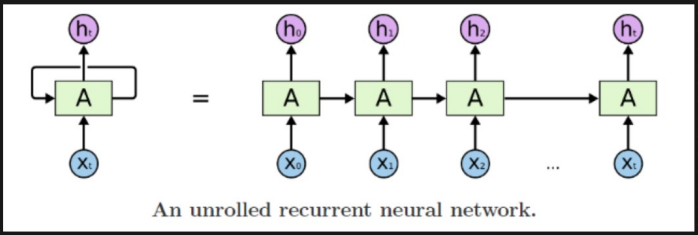
LSTM RNN [Long Short Term Memory RNN]

RNN ⟶ Long Term Dependencies ⟶ Vanishing Gradient Problem

1. RNN ⟶ Problem ? ✓
2. Why LSTM RNN ? ✓ Basic Representation.
3. How LSTM RNN Works
   - Long Term Memory ✓
   - Short Term Memory ✓
4. LSTM Architecture
5. Working of LSTM RNN

Problems With RNN → Long Term Dependency



An unrolled recurrent neural network.

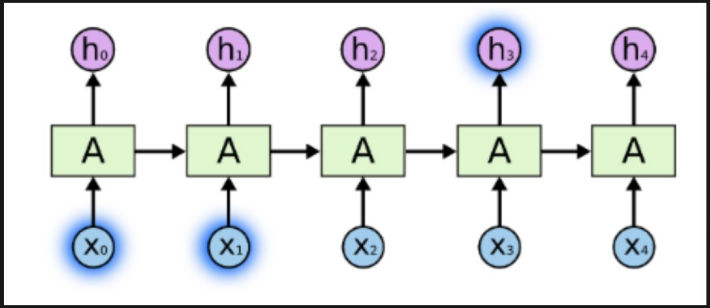
Vanishing Gradient Problem

Task:

Next Word In a Sentence

The color of the sky is blue

further context



Gap is less

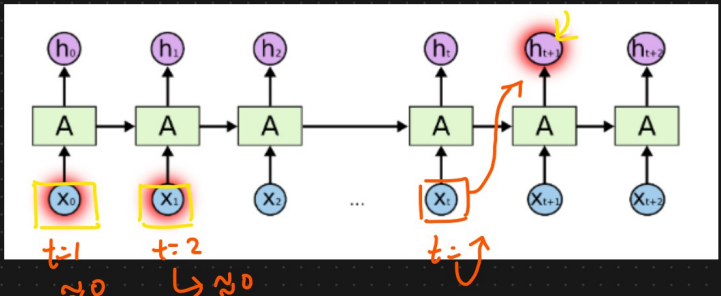
Huge gap O/p ← Context

I grew up in India .... I speak fluent

Context

Name of language

further Context



0-0.25

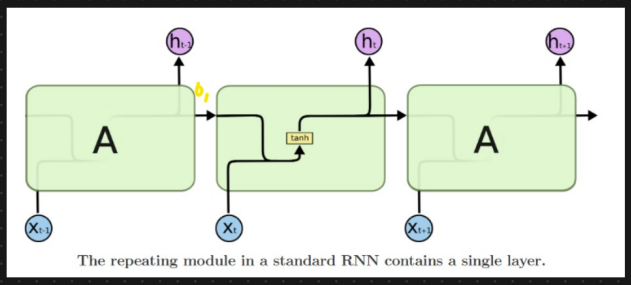
0-1

Huge Gap → Long Term Dependency

RNN → Long Term Dependency → Vanishing Gradient Problem

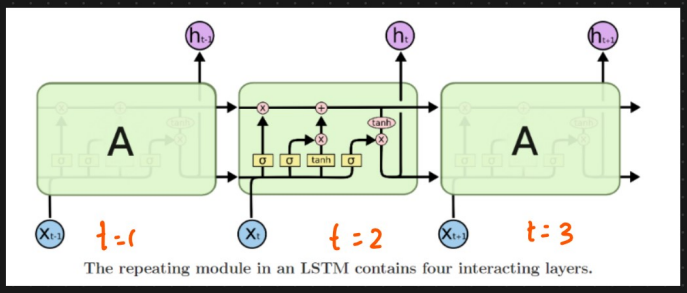
Chain Rule → ≈0.

