Sequential Data

22 October 2022 13:09

ANN → tabular data

CNN → images

RNN → Recurrent NN

↳ is type of sequential model to work on sequential data

iq | marks | gender | placement

↘ does not matter

No

iq

marks

gender

RNN → NLP → ML

NLP → RNN

CNN → images → computer vision

eg → text → sequential data

Hi my name is Nitish

Time series

sequential

2001 2002 2003

Speech → sequential

DNA sequence

RNN

ANN / CNN

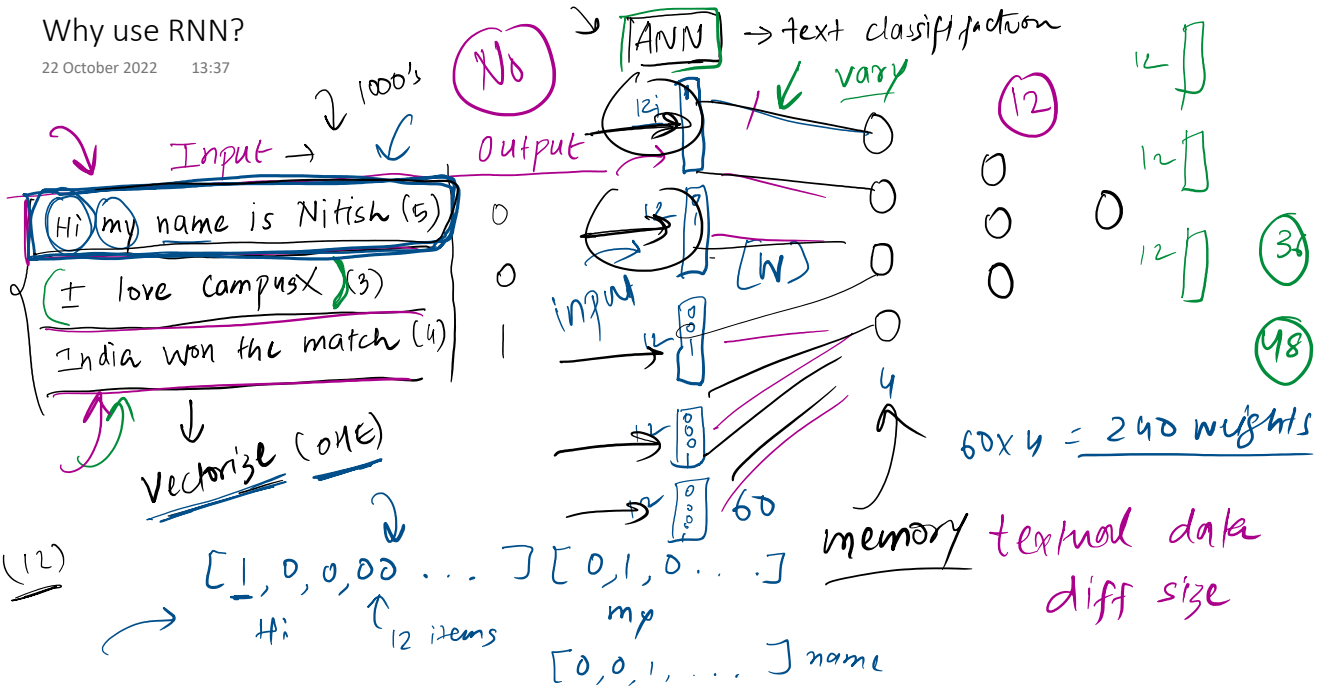
RNN Why?

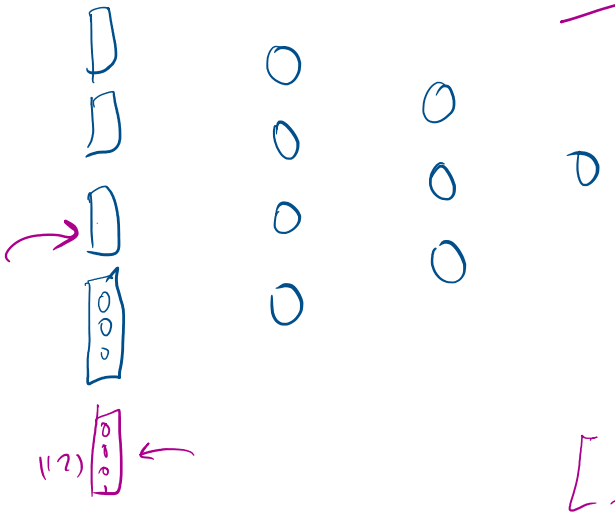
Application → RNN

Roadmap

Why use RNN?

22 October 2022 13:37





unneccessary computation

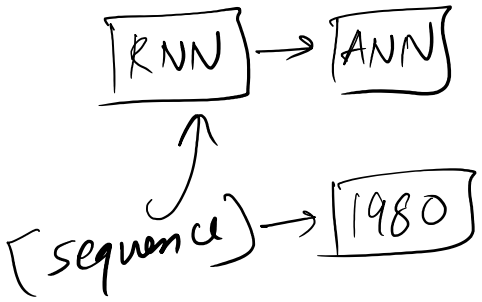
fail

predidin

200 words

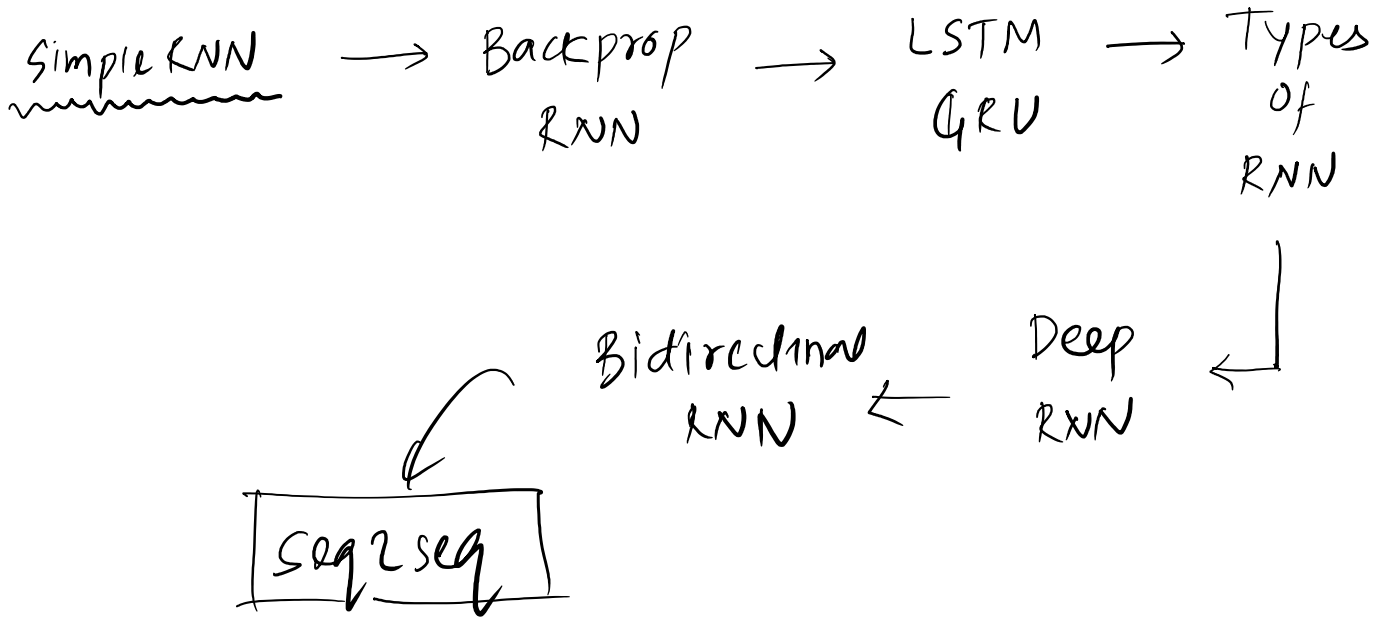
text input → varying size

1) text input → varying size
2) zero padding → unnecessary computation
3) Prediction problem
4) Totally disregarding the sequence info



