Attention-based models

Elena Voita

Yandex, University of Amsterdam

lena-voita@yandex-team.ru

NOVEMBER 9, 2017

Plan

- Machine translation task
- A couple of words on Neural Networks
- Encoder-decoder model
- Bahdanau attention model
- Attention is all you need: Transformer
- Attention: other use cases

Machine translation task

$\boldsymbol{x} = (x_1, x_2, \ldots, x_{\mathrm{Tx}})$ - source sentence

$\boldsymbol{y} = (y_1, y_2, \ldots, y_{\mathrm{Ty}})$ - target sentence

Any type of machine translation system can be defined as a function

$$\hat{\boldsymbol{y}} = \mathrm{mt}(\boldsymbol{x})$$

Translation is equivalent to finding a target sentence that maximizes the conditional probability of $\boldsymbol{y}$ given a source sentence $\boldsymbol{x}$, i.e.

$$\arg\max_{y} \mathrm{p}(\boldsymbol{y}|\boldsymbol{x})$$

Machine translation task

Machine translation systems create a probabilistic model for the probability of $\boldsymbol{y}$ given $\boldsymbol{x}$,

$$p(\boldsymbol{y}|\boldsymbol{x}, \theta),$$

and find the target sentence that maximizes this probability:

$$\hat{\boldsymbol{y}} = \arg\max_{y} p(\boldsymbol{y}|\boldsymbol{x}, \theta).$$

(θ – the parameters of the model specifying the probability distribution)

Machine translation task

- Modeling

What the model $\mathrm{p}(\boldsymbol{y}|\boldsymbol{x}, \theta)$ will look like?

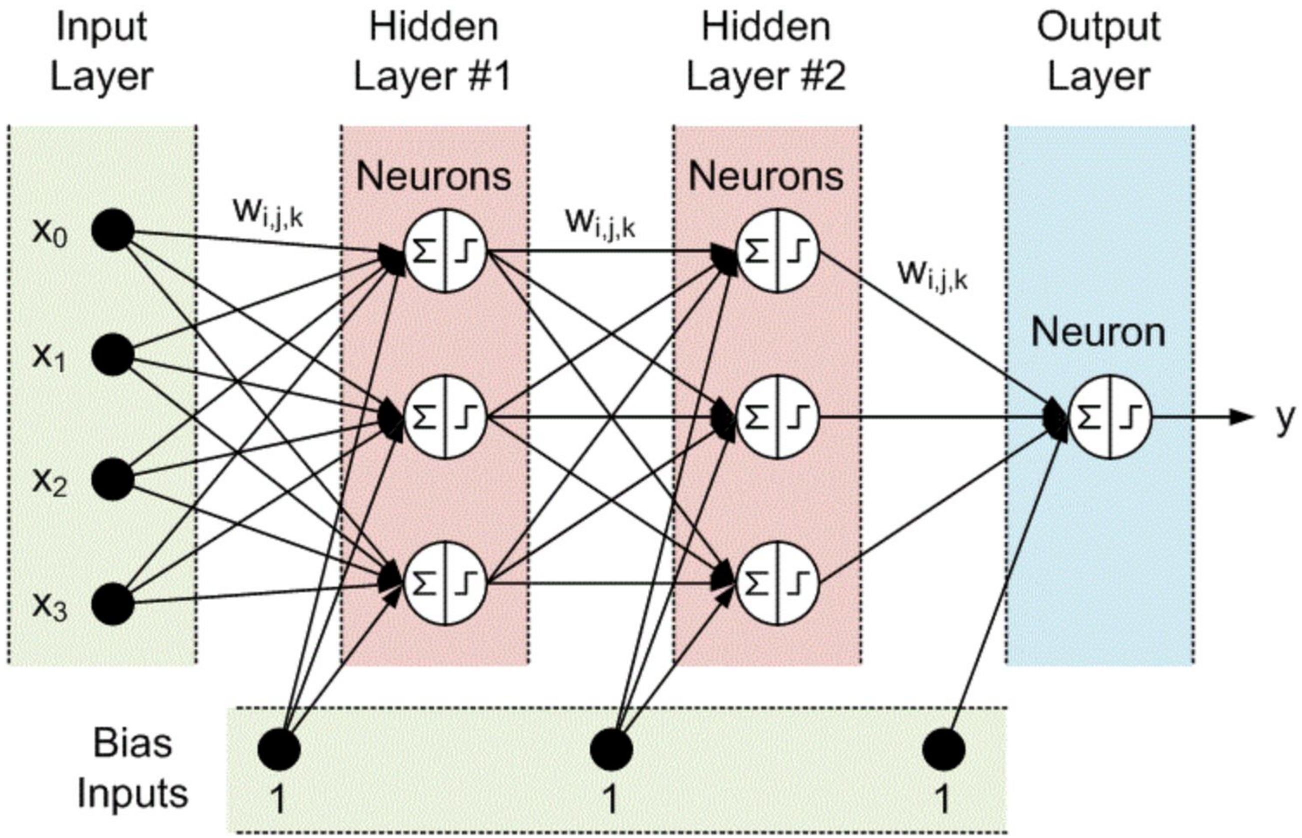
- Learning

We need a method to learn appropriate values for parameters θ from training data.

- Search

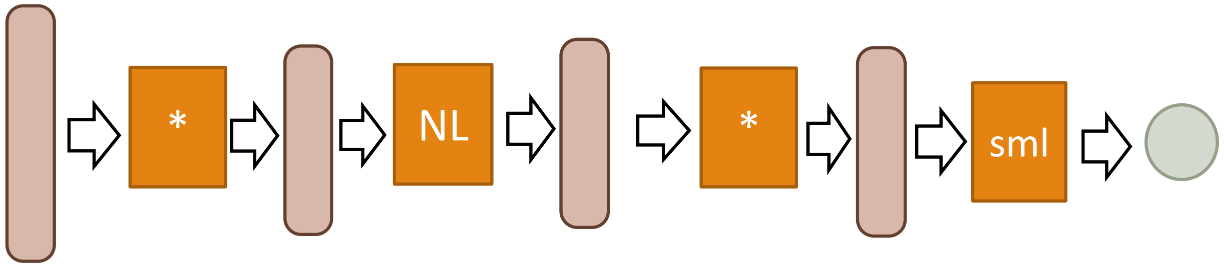
We need to solve the problem of finding the most probable sentence (solving “argmax”)

Neural Networks: preliminaries



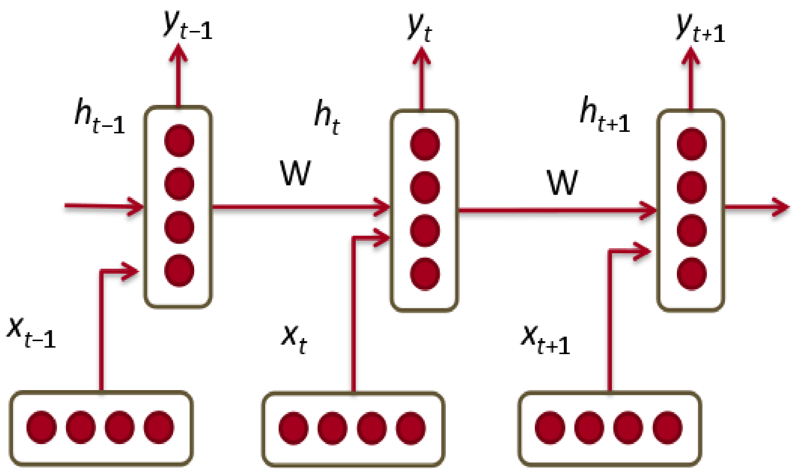
Neural Networks recap

- neural network is a composition of functions
- each layer represents a function from a particular family of functions
- constructing a network structure is equivalent to taking a composition of functions
- each layer has to be differentiable w.r.t. its inputs and parameters
- the whole network is then trained by gradient descent



Neural Networks: RNN

Recurrent neural network (RNN)



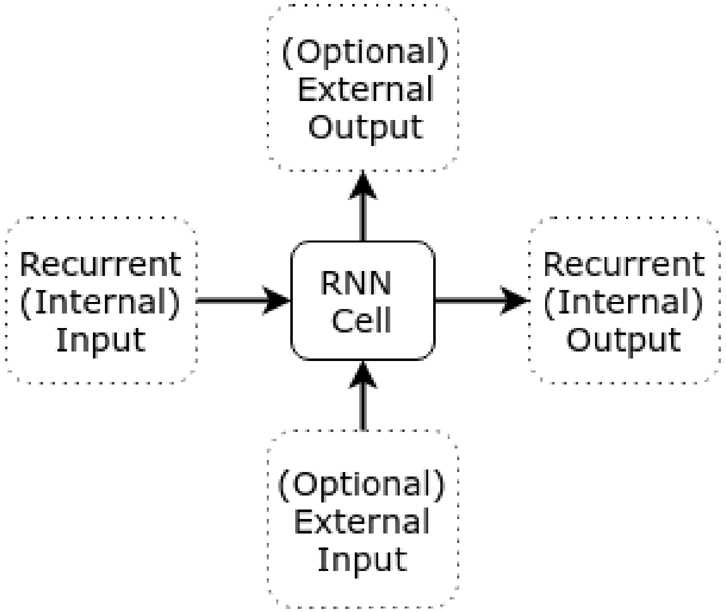
$x_1, x_2, ..., x_n$

$h_0 = 0$

$h_t = \tanh(h_{t-1}W_h + x_t W_x)$

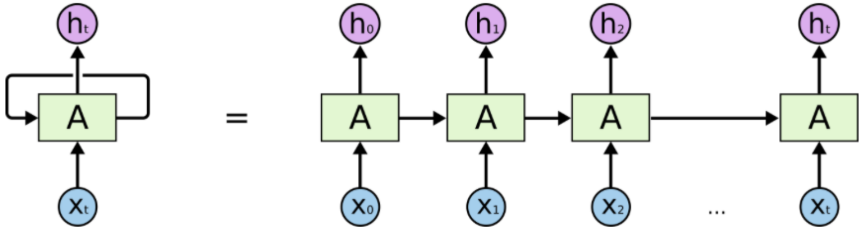
Последний вектор, h_n, содержит всю информацию про $x_1, ..., x_n$

Neural Networks: RNN (LSTM, GRU)

