Long Short-term memory (LSTM)

* LSTM is one of the varient of RNN.

Why LSTM is required ?

Some of the usecases are

a) Sometimes model needs to remember context and sometimes model needs to remember previous word.

b) Sometimes sentences was too long and supposed to remember everything.

c) Sometimes need to remember previous small context not all sentences.

Examples:

a) How are ______ ? you → short term memory

* 'you' can have high probability, because in most of the scenarios we use it.

* Why short term ?

Because 'you' is predicted from sequence 'How are' which is short context. Model didn't predict 'you' from the overall context in this page.

b) I am going to learn ________ LSTM

↓

Long term memory

* It can be anything like NLP, DL, ML (or) LSTM.

* Why LSTM?

Because from the starting, we are learning about LSTM. So, it will have high probability.

* Why Long term?

→ Because 'LSTM' is predicted from long context (Mostly all sentences from start).

→ It didn't predict from short context i.e 'I am going to learn'

* So, sometimes we need short context, sometimes we need long context, sometimes we need very long memory, sometimes we need to change the context.

c) changing context

After learning this , I am going to do ______

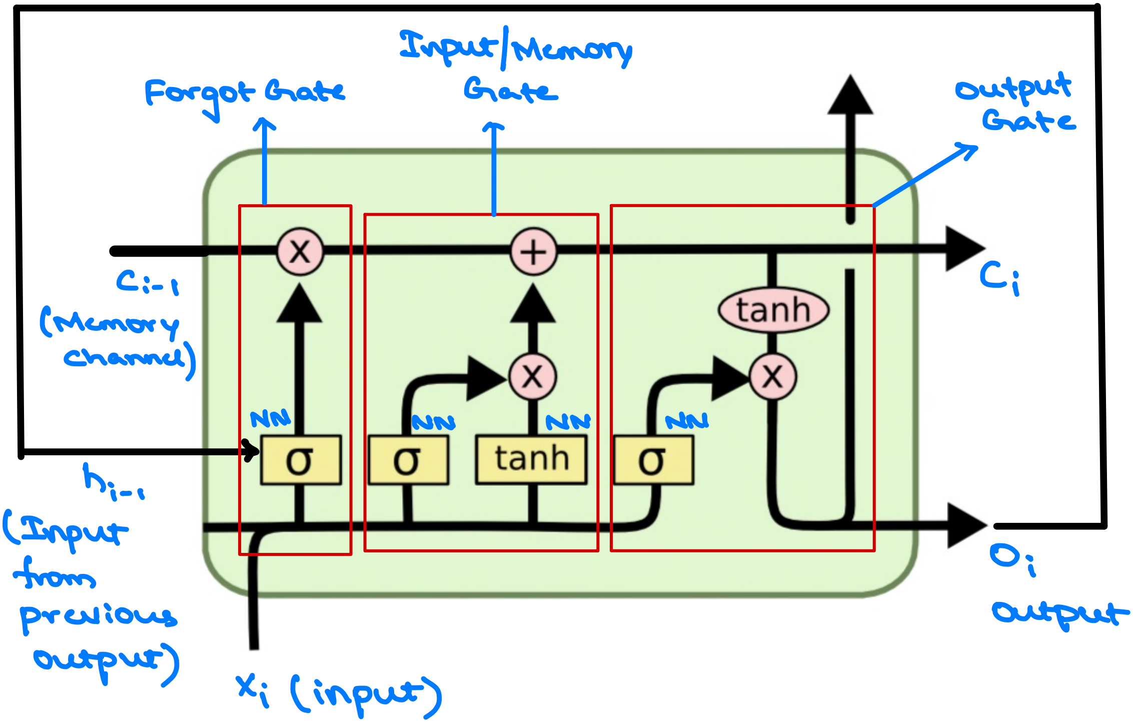
assignment
cooking
rest
project

Here we don't need to remember previous context.

So, we need a model that can satisfy all previous criteria. This is why we need LSTM.

