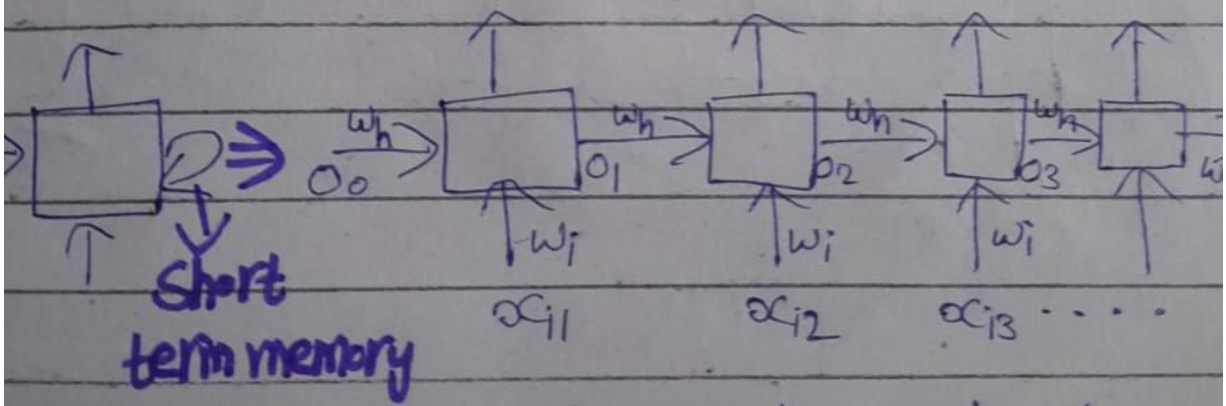
⇒ LSTM RNN (Long Short Term Memory RNN):

- **Problem with Simple RNN:**

  RNN → Can't handle Long term dependencies → **Vanishing Gradient Problem**

- **Basic Representation of LSTM RNN:**

Long Term memory (which remember the context till when it is required) → Memory cell → (after that remove it)

