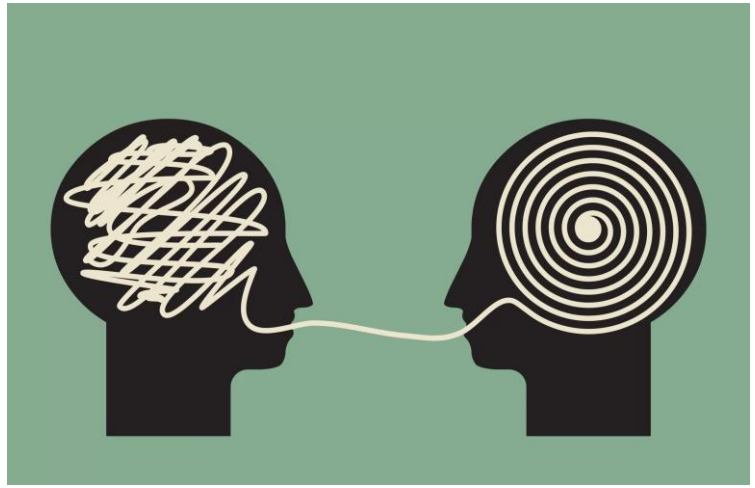
1D RNNs

Обработка естественного языка Natural Language Processing (NLP)

Обработка естественного языка (*Natural Language Processing, NLP*) — общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков.

Wikipedia



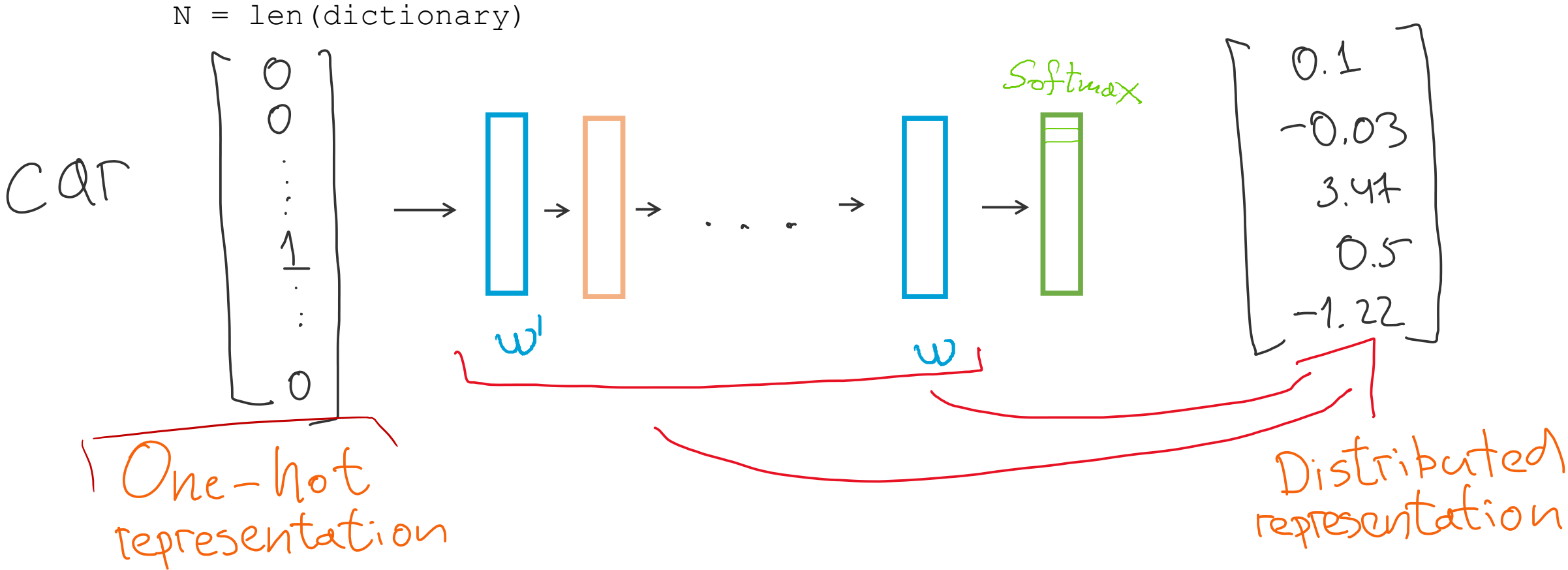
東 = est = ESTE = **east**

西 = ouest = OESTE = **west**

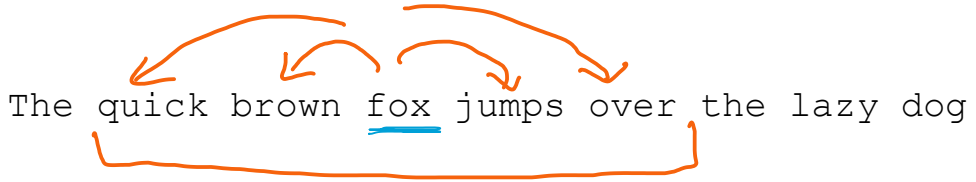
北 = nord = NORTE = **north**

南 = sud = SUR = **south**

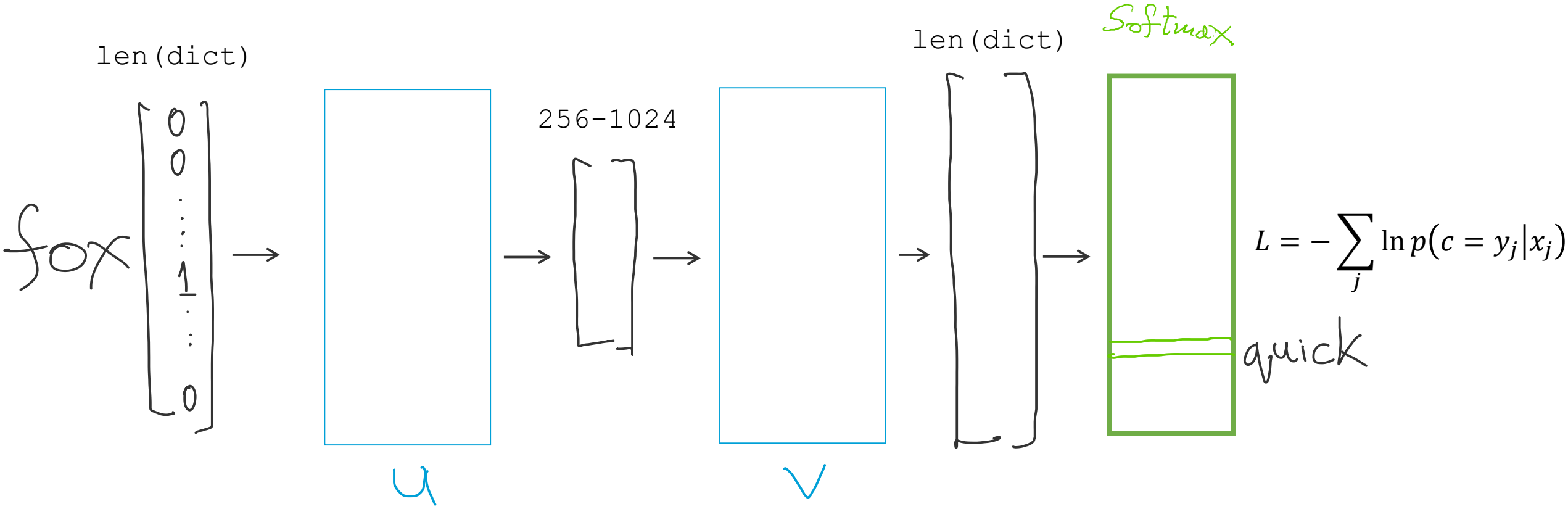
Deep NLP
Из символьного в непрерывное



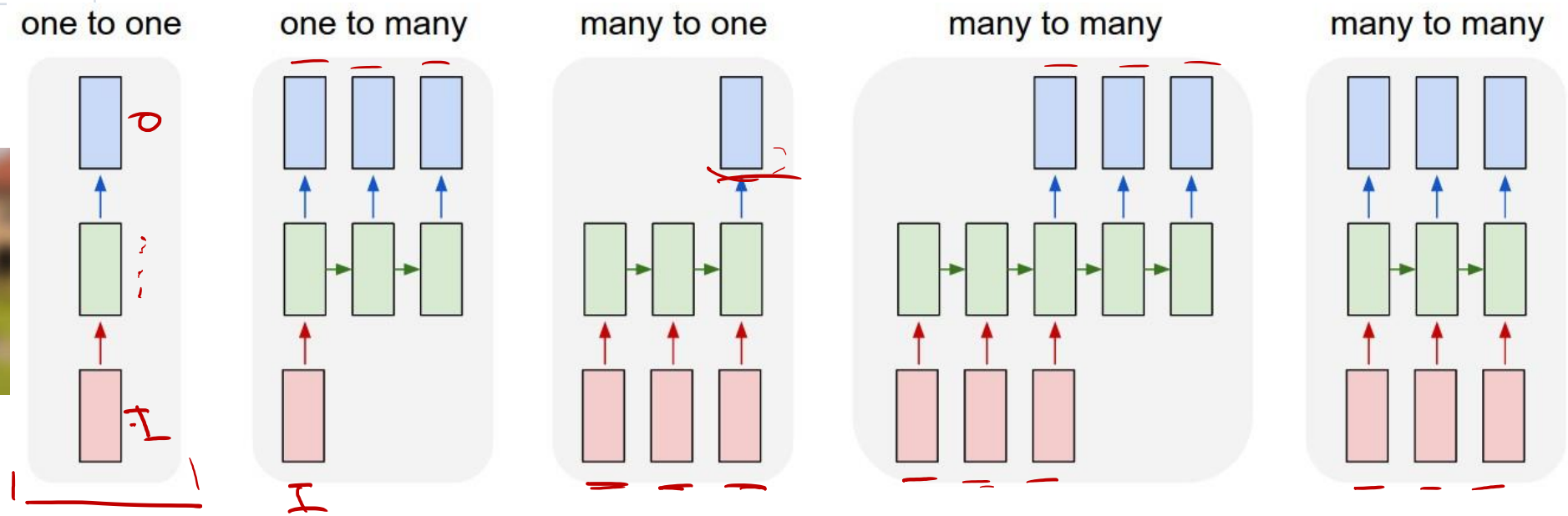
word2vec




```
fox -> quick
fox -> brown
fox -> jumps
fox -> over
```

