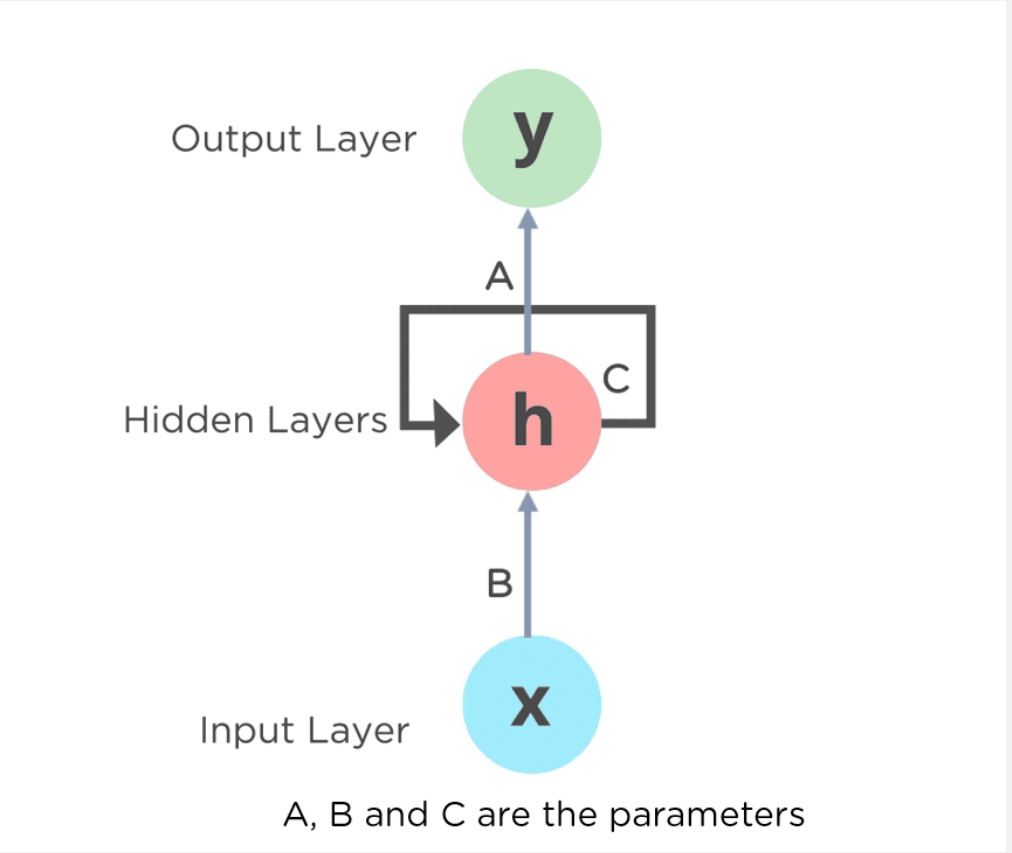
SIECI REKURENCYJNE

Paulina Tomaszewska

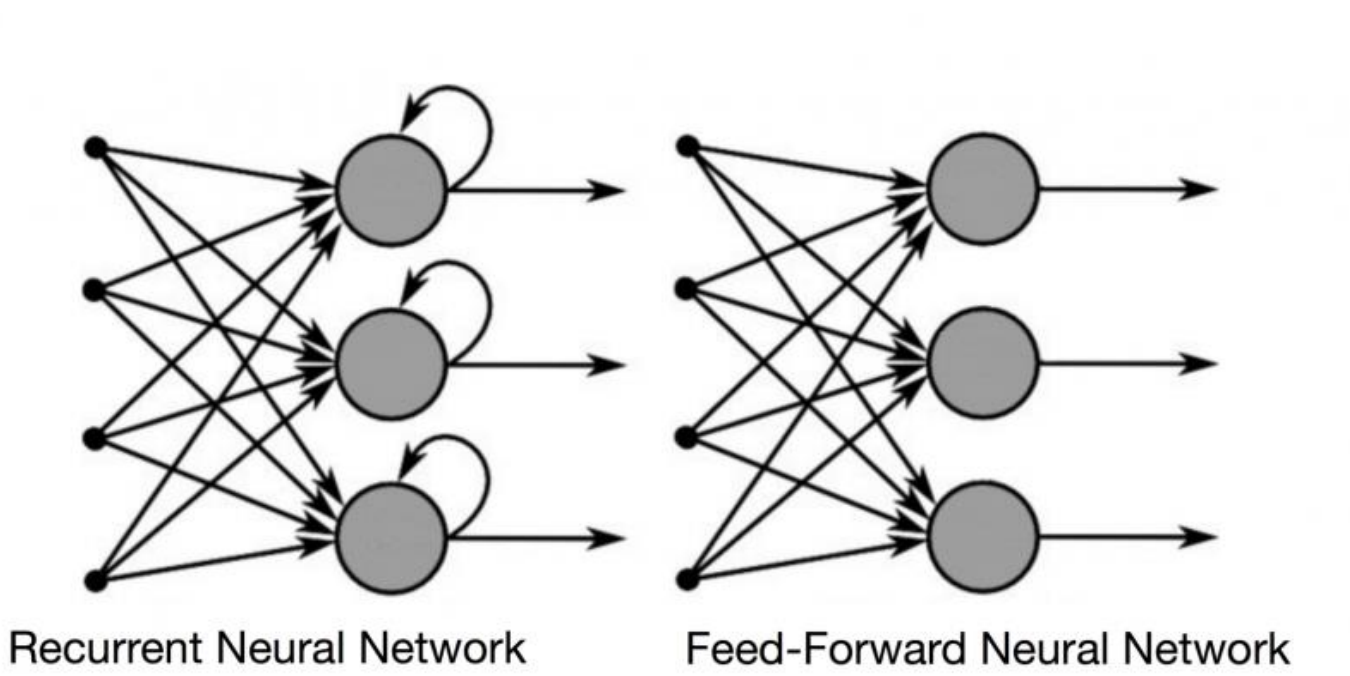
RNN

- For sequential data:
  - time series
  - text
  - audio
  - video
- Applications:
  - Siri, Cortana,
  - Google Translate
- Internal memory – short term
- Backpropagation through time



https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

RNN VS. MLP



MLP:

- the information only moves in one direction
- no memory of the input
- no notion of order in time

RNN:

- the information cycles through a loop
- considers the current input and historical

https://builtin.com/data-science/recurrent-neural-networks-and-lstm

EXAMPLE FOR BETTER UNDERSTANDING

- input to the network: „neuron”
- MLP: processes the word character by character. By the time it reaches the character "r," it has already forgotten about "n," "e" and "u," which makes it almost impossible for this type of neural network to predict which character would come next.
- RNN: remember previous characters because of its internal memory. It produces output, copies that output and loops it back into the network.

https://builtin.com/data-science/recurrent-neural-networks-and-lstm

INPUTS, OUTPUTS

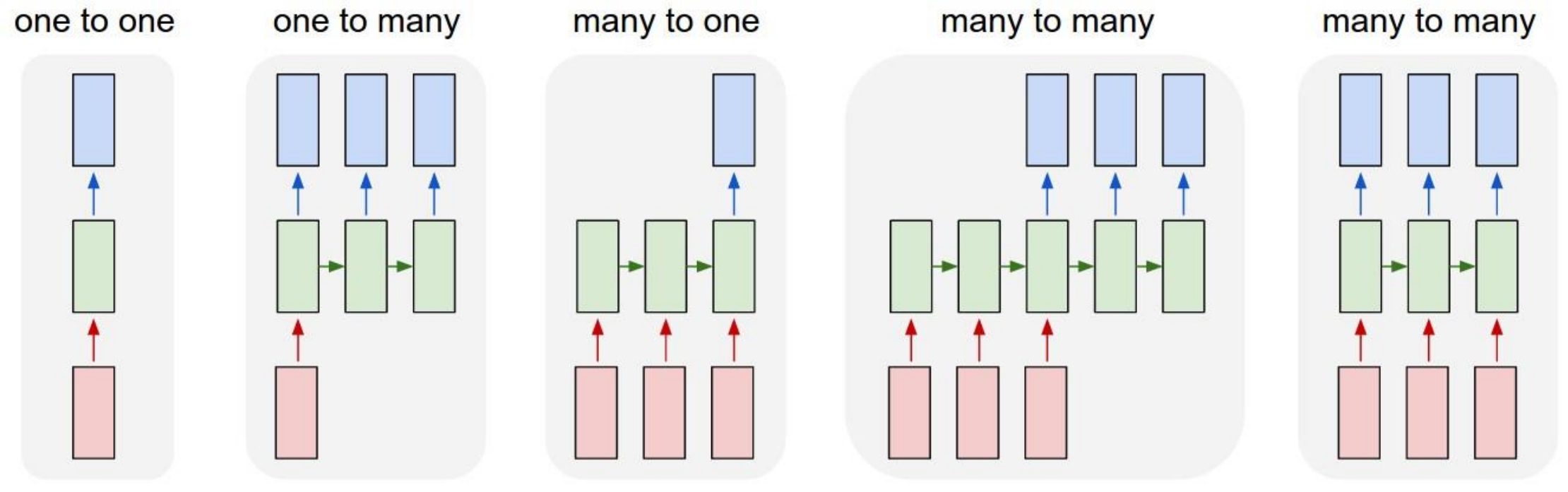


Image classification

Image captioning

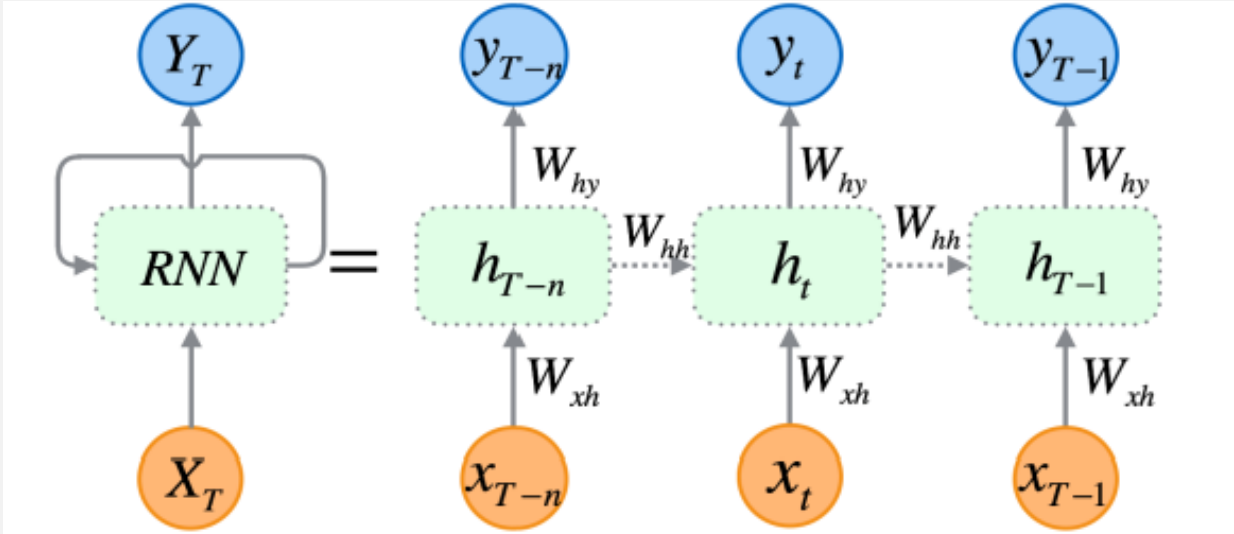
Sentiment analysis

Machine translation

Classification of video frames

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

