An Introduction to Neural Networks

Long Short Term Memory (LSTM) and the Attention mechanism

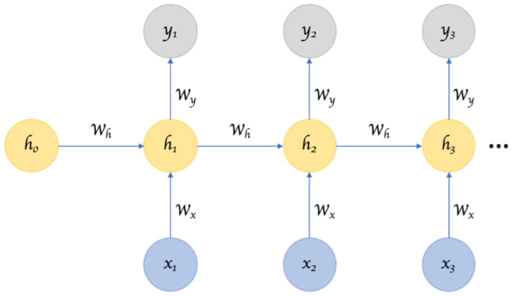
Ange Tato
Université du Québec à Montréal
Montreal, Canada

Agenda

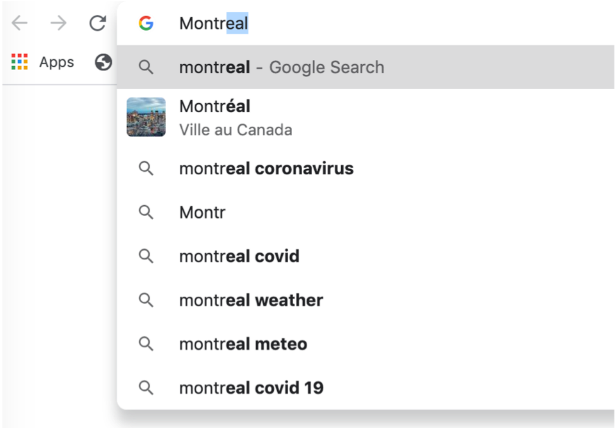
- Recurrent Neural Network (RNN)
- Long Short Term Memory (LSTM)
- Backpropagation Through Time (BPTT)
- Deep Knowledge Tracing (DKT)
- Attention Mechanism in Neural Networks

Do you know how Google's autocomplete feature predicts the rest of the words a user is typing ?

Collection of large volumes of most frequently occurring consecutive words



Fed to a Recurrent Neural Network



Prediction

Recurrent Neural Network (RNN)

- **Feed forward Network (FFN)** :
  - Information flows only in the forward direction. **No cycles or Loops**
  - Decisions are based on current input, **no memory** about the past
  - Doesn't know how to handle sequential data
- Solution to FFN : Recurrent Neural Network
  - Can handle sequential data
  - Considers the current input and also the previously received inputs
  - Can memorize previous inputs due to its internal memory

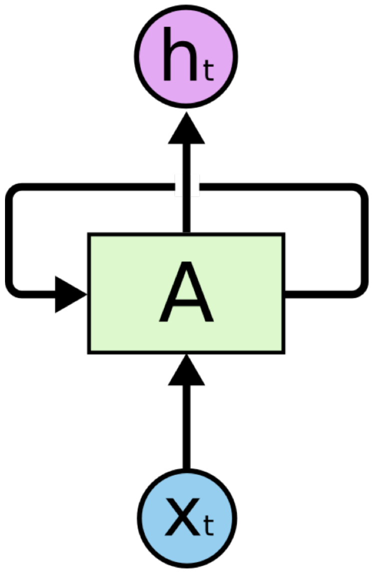


Fig1: RNN [4]

- **RNN**

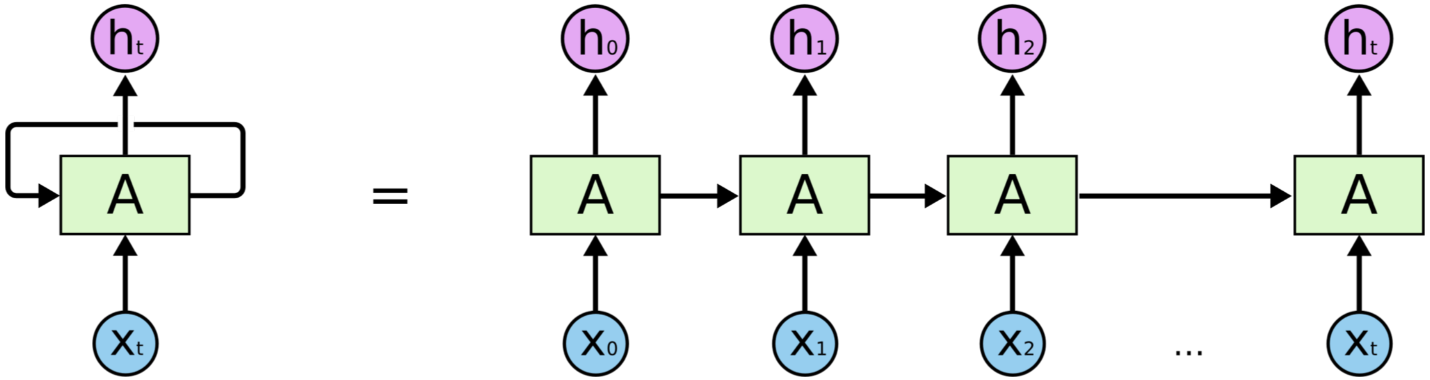


Fig2: An unrolled recurrent neural network [4]

- Useful in a variety of problems :
  - Speech recognition
  - Image captioning
  - Translation
  - Etc.