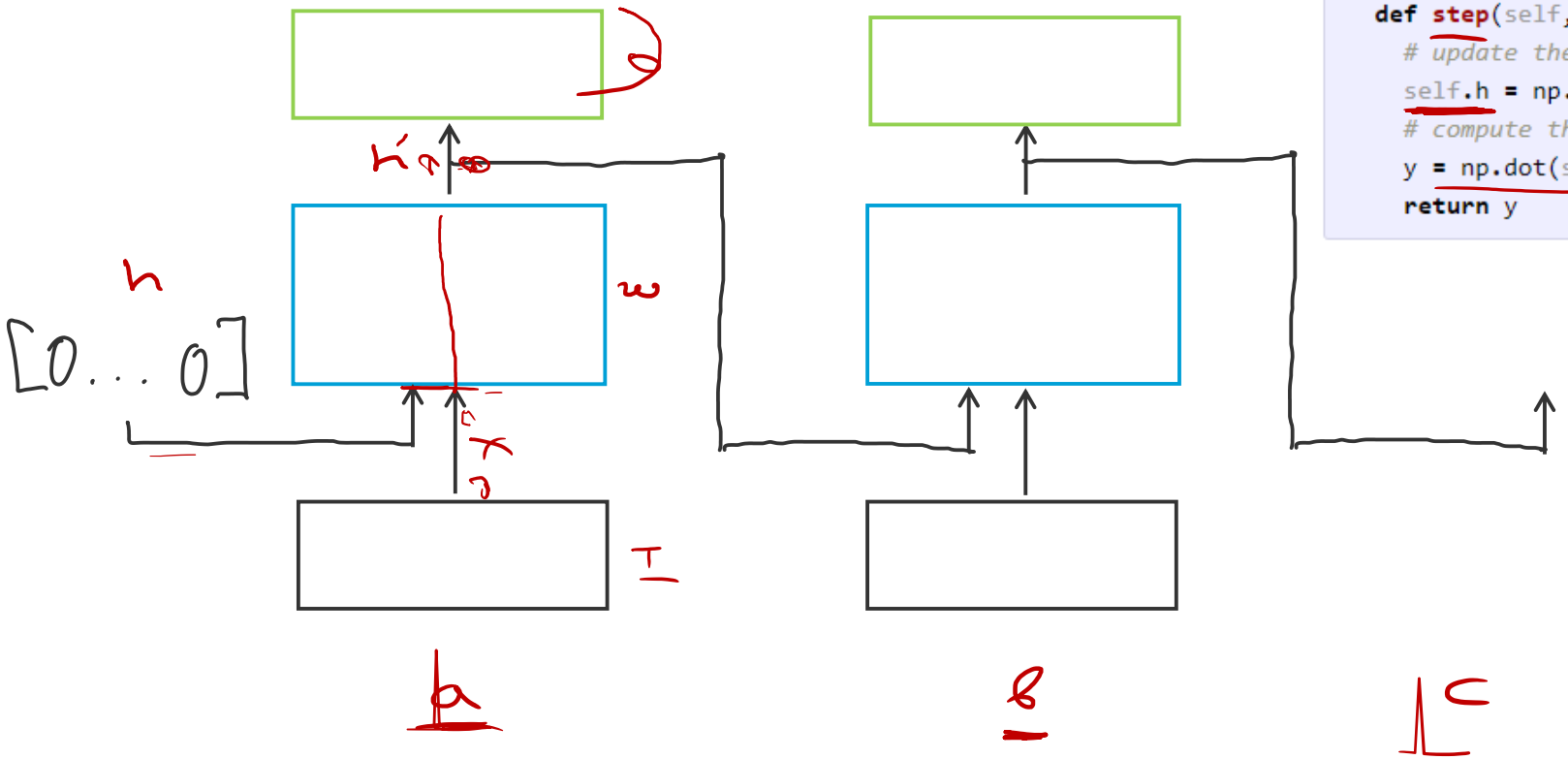


Рекуррентные нейронные сети
Recurrent Neural Networks

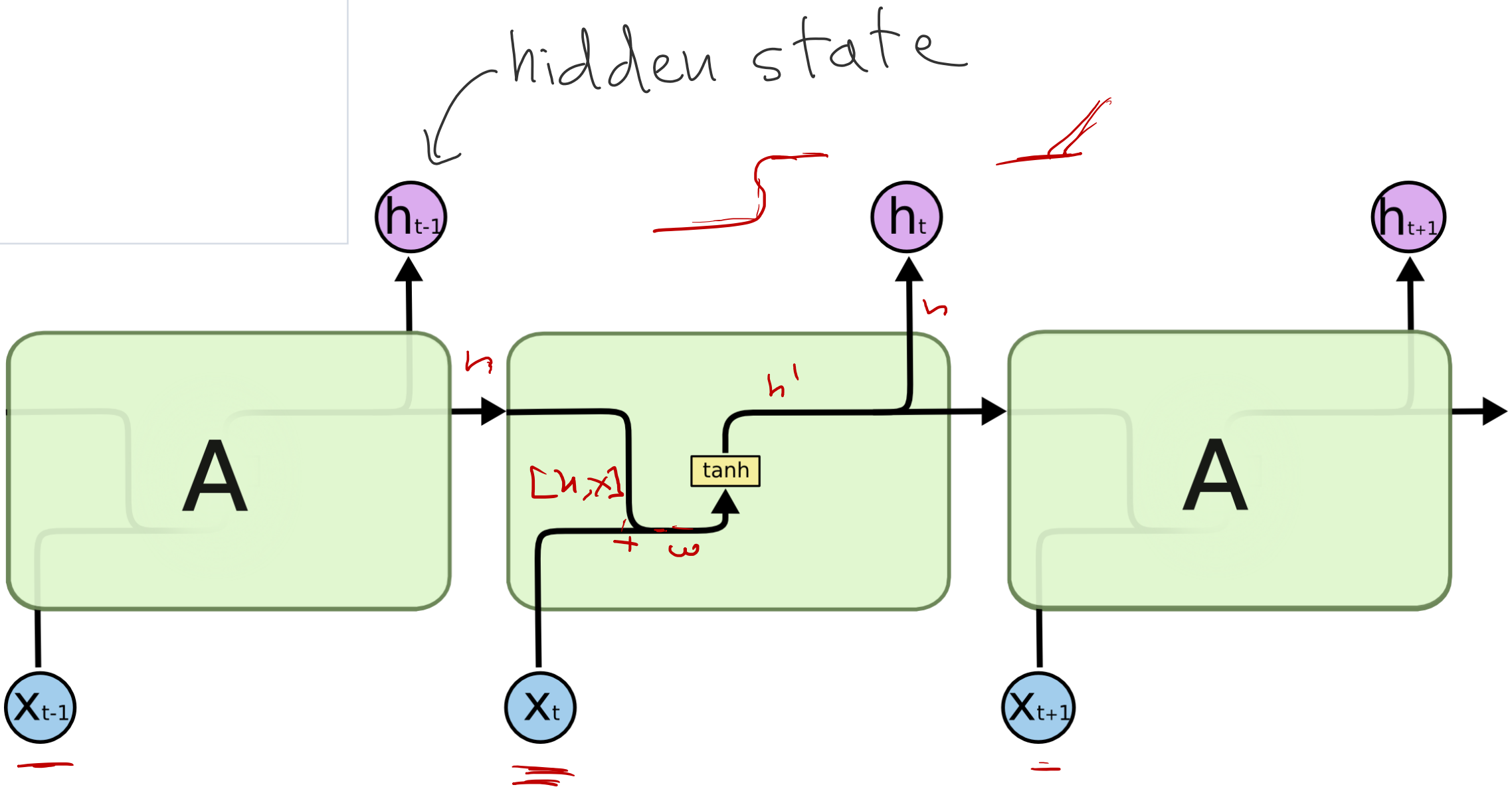


[The Unreasonable Effectiveness of Recurrent Neural Networks]

Основной шаг

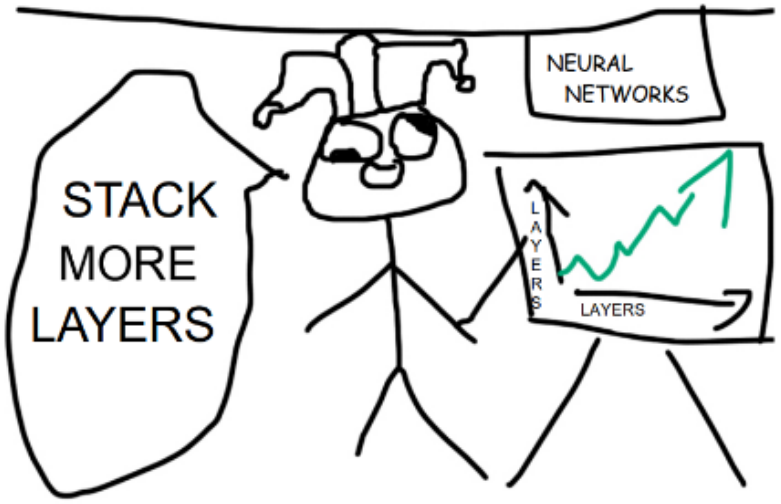


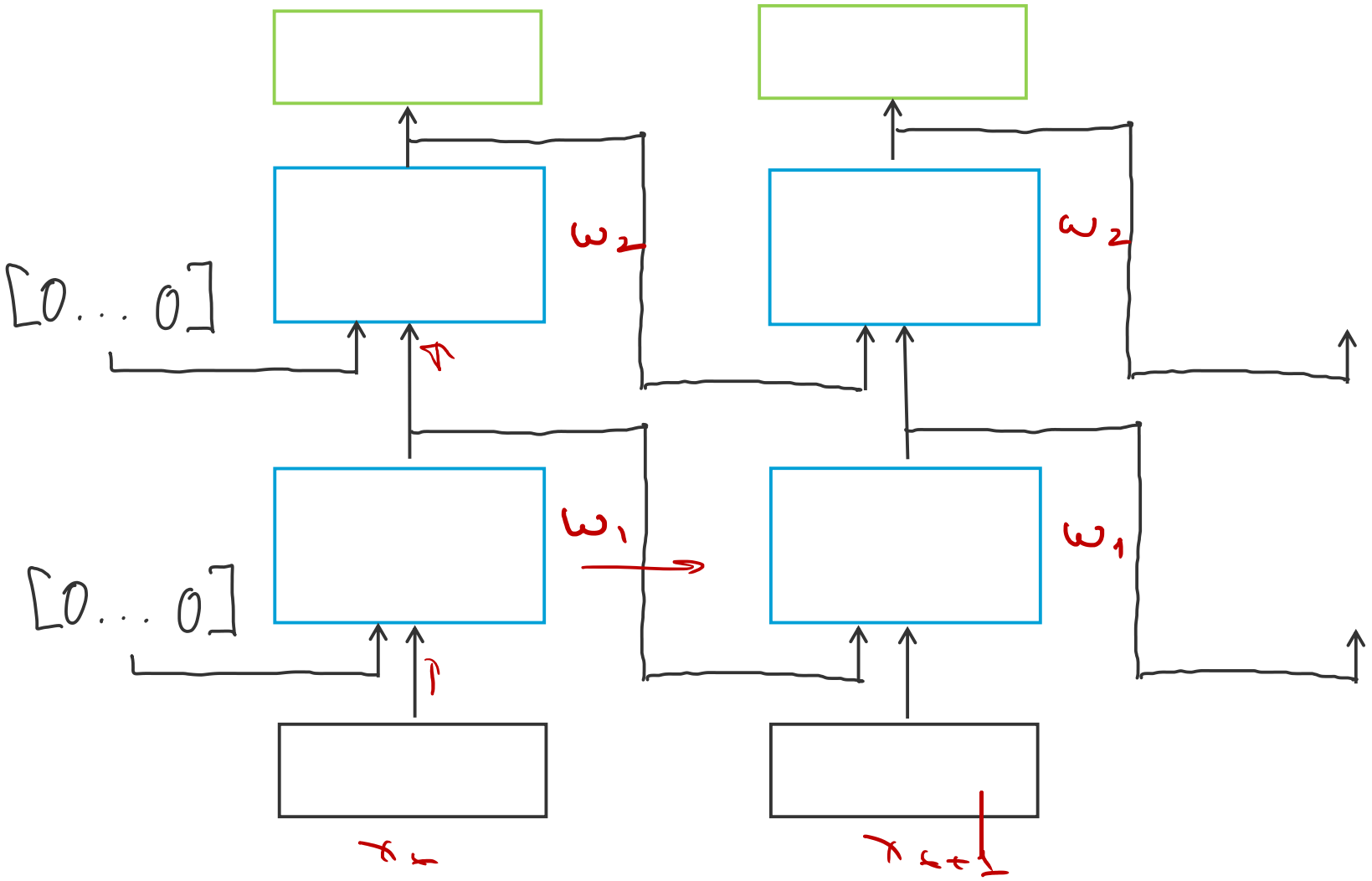
```python
class RNN:
  # ...
  def step(self, x):
    # update the hidden state
    self.h = np.tanh(np.dot(self.W_hh, self.h) + np.dot(self.W_xh, x))
    # compute the output vector
    y = np.dot(self.W_hy, self.h)
    return y
```