Basic Representation of RNN And LSTM RNN

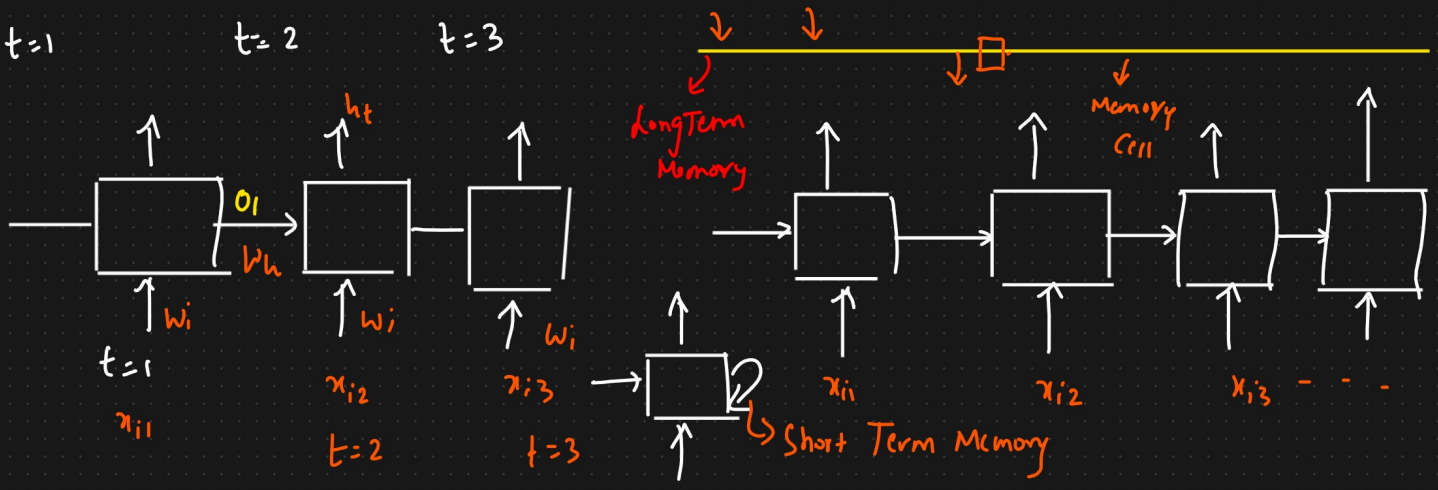


The repeating module in a standard RNN contains a single layer.

LSTM RNN



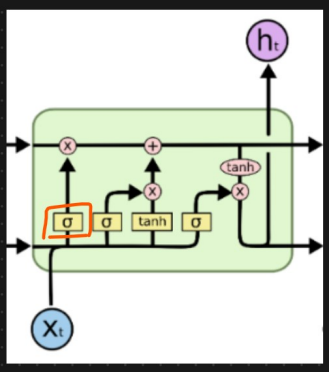
The repeating module in an LSTM contains four interacting layers.



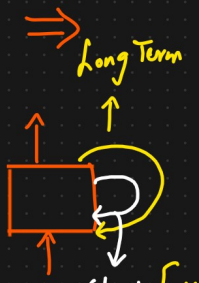
LSTM RNN → Long Term Memory
LSTM RNN → Short Term Memory