Next →

Architecture



Forgot Gate:

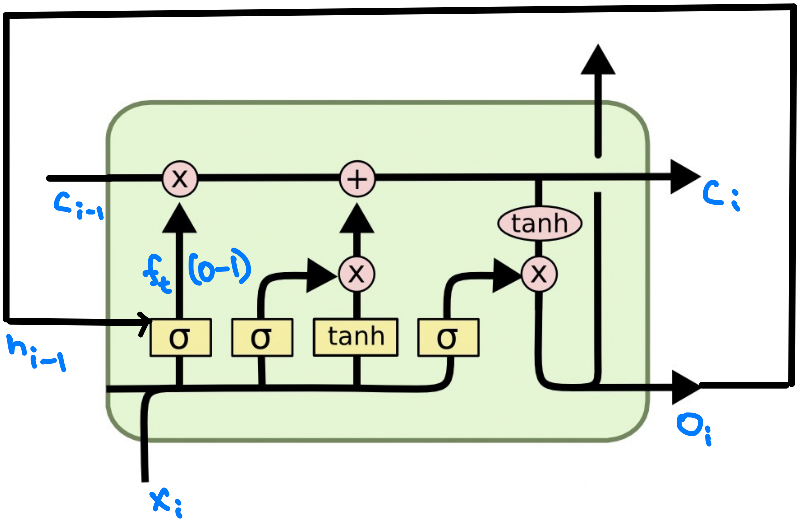
* At t=0, let us assume x_i = H. Now convert 'H' into vector and send to NN which is using sigmoid activation function.

* As t=0, h_{i-1} is empty. No previous output.

* This NN always gives output b/w 0 to 1.

close to 0 → NN learnt that forgot about previous context

close to 1 → NN learnt that remember previous context

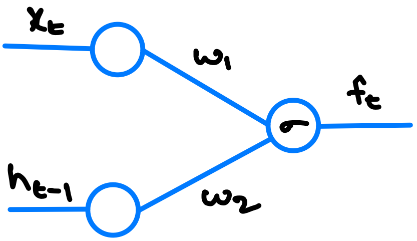


* At t=0, Memory channel (c_{i-1}) is 0. So, the output of forgot gate at t=0, will be zero always because 0 * (0-1) = 0.

* Basically forgot gate tells you, how much I need to forgot from the previous context.

Equation

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

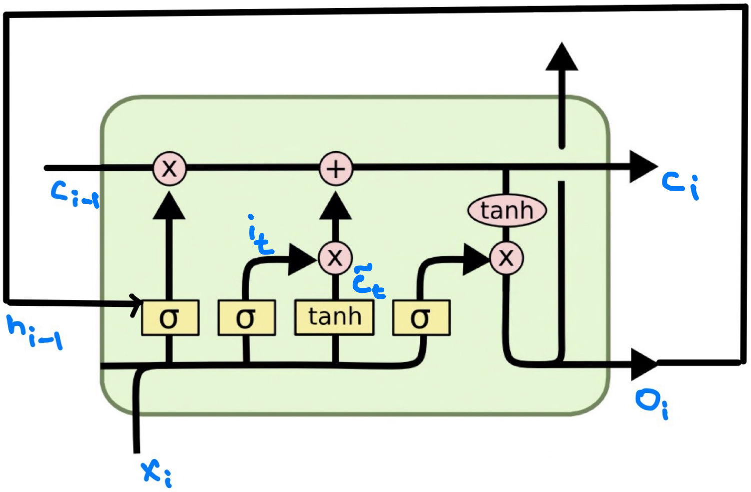


$$w_f = [w_1, w_2]$$

Input / Memory Gate:

- In forgot gate, we learnt how to forgot previous word / context.

- In Input / Memory gate, we learn how to memorize current word.



At $t=0$

$x_i =$ 'H'

$h_{i-1} = 0$

- Sigmoid NN → Decides whether current input we are supposed to add into memory (or) not. [Important (or) not]

- tanh NN → If I have to add current input to the memory, then howmuch need to be added is decided by this NN. (Regulates NN)