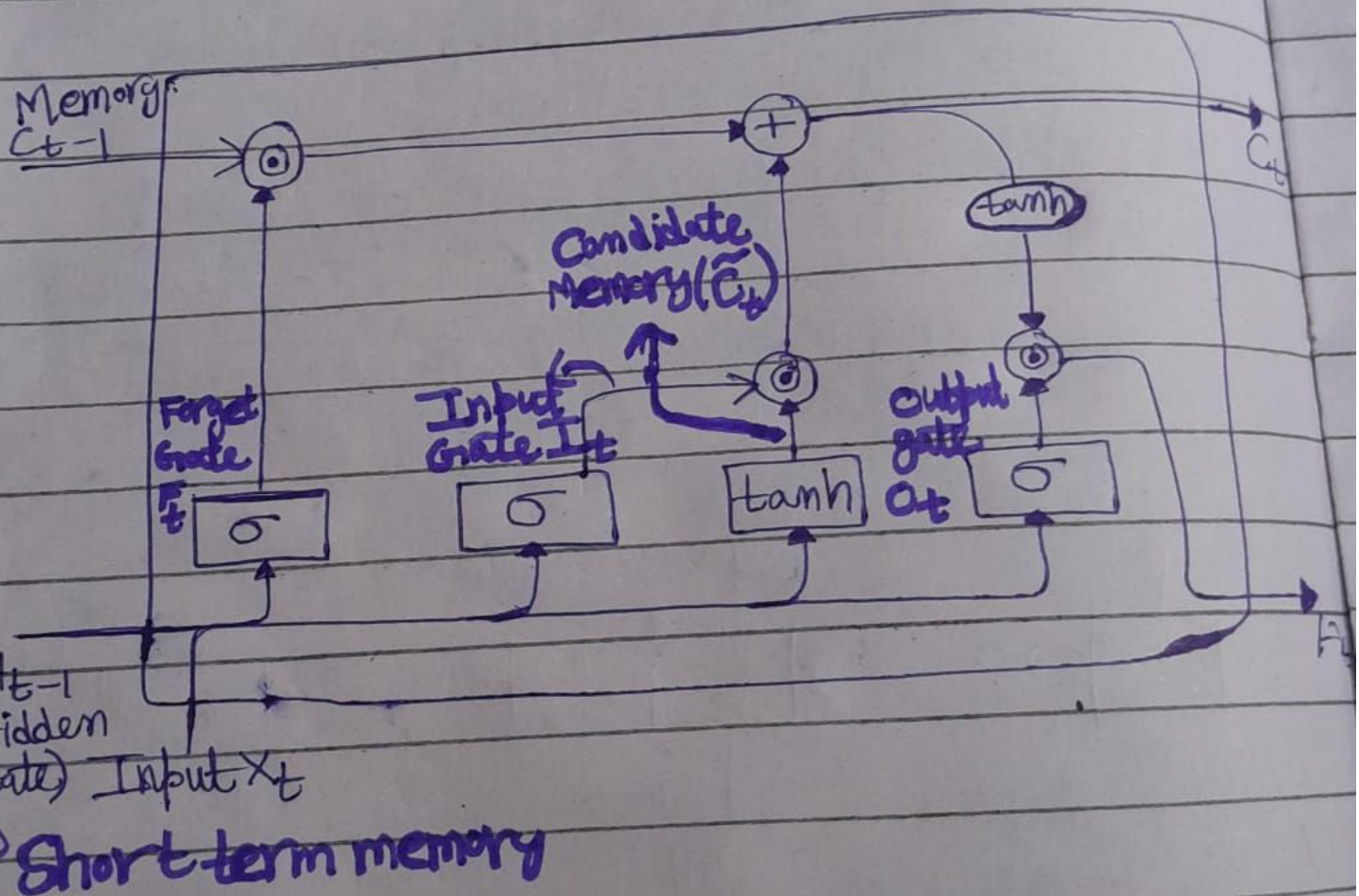


In LSTM RNN, we have short term memory like Simple RNN but we also have Long Term memory which remember the context till it

is required then drop it

LSTM RNN — Long Term Memory
LSTM RNN — Short Term Memory

⇒ LSTM Architecture:



