RNN → FFN, BP

2 Partical

LSTM

⇒ Word embedding ⇒ NN { Semantic meaning / Dense vector

Normal encoding

{ Doctmal- / ONM / TFIDF / Integer encoding }

1. Why RNN?
2. RNN ⇒ FP
3. RNN - code ⇒ (keras)
4. How BP works.
5. Problems with RNN ⇒ Vanishing gradient and exploding gradient

Long term memory issue

6. LSTM ⇒ World story

RNN ⇒ (gajani)

Long term

7. LSTM Partical ⇒ keras ⇒ 1990 - 2000

Decode the LSTM

8. GRU ← LSTM

two gate.

9. Deep RNN | Stacked RNN
10. Bidirectional RNN / LSTM / GRU

Seq to seq

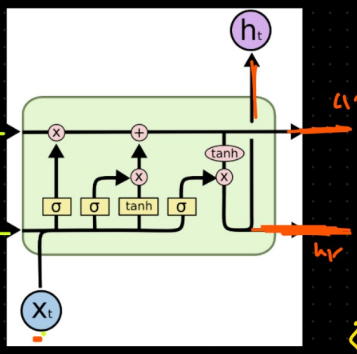
Encoder Decoder

Attention

Self Attention

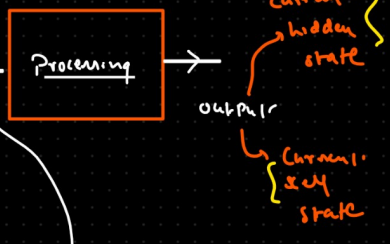
⇒ (End to end)

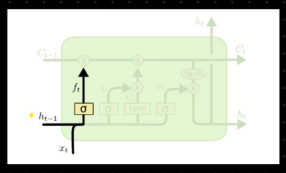
⇒ LSTM



{ Forget, update, output }

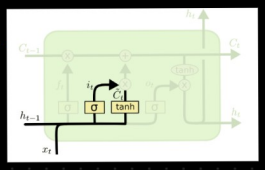
3 things involve





Forget gate

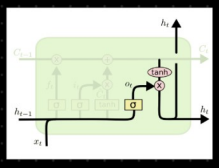
to remove something from the C_t



Input gate

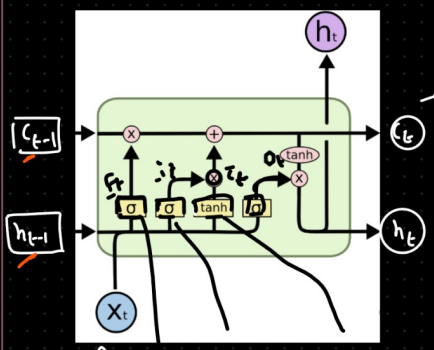
Add info to the cell state

C_t



Output gate

Calculate the hidden $\{C_t / h_t\}$



⊗ ⇒ Point wise multiplication

⊕ ⇒ Point wise addition

tanh ⇒ (act ⇒ ctr)

NN (Hidden) — (Hidden) Node — hidden tanh act

Hidden vector h_t, C_t

Vector

[0.1 0.2 0.3]

C_t Cell, h_t Hidden → Same

x_t ⇒ Input on the current TS

Vec

Sentiment

OHE [good bad better worst]

