#RNN

- Type of NN used for Sequential Data.

ANN → Tabular Data.
CNN → Image
RNN → Recurrent NN.

→ Like if there is data with iq, marks, gender here sequence wont matter but data like text ("Hi my name is") this is sequential data like speech, DNA sequence thats why RNN used for NLP related areas. RNN can also remember what came before or after.

~~ANN~~

RNN Architecture & forward propagation.

→ when u put data in RNN it gets input to as shape;
(time steps, input - features).

Eg: (3,5) means 3 timestep & 5 input features.

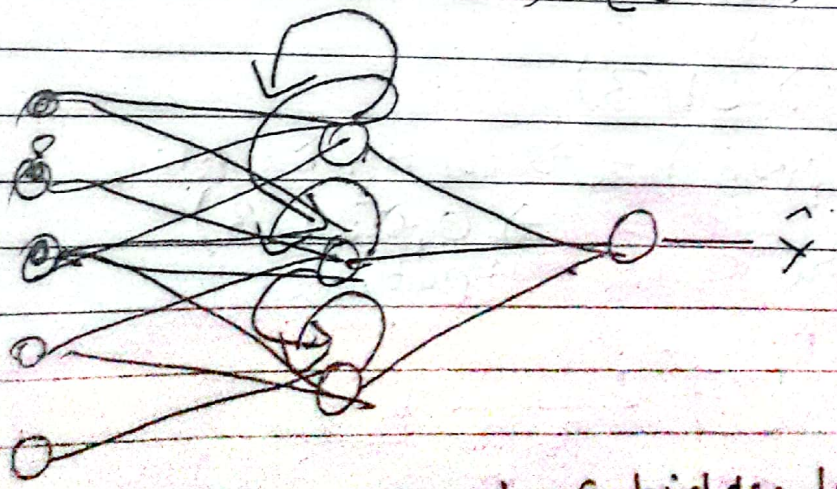
(x)

	Review	Sentiment
x_1	Movie was good (x_{11} x_{12} x_{13})	1
x_2	Movie was bad (x_{21} x_{22} x_{23})	0
x_3	Movies was not good (x_{31} x_{32} x_{33} x_{34})	0

In ANN we feed all input directly but in RNN we do one by one.

Convert above words to tokens using one unique.

movie (10000) was (01000) good (00100) bad (00010) not (00001)



Like output of hidden layer will again go back to it as input but for new timestep eg (t = 2) & we repeat the cycle. Each repeated will have its own new weight.

Review	sentient.
x_1 x_{12} x_{13}	1
x_{21} x_{22} x_{23}	0
x_{31} x_{32} x_{33} x_{34}.	0

Box will represent hidden layer

$t = 1$ $\uparrow w_i$

x_{11}

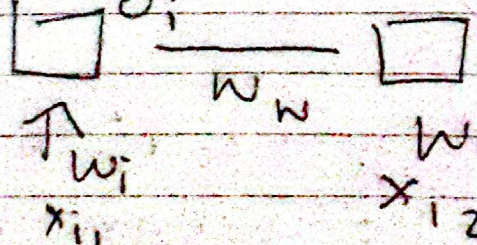
$x_i \cdot w_i \rightarrow (1, 3)$

$f(x_i w_i) \rightarrow$ tanh / ReLU

$\rightarrow o_1 = (1, 3)$

where o_1 is output$_1$

$t = 2$



$f(x_{12} w_i + o_1 w_w)$.

& it will repeat & we will get dot product of all output & apply sigmoid function to get Output.

U give new ~~ipt~~ input but use same weight & biases.

Types RNN

① Many 2 one

→ input → sequence →
output → non-sequential

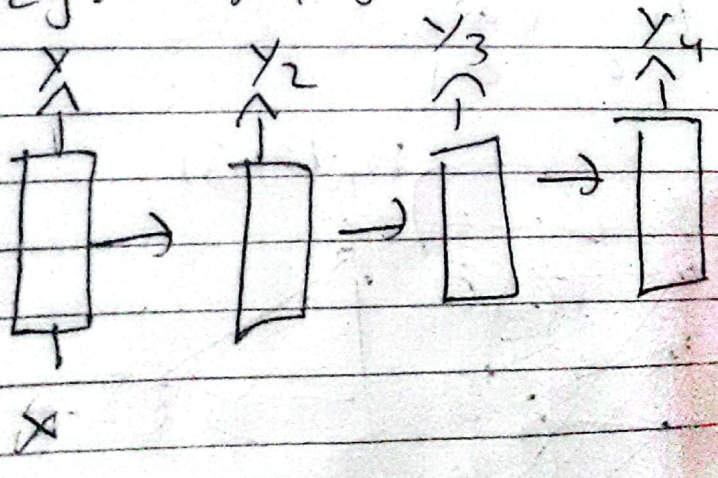
eg: sentimenat analysis where u give input in sequence but get output in non sequence.

u give multiple ipu but get one output.