- **Math behind RNN**

$$h_t \; = f\,(W_{xh}\, x_t + W_{hy}\, h_{t-1})$$

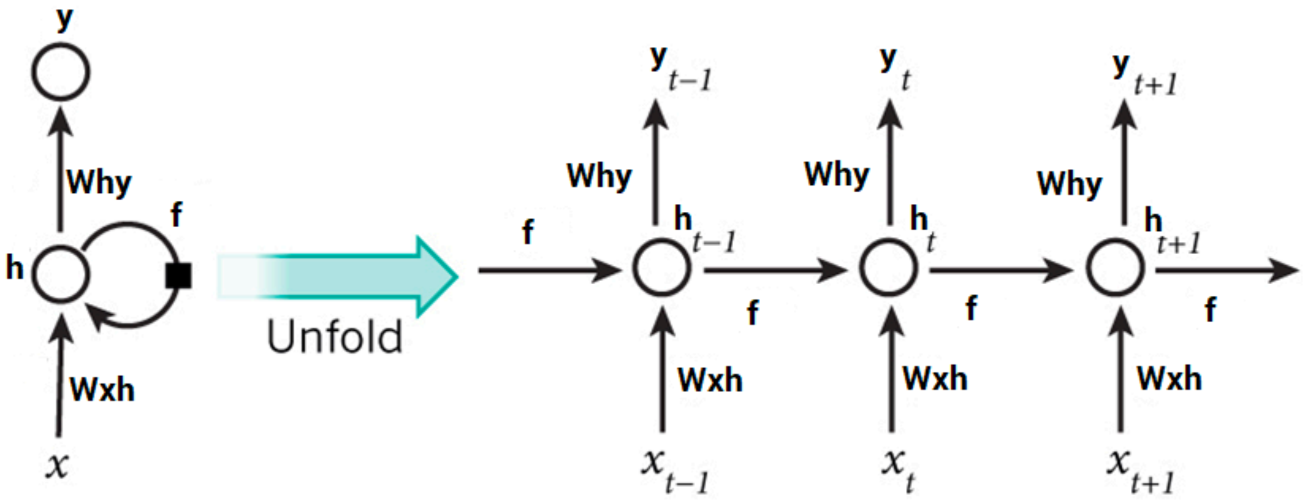


Fig3: Unfolded RNN [5]

- h_t : hidden state at time step t
- x_t : input at time step t
- W_{xh} and W_{hy} : weight matrices. Filters that determine how much importance to accord to both the present input and the past hidden state.

Long Short Term Memory (LSTM)

- A small example where RNN can work perfectly :
  - Prediction of the last word in the sentence : “The clouds are in the sky”

- RNN can’t handle situation where the **gap** between the **relevant information** and the point where it is needed is **very large**.

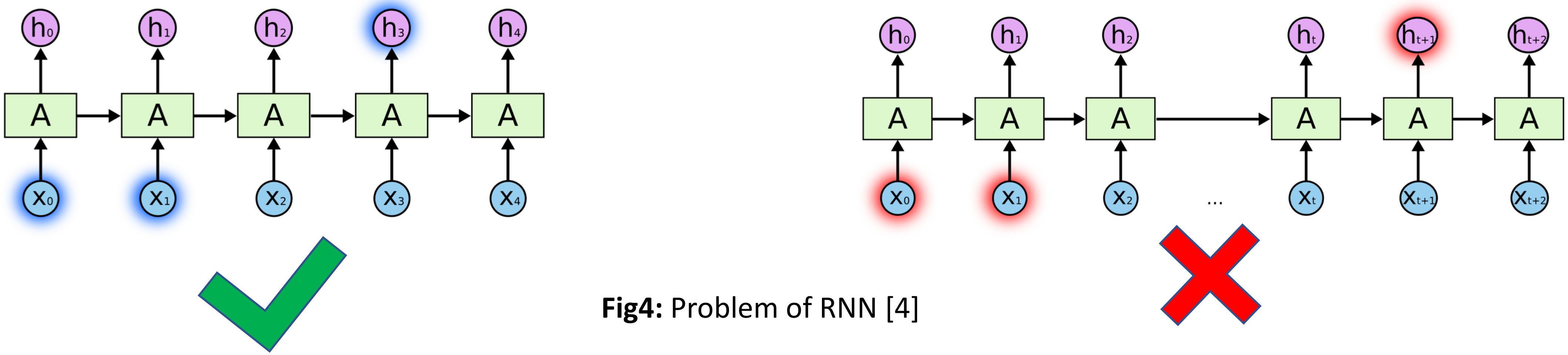


Fig4: Problem of RNN [4]

- LSTM can !