Conveyance But : Luggages

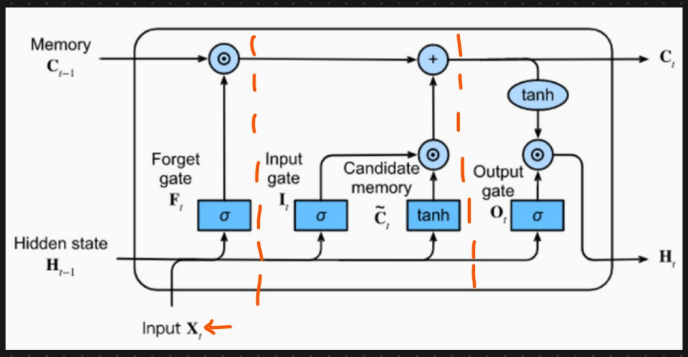
LSTM Architecture



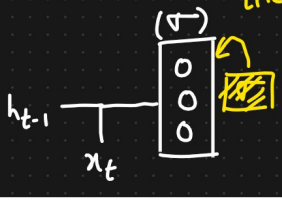
Basic Architecture

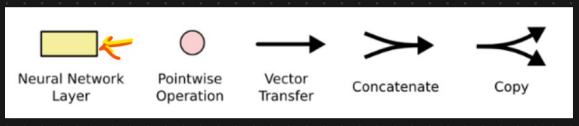


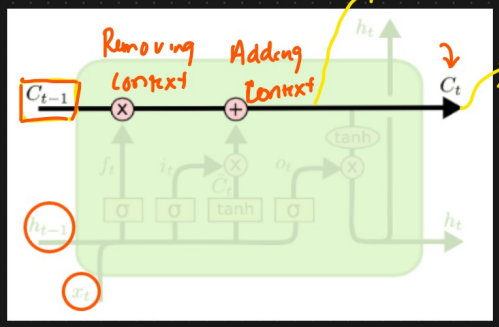
LSTM RNN



{Neural Nw Layer}





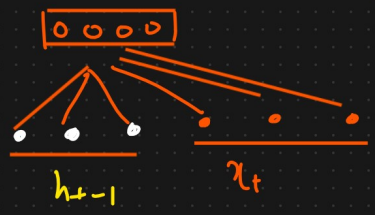


Long Term Memory ⊗ ⊕ tanh

Combining 2 vectors

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

$x_t = [2 \quad 3 \quad 4]$



Duplicate copy

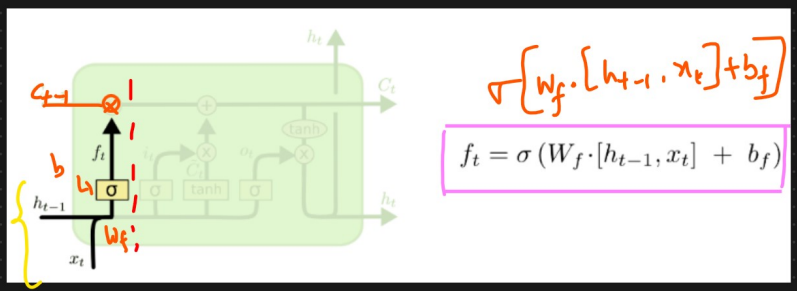
V1 $[1 \quad 2 \quad 3]$ ← tanh

V2 $[4 \quad 5 \quad 6]$

⊗ ⇒ [4 10 18]

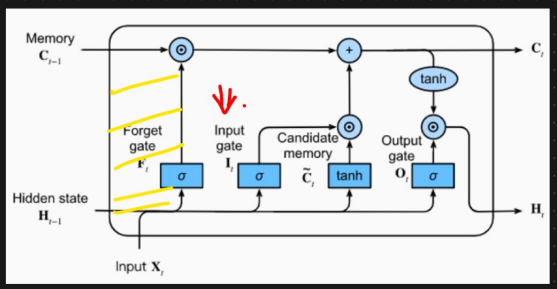
⊕ ⇒ [5 7 9]

tanh ⇒ [tanh(1) tanh(2) tanh(3)]



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

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



Forget Gate

h_{t-1} = Hidden state of previous time stamp

x_t = Word passed as i/p in the current time stamp

Text — Next Word

$x_{11} \quad x_{12} \quad x_{13} \quad x_{14}$ — y_{15}

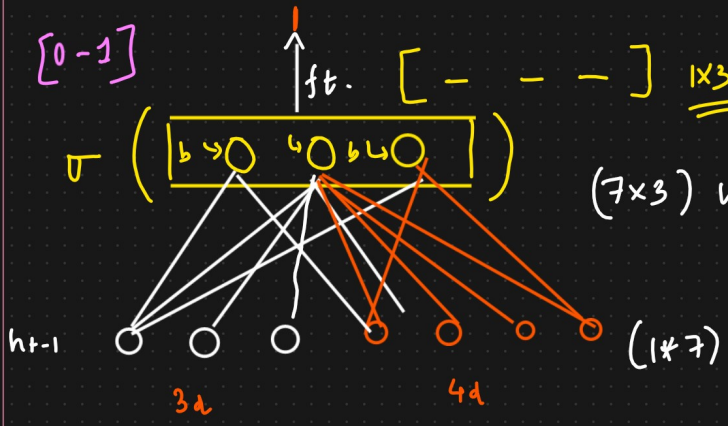
$[0 \quad 2 \quad 4 \quad 1]$ $[4 \quad 5 \quad 12]$ - - - - -