Understanding LSTM Networks

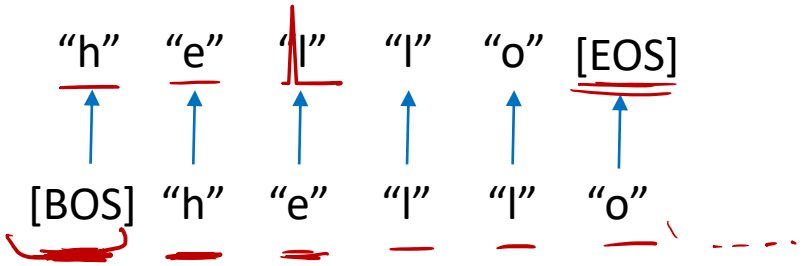
Тоже можно “stack more layers”



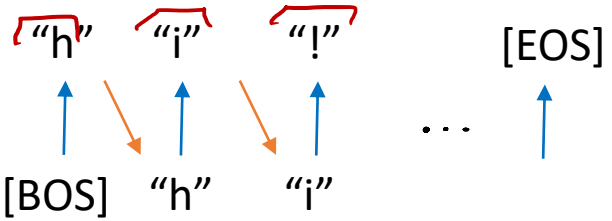


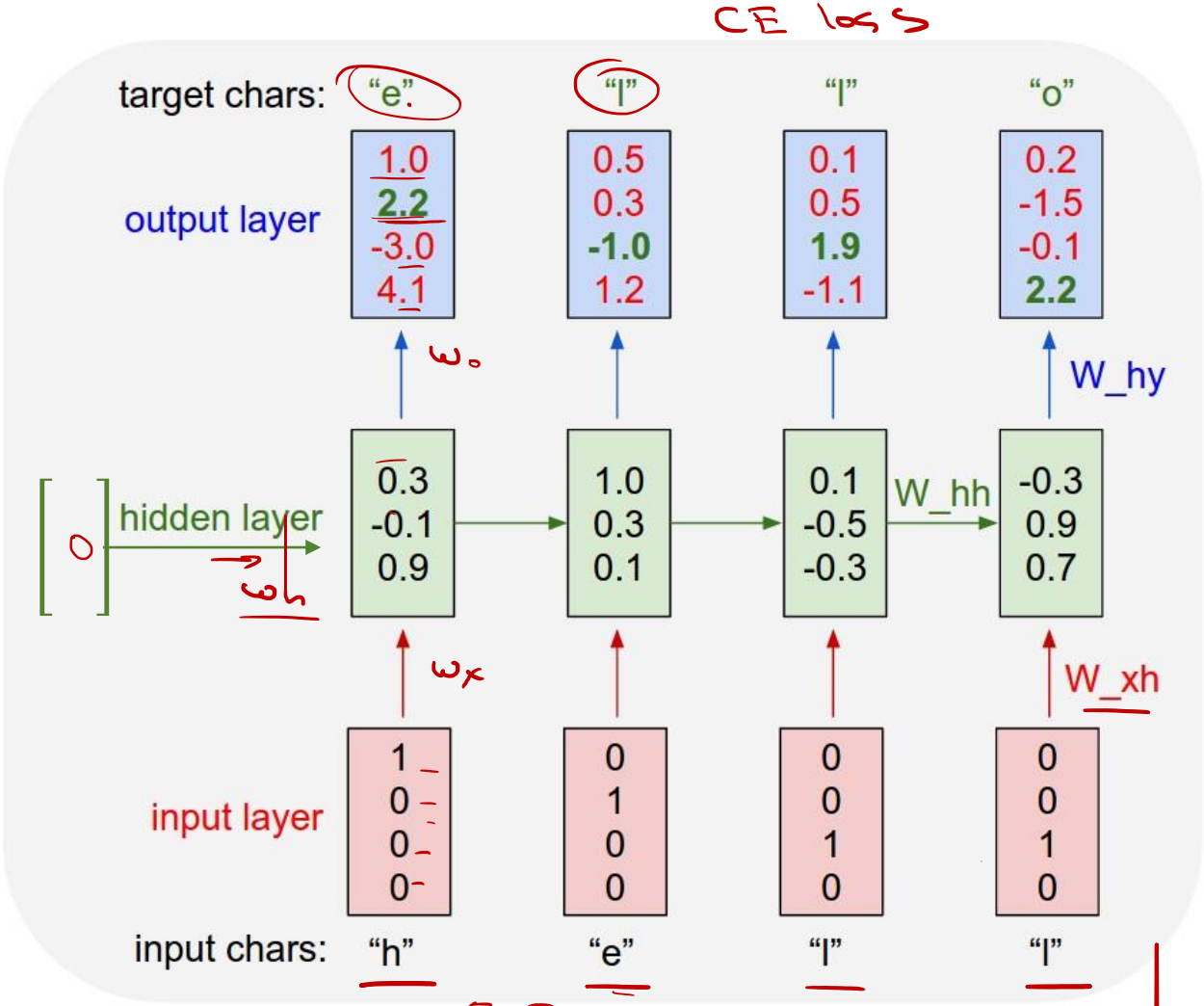
Пример: генерация текста

Training:



Inference:





[The Unreasonable Effectiveness of Recurrent Neural Networks]

Пример: генерация текста

Paul Graham

"The surprised in investors weren't going to raise money. I'm not the company with the time there are all interesting quickly, don't have to get off the same programmers. There's a super-angel round fundraising, why do you can do. If you have a different physical investment are become in people who reduced in a startup with the way to argument the acquirer could see them just that you're also the founders will part of users' affords that and an alternation to the idea. [2] Don't work at first member to see the way kids will seem in advance of a bad successful startup. And if you have to act the big company too."

```
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.
```

[The Unreasonable Effectiveness of Recurrent Neural Networks]

Iteration: 100

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

Iteration: 300

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

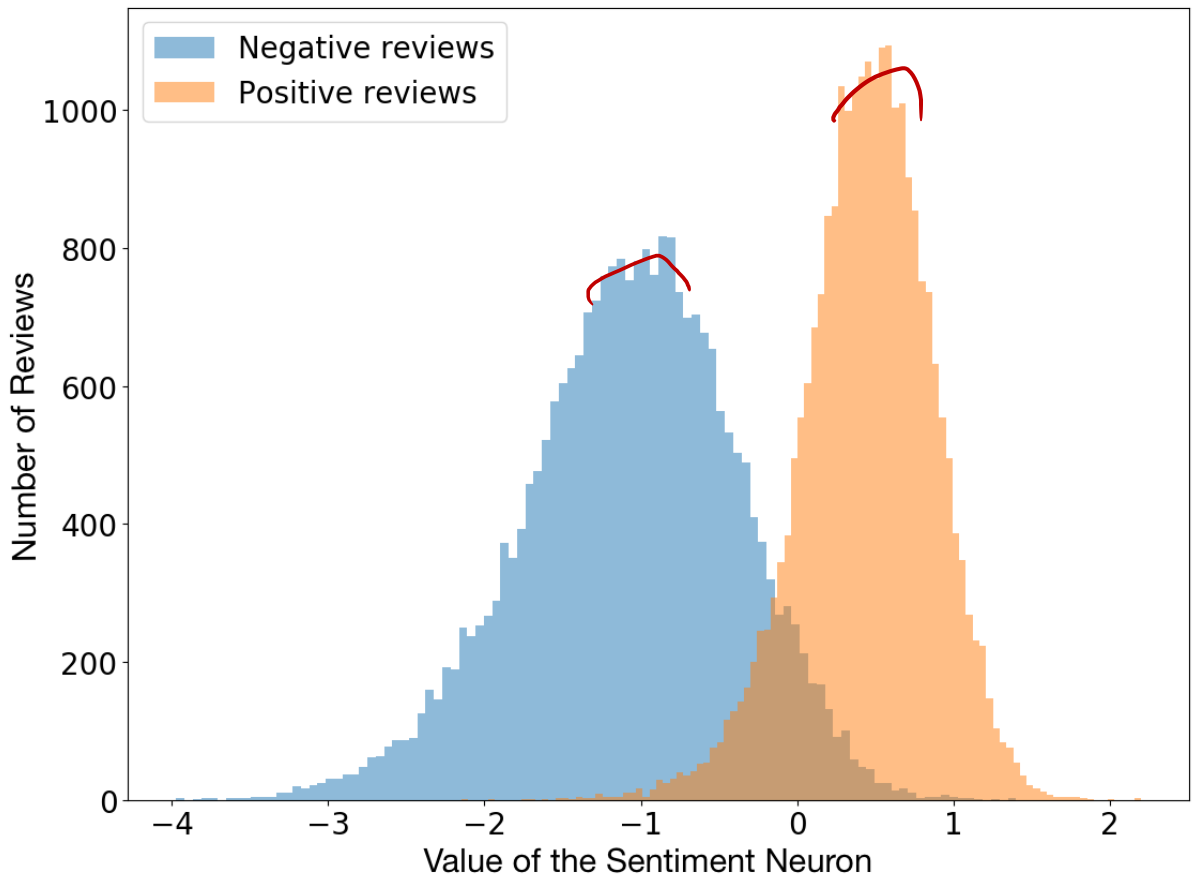
Iteration: 700

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

Iteration: 2000

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

[The Unreasonable Effectiveness of Recurrent Neural Networks]

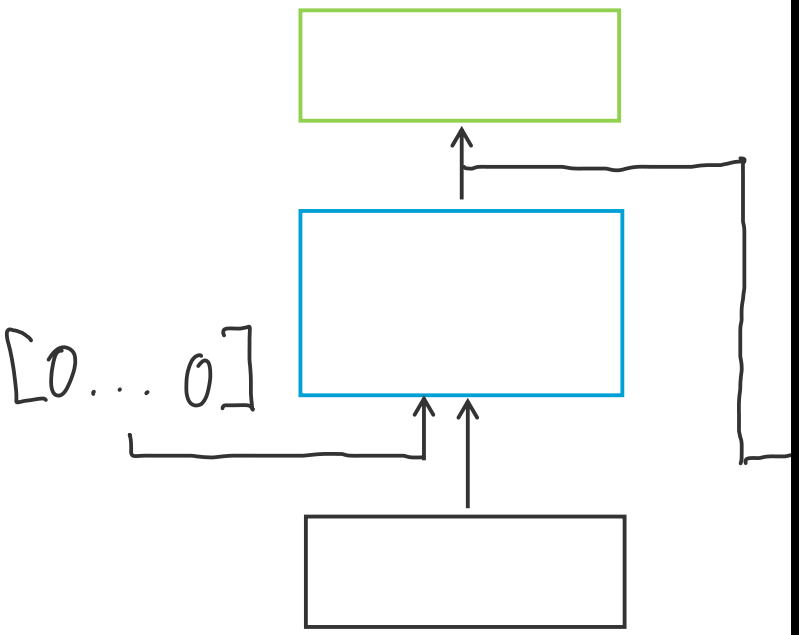


[Unsupervised Sentiment Neuron]

SENTIMENT FIXED TO POSITIVE	SENTIMENT FIXED TO NEGATIVE
I couldn't figure out the shape at first but it definitely does what it's meant to do. It's a great product and I recommend it highly	I couldn't figure out how to use the product. It did not work.At least there was no quality control; this tablet does not work. I would have given it zero stars, but that was not an option.
I couldn't figure out why this movie had been discontinued! Now I can enjoy it anytime I like. So glad to have found it again.	I couldn't figure out how to set it up being that there was no warning on the box. I wouldn't recommend this to anyone.
I couldn't figure out how to use the video or the book that goes along with it, but it is such a fantastic book on how to put it into practice!	I couldn't figure out how to use the gizmo. What a waste of time and money. Might as well through away this junk.
I couldn't figure out how to use just one and my favorite running app. I use it all the time. Good quality, You cant beat the price.	I couldn't figure out how to stop this drivel. At worst, it was going absolutely nowhere, no matter what I did.Needles to say, I skim-read the entire book. Don't waste your time.
I couldn't figure out how to attach these balls to my little portable drums, but these fit the bill and were well worth every penny.	I couldn't figure out how to play it.