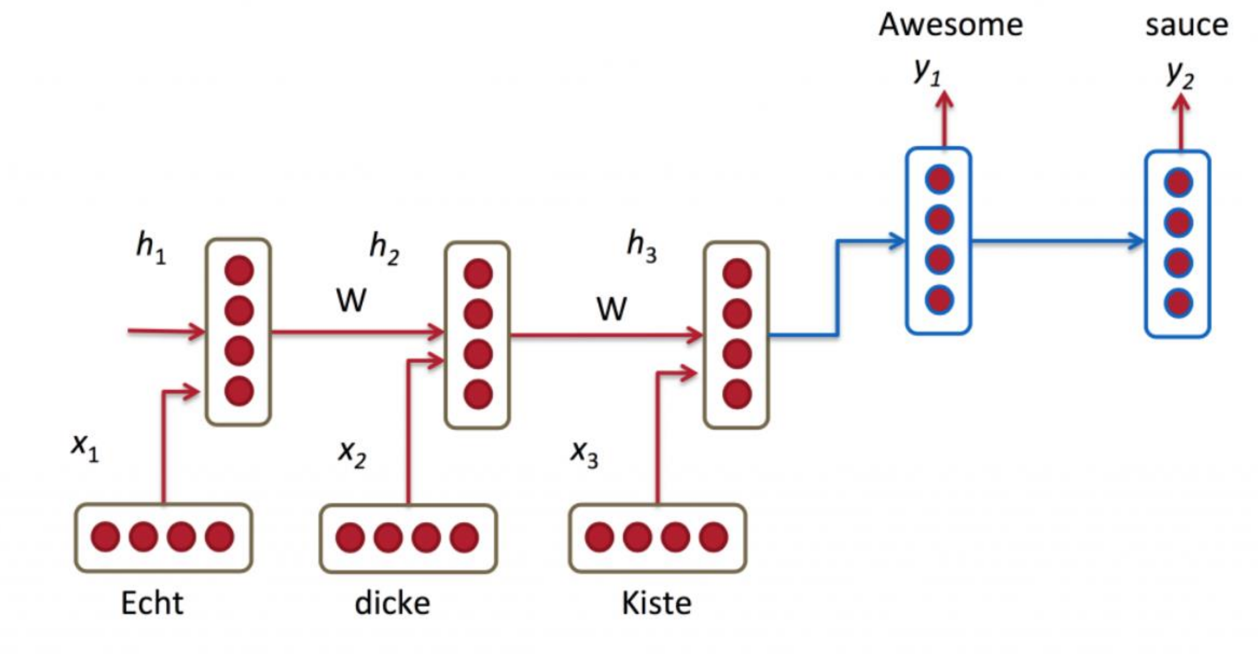


h_t – состояние

Ячейка берет предыдущее состояние h_{t-1} и добавляет к нему информацию об очередном элементе x_t .

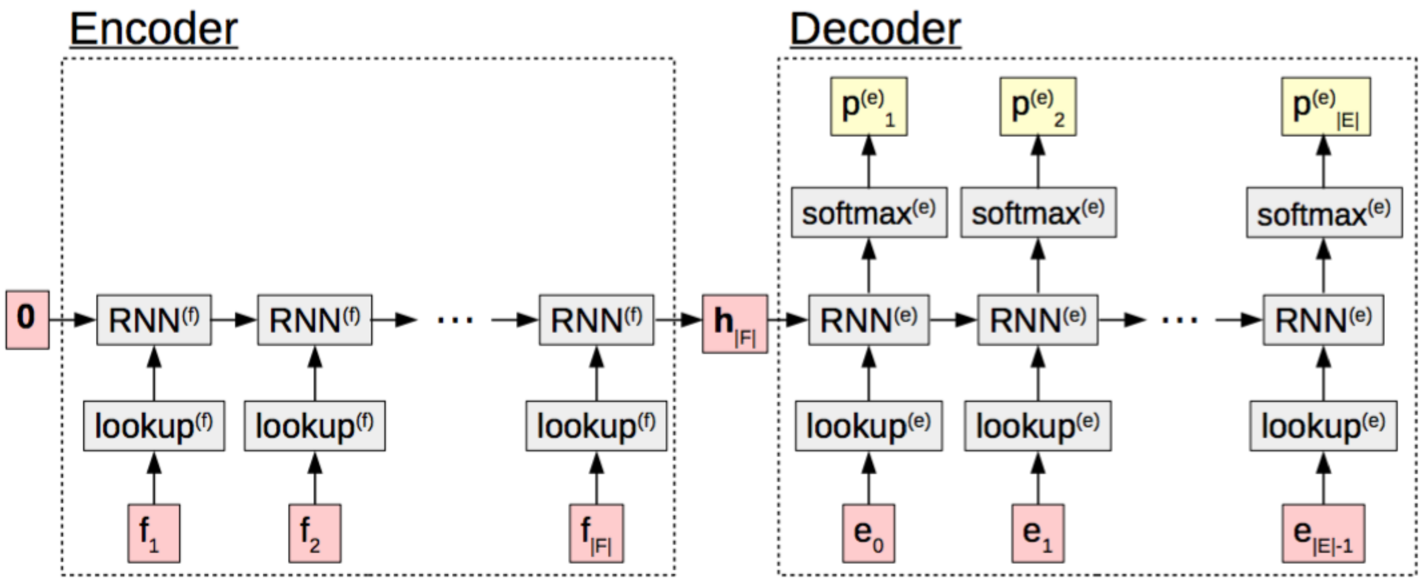


Neural machine translation: encoder-decoder



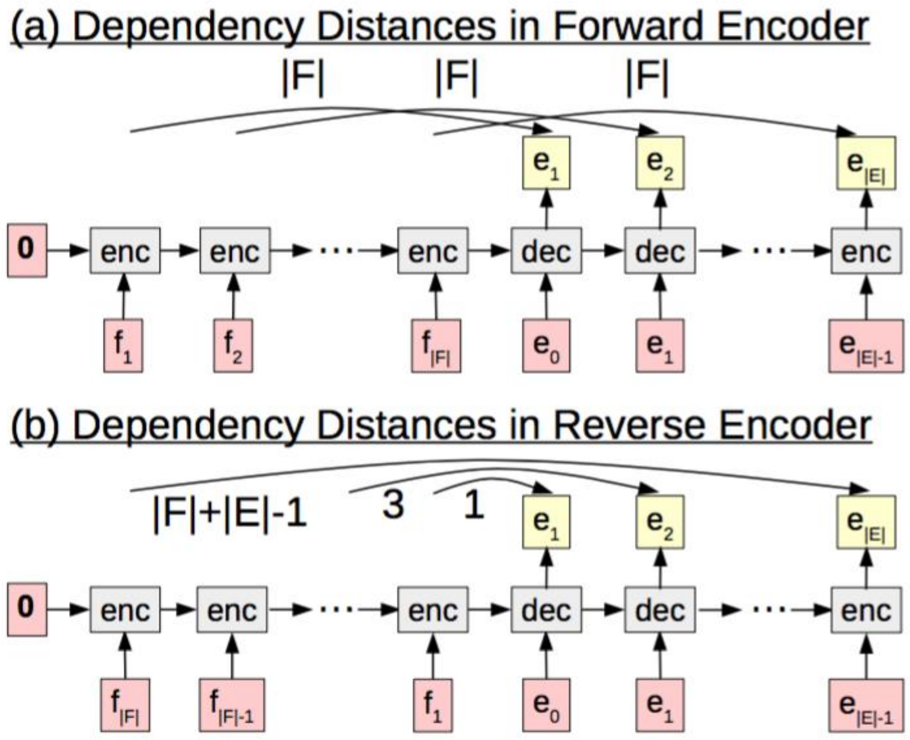
$$p\left(y_1, y_2, \dots, y_{T_y} \middle| x_1, x_2, \dots, x_{T_x}\right) = \prod_{i=1}^{T_y} p(y_i | y_1, \dots, y_{i-1}, \boldsymbol{x})$$

Neural machine translation: encoder-decoder



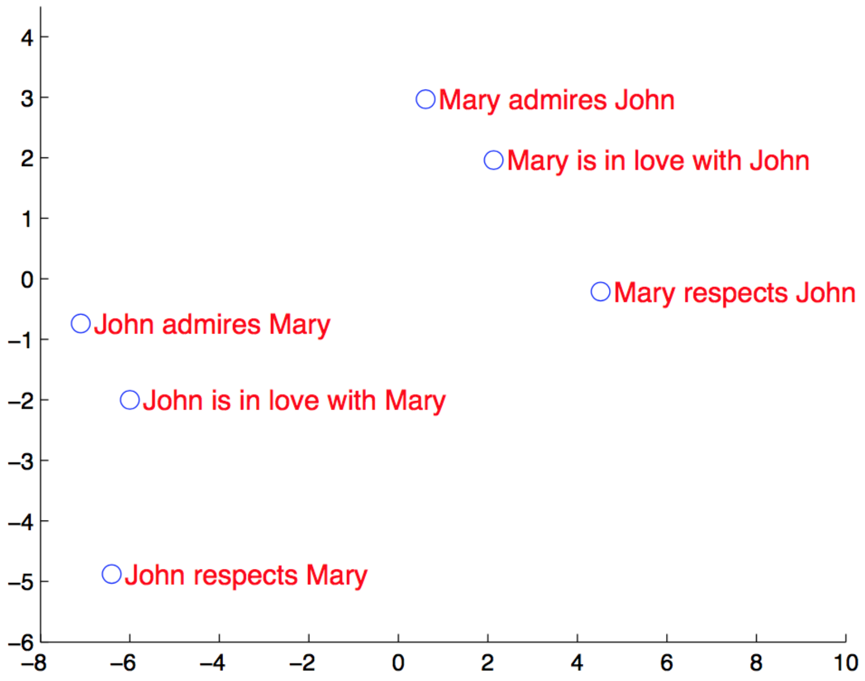
- Encode source sentence with deep LSTM
- Generate target words from decoder LSTM after <EOS>
- Bootstrap training by reversing the source sentence (+5 BLEU score – huge profit!)

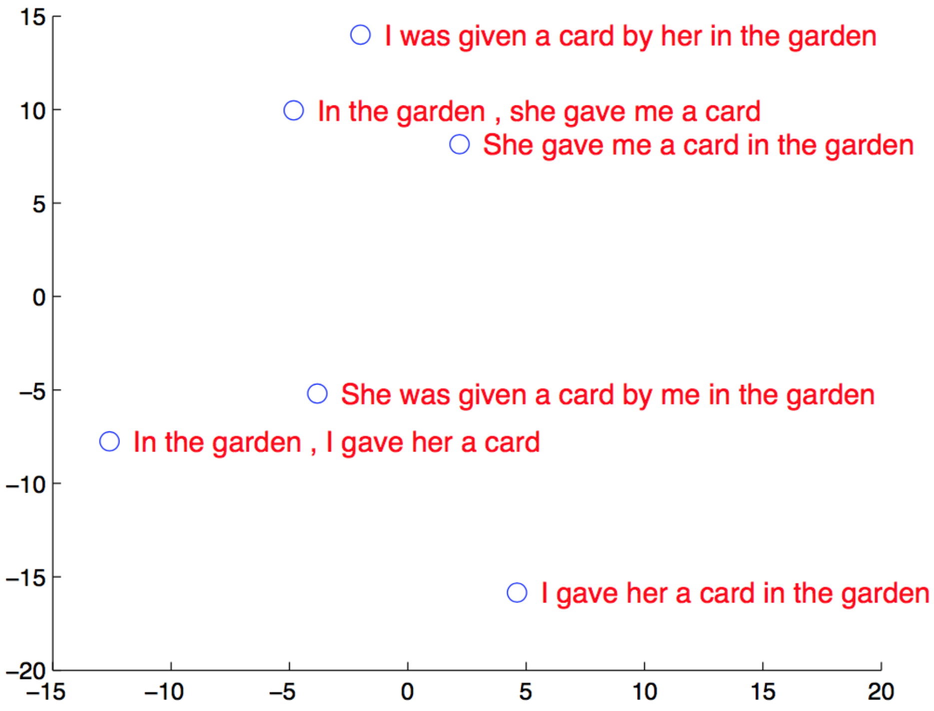
Why reversing the source sentence is good?