- For example, $i_t = 1$, $\tilde{C}_t = 0.09$ then output of memory channel

$$c_{i-1} = (1 * 0.09) + 0 = 0.09 \Rightarrow$$ vector 'H' memory

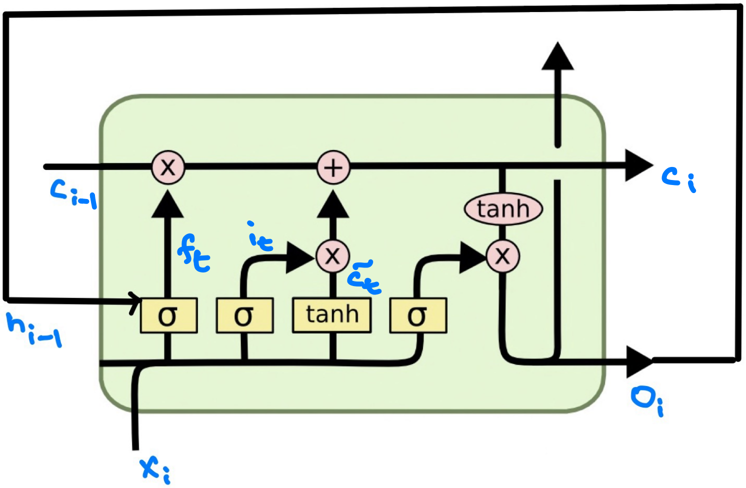
* Input / Memory gate learns new things (or) context and adds the context to the previous one.

Equations

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

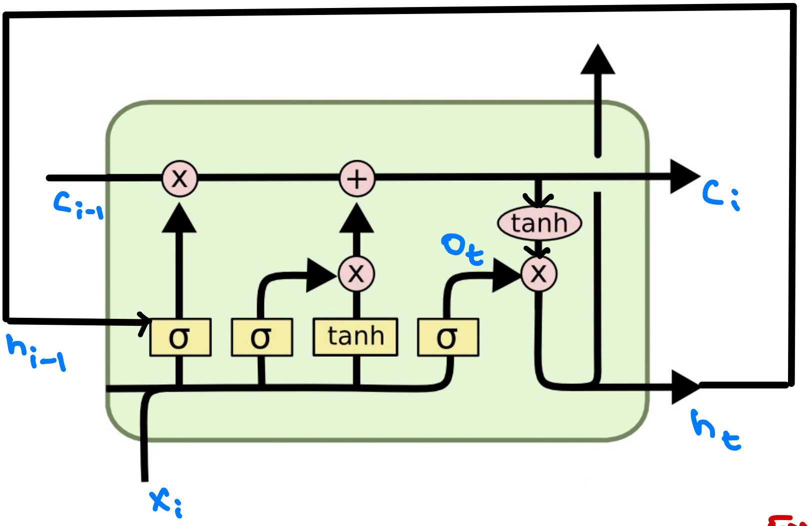
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Memory Channel



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

Output Gate



Ex:

$c_t = 0.09$

$\tanh(c_t) = 0.089$

$o_t = 1$

$h_t = 0.089$

Equations:

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

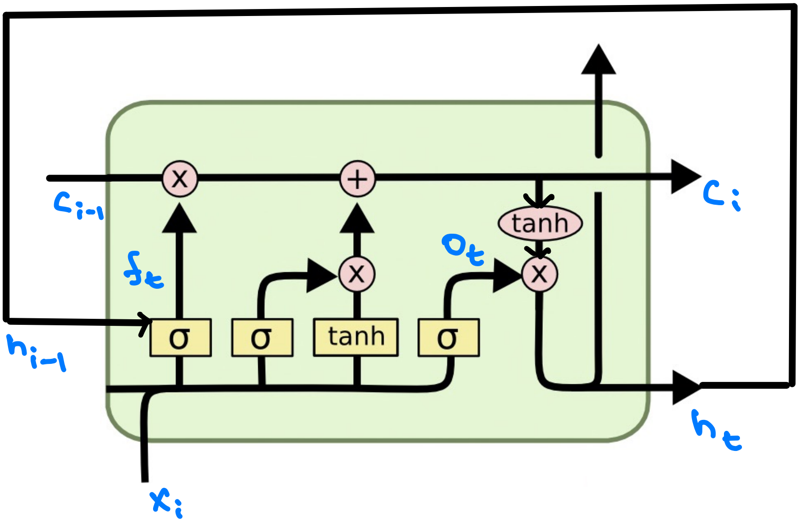
$$h_t = o_t * \tanh(c_t)$$

* Output gate takes memory channel (which captures context till the current word) and output of sigmoid NN as input.

* tanh converts memory channel output into [-1, 1] range.

* The output of output gate is sent to input of next word to memorize the previous context.

