещал, на рождество, но спустя семь лет. имя главного рода собственно вновь образовалась в россии и народном состоянии. после присказа с постановлением союза писателей россии и генеральной диссертации о спортивном училище с 1980 года. в 1970-х годах был основан в составе комитета высшего совета театра в польши. в 1957 - 1962 годах - начальник батальона сан-аухаров...

- Высокий diversity – хлебниковская заумь:

обещал, на рождество, но спустя семь лет. имя главы философии пововпели nol-lнози - врайу-7 на луосече. человеческая восстания покторов извоенного чомпде и э. дроссенбурга, ... карл уним-общекрипских. эйелем хфечак от этого списка сравнивала имущно моря в юнасториансический индристское носительских женатов в церкви испании....

ПОРОЖДЕНИЕ ТЕКСТА С RNN

- Ещё пример – «12 стульев», 3 слоя LSTM размерности 128.
- Низкий diversity:

– вы думаете, что он подвергается опасности? не понимаете на девушки и со всего большого секретара. на поставитель из столики с колодции под собой по столовом под нарипальное одного обедать вы получить стулья. но все не собирался. под водой под события не подошел к двери. он серебренной при столики под водом воробьяниновской порочение и подошел к стулом.

- Средний diversity:

– что это значит?– спросил ипполит матвеевич, вспоминая только подкладка, идиость выкрасть, что уже совершенно всего упасы, по рексе оборанный решали на ним ответственное колоно горячи облиганта ветерность ”правосудель” за стояли пределицу и из одобрания из на порахнитостью. но кричался воему тогу. его не смотрел ордеров с мы толстений принимать выдержание то преходитель.

- Высокий diversity:

– ну, и я вы умоли полтуча,– сказал остап, нади гадалкий во столбор не черта не надо предражало. ответил золотый и стулья и нов. срековое зарабоварил сто оспастук, и обычно, и строи тираживым господура моя животую столу, почто не уличного беспарные такиме судьберского есть денегальный извер.

ПОРОЖДЕНИЕ ТЕКСТА С RNN

- Последний пример – те же 3 слоя по 128, «Евгений Онегин».
- Низкий diversity – учим стихи наизусть:

но вот уж близко. перед ними
уж белокаменной москвы
как жар, крестами золотыми
горят старинные главы.

- Средний diversity – цитируем большими кусками:

не правда ль? вам была не новость
смиренной девочки, поврама
он был любим... по крайней мере
так думал он на супруге.

- Высокий diversity – опять заумь:

простой живеть по полном в,
бал уж канит; три несала
до глаза подерень преданьем
поедет, смертаю себя.

ПОРОШКИ

А вот порошки из seq2seq-архитектуры на достаточно маленьком датасете (спасибо Артуру Кадурину):

заходит к солнцу отдаётесь
что он летел а может быть
и вовсе не веду на стенке
на поле пять и новый год
и почему то по башке
в квартире голуби и боли
и повзрослел и умирать

страшней всего когда ты выпил
без показания зонта

однажды я тебя не вышло
и ты

я захожу в макдоналисту
надену отраженный дождь
под ужин почему местами
и вдруг подставил человек

ты мне привычно верил крышу
до дна

я подползает под кроватью
чтоб он исписанный пингвин
и ты мне больше никогда
но мы же после русских классик
барто солдаты для любви

