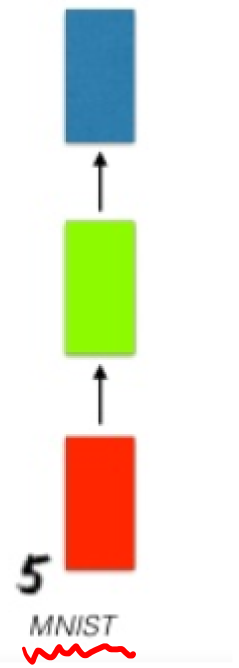
RNN Types

① One-to-One / Vanilla Mode: Vanilla mode of processing w/o RNN

fixed size i/p to fixed sized o/p.

eg: Image Classification



RNN Types

② **One-to-Many:**

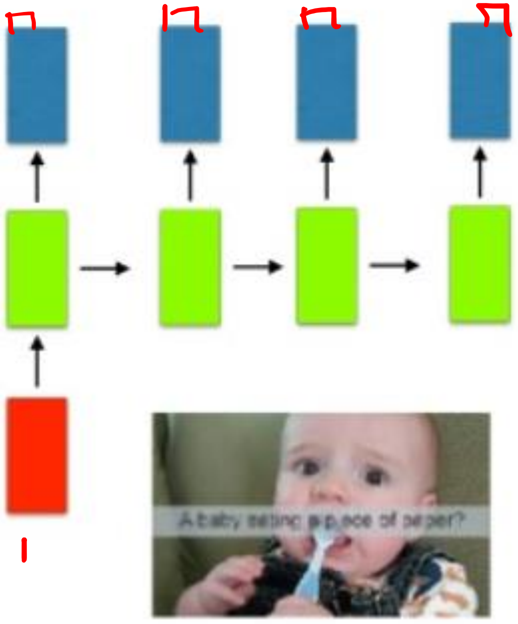
Sequence O/P

Image Captioning:

Input: An image

o/p: Sentence of words

Image Captioning



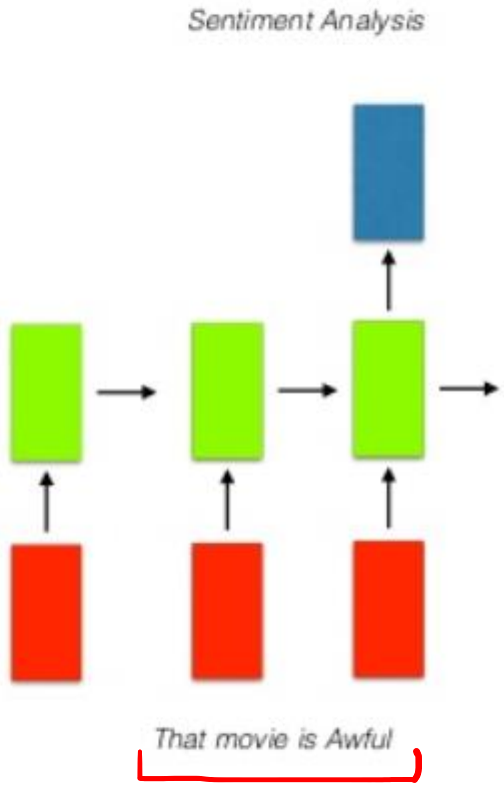
A Baby Eating a piece of paper

RNN Types

3 Many-to-one:

Sequence input

eg: Sentiment analysis where a given sentence is classified as expressing +ve or -ve Sentiment.



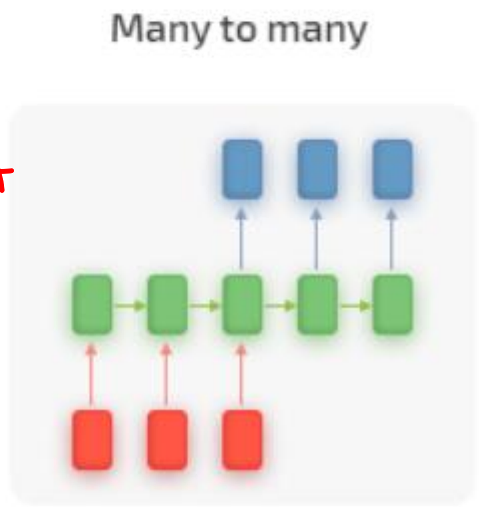
RNN Types

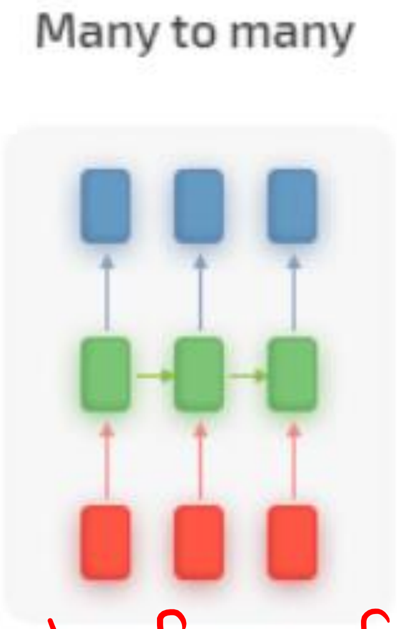
(4) Many to Many:-

Sequence input & Sequence Input

eg: Google Translate

Machine Translation

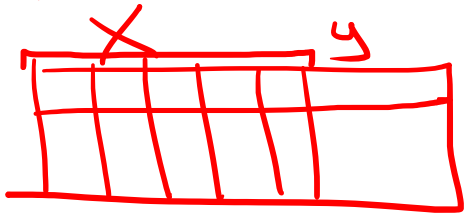




Video Classification where we wish to label each frame of the video.

