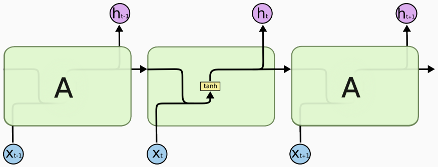
Long Short Term Memory Networks

Recurrent Neural Networks

Laura Bravo, Alejandro Pardo, Juan Pérez
Summer BCV

July 12, 2017

LSTM vs RNN



1

In an **RNN** there are multiple inputs. Inside each cell the input is processed by a tanh gate and the produced output is called a hidden state. The network is able to *remember* relationships between inputs by sharing the hidden state between cells.

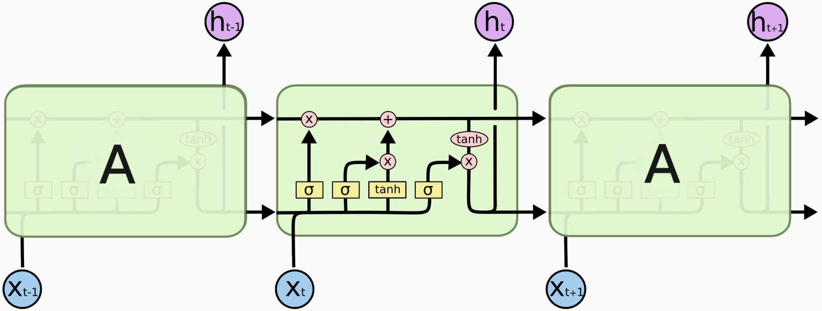
[1] All nice figures were taken from Christopher Olah's Understanding LSTM Networks, August, 2015. *http : //colah.github.io/posts/*2015 − 08 − *Understanding* − *LSTMs*/

LSTM vs RNN

So why do we like LSTMs?

- Gradient doesn’t vanish
- They have a longer term memory

How do LSTMs work?



There are two states inside each cell that are passed on through time, the cell state and the hidden state. They are what allow long and short term memory.

As opposed to RNNs, what we forget of the previous information is not completely lost we are keeping it in any of the two internal states.

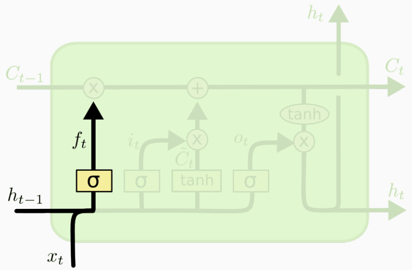
How do LSTMs work?

There are 3 internal gates that decide what information from the previous cell and the input is passed on to the next cell and outputted.

- i: input gate layer.
- f: forget gate layer.
- o: output gate layer.

How do LSTMs work?

Forget gate layer This layer looks at the previous hidden state (h_{t-1}) and the current input (x_t) and decides whether or not keep the past cell state (C_{t-1}). This is done by using the sigma operation, taking the output 0 to forget and 1 to remember.

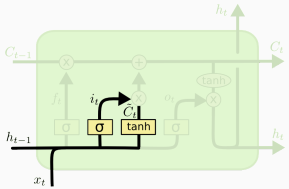


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

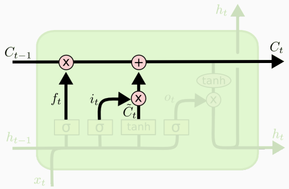
Forget gates could lead to the gradient vanishing, to avoid killing it too soon in the learning process their **bias is initialized positive**. Thus allowing the forget gates to be active later in the training.

How do LSTMs work?

Input gate layer This layer decides what new information x_t is important to keep in C_t, taking into account what happened before h_{t-1}. C_t is updated by using the i gate and the decision taken in the f gate.



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

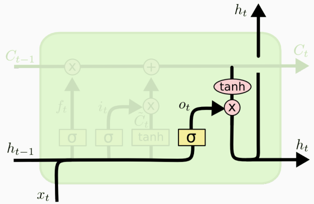


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Why are $\tilde{C}_t$ and i_t so similar? We are learning independently what new information we want to keep and whether or not we want to keep it.

How do LSTMs work?

Output gate layer This gate coupled with C_t defines the hidden state h_t which is what is later interpreted as the networks output.



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

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