Why RNN?

Already discussed in the last lecture

1) Sequence → any Sentence → ~~ANN~~ → {Fixed input size} {Padding}

↑ length very huge

↑ computational expensive

2) RNN ⇒ Recurrent Neural Network

<again and again>

What is Sequence?
Ans: Something that contains some meaning

Sequence ⇒ Contain some meaning

1 Data for RNN

good ⇒ 1
bad ⇒ 0

Sentiment prediction dataset taken as an example

Reviews	Sentiment
movie was good	1
movie was bad	0
movie was not good	0

Review 1 ⇒ [movie was good]

Review 1 [[1,0,0,0,0], [0,1,0,0,0], [0,0,1,0,0]]

(3,5) Size of encoded Review 1

for every corpus

OHE

Vocab

= [movie, was, good, bad, not]

(3 words in the sentence and 5 vocabulary)

As a first step we are going to convert the text into numbers/embeddings

NLP Project

text → Num

= [encoding]

1. Doc map
2. TF-IDF
3. OHE
4. word2vec
5. glove
6. elmo

(4, 6 → ANN)

Review 2 [[1,0,0,0,0], [0,1,0,0,0], [0,0,0,1,0]]

(3,5) Size of encoded Review 2

Review 3 [[1,0,0,0,0], [0,1,0,0,0], [0,0,0,0,1], [0,0,1,0,0]]

(4,5) Size of encoded Review 3

Keras for encoding

Keras → RNN → Batch Size, input feature

(3, 3, 5)

How RNN works

RNN ⟹ ANN

Difference b/w RNN and ANN:
2 big difference:
1. In ANN we are going to pass all the words in a sentence (present in the encoded form) in the single shot which raises length mismatch issue, Whereas, in RNN we are passing words based on timestamp and not in a single go(For ex; at t=0 will pass X11 then t=1 will pass X12 ..so on).

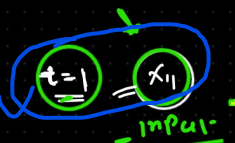
2. Another difference is ANN cannot preserve the sequence (No memory) whereas, RNN can preserve sequence(can retain memory) using feedback loop

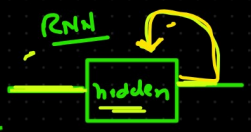
Review			Sentiment
x_1 movie	was	good	1
x_2 movie	was	bad	0
x_3 movie	was not	good	0

2 big Diff-

ANN
Hidden
Input
Output

RNN ⇒ Recurrent (Recurrency)



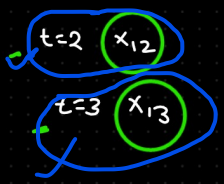


Sentence 1

[[1,0,0,0,0] [0,1,0,0,0], [0,0,1,0,0]]

No.of neurons needed in I/P layer → 15

Altogether (Missmatch)



Feedback loop

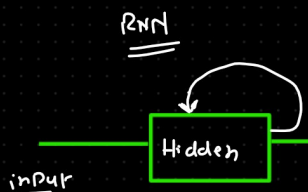
t=1
t=2
t=3

Sentence 3

[[1,0,0,0,0], [0,1,0,0,0], [0,0,0,0,1], [0,0,1,0,0]]

No.of neurons needed in I/P layer → 20 (Sequence)

Question: How many times will recurrency happen?
Ans: Until all the words are completed.



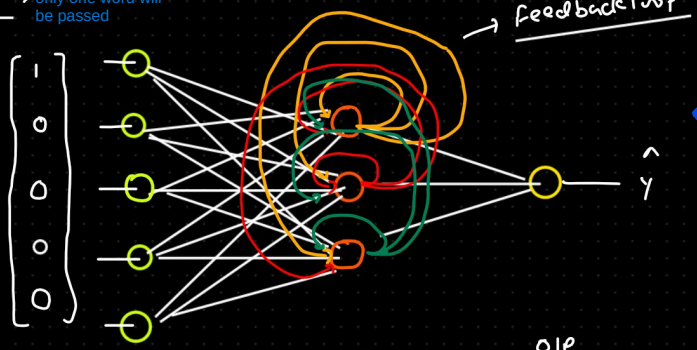
[[1,0,0,0,0] [0,1,0,0,0], [0,0,1,0,0]]

Encoded first sentence: Movie was good.