Long Short Term Memory (LSTM)

- **Long Short Term Memory networks** – usually just called "**LSTMs**" – are a special kind of RNN, capable of learning **long-term dependencies**. *Hochreiter & Schmidhuber (1997)*

- All recurrent neural networks have the form of a **chain of repeating modules** of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

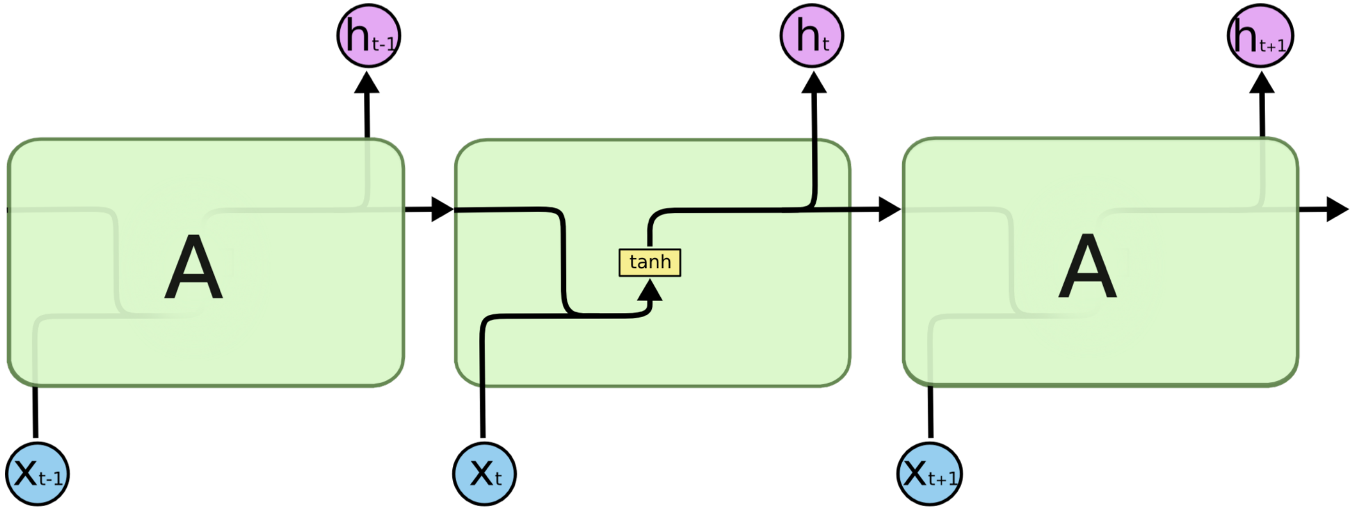


Fig5: The repeating module in a standard RNN contains a single layer [4]

- **LSTM** have the same chain like structure except for the repeating module.

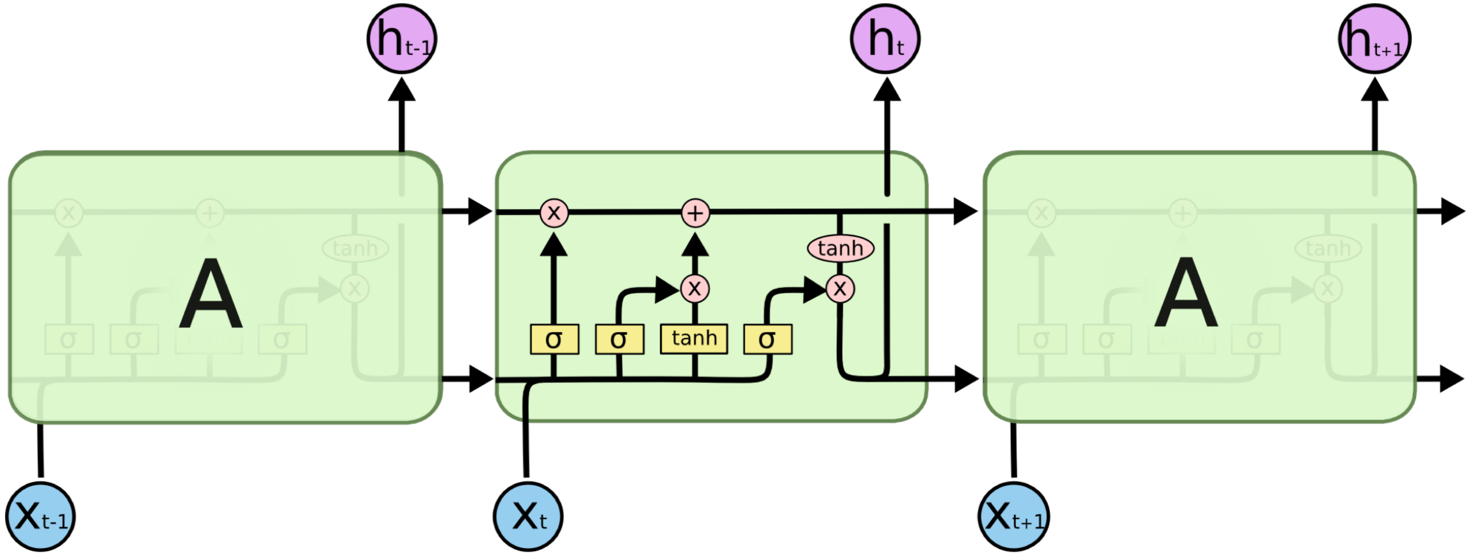
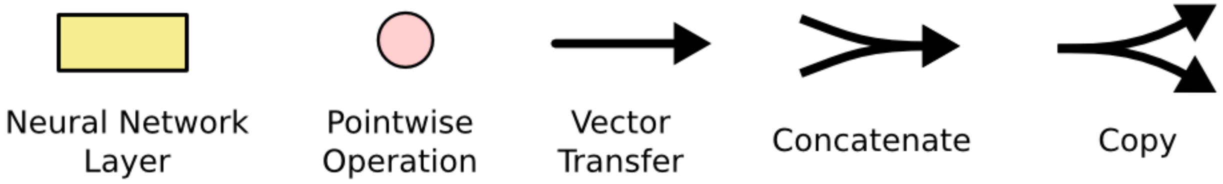
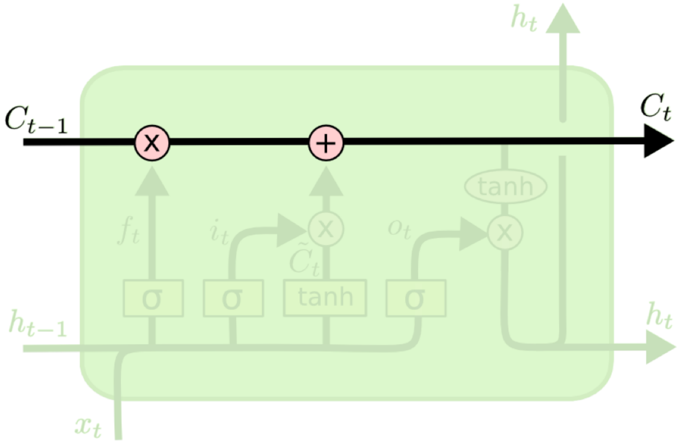


