

ORDERED NEURONS: INTEGRATING TREE STRUCTURES INTO RECURRENT NEURAL NETWORKS

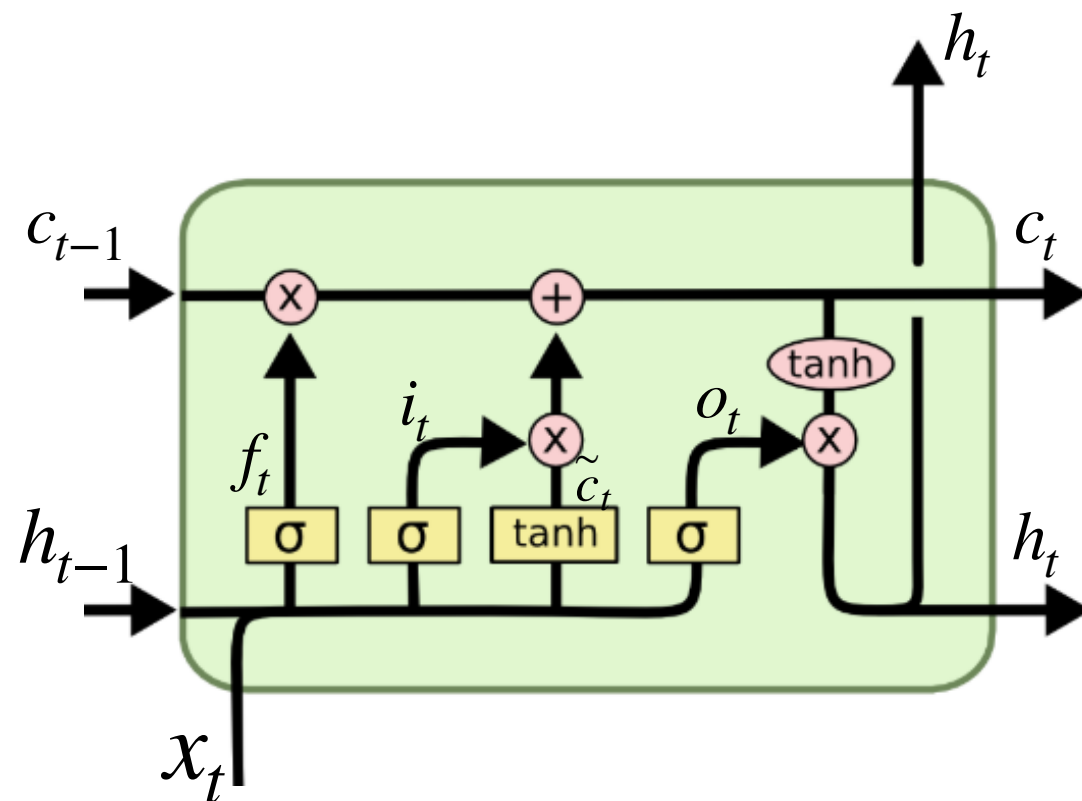
Николай Пальчиков БПМИ162
НИУ ВШЭ
2019

1) Сам язык в речи и письме имеет последовательную структуру.

- 1) Сам язык в речи и письме имеет последовательную структуру.**
- 2) Структура языка древовидна.**

- 1) Сам язык в речи и письме имеет последовательную структуру.**
- 2) Структура языка древовидна.**
- 3) Есть работы, подтверждающие, что LSTM с достаточной вместимостью может неявно учитывать иерархические зависимости.
(Gulordava et al. (2018); Kuncoro et al. (2018) Lakretz et al. (2019))**

Vanilla LSTM



$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

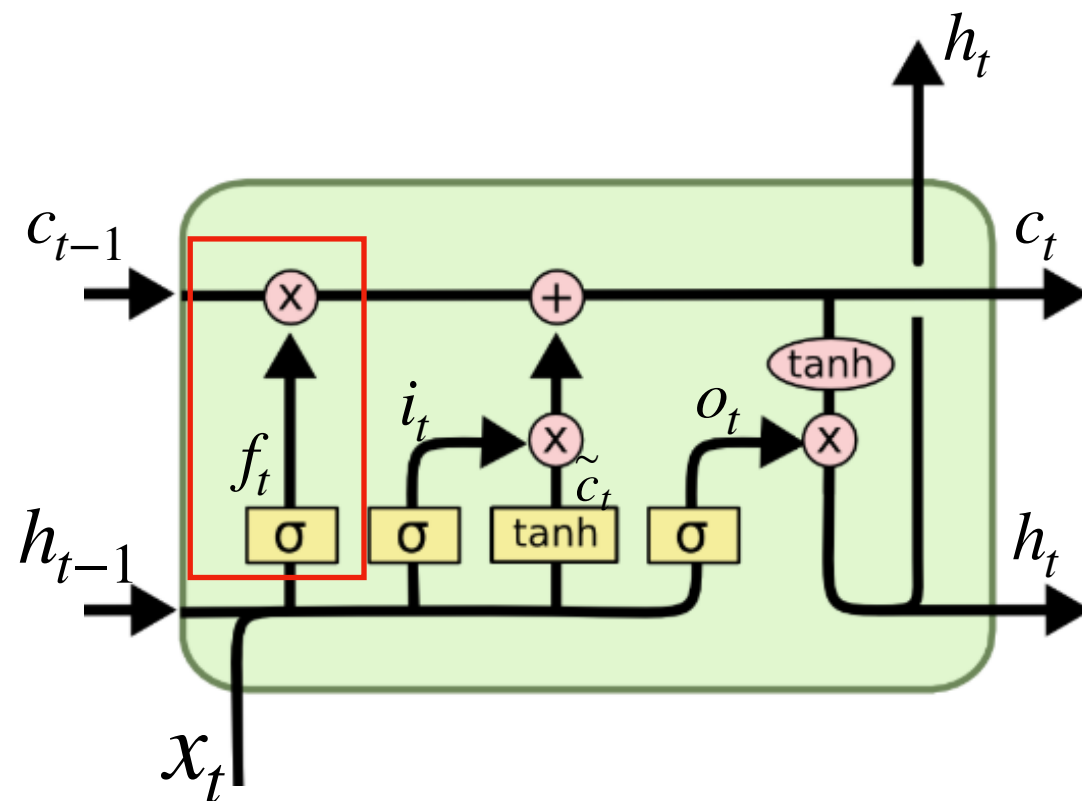
$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = c_{t-1} \odot f_t + \tilde{c}_t \odot i_t$$