This is one of the ideas, but what do you think?

Neural machine translation: encoder-decoder



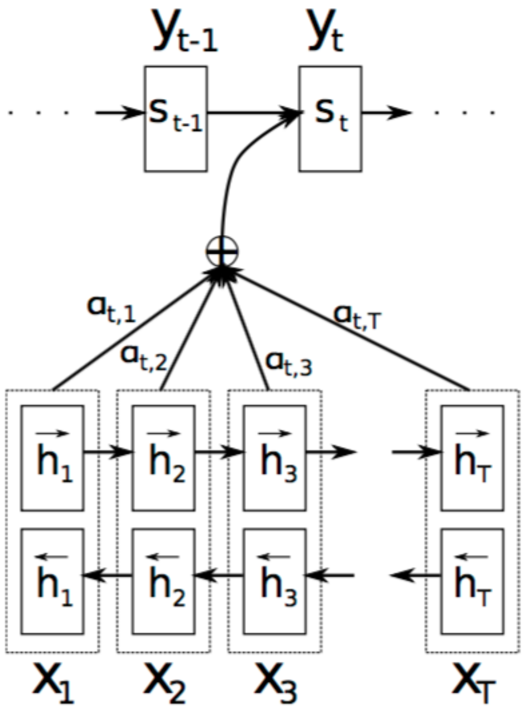


Attention: what is it?

According to Scholaropedia:

“Attention refers to the process by which organisms select a subset of available information upon which to focus for enhanced processing (often in a signal-to-noise-ratio sense) and integration.”

Neural machine translation: attentional encoder-decoder (Bahdanau)



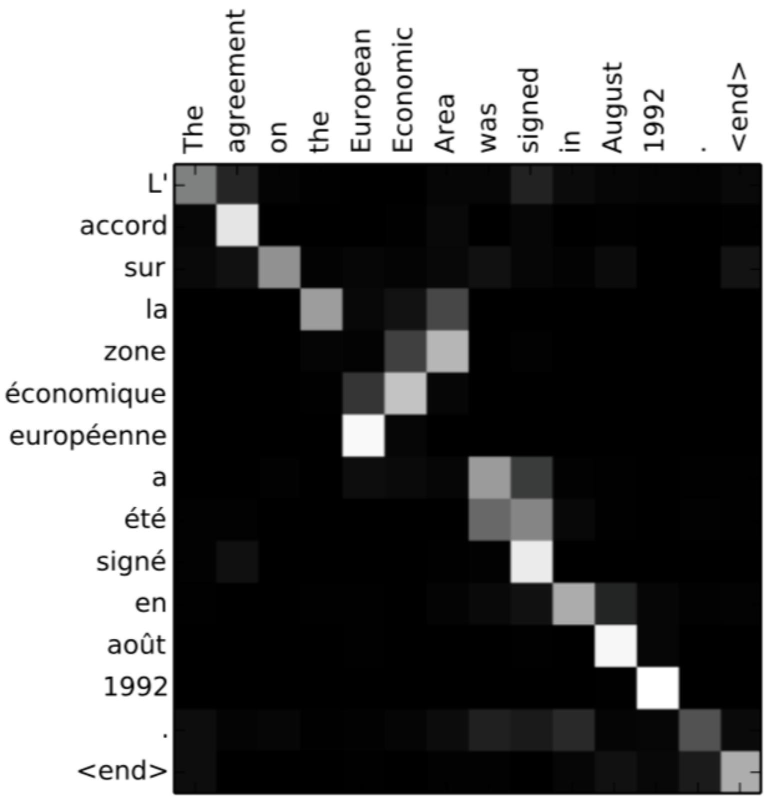
$$p(y_i | y_1, \dots, y_{i-1}, \boldsymbol{x}) = g(y_{i-1}, s_i, c_i)$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

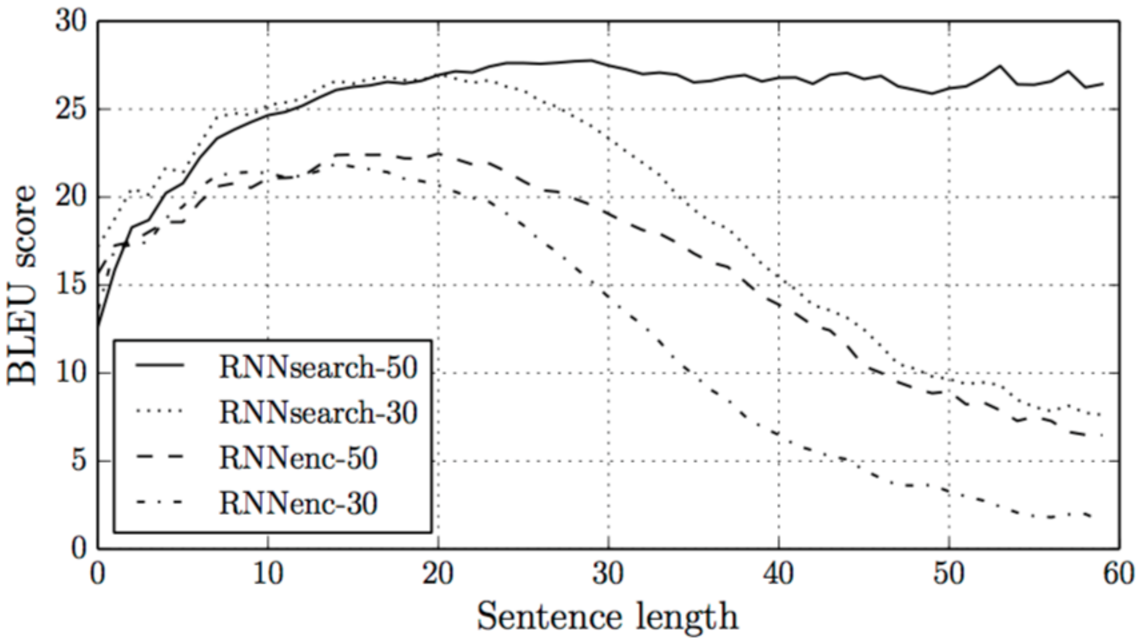
$$e_{ij} = a(s_{i-1}, h_j)$$

Neural machine translation: attentional encoder-decoder (Bahdanau)



Attention picks the words from the source sentence, which are useful for generating current target sentence. The model literally ”pays attention” to some source words

Neural machine translation: attentional encoder-decoder (Bahdanau)



Attention score: how to compute it?

- dot product
- bilinear function
- multi-layer perceptron
- any, literally, ANY function you can imagine

$$f_{att}(h_i, s_j) = h_i^{\top} s_j$$

$$f_{att}(h_i, s_j) = h_i^{\top} \mathbf{W}_a s_j$$

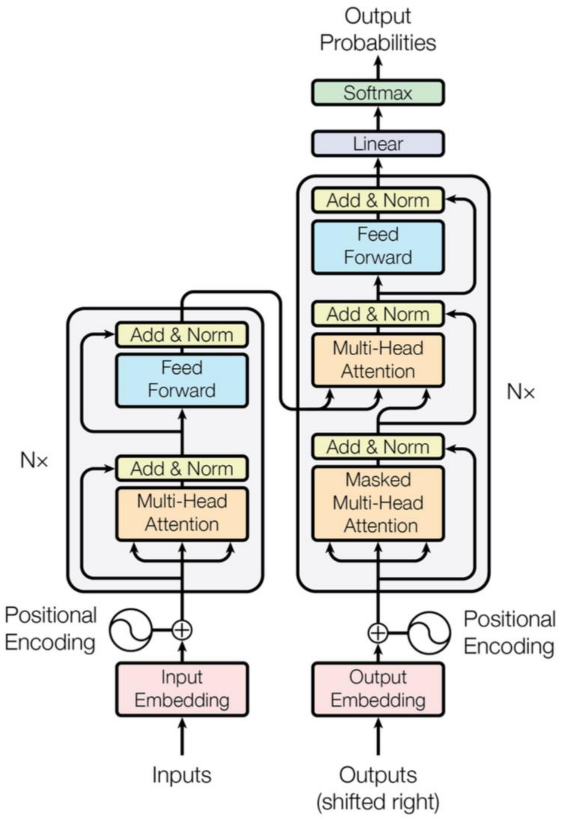
$$f_{att}(\mathbf{h}_i, \mathbf{s}_j) = \mathbf{v}_a{}' \tanh(\mathbf{W}_a[\mathbf{h}_i; \mathbf{s}_j])$$

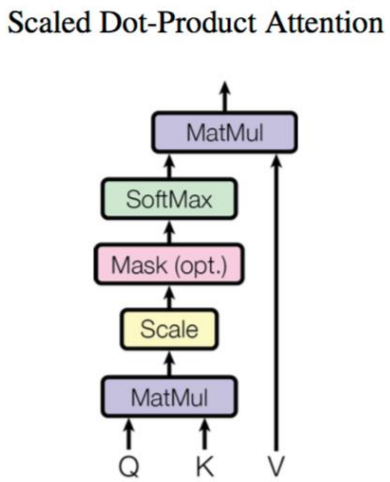
$$z(c, m, q) = \left[c, m, q, c \circ q, c \circ m, |c - q|, |c - m|, c^T W^{(b)} q, c^T W^{(b)} m\right]$$

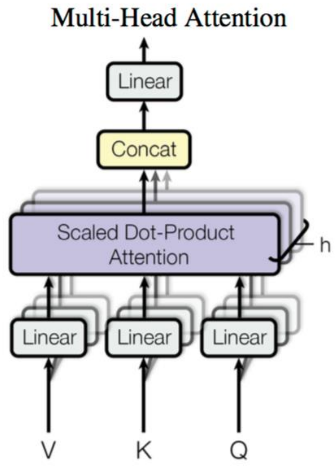
$$G(c, m, q) = \sigma\left(W^{(2)} \tanh\left(W^{(1)} z(c, m, q) + b^{(1)}\right) + b^{(2)}\right)$$

A bit crazy, huh?