② One to Many

input → non sequential
output → sequence

eg: Image (you send images & gets its output in text of what its doing).



③ Many to Many

Input → sequential

Output → sequential

also called seq2seq.

* Backpropagation through time (BPTT)

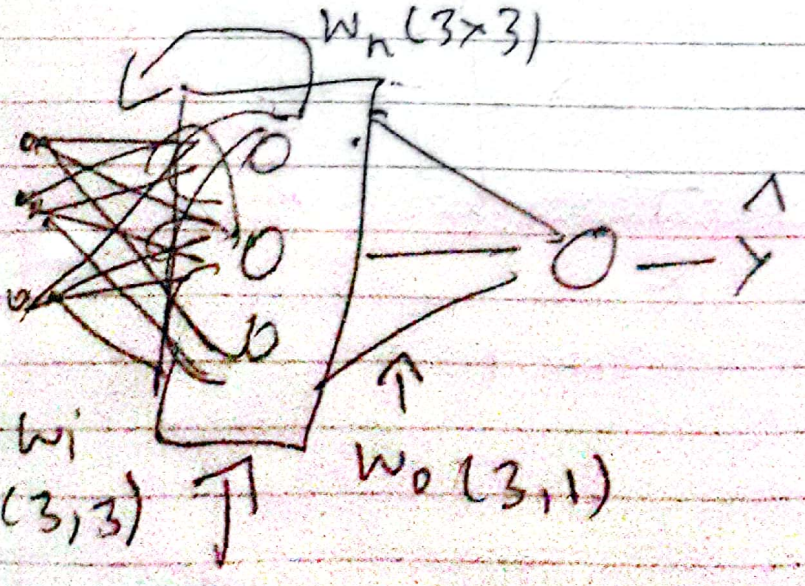
- Eg of Sentiment Analysis in Many to one RNN

text				
cat mat rat	1	x_1 100	010	001
rat rat mat	1	x_2 001	001	010
mat mat cat	0	x_3 010	010	100

Calculate unique words in vocab

Cat Mat Rat.

100 010 001



Forward propagation

In

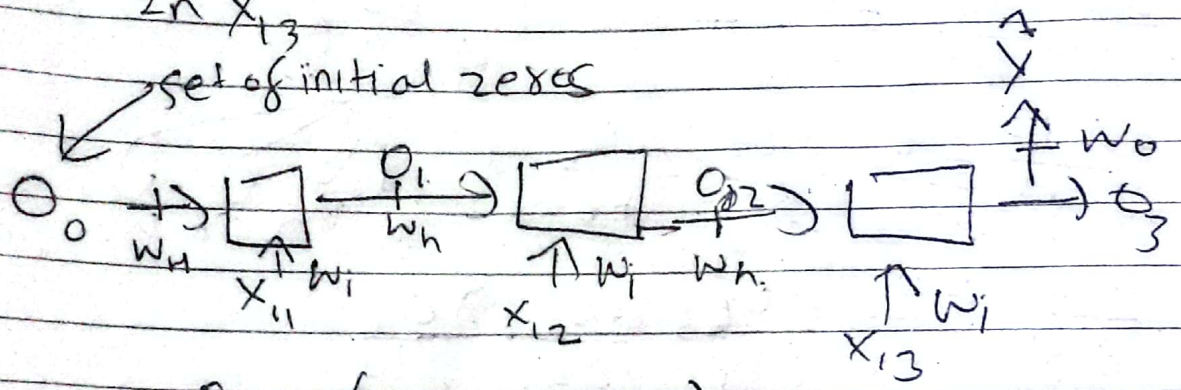
x_1 : 100 010 001

In x_{11} (100)

In x_{12}

In x_{13}

set of initial zeros



$$O_1 = f(X_{11} W_i + O_0 W_h)$$

$$O_2 = f(X_{12} W_i + O_1 W_h)$$

$$O_3 = f(X_{13} W_i + O_2 W_h)$$

$$\hat{Y} = \sigma(O_3 W_o)$$

$$L = -Y_i \log \hat{Y}_i - (1-y) \log (1-\hat{y})$$

Minimize L (loss) using gradient descent

Gradient descent: (Find w_i such that L is min)