$$h_t = \tanh(c_t) \odot o_t$$

Vanilla LSTM



$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

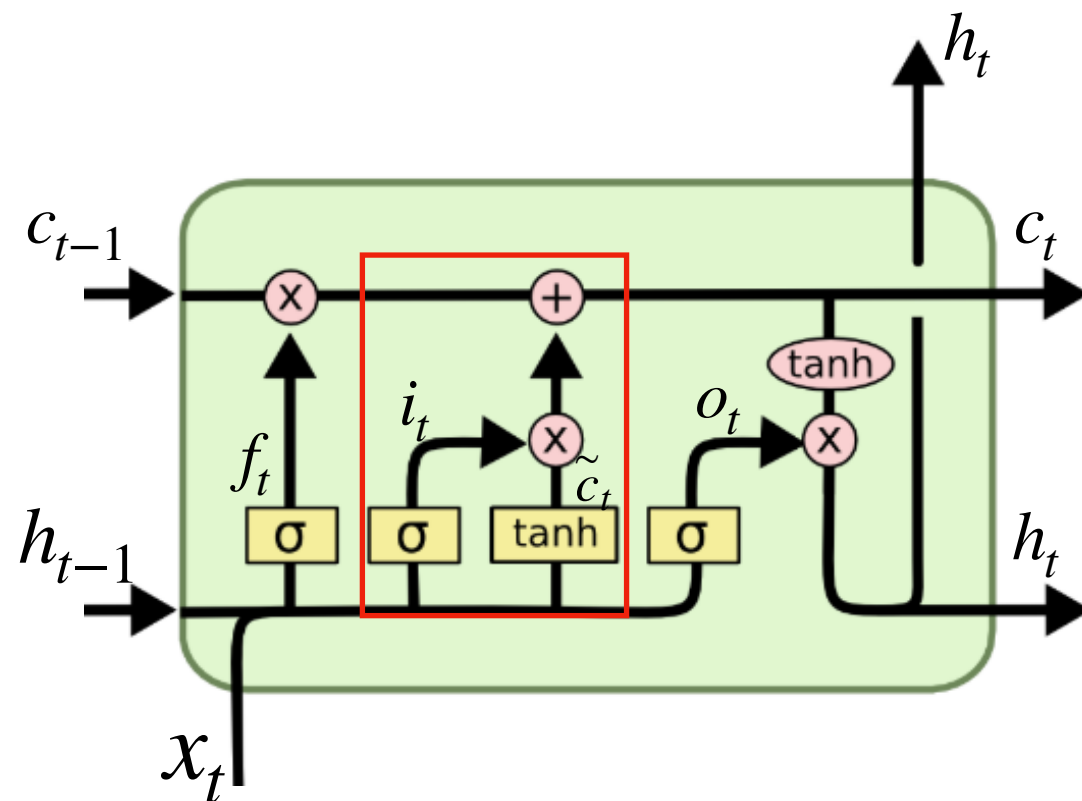
$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = c_{t-1} \odot f_t + \tilde{c}_t \odot i_t$$

$$h_t = \tanh(c_t) \odot o_t$$

Vanilla LSTM



$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

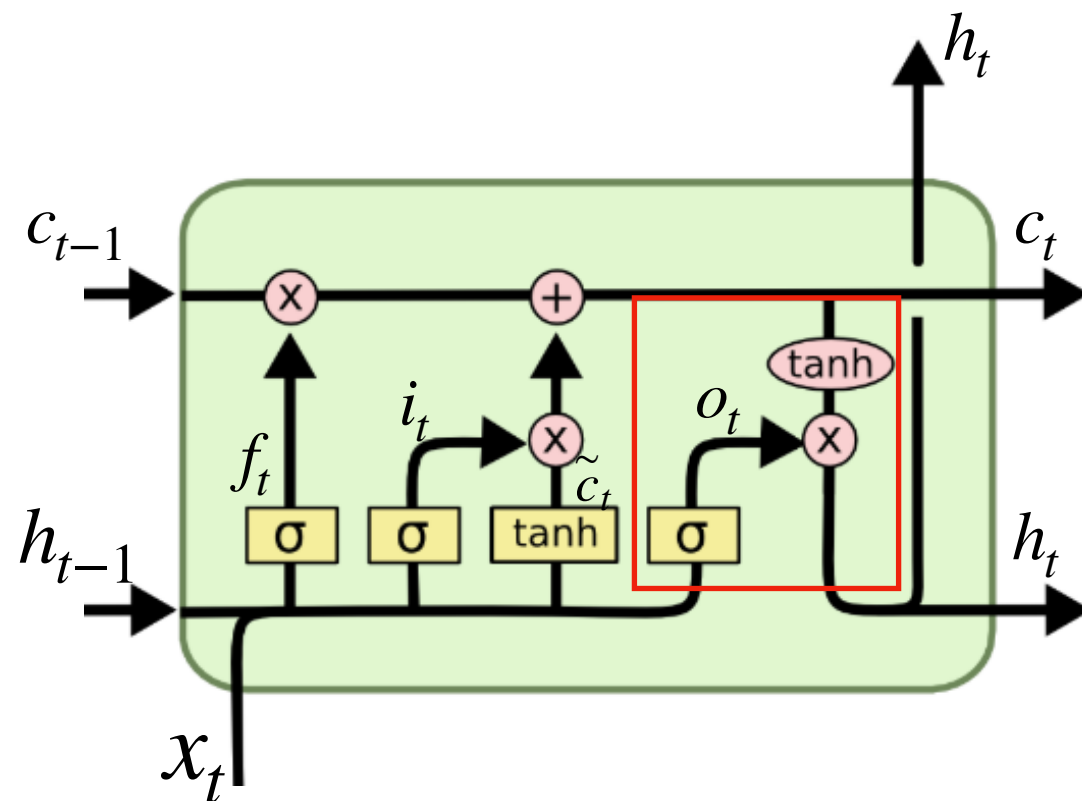
$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = c_{t-1} \odot f_t + \tilde{c}_t \odot i_t$$

$$h_t = \tanh(c_t) \odot o_t$$

Vanilla LSTM



$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

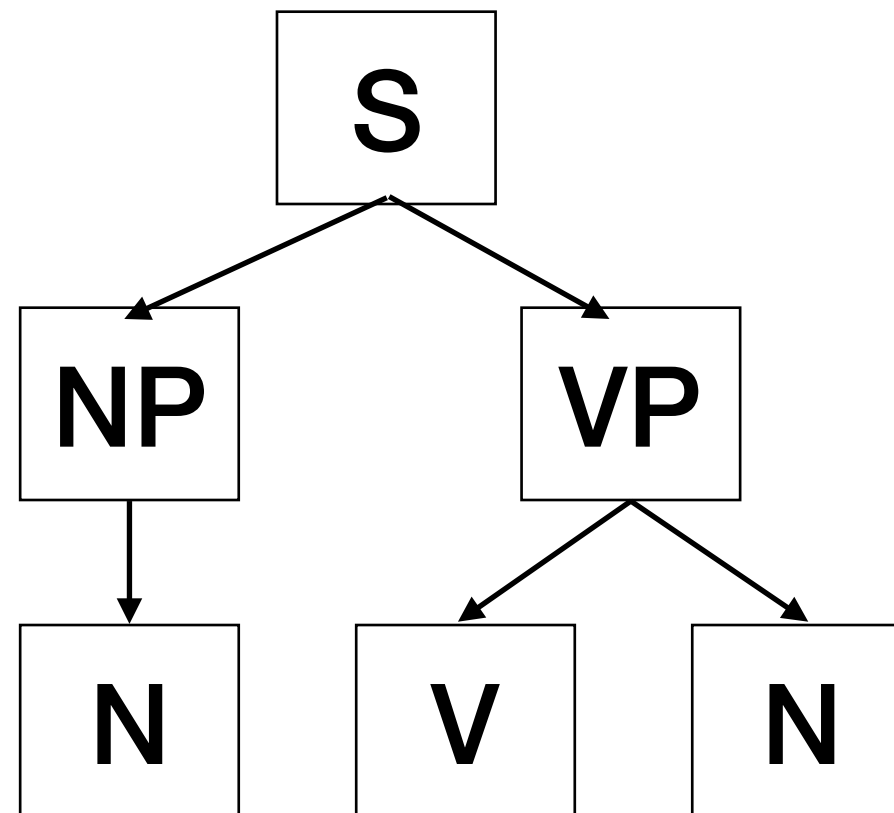
$$c_t = c_{t-1} \odot f_t + \tilde{c}_t \odot i_t$$