good bad better	0
bad bad better	1
worst bad good	0

[1,0,0,0] [0,1,0,0] [0,0,1,0] ⇒ OHE

$x_t \rightarrow x_{t+1} \rightarrow x_{t+2}$

Word by word ⇒ Lstm

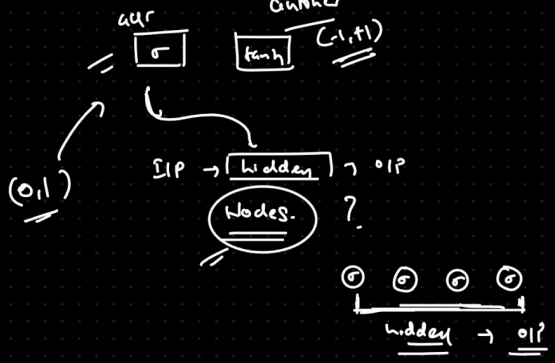
f_t → forget gate

$i_t, \tilde{C}_t$ → input gate

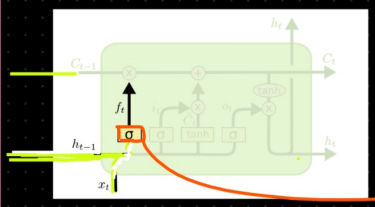
O_t → O/P gate

→ vector

I/P → architecture → fully → vector



Forget gate



Node = 3

4 dimension

$[x_{i1}\ x_{i2}\ x_{i3}\ x_{i4}]$

$[f_1\ f_2\ f_3] \Rightarrow (F)$

(3×1)

$[b_1\ b_2\ b_3] \Rightarrow 3$

7×3 = 21 weights → 24

3 bias

h_{t-1} x_t ⇒ (7×1)

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

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

$[1\ 1\ 1] \Rightarrow [4\ 5\ 6]$

remaining updating

cell

$[4\ 5\ 6]$ C_{t-1}

$\otimes$ ⇒ $[2, 2.5, 3]$

$[\frac{1}{2}\ \frac{1}{2}\ \frac{1}{2}]$ F_t

$[0\ 0\ 0]$ h_t

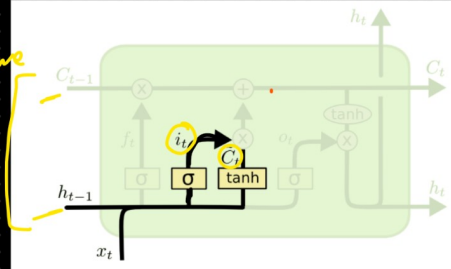
$[0\ 0\ 0]$

(no. of remove)

$F_t \otimes C_{t-1}$ ←

Input

Same



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)$$

$i_t = \sigma\left(W_i [h_{t-1}, x_t] + b_i\right)$ → (3×1)

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

$\tilde{C}_t = (3\times1)$ $[\cdot, \cdot, \cdot]$ T = tanh

b_1 b_2 b_3

T T T

3×1 4×1 (7×1)

$7\times3 = 21 + 3 = 24$

$x_t = 4$

$h_{t-1} = 3$

$i_t \otimes \tilde{C}_t = \tilde{C}_t$

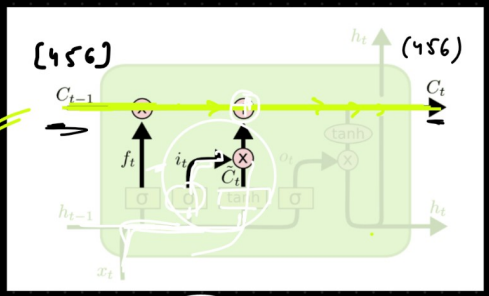
⇓

$C_t = f_t \otimes C_{t-1} \oplus (i_t \otimes \tilde{C}_t)$ ⇒ Current cell state

↓

(t)

RNN



( )

Recall

(1, 2, 3, 4)

$C_{t-1} = [4\ 5\ 6]$

$f_t = [1\ 1\ 1]$ ⊗ $[4\ 5\ 6]$

$i_t \otimes \tilde{C}_t = [0\ 0\ 0]$

$C_t = [4\ 5\ 6] \oplus [0\ 0\ 0]$

$= [4\ 5\ 6]$

LSTM

Update

[t-1 t]