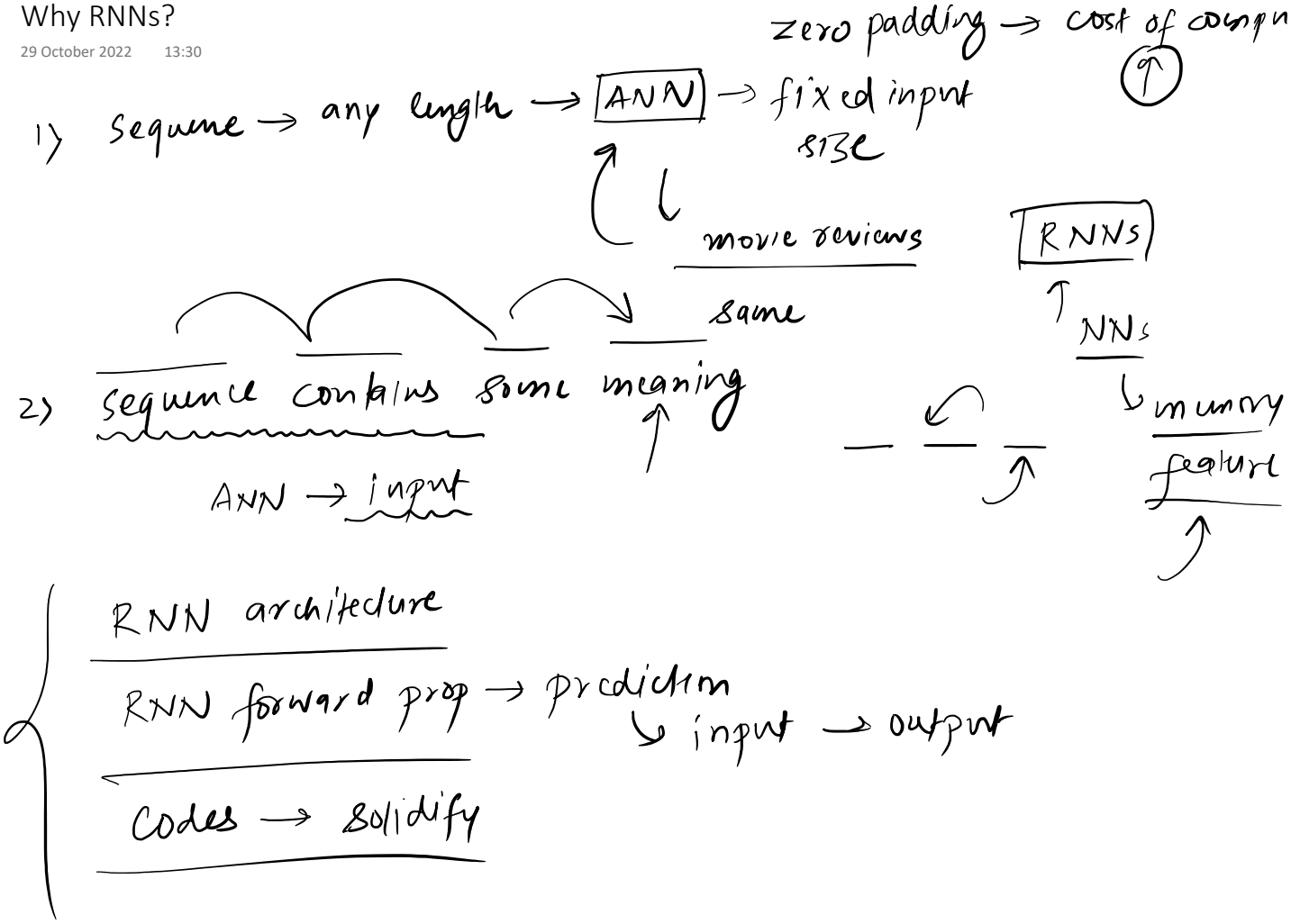
Roadmap

22 October 2022 13:37



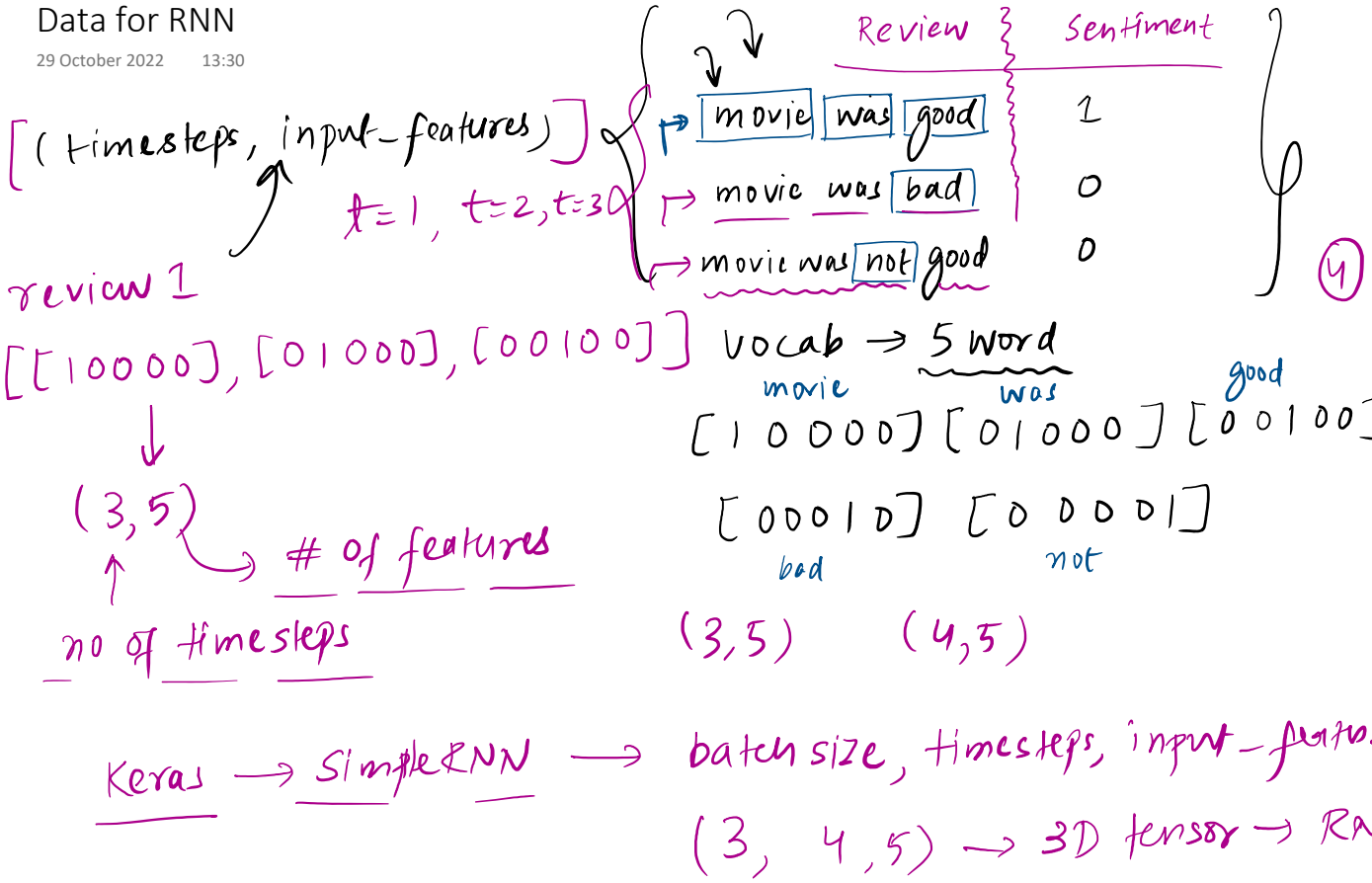
Why RNNs?