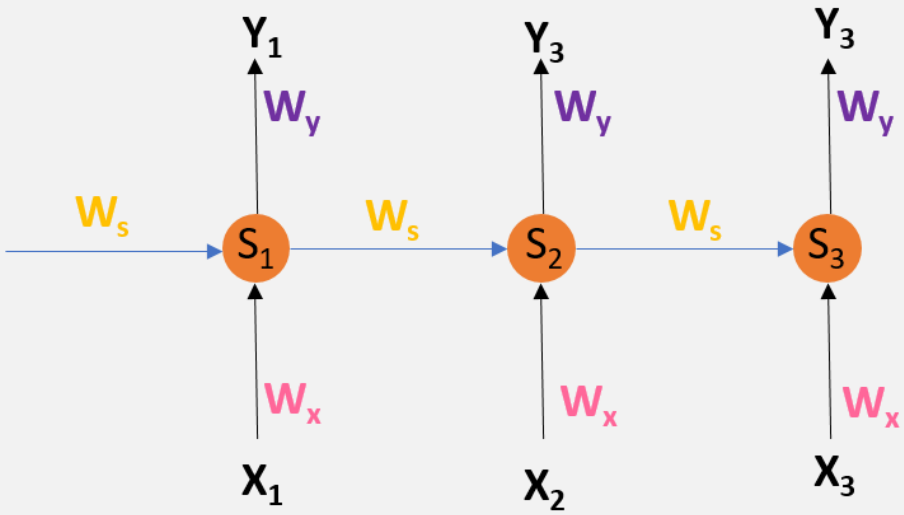
UNROLLING



Standard RNN architecture and an unfolded structure with T time

https://www.i2tutorials.com/what-is-the-difference-between-bidirectional-rnn-and-rnn/

BACKPROPAGATION THROUGH TIME



1. Compute loss $E_3 = (d_3 - Y_3)^2$

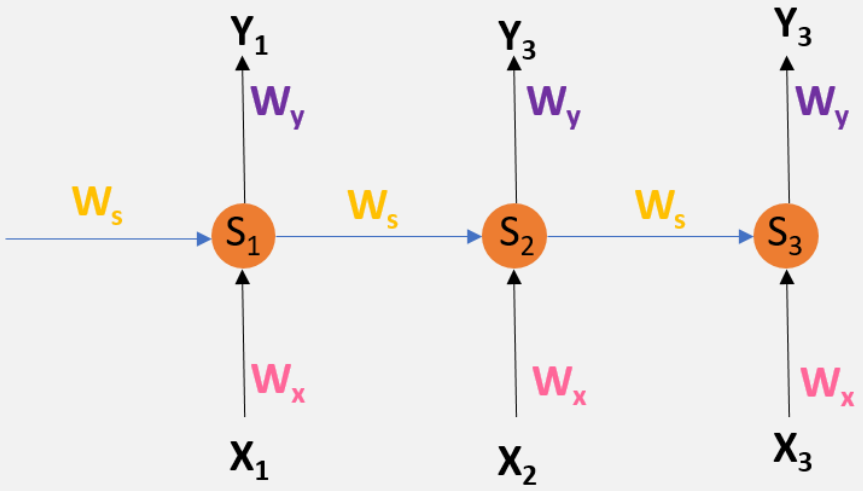
2. Adjusting Wy

$$S_t = g_1(W_x x_t + W_s S_{t-1})$$
$$Y_t = g_2(W_Y S_t)$$

$$\frac{\partial E_3}{\partial W_y} = \frac{\partial E_3}{\partial Y_3} \cdot \frac{\partial Y_3}{\partial W_Y}$$

https://www.geeksforgeeks.org/ml-back-propagation-through-time/

BACKPROPAGATION THROUGH TIME



$$S_t = g_1(W_x x_t + W_s S_{t-1})$$
$$Y_t = g_2(W_Y S_t)$$

3. Adjusting Ws

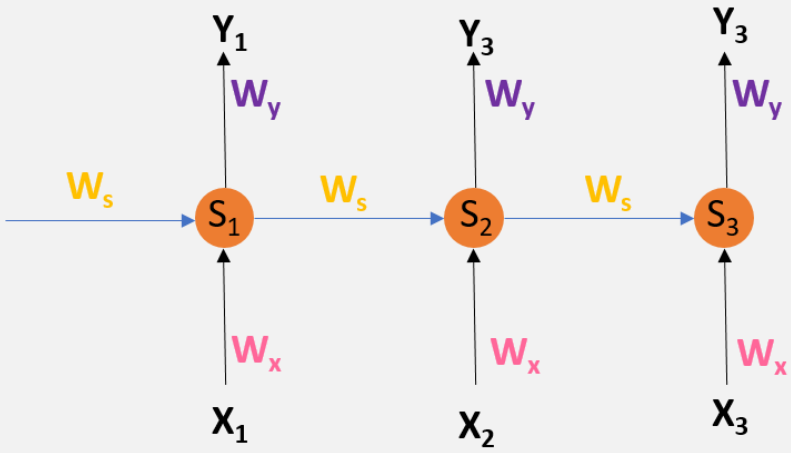
$$\frac{\partial E_3}{\partial W_S} = (\frac{\partial E_3}{\partial Y_3} \cdot \frac{\partial Y_3}{\partial S_3} \cdot \frac{\partial S_3}{\partial W_S}) +$$

$$(\frac{\partial E_3}{\partial Y_3} \cdot \frac{\partial Y_3}{\partial S_3} \cdot \frac{\partial S_3}{\partial S_2} \cdot \frac{\partial S_2}{\partial W_S}) +$$