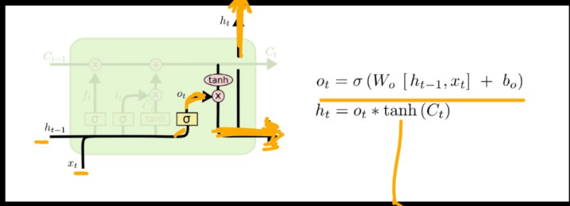
(lw , rb , tr) o

−∞ → ∞

cell state

Output gate

next hidden   Stacked RNN



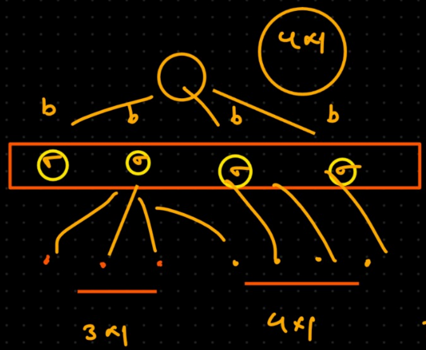
⇒ $h_t = O_t \otimes \tanh(C_t)$

[σ] ⇒ hidden layer

x_t , h_{t-1}

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

(3×1)   (4×1)



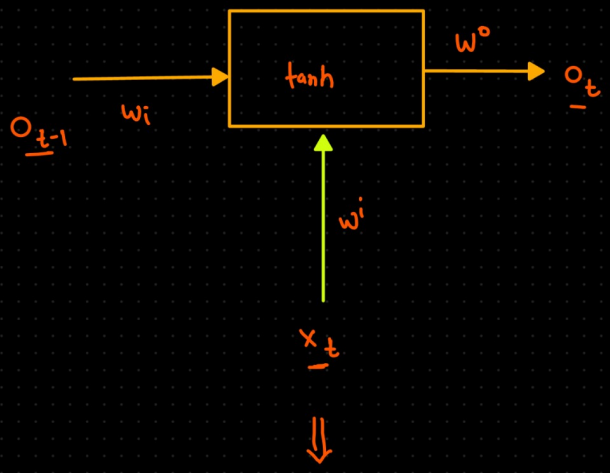
4

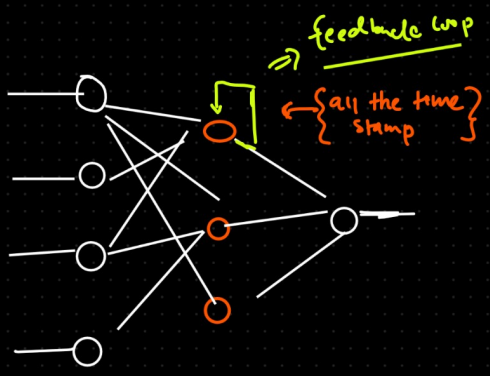
7×4 = 28 + 4 → 32

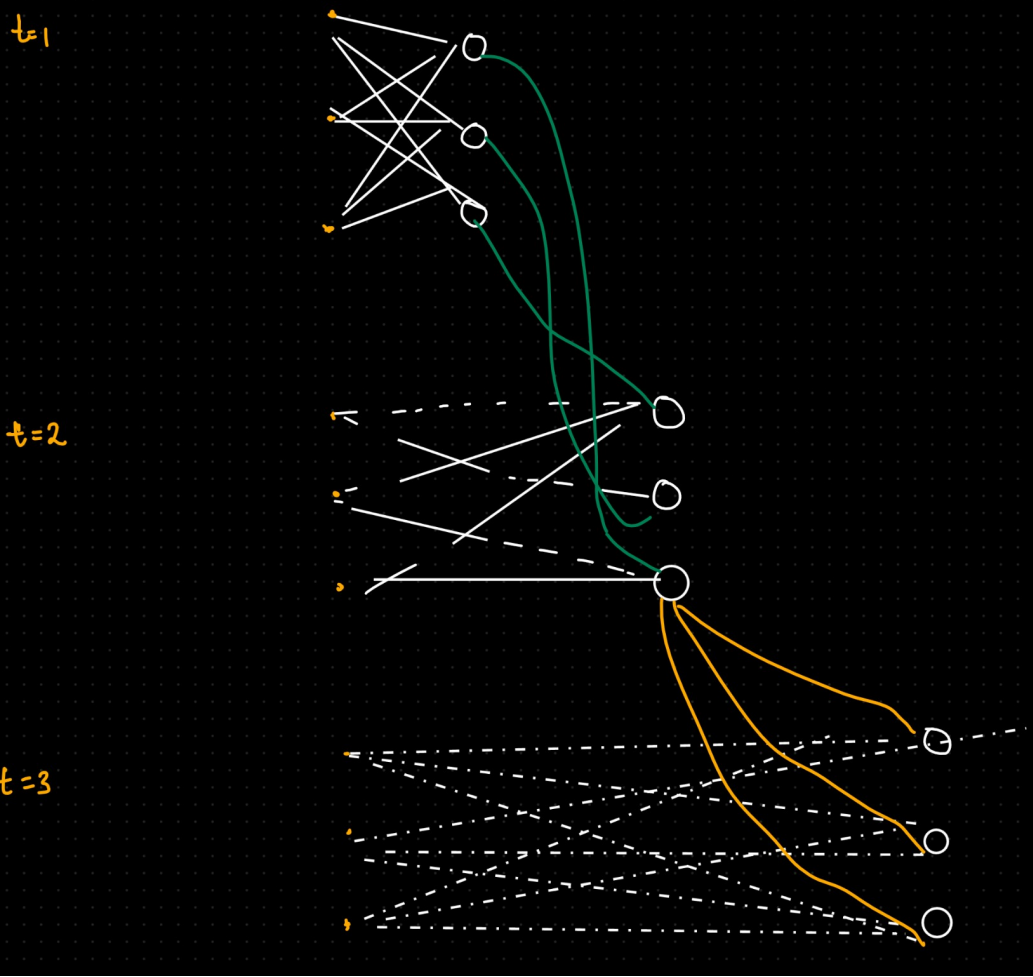
GRU ⇒ 2 gate

Deep RNN | Deep LSTM | Deep GRU

Feedback loop



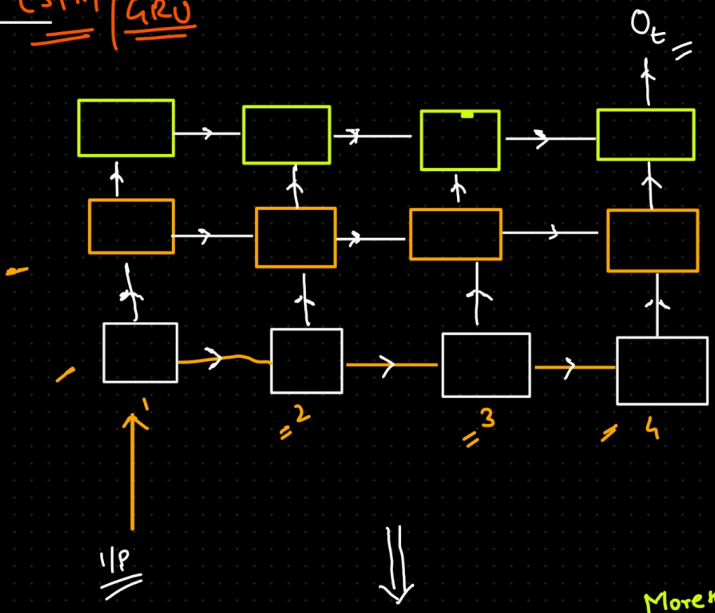




Deep RNN | LSTM | GRU

↓

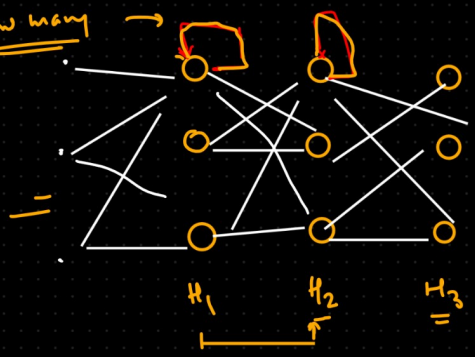
Hidden.