Attention is all you need: Transformer

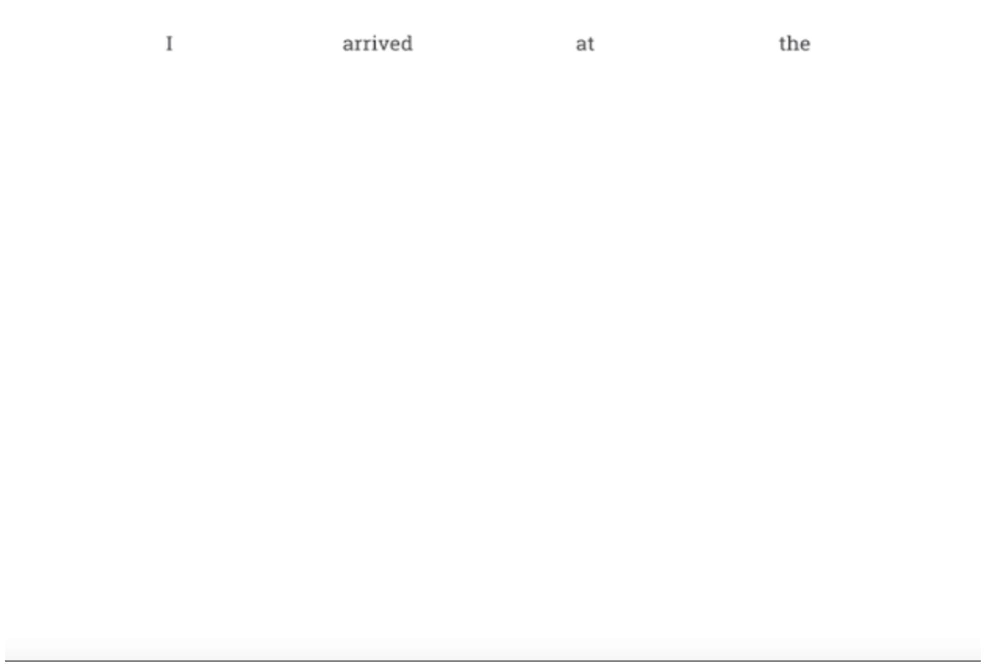






$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Attention is all you need: Transformer



I arrived at the bank after crossing the…

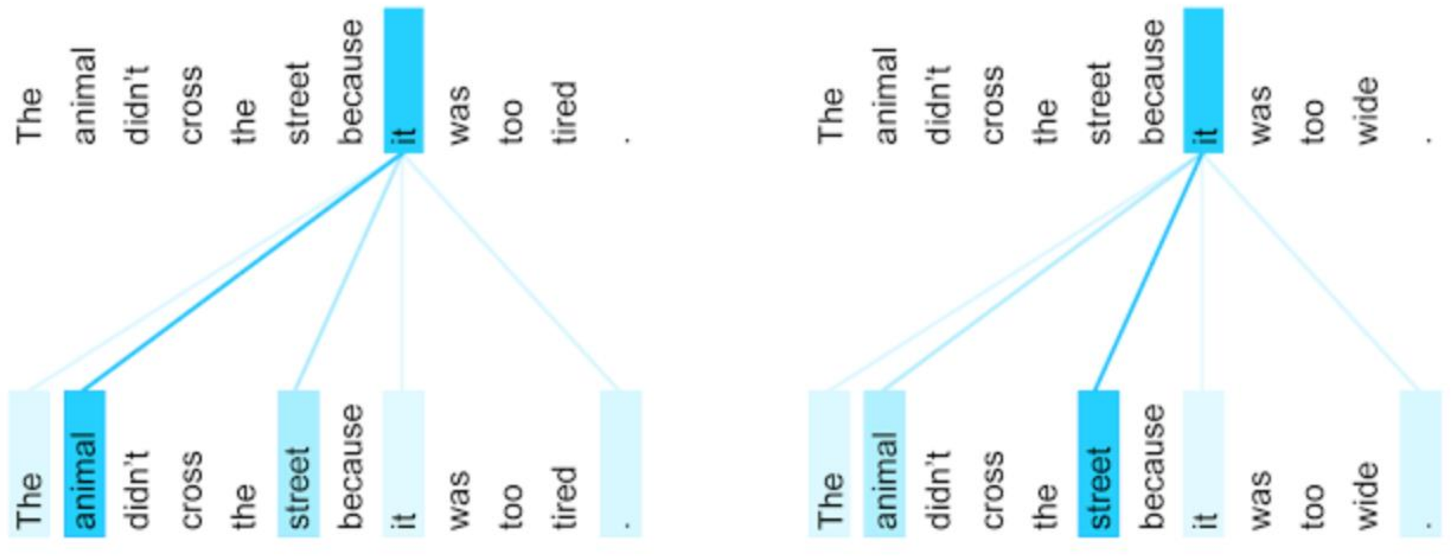
… river? …road?

Attention is all you need: Transformer

The animal didn't cross the street because it was too tired.
L'animal n'a pas traversé la rue parce qu'il était trop fatigué.

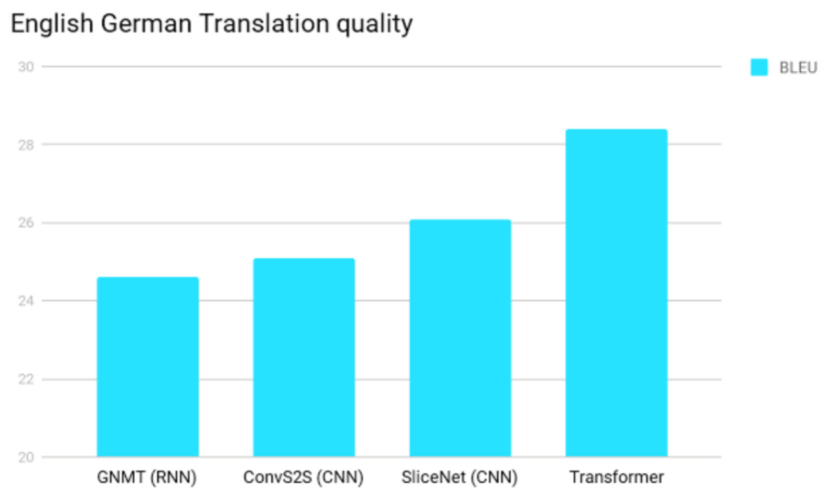
The animal didn't cross the street because it was too wide.
L'animal n'a pas traversé la rue parce qu'elle était trop large.

Attention is all you need: Transformer

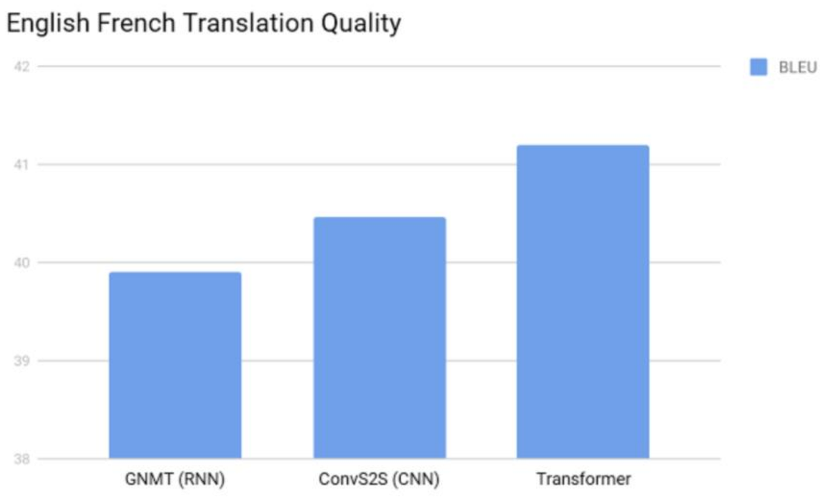


The encoder self-attention distribution for the word “it” from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads).

Attention is all you need: Transformer

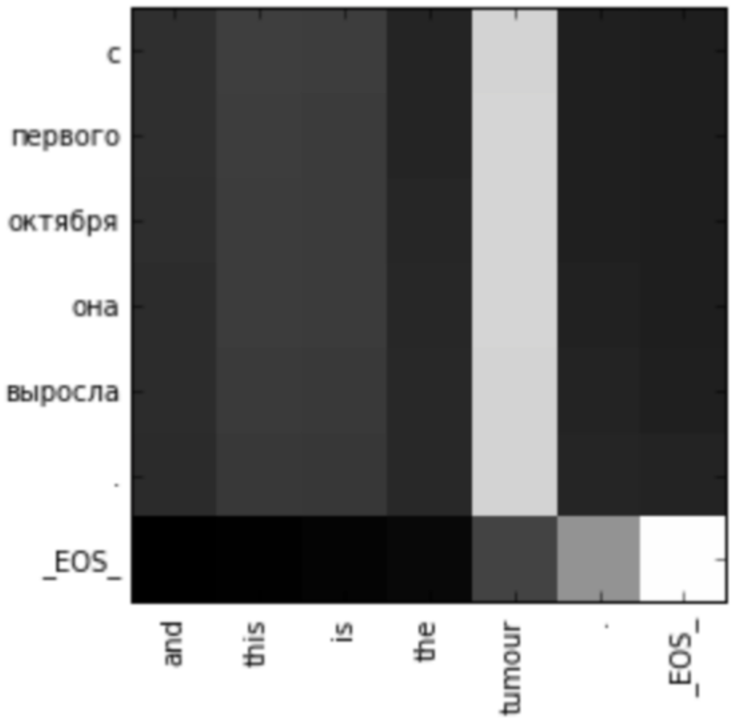


BLEU scores (higher is better) of single models on the standard WMT newstest2014 English to German translation benchmark.



BLEU scores (higher is better) of single models on the standard WMT newstest2014 English to French translation benchmark.

What if we attend to some context for translation?