Why RNN? Used popularly in NLP problems.

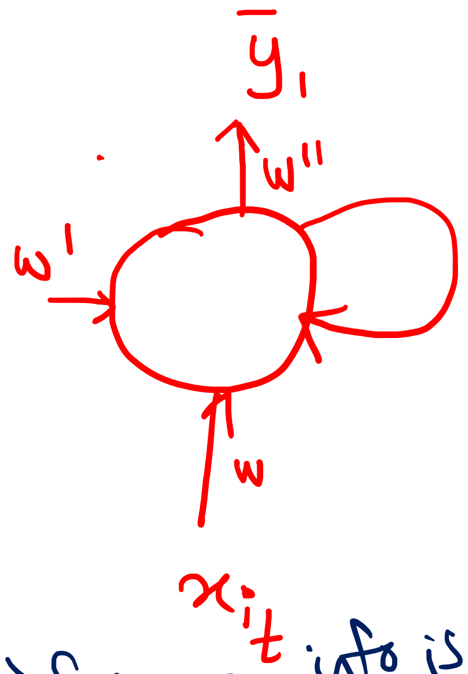
NLP: Input Data



X = text format → Preprocessing → + Bag of words (CountVectorizer)

processed X = Number format

Naive Baye's Algo.



0	love play
1	love play cricket
2	play fun

	love	playing	cricket	fun
0	1	1	0	0
1	1	1	1	0
2	0	1	0	1
Count:	2	3	1	1

Descending order of Count → Sequence info is LOST.

	play	love	cricket	fun
0	1	1	0	0
1	1	1	1	0
2	1	0	0	1

$\left.\begin{array}{l} \text{① Bag of words} \\ \text{② TF-IDF} \\ \text{③ WORD2VEC} \end{array}\right\}$ Sequence info is lost.

$$0 = \begin{array}{cccc} x_0 & x_1 & x_2 & x_3 \end{array} \; [1, 1, 0, 0]$$

$$1 = [1, 1, 1, 0] \quad \begin{array}{cccc} x_0 & x_1 & x_2 & x_3 \end{array}$$

$$2 = [1, 0, 0, 1] \quad \begin{array}{cccc} x_0 & x_1 & x_2 & x_3 \end{array}$$

Why RNN?

Time Series Prob

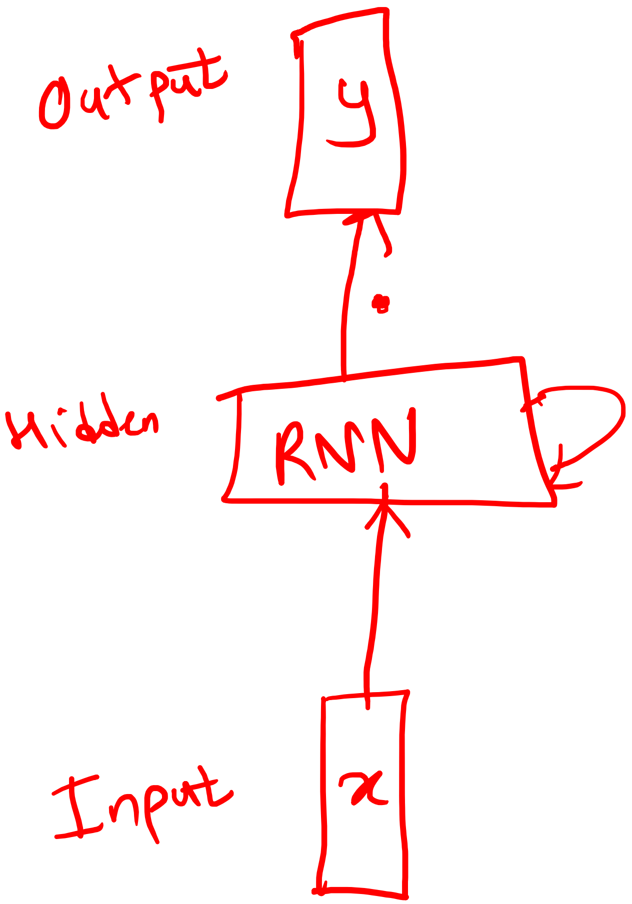
exponential smoothing

eg: Image Captioning,
Google Translate
Sentiment analysis for restaurant reviews
Amazon alexa
Google Assistant, Siri | Microsoft Cortana

Imp: Sequence info.

