

Leet-56)

# RNN

	review	sentiment
$x_1$	movie was good	1
$x_2$	movie was bad	0
$x_3$	movie was not good	0

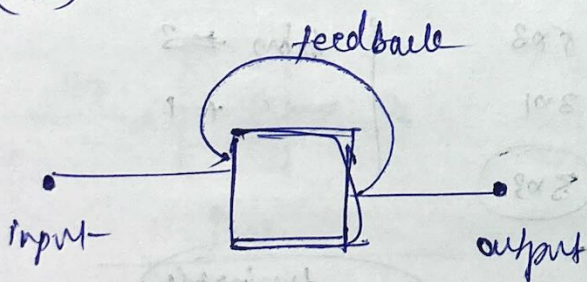
↓  
Total 5 unique words

movie    was    good    bad    not  
 $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow$  "movie" word  
 $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} \rightarrow$  "was" word

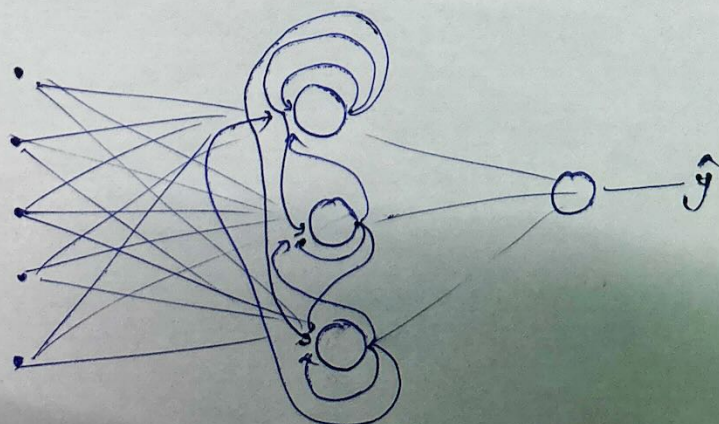
$x_1 = \left[ \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \right]$   
 $\quad \quad \quad t_1 \quad \quad \quad t_2 \quad \quad \quad t_3$   
 $\quad \quad \quad \uparrow$  no. of features, no. of times steps

$x_2 = (3 \times 5)$

$x_3 = (4 \times 5)$

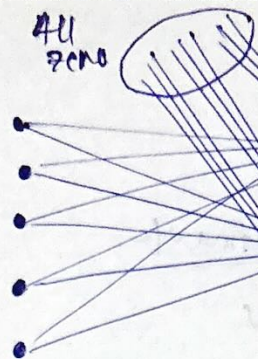


$x_1, t_1$   
 $x_2, t_2$   
 $x_3, t_3$

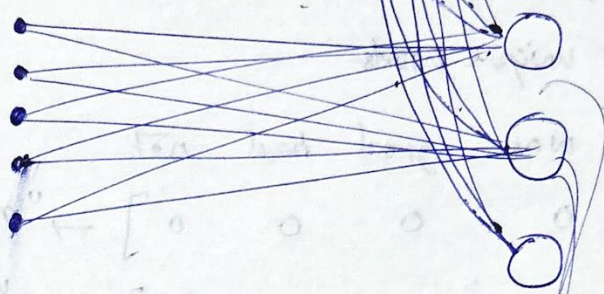




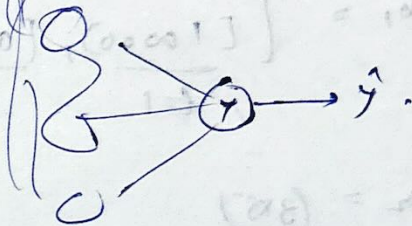
$f=1$   
 $x_{11}$



$f=2$   
 $x_{12}$



$f=3$   
 $x_{13}$



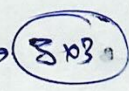
weights =  $5 \times 3$

$3 \times 1$

bias = 3

+ 1

neural layer



trainable  
 31 parameters



from keras.layers import Dense, SimpleRNN.

