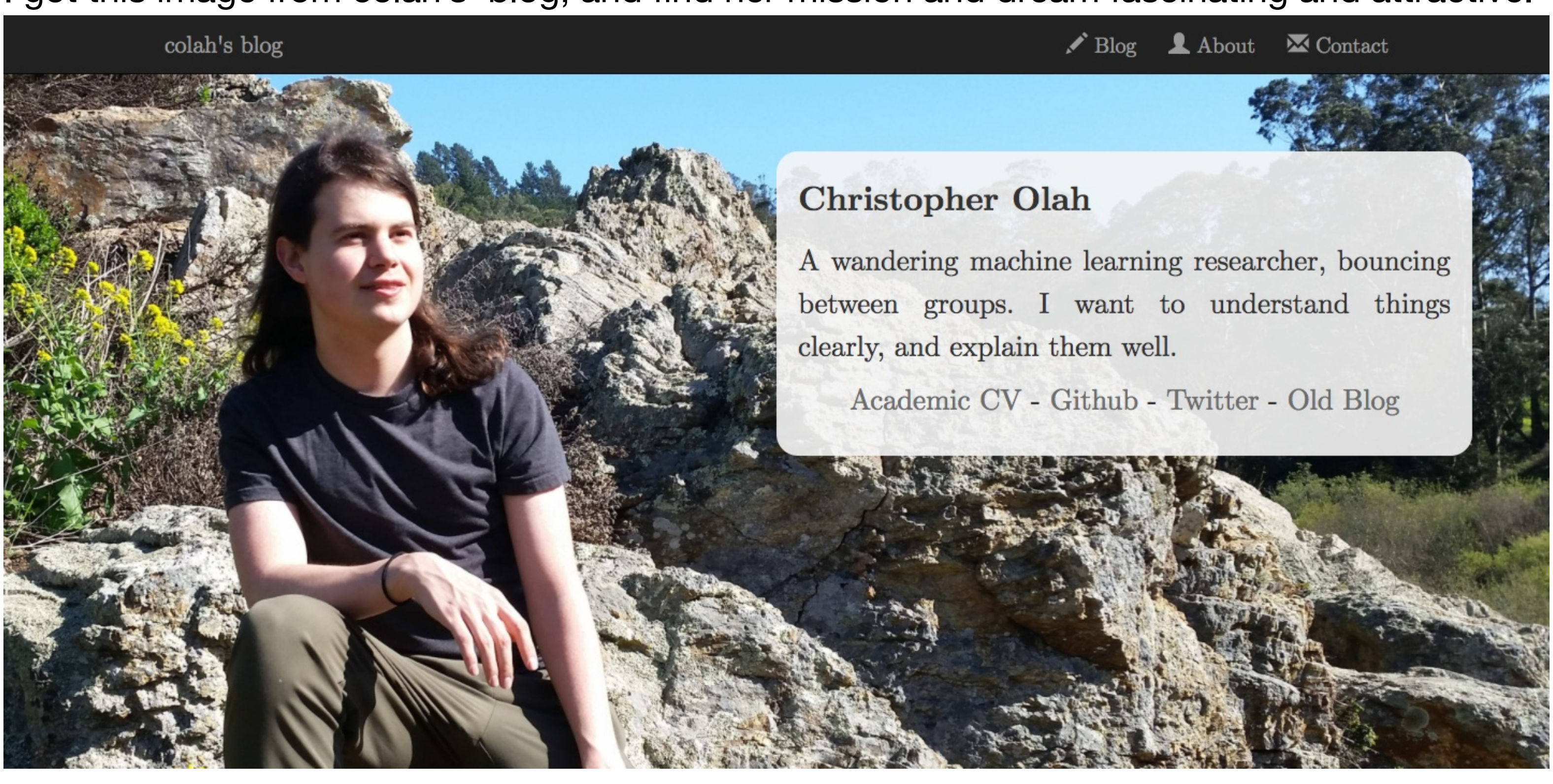
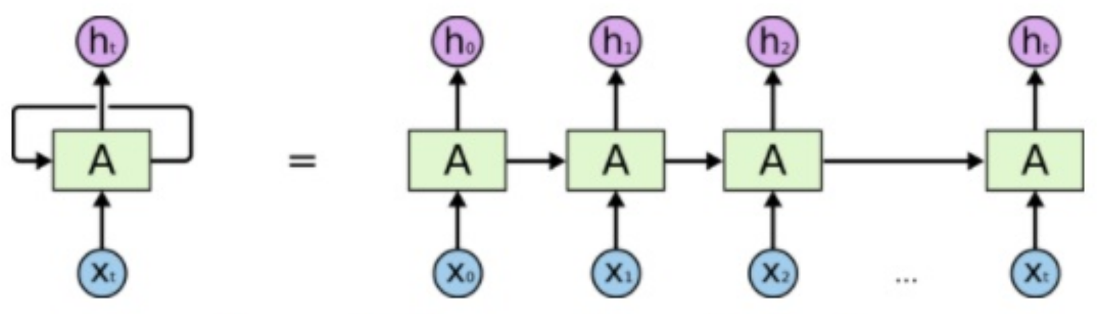
source:[colah's blog](#)
[Tensorflow](#)