$$h_t = \tanh(c_t) \odot o_t$$

Предположение: язык имеет иерархическую структуру:

Constituency-based parse tree для предложения:

John sees Bill



***S = Sentence**

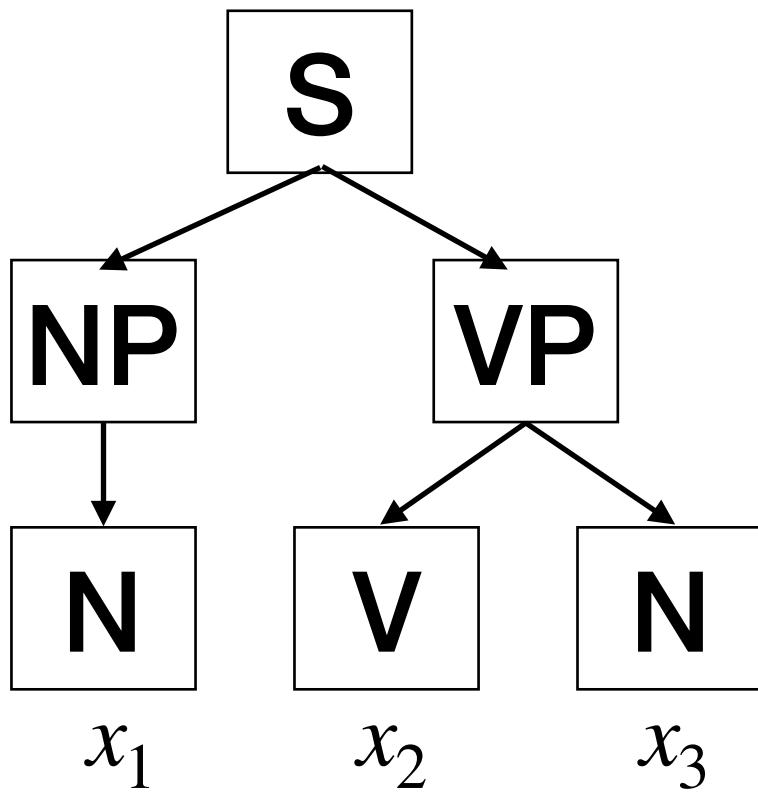
NP = Noun Phrase

VP = Verb Phrase

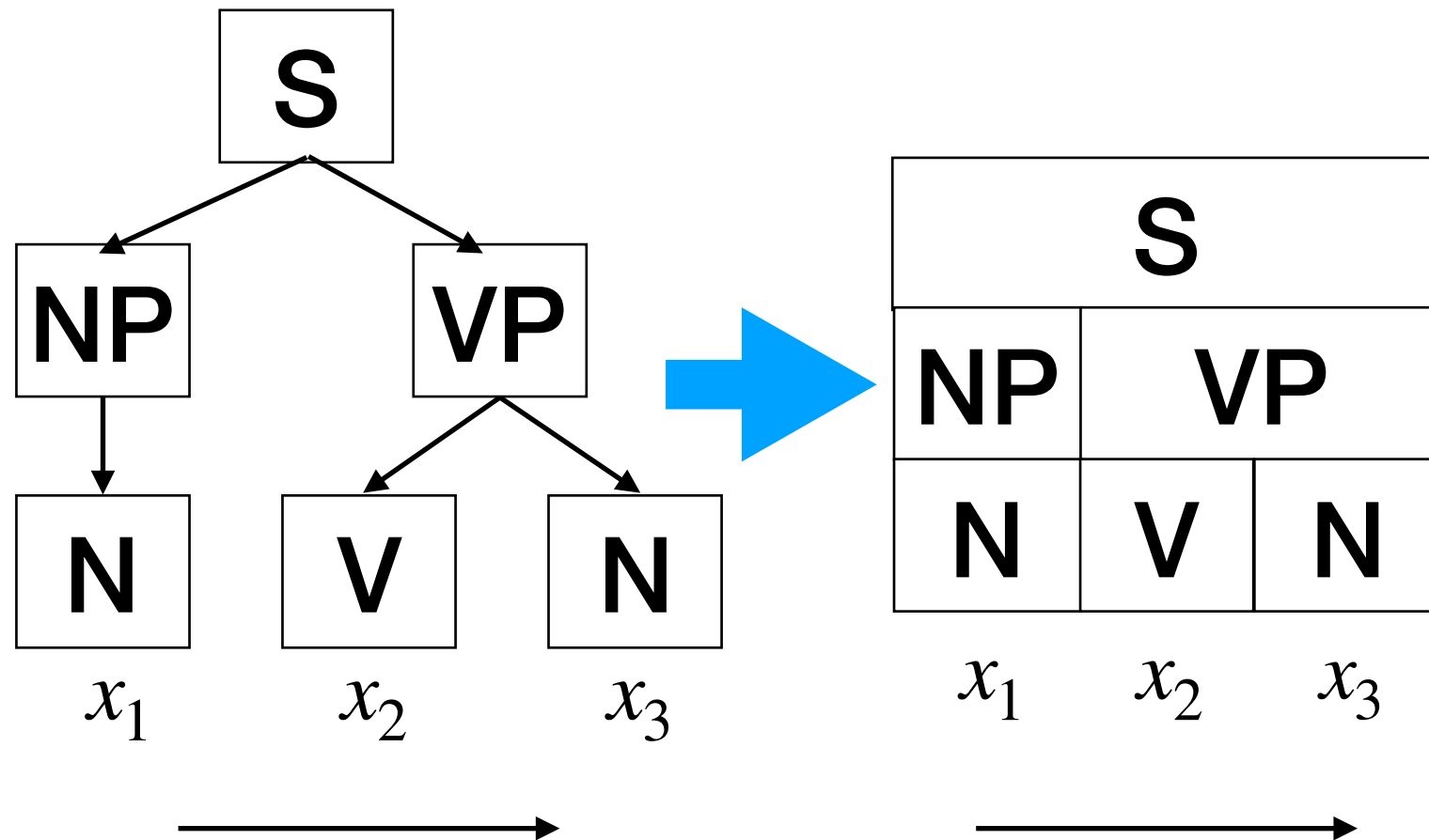