(RNN)

t=1, x_{11} ⇒ In RNN at a time only one word will be passed

Feedback loop



Please note that output to each neuron will be passed to all other neurons via feedback loop, which will act as an input in the next time stamp. Just observe the Orange, Red and Blue curves in the figure on left hand side.

Ques: Why there are 5 neurons in the input layer?
Ans: Always remember that in case of textual data in RNN neurons at i/p layer is equivalent to vocabulary size in that textual dataset.

I/P Hidden O/P

Neuron ⇒ [I/P * weight + b]

↓

{activation}

tanh

In research paper every neuron in RNN architecture was using the tanh activation function

word2vec ⇒ [illegible]

Understanding how feedback loop works in different timestamp and give RNN capability to memorize sequence

Review

RNN

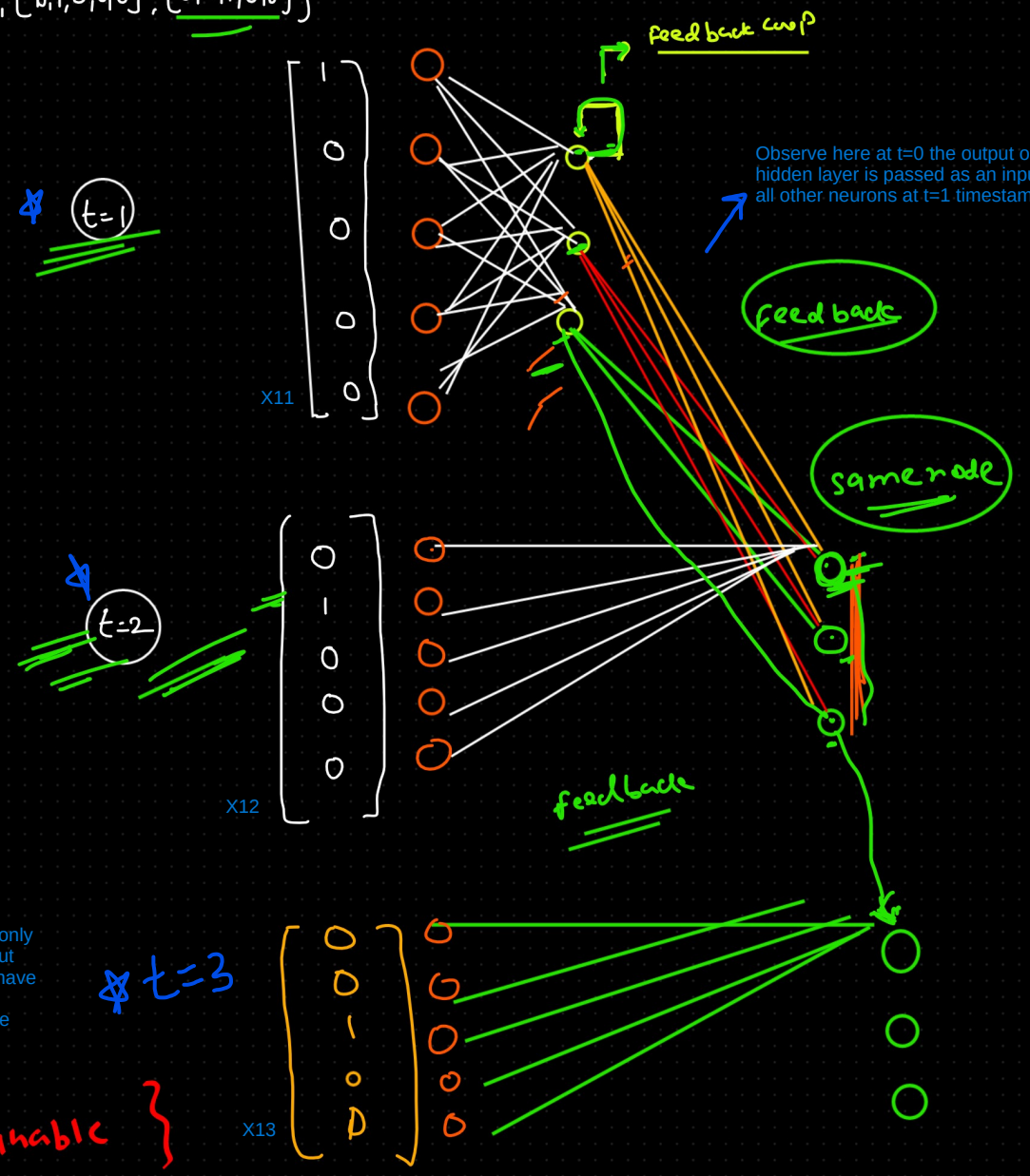
Review-1 ⇒ movie was Good

{every layer} = [tanh(IIPxW+b)]