⇒ Neural Network Layer

⇒ Vector transfer

⇒ Copy

$\rightarrow\!\!\!\!\rightarrow$ ⇒ Concatenate

e.g Combining two vectors

$h_{t-1} = [1\ 2\ 3]$

$x_t = [4\ 5\ 6]$

So concatenate $[1\ 2\ 3\ 4\ 5\ 6]$

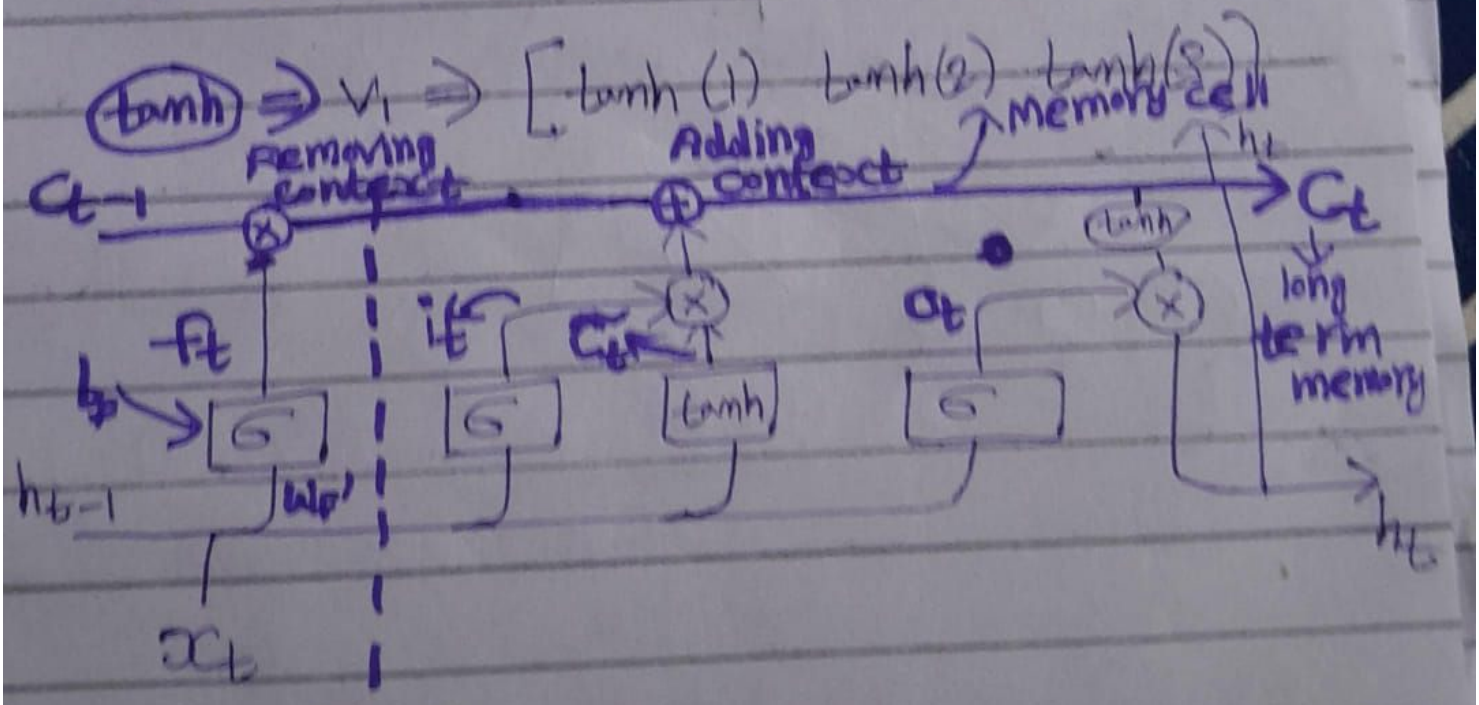
⊙ ⇒ Pointwise operation

Let $V_1 = [1\ 2\ 3]$, $V_2 = [4\ 5\ 6]$

⊗ ⇒ $[1\times4\ \ 2\times5\ \ 3\times6] \Rightarrow [4\ 10\ 18]$

⊕ ⇒ $[1+4\ \ 2+5\ \ 3+6] \Rightarrow [5\ 7\ 9]$

tanh ⇒ $V_1 \Rightarrow [\tanh(1)\ \ \tanh(2)\ \ \tanh(3)]$



Forget gate f_t

So

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

Forget Gate:

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

Text

Next word to predict

y_{15}

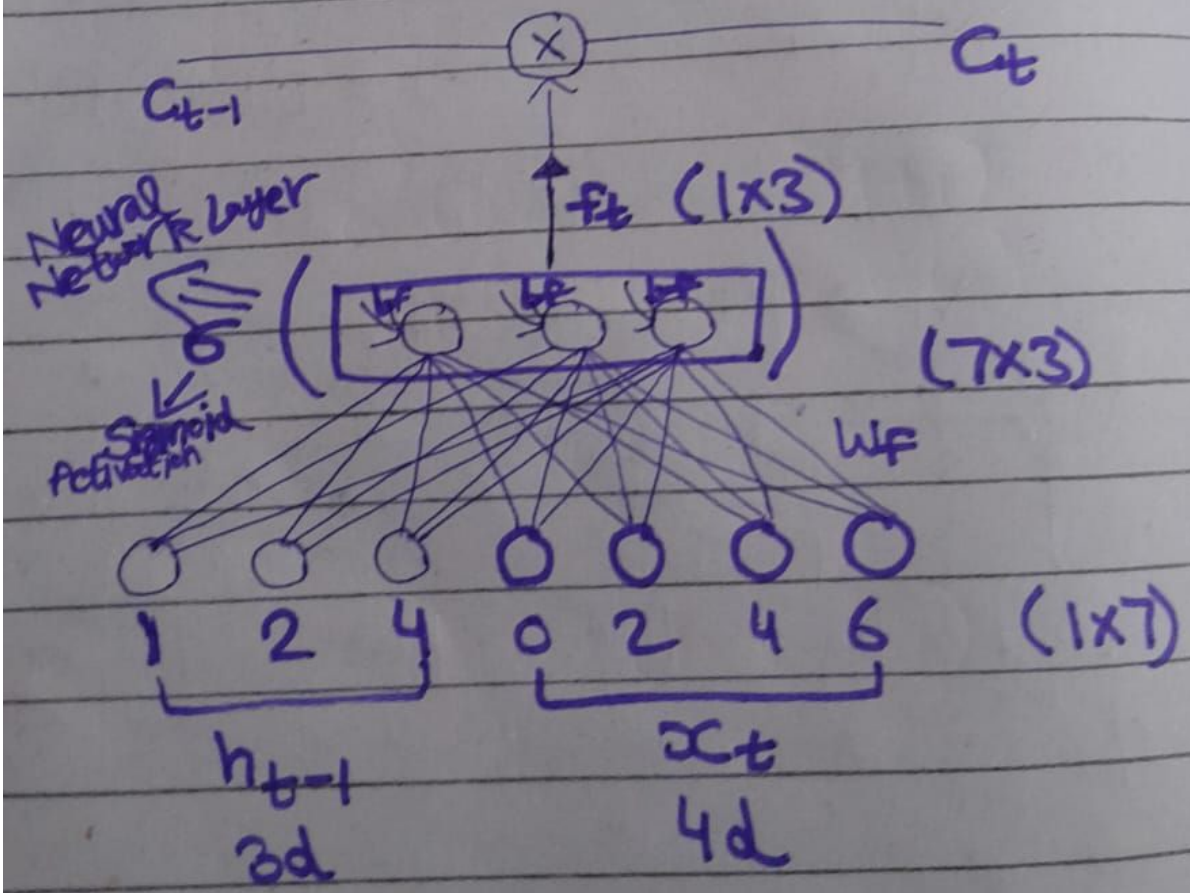
x_{11} x_{12} x_{13} x_{14}

vector of word

[0 2 46] [4512] ... ...

x_t x_{t+1}

Let $h_{t-1} = [1\ 2\ 4]$ (3 dimensional)