$$W_i = W_i - \eta \frac{dL}{dW_i}$$

$$W_h = W_h - \eta \frac{dL}{dW_h}$$

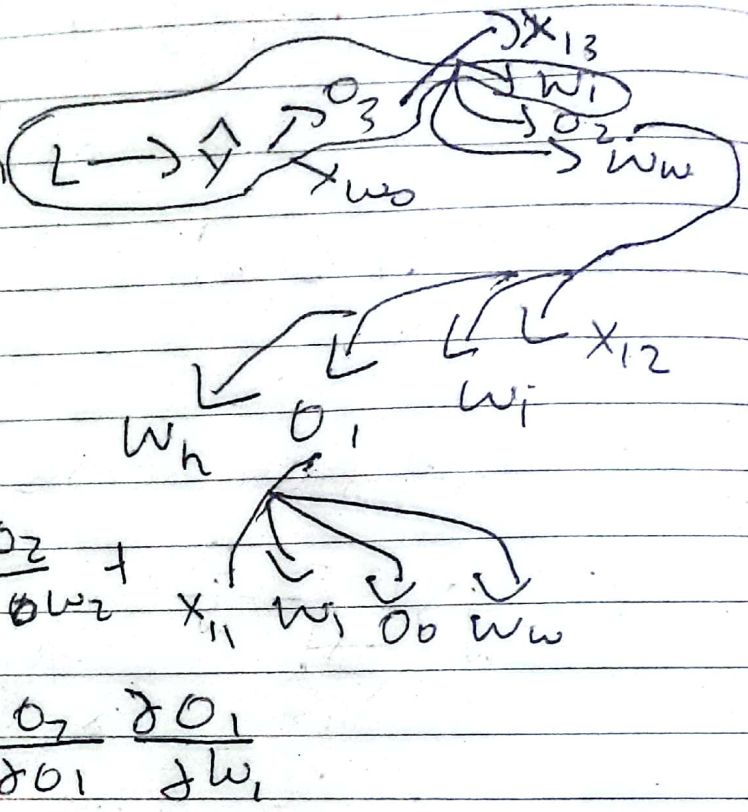
$$W_o = W_o - \eta \frac{dL}{dW_o}$$

depends on $L \to \hat{y} \to o_3 \to w_0$

chain rule

① $\frac{\partial L}{\partial w_0} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_0}$

② $\frac{\partial L}{\partial w_i}$ → depends on



$= \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial w_i} +$

$\frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial w_2} +$

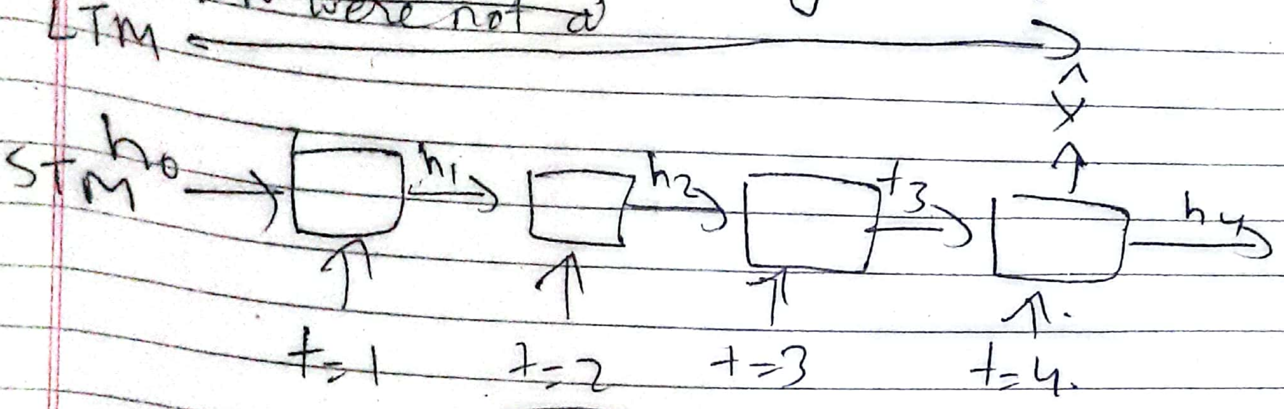
$\frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial o_1} \frac{\partial o_1}{\partial w_i}$

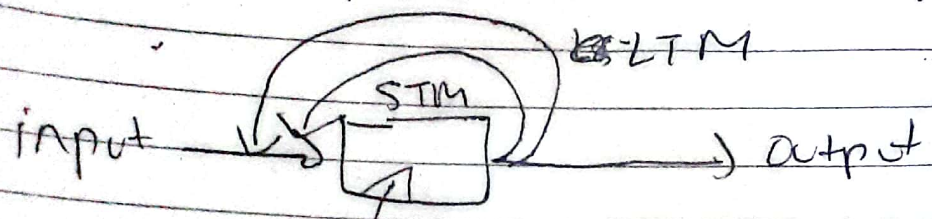
This isn't feasible to summarize

$$\frac{\partial L}{\partial w_i} = \sum_{j=1}^{3} \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_j} \frac{\partial o_j}{\partial w_i}$$

Long Short Term Memory (LSTM)

~~RNN were not a~~





~~difficult~~
~~architecture~~
because

STM will need to communicate to LTM

↓

in prev repo