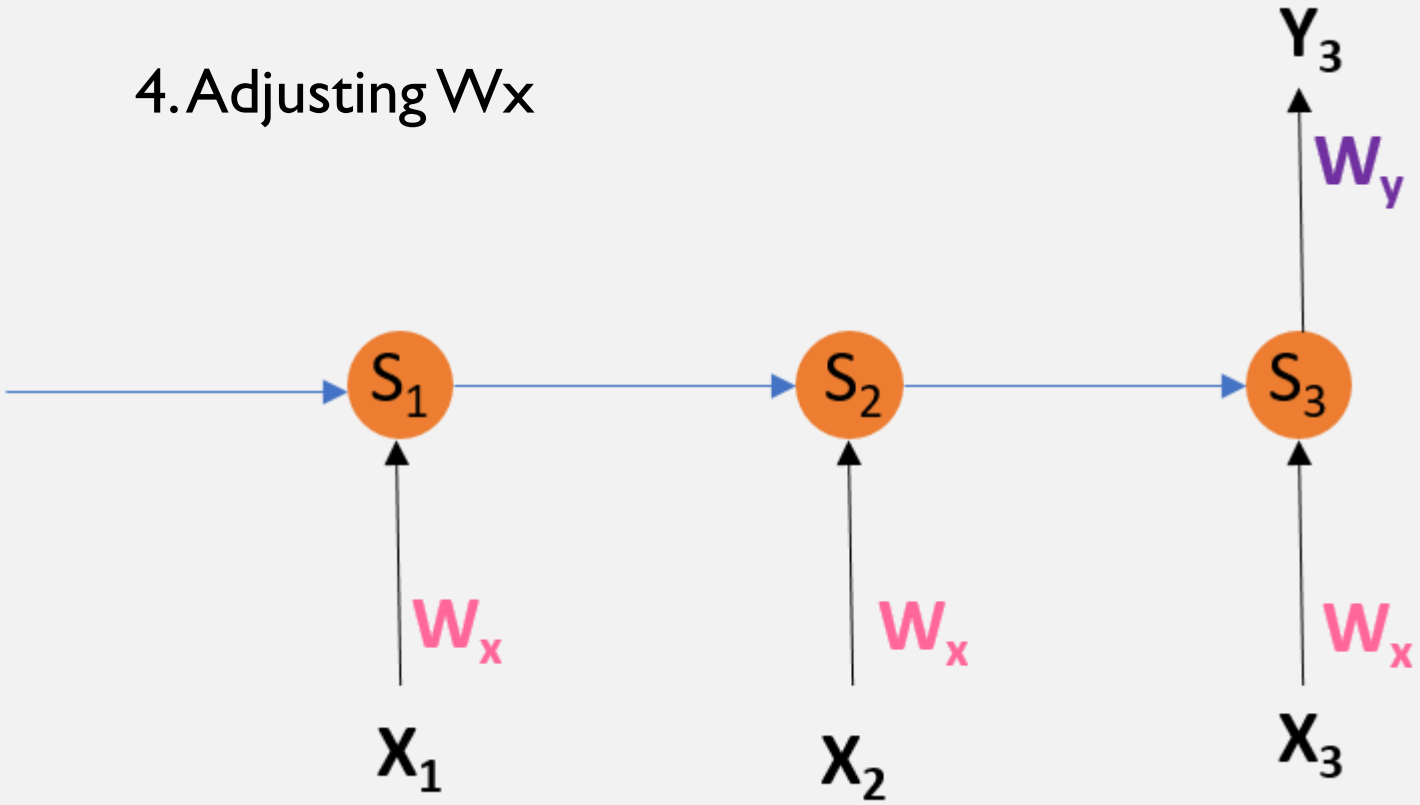
$$(\frac{\partial E_3}{\partial Y_3} \cdot \frac{\partial Y_3}{\partial S_3} \cdot \frac{\partial S_3}{\partial S_2} \cdot \frac{\partial S_2}{\partial S_1} \cdot \frac{\partial S_1}{\partial W_S})$$

https://www.geeksforgeeks.org/ml-back-propagation-through-time/

BACKPROPAGATION THROUGH TIME



4. Adjusting Wx



$$S_t = g_1(W_x x_t + W_s S_{t-1})$$
$$Y_t = g_2(W_Y S_t)$$

$$\frac{\partial E_3}{\partial W_X} = (\frac{\partial E_3}{\partial Y_3}.\frac{\partial Y_3}{\partial S_3}.\frac{\partial S_3}{\partial W_X})+$$

$$(\frac{\partial E_3}{\partial Y_3}.\frac{\partial Y_3}{\partial S_3}.\frac{\partial S_3}{\partial S_2}.\frac{\partial S_2}{\partial W_X})+$$

$$(\frac{\partial E_3}{\partial Y_3}.\frac{\partial Y_3}{\partial S_3}.\frac{\partial S_3}{\partial S_2}.\frac{\partial S_2}{\partial S_1}.\frac{\partial S_1}{\partial W_X})$$