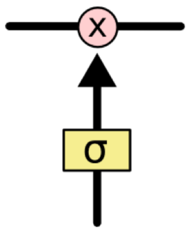
Fig6: The repeating module in a standard RNN contains a single layer [4]



- The core idea behind LSTMs is the **cell state**.

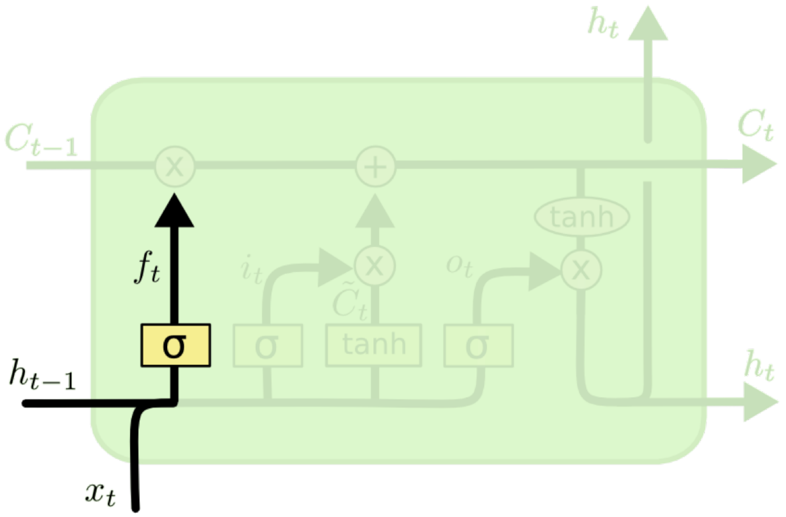


- The LSTM has the ability to **remove** or **add** information to the cell state : thanks to **gates**



- Gates are composed out of a sigmoid neural net layer and a pointwise multiplication operation

- Step-by-Step LSTM Walk Through
  - **Step 1:** Decide what information to **throw away** from the cell state, **forget layer.**



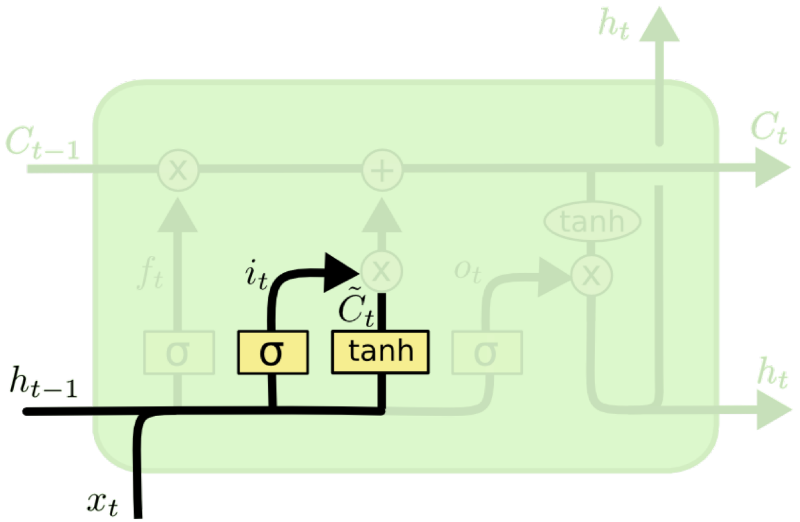
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

- **1** represents "completely keep this"
- **0** represents "completely get rid of this."