Let us assume word is 'HELLO'



At $t=0$, $x_i = H$, $C_i = 0.09$, $h_t = 0.089$

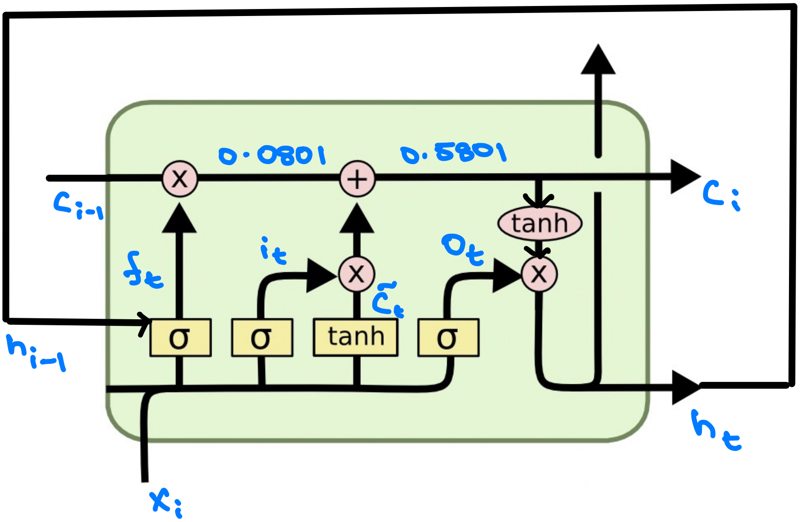
At $t=1$, $x_i = E$

* In the forgot gate, sigmoid NN takes h_t='H' and x_i='E' as input and checks for a pattern. If our data doesn't have 'HE' pattern it gives 0 as output. Similarly if our data contains pattern 'HE', it gives close to '1' as output.

* For example, if the pattern is 'XH' it gives output close to 0, because it's hard to find words with 'XH' pattern.

$f_t = 0.89$, $C_{i-1} = 0.09$

output $= 0.89 * 0.09 = 0.0801$



* In the input/Memory gate, we are learning the pattern 'HE' and adding it to the memory channel.

$$\text{output} = (H + HE)\,\text{memory}$$

$$= 0.0801 + (i_t * \tilde{C}_t)$$

$$= 0.0801 + (1 * 0.5)$$

$$= 0.5801$$

* In the output gate

$$h_t = O_t + \tanh(0.5801)$$

$$= 1 + 0.5227 = 1.5227$$

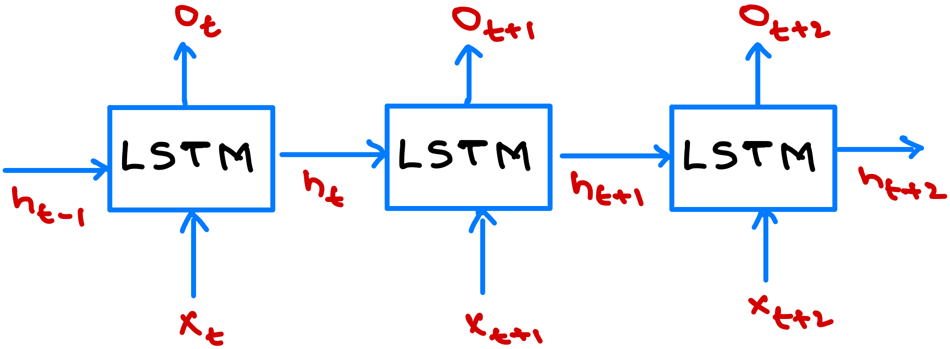
Example

* Assume we passed entire wikipedia data to the model. Input is 'HEZ'.
* LSTM tries to remember the previous context until input is letter 'E'.
* When 'Z' is given as input, forgot gate gives output close to zero, because words with patter 'HEZ' are hard to find.
* So, basically it forgots the previous context when 'Z' is given as input.

Key points

* Role and responsibilities of every sigmoid NN is different. We can't use same sigmoid NN in entire architecture.

RNN Architecture


O_t
O_{t+1}
O_{t+2}
LSTM
LSTM
LSTM
h_{t-1}
h_t
h_{t+1}
h_{t+2}
x_t
x_{t+1}
x_{t+2}