x_t x_{t+1}

Memory Cell

h_{t-1} $[1 \quad 2 \quad 4]$ 3d

C_{t-1} $[4 \quad 2 \quad 1]$ 3d ⇐



$$f_t = [0 \quad 0 \quad 0]$$ ← Complete Sentence

Context changed

① $C_{t-1} = [6 \quad 8 \quad 9]$ ⊗ $[0 \quad 0 \quad 0]$

$= [0 \quad 0 \quad 0]$ ← Removing all the previous Context

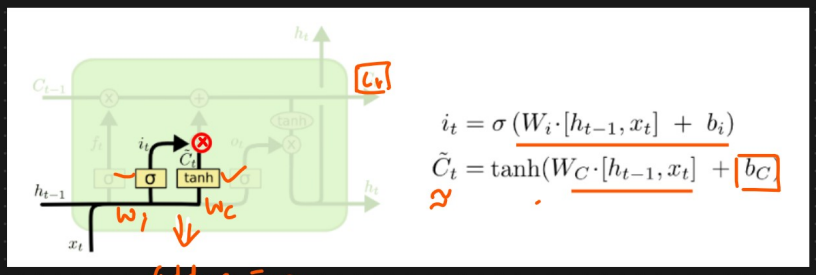
$f_t = [1 \quad 1 \quad 1]$

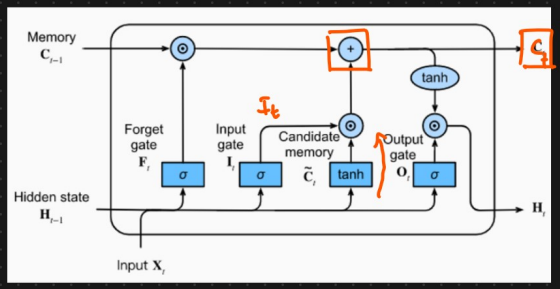
② $C_{t-1} = [6 \quad 8 \quad 9]$ ⊗ $[1 \quad 1 \quad 1] = [6 \quad 8 \quad 9]$

③ $C_{t-1} = [6 \quad 8 \quad 9]$ ⊗ $[0.5, 1, 0.5] = [3 \quad 8 \quad 4.5]$

Conclusion : Based on the context → Forget gate will let go some information or will not let go some Info {Forgetting}.

② Input Gate And Candidate Memory

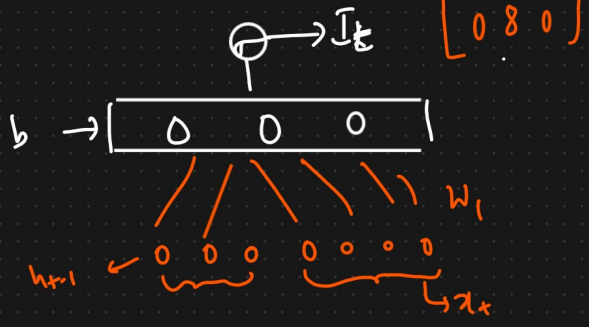


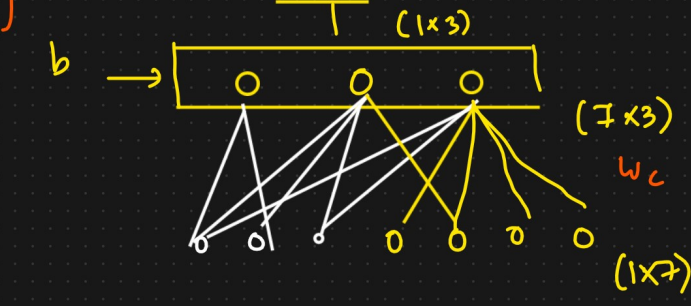


Adding Info

$I_t = [\,2\ \boxed{4}\ 6\,]$ —— ⊛ —— C_t $[\,0\ 2\ 0\,]$ ⇒ I/p Gate

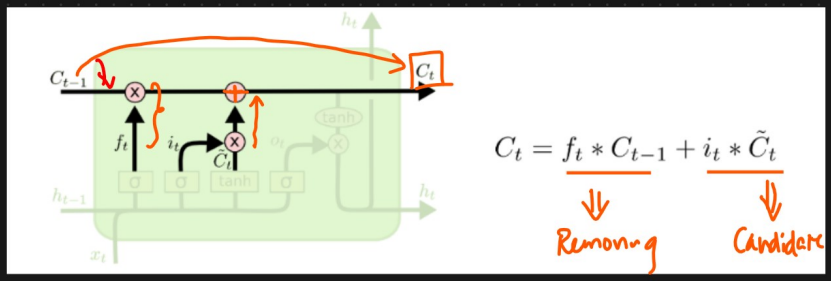
$[\,0\ 8\ 0\,]$





Context = If any information needed to be added it the memory

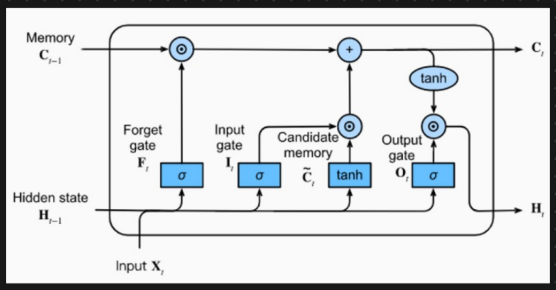
C_{t-1} ⟶ The information will be added



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

f_t ⇓ Removing

$i_t * \tilde{C}_t$ ⇓ Candidate



I stay in India . . . . . and I speak → English → Hindi

Or Forgetting Some info

Memory.