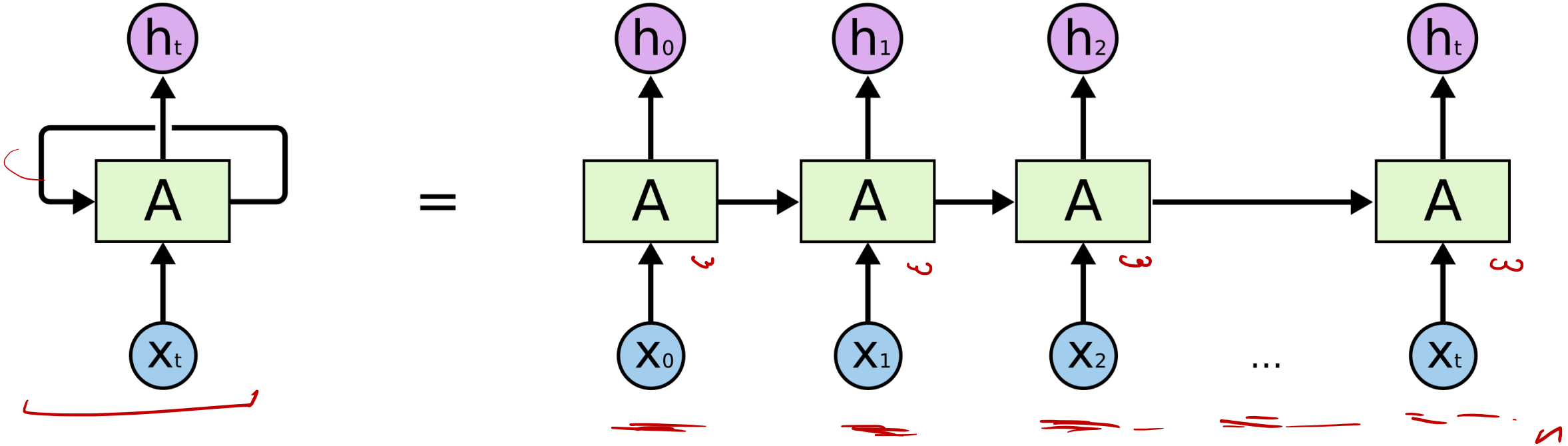
Небольшой секрет...

Основной шаг




```
dot(self.W_xh, x))
```

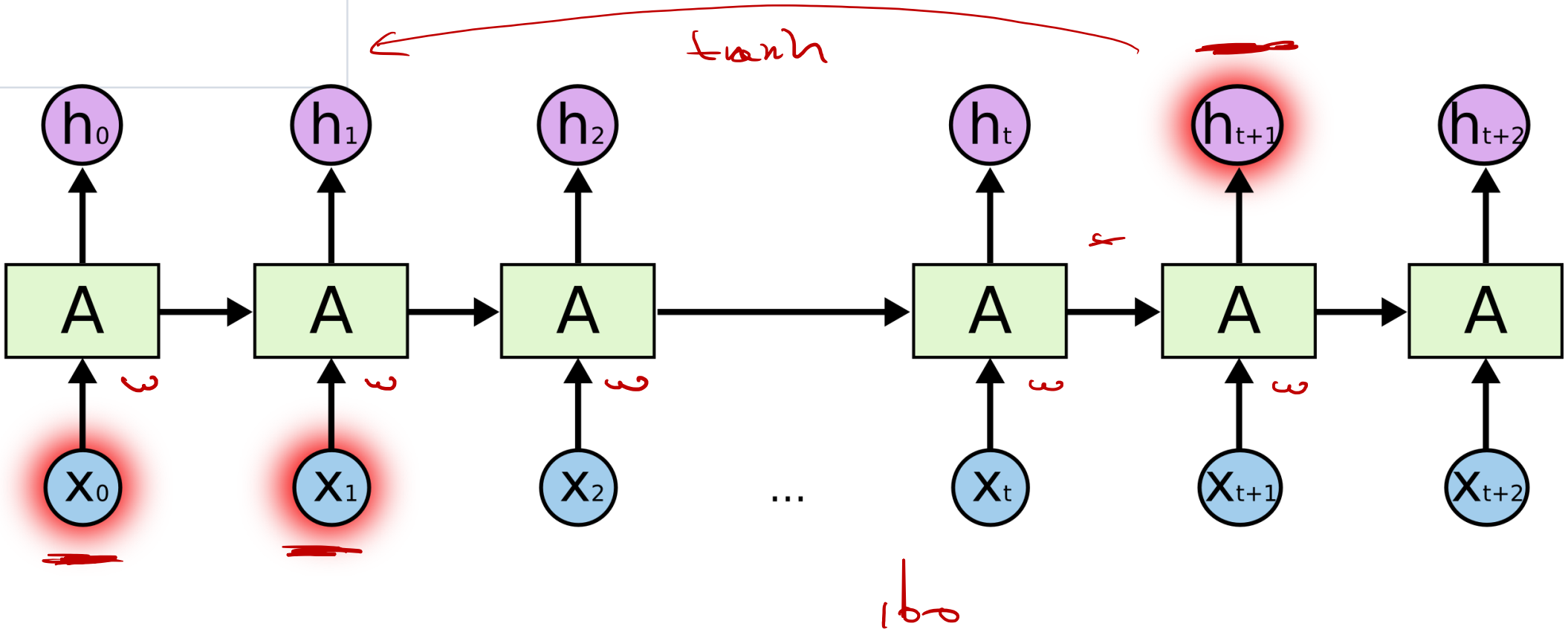
Как происходит тренировка



Backpropagation through time

[Understanding LSTM Networks]

Проблема длинных зависимостей



[Understanding LSTM Networks]

LONG SHORT-TERM MEMORY

LSTM

Neural Computation 9(8):1735–1780, 1997

Sepp Hochreiter
Fakultät für Informatik
Technische Universität München
80290 München, Germany
hochreit@informatik.tu-muenchen.de
http://www7.informatik.tu-muenchen.de/~hochreit

Jürgen Schmidhuber
IDSIA
Corso Elvezia 36
6900 Lugano, Switzerland
juergen@idsia.ch
http://www.idsia.ch/~juergen

Abstract

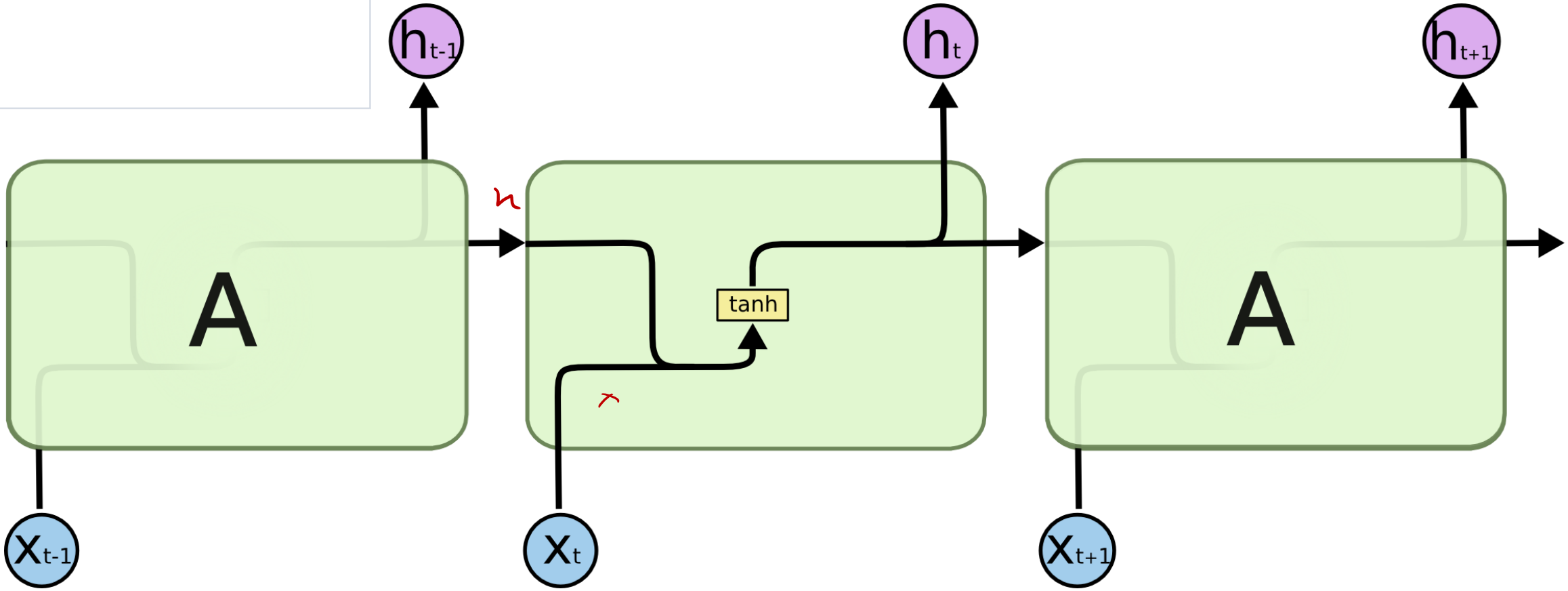
Learning to store information over extended time intervals via recurrent backpropagation takes a very long time, mostly due to insufficient, decaying error back flow. We briefly review Hochreiter's 1991 analysis of this problem, then address it by introducing a novel, efficient, gradient-based method called "Long Short-Term Memory" (LSTM). Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete time steps by enforcing *constant* error flow through "constant error carrousels" within special units. Multiplicative gate units learn to open and close access to the constant error flow. LSTM is local in space and time; its computational complexity per time step and weight is $O(1)$. Our experiments with artificial data involve local, distributed, real-valued, and noisy pattern representations. In comparisons with RTRL, BPTT, Recurrent Cascade-Correlation, Elman nets, and Neural Sequence Chunking, LSTM leads to many more successful runs, and learns much faster. LSTM also solves complex, artificial long time lag tasks that have never been solved by previous recurrent network algorithms.

1 INTRODUCTION

Recurrent networks can in principle use their feedback connections to store representations of recent input events in form of activations ("short-term memory", as opposed to "long-term memory" embodied by slowly changing weights). This is potentially significant for many applications, including speech processing, non-Markovian control, and music composition (e.g., Mozer 1992). The most widely used algorithms for learning *what* to put in short-term memory, however, take too much time or do not work well at all, especially when minimal time lags between inputs and corresponding teacher signals are long. Although theoretically fascinating, existing methods do not provide clear *practical* advantages over, say, backprop in feedforward nets with limited time windows. This paper will review an analysis of the problem and suggest a remedy.