Summary

Simple form of Vanilla RNN

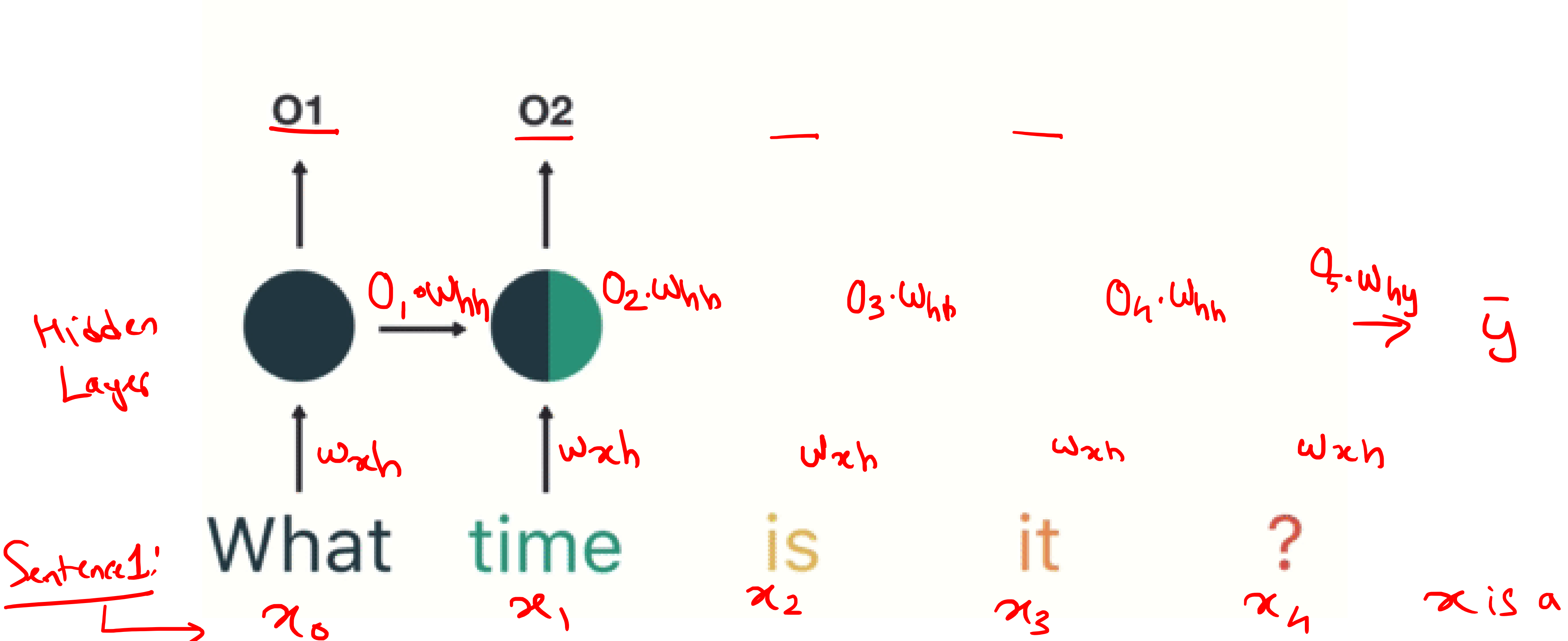


$$h_t = f_W(h_{t-1}, x_t)$$

$$\therefore h_t = \tanh(W_{hh} \cdot h_{t-1} + W_{xh} \cdot x_t)$$

$$\boxed{y_t = W_{hy} \cdot h_t}$$

Summary



x is a vector of length = 5

Sentence 2: This place is amazing. ⇒ x is a vector of length = 4.

x_0 x_1 x_2 x_3

Can I conclude, that x can have different lengths & still RNN can process it?