Now Let

$C_{t-1} = [6\ 8\ 9]$

① Let $f_t = [0\ 0\ 0]$

$C_t = C_{t-1} \otimes f_t = [6\ 8\ 9] \otimes [0\ 0\ 0]$

$\boxed{C_t = [0\ 0\ 0]} \Rightarrow$ Removed all the previous context

② let $f_t = [1\ 1\ 1]$

$C_t = [6\ 8\ 9] \otimes [1\ 1\ 1]$

$\boxed{C_t = [6\ 8\ 9]} \Rightarrow$ No context is removed

③ let $f_t = [0.5, 1, 0.5]$

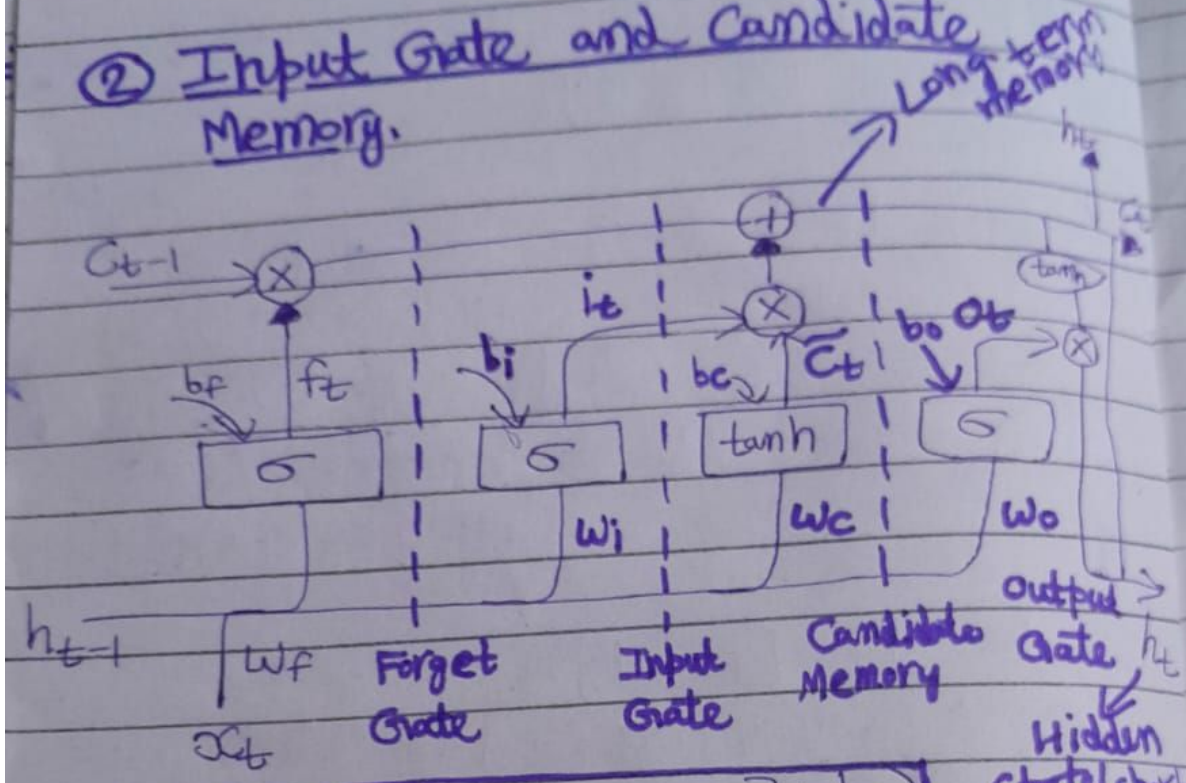
$C_t = [6\ 8\ 9] \otimes [0.5, 1, 0.5]$

$\boxed{C_t = [3\ 8\ 4.5]} \Rightarrow$ Some previous context is removed

Conclusion:

"Based on the context Forget gate will forget some information or not forget some information."

② Input Gate and Candidate Memory.



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

$$\tilde{C}_t = \tanh\left(W_c[h_{t-1}, x_t] + b_c\right)$$

$$\text{Final Input} = i_t \otimes \tilde{C}_t$$

$$C_t = C_{t-1} \oplus (i_t \otimes \tilde{C}_t)$$

→ Final Input

Conclusion:

If any information needed to add in memory it will get added.

So, till now

$$C_t = \underbrace{C_{t-1} \otimes f_t}_{\text{Removing Context (Forget gate)}} \oplus \underbrace{i_t \otimes \tilde{C}_t}_{\text{Adding Context (Input gate and candidate memory)}}$$

3) Output gate:

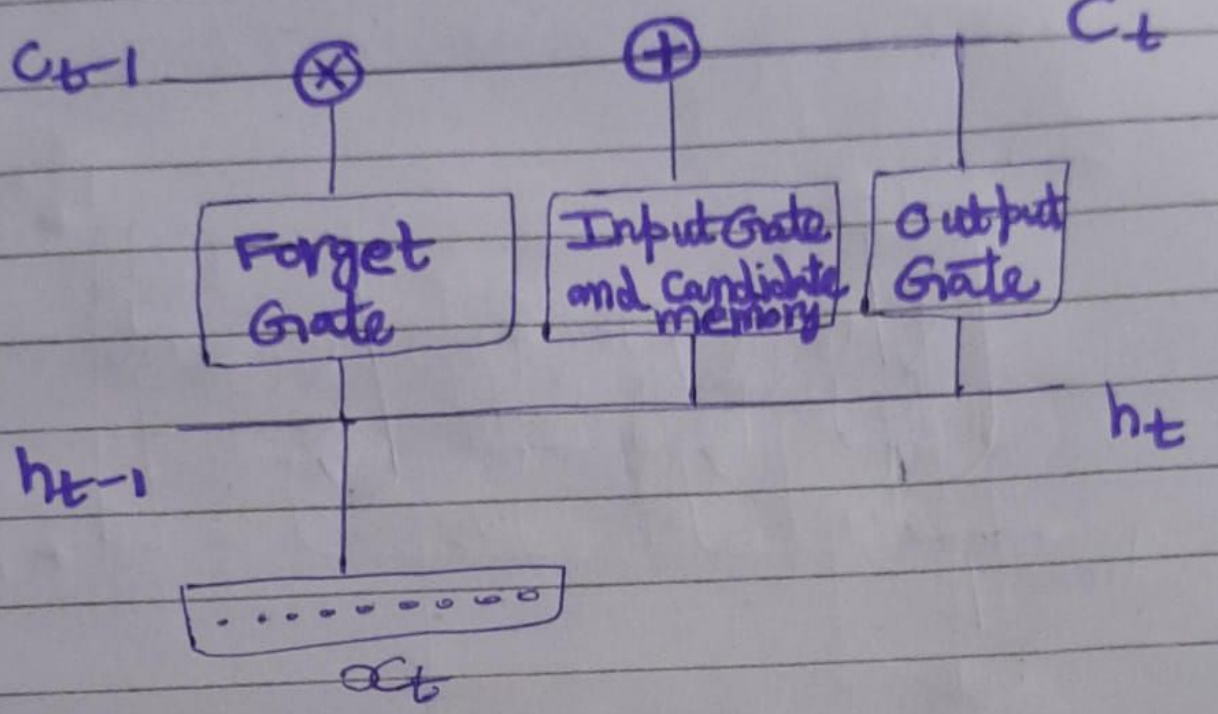
$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = O_t \otimes \tanh(C_t)$$

Training Data with LSTM RNN.

Text Paragraph	Output (good/bad)
I went to Restuarant and ordor burger. The burger looked tasty and crispy But burger is not good for health. It has lot of fats, cholestrol But this burger was made with whey protien and only vegetables were used so it was good	1

Training:



Step1: Words → Vectors → Embedding Layer

- Word2vec can be used

Let Word2vec (3 dimensions)