N = Noun

V = Verb

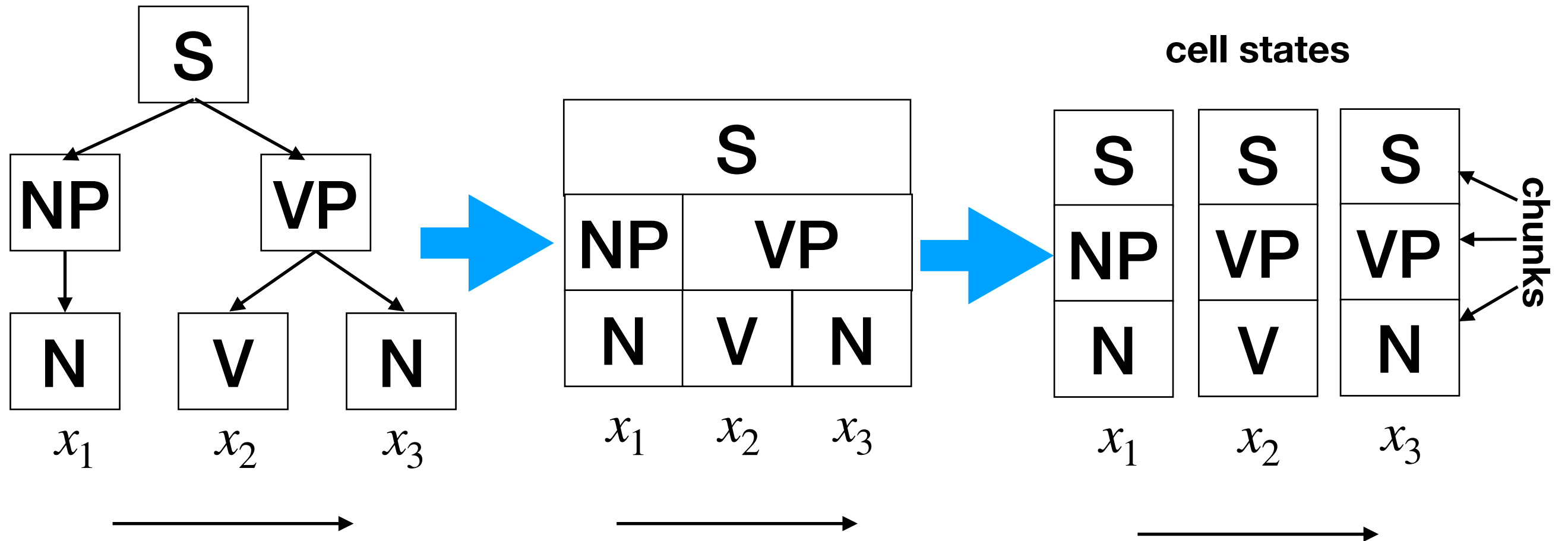
Constituent (единое целое) — слово или набор слов, функционирующих как единое целое в предложении.



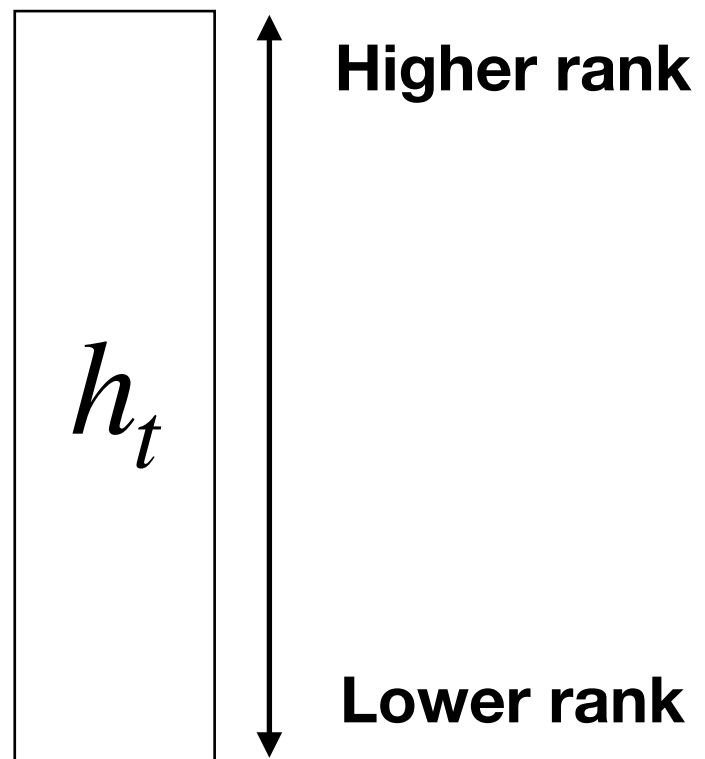
Когда заканчивается составляющая,
заканчиваются и все её наследники