https://www.geeksforgeeks.org/ml-back-propagation-through-time/

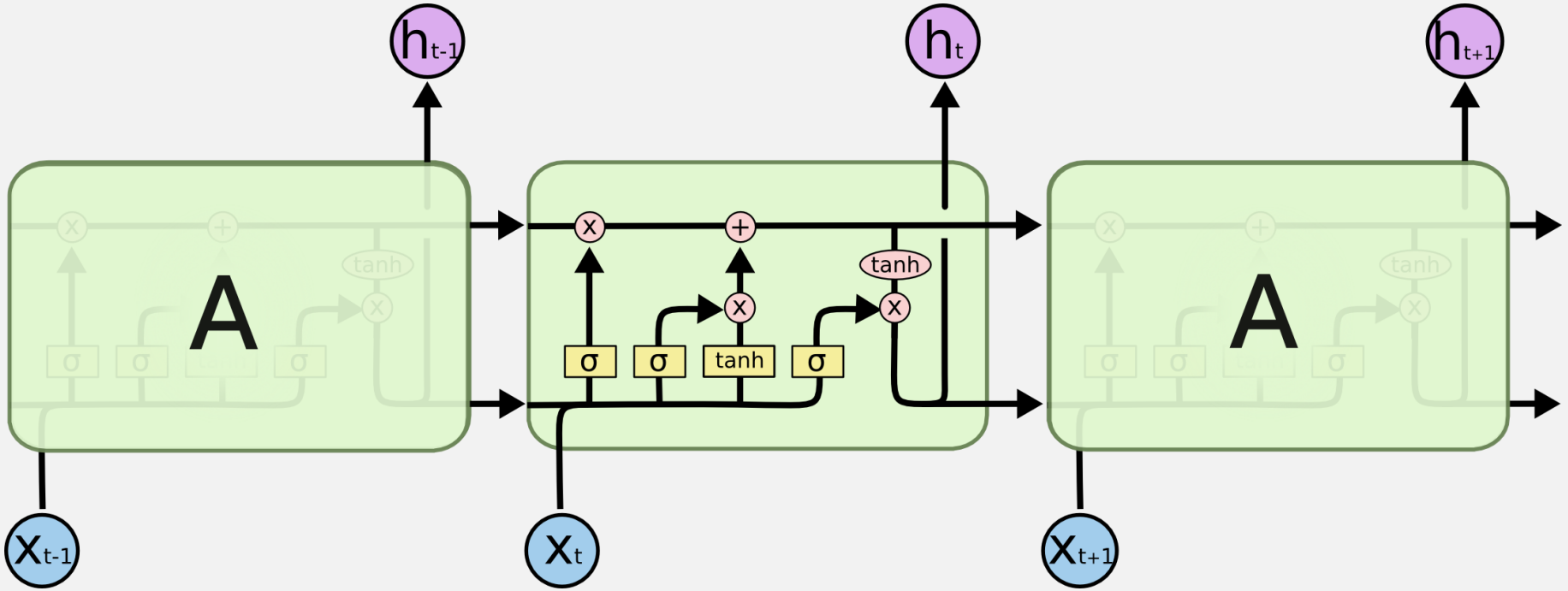
RNN - PROBLEMS

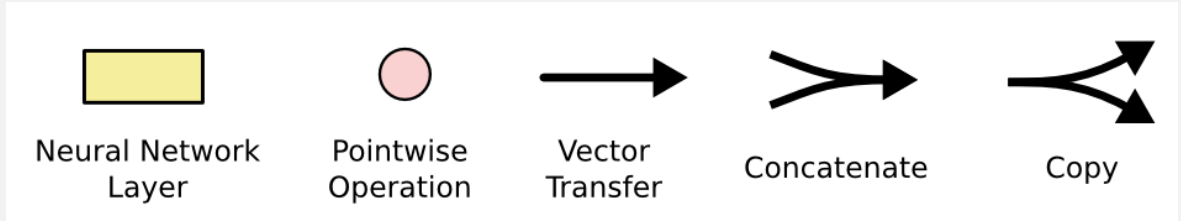
- Exploiding gradient
  - gradient grows uncontrollably large

  Solution: gradient clipping
- Vanishing gradient
  - when long input sequences (>10 time steps)
  - the gradient (during backpropagation through time) becomes too small
  - the contribution of information decays geometrically over time

  Solution: Long-short term memory unit (LSTM)

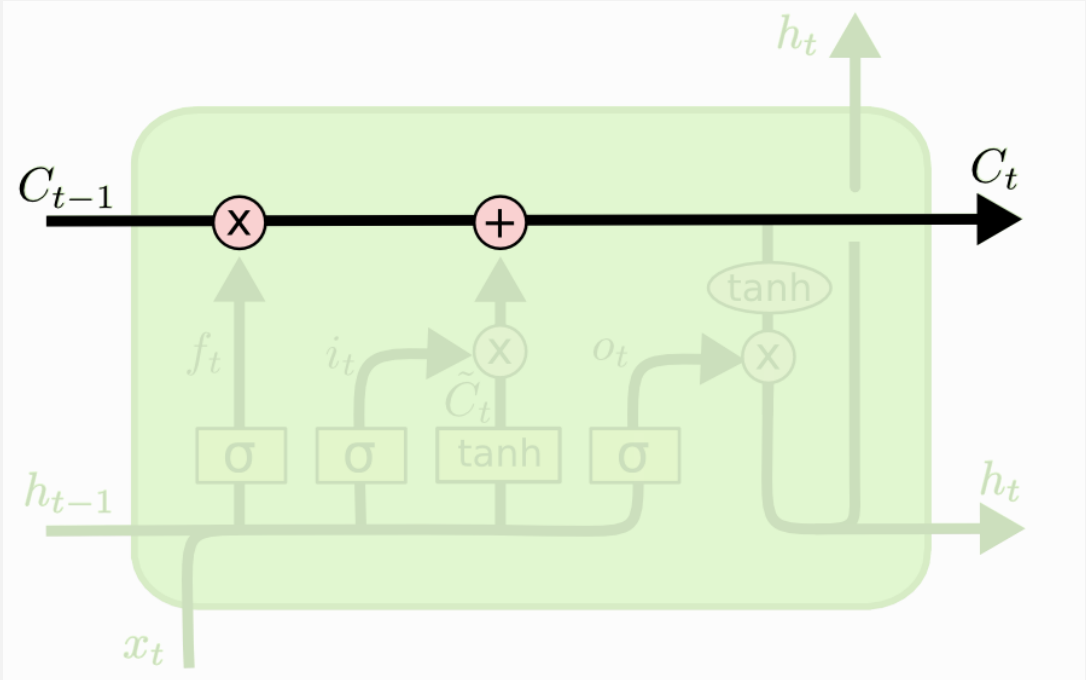
LSTM





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

CELL STATE – „MEMORY”



- The information flow to cell state is regulated by system of gates

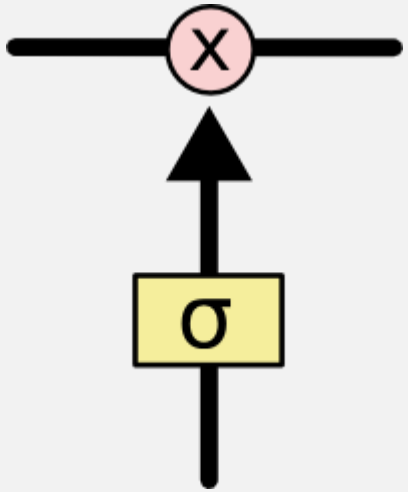
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

GATE

- optionally let information through
- are composed out of a sigmoid neural net layer and a pointwise multiplication operation

The sigmoid layer outputs numbers from [0,1]:

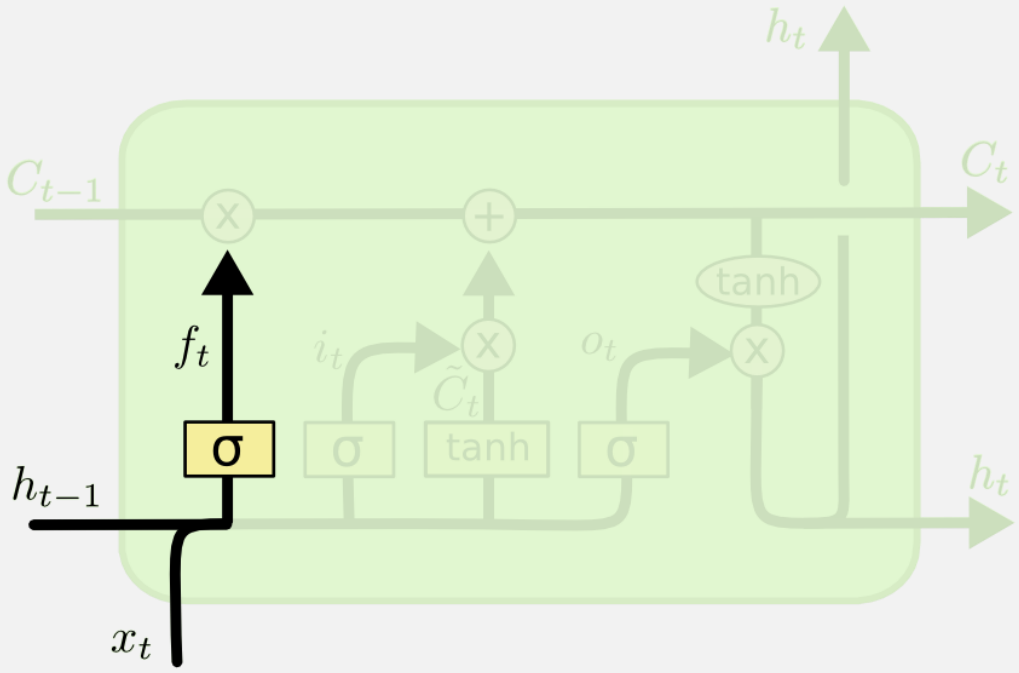
- 0 - “let nothing through”
- 1 - “let everything through”



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

FORGET GATE

- what information to throw away from the cell state



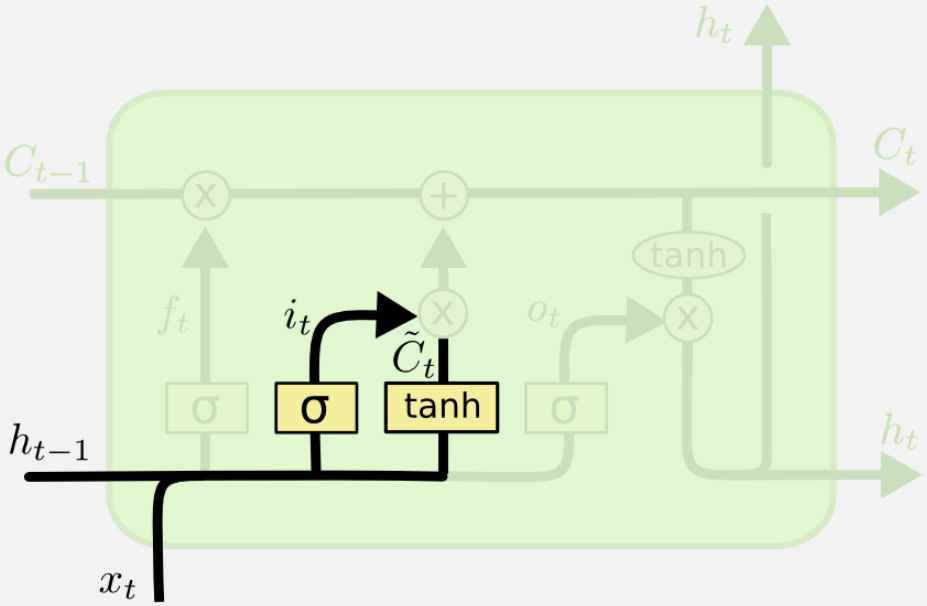
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

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

Example
Model tries to predict the next word in a sentence based on all the previous ones. In such a problem, the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.

INPUT GATE

1. decide which values to update
2. create a vector of candidate values to be added to cell state



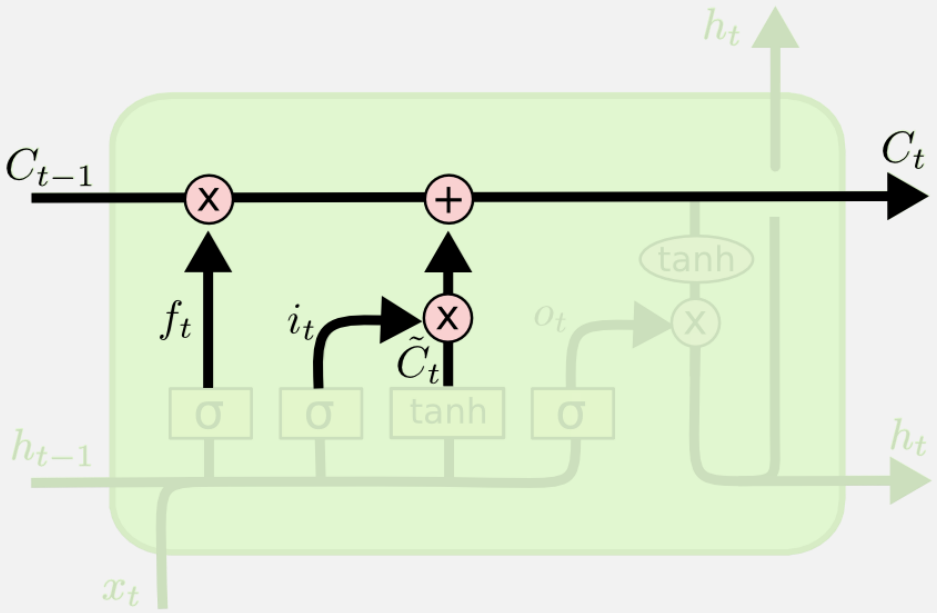
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

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

Example
In the case of our model, we'd want to add the gender of the new subject to the cell state, to replace the old one we're forgetting.

