Name = Harshit Yadav

Semester = 2

Registration no = 19091B017

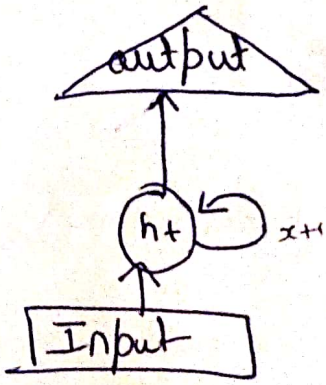
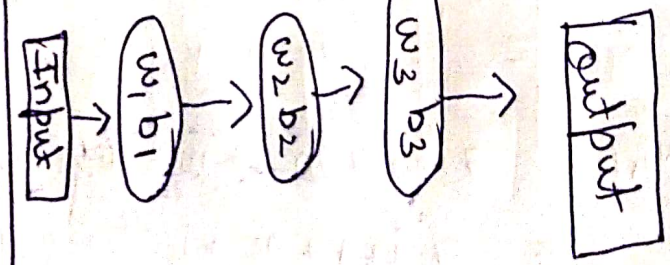
Course Name = Deep learning

Signature = Harsh

Date = 30 may 2020

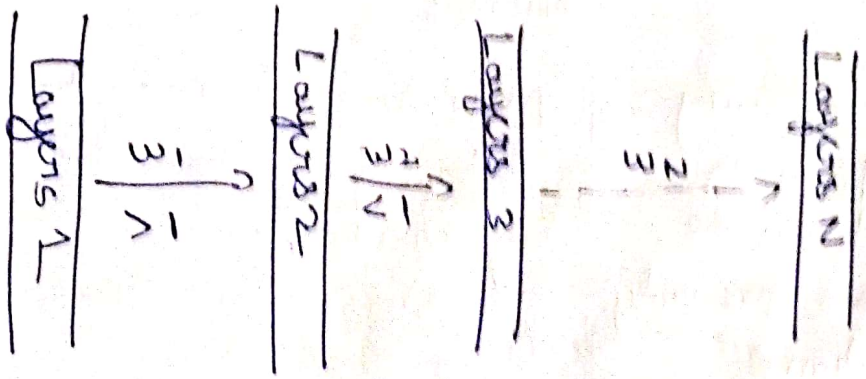
Assignment-4

Q Compare simple RNN and with ANN, Discuss and prove the vanishing/exploding gradient problem is much worse in simple RNN than DNN.

RNN	ANN
① 	
$h_t = f(h_{t-1}, x_t)$ ~~$h_t = \tanh(w_{hh} h$~~. $h_t = \tanh(w_{hh} h_{t-1} + w_{x+1} x_t)$ $y_t = w_{hy} h_t$	$Z = x_1 w_1 + b$ $Act(z) = output$
② It follows a recurrence relation instead of a fwd forward pass and uses ~~fwd~~ back propogation through time to learn.	ANN network travel in linear direction ~~dwd~~ during fwd forward process or the back propagation process.

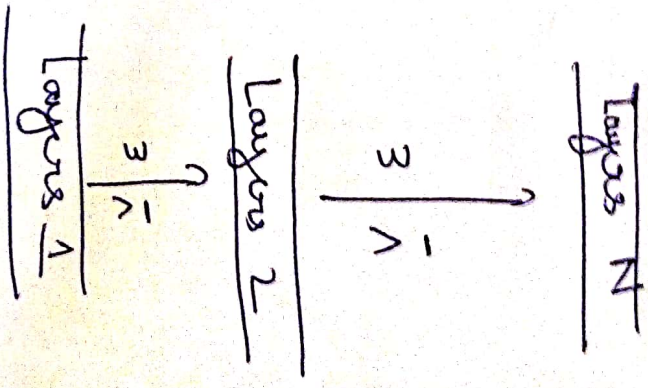
③ RNN does not have a limit on its size.	ANN works with fixed set of layers and number of neurons
④ RNN have the capacity to store the previous information. It has the memory to store previously processed information.	ANN does not have the capacity to store previous information no, such memory to store information
⑤ Weights and bias are dependent across the layers. same weight and bias across all layers.	weights and bias are independent of each other there will be different weights and bias at each layer
eg. Used in time series data such as stock market price or covid-19 data	used to process text data like word prediction etc

Multilayer NN



In DNN, between one layers we have weights have value can be more than 1 or less than 1, because of this effect it will cancel each other and hence deep neural network will occasionally come across problem of vanishing and exploding gradient

Recurrent NN



In RNN same weight parameters will recurr between one layers. If value of weight >1, then effect of it further grows as Iteration grows and hence RNN meet exploding gradient problem.