ANSWER: YES

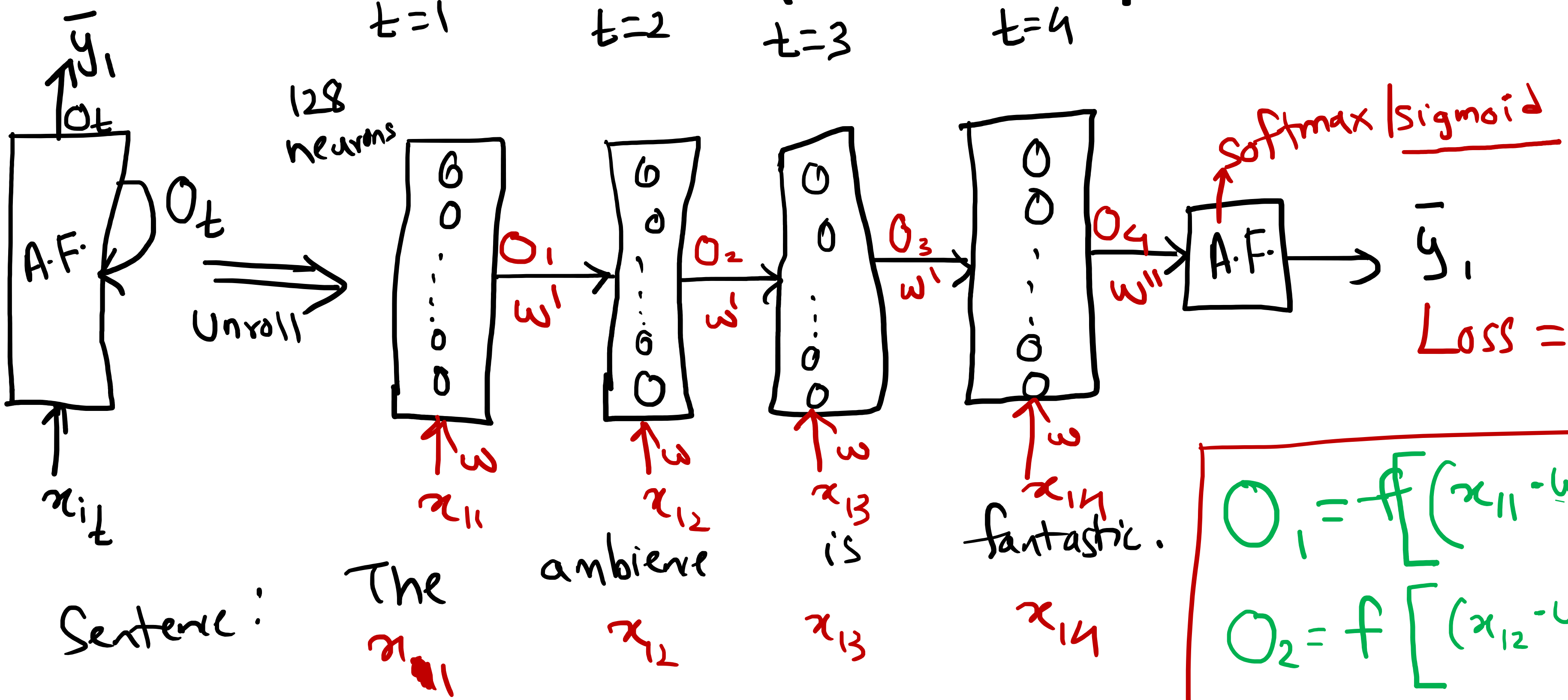
F.P. over time

RNN Architecture (Fwd Prop over Time)

$w \approx w_{xh}$

$w' \approx w_{hh}$

$w'' \approx w_{hy}$



Loss = $(\hat{y} - y)$ ↓ reduce

$$O_1 = f[(x_{11} \cdot w + b) + (O_0 \cdot w' + b')]$$

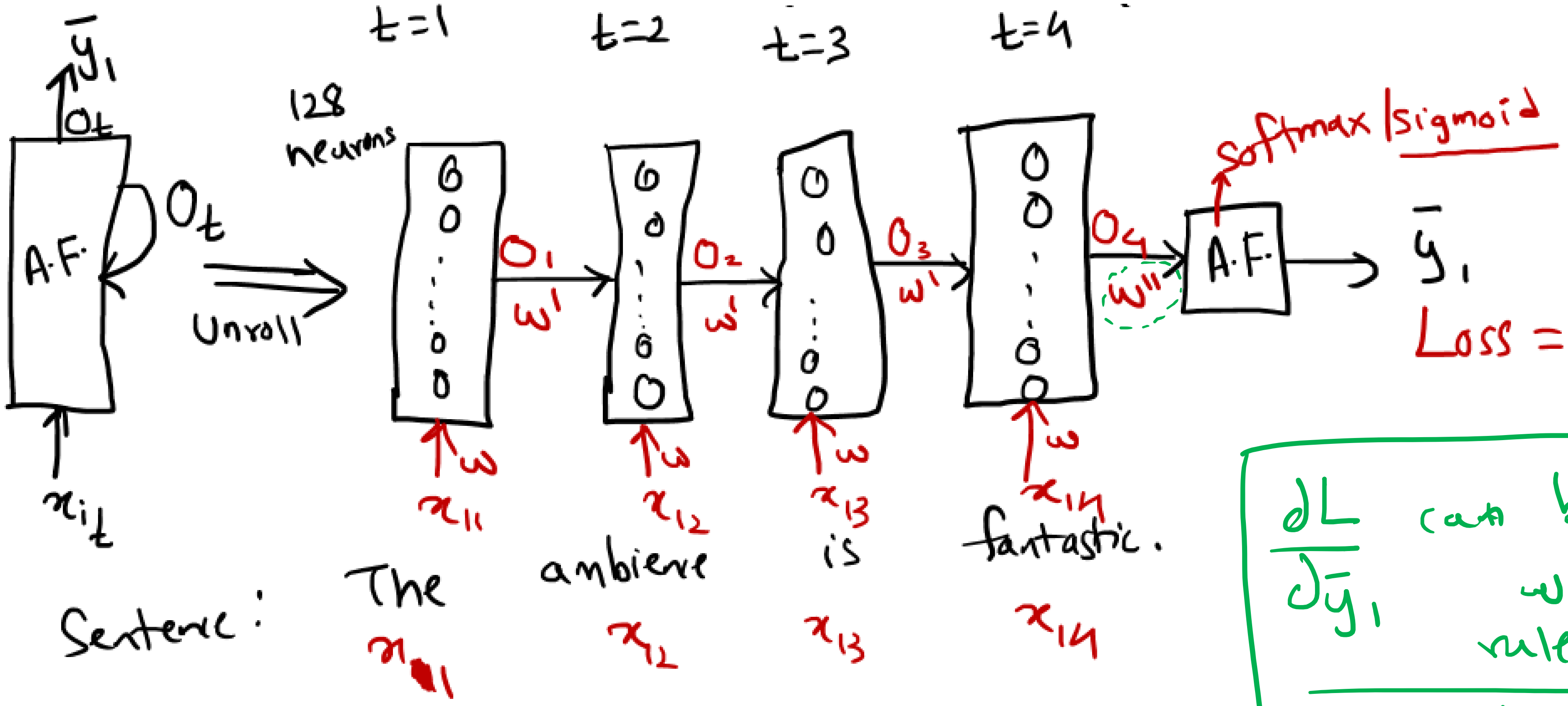
$$O_2 = f[(x_{12} \cdot w + b) + (O_1 \cdot w' + b')]$$