Long Short-term Memory'97

Vanilla RNN



[Understanding LSTM Networks]

Long Short-term Memory (LSTM)

Forget gate:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Input gate:

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

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

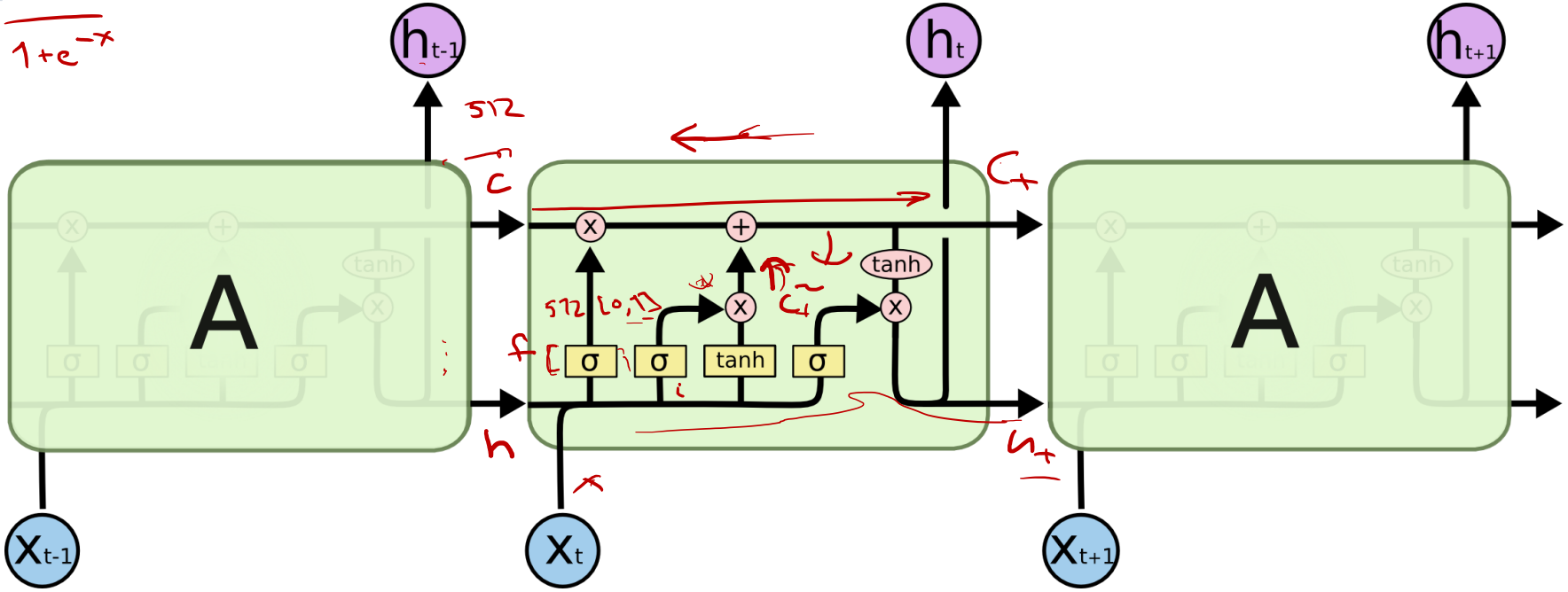
Cell update:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Output gate:

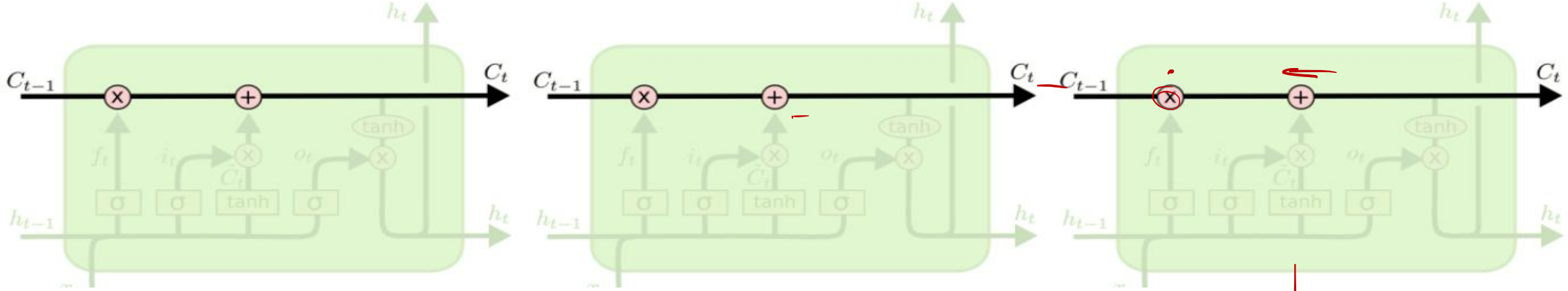
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

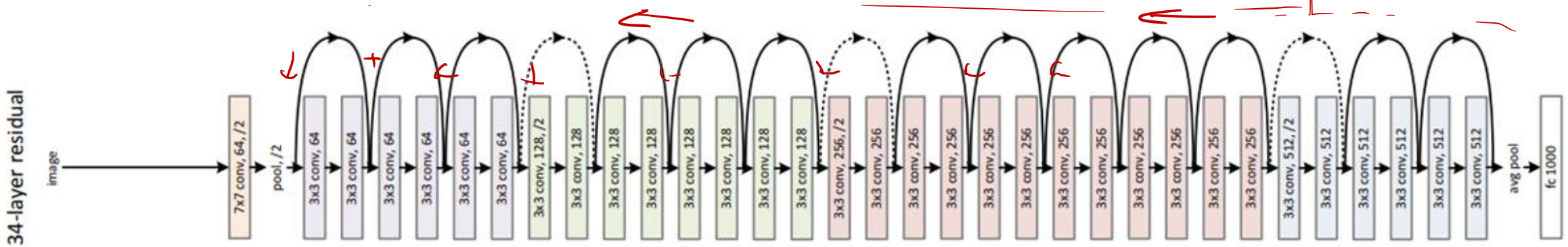
$$h_t = o_t * \tanh\left(C_t\right)$$



[Understanding LSTM Networks]

Почему это работает?

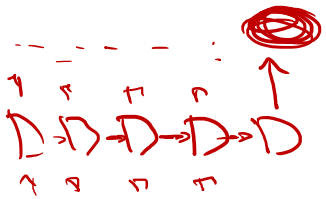




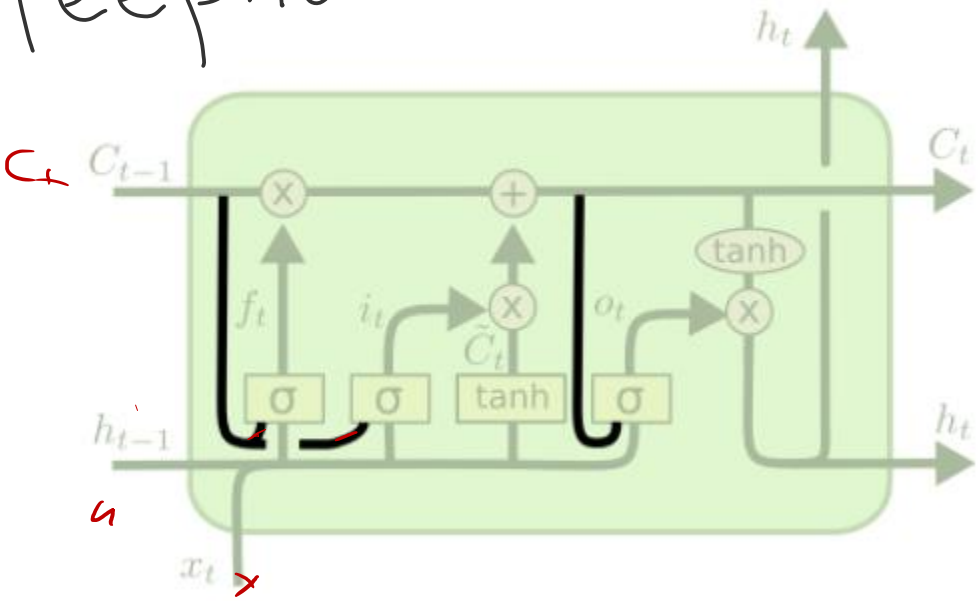
[Understanding LSTM Networks]

[ResNet'15]

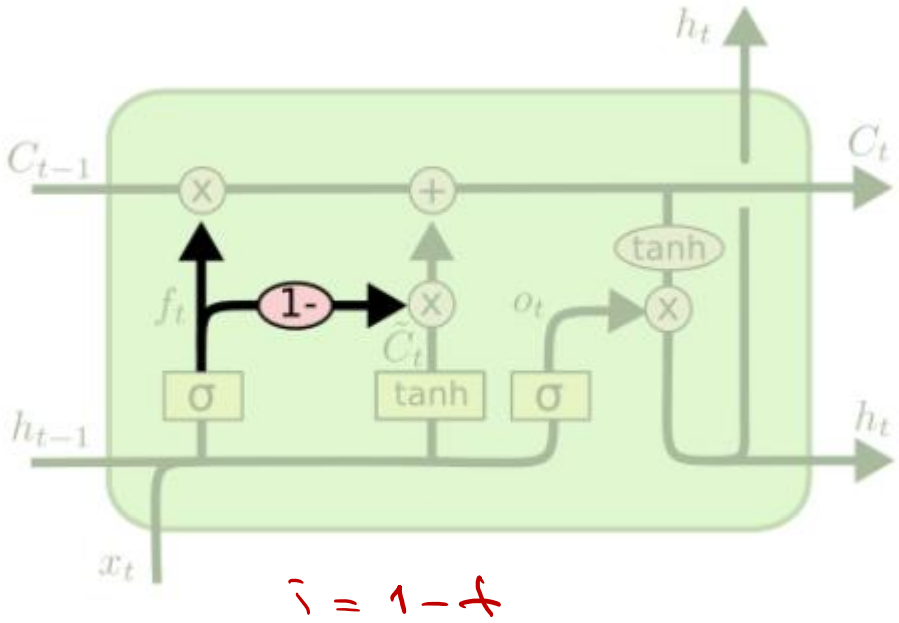
Варианты LSTM



Peephole connections



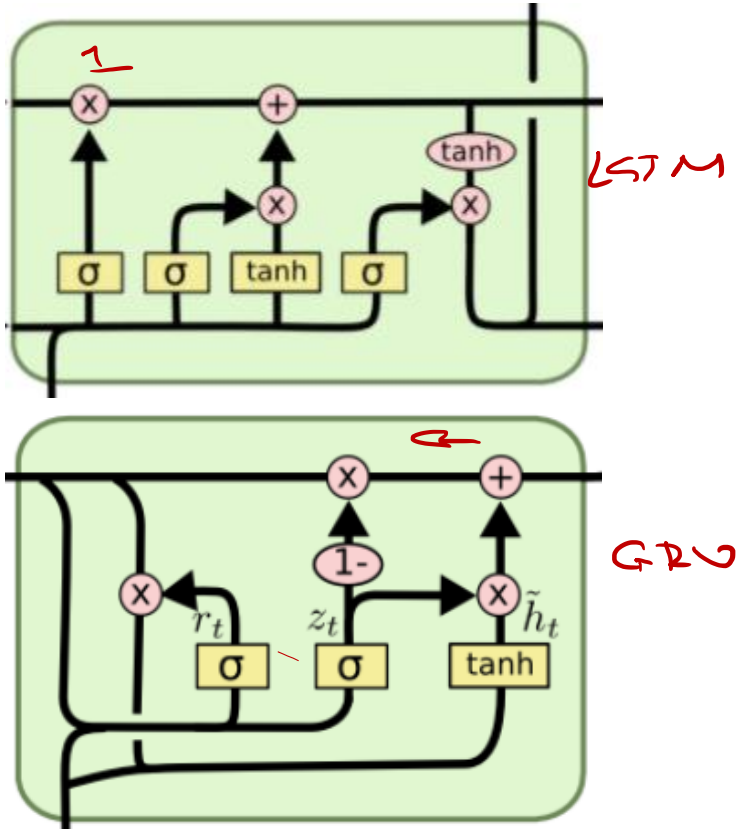
No input gate



$i = 1 - f$

[Understanding LSTM Networks]

А оно надо?