Similarly when weight <1, it further becomes less with the no. of iteration and hence RNN meets vanishing gradient problem.

② ~~LSTM~~ LSTM & cell state, forget gate, I/P Gate, O/P Gate
intermediate cell

→ The core idea of LSTM cell is to remember and forget the content in the memory unit, that memory unit is called cell state, LSTM cell units are an substitute for hidden layer neurons in the case of simple RNN.

cell state : Each LSTM recurrent unit maintain this which conceptually describe the information that was chosen to be retained by the previous LSTM recurrent unit.

$$C_t = (I_t + \tilde{C}_t) + (f_t + C_{t-1})$$

I_t = input gate value
$\tilde{C}_t$ = intermediate cell state value
f_t = forget gate output
C_{t-1} = cell state result of previous step.

Forget Gate: It determines to what extent to forget the previous data

$$f_t = \sigma(w_f S_{t-1} + w_f X_t)$$

f_t = forget gate value at time 't'
w_f = weights for forget gate
S_{t-1} = previous state value
X_t = input at time 't'

Input Gate = It determines the extent of information to be written onto internal cell state

$$i_t = \sigma(w_i S_{t-1} + w_i x_t)$$

w_i = weight for i/p gate

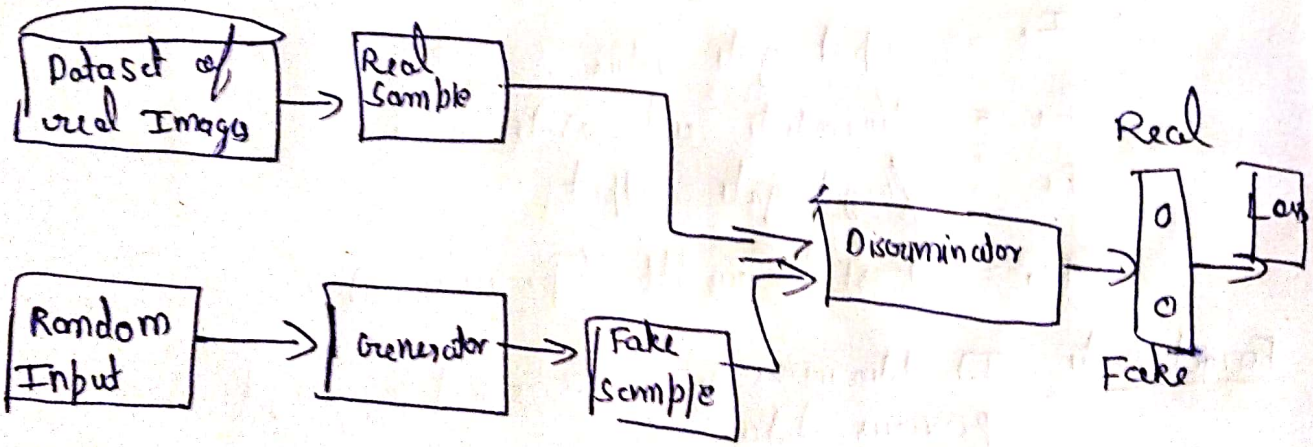
output gate = It determines what output to generate from the current internal all state

$$o_t = \sigma(w_o s_{t-1} + w_o x_t)$$

w_o = weights for o/p gate

Intermediate state = Tanh creates a vector of new values and it is stored in intermediate state

GAN



objective

→ The generator take sample of data (image, audio, etc) and tries to fool the discriminator to produce as real image as possible

→ the discriminator on the other hand tries to distinguish between real and fake samples.

→ the generator discriminator are both neural n/w and they both own in competion with each other in training phase

→ the steps are repeated several times, the generator & discriminator gets both better and better in their own classification after iteration.