I got this image from colah's blog, and find her mission and dream fascinating and attractive.



Here is the picture of an unrolled example in RNN that we usually saw.



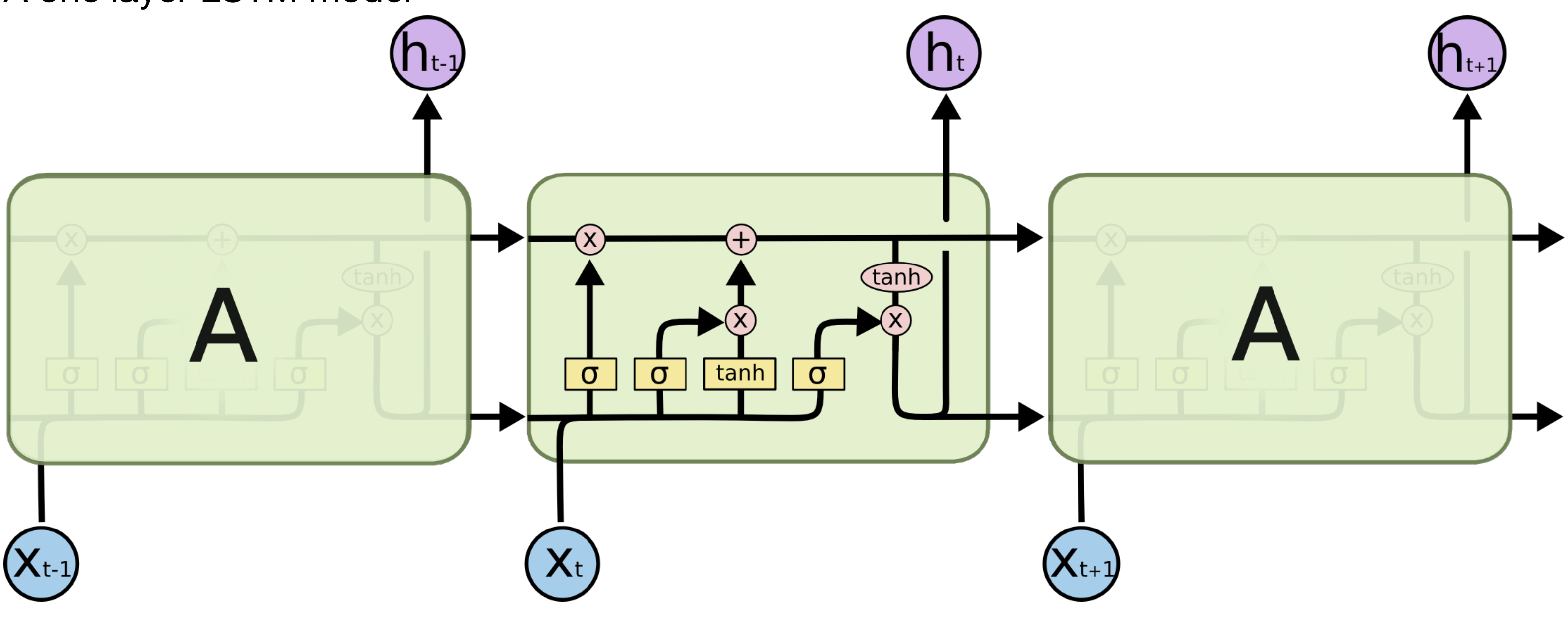
An unrolled recurrent neural network.