Когда заканчивается составляющая,
заканчиваются и все её наследники

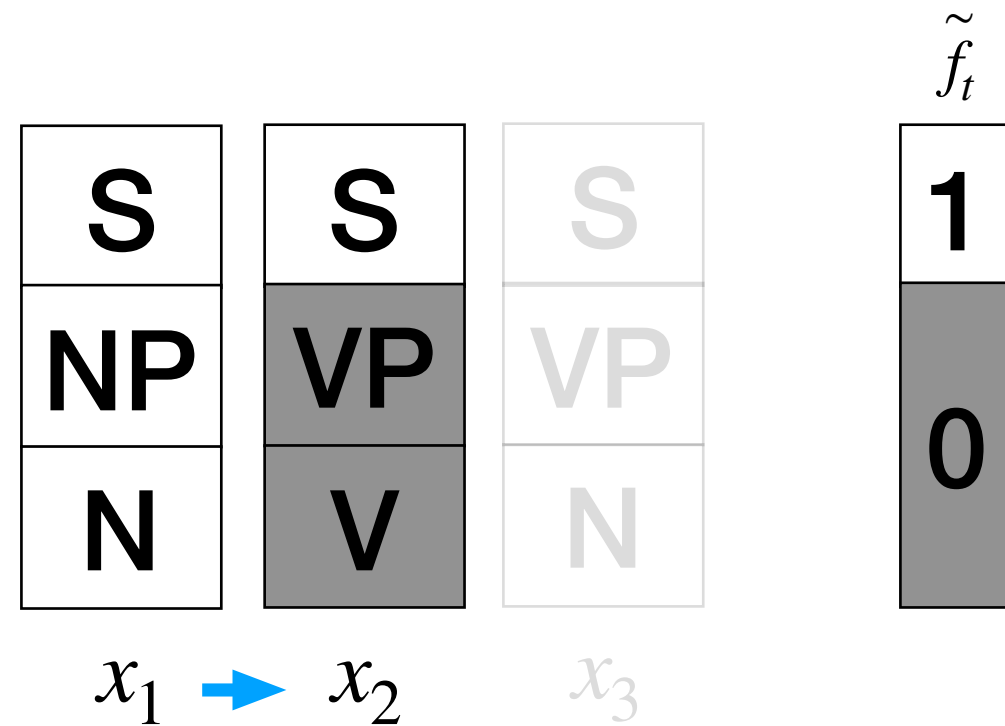
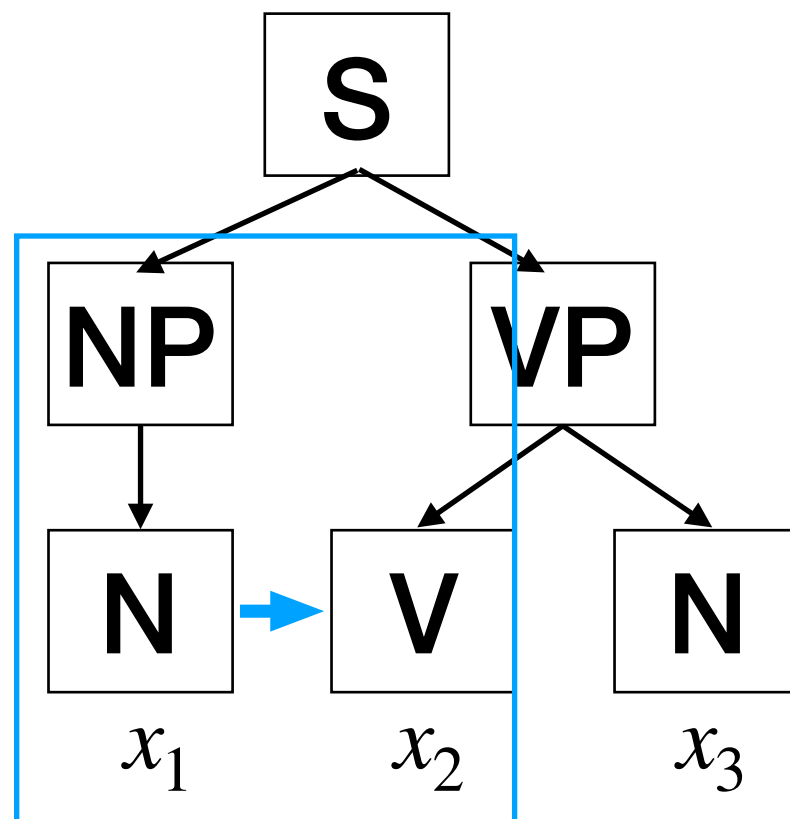


Ordered Neurons



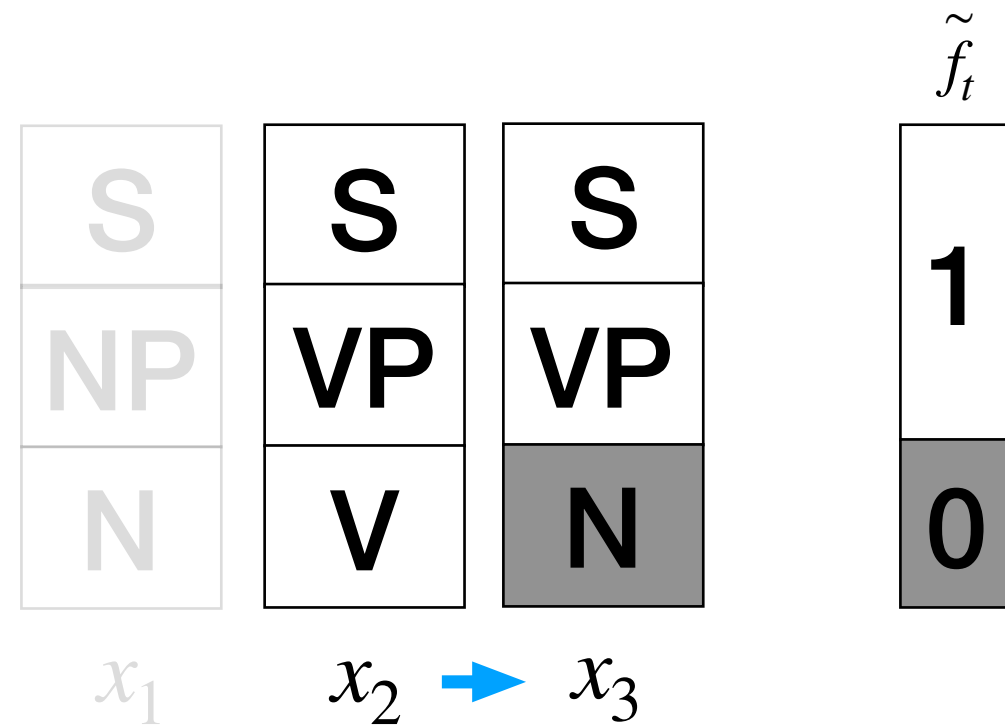
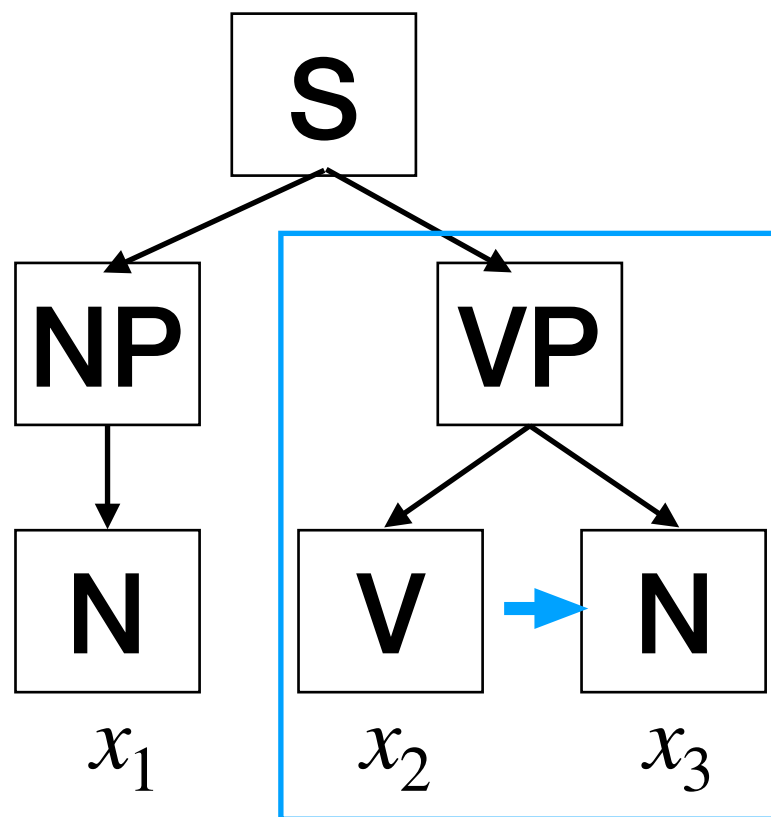
Когда заканчивается составляющая,
заканчиваются и все её наследники