29 October 2022 13:30



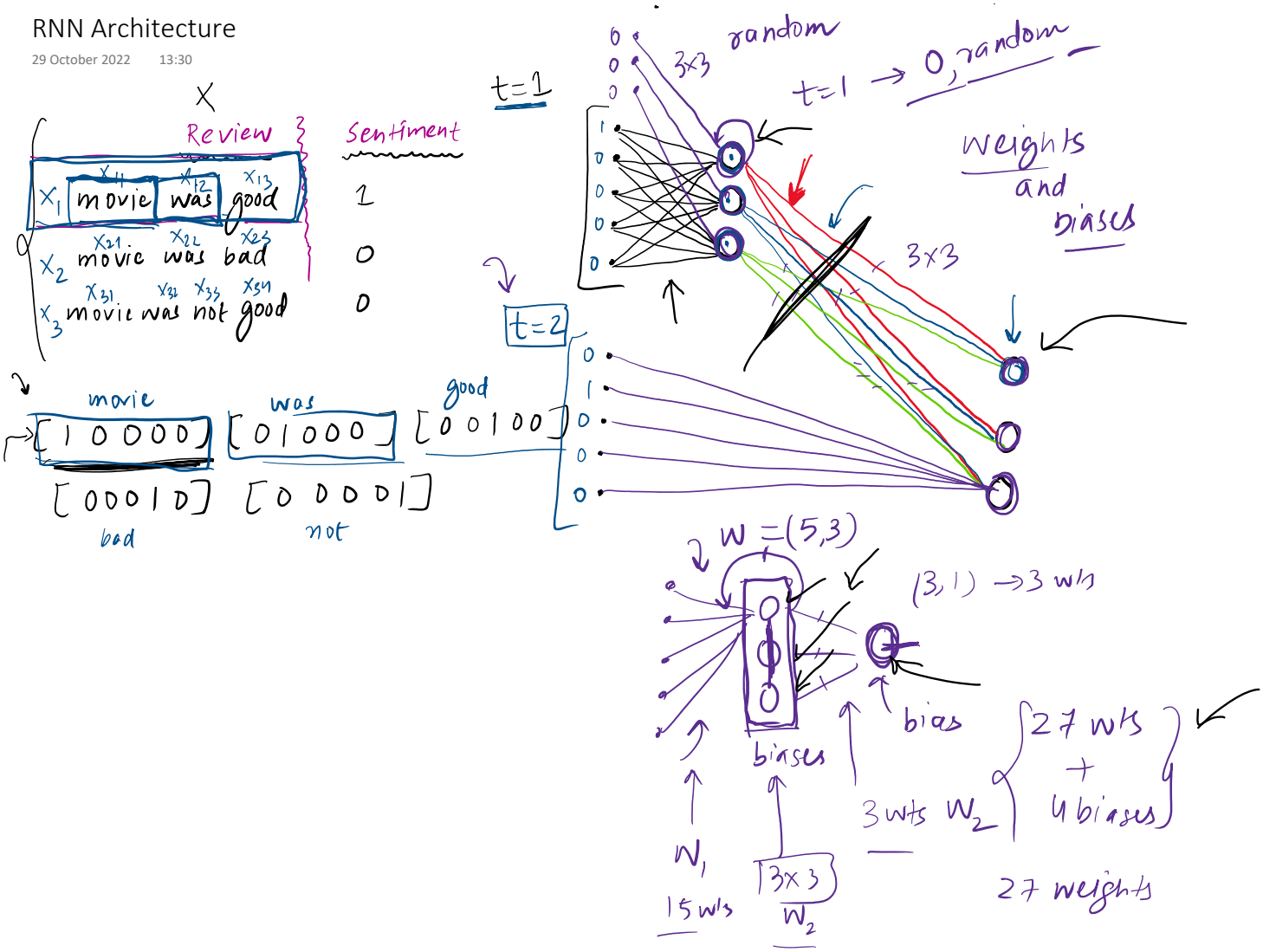
Data for RNN

29 October 2022 13:30



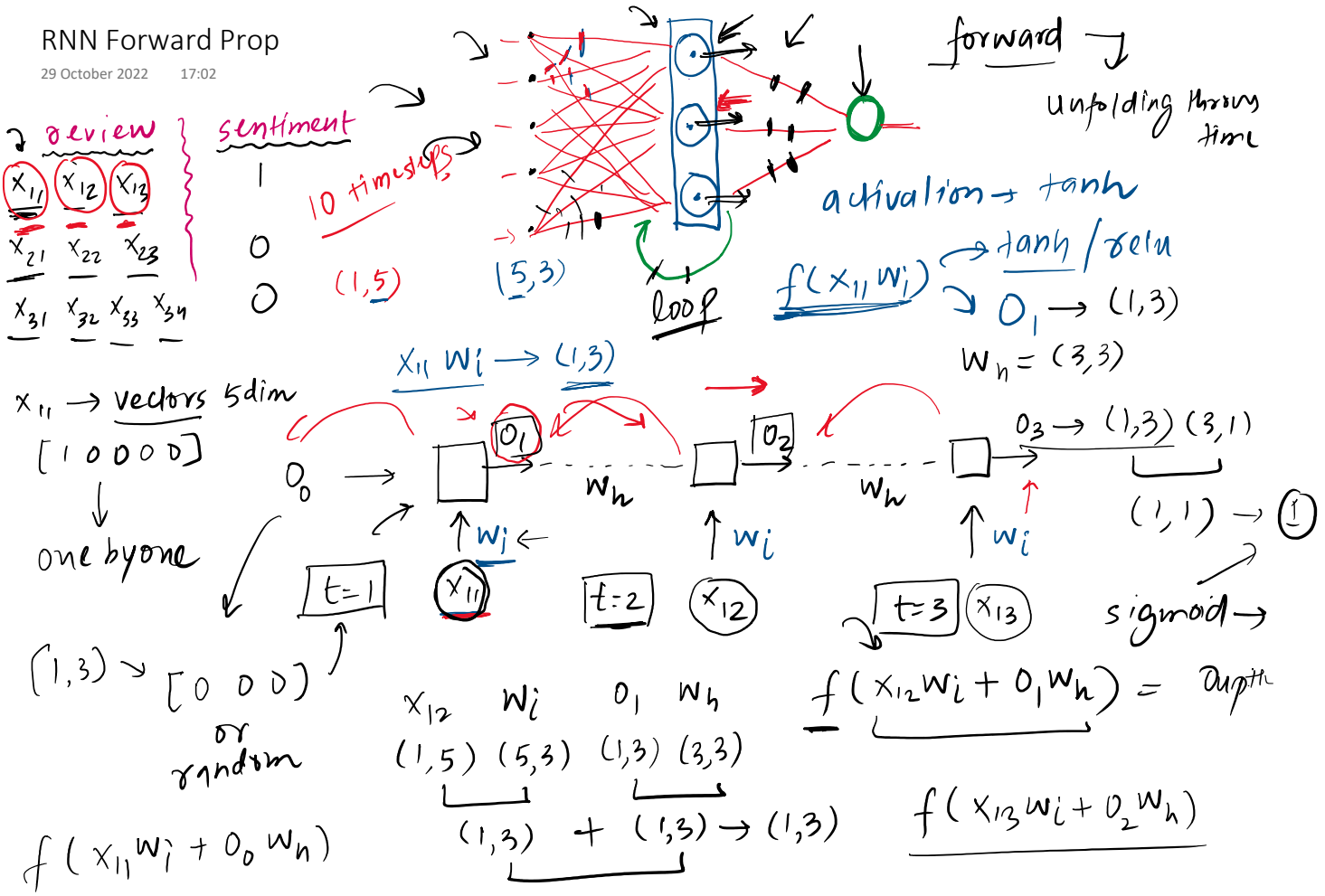
RNN Architecture

29 October 2022 13:30



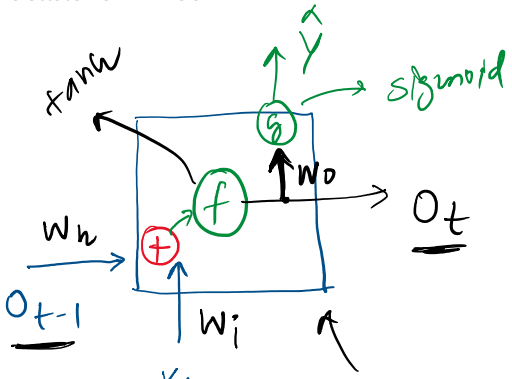
RNN Forward Prop

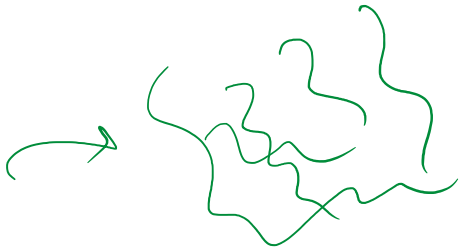
29 October 2022 17:02



Simplified Representation

29 October 2022 13:32





$\hat{y} = g(O_t W_o)$

$f(X_{it} W_i + O_{t-1} W_h) \rightarrow$ vector $\rightarrow O_t$

i = row #

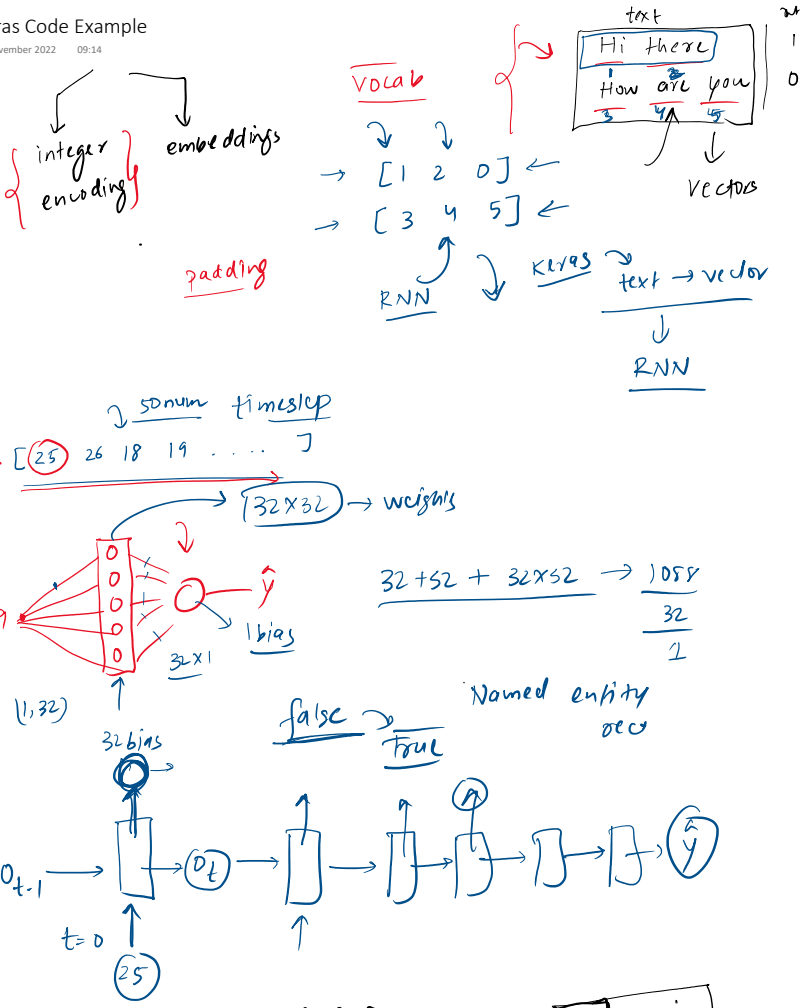
t = time step

$X_{11} \rightarrow$

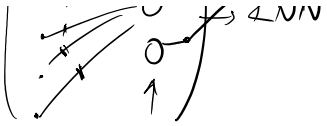
Hello how are you ?

Keras Code Example

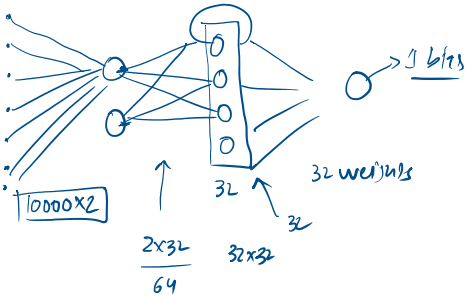
06 November 2022 09:14



In natural language processing, word embedding is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning.



17 nodes



population in 9th year = x

$x + 10\%\text{ of } x = 10000$

$x + 0.1x = 10000$

$1.1x = 10000$

$x = \frac{10000}{1.1}$

$t(n-1)$

$\frac{x-1}{x} + \frac{1}{2}\left(\frac{x-1}{x}\right)^2 + \frac{1}{2}\left(\frac{x-1}{x}\right)^3 + \frac{1}{2}\left(\frac{x-1}{x}\right)^4 + \cdots$

$+2+$

$+(n-1)$

50 → 500

A

A[35] → t sec

A[35] →

35

1 x 4 x35

x

1 25 50 100

12

O(n)

O(n²) → nested loops