In this essay , the author is specially interested in LSTM a special type of RNN, which , i think is the staple of mainstream nowadays

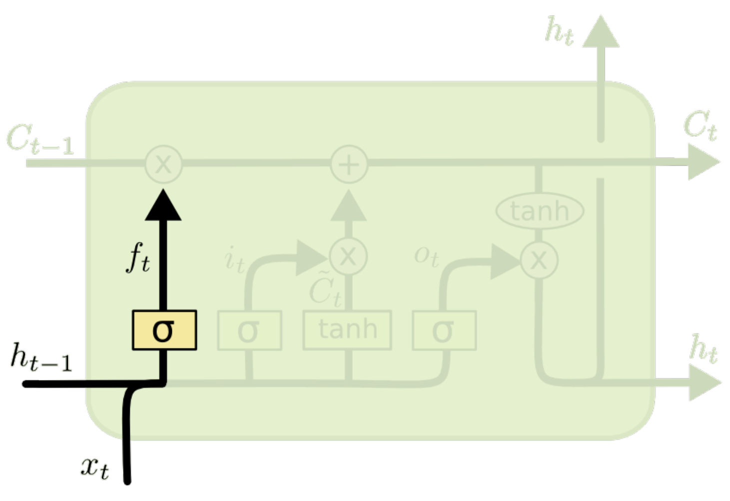
Why is LSTM cell so hot in recent days.

well one thought from previous learning experience is the Long term dependency problem when previous state is too long from current state it will have gradient explosion(not sure if i used this words correctly). I'm not sure if underflow or overflow is the proper words here.

Here is a basic cells graph
A one layer LSTM model

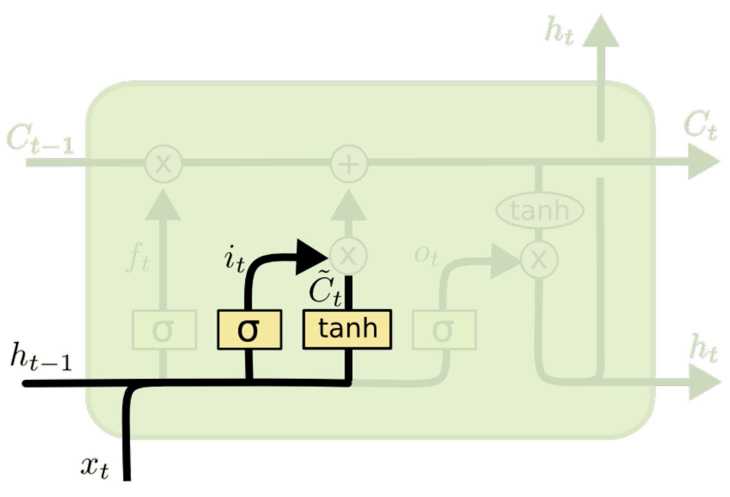


As we can see in the above picture, the cell contains multiple layers
first, previous state info and input info with same weight will be pass through sigmoid function to determine whether it will be added to the cell



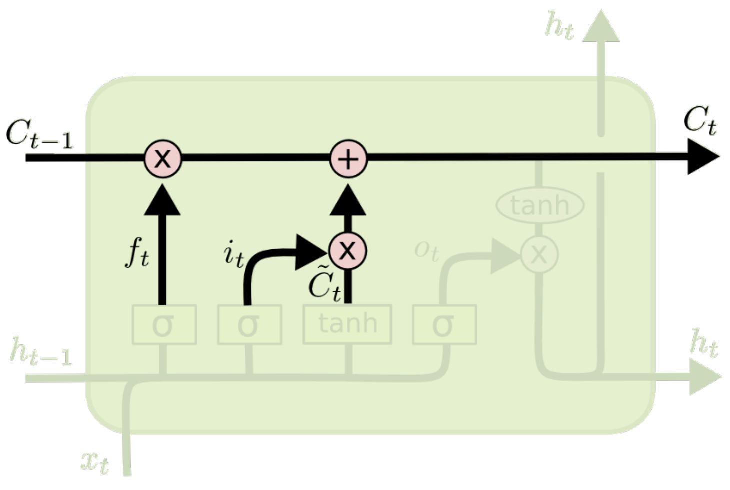
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Then, previous state info and input info with another weight will be pass through sigmoid function and a tanh function