SRC: since october 1st **it** has grown from there

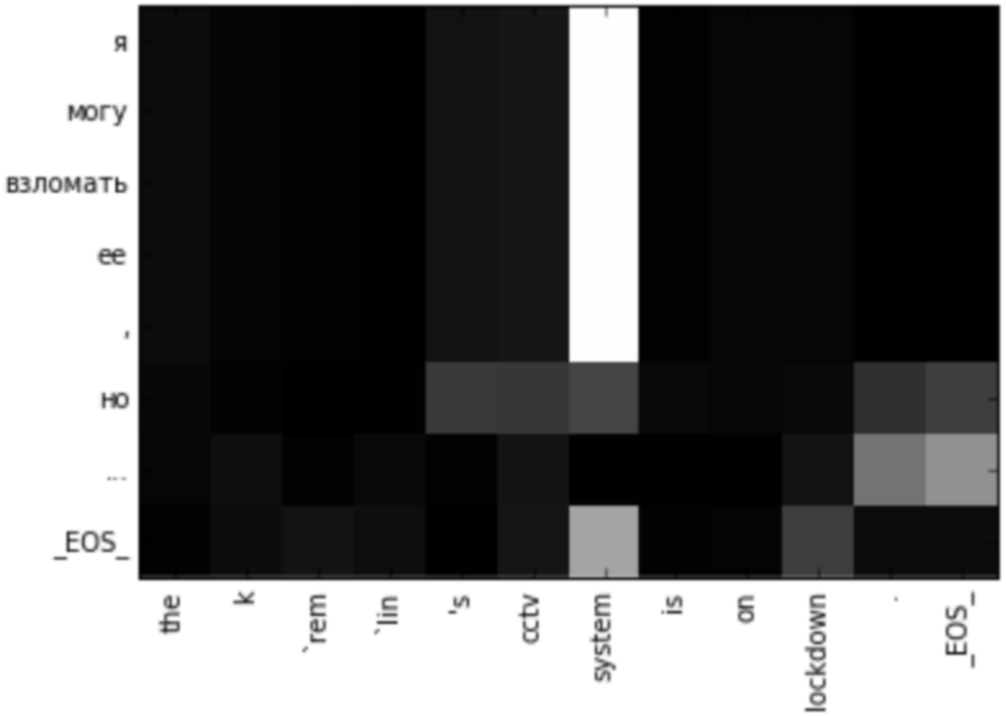
CTX: and this is the tumour .

DST: с первого октября она росла отсюда

NO CONTEXT: с первого октября он вырос .

WITH CONTEXT: с первого октября **она** выросла .

What if we attend to some context for translation?



SRC: i can hack **it** , but ...

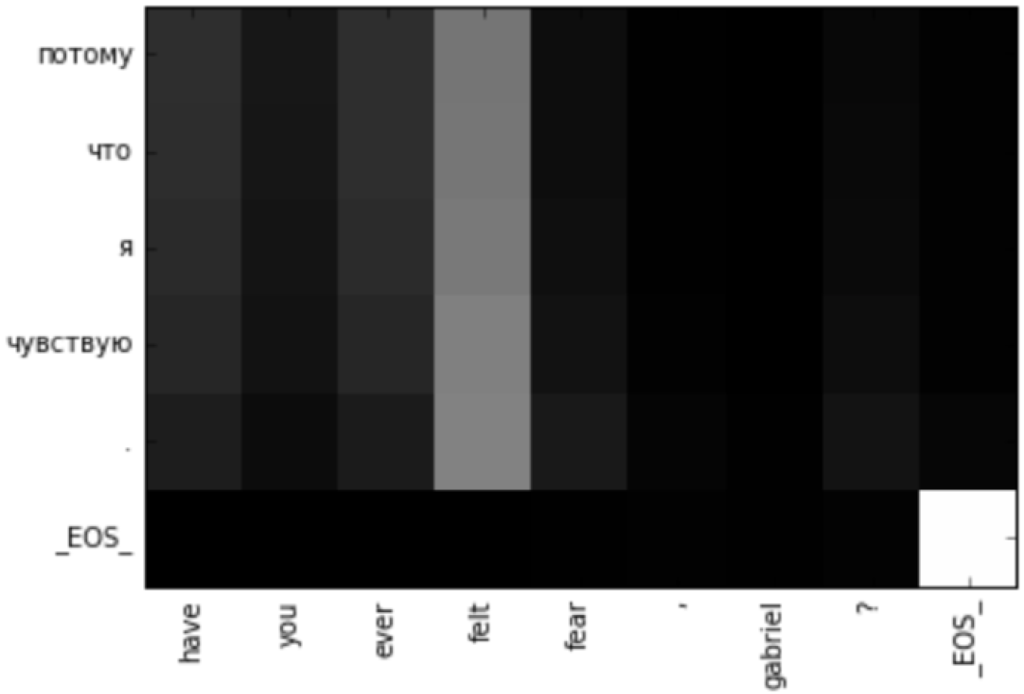
CTX: the k `rem `lin 's cctv system is on lockdown .

DST: я могу взломать ее , но ..

NO CONTEXT: я могу взломать его , но ...

WITH CONTEXT: я могу взломать **ее** , но ...

What if we attend to some context for translation?



SRC: because i **do** .

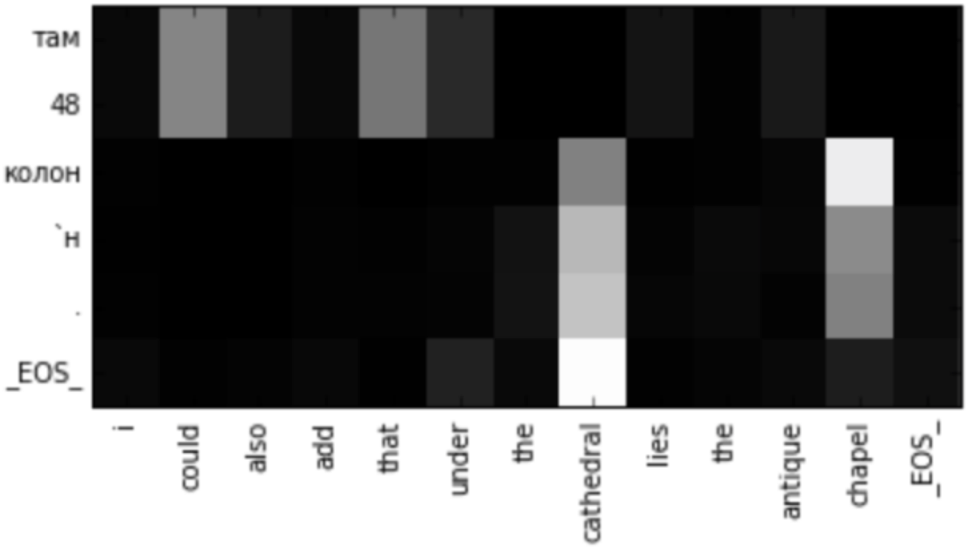
CTX: have you ever felt fear , gabriel ?

DST: а я чувствую .

NO CONTEXT: потому что я знаю .

WITH CONTEXT: потому что я **чувствую** .

What if we attend to some context for translation?



SRC: there are 48 **columns** .

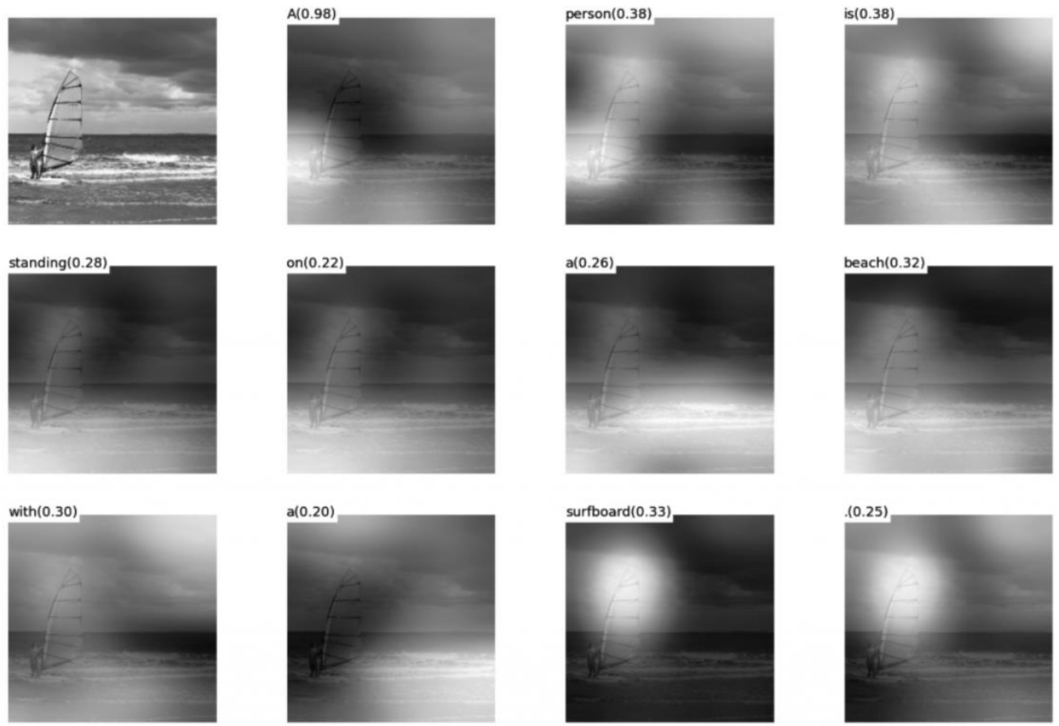
CTX: i could also add that under the cathedral lies the antique chapel

DST: там 48 колон `ок .

WITH CONTEXT: там 48 **колон `н** .

NO CONTEXT: там 48 колон `ок .

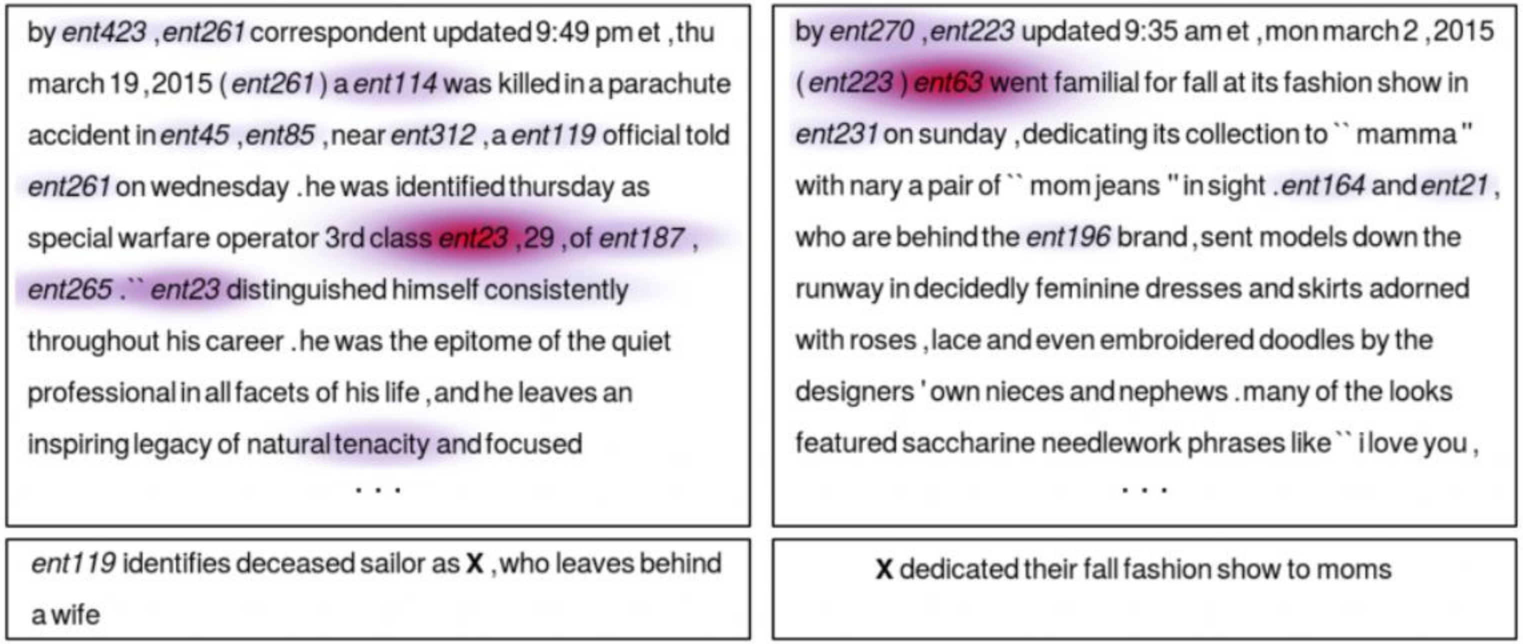
Attention: other use cases
Image caption generation



(b) A person is standing on a beach with a surfboard.