model.add(SimpleRNN(nodes, shape=(1,)))

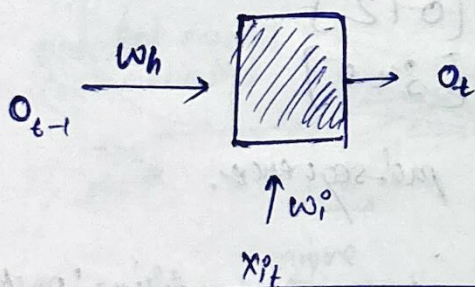
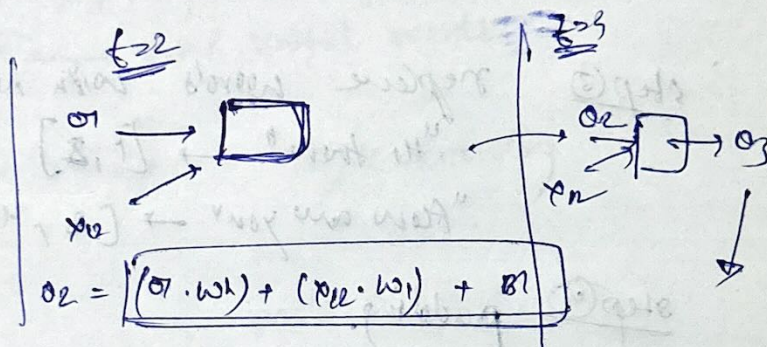
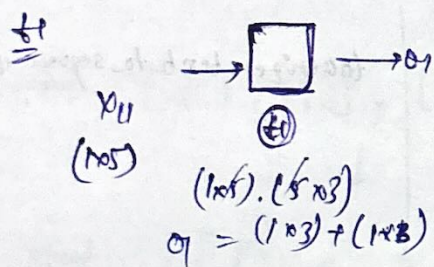
maximum time stamp  
features (words) in sequence

(eg longest sentence has 5 words, then 5)

$$w_h = (3 \times 3)$$

$$w_i = (5 \times 3)$$

$$b_i = (1 \times 3)$$



$$o_t = f((o_{t-1} \cdot w_h) + (x_{it} \cdot w_i) + b_i)$$

Activation /  $f$



# lect 57

text  $\rightarrow$  Numbers

① Integer encoding

② embeddings-

## Integer Encoding

from keras.preprocessing.text import Tokenizer  
tokenizer = Tokenizer(oov\_token='[UNK]')

step ① create vocab from given inputs.

(let say there are total of  $N$  unique words).

step ② Rank those words, ex

Hi $\rightarrow$ 1
there $\rightarrow$ 2
How $\rightarrow$ 3
Are $\rightarrow$ 4
You $\rightarrow$ 5

tokenizer.fit\_on\_texts(sentences)

step ③ replace words with no.

"Hi there"  $\rightarrow$  [1, 2]

"How are you"  $\rightarrow$  [3, 4, 5]

tokenizer.texts\_to\_sequences(sentences)

step ④ padding.

$\begin{bmatrix} 1 & 2 & 0 \\ 3 & 4 & 5 \end{bmatrix}$

$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}$

from keras.utils import pad\_sequences.

~~sequence~~  
sentences = pad\_sequences(sentences, padding='post')  
= 'pre'