LSTM

GRU

1. If the current node is an activation function, replace it with a randomly chosen activation function. The set of permissible nonlinearities is $\{\tanh(x)$, sigmoid(x), ReLU(x), Linear$(0, x)$, Linear$(1, x)$, Linear$(0.9, x)$, Linear$(1.1, x)\}$. Here the operator Linear(a, x) introduced a new square parameter matrix W and a bias b which replaces x by $W \cdot x + b$. The new matrix is initialized to small random values whose magnitude is determined by the scale parameter, and the scalar value a is added to its diagonal entries.
2. If the current node is an elementwise operation, replace it with a different randomly-chosen elementwise operation (multiplication, addition, subtraction) .
3. Insert a random activation function between the current node and one of its parents.
4. Remove the current node if it has one input and one output.
5. Replace the current node with a randomly chosen node from the current node's ancestors (i.e., all nodes that A depends on). This allows us to reduce the size of the graph.
6. Randomly select a node (node A). Replace the current node with either the sum, product, or difference of a random ancestor of the current node and a random ancestor of A. This guarantees that the resulting computation graph will remain well-defined.

MUT1:

$$\begin{aligned} z &= \text{sigm}(W_{xz}x_t + b_z) \\ r &= \text{sigm}(W_{xr}x_t + W_{hr}h_t + b_r) \\ h_{t+1} &= \tanh(W_{hh}(r \odot h_t) + \tanh(x_t) + b_h) \odot z \\ &+ h_t \odot (1 - z) \end{aligned}$$

MUT2:

$$\begin{aligned} z &= \text{sigm}(W_{xz}x_t + W_{hz}h_t + b_z) \\ r &= \text{sigm}(x_t + W_{hr}h_t + b_r) \\ h_{t+1} &= \tanh(W_{hh}(r \odot h_t) + W_{xh}x_t + b_h) \odot z \\ &+ h_t \odot (1 - z) \end{aligned}$$

MUT3:

$$\begin{aligned} z &= \text{sigm}(W_{xz}x_t + W_{hz}\tanh(h_t) + b_z) \\ r &= \text{sigm}(W_{xr}x_t + W_{hr}h_t + b_r) \\ h_{t+1} &= \tanh(W_{hh}(r \odot h_t) + W_{xh}x_t + b_h) \odot z \\ &+ h_t \odot (1 - z) \end{aligned}$$

Arch.	5M-tst	10M-v	20M-v	20M-tst
Tanh	4.811	4.729	4.635	4.582 (97.7)
LSTM	4.699	4.511	4.437	4.399 (81.4)
LSTM-f	4.785	4.752	4.658	4.606 (100.8)
LSTM-i	4.755	4.558	4.480	4.444 (85.1)
LSTM-o	4.708	4.496	4.447	4.411 (82.3)
LSTM-b	4.698	4.437	4.423	**4.380 (79.83)**
GRU	4.684	4.554	4.559	4.519 (91.7)
MUT1	4.699	4.605	4.594	4.550 (94.6)
MUT2	4.707	4.539	4.538	4.503 (90.2)
MUT3	4.692	4.523	4.530	4.494 (89.47)

[An Empirical Exploration of Recurrent Network Architectures'15]

Bidirectional RNN

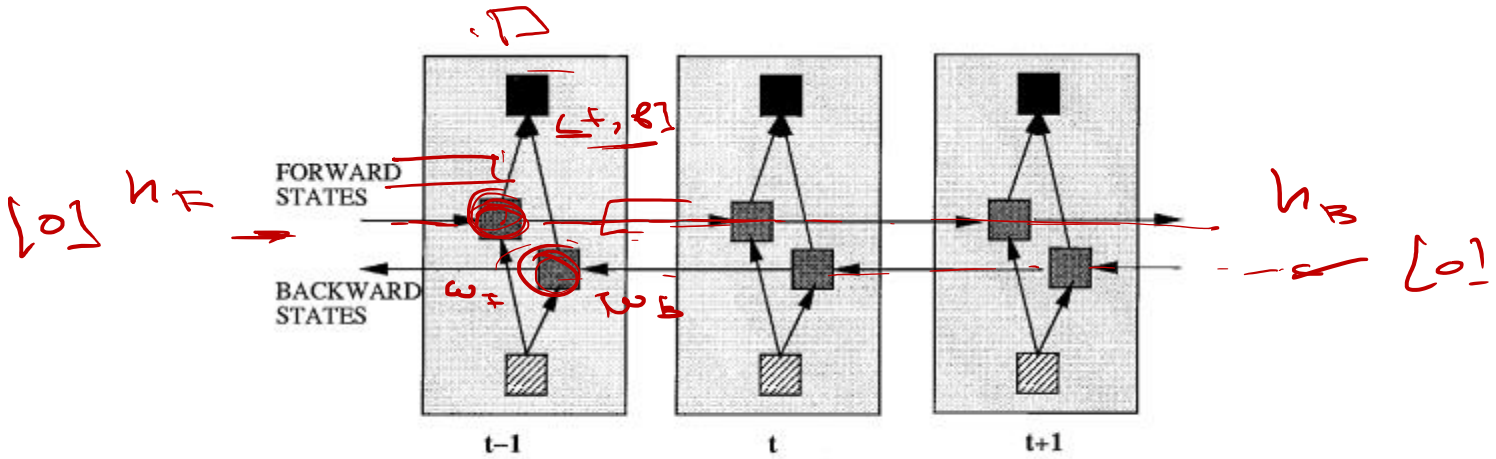
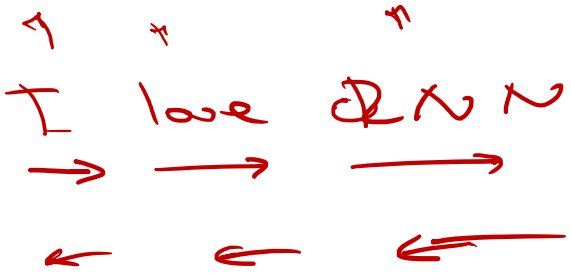


Fig. 3. General structure of the bidirectional recurrent neural network (BRNN) shown unfolded in time for three time steps.



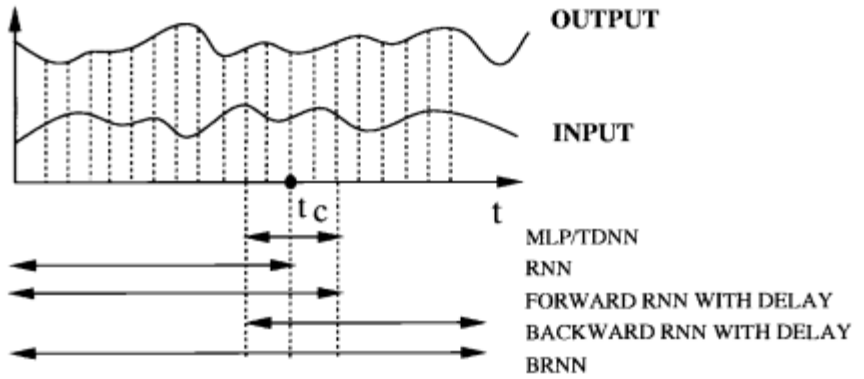
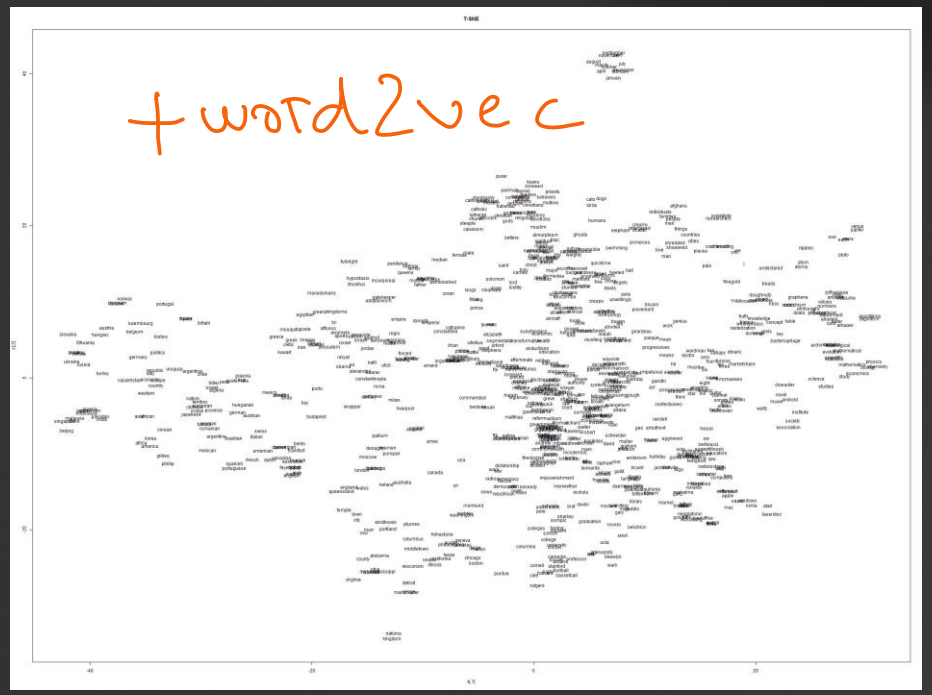


Fig. 2. Visualization of the amount of input information used for prediction by different network structures.

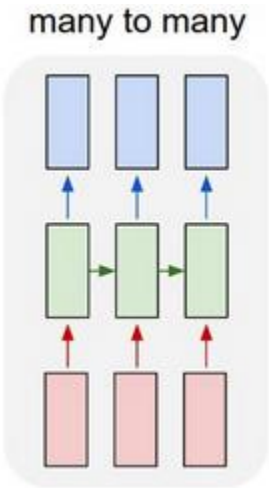
[Bidirectional Recurrent Neural Networks'97]



Part of Speech Tagging
Разбор частей речи

```
the big green fire truck
```

Wikipedia



2015-2016

Bidirectional LSTM-CRF Models for Sequence Tagging

Zhiheng Huang
Baidu research
huangzhiheng@baidu.com

Wei Xu
Baidu research
xuwei06@baidu.com

Kai Yu
Baidu research
yukai@baidu.com

arxiv

Named Entity Recognition

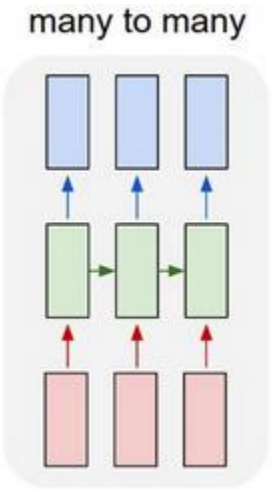
Most research on NER systems has been structured as taking an unannotated block of text, such as this one:

Jim bought 300 shares of Acme Corp. in 2006.

And producing an annotated block of text that highlights the names of entities:

[Jim]$_{\text{Person}}$ bought 300 shares of [Acme Corp.]$_{\text{Organization}}$ in [2006]$_{\text{Time}}$.