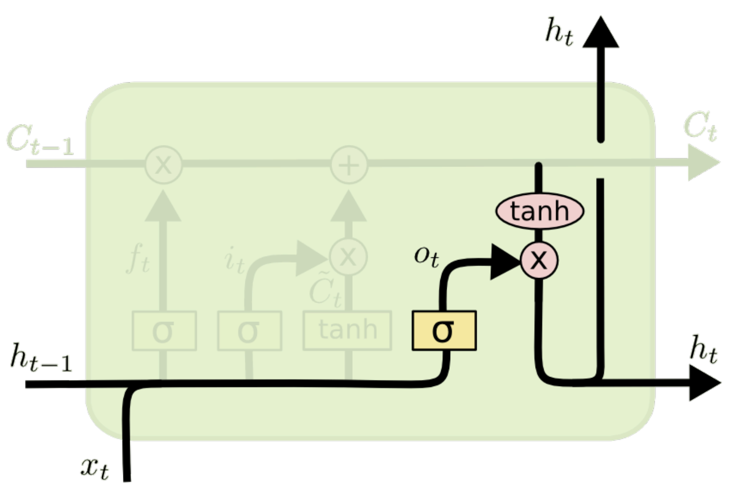
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

the next step is to combine them together



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

then, it will come with the next step

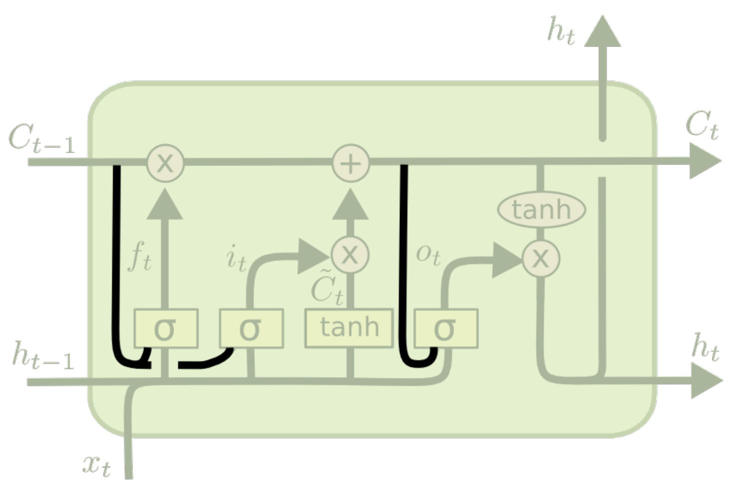


$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Even now I understand how it works, but I guess i'll never know why this entire function works.

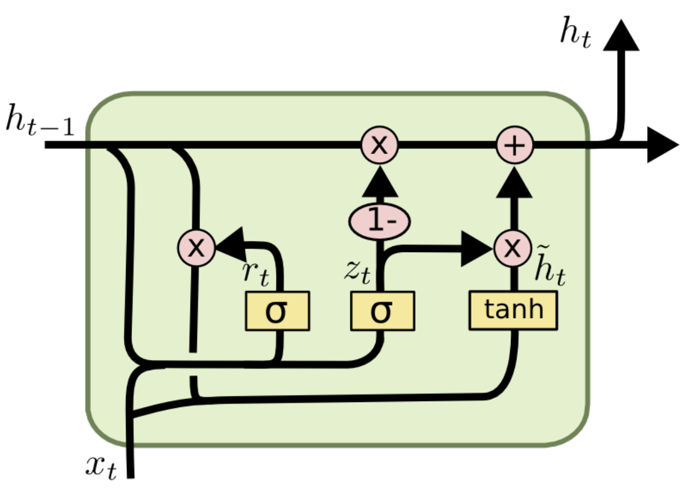
Variants on Long Short Term Memory

The next topic discussed the various LSTM cells.



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

this differs from basic LSTM is that it contains a peephole to all the gates, not sure how this will help improve the model



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

this one looks more simple than the previous one

So according to paper published in 2015 by Greff, the result from different LSTM cells seems more or less the same.

I guess there people who created these LSTM cells are just guessing the that it will work, not really understand what's exactly going on.

conclusion

the next possible area to explore is attention , here is a [2015 paper](#) written by kelvin xu introducing the concept of attention