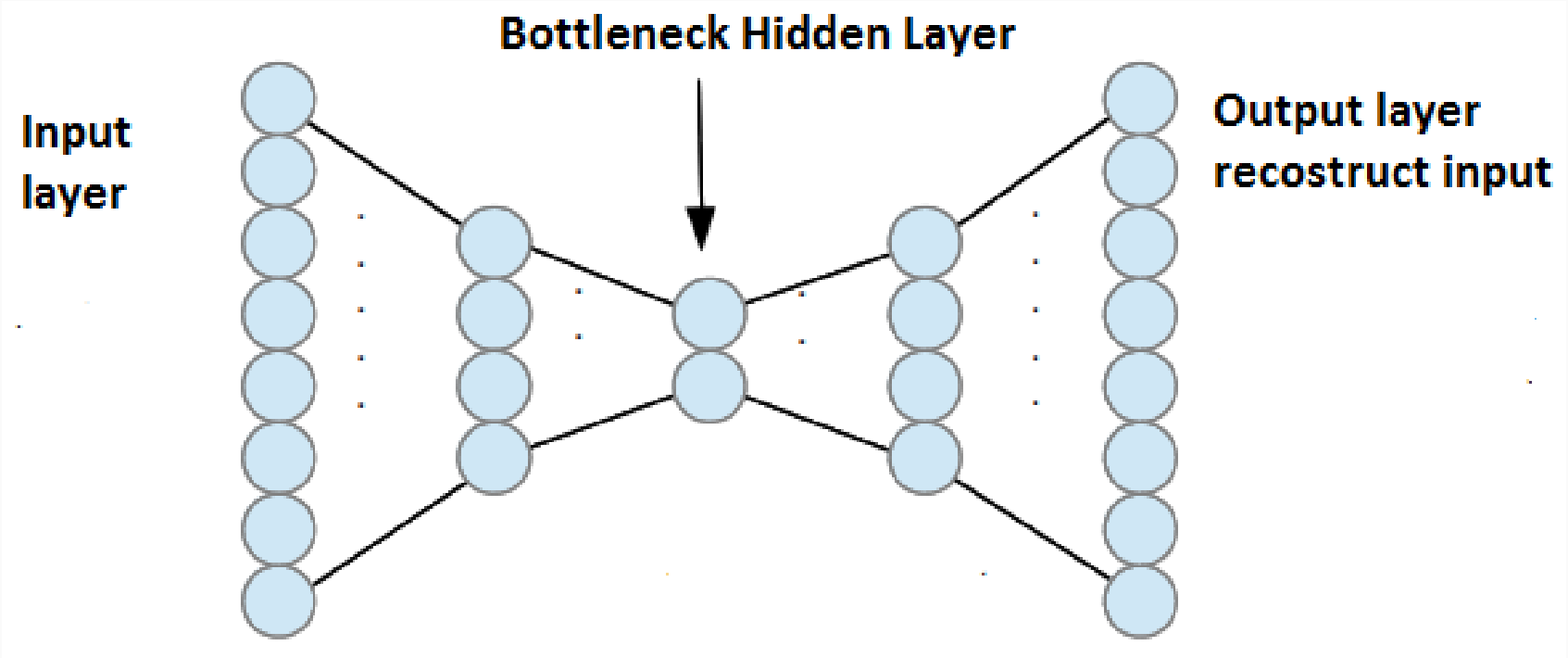
АВТОКОДИРОВЩИКИ

Извлечение признаков

- Мы говорили об извлечении признаков.
- Но все наши сети до сих пор работали с учителем.
- Как извлечь признаки из неразмеченных данных?
- Например, просто датасет фотографий или рукописных цифр, безо всяких меток.

АВТОКОДИРОВЩИКИ

- *Автокодировщики* (autoencoders) появились в (Rumelhart, Hinton, Williams 1986).
- Очень простая и мощная идея — давайте восстановим вход:



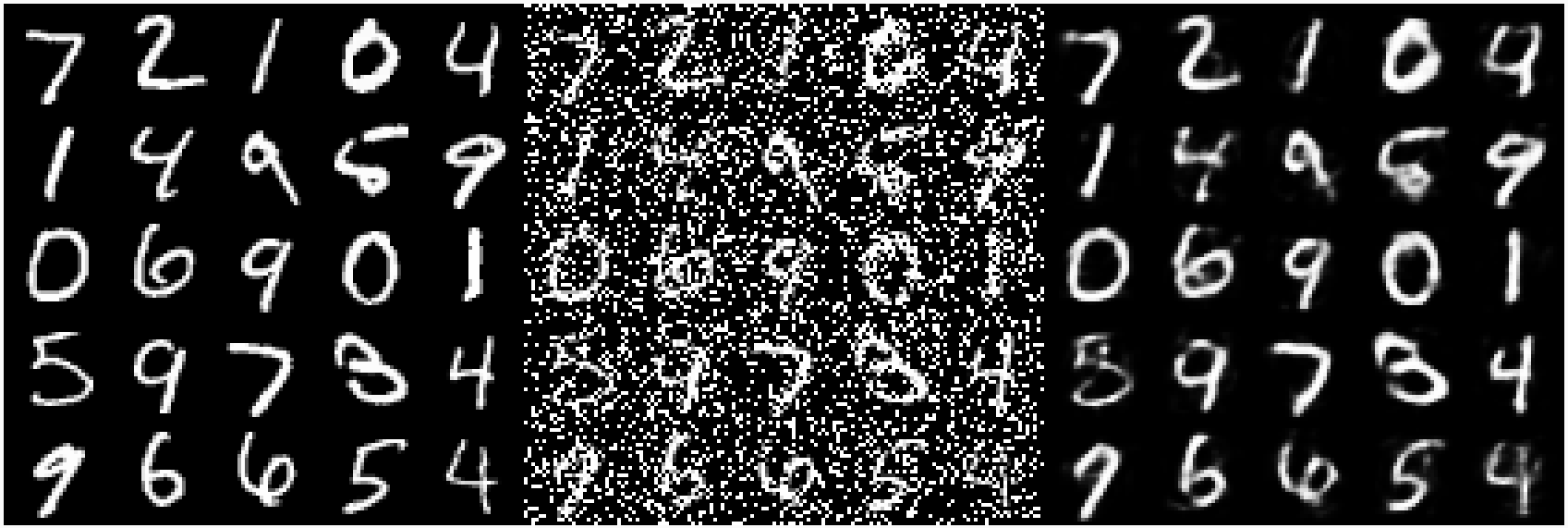
- Почему нельзя просто скопировать вход в выход?

АВТОКОДИРОВЩИКИ

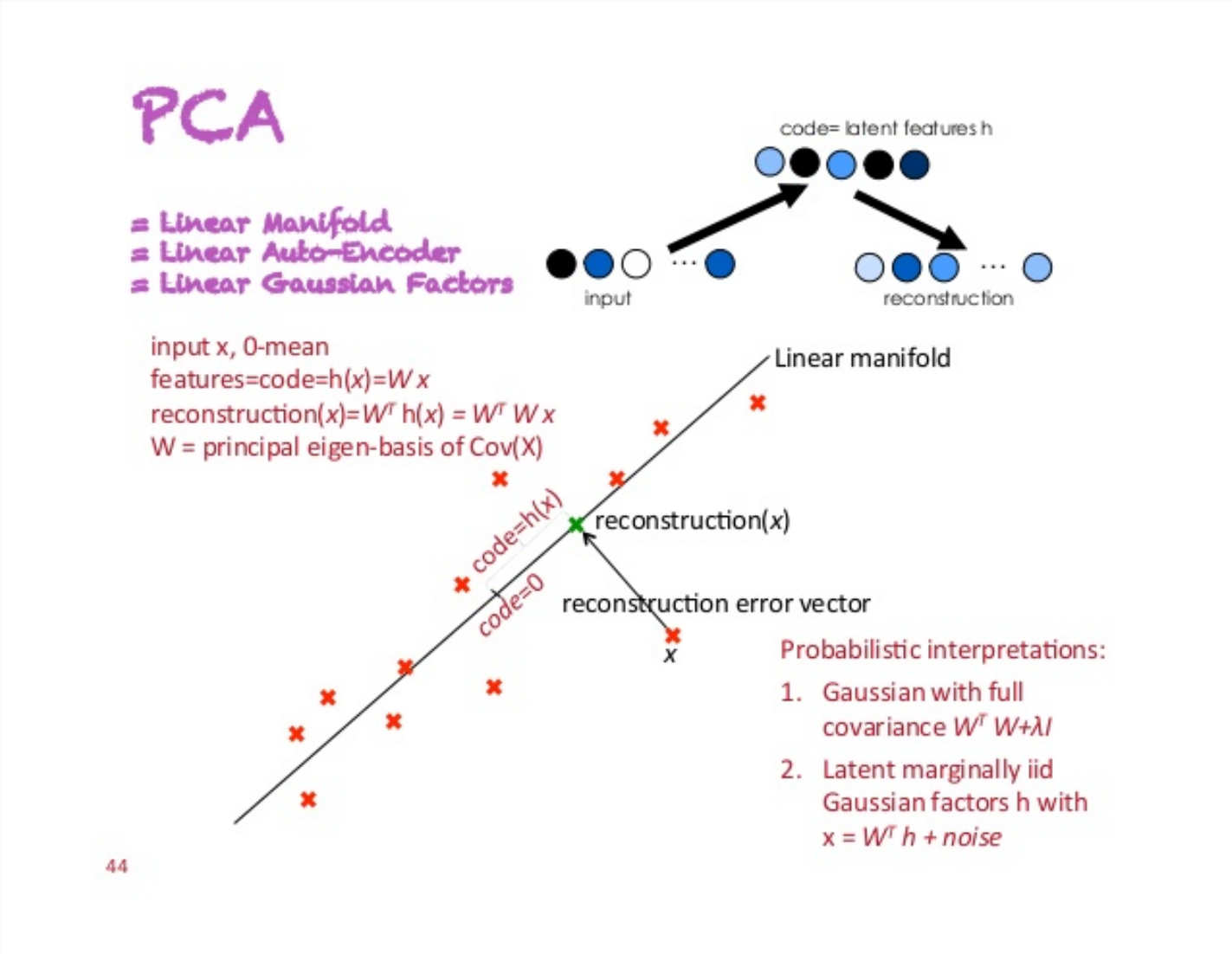
- Undercomplete vs. overcomplete autoencoders.
- Для undercomplete AE идея – *обучение многообразий* (manifold learning): ищем данные вблизи многообразия малой размерност.
- Хотим научиться “разворачивать” его, получив сжатое представление данных; фактически метод снижения размерности.
- Overcomplete AE может скопировать, но мы его регуляризуем; дропаут очень помогает.