Wikipedia



Named Entity Recognition with Bidirectional LSTM-CNNs

Jason P.C. Chiu
University of British Columbia
jsonchiu@gmail.com

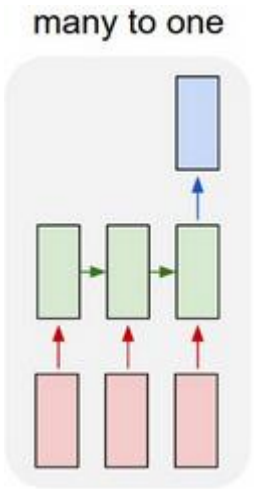
Eric Nichols
Honda Research Institute Japan Co.,Ltd.
e.nichols@jp.honda-ri.com

arxiv

Sentiment Analysis

The film is powerful, accessible and funny
The movie is well done, but slow
Ah yes, and then there's the music

Stanford Sentiment Treebank

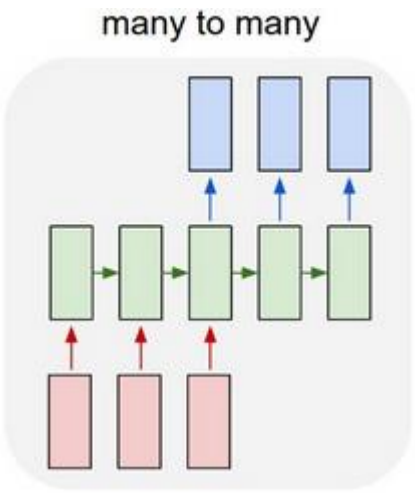


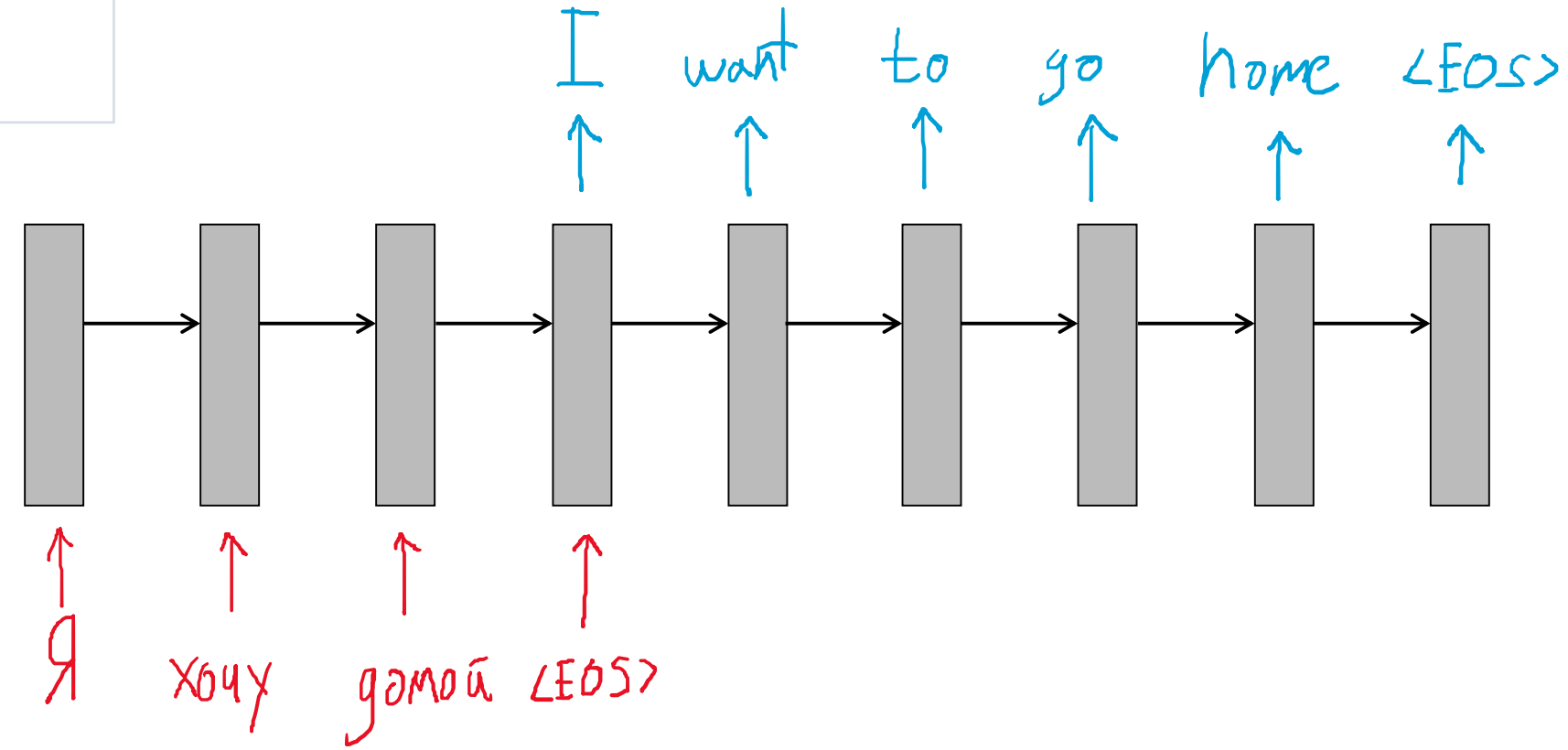
Assessing State-of-the-Art Sentiment Models on State-of-the-Art Sentiment Datasets

Jeremy Barnes, Roman Klinger, and **Sabine Schulte im Walde**
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart, Germany
{barnesjy,klinger,schulte}@ims.uni-stuttgart.de

Source

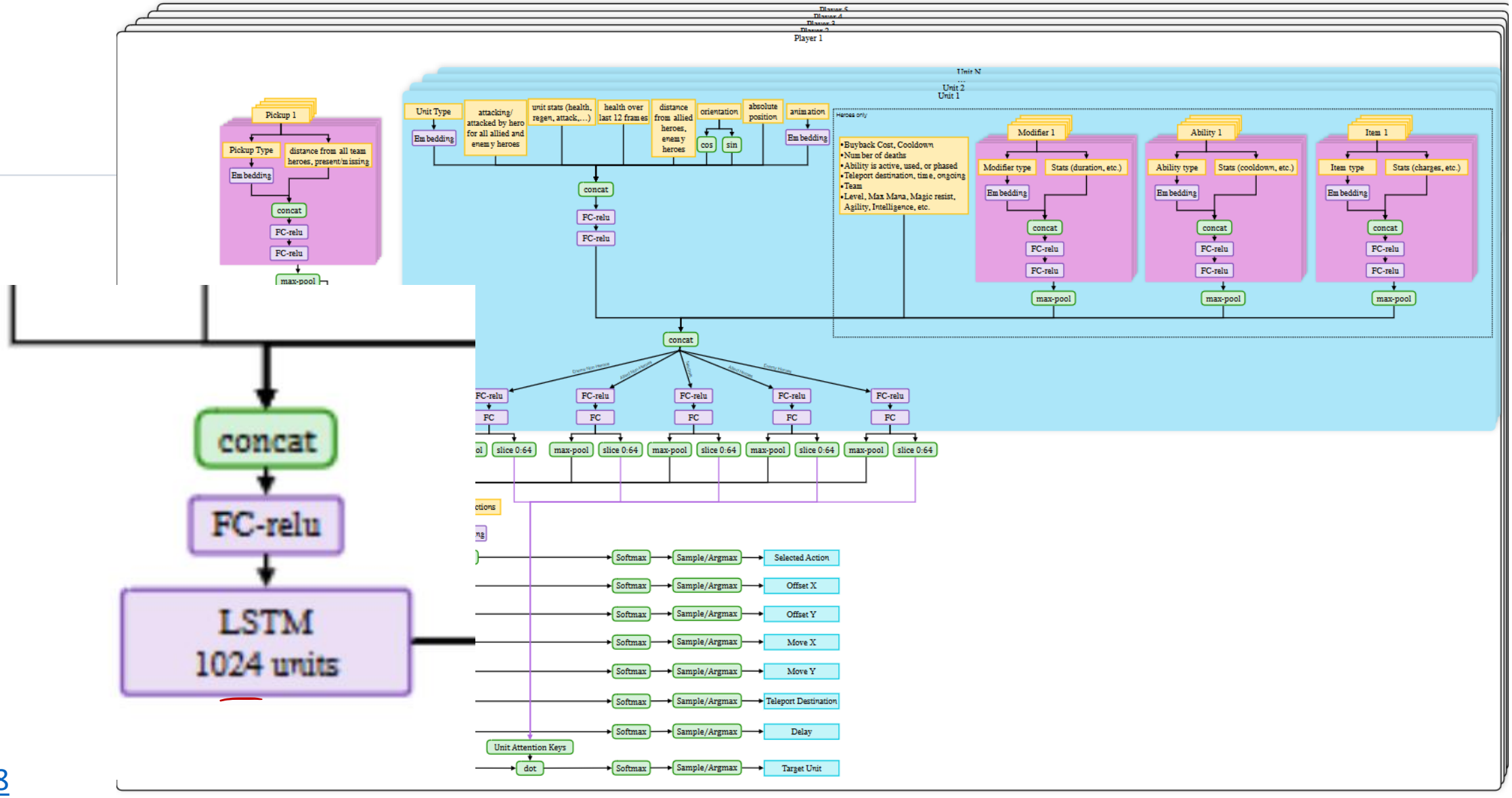
Машинный перевод
Machine Translation





Sequence to Sequence Learning with Neural Networks'14

И даже OpenAI Five для Dota



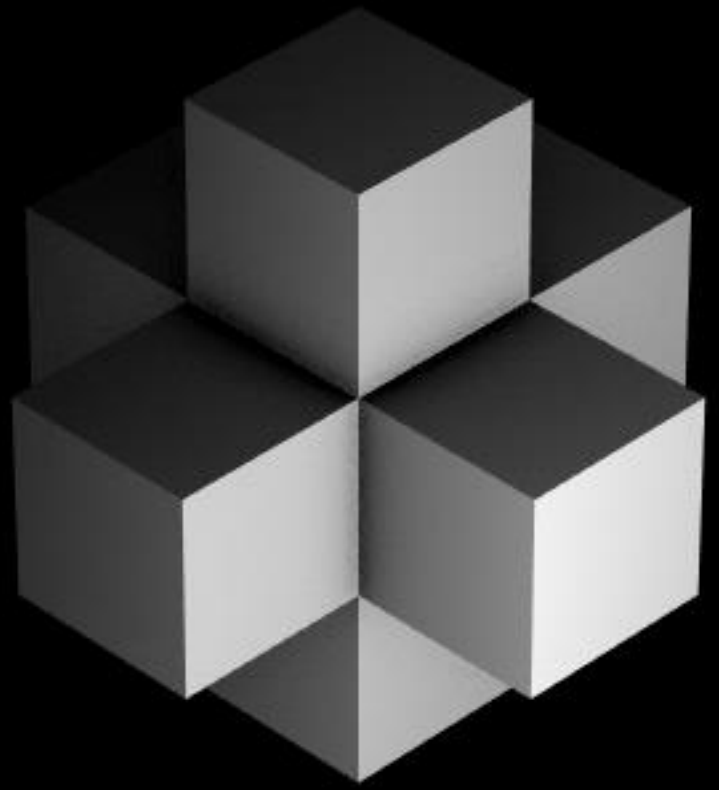
[OpenAI blog'18]

Есть в каждом фреймворке

PYTORCH

```
class LSTMTagger(nn.Module):

    def __init__(self, embedding_dim, hidden_dim, vocab_size, tagset_size):
        super(LSTMTagger, self).__init__()
        self.hidden_dim = hidden_dim

        self.word_embeddings = nn.Embedding(vocab_size, embedding_dim)

        # The LSTM takes word embeddings as inputs, and outputs hidden states
        # with dimensionality hidden_dim.
        self.lstm = nn.LSTM(embedding_dim, hidden_dim)

        # The linear layer that maps from hidden state space to tag space
        self.hidden2tag = nn.Linear(hidden_dim, tagset_size)
        self.hidden = self.init_hidden()

    def init_hidden(self):
        # Before we've done anything, we dont have any hidden state.
        # Refer to the Pytorch documentation to see exactly
        # why they have this dimensionality.
        # The axes semantics are (num_layers, minibatch_size, hidden_dim)
        return (autograd.Variable(torch.zeros(1, 1, self.hidden_dim)),
                autograd.Variable(torch.zeros(1, 1, self.hidden_dim)))

    def forward(self, sentence):
        embeds = self.word_embeddings(sentence)
        lstm_out, self.hidden = self.lstm(
            embeds.view(len(sentence), 1, -1), self.hidden)
        tag_space = self.hidden2tag(lstm_out.view(len(sentence), -1))
        tag_scores = F.log_softmax(tag_space, dim=1)
        return tag_scores
```