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

UPDATE THE OLD CELL STATE



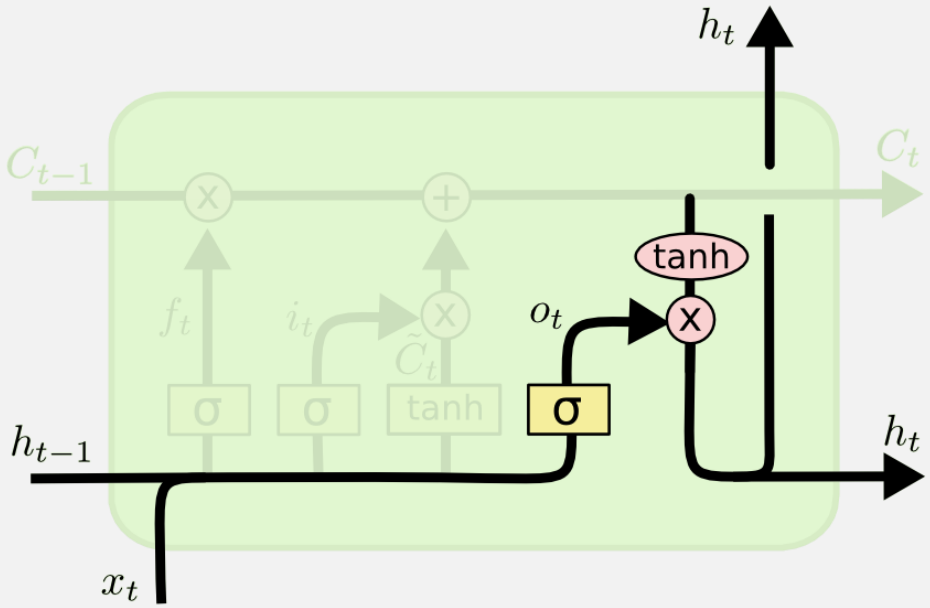
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Example
In the case of the language model, this is where we'd actually drop the information about the old subject's gender and add the new information, as we decided in the previous steps.

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

OUTPUT GATE

- decide what to output
- output is a filtered version of cell state



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

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

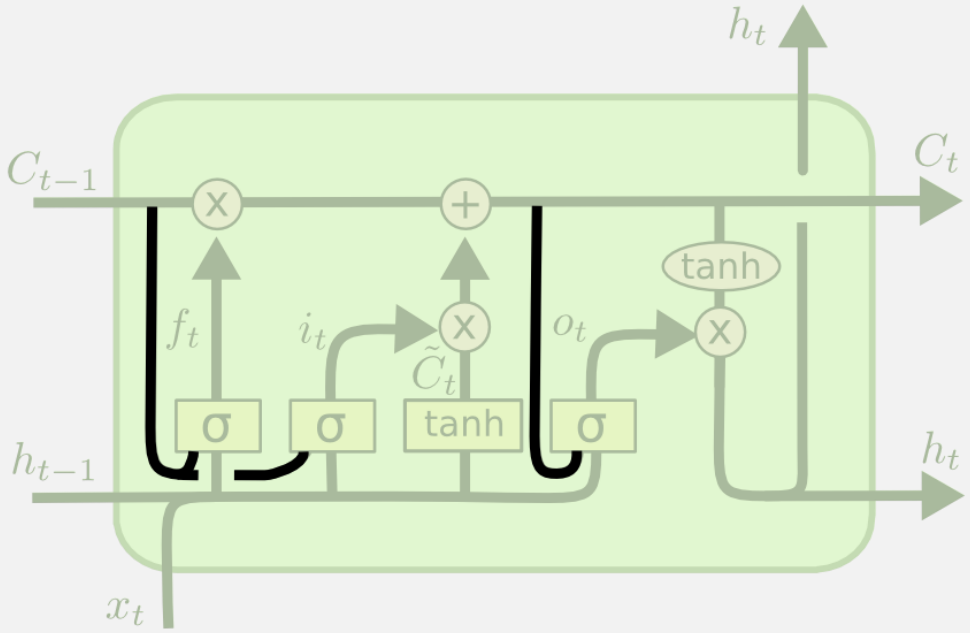
Example

For the language model example, since it just saw a subject, it might want to output information relevant to a verb, in case that's what is coming next. For example, it might output whether the subject is singular or plural, so that we know what form a verb should be conjugated into if that's what follows next.

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

VARIANTS OF LSTM – PEEPHOLE CONNECTIONS

- we let the gate layers look at the cell state
- it can be applied to only some gates

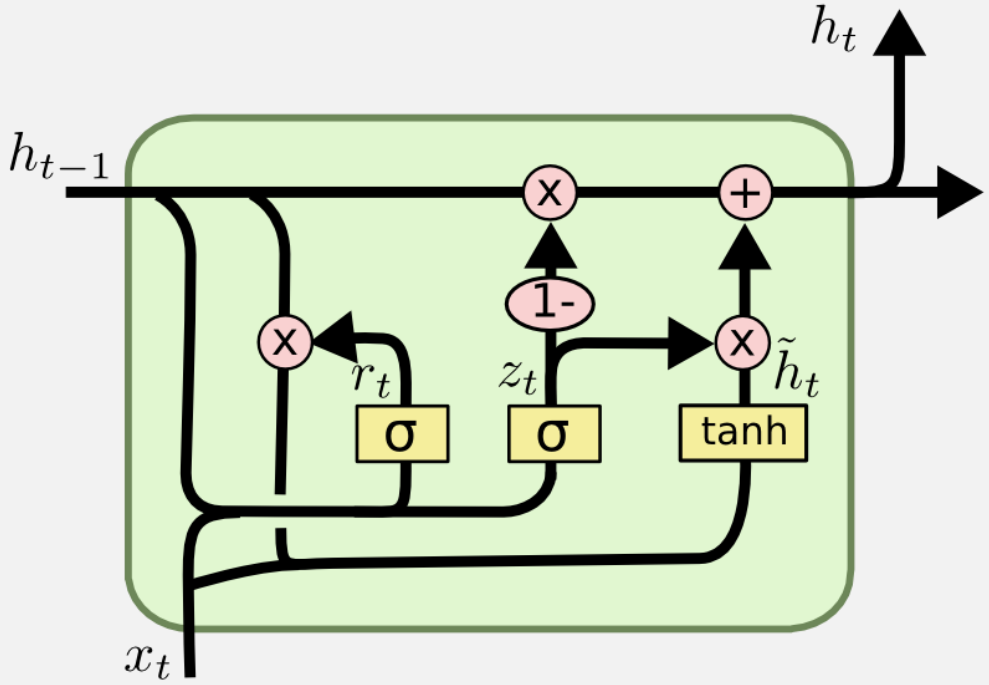


$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

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

GATED RECURRENT UNIT (GRU)