АВТОКОДИРОВЩИКИ

- А можно и ещё сильнее, чем дропаут: *шумоподавляющие автокодировщики* (denoising autoencoders).

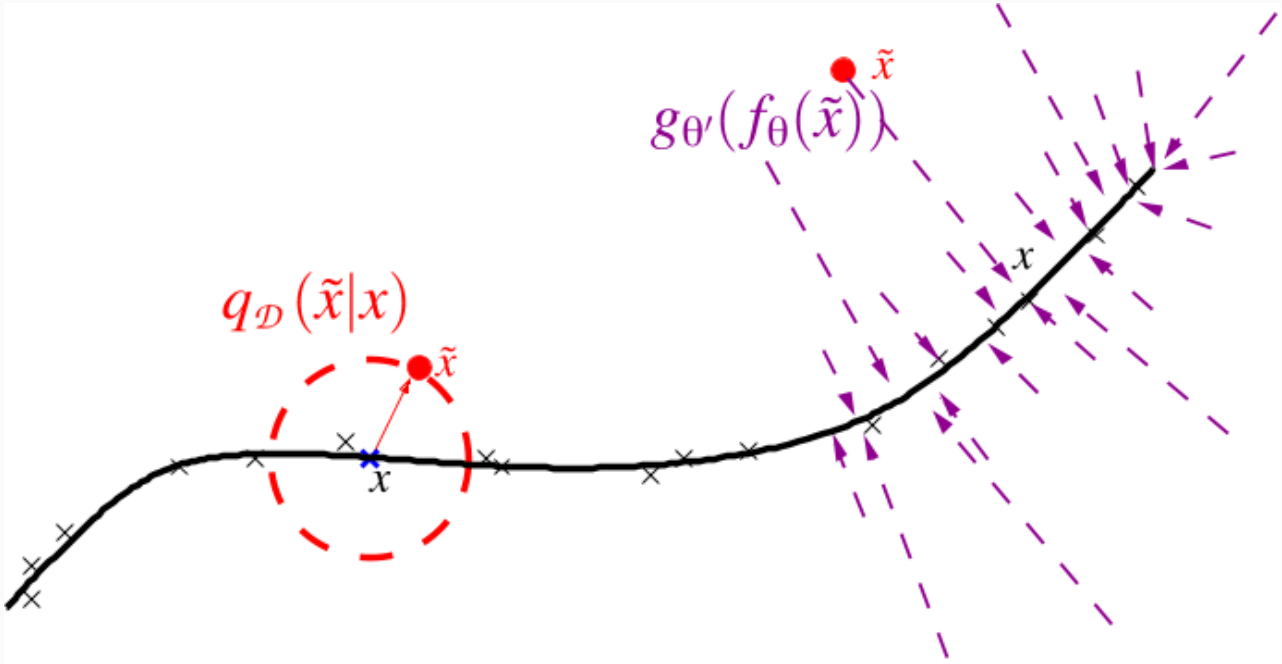


- Можно сравнить с методом главных компонент (PCA):