input → 1 10 100 1000

time → 1 2 3 4

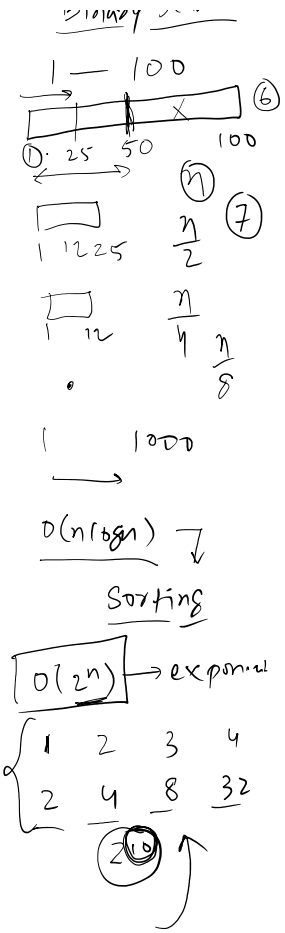
O(log(n))

Binary search

1 — 100

→ (6)

$\sqrt{(x_1$


1 — 100
6
1 25 50 100
7
n/2
7
1 1225
n/4
n/8
1 1000
O(n log n)
Sorting
O(2^n) → exponential
1 2 3 4
2 4 8 32
2^10
for i in range
— O(n)
for j in (n^2)
O(n)
O(n+n) →
O(2n)
→ O(n)
O(n+n^2) O(n^2)
for i in range
for j in
0 1 2 3 4 5 6 7 8 9 n^2
25 → '25'
str()
n = 345 % 10
digits[5]
5 + ''
= '5'
345//10 → 34 != 0
34//10 → 3

34 // 10 → 4

4 + '5' →
45

34 // 10 → 3

3 != 0

3 // 10 => 3

digits[3]

3 + '45'
= '345'

Loop n → 10 times

345 = n ⇒ 3450

3 → 4

inp → 10 → log
time → 1

O(n) → O(n+n)
O(n) O(2n) → O(n)

O(n) → O(n*n)
O(n) O(n²)

O(n)
O(n)
O(10000000)

n² 10000000
×
O(n²)

len(L)/2
1 → n/2

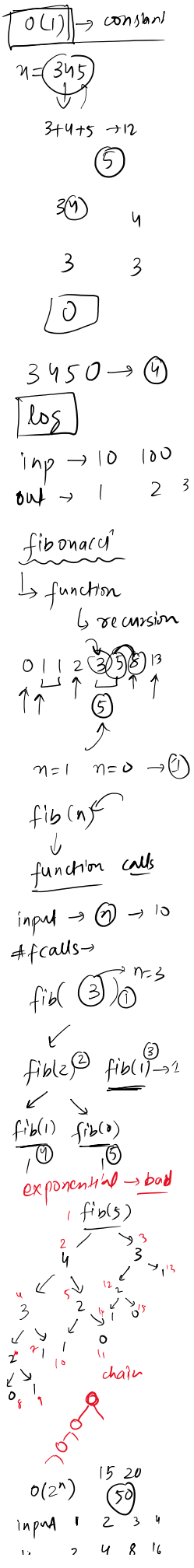
4 n/2 24

O(n) n=100

50, 100 n/2
↳
2 → 100=n n/2

j=1 → 2
j=2 → 4 2-100
j=3 → 8
j=4 → 32

n/2 * logn
nlogn


O(1) → constant
n = 345
3+4+5 → 12
5
3 4
4
3 3
0
3450 → 4
log
inp → 10 100
out → 1 2 3
fibonacci
↳ function
↳ recursion
0 1 1 2 3 5 8 13
5
n=1 n=0 → 1
fib(n)
↓
function calls
input → n → 10
#fcalls →
fib(3) n=3
fib(2) fib(1) → 1
fib(1) fib(0)
exponential → bad
fib(5)
chain
O(2^n)
15 20
50
input 1 2 3 4
1 2 4 8 16