# # Embeddings

step 1) use ~~import~~ Embedding

step 2)

from keras.layers import Embedding

(each word is represented by 2 numbers)

model.add(Embedding(5, output\_dim=2, input\_length=4))

no. of unique words in vocab

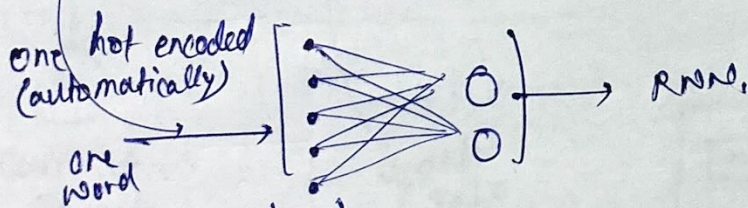
no. of words in each sentence

sent = ["How are you",  
"you are welcome though"]

unique words → How, are, you, welcome, though  
5 → total unique words

sent = [[1, 2, 3], [3, 2, 4, 5]] Integer Encoding

sent = [[1 2 3 0],  
[3 2 4 5]] padded (post)



unique words in vocab = 5  
output dim = 2  
input-length = 4

length of each sentence



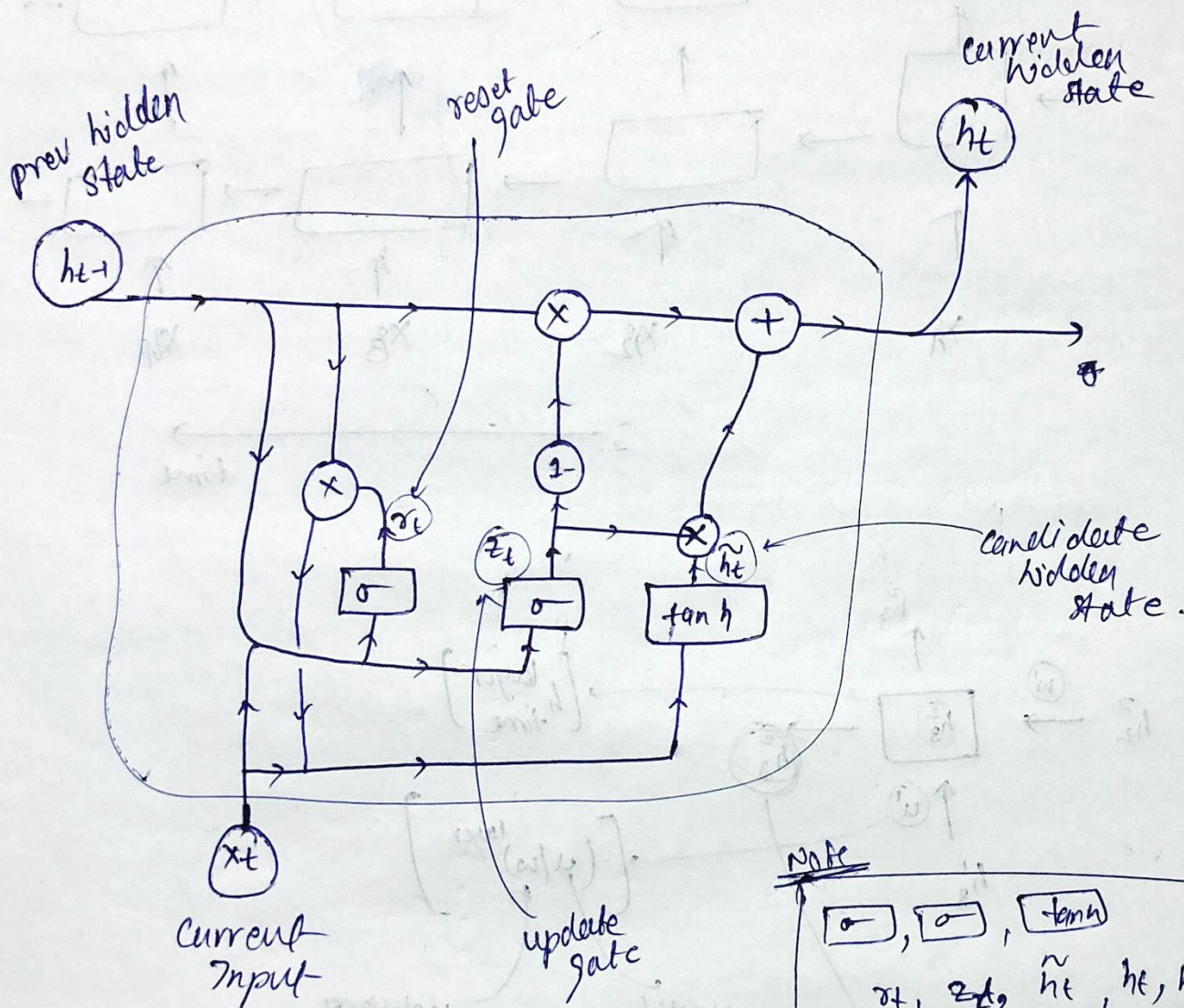
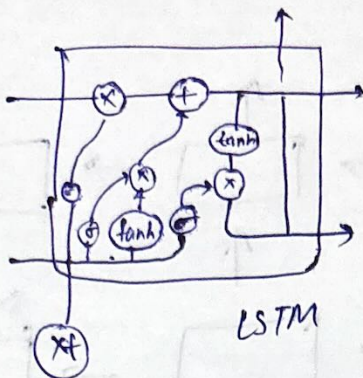
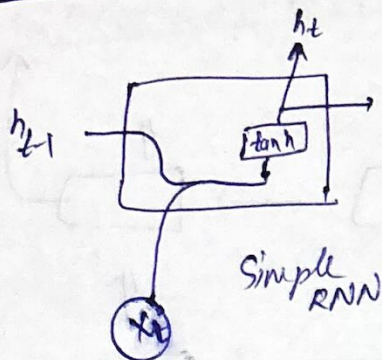
ref 027