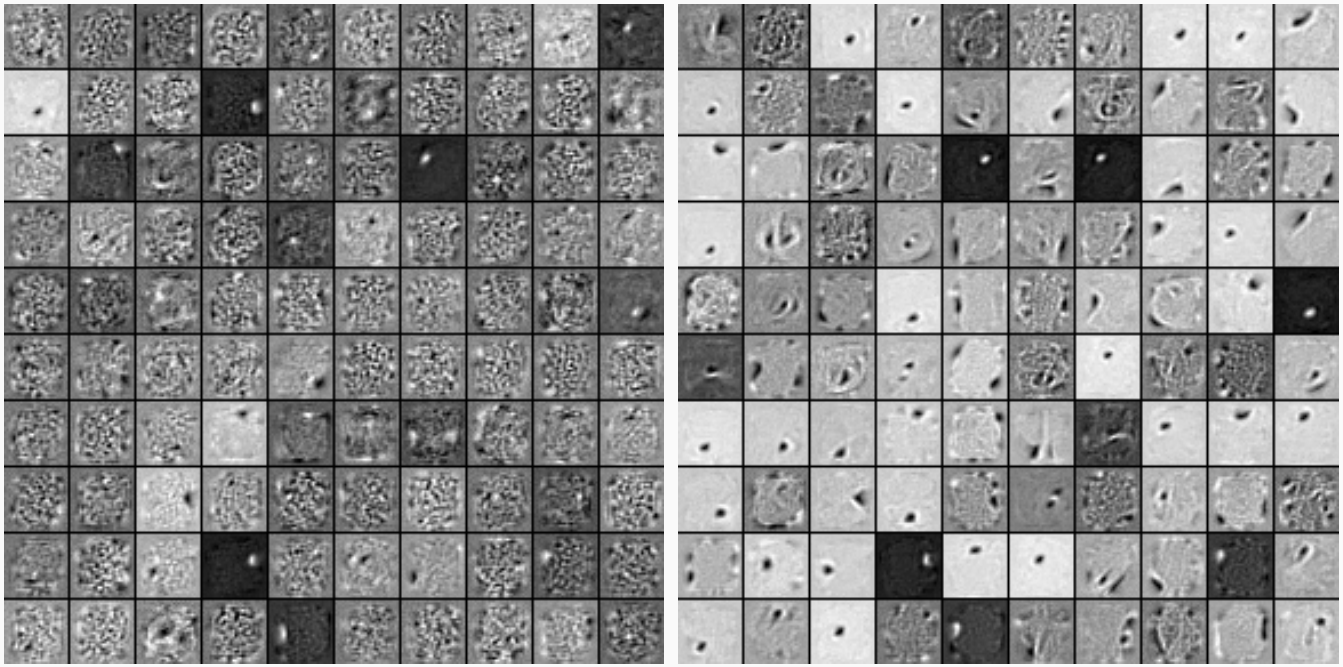
АВТОКОДИРОВЩИКИ

- Шумоподавляющий автокодировщик:



АВТОКОДИРОВЩИКИ

- Фильтры, обученные на MNIST без шума и с шумом:



АВТОКОДИРОВЩИКИ

- Разреженный автокодировщик: опять что-то вроде энергоэффективного человеческого мозга.
- Как добавить разреженности?
- Давайти просто скажем, что только доля ρ нейронов должна быть активной, и добавим регуляризатор об этом.
- Но что это может быть за регуляризатор?

АВТОКОДИРОВЩИКИ

- Регуляризатор — расстояние Кульбака–Лейблера между собственно активациями и монеткой с вероятностью ρ:

$$\mathrm{KL}\left(\rho \| \hat{\rho}\right) = \rho \log \frac{\rho}{\hat{\rho}}.$$

- Пример: `https://www.youtube.com/watch?v=qAyJkITp4AI`

СПАСИБО!

Спасибо за внимание!