$O(2^n)$ (50)

input 1 2 3 4

ti 2 4 8 16

$n = 50, 100, 500$

↑ weeks

exponential

↳ days/ weeks

Subset → $O(?)$

power set

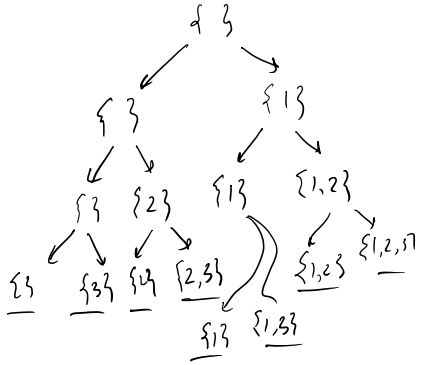
$\{1,2\} \rightarrow \{\{\},\{1\},\{2\},\{1,2\}\}$

$\{1,2,3\} \rightarrow$

$\{\},\{1\},\{2\},\{3\},\{1,2\}\{2,3\}$

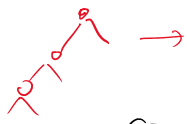
$\{3,1\}\{1,2,3\}$

$\{1,2,3\}$



reduce → divide → log

Increase → multi → exp



→ exponential

$\{1,2\} \rightarrow 4$ $\quad 2^2 = 4$

$2^3 = 8$

$\{1,2,3\} \rightarrow 8$ $\quad 2^4 = 16$

$O(2^n)$

$O(?)$

$$T(n) = \begin{cases} 3T(n-1) \text{ if } n>0 \\ 1, \text{ otherwise} \end{cases}$$

$n > 0$

$T(n) = 3T(n-1)$

$= 3[3T(n-2)]$

$= 3^2 T(n-2)$

$= 3^2[3T(n-3)]$

$= 3^3 T(n-3)$

$= 3^n T(n-n)$

$= 3^n T(0)$

$T(n) = \boxed{3^n} \rightarrow O(3^n)$

$$T(n) = \begin{cases} 2T(n-1)-1 & \text{if } n>0 \\ 1, \text{ otherwise} \rightarrow \text{constant} \end{cases}$$

$=$

$T(n) = 2T(n-1)-1$

$= 2[2T(n-2)-1]-1$

$= 2^2 T(n-2) - 2 - 1$

$= 2^2[2T(n-3)-1] - 2 - 1$

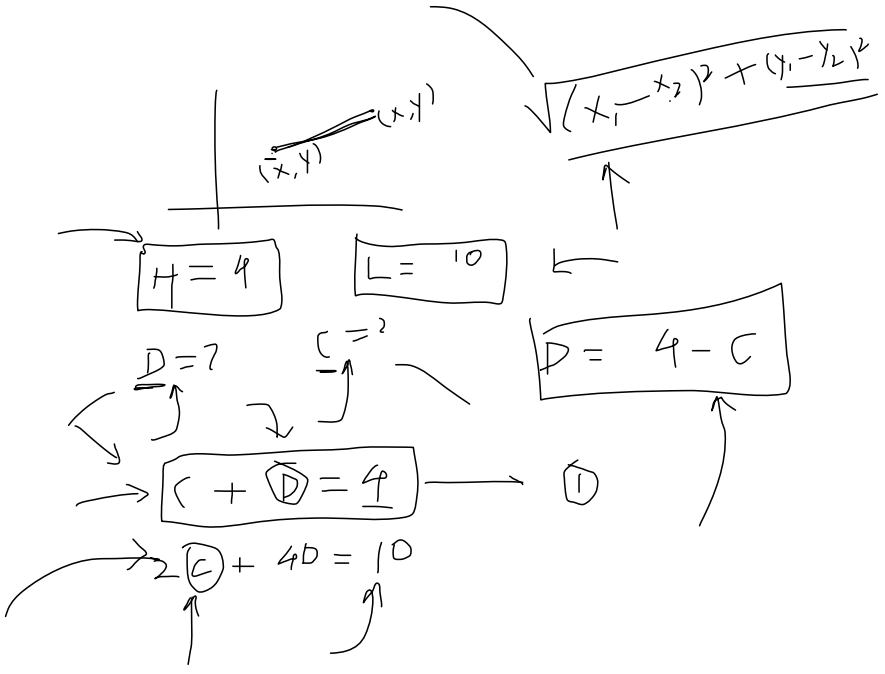
$= 2^3 T(n-3) - 2^2 - 2^1 - 2^0$

$= 2^n T(n-n) - 2^{n-1} - 2^{n-2} \ldots - 2^1 - 2^0$

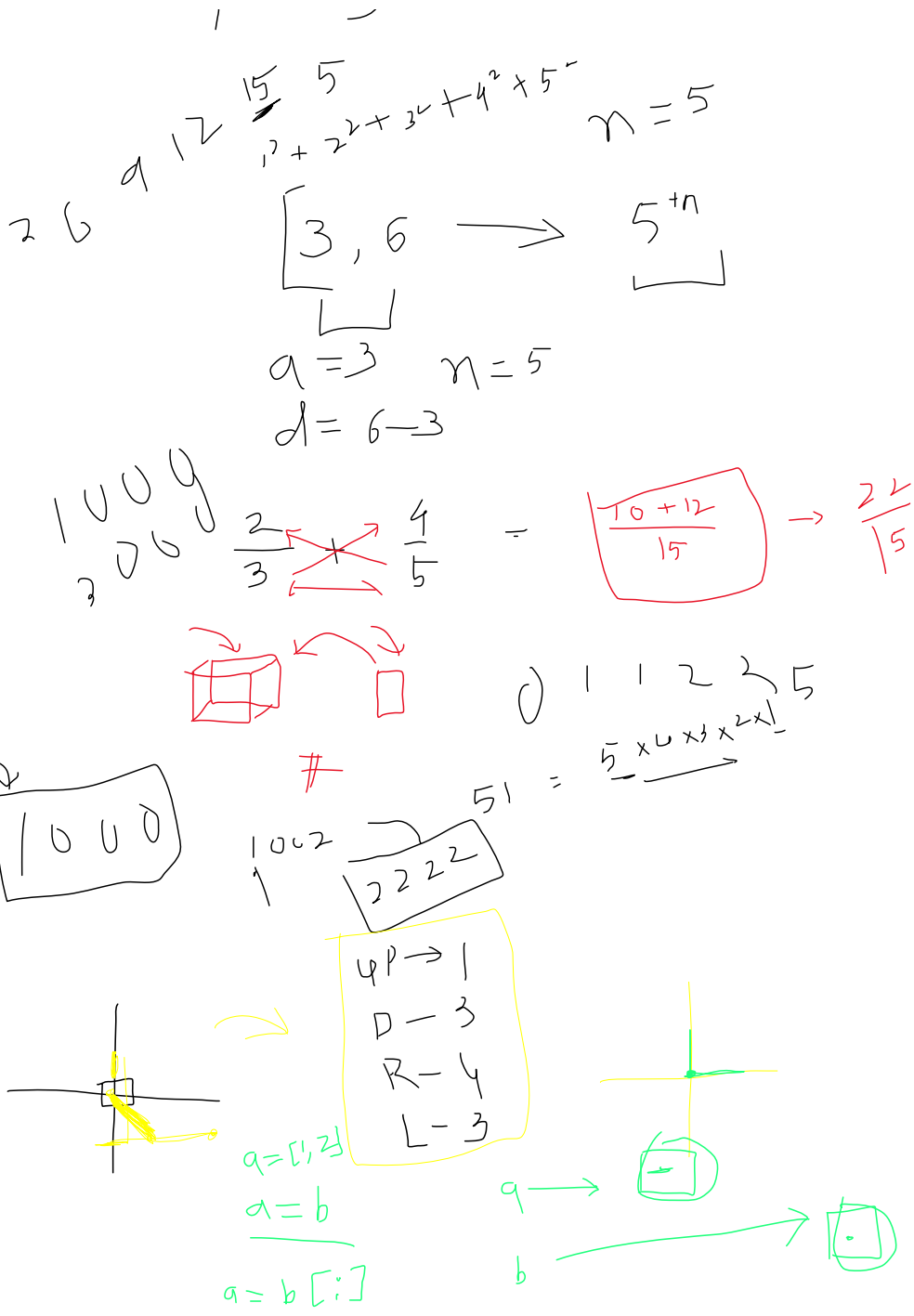
$= 2^n - [2^{n-1} + 2^{n-2} + \ldots + 2^1 + 2^0]$