Смоделируем процесс:



Когда заканчивается составляющая,
заканчиваются и все её наследники

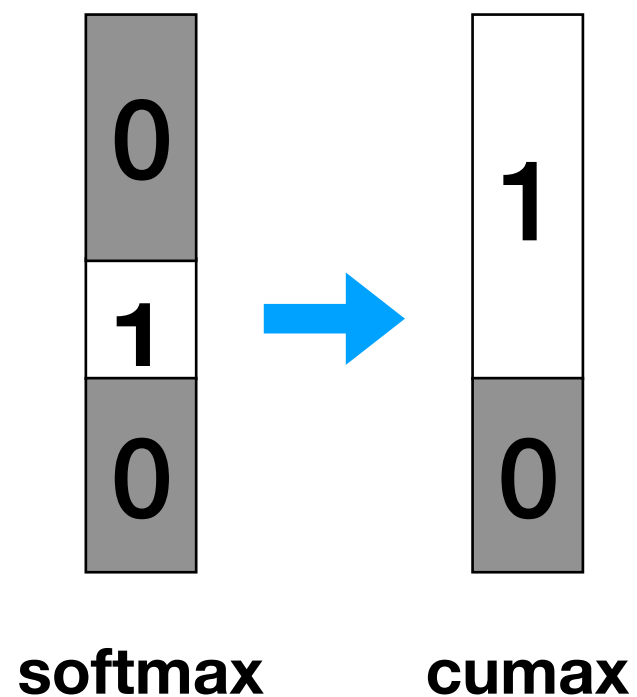
Смоделируем процесс:



Функция активации CUMAX

$$\hat{g} = cumax(\cdot) = cumsum(softmax(\cdot))$$

$$\hat{g}_k = cumax(\cdot)_k = \sum_{i=1}^k softmax(\cdot)_i$$



\hat{g} — математическое ожидание
разделяющего вектора

$$g = (0, 0, \dots, 0, 1, 1, \dots, 1)$$

Функция активации CUMAX

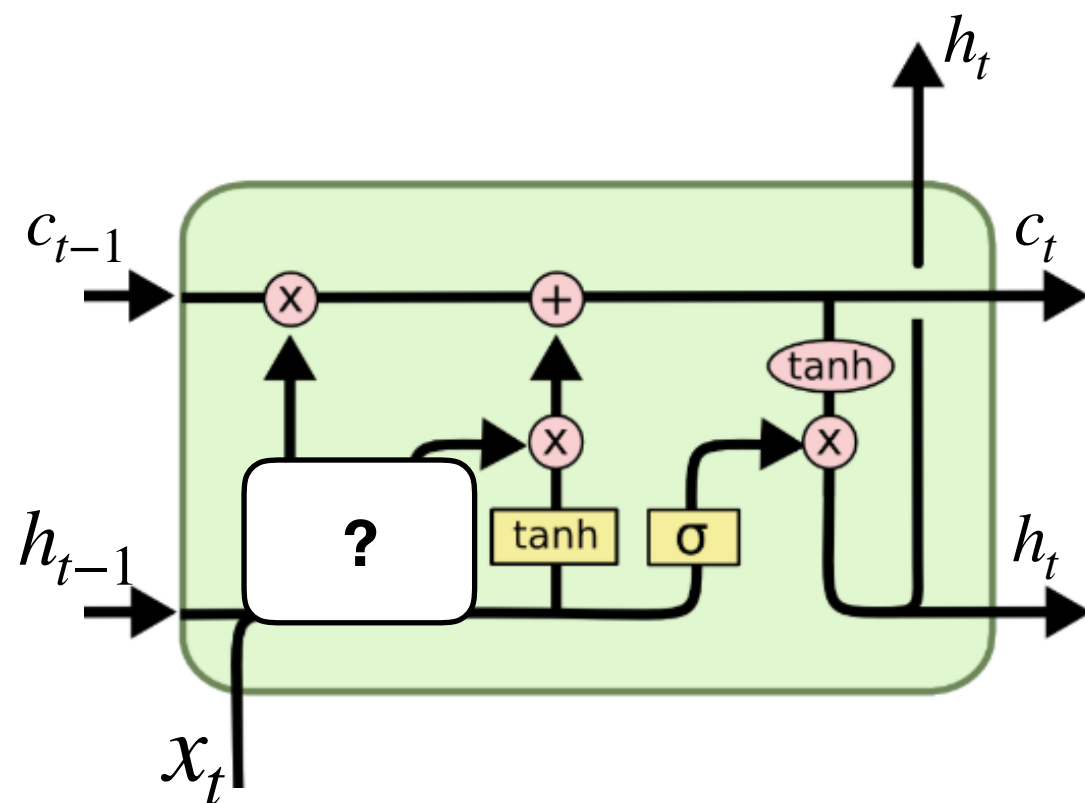
$$\hat{g} = cumax(\cdot) = cumsum(softmax(\cdot))$$
$$\hat{g}_k = cumax(\cdot)_k = \sum_{i=1}^k softmax(\cdot)_i$$

Пусть g – случайный разделяющий вектор вида $(0, 0, \dots, 0, 1, 1, \dots, 1)$. Пусть d – случайная категориальная переменная, отвечающая за первую единицу в g .

$$p(d) = softmax(\cdot)$$

$$p(g_k = 1) = p(d \leq k) = \sum_{i \leq k} p(d = i) = \hat{g}_i \Rightarrow$$

$$\Rightarrow [g \text{ is binary}] \hat{g} = \mathbb{E}(g)$$



$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$\hat{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

~~$$c_t = c_{t-1} \odot f_t + \hat{c}_t \odot i_t$$~~

$$h_t = \tanh(c_t) \odot o_t$$

New gates:

Master forget gate:

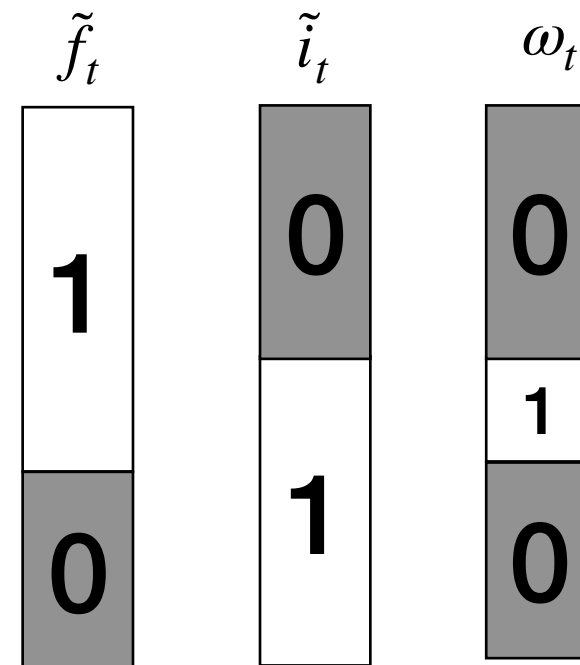
$$\tilde{f}_t = \text{cumsum}(W_{\tilde{f}}x_t + U_{\tilde{f}}h_{t-1} + b_{\tilde{f}})$$