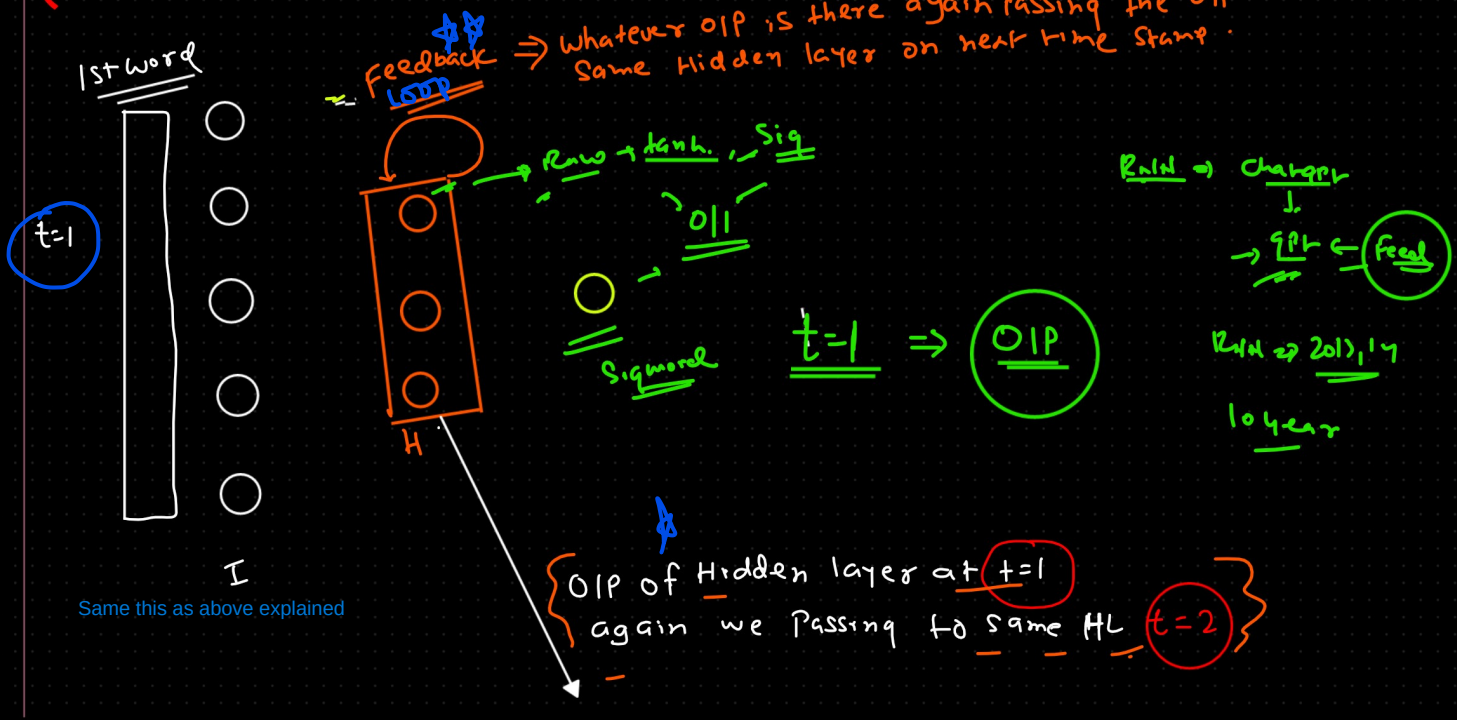
Review1 → [[1,0,0,0,0], [0,1,0,0,0], [0,0,1,0,0]]