$$O_3 = f[(x_{13} \cdot w + b) + (O_2 \cdot w' + b')]$$

$$O_4 = f[(x_{14} \cdot w + b) + (O_3 \cdot w' + b')]$$

RNN Architecture (Fwd Prop over Time)

RNN Architecture (Back Prop over Time)



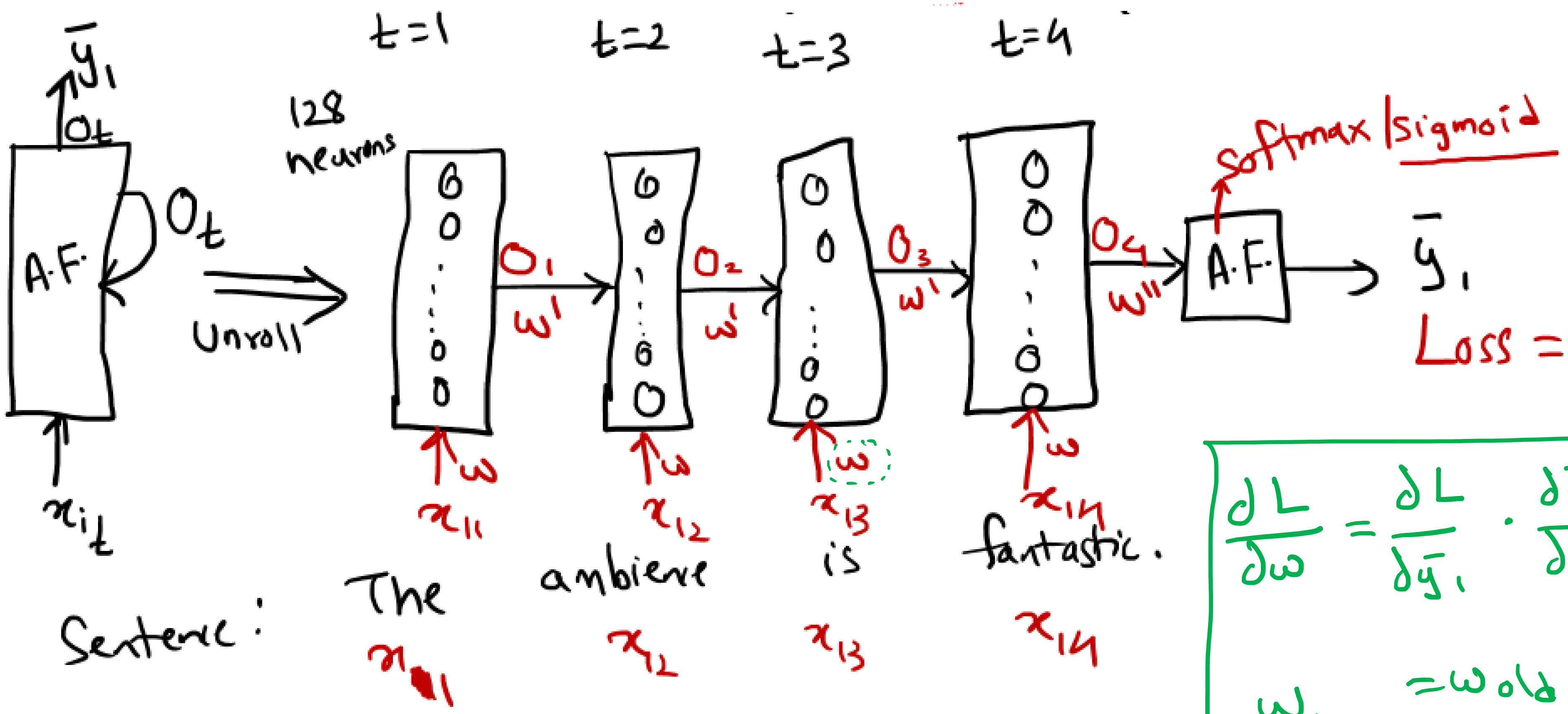
$Loss = (\hat{y} - y)$ ↓ reduce

$\frac{\partial L}{\partial \bar{y}_1}$ can be computed directly w/o using chain rule. ∴ SORTED.

$$\frac{\partial L}{\partial w''} = \frac{\partial L}{\partial \bar{y}_1} \cdot \frac{\partial \bar{y}_1}{\partial w''}$$

$$\therefore w''_{new} = w''_{old} - \eta \cdot \frac{\partial L}{\partial w''}$$

RNN Architecture (Back Prop over Time)



$Loss = (\hat{y} - y)$ ↓ reduce

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \bar{y}_1} \cdot \frac{\partial \bar{y}_1}{\partial O_4} \cdot \frac{\partial O_4}{\partial w}$$

$$w_{new} = w_{old} - \eta \cdot \left\{ \frac{\partial L}{\partial w} \right\}$$

Problems with RNN

① Vanishing Gradient Problem:

tanh() most comman A.F. in RNN → [-1, +1]

Der. of tanh → [0, 1]

② Exploding Gradient Problem: It is due to the higher assigned weights

Problems with RNN

Due to these problems, we cannot do a lot of B.P.,

∴ we cannot consider older states i.e. we cannot process a lot of sequence data.