$= 2^n - [2^n - 1] = 2^n - 2^n + 1$

$O(1) \rightarrow$ constanst

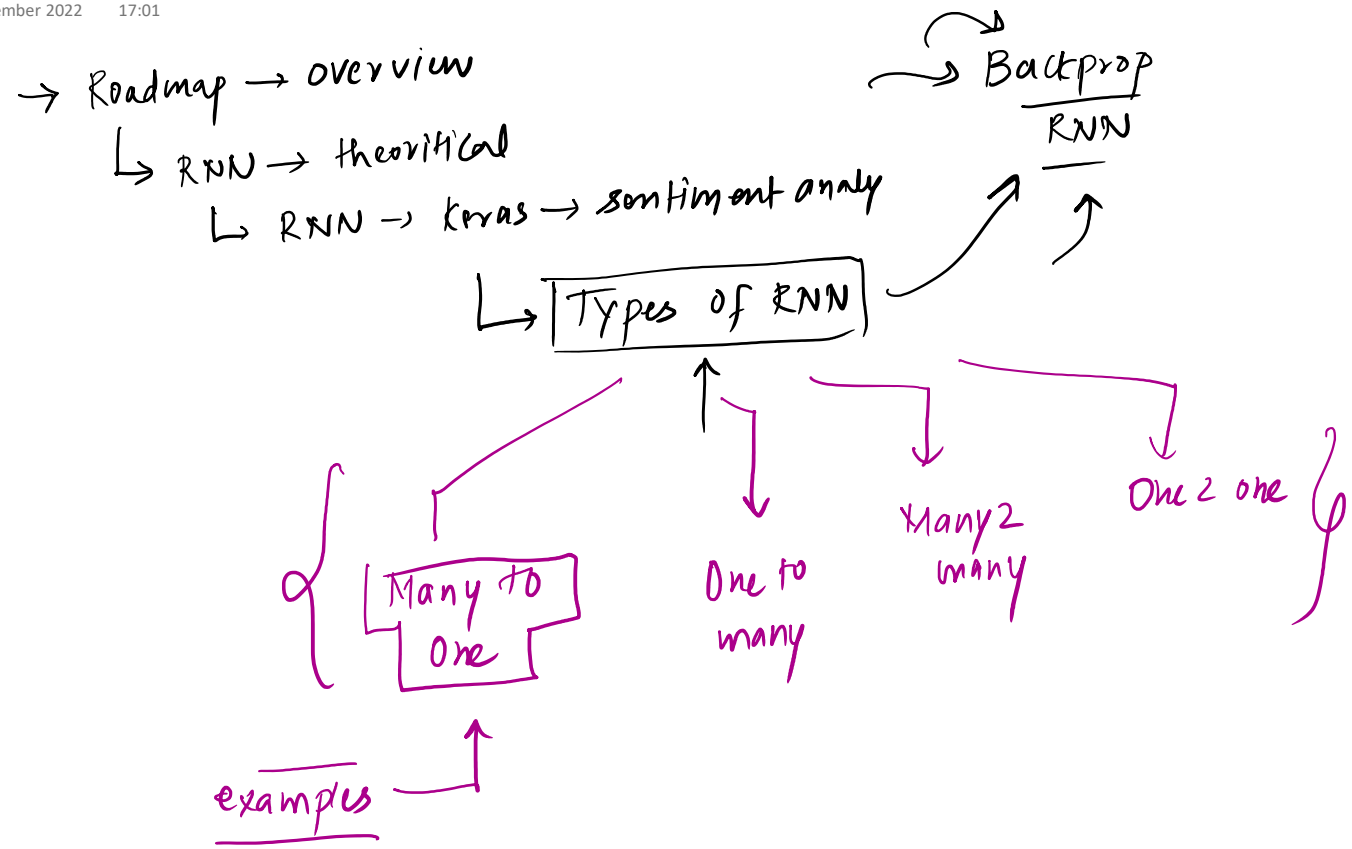


15 5


n = 5
3, 6
5^n
a = 3
n = 5
d = 6 − 3
10 + 12
15
22
5
0 1 1 2 2 5
1000
1002
2222
51
up → 1
D − 3
R − 4
L − 3
a = [1, 2]
a = b
a = b [:]
a
b

