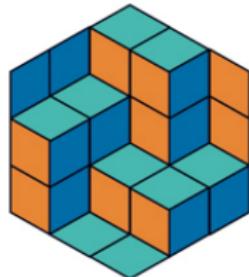


Лекция 3. Рекуррентные нейронные сети

Александр Юрьевич Авдюшенко

МКН СПбГУ

3 марта 2022



Факультет
математики
и компьютерных
наук
СПбГУ

Пятиминутка

- ▶ Выпишите несколько названий методов оптимизации нейронных сетей
- ▶ Опишите пару методов регуляризации при обучении нейронных сетей
- ▶ Нарисуйте (или запишите), как устроен блок skip-connection

Недостатки сверточных нейронных сетей

или зачем нужны рекуррентные

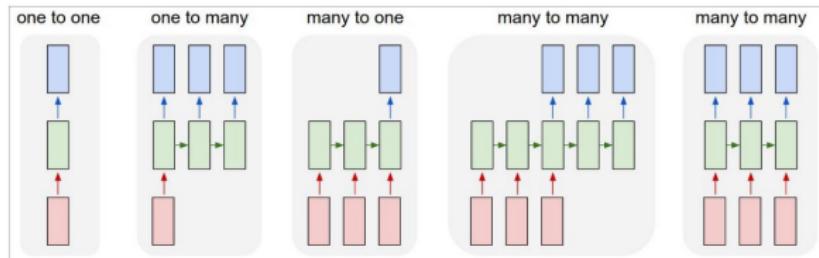
- ▶ на входе только вектора фиксированной размерности (например, изображения 28×28)
- ▶ на выходе тоже размерность фиксирована (например, вероятности 1000 классов)
- ▶ фиксированное число вычислительных шагов (т.е. архитектура сети)

Недостатки сверточных нейронных сетей

или зачем нужны рекуррентные

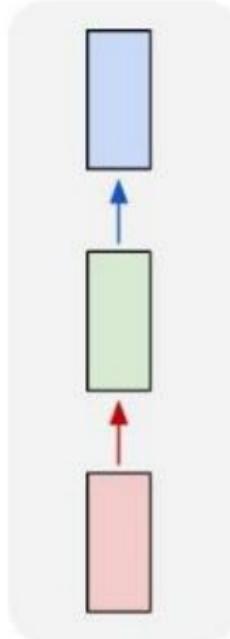
- ▶ на входе только вектора фиксированной размерности (например, изображения 28×28)
- ▶ на выходе тоже размерность фиксирована (например, вероятности 1000 классов)
- ▶ фиксированное число вычислительных шагов (т.е. архитектура сети)

A. Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks

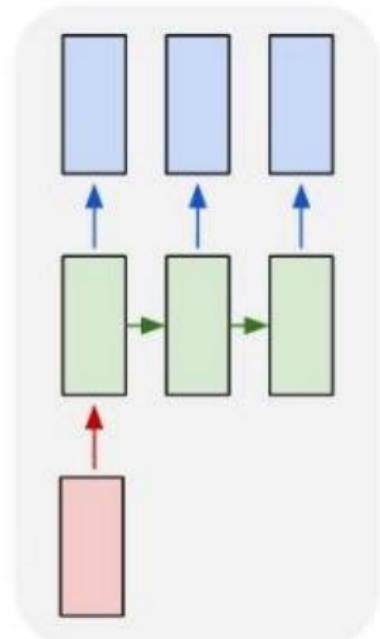


Архитектуры рекуррентных сетей

one to one



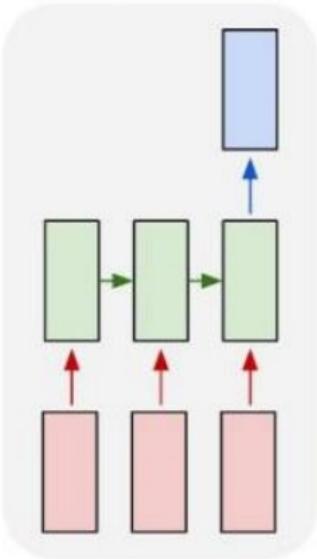
one to many



Vanilla Neural Networks

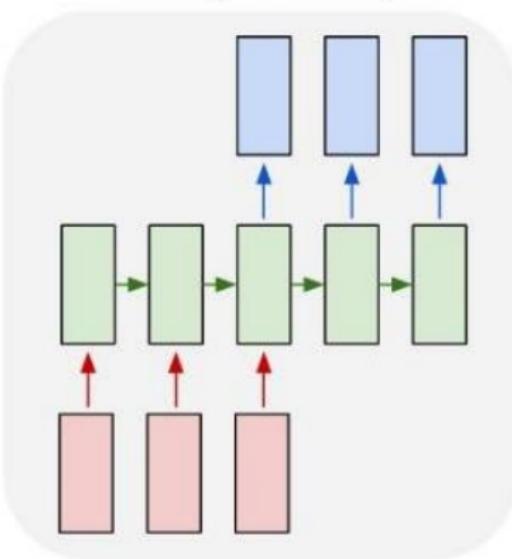
Image Captioning
image → (sequence of words)

many to one



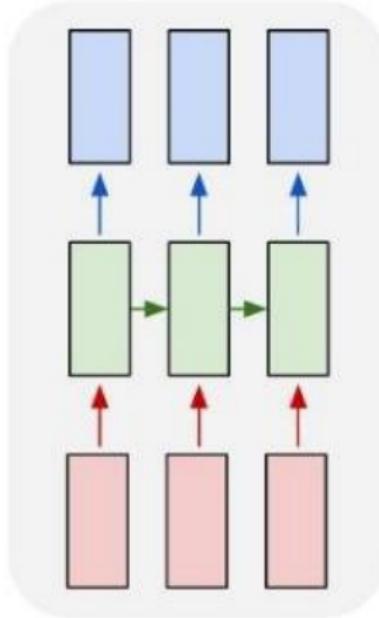
Sentiment Classification
(sequence of words) → sentiment

many to many



Machine Translation
(seq of words) → (seq of words)

many to many



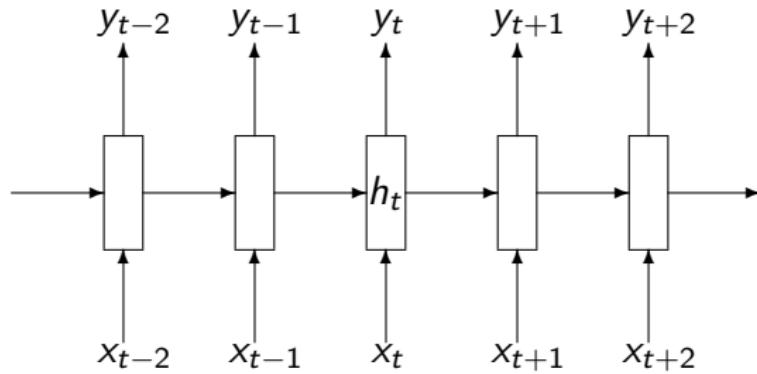
Video Classification
(on frame level)

Последовательная обработка фиксированного входа

Последовательная генерация фиксированного выхода

K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, D. Wierstra.
DRAW: A Recurrent Neural Network For Image Generation

Рекуррентная нейронная сеть (RNN)



Рекуррентная нейронная сеть (RNN)

Обрабатываем последовательность векторов x **одной и той же** функцией с параметрами:

$$h_t = f_W(h_{t-1}, x_t)$$

f_W — функция, параметризованная W

x_t — очередной входной вектор

h_t — скрытое состояние

Рекуррентная нейронная сеть (RNN)

Обрабатываем последовательность векторов x **одной и той же** функцией с параметрами:

$$h_t = f_W(h_{t-1}, x_t)$$

f_W — функция, параметризованная W

x_t — очередной входной вектор

h_t — скрытое состояние

Вопрос

Какую функцию взять в качестве f_W ?