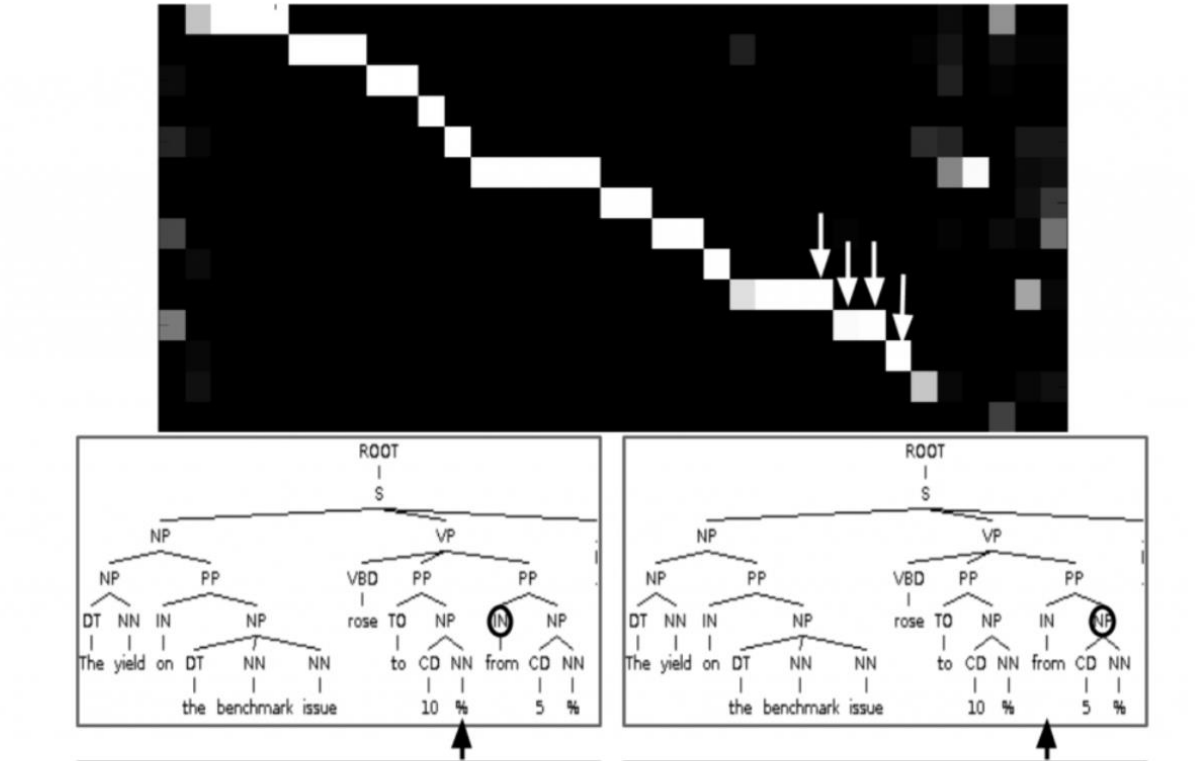
Use a Convolutional Neural Network to “encode” the image, and a Recurrent Neural Network with attention mechanisms to generate a description.

Attention: other use cases
Question answering task



Use a RNN to read a text, read a (synthetically generated) question, and then produce an answer.

Attention: other use cases
Generate sentence parse trees



Use a Recurrent Neural Network with attention mechanism to generate sentence parse trees

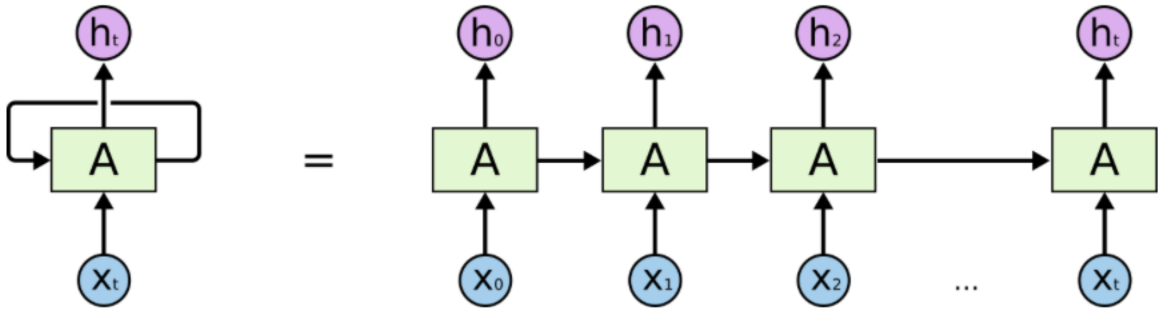
Seminar

NOVEMBER 9, 2017

Seminar plan

- Recurrent units in detail
- LSTM and GRU
- Dropout/ensembling
- Homework description

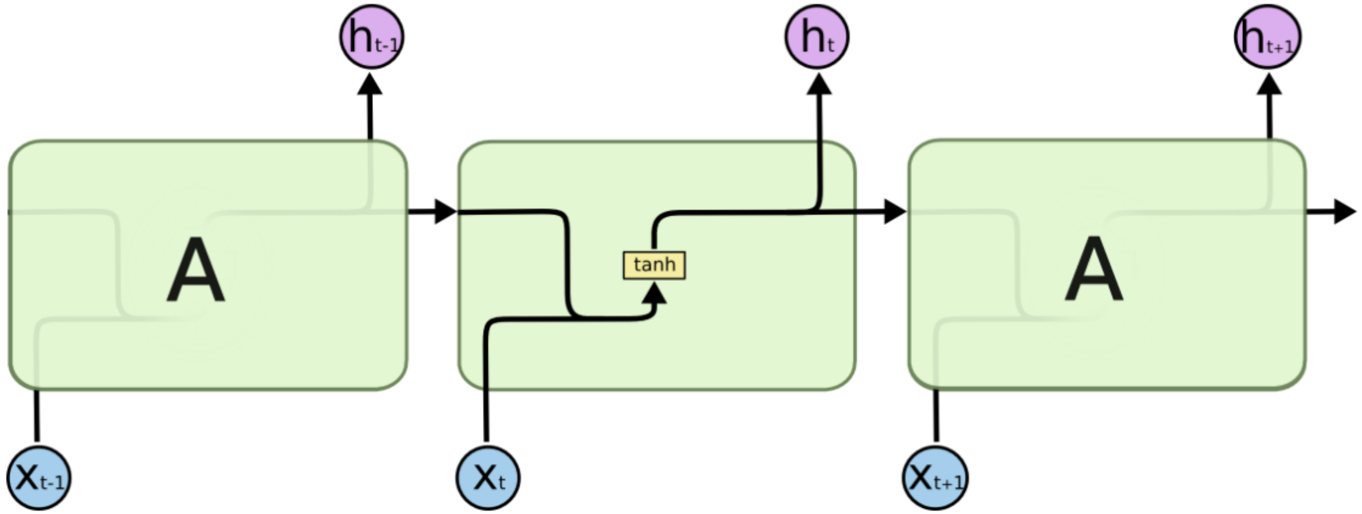
RNNs in detail



An unrolled recurrent neural network.

$$h_t = \tanh(h_{t-1}W_h + x_t W_x)$$

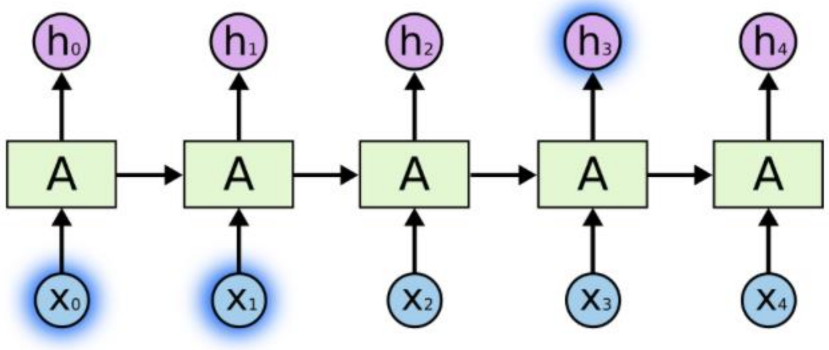
RNNs in detail



The repeating module in a standard RNN contains a single layer.

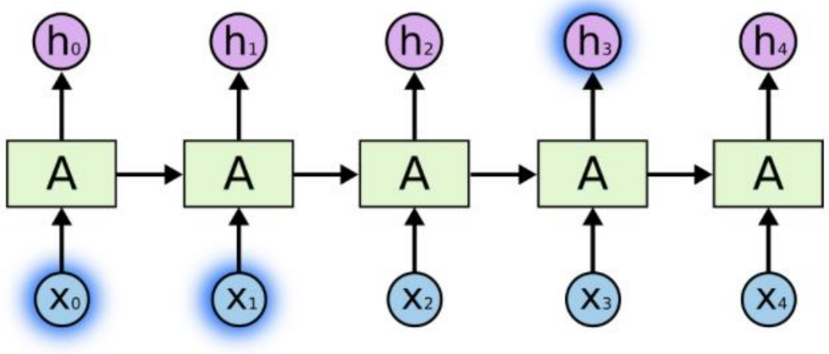
$$h_t = \tanh(h_{t-1} W_h + x_t W_x)$$

RNNs in detail



$$h_t = \tanh(h_{t-1} W_h + x_t W_x)$$

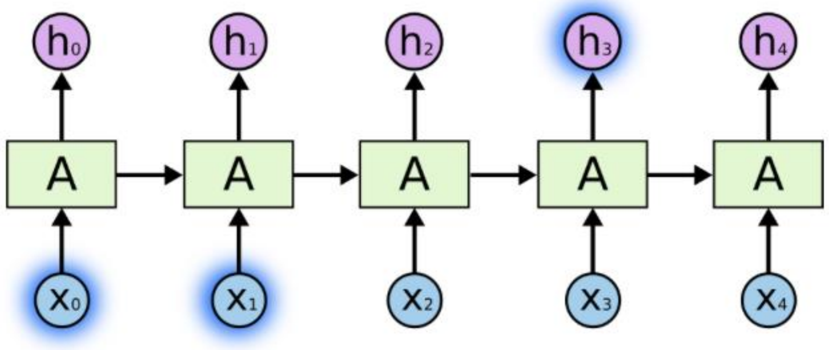
RNNs in detail



$$h_t = \tanh(h_{t-1} W_h + x_t W_x)$$

$$\frac{\partial h_t}{\partial W_h} = \sum_{k=0}^{t} \frac{\partial h_t}{\partial h_k} \cdot \frac{\partial h_k}{\partial W_h}$$