## # LSTM

$$R_1 = \text{spol. } k[x, y]$$



# test 64 ] # GRU's



Note

$\sigma$ ,  $\sigma$ ,  $\tanh$

$x_t, z_t, \tilde{h}_t, h_t, h_{t-1}$

\* all have same dim

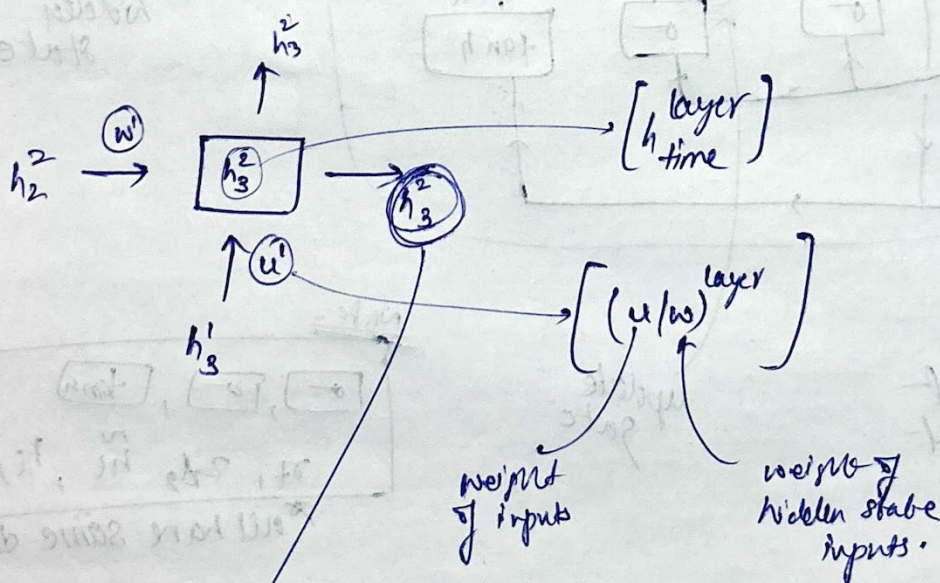
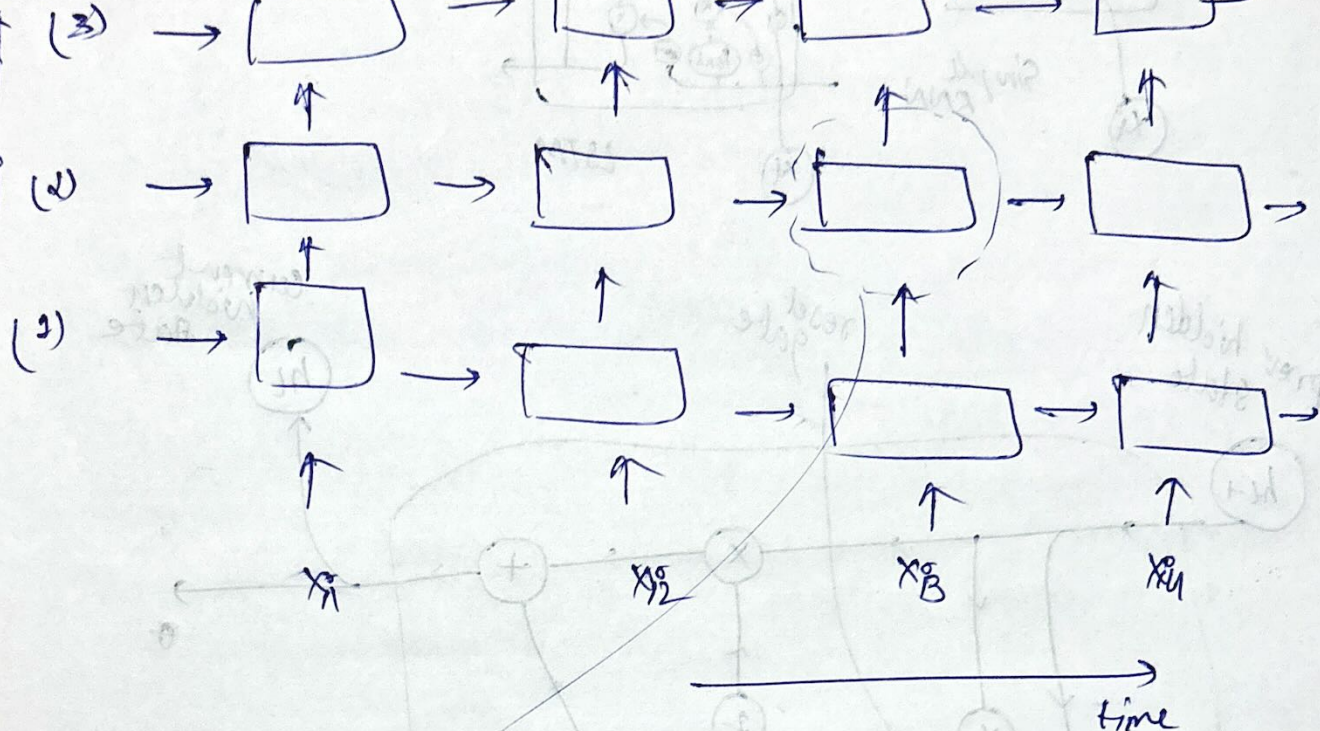
$$h_t = [(1 - z_t) * h_{t-1}] \oplus [z_t * \tilde{h}_t]$$



# test 65] # Deep RNN's / LSTM / GRU

for RNN's

layers (l)



Recall

$$h_t^e = h_3^2 = \tanh(w^e \cdot h_{t-1}^e + u^e h_t^{e-1} + b)$$



# lec66

11 Bidirectional RNN/LSTM/GRU

lec 66

11/11/2024

1 -> Left to Right

2 -> Right to Left

3 -> Bi-directional

4 -> Forward pass

5 -> Backward pass

6 -> Initialization

7 -> Propagation

8 -> Output layer

9 -> Test set processing

10 -> Part of speech processing (noun, verb, etc.)

11 -> Named Entity Recognition

12 -> Sentiment Analysis

13 -> Text classification (spam)

14 -> Word count

15 -> Part of speech

16 -> Named Entity

17 -> Sentiment

18 -> Text classification