Простейшая (vanilla) рекуррентная нейронная сеть

$$h_t = f_W(h_{t-1}, x_t)$$

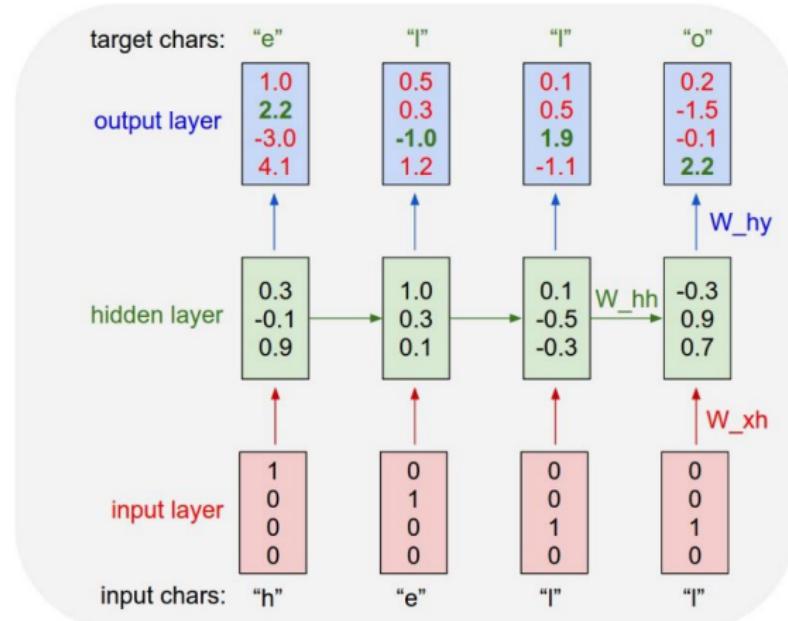
В качестве функции f_W задаём линейное преобразование с нелинейной «сигмоидой» по компонентам:

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

$$y_t = W_{hy} h_t$$

Пример модели на уровне символов

Весь словарь из четырёх букв: $[h, e, l, o]$



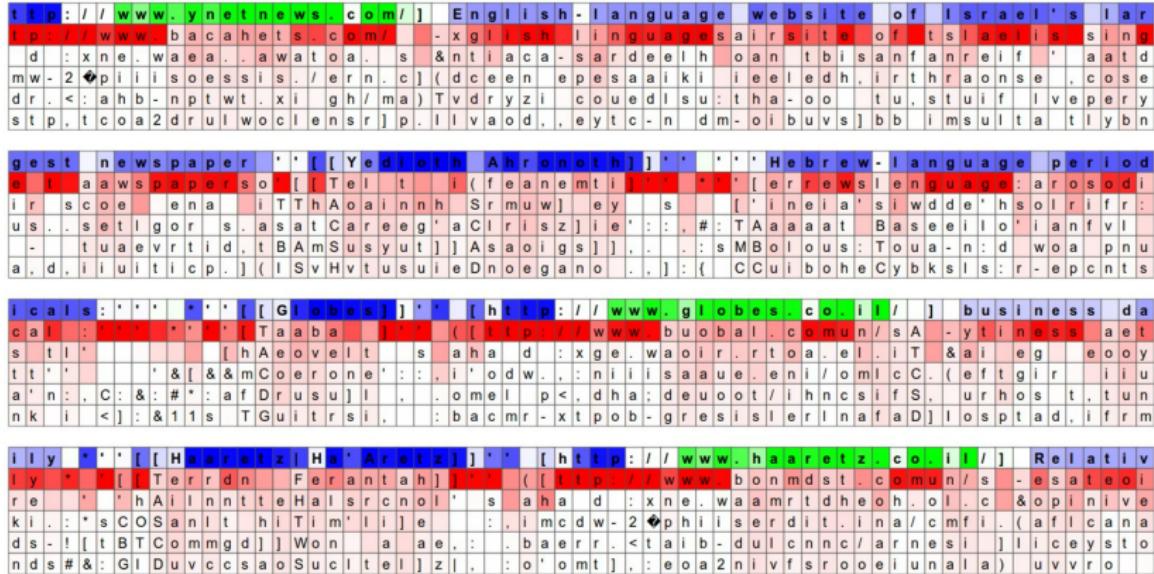
к значениям выходного слоя для получения функции потерь
ещё применяется **Softmax**