RNNs in detail

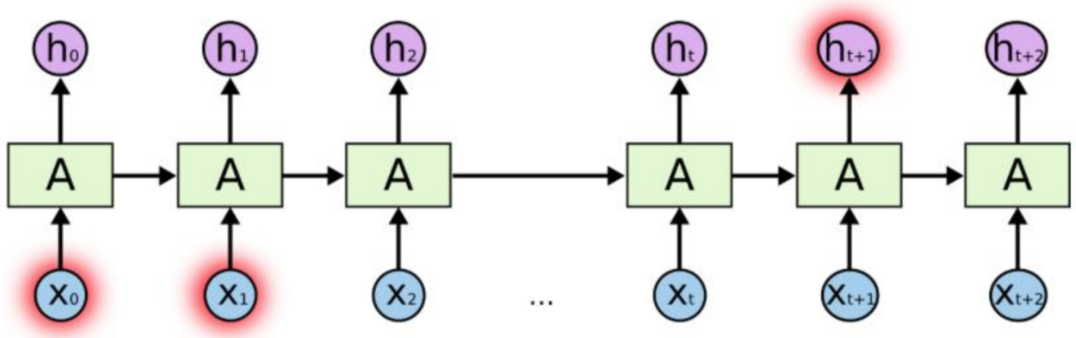


$$h_t = \tanh(h_{t-1} W_h + x_t W_x)$$

$$\frac{\partial h_t}{\partial W_h} = \sum_{k=0}^{t} \frac{\partial h_t}{\partial h_k} \cdot \frac{\partial h_k}{\partial W_h}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}}$$

RNNs in detail



Houston, we have a problem

$$h_t = \tanh(h_{t-1} W_h + x_t W_x)$$

$$\frac{\partial h_t}{\partial W_h} = \sum_{k=0}^{t} \frac{\partial h_t}{\partial h_k} \cdot \frac{\partial h_k}{\partial W_h}$$

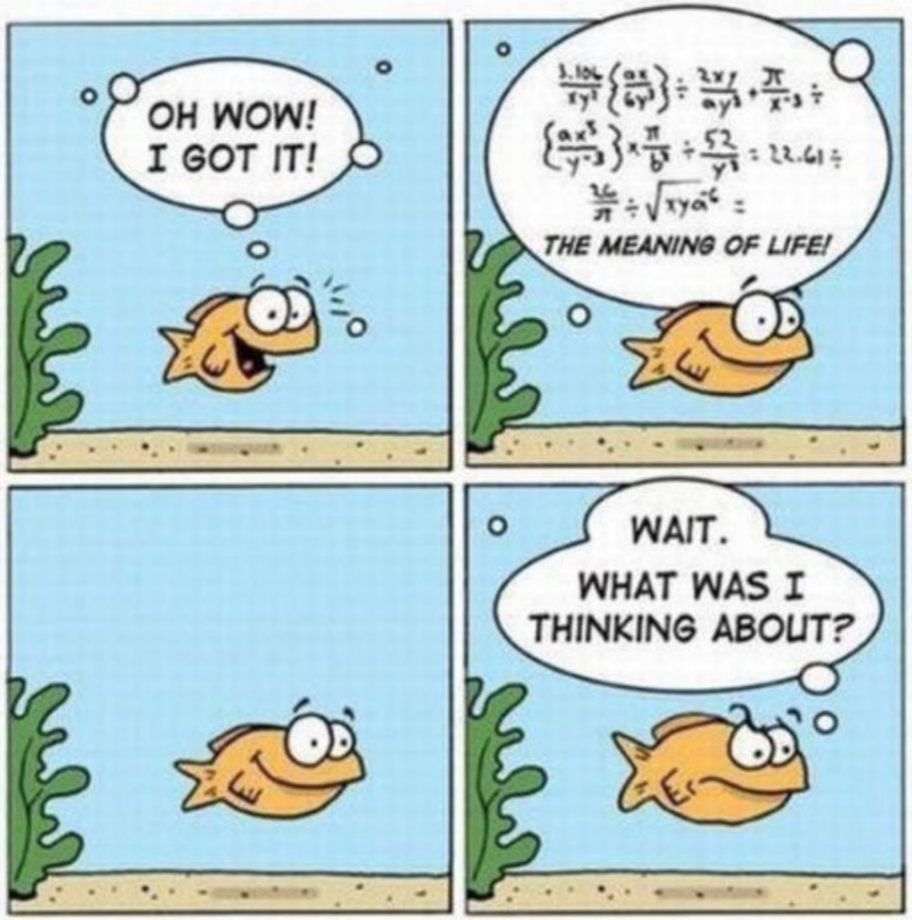
$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}}$$

Why should we care?

An example of long-distance dependencies in language:

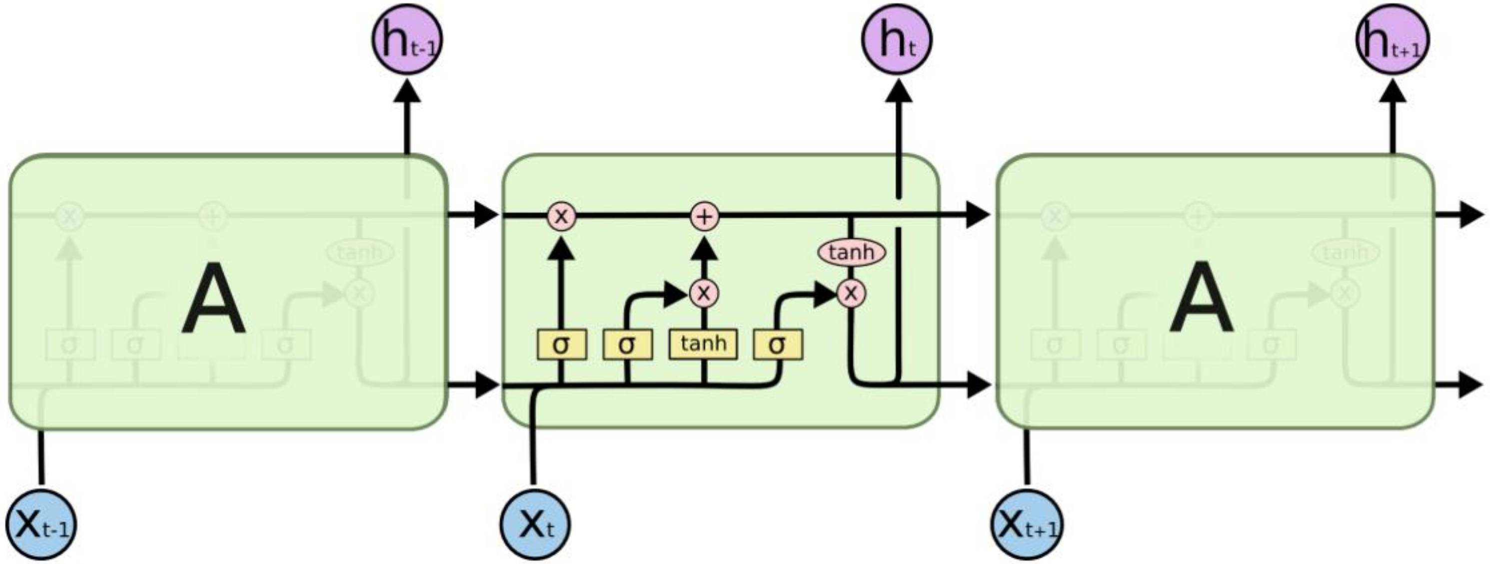
He doesn't have very much confidence in himself

She doesn't have very much confidence in herself



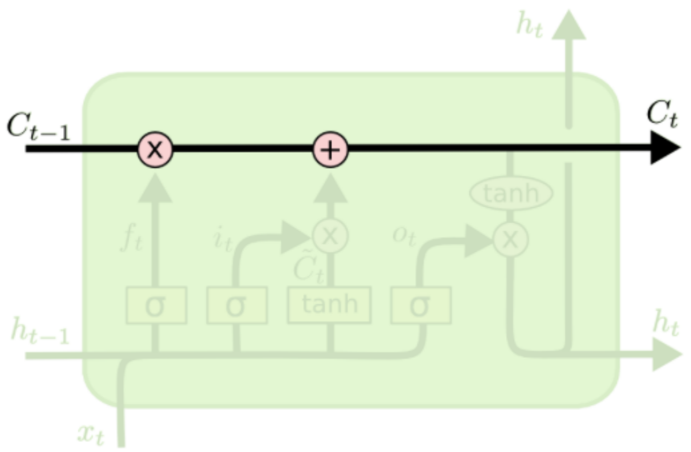
THE TRAGEDY OF A THREE SECOND MEMORY

LSTM: handling long-term dependences



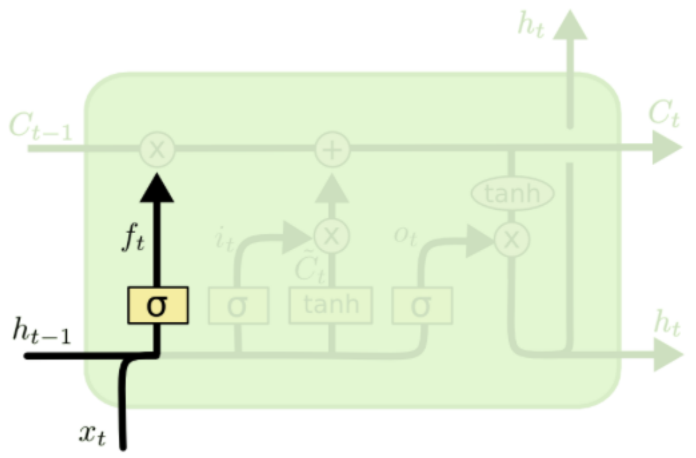
The repeating module in an LSTM contains four interacting layers.

LSTM: handling long-term dependences



The key to LSTMs is the cell state C_t.
The cell state is kind of like a conveyor belt.
It runs straight down the entire chain, with only some minor linear interactions.
It's very easy for information to just flow along it unchanged.