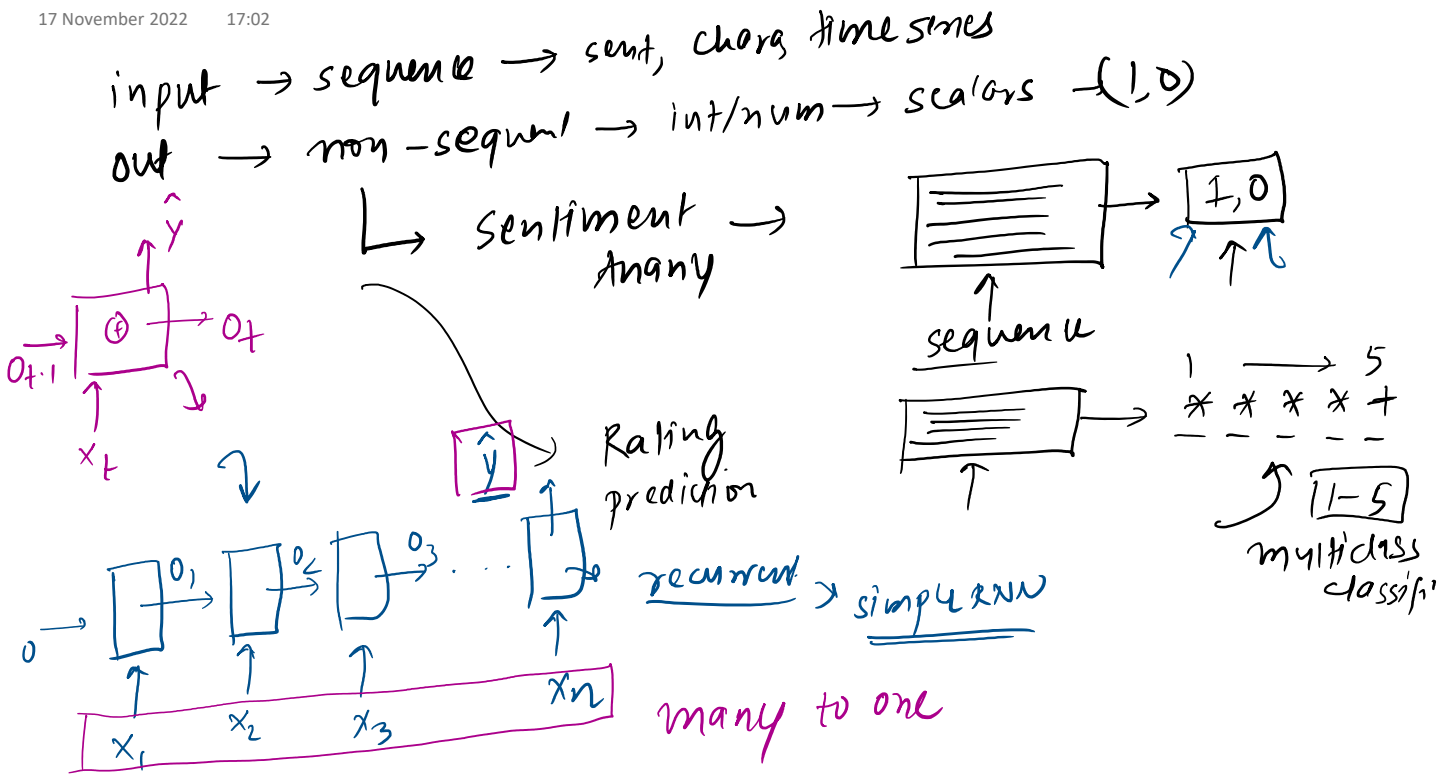
Till Now

17 November 2022 17:01



Many to One

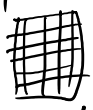
17 November 2022 17:02



One to Many

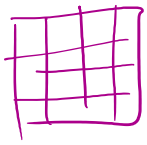
17 November 2022 17:02

→ normal non sequential

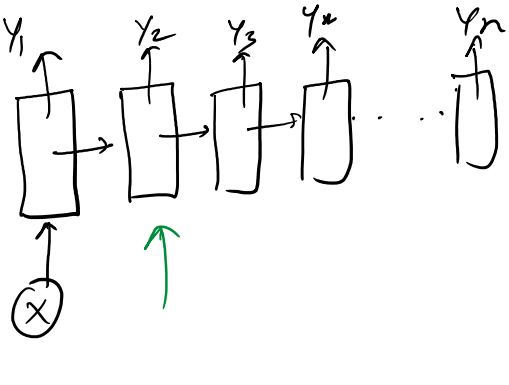
↳ 

→ Output → sequences

image captioning



→ textual depres

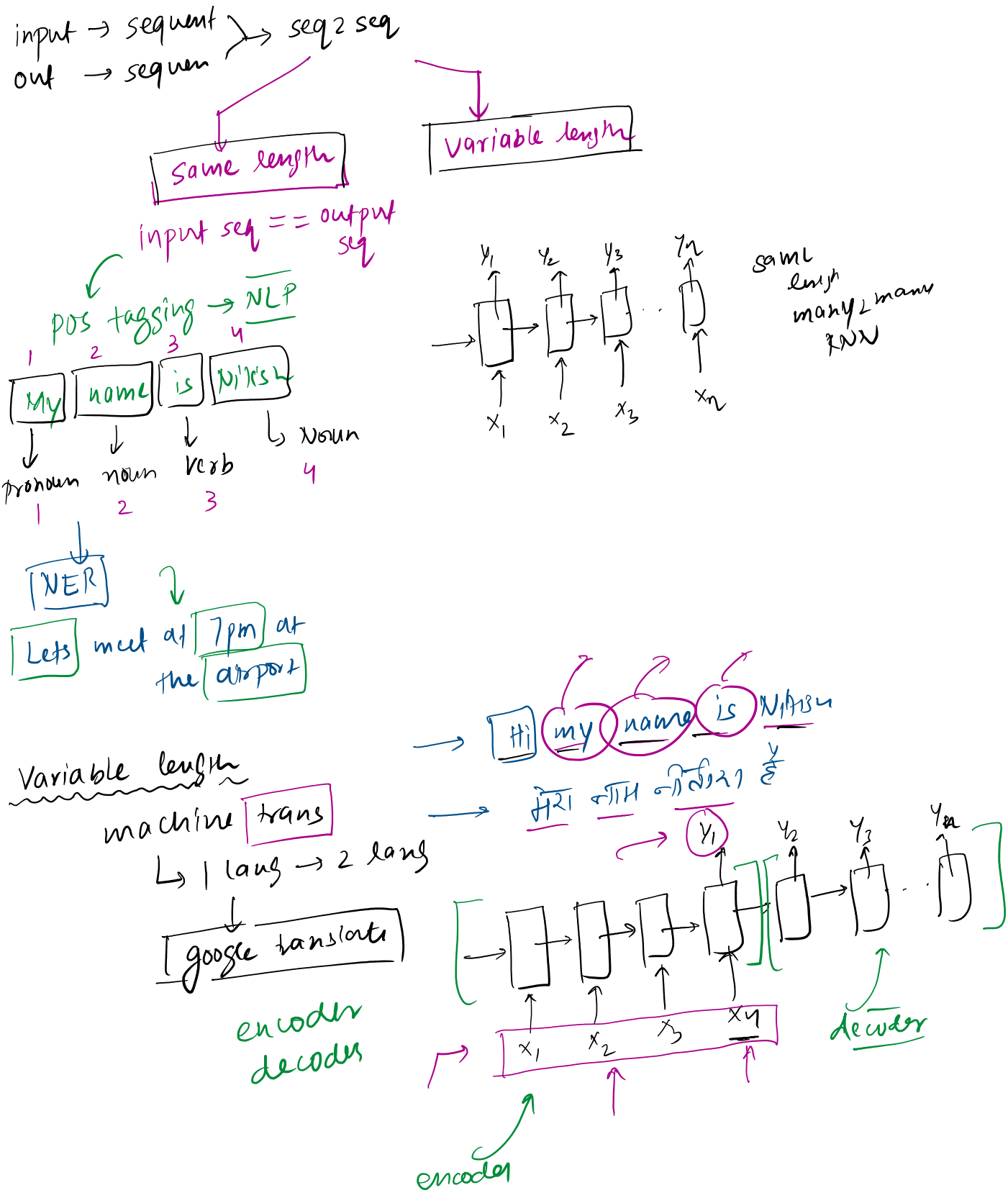


→ NN → A man is playing cricket

→ music generation

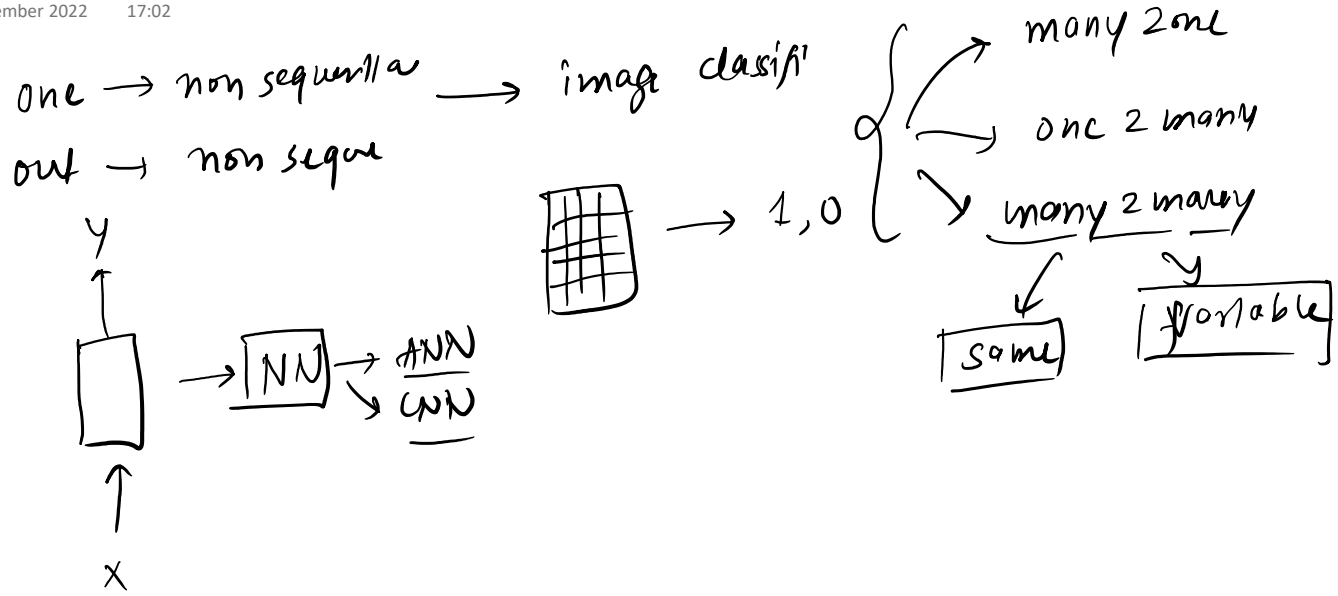
Many to Many

17 November 2022 17:02



One to One

17 November 2022 17:02



Backpropagation in RNN

01 December 2022 16:43

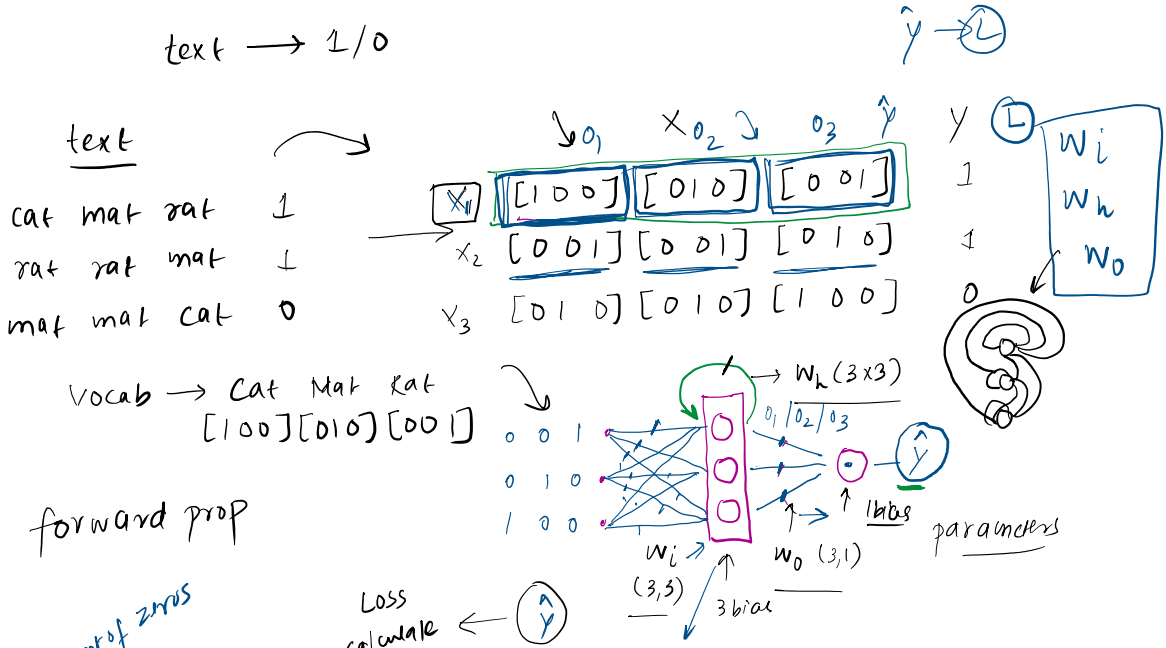
RNN Intro → RNN → practical → types of RNN

→ RNN → Backprop → BPTT