Sequence Models and Long-Short Term Memory Networks, Pytorch Tutorial



Например, Part of Speech Tagging

A	DET
revolving	VERB
fund	NOUN

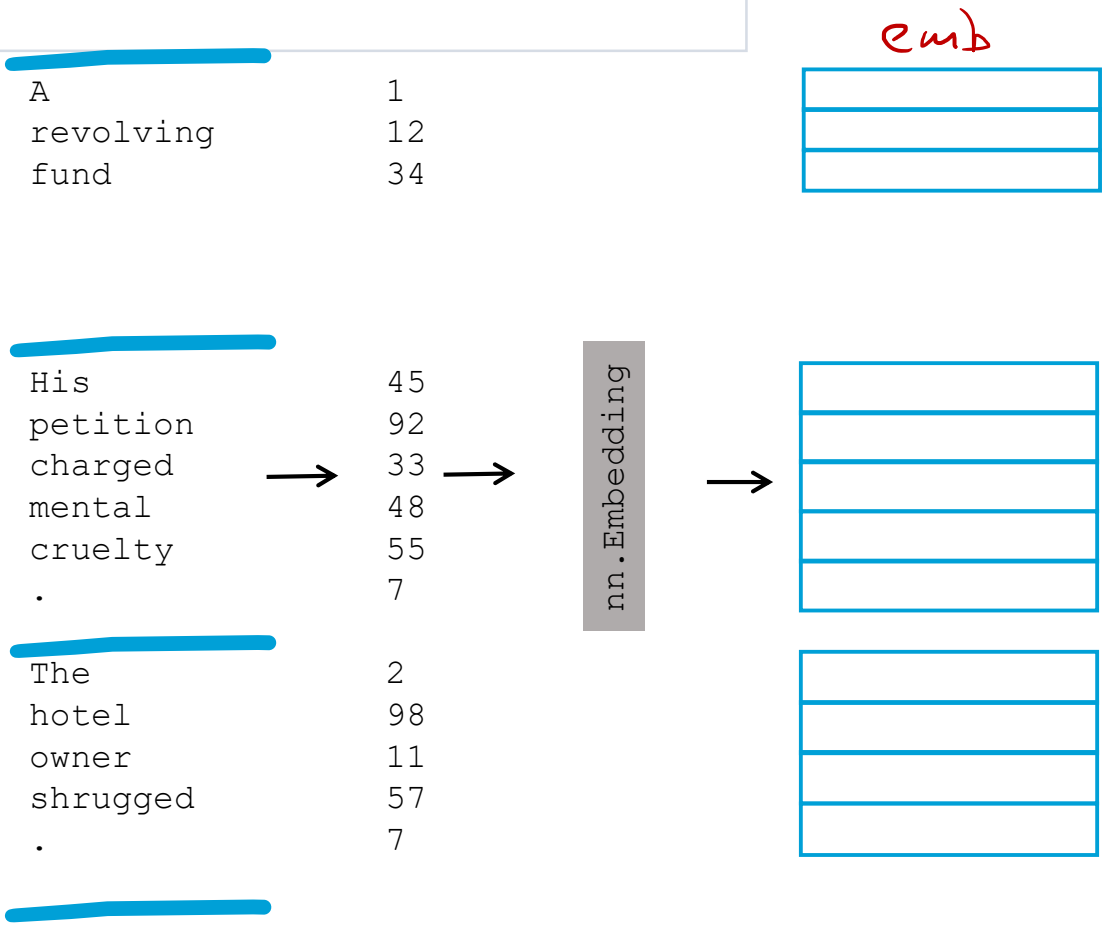
His	DET
petition	NOUN
charged	VERB
mental	ADJ
cruelty	NOUN
.	.

The	DET
hotel	NOUN
owner	NOUN
shrugged	VERB
.	.

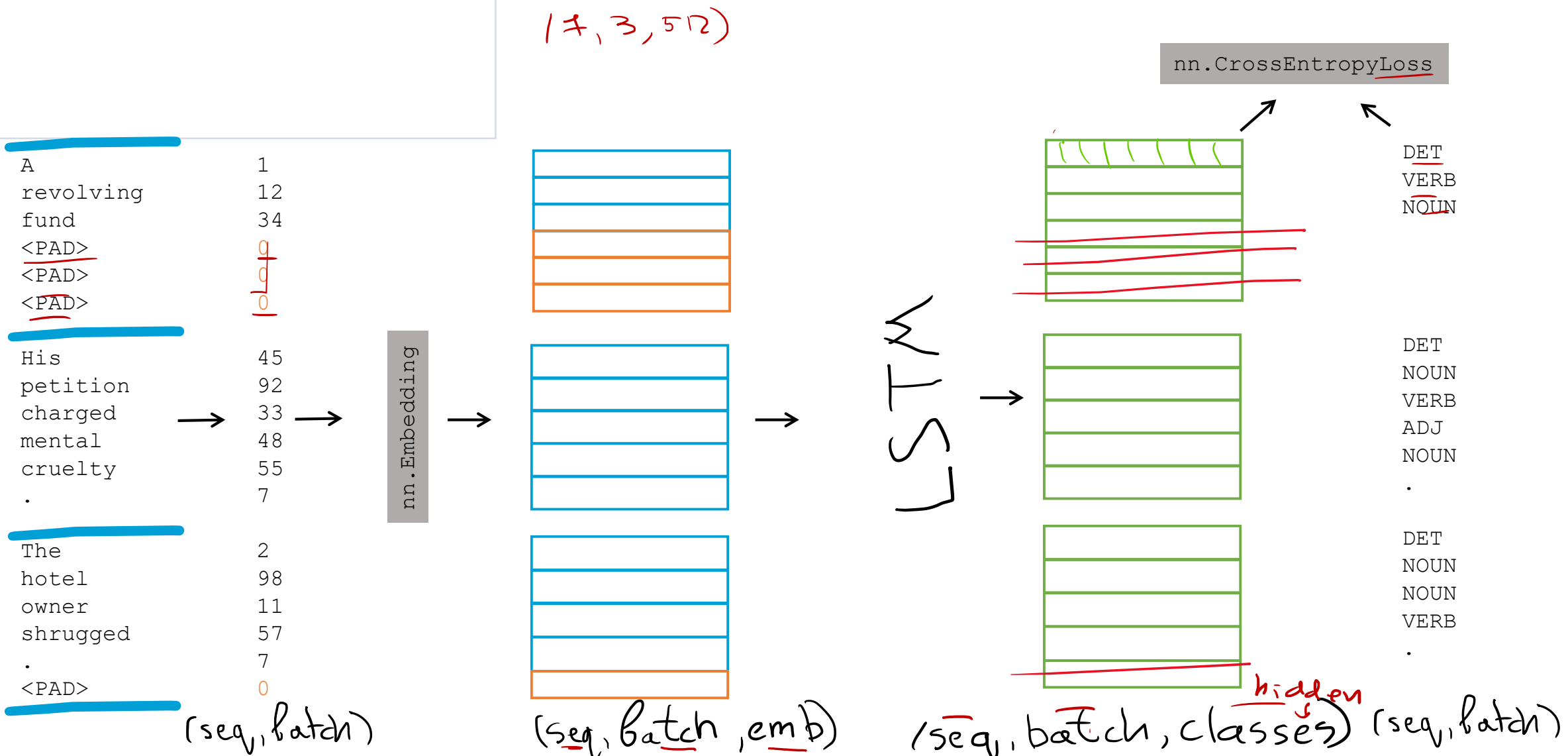
batch_size = 3

Brown corpus, NLTK

Например, Part of Speech Tagging

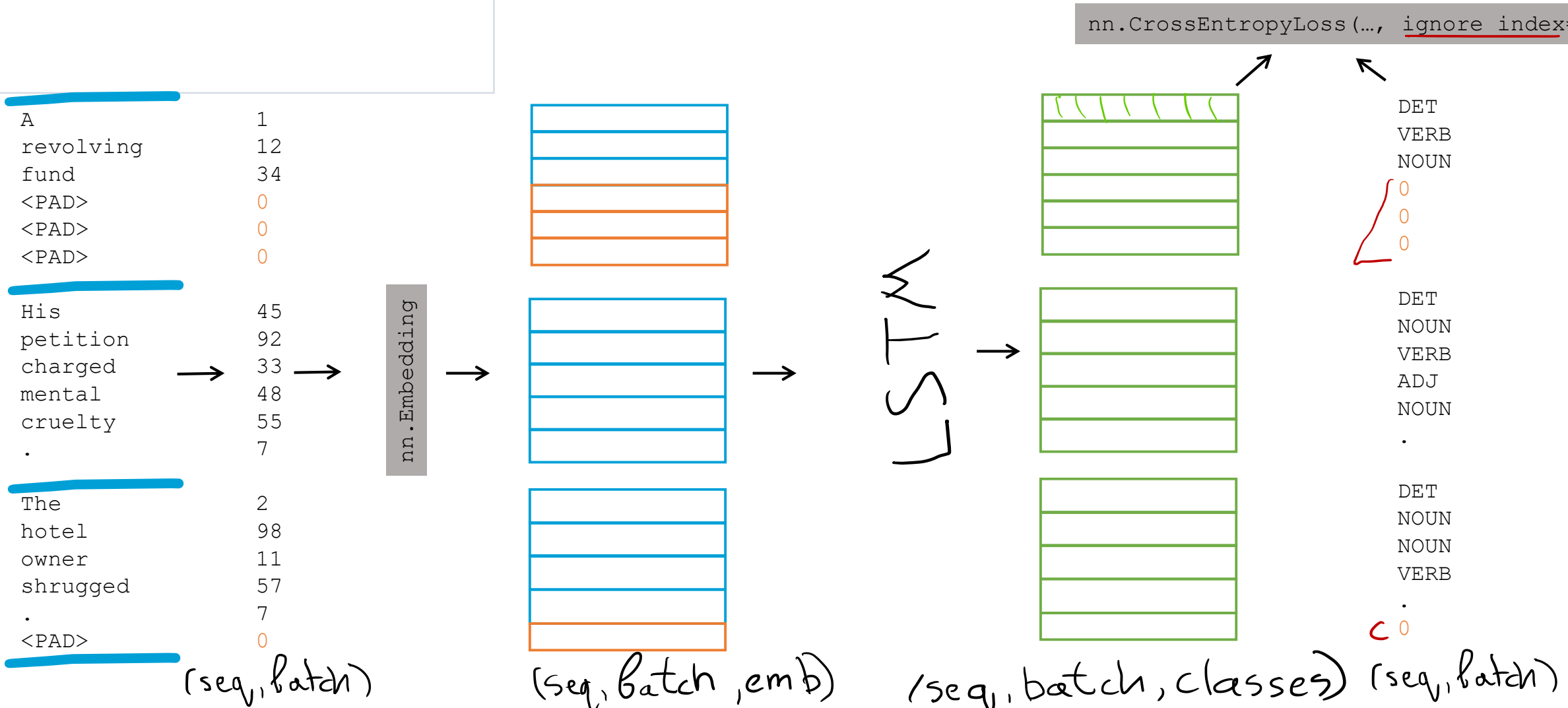


Например, Part of Speech Tagging



Например, Part of Speech Tagging

Video: CNN + LSTM



Бонус для тех, кто делает задания!

2.1, 2.2
3.1, 3.2