Демо

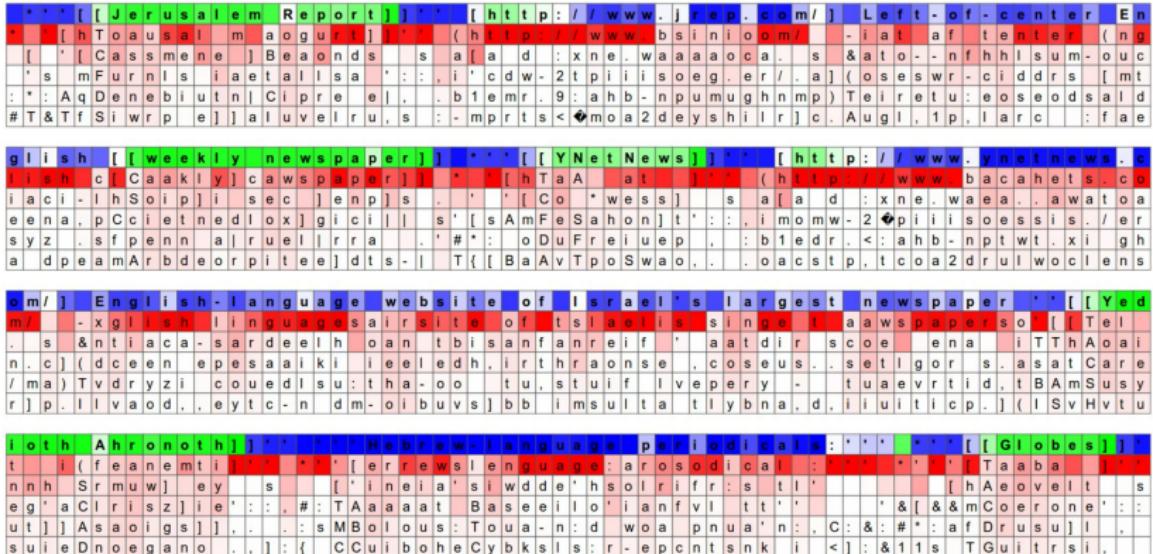
Реализация на numpy от Karpathy

Разбираемся с кодом!

Как это работает?



The neuron highlighted in this image seems to get very excited about URLs and turns off outside of the URLs. The LSTM is likely using this neuron to remember if it is inside a URL or not.



The highlighted neuron here gets very excited when the RNN is inside the `[[]]` markdown environment and turns off outside of it. Interestingly, the neuron can't turn on right after it sees the character "`"`", it must wait for the second "`"`" and then activate. This task of counting whether the model has seen one or two "`"`" is likely done with a different neuron.

Cell sensitive to position in line:

The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action--the one Kutuzov and the general mass of the army demanded--namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all--carried on by vis inertiae--pressed forward into boats and into the ice-covered water and did not, surrender.

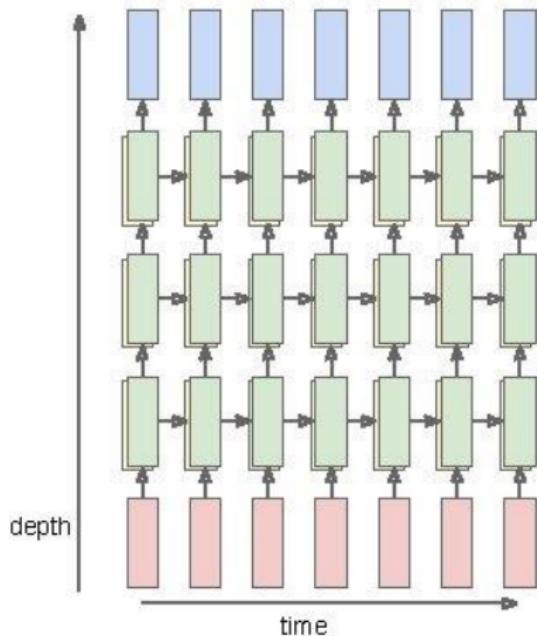
Cell that turns on inside quotes:

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

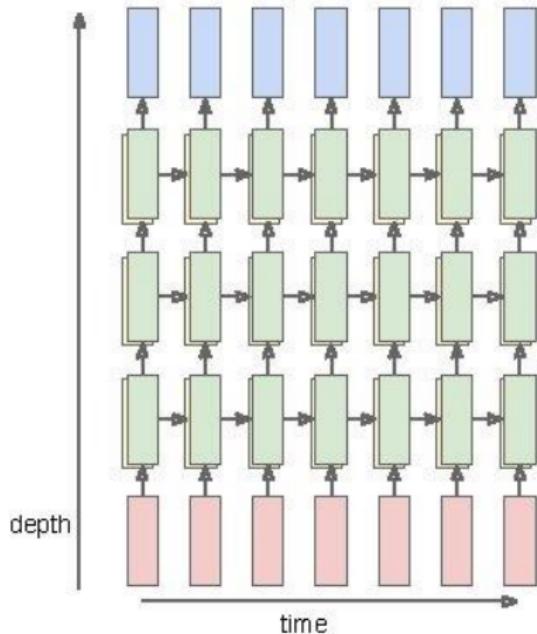
Глубокие рекуррентные сети

$$h_t^\ell = \tanh W^\ell \begin{pmatrix} h_t^{\ell-1} \\ h_{t-1}^\ell \end{pmatrix}$$
$$h \in \mathbb{R}^n, \quad W^\ell[n \times 2n]$$



Глубокие рекуррентные сети

$$h_t^\ell = \tanh W^\ell \begin{pmatrix} h_t^{\ell-1} \\ h_{t-1}^\ell \end{pmatrix}$$
$$h \in \mathbb{R}^n, \quad W^\ell[n \times 2n]$$



Вопрос

Какая главная проблема *vanilla RNN*?

Долгая кратковременная память (Long short-term memory, LSTM)

$$W^\ell[4n \times 2n]$$

$$\begin{pmatrix} i \\ f \\ o \\ c'_t \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^\ell \begin{pmatrix} h_t^{\ell-1} \\ h_{t-1}^\ell \end{pmatrix}$$

$$c_t^\ell = f \odot c_{t-1}^\ell + i \odot c'_t$$
$$h_t^\ell = o \odot \tanh(c_t^\ell)$$

⊗ - покомпонентное произведение

Мотивация и схема LSTM

Сеть должна долго помнить контекст. Какой именно сеть выучивает сама. Для этого вводится вектор c_t — вектор состояния сети в момент t .

$$c'_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_{c'}) \quad \text{candidate cell state}$$