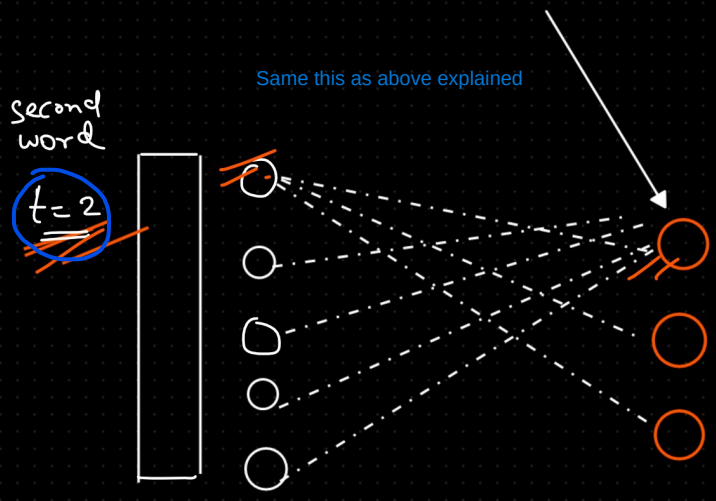
↓

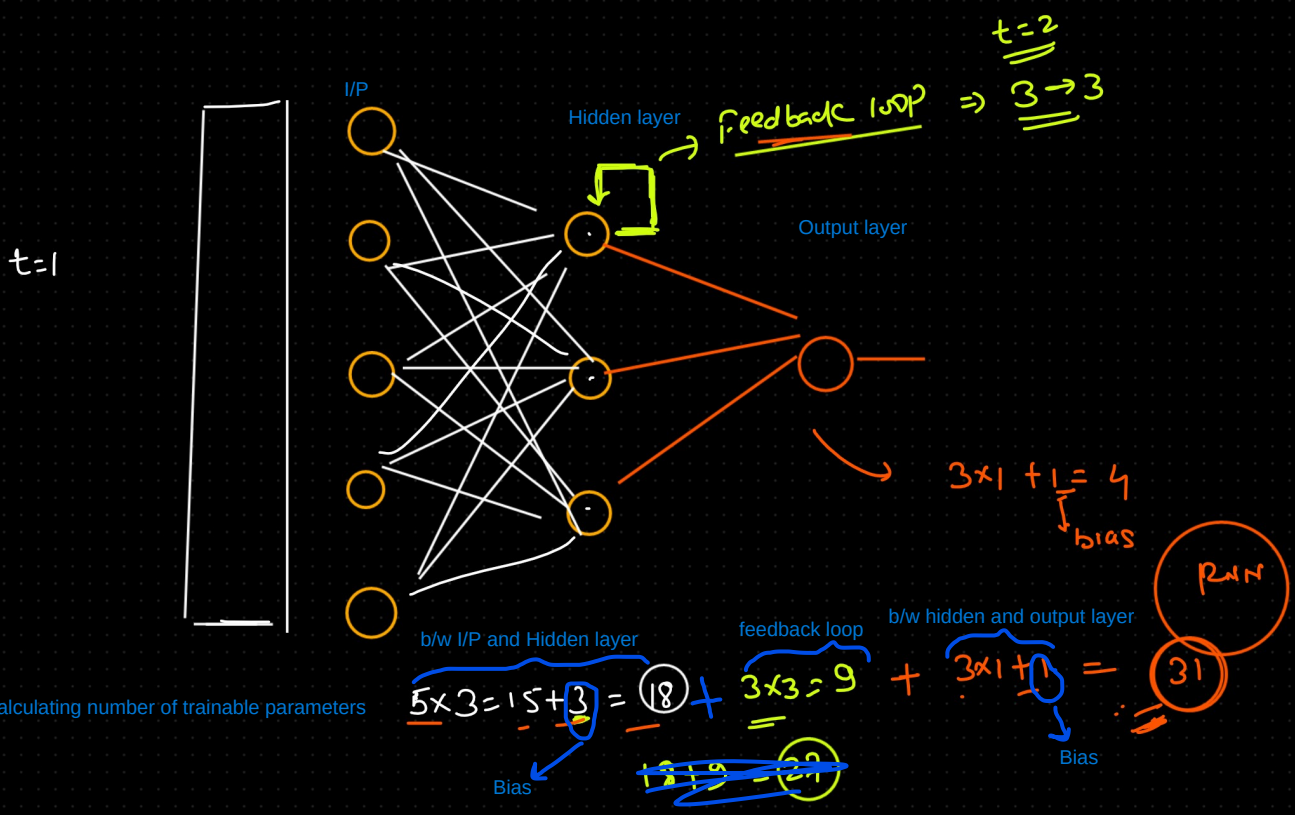
OHE



{Calculate trainable Param}





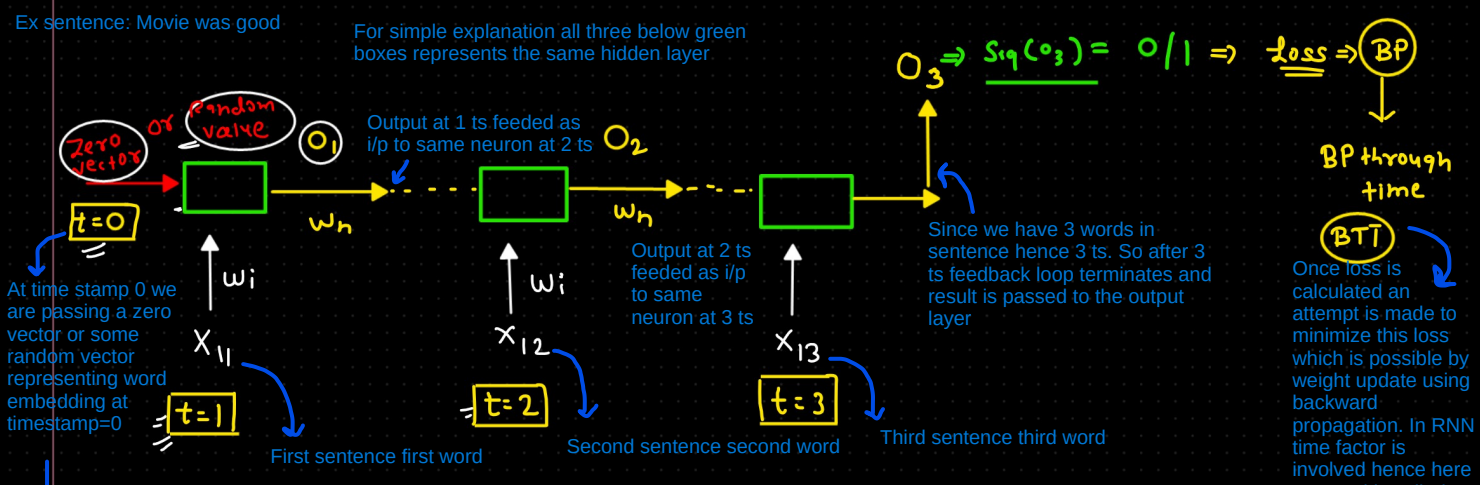


⇒ this Box it is a same Hidden Layer

Breaking down the functioning of feedback loop:

Ex sentence: Movie was good

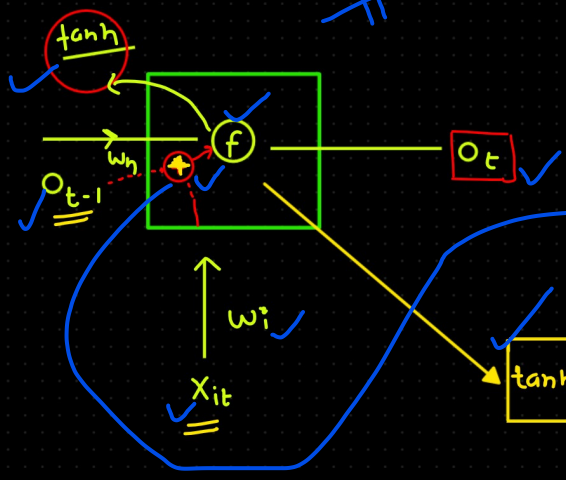
For simple explanation all three below green boxes represents the same hidden layer



Need of this?