∴ We need a better or Modified RNN Model.

Solution: 1997 : Long Short Term Memory (LSTM).

eg1: I live in Gujrat & I know Gujrati ⟹ RNN can process it.

eg2: I live in Gujrat & I am a businessman and I know ___ ?

⟹ RNN fails

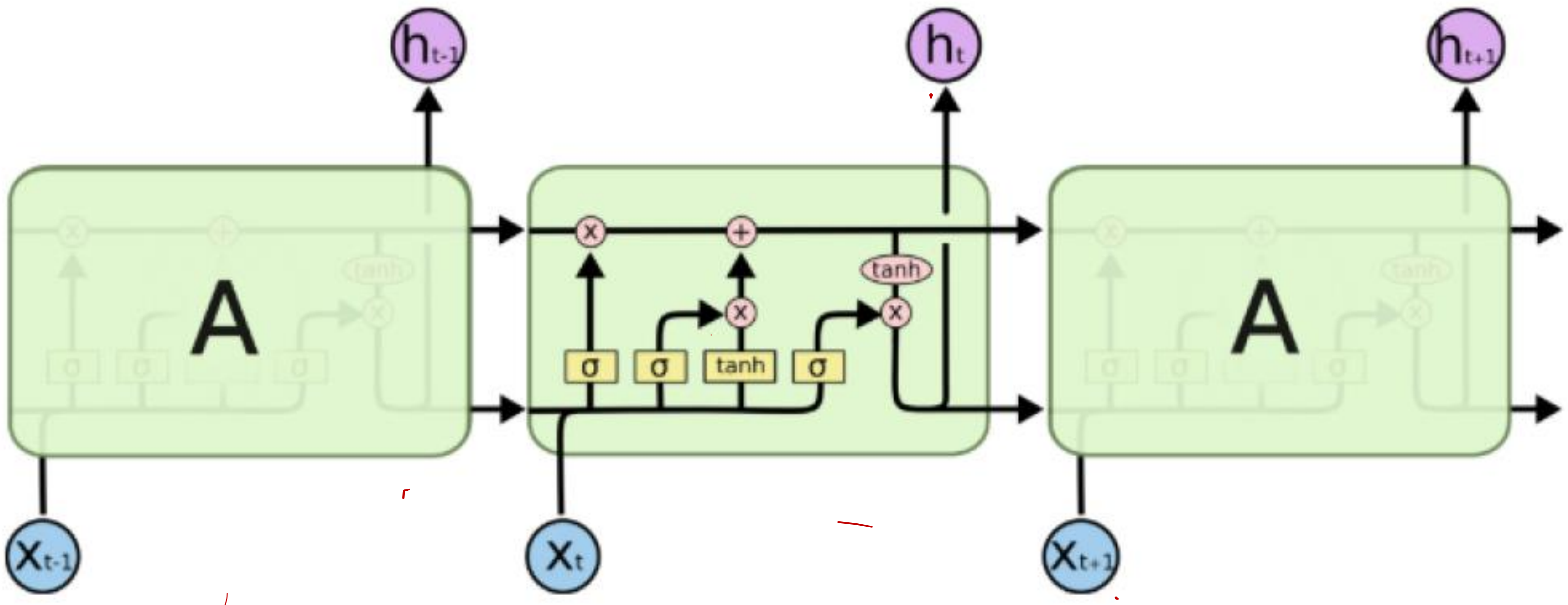
bcz the context is far away.

far

① We have not studied entire NLP course. ② LSTM requires a good background of NLP.

LSTM Architecture

③ LSTM is easy if you compare this to learn car driving.