Long Short Term Memory (LSTM)

- Step-by-Step LSTM Walk Through

  - **Step 2**: Decide what new information we're going to store in the cell state

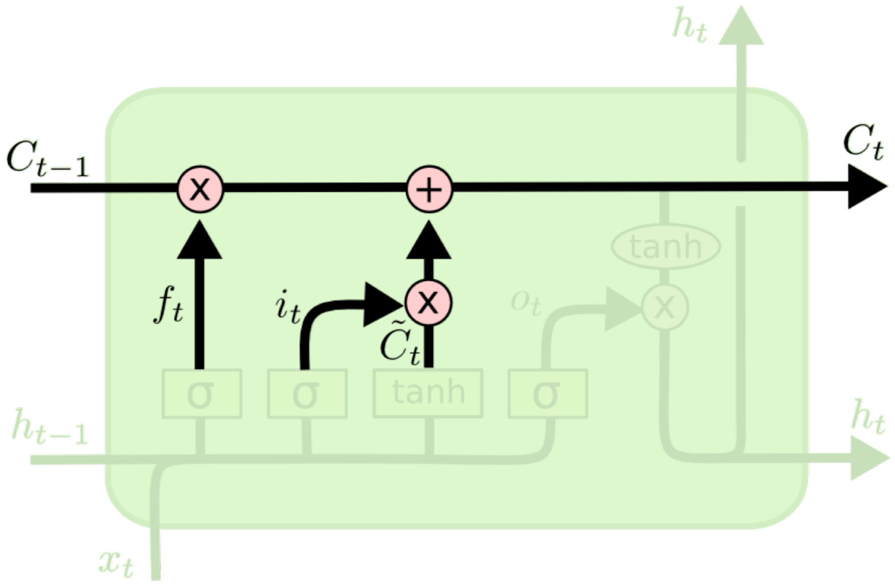


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

  - **Input gate layer** : decides which values we will update
  - **Tanh layer** : creates a vector of new candidate values

- **Example** : "I grew up in France… I speak fluent *French*."

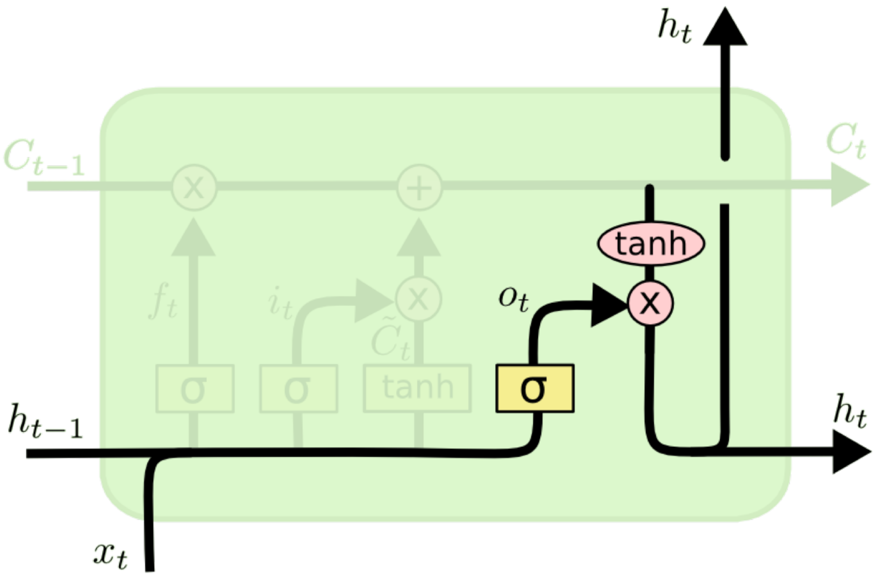
- Step-by-Step LSTM Walk Through
  - **Step 3**: Update the cell state



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

- **Example** : "I grew up in France… I speak fluent *French*."

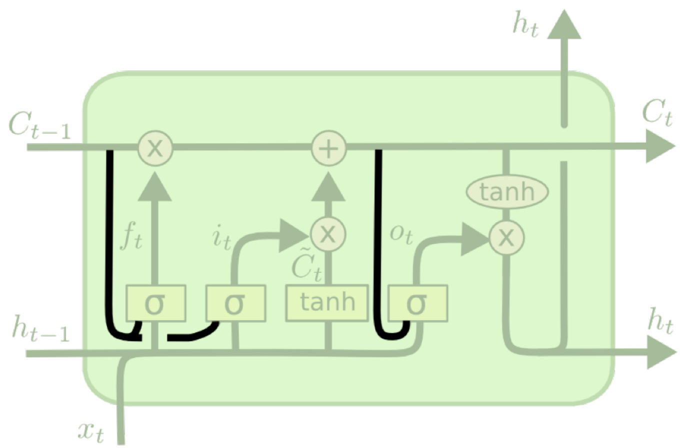
- Step-by-Step LSTM Walk Through
  - **Step 4**: Decide what is the output