Master input gate:

$$\tilde{i}_t = 1 - \text{cumsum}(W_{\tilde{i}}x_t + U_{\tilde{i}}h_{t-1} + b_{\tilde{i}})$$

Overlap:

$$\omega_t = \tilde{f}_t \odot \tilde{i}_t$$



$$\begin{aligned}\tilde{f}_t &= cumsum(W_{\tilde{f}}x_t + U_{\tilde{f}}h_{t-1} + b_{\tilde{f}}) \\ \tilde{i}_t &= 1 - cumsum(W_{\tilde{i}}x_t + U_{\tilde{i}}h_{t-1} + b_{\tilde{i}}) \\ \omega_t &= \tilde{f}_t \odot \tilde{i}_t \\ f_t &= \sigma(W_fx_t + U_fh_{t-1} + b_f) \\ i_t &= \sigma(W_ix_t + U_ih_{t-1} + b_i)\end{aligned}$$

$$\begin{aligned}\hat{f}_t &= f_t \odot \omega_t + (\tilde{f}_t - w_t) \\ \hat{i}_t &= i_t \odot \omega_t + (\tilde{i}_t - w_t)\end{aligned}$$

$$\begin{aligned}\hat{c}_t &= \tanh(W_cx_t + U_ch_{t-1} + b_c) \\ o_t &= \sigma(W_ox_t + U_oh_{t-1} + b_o)\end{aligned}$$

~~$$c_t = c_{t-1} \odot f_t + \hat{c} \odot i_t$$~~


$$h_t = \tanh(c_t) \odot o_t$$

$$\begin{aligned}
\tilde{f}_t &= \text{cumsum}(W_{\tilde{f}}x_t + U_{\tilde{f}}h_{t-1} + b_{\tilde{f}}) \\
\tilde{i}_t &= 1 - \text{cumsum}(W_{\tilde{i}}x_t + U_{\tilde{i}}h_{t-1} + b_{\tilde{i}}) \\
\omega_t &= \tilde{f}_t \odot \tilde{i}_t \\
f_t &= \sigma(W_fx_t + U_fh_{t-1} + b_f) \\
i_t &= \sigma(W_ix_t + U_ih_{t-1} + b_i)
\end{aligned}$$

$$\begin{aligned}
\hat{f}_t &= f_t \odot \omega_t + (\tilde{f}_t - w_t) \\
\hat{i}_t &= i_t \odot \omega_t + (\tilde{i}_t - w_t)
\end{aligned}$$

$$\begin{aligned}
\hat{c}_t &= \tanh(W_cx_t + U_ch_{t-1} + b_c) \\
o_t &= \sigma(W_ox_t + U_oh_{t-1} + b_o) \\
c_t &= c_{t-1} \odot f_t + \hat{c} \odot i_t \\
h_t &= \tanh(c_t) \odot o_t
\end{aligned}$$

Controls erasing mechanism


$$\begin{aligned}\tilde{f}_t &= \text{cumsum}(W_{\tilde{f}}x_t + U_{\tilde{f}}h_{t-1} + b_{\tilde{f}}) \\ \tilde{i}_t &= 1 - \text{cumsum}(W_{\tilde{i}}x_t + U_{\tilde{i}}h_{t-1} + b_{\tilde{i}}) \\ \omega_t &= \tilde{f}_t \odot \tilde{i}_t \\ f_t &= \sigma(W_fx_t + U_fh_{t-1} + b_f) \\ i_t &= \sigma(W_ix_t + U_ih_{t-1} + b_i)\end{aligned}$$

$$\begin{aligned}\hat{f}_t &= f_t \odot \omega_t + (\tilde{f}_t - w_t) \\ \hat{i}_t &= i_t \odot \omega_t + (\tilde{i}_t - w_t)\end{aligned}$$

$$\begin{aligned}\hat{c}_t &= \tanh(W_cx_t + U_ch_{t-1} + b_c) \\ o_t &= \sigma(W_ox_t + U_oh_{t-1} + b_o) \\ c_t &= c_{t-1} \odot \hat{f}_t + \hat{c} \odot \hat{i}_t \\ h_t &= \tanh(c_t) \odot o_t\end{aligned}$$

Controls erasing mechanism

Controls writing mechanism

$$\tilde{f}_t = \text{cumsum}(W_{\tilde{f}}x_t + U_{\tilde{f}}h_{t-1} + b_{\tilde{f}})$$

$$\tilde{i}_t = 1 - \text{cumsum}(W_{\tilde{i}}x_t + U_{\tilde{i}}h_{t-1} + b_{\tilde{i}})$$

$$\omega_t = \tilde{f}_t \odot \tilde{i}_t$$

$$f_t = \sigma(W_fx_t + U_fh_{t-1} + b_f)$$

$$i_t = \sigma(W_ix_t + U_ih_{t-1} + b_i)$$

$$\hat{f}_t = f_t \odot \omega_t + (\tilde{f}_t - w_t)$$

$$\hat{i}_t = i_t \odot \omega_t + (\tilde{i}_t - w_t)$$

$$\hat{c}_t = \tanh(W_cx_t + U_ch_{t-1} + b_c)$$

$$o_t = \sigma(W_ox_t + U_oh_{t-1} + b_o)$$

$$c_t = c_{t-1} \odot \hat{f}_t + \hat{c} \odot \hat{i}_t$$

$$h_t = \tanh(c_t) \odot o_t$$

Controls erasing mechanism

$$\tilde{f}_t = \text{cumsum}(W_{\tilde{f}}x_t + U_{\tilde{f}}h_{t-1} + b_{\tilde{f}})$$

Controls writing mechanism

$$\tilde{i}_t = 1 - \text{cumsum}(W_{\tilde{i}}x_t + U_{\tilde{i}}h_{t-1} + b_{\tilde{i}})$$

Incomplete constituent

$$\omega_t = \tilde{f}_t \odot \tilde{i}_t$$

$$f_t = \sigma(W_fx_t + U_fh_{t-1} + b_f)$$

$$i_t = \sigma(W_ix_t + U_ih_{t-1} + b_i)$$

$$\hat{f}_t = f_t \odot \omega_t + (\tilde{f}_t - w_t)$$

$$\hat{i}_t = i_t \odot \omega_t + (\tilde{i}_t - w_t)$$

$$\hat{c}_t = \tanh(W_cx_t + U_ch_{t-1} + b_c)$$

$$o_t = \sigma(W_ox_t + U_oh_{t-1} + b_o)$$

$$c_t = c_{t-1} \odot \hat{f}_t + \hat{c} \odot \hat{i}_t$$

$$h_t = \tanh(c_t) \odot o_t$$

На практике моделировать master input gate и master output gate такого же размера как hidden state бессмысленно и затратно.