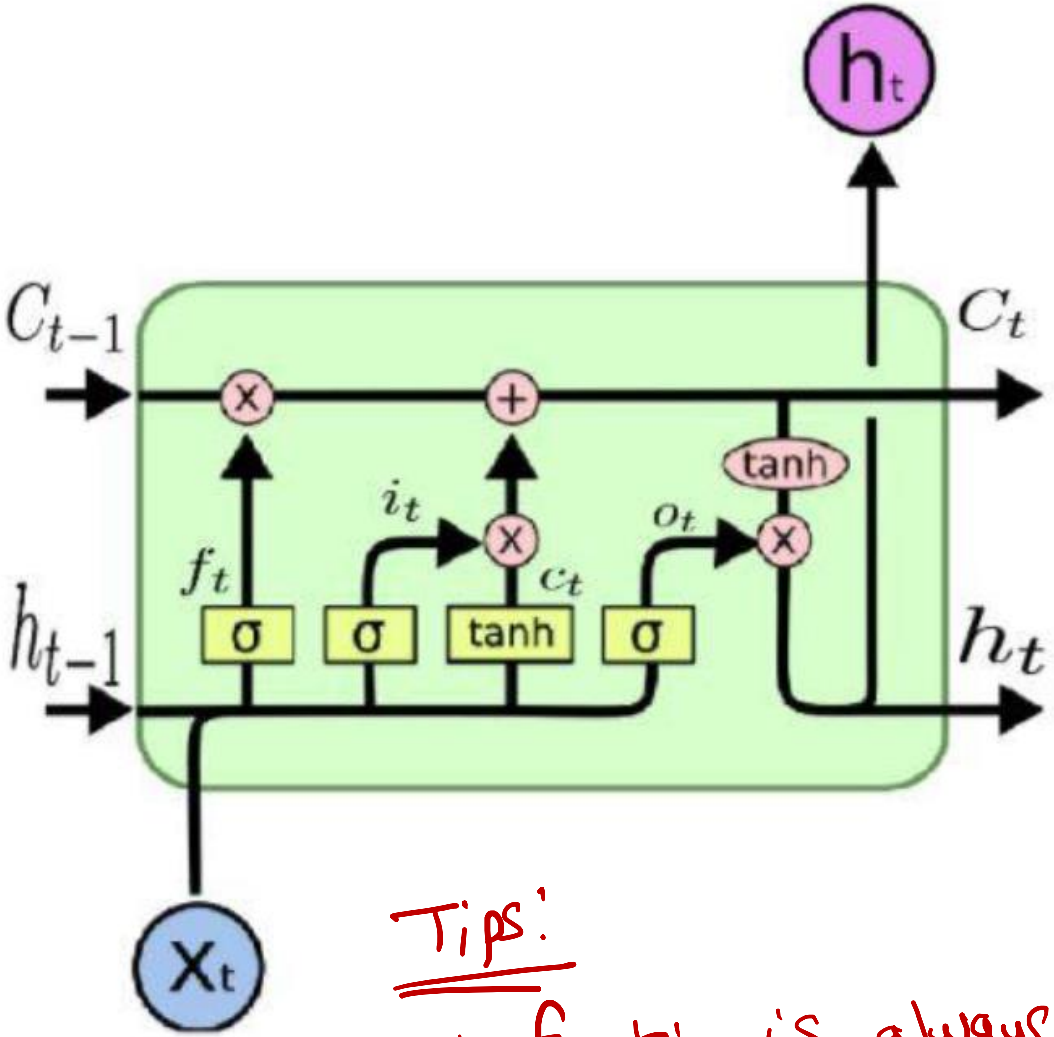
The repeating module in an LSTM contains four interacting layers.

① f : Forget gate: whether to erase the cell data

② i : Input gate: whether to write to the cell

③ g : Gate gate: How much to write to the cell?

④ o : Output gate: How much to reveal cell.



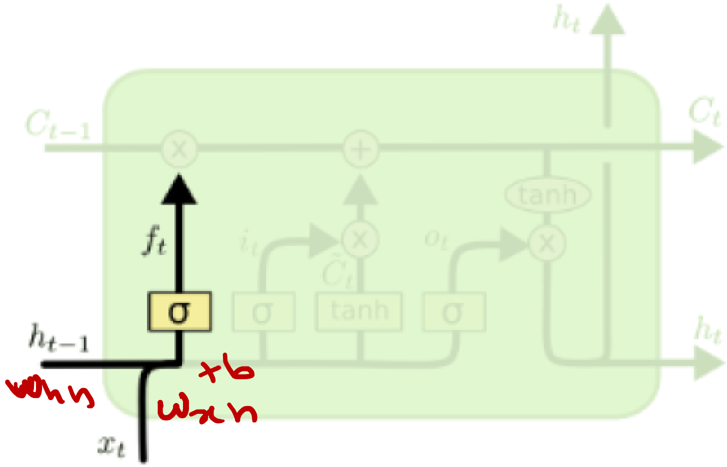
C : Cell State
h : hidden State

f : forget gate
i : input gate
o : output gate

Tips:
① Sigmoid function is always used to filter out information.
② tanh function is used for adding new or modifying existing information.

Forget Gate

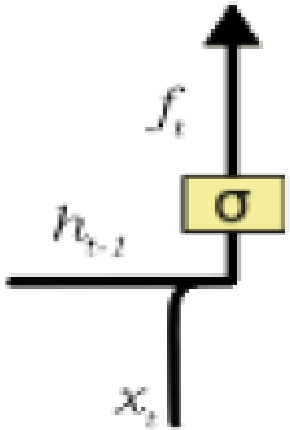
After getting o/p from h_{t-1} i.e. prev. state, Forget gate helps me to take decision about what must be removed from h_{t-1} & thus keep only relevant info.