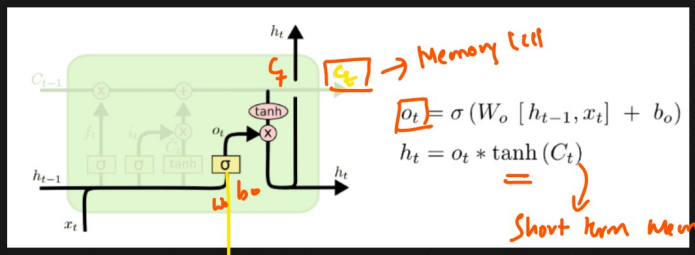
Forget Gate

I/p Gate (*) Candidate memory

+

Ct-1 => Cf.

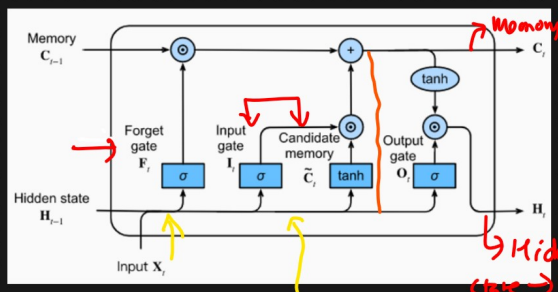
Output gate LSTM RNN



→ Memory Cell

→ Short term Memory.

→ Neural Network

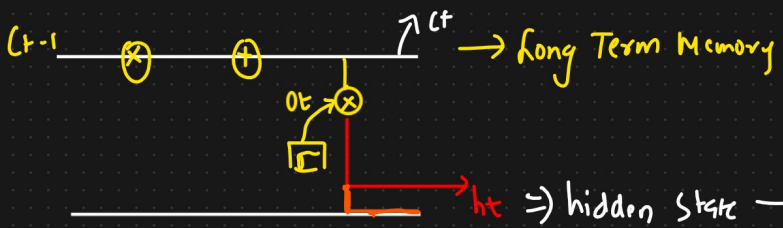


Memory

→ Hidden state → ST Memory.

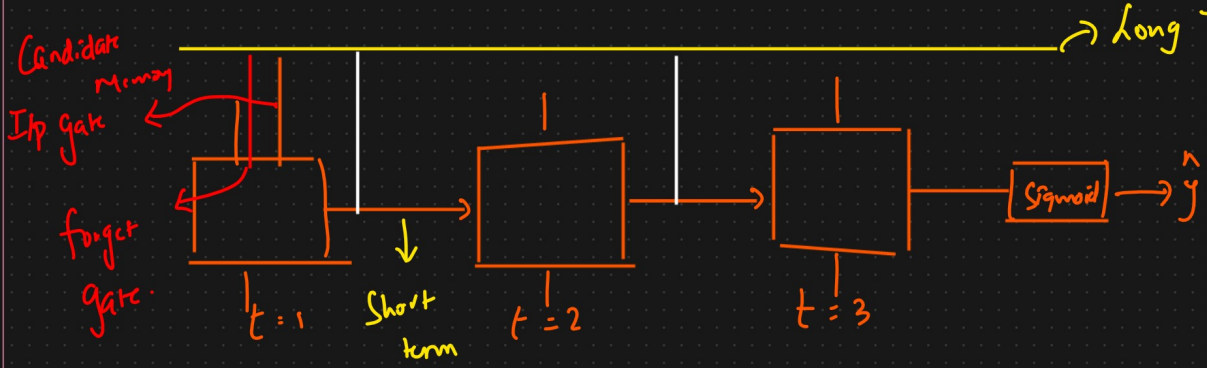
Forget → Some Info → context → Memory Cell

Add Info → Context



→ Long Term Memory

ht ⇒ hidden state → Short term Memory



→ Long Term

loss ↓↓.

[Wi, Wc, Wo] → Updating ← Back Propogation

GRU RNN ⇒ LSTM Variant