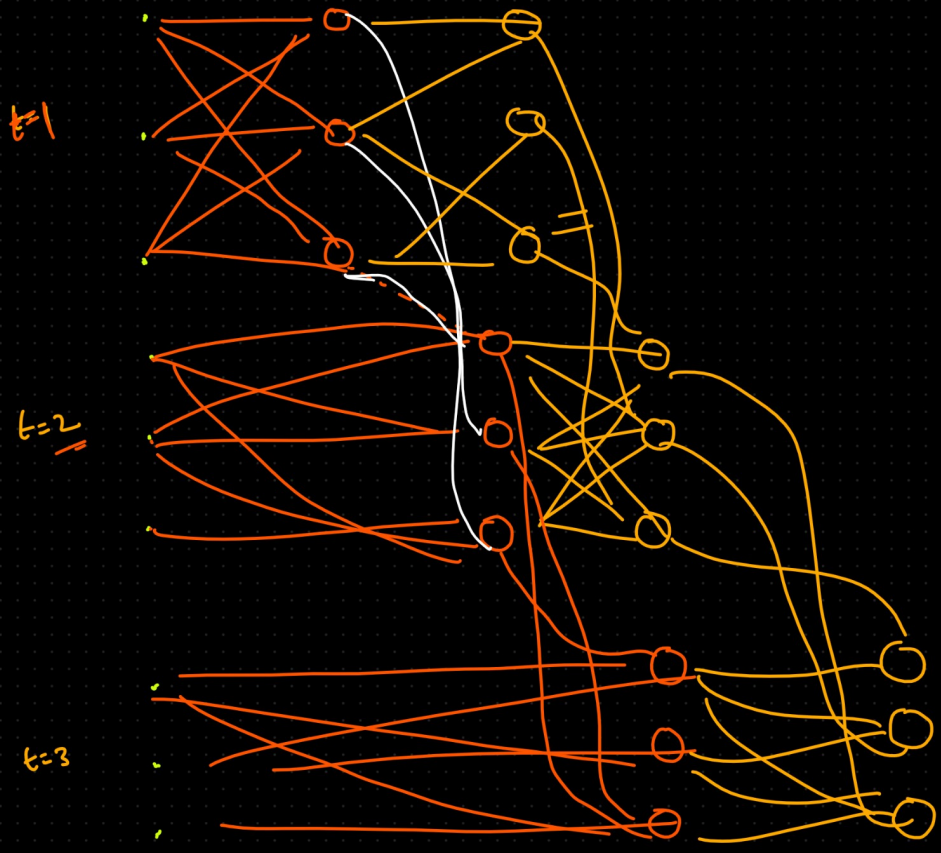


Moreman 1 hidden layer

until the all time stamp →

how many →





Why RNN / LSTM / GRU ?

Hidden layer ⇒ retrive complex

Stacked RNN

More hidden

text ⇒ hirarchical text

Complex

speech recog

the hidden layer

Complex ⇒ DRNN / SRNN

NLP

GRU ⇒ hirarchical ⇒ seq to seq

↳ Encoder 4 encoder ⇒ transformer

context vector

Attention

⇒ seq to seq

↳ RNN why ?