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

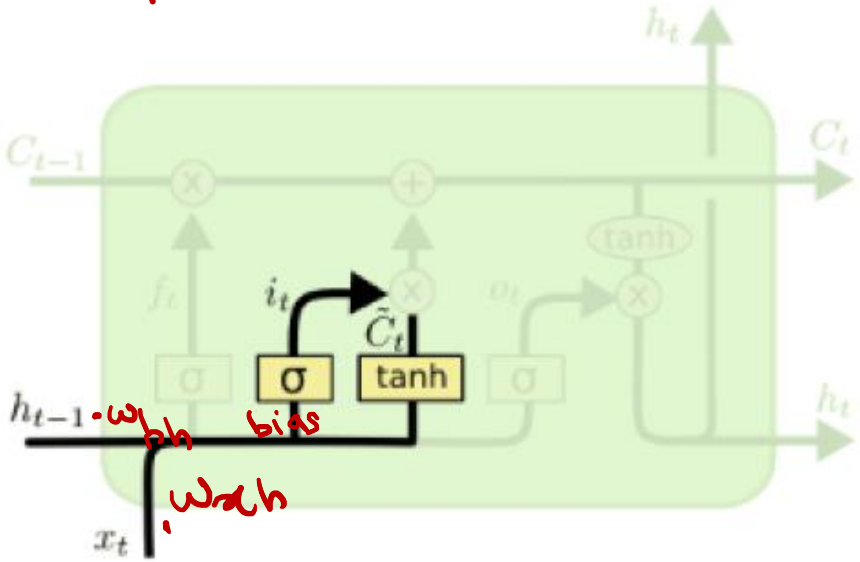
Forget Gate Example

Bob is a nice person. Dan on the other hand is evil.



Input Gate

In input gate, we decide to add new info from present input to our present cell state scaled by how much we wist to add them.



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

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

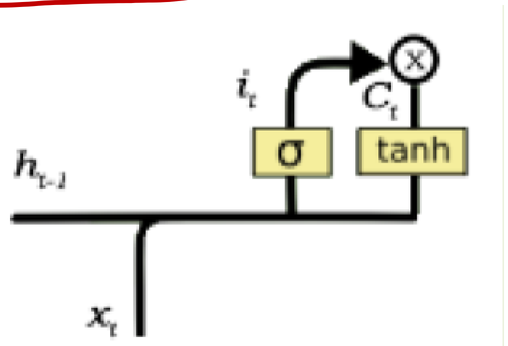
Sigmoid layer decides which value shud be updated & tanh create a vector for new info to be added to present cell state.

Input Gate Example

Bob knows swimming. He told me $\boxed{\text{over the phone}}$ that he had served the navy for 4 long years.

ignored

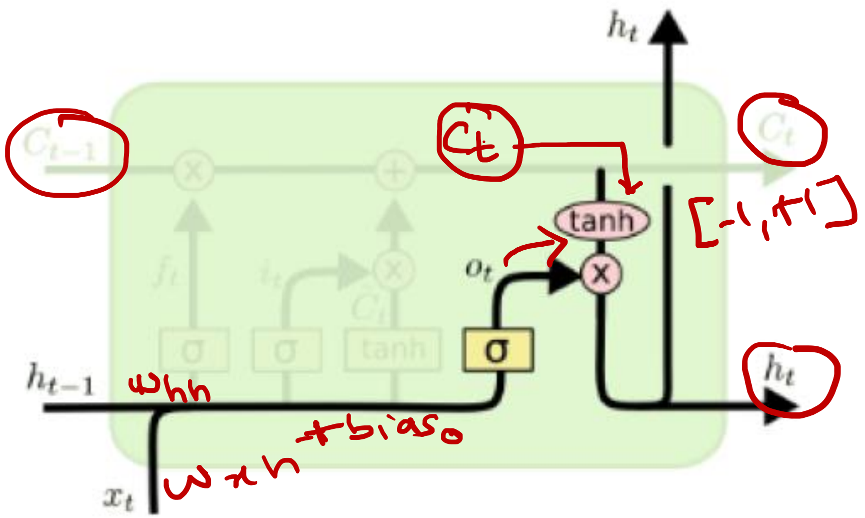
added to cell state



$$\begin{bmatrix} 1 \\ 0.002 \\ 1 \\ 0.1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 5 \\ 6 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \checkmark \\ 0.004 \\ 6 \checkmark \\ 0.3 \end{bmatrix}$$

Output Gate

we decide finally what to output from our cell state (which will be done by sigmoid).



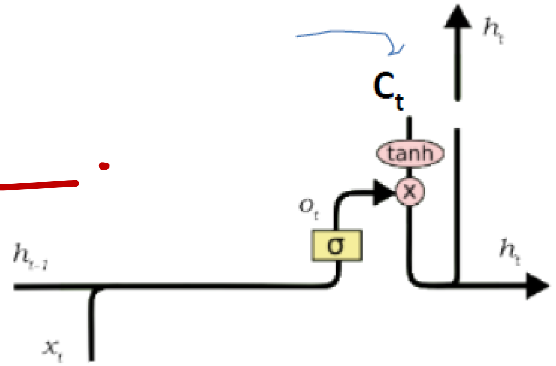
$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

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

Output Gate Example

Bob fought single handedly with the enemy and died for his country. For his contributions brave ___?___.

adjective
i.e. to
describe a noun



Additional Resource

- https://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/
- https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21
- https://medium.com/@aidangomez/let-s-do-this-f9b699de31d9