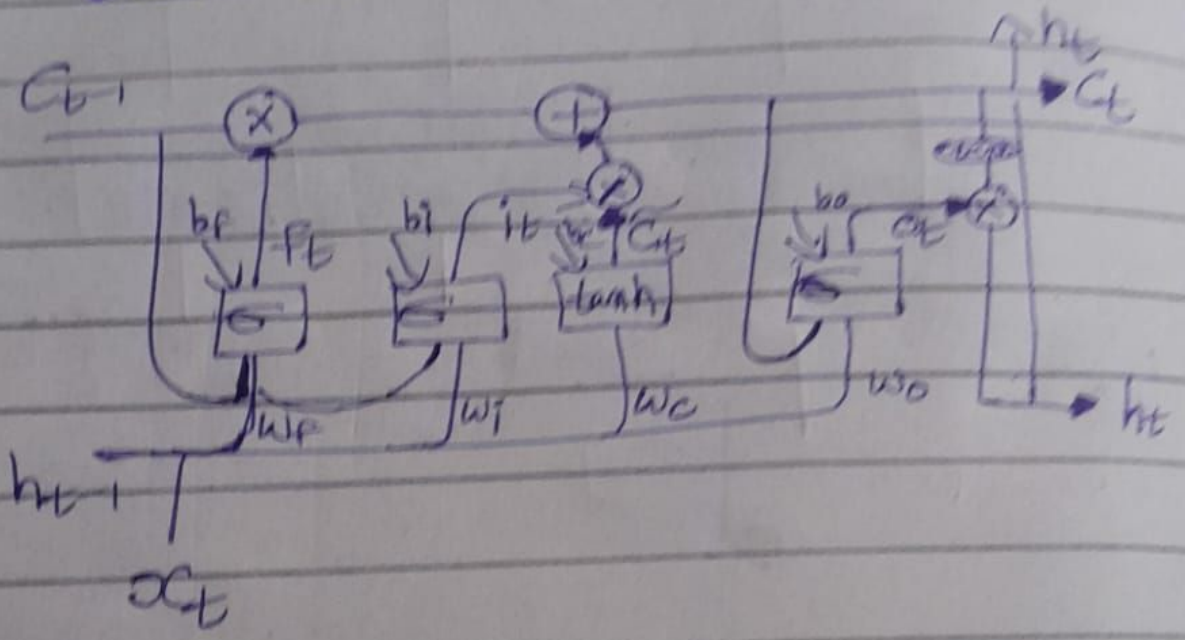
Good bad Healthy ← Black Box

Tasty [0.9 0.0 0.1] ← 3d

- We will forget the non-important context through forget gate and add further context thraugh Input gate and candidate memory.

⇒ Variants of LSTM RNN:

LSTM variants introduced by Gers & Schmidhuber (2000):



Eqn

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

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

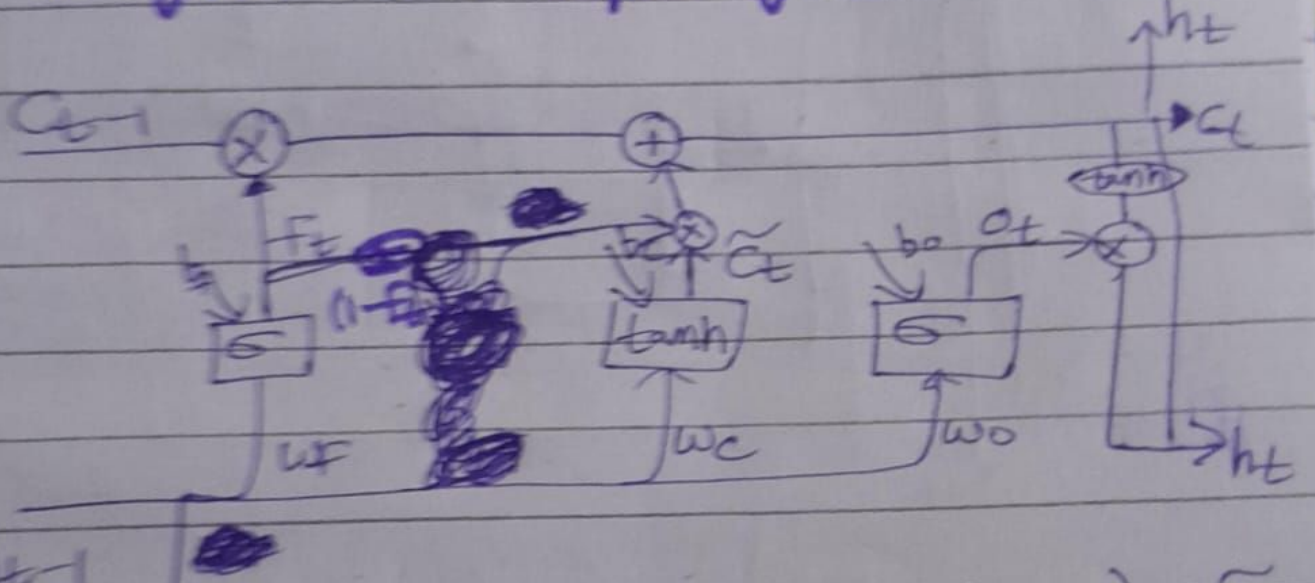
$$O_t = \sigma(W_o [C_t, h_{t-1}, x_t] + b_o)$$