2

RNN LSTM

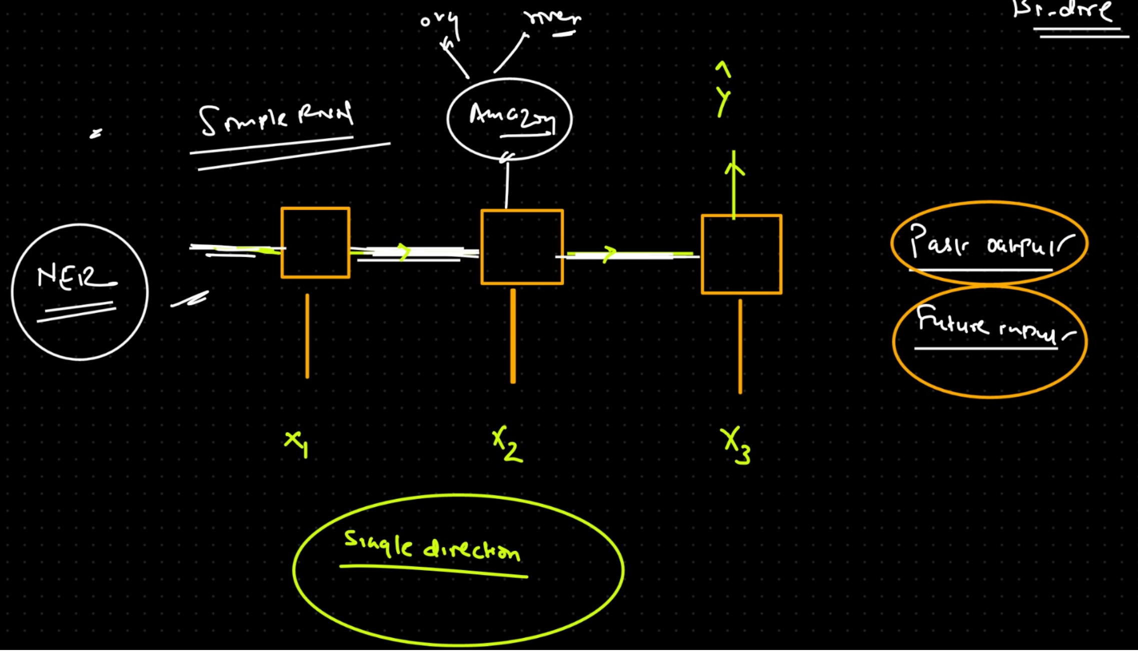
FFNN ⇒ { FF, key, v, lum, Pos }

Research ⇒

won → 3 to 4 PMISC

Bi directional RNN / Bi directional LSTM / Bi direction GRU

Bi-dire

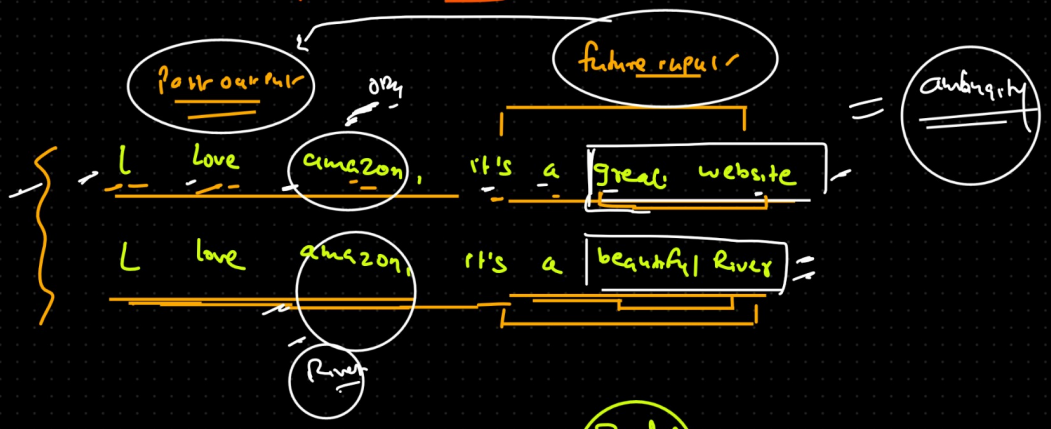


NER ⇒ Named entity Recognition

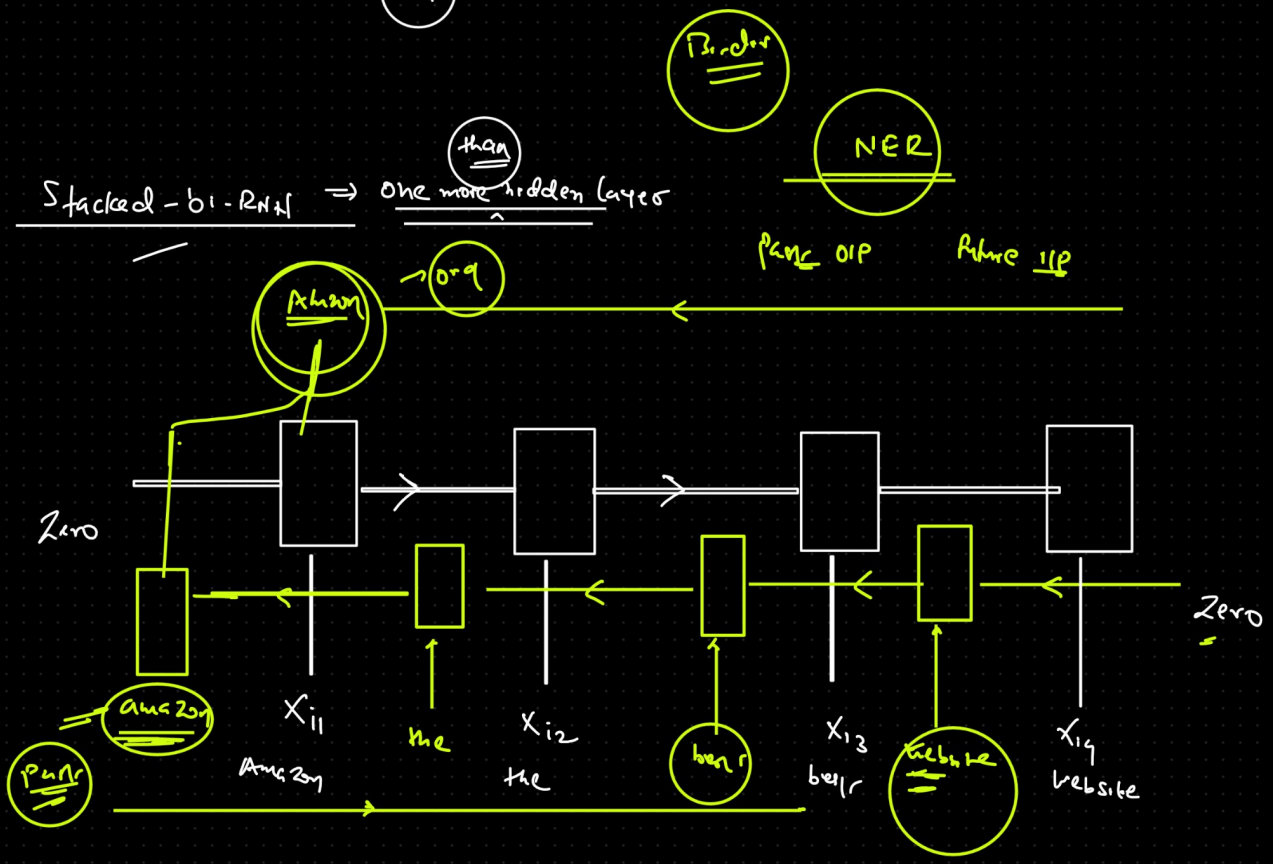