Ans: So this acts as an initiator of feedback loop. like at t=1 the hidden neurons is suppose to have output from neurons in previous timestamp, which is not possible for neurons at ts=1. So to balance out this we are introducing the zero vector

Forward Propogation in RNN



$\hat{Y} = Sig(O_t \times W_o)$

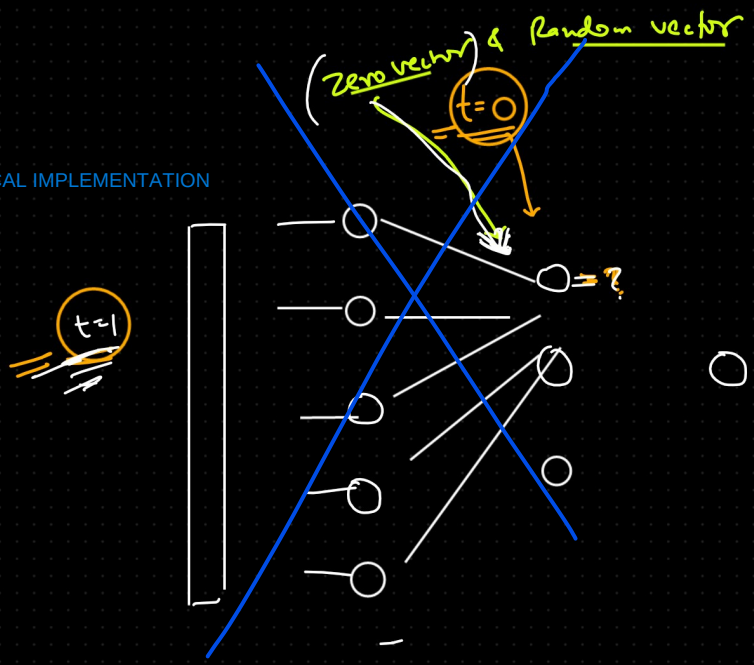
Loss, BTT

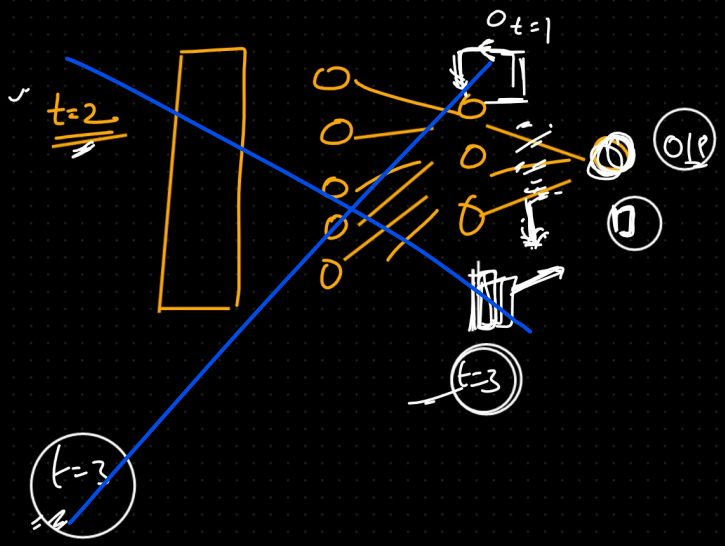
\+ represents this combine operation and passing to the passing through tanh activation fun^

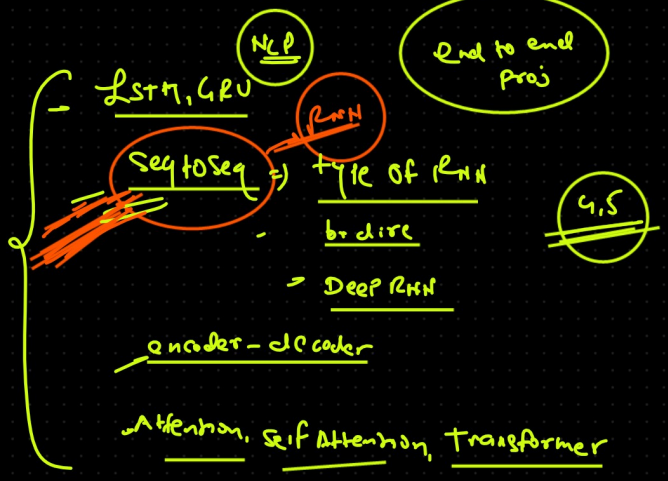
$\tanh(x_{it} \cdot W_i + O_{t-1} \cdot W_h)$

This is the output of tanh activated neurons that we are going to pass to the another layer

SEE .ipynb file for PRACTICAL IMPLEMENTATION







Data anal, Data eng

Sentiment analysis { m to m / 0 to m / m to 0 }

entire sent → single o/p