Connections → From memory cell to Forget gate, input gate and output gate ⇒ Peephole Connections

Peephole Connections:

We let the gate layers look at the cell state

Another variant → Coupling Forget and input gates:



$$C_t = f_t \otimes C_{t-1} \oplus (1 - f_t) \otimes \tilde{C}_t$$

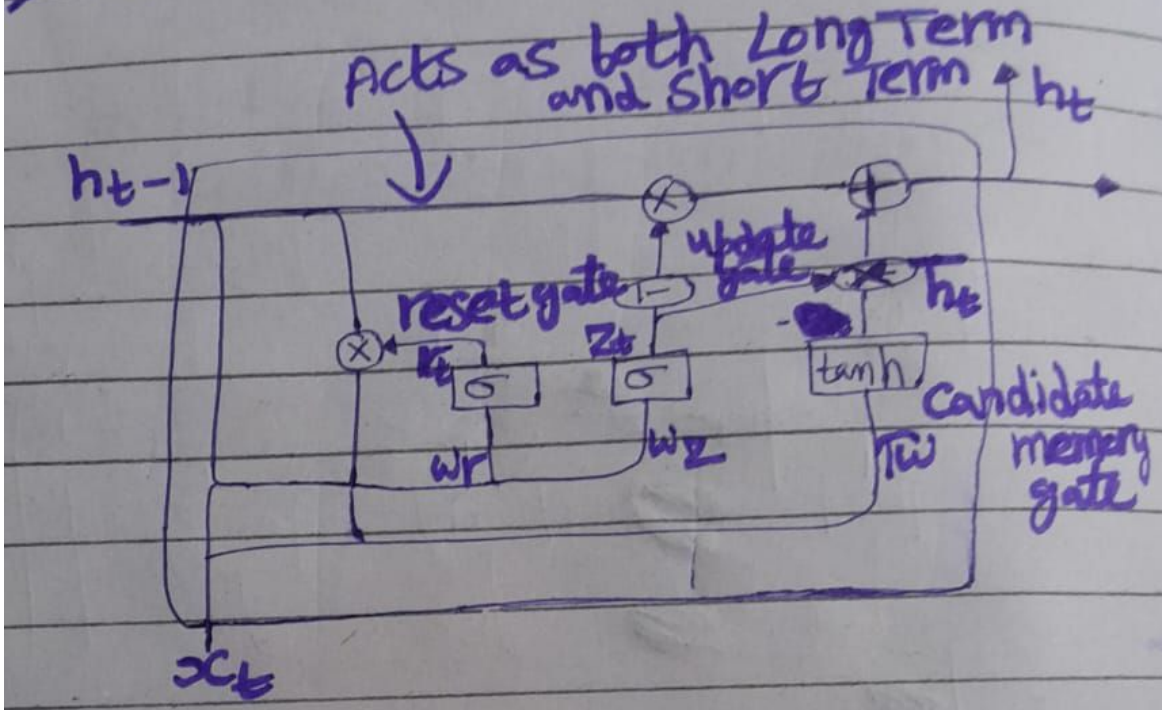
⇒ Instead of decididing seperately what to forget and what to add new info we make this decision together

Conclusion:

We only forget when we are going to input something in its place or we only add/input new

values only when we forget.

⇒ GRU (Gated Recurrent unit)



- In LSTM RNN we have too many number of gates, weights and bias and complex architecture due to which training time increase because there are many trainable parameters.

Update gate

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

Reset gate

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

Temporary Hidden State

$$\tilde{h}_t = \tanh(W \cdot [r_t \circledast h_{t-1}, x_t])$$

$$h_t = (1 - z_t) \otimes h_{t-1} \oplus z_t \otimes \tilde{h}_t$$

Reset gate (r_t):

Responsible for resetting / forgetting some information from ~~h_t~~ h_{t-1}

Update Gate (z_t):

"It defines what context info needs to be added"

↓ Depends on

Candidate Hidden State [Current Context]