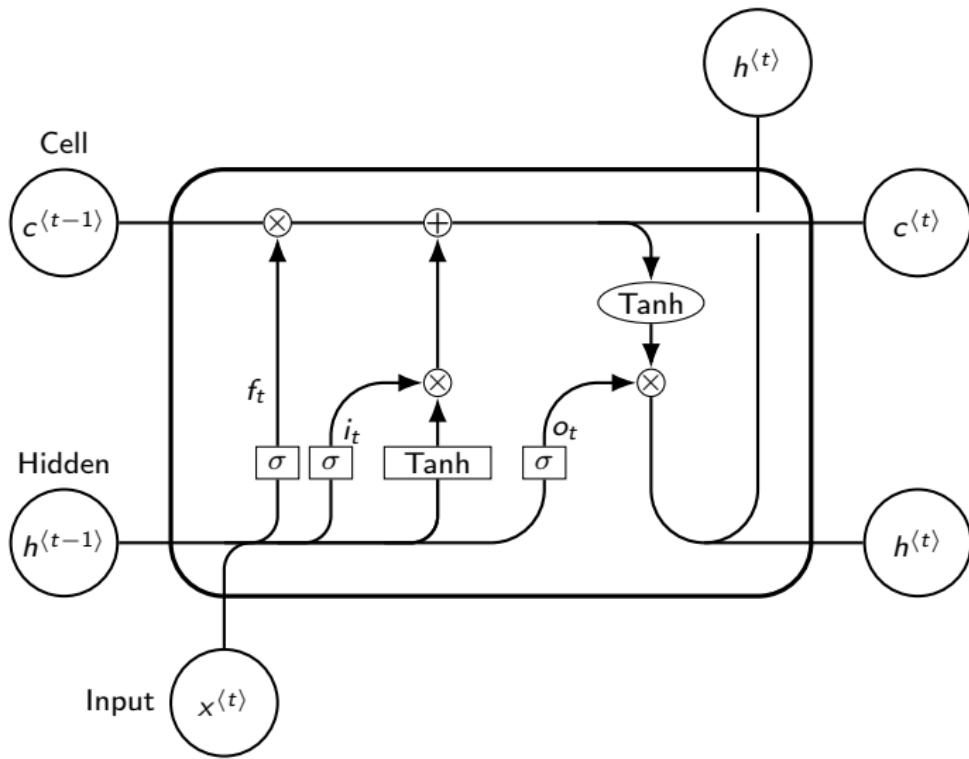
$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad \text{input gate}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad \text{forget gate}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad \text{output gate}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c'_t \quad \text{cell state}$$

$$h_t = o_t \odot \tanh(c_t) \quad \text{block output}$$



LSTM: forget gate

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

Пример: удаление пола действующего лица при генерации текста

LSTM: input/candidate gates

$$c'_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_{c'}) \quad \text{candidate cell state}$$
$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad \text{input gate}$$

Пример: добавление пола действующего лица при генерации текста

LSTM: обновление состояния ячейки памяти

$$c_t = f_t \odot c_{t-1} + i_t \odot c'_t \quad (\text{cell state})$$

Новое состояние c_t получается суммой предыдущего состояния c_{t-1} с фильтром f_t и вектора значений-кандидатов c'_t с фильтром i_t

LSTM: генерация выхода

$$\begin{aligned} o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) && \text{output gate} \\ h_t &= o_t \odot \tanh(c_t) && \text{block output} \end{aligned}$$

GRU: Gated Recurrent Unit

$$u_t = \sigma(W_{xu}x_t + W_{hu}h_{t-1} + b_u)$$

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$

$$h'_t = \tanh(W_{xh'}x_t + W_{hh'}(r_t \odot h_{t-1}))$$

$$h_t = (1 - u_t) \odot h'_t + u_t \odot h_{t-1}$$

Используется только h_t , вектор c_t не вводится.

Фильтр обновления (update gate) вместо входного и забывающего.

Фильтр перезагрузки (reset gate) определяет, какую часть памяти перенести дальше с предыдущего шага.

Резюме

- ▶ Рекуррентные нейронные сети — простой, мощный и гибкий подход решения различных задач машинного обучения
- ▶ Vanilla RNN просты, но всё-таки недостаточно хороши
- ▶ Поэтому нужно использовать LSTM или GRU
- ▶ Зануление градиентов предотвращает LSTM
- ▶ От «взрыва градиентов» помогает clipping
- ▶ Необходимо более глубокое понимание, как теоретическое, так и практическое — много открытых вопросов

Резюме

- ▶ Рекуррентные нейронные сети — простой, мощный и гибкий подход решения различных задач машинного обучения
- ▶ Vanilla RNN просты, но всё-таки недостаточно хороши
- ▶ Поэтому нужно использовать LSTM или GRU
- ▶ Зануление градиентов предотвращает LSTM
- ▶ От «взрыва градиентов» помогает clipping
- ▶ Необходимо более глубокое понимание, как теоретическое, так и практическое — много открытых вопросов

Что ещё можно посмотреть?

- ▶ [Лекция 10](#) курса «CS231n» Andrej Karpathy в Стенфорде
- ▶ [Как тренировать нейросети?](#)
- ▶ [Аналогичное занятие](#) курса Школы анализа данных