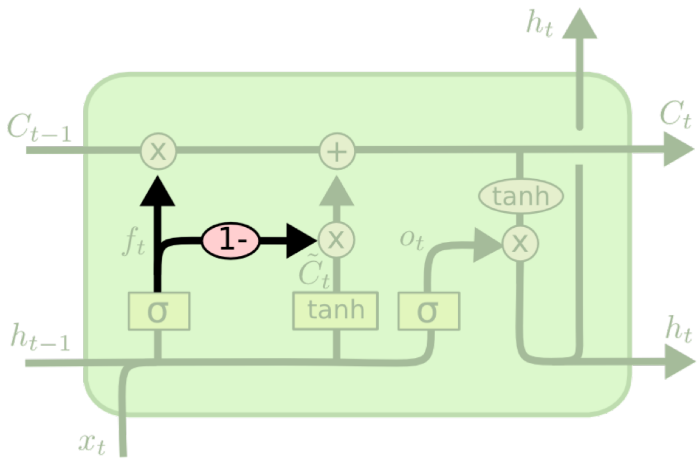
$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

- **Example** : “I grew up in France… I speak fluent *French*.”

- Variants of LSTM



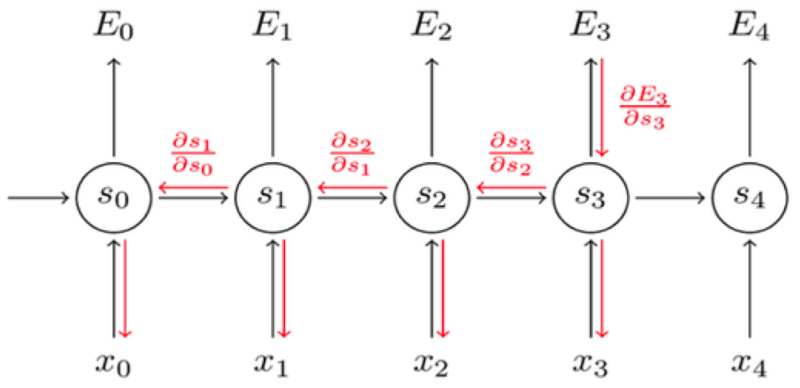
$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \ + \ b_o\right)$$



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

- **Backpropagation**: Uses partial derivatives and the chain rule to calculate the change for each weight efficiently. Starts with the derivative of the loss function and propagates the calculations backward.

- **Backpropagation Through Time**, or BPTT, is the training algorithm used to update weights in recurrent neural networks like LSTMs.



Backpropagation Through Time

Long Short Term Memory (LSTM)

- The good news !
- You don’t have to worry about all those intern details when using libraries such as Keras.

- Deep Knowledge Tracing (DKT) : Application of RNN/LSTM in education

- **Knowledge tracing** : modeling student knowledge over time so that we can accurately predict how students will perform on future interactions.

- Recurrent Neural Networks (RNNs) map an input sequence of vectors $x_1, \ldots, x_T$, to an output sequence of vectors $y_1, \ldots, y_T$. This is achieved by computing a sequence of 'hidden' states $h_1, \ldots, h_T$.

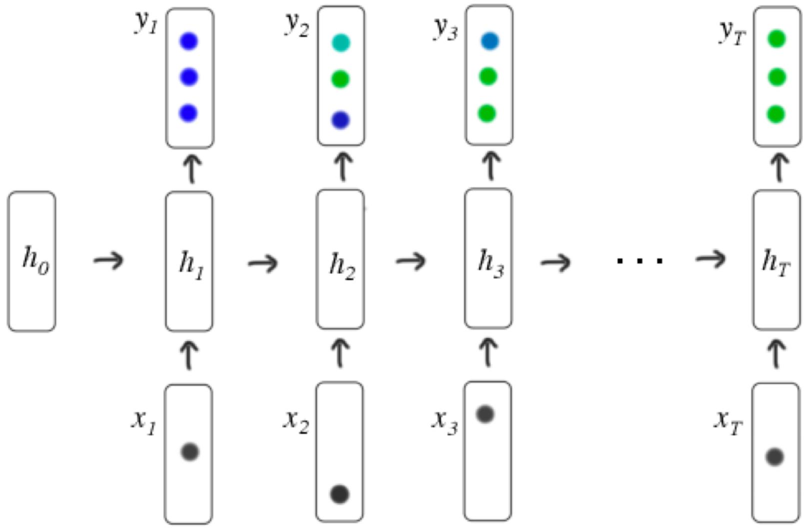
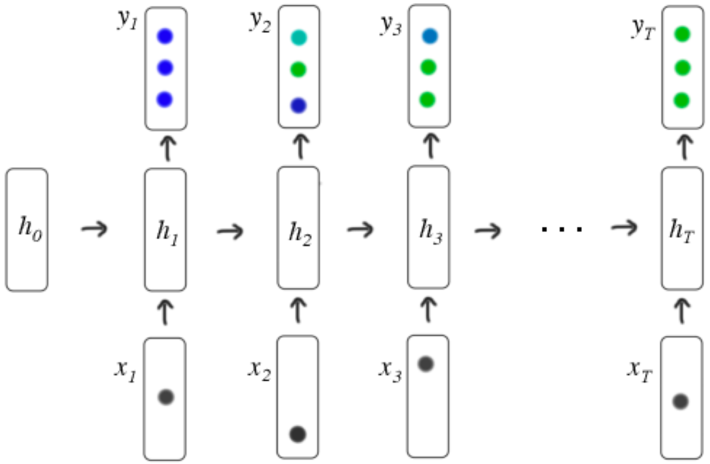