- combines the forget and input gates into a single "update gate"



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

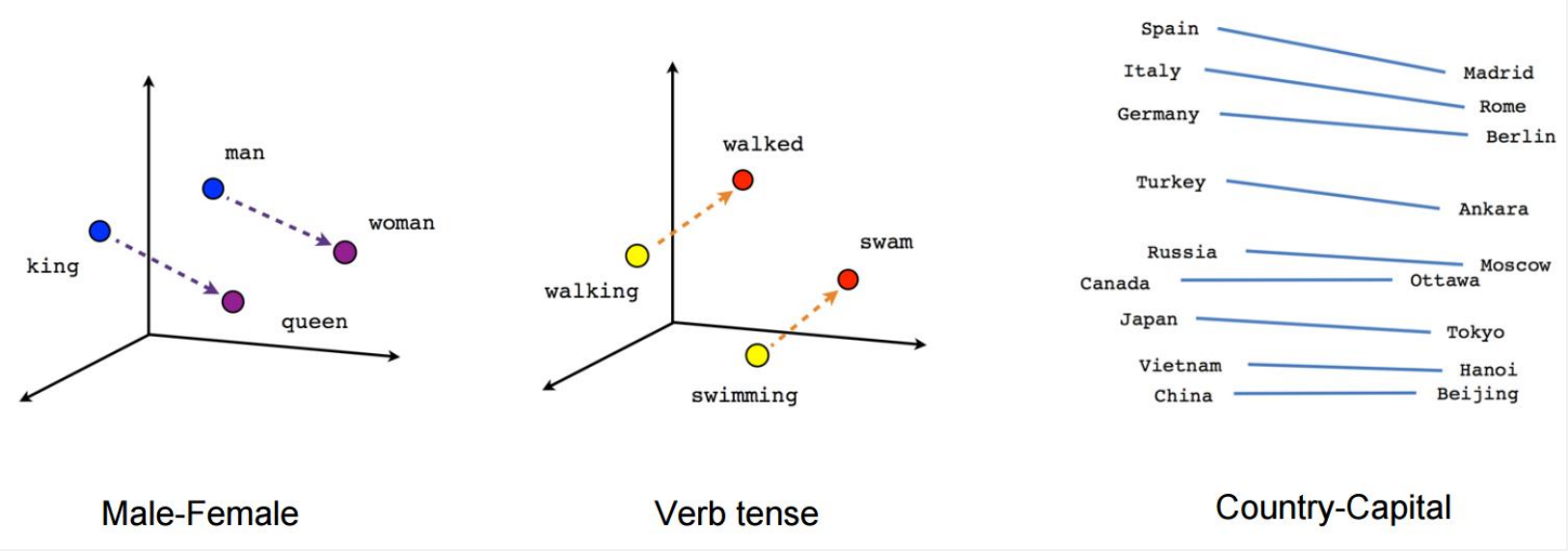
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

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

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

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

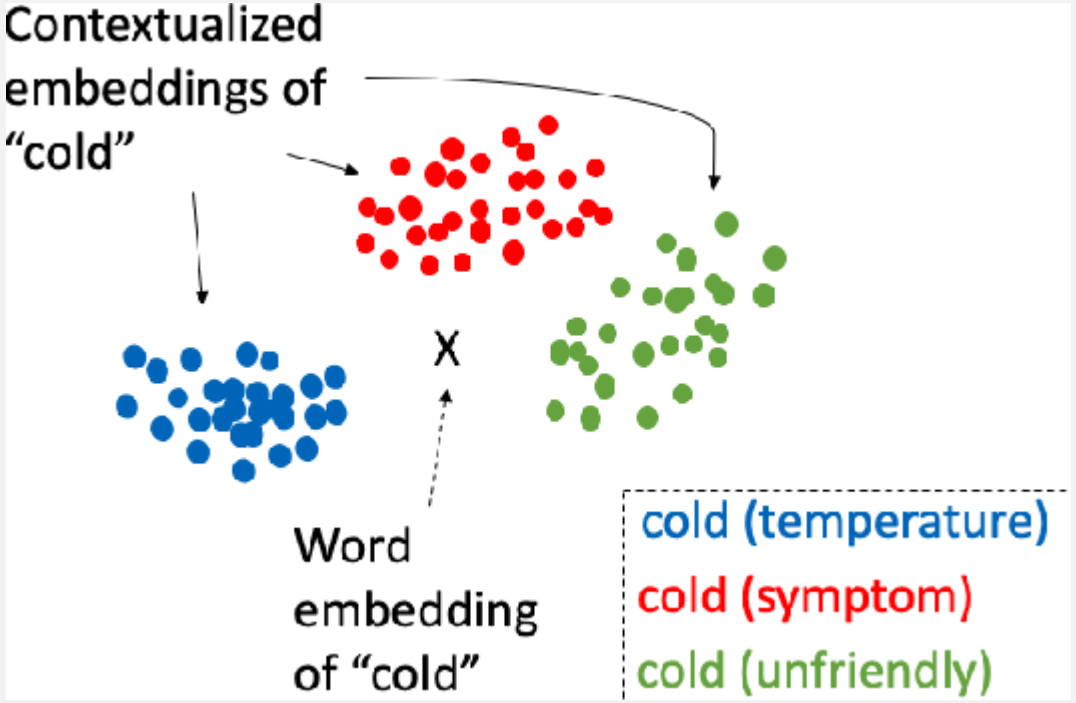
EMBEDDINGS – WORD2VEC



https://cnkndmr.gitlab.io/others/en/neural-network-word2vec.html

CONTEXTUAL EMBEDDINGS

- GloVe
- ELMo



https://www.semanticscholar.org/paper/Enhancing-Clinical-Concept-Extraction-with-Si-Wang/06b36e744dca445863c9f9aefe76aea95ba95999