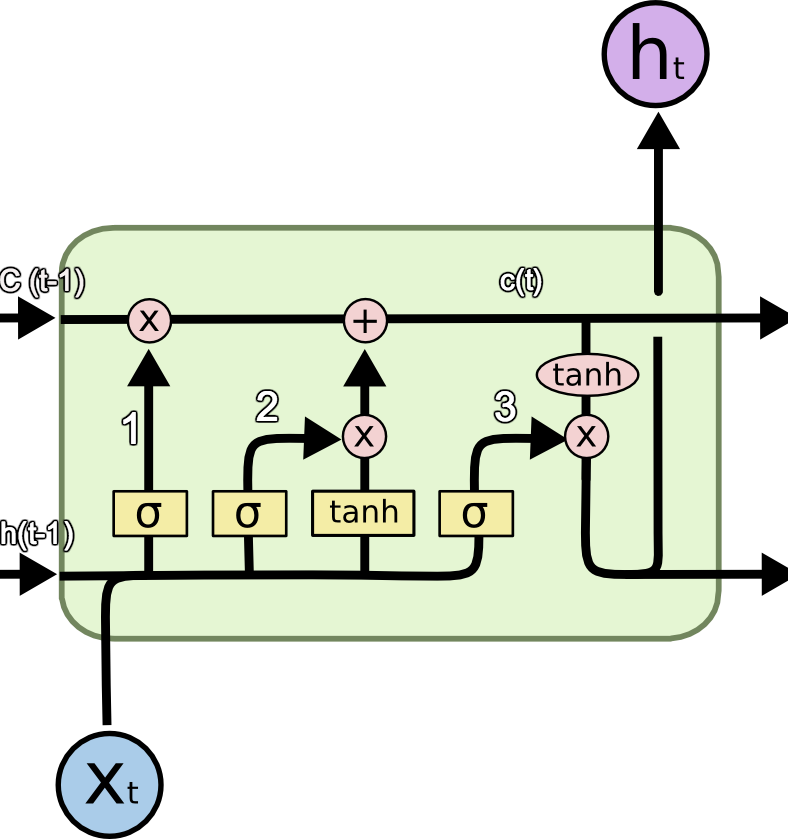
Problem with RNN

During the training of RNN, as the information goes in loop again and again which results in very large updates to neural network model weights. This is due to the accumulation of error gradients during an update and hence, results in an unstable network. At an extreme, the values of weights can become so large as to overflow and result in NaN values.The explosion occurs through exponential growth by repeatedly multiplying gradients through the network layers that have values larger than 1 or vanishing occurs if the values are less than 1.

**Long Short Term Memory**

The above drawback of RNN pushed the scientists to develop and invent a new variant of the RNN model, called Long Short Term Memory. LSTM can solve this problem, because it uses gates to control the memorizing process.



Understanding LSTM...

The top most line is cell state, we can think it like a conveyor belt.

STEP-1:

The first step is to identify those information that are not required and will be thrown away

from the cell state. his decision is made by a sigmoid layer called as forget layer(ft).

sigmoid(weight[output from previous timestamp,new input] + bias )

we get an output ie 0 or 1

0 means get rid of it and 1 means completely keep.

STEP-2:

The next step to decide, what info. we are going to store in cell state. It has follo. steps:

(it)A sigmoid layer called "input gate layer" decides which which values will b updated.

(ct)Next, a tanh layer creates a vector of new candidate values, that could be added to the state.

STEP-3:

update cell state

ft \* ct-1 + it \* ct

STEP-4:

will run a sigmoid layer which decides what parts of the cell state we are going to output.

then we put cell state through tanh and \* it by output of sigmoid gate,

so that we only output the parts we decide to.