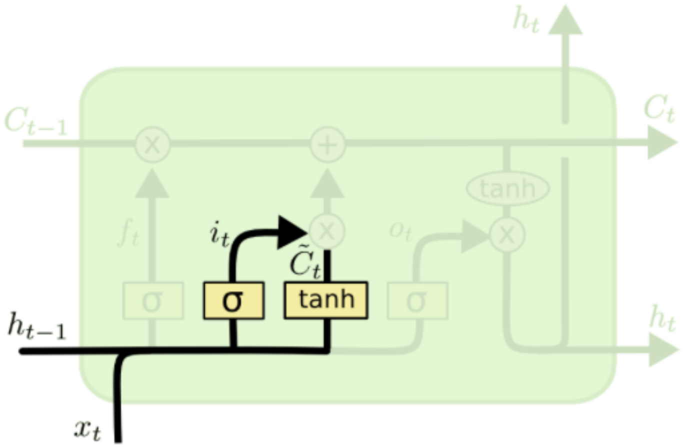
LSTM: handling long-term dependences



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

The first step in our LSTM is to decide what information we're going to throw away from the cell state.

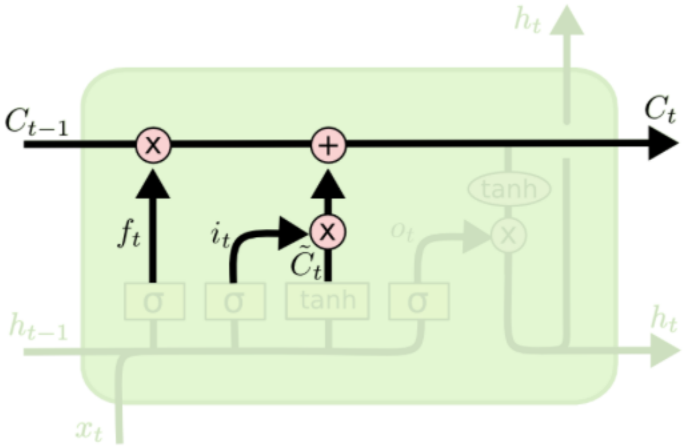
LSTM: handling long-term dependences



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

The next step is to decide what new information we're going to store in the cell state.

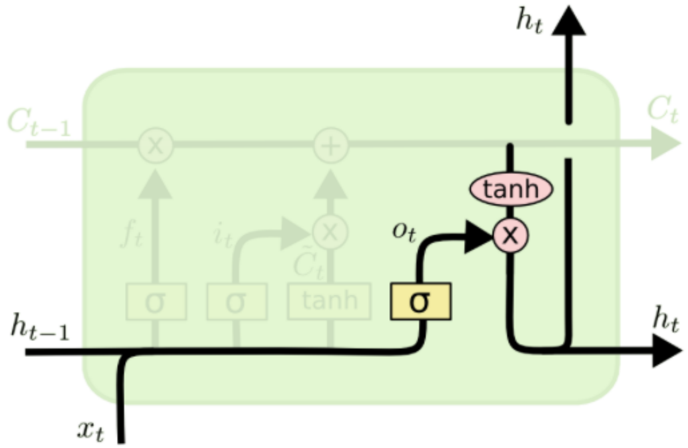
LSTM: handling long-term dependences



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

It's now time to update the old cell state, C_{t-1}, into the new cell state C_t. The previous steps already decided what to do, we just need to actually do it.

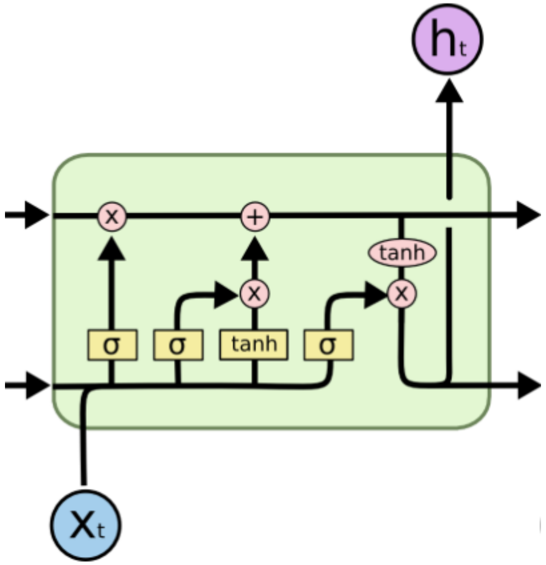
LSTM: handling long-term dependences



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

Finally, we need to decide what we're going to output. This output will be based on our cell state, but will be a filtered version.

LSTM: handling long-term dependences



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

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

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

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

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

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

or

Original version:

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

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

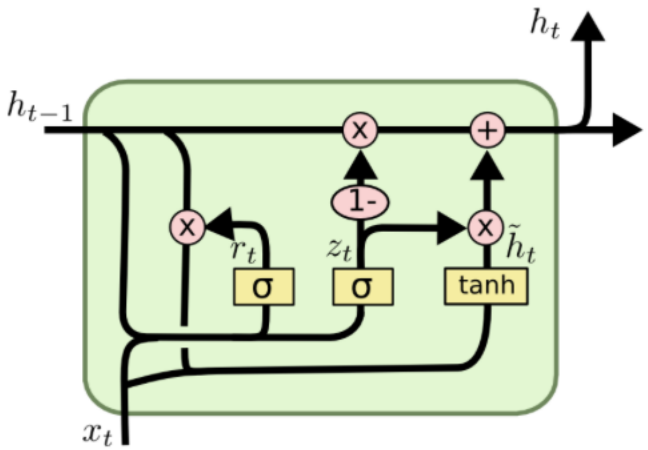
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

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

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

Why the problem of vanishing/exploding gradient here is not such urgent?

GRU: Gated Recurrent Unit



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

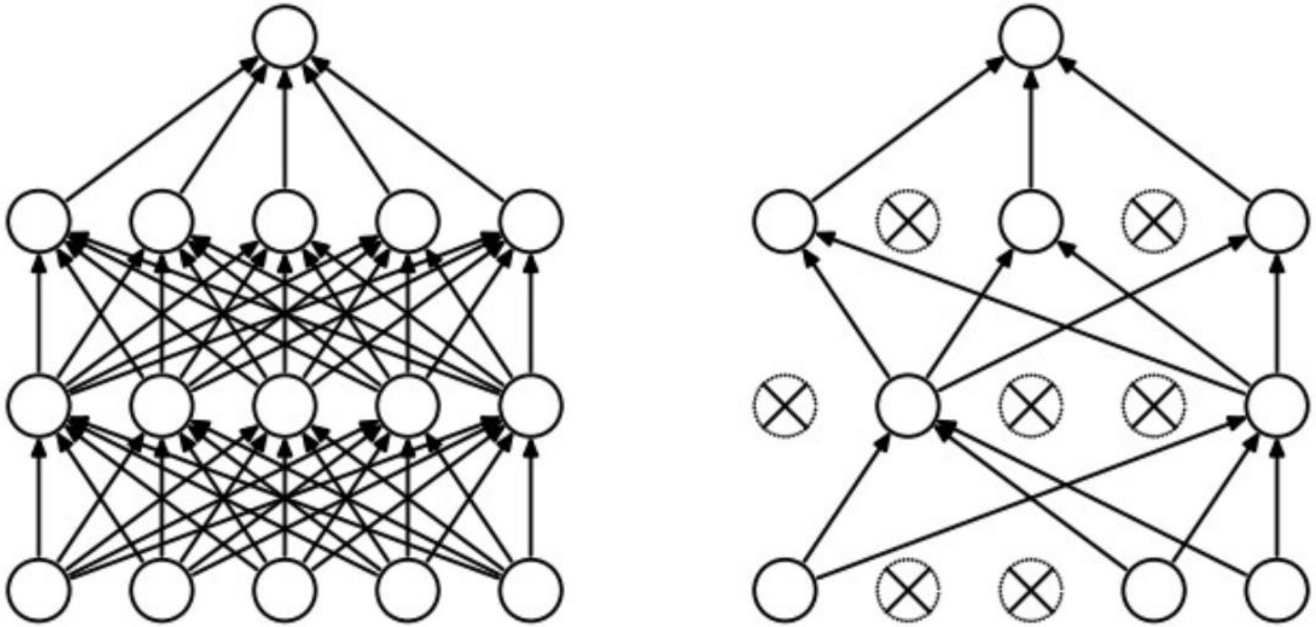
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Ensembles

- independently train several NNs
- (each time initialization is random)
- average these networks
- Profit!

Dropout



(a) Standard Neural Net (b) After applying dropout.

- h_train = m ⊙ h, mj ~ Bernoulli(p).
- h_test = ph.

References

Dzmitry Bahdanau et al., Neural Machine Translation by Jointly Learning to Align and Translate, In: arXiv preprint arxiv: 1409.0473

Eunsol Choi et al., Coarse-to-Fine Question Answering for Long Documents, In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 209–220

Yiming Cui et al., Attention-over-Attention Neural Networks for Reading Comprehension, In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 593–602

Bhuwan Dhingra & Hanxiao Liu, Gated-Attention Readers for Text Comprehension, In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1832–1846

References

Yanchao Hao et al., An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge, In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 221–231

Sebastien Jean et al., Does Neural Machine Translation Benefit from Larger Context?, In: arXiv preprint arxiv:1704.05135

Rudolf Kadlec et al., “Text Understanding with the Attention Sum Reader Network”, In: arXiv preprint arxiv:1603.01547

Minh-Thang Luong et al., Effective Approaches to Attention-based Neural Machine Translation, In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421

References

Adam Trischler et al., Natural Language Comprehension with the EpiReader, In: arXiv preprint arxiv: 1606.02270 (EMNLP, 2016)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need, In: arXiv preprint arxiv:1706.03762

Longyue Wang et al., Exploiting Cross-Sentence Context for Neural Machine Translation, In: arXiv preprint arxiv:1704.04347

Zichao Yang et al., Hierarchical Attention Networks for Document Classification, In: Proceedings of NAACL-HLT 2016, pp. 1480–1489

References

Pascanu et al., On the difficulty of training recurrent neural networks. In: Proceedings of the 30th International Conference on Machine Learning

Sutskever et al., Sequence to Sequence Learning with Neural Networks. In: arXiv preprint arxiv:1409.3215

Xu et al., Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In: arXiv preprint arxiv:1502.03044

Hermann et al., Teaching Machines to Read and Comprehend. In: arXiv preprint arxiv:1506.03340

References

Vinyals et al., Grammar as a Foreign Language. In: arXiv preprint arxiv:1412.7449

Pham et al., Dropout improves Recurrent Neural Networks for Handwriting Recognition. In: arXiv preprint arxiv:1312.4569

Gal & Grahramani, A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In: arXiv preprint arxiv:1512.05287

Cho et al., Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In: arXiv preprint arxiv:1406.1078

http://colah.github.io/posts/2015-08-Understanding-LSTMs/