Fig7: Deep Knowledge Tracing [1]

- How to train a RNN/LSTM on students interactions?



- Convert student interactions into a sequence of fixed length input vectors x_t: one-hot encoding of the student interaction tuple h_t = {q_t, a_t}. Size of x_t = 2M (number of unique exercises)

- Y_t is the output : vector of length equal to the number of problems, each entry represents the predicted probability that the student would answer that particular problem correctly.

- **Optimization**
  - **Training objective** : negative log likelihood of the observed sequence of student responses under the model.
  - $\delta(q_{t+1})$: the one-hot encoding of which exercise is answered at time t + 1
  - ℓ : binary cross entropy
  - The loss for a single student is :

$$L = \sum_t \ell(\mathbf{y}^T \delta\left(q_{t+1}\right), a_{t+1})$$

Attention Mechanism

- In psychology, attention is the cognitive process of selectively concentrating on one or a few things while ignoring others.

> *A neural network is considered to be an effort to mimic human brain actions in a simplified manner. Attention Mechanism is also an attempt to implement the same action of selectively concentrating on a few relevant things, while ignoring others in deep neural networks.*

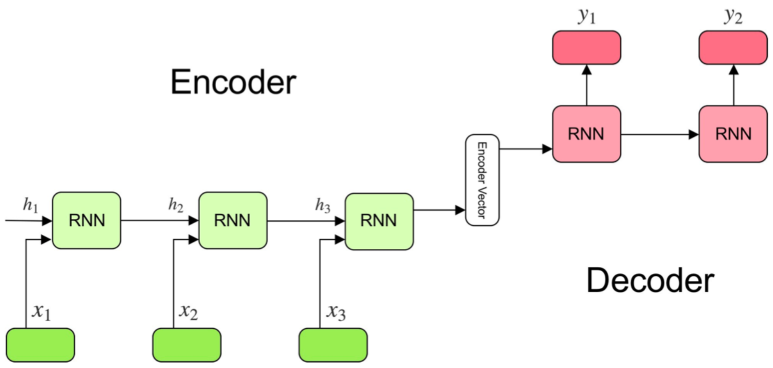
Attention Mechanism

- The attention mechanism emerged as an improvement over the encoder decoder-based neural machine translation system in natural language processing (NLP). Later, this mechanism, or its variants, was used in other applications, including computer vision, speech processing, etc.

- Before attention, neural machine translation was based on encoder decoder RNN/LSTM (Seq2Seq models). Both encoder and decoder are stacks of LSTM/RNN units. It works in the two following steps:

  - The encoder LSTM is used to process the entire input sentence and encode it into a context vector,

  - The decoder LSTM or RNN units produce the words in a sentence one after another



Attention Mechanism

- The main drawback of this approach : If the encoder makes a bad summary, the translation will also be bad !
- **Long-range dependency problem of RNN/LSTMs :** the encoder creates a bad summary when it tries to understand longer sentences.
- So is there any way we can keep all the relevant information in the input sentences intact while creating the context vector?
- Attention mechanism !

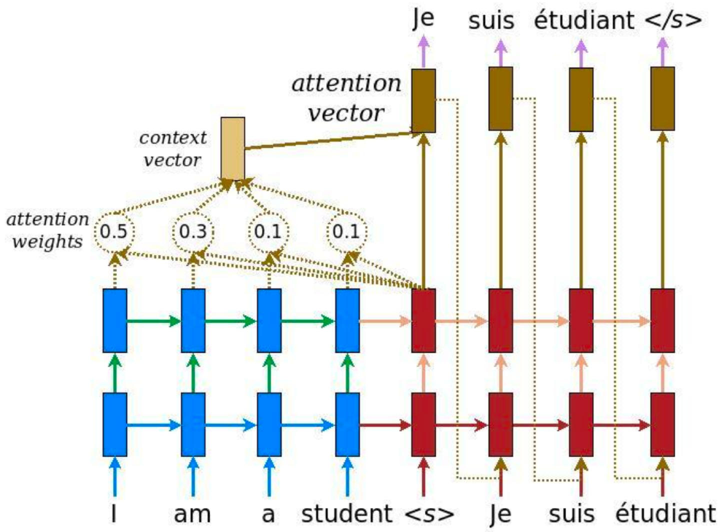


Fig8: attention mechanism applied to encoder-decoder [6]

Attention Mechanism

- How the attention mechanism work ?