Name entity

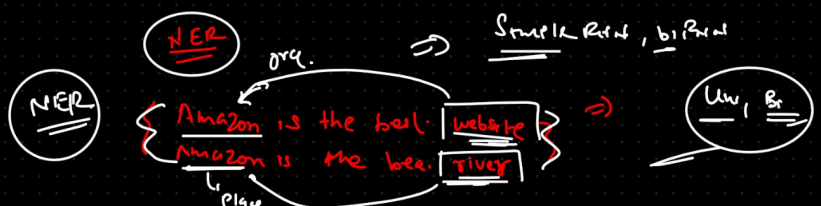
Sunny → Person

Bengaluru → Place

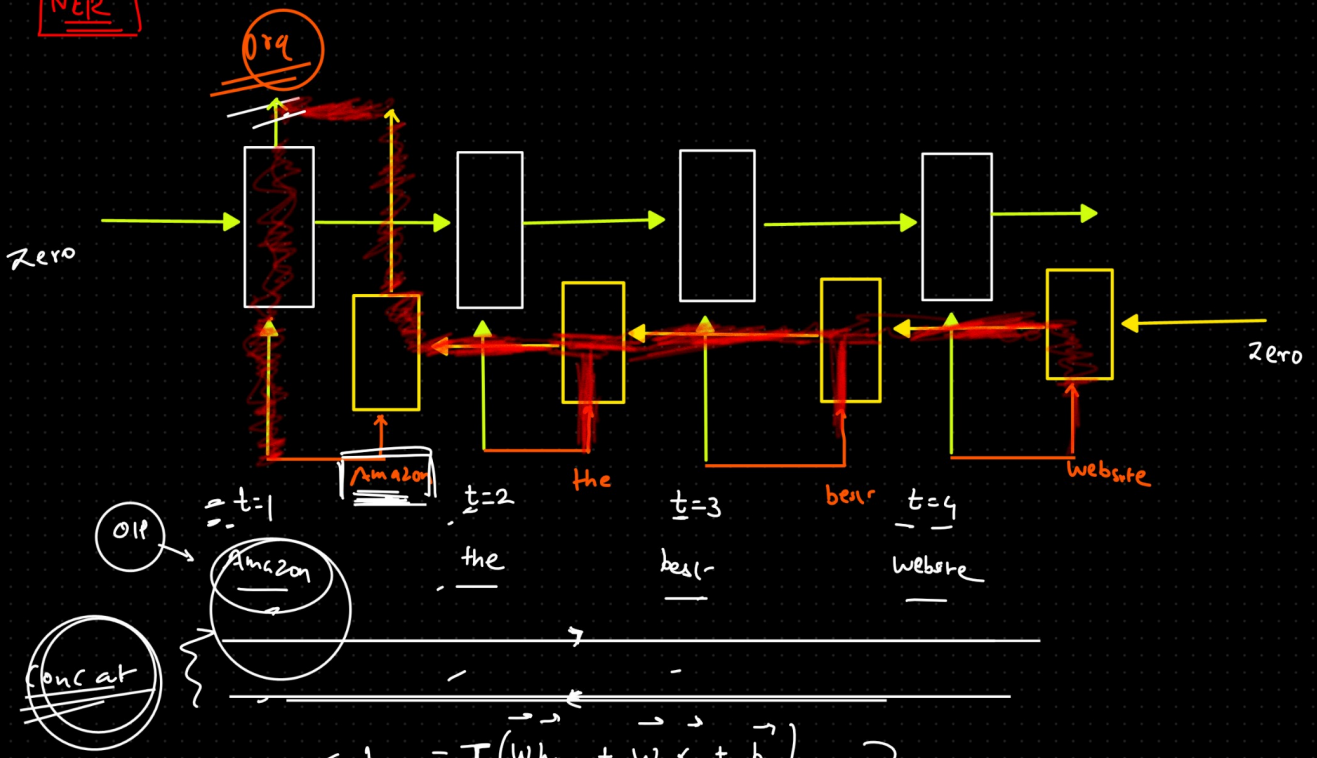


Stacked-bi-RNN ⇒ one more hidden layer than





NER



$$\begin{cases} \overrightarrow{h_t} = T\left(\overrightarrow{w_h} \overrightarrow{h_{t-1}} + \overrightarrow{w_i} \overrightarrow{x_t} + \overrightarrow{b}\right) \\ \overleftarrow{h_t} = T\left(\overleftarrow{w_h} \overleftarrow{h_{t-1}} + \overleftarrow{w_i} \overleftarrow{x_t} + \overleftarrow{b}\right) \end{cases}$$

Bi-directional

$$y_t = \text{Activ}\left(w\left[\overrightarrow{h_t}\ \overleftarrow{h_t}\right] + b\right)$$