Авторы предлагают делать их размера $\frac{D}{C}$, D — hidden state size, C — chunk size

Эксперименты

Language Modeling

Model	Parameters	Validation	Test
Zaremba et al. (2014) - LSTM (large)	66M	82.2	78.4
Gal & Ghahramani (2016) - Variational LSTM (large, MC)	66M	—	73.4
Kim et al. (2016) - CharCNN	19M	—	78.9
Merity et al. (2016) - Pointer Sentinel-LSTM	21M	72.4	70.9
Grave et al. (2016) - LSTM	—	—	82.3
Grave et al. (2016) - LSTM + continuous cache pointer	—	—	72.1
Inan et al. (2016) - Variational LSTM (tied) + augmented loss	51M	71.1	68.5
Zilly et al. (2016) - Variational RHN (tied)	23M	67.9	65.4
Zoph & Le (2016) - NAS Cell (tied)	54M	—	62.4
Shen et al. (2017) - PRPN-LM	—	—	62.0
Melis et al. (2017) - 4-layer skip connection LSTM (tied)	24M	60.9	58.3
Merity et al. (2017) - AWD-LSTM - 3-layer LSTM (tied)	24M	60.0	57.3
ON-LSTM - 3-layer (tied)	25M	58.29 ± 0.10	56.17 ± 0.12
Yang et al. (2017) - AWD-LSTM-MoS*	22M	56.5	54.4

Table 1: Single model perplexity on validation and test sets for the Penn Treebank language modeling task. Models labelled *tied* use weight tying on the embedding and softmax weights (Inan et al., 2016; Press & Wolf, 2017). Models labelled * focus on improving the softmax component of RNN language model. Their contribution is orthogonal to ours.

UNSUPERVISED CONSTITUENCY PARSING

1) $\hat{d}_t^f = D_m - \sum_{k=1}^{D_m} \tilde{f}_{tk}$ D_m — размер hidden state
 \tilde{f}_{tk} — k-й элемент master forget gate на шаге t

UNSUPERVISED CONSTITUENCY PARSING

- 1) $\hat{d}_t^f = D_m - \sum_{k=1}^{D_m} \tilde{f}_{tk}$ D_m — размер hidden state
 \tilde{f}_{tk} — k-й элемент master forget gate на шаге t
- 2) Отсортируем \hat{d}_t^f по возрастанию

UNSUPERVISED CONSTITUENCY PARSING

$$\mathbf{1)} \quad \hat{d}_t^f = D_m - \sum_{k=1}^{D_m} \tilde{f}_{tk} \quad \begin{array}{l} D_m - \text{размер hidden state} \\ \tilde{f}_{tk} - \text{k-й элемент master forget gate на шаге } t \end{array}$$

2) Отсортируем \hat{d}_t^f по возрастанию

Посмотрим на первый \hat{d}_i^t в отсортированной последовательности.

3) Разобьем входную последовательность на синтаксические составляющие $\{x_{k < i}\}, \{x_{k > i}\}$

UNSUPERVISED CONSTITUENCY PARSING

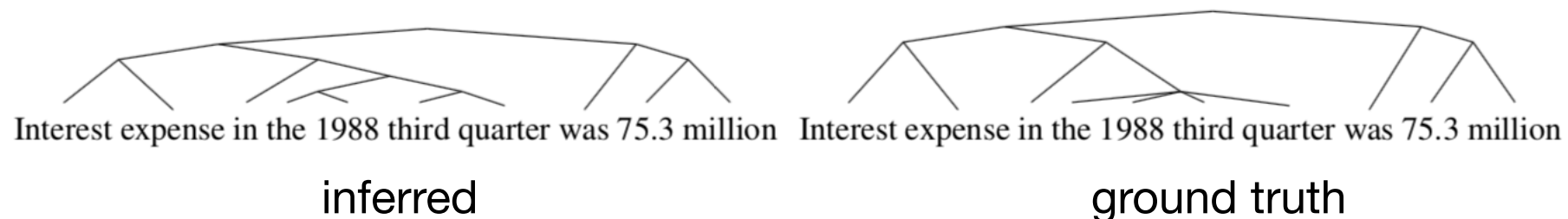
1) $\hat{d}_t^f = D_m - \sum_{k=1}^{D_m} \tilde{f}_{tk}$ D_m — размер hidden state
 \tilde{f}_{tk} — k-й элемент master forget gate на шаге t

2) Отсортируем \hat{d}_t^f по возрастанию

Посмотрим на первый \hat{d}_i^t в отсортированной последовательности.

3) Разобьем входную последовательность на синтаксические составляющие $\{x_{k < i}\}, \{x_{k \geq i}\}$

4) Для $\{x_{k < i}\}, \{x_{k \geq i}\}$ рекурсивно повторим шаги 1-3



LOGICAL INFERENCE

С помощью рекуррентной архитектуры посчитаем эмбединги двух предложений — h_1, h_2

На вход классификатору подадим $(h_1, h_2, h_1 \odot h_2, abs(h_1 - h_2))$

LOGICAL INFERENCE

	ON-LSTM	LSTM
Short-Term Dependency		
SUBJECT-VERB AGREEMENT:		
Simple	0.99	1.00
In a sentential complement	0.95	0.98
Short VP coordination	0.89	0.92
In an object relative clause	0.84	0.88
In an object relative (no <i>that</i>)	0.78	0.81
REFLEXIVE ANAPHORA:		
Simple	0.89	0.82
In a sentential complement	0.86	0.80
NEGATIVE POLARITY ITEMS:		
Simple (grammatical vs. intrusive)	0.18	1.00
Simple (intrusive vs. ungrammatical)	0.50	0.01
Simple (grammatical vs. ungrammatical)	0.07	0.63
Long-Term Dependency		
SUBJECT-VERB AGREEMENT:		
Long VP coordination	0.74	0.74
Across a prepositional phrase	0.67	0.68
Across a subject relative clause	0.66	0.60
Across an object relative clause	0.57	0.52
Across an object relative (no <i>that</i>)	0.54	0.51
REFLEXIVE ANAPHORA:		
Across a relative clause	0.57	0.58
NEGATIVE POLARITY ITEMS:		
Across a relative clause (grammatical vs. intrusive)	0.59	0.95
Across a relative clause (intrusive vs. ungrammatical)	0.20	0.00
Across a relative clause (grammatical vs. ungrammatical)	0.11	0.04

Источники:

Изображение схемы LSTM: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Статья: <https://arxiv.org/abs/1810.09536>

Вопросы:

- 1) Выведите cusum loss как математическое ожидание разделяющего вектора $g = (0, 0, 0, \dots, 0, 1, \dots, 1)$
- 2) В формулах для пересчёта внутреннего состояния c_t архитектуры ON-LSTM присутствует master forget gate. Запишите его формулу и объясните интуицию его работы.
- 3) ON-LSTM учитывает иерархические зависимости в текстовых данных. Эти зависимости канонично изображаются деревом синтаксического разбора. Разные нейроны в скрытом состоянии ON-LSTM отвечают за разные уровни в данном дереве. При этом листья в дереве разбора могут быть расположены на разной глубине. Как это учитывается в ON-LSTM?