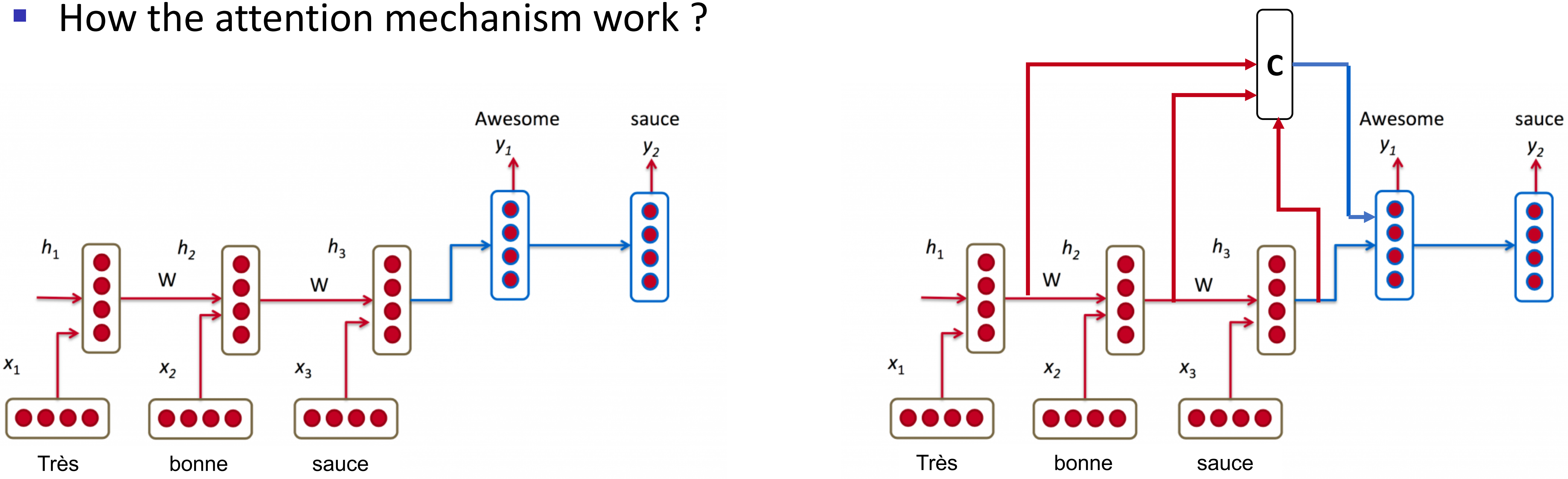
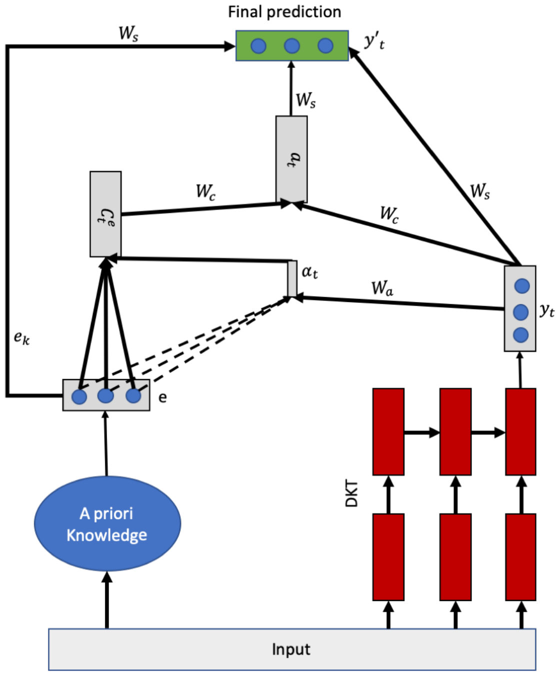


Fig9: Seq2seq model without and with attention mechanism

Attention Mechanism

- Attention mechanism in Education
- DKT + Attention mechanism (Tato et al. 2019)
- Use attention to incorporate expert knowledge to the DKT
- Expert knowledge = Bayesian network computed by experts
- Improve the original DKT if you have external knowledge



Application

References

1. C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, andJ. Sohl-Dickstein, “Deep knowledge tracing,” inAdvances in NeuralInformation Processing Systems, 2015, pp. 505–513

2. M.-T. Luong, H. Pham, and C. D. Manning, “Effective ap-proaches to attention-based neural machine translation,”arXiv preprintarXiv:1508.04025, 2015

3. A. Tato and R. Nkambou. Some Improvements of Deep Knowledge Tracing. 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 2019, pp. 1520-1524, doi: 10.1109/ICTAI.2019.00217.

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

5. https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/

6. https://medium.com/syncedreview/a-brief-overview-of-attention-mechanism-13c578ba9129