Many to One RNN

Sentiment Analysis

text → 1/0



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

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

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

$$\hat{y} = \sigma(O_3 W_0)$$

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

$$W_i = W_i - \eta \frac{\partial L}{\partial W_i}$$

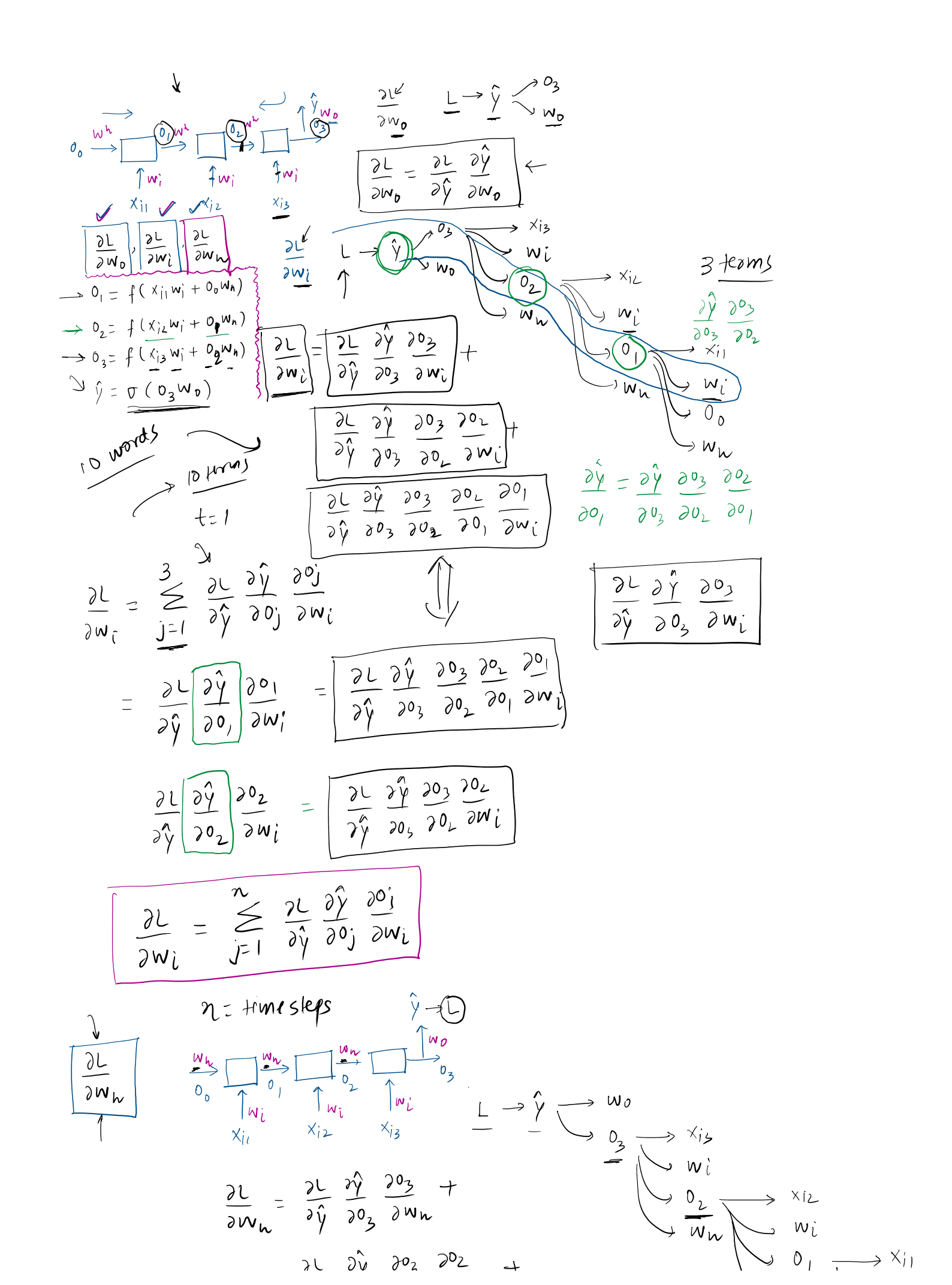
$$W_h = W_h - \eta \frac{\partial L}{\partial W_h}$$

$$W_0 = W_0 - \eta \frac{\partial L}{\partial W_0}$$

W_i W_h W_0

$\frac{\partial \hat{y}}{\partial W_0}$ $\frac{\partial L}{\partial \hat{y}}$ $\frac{\partial L}{\partial W_0}$

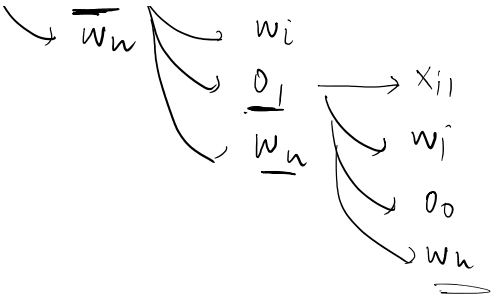
L → $\hat{y}$ → O_3, W_0


3 terms
10 words
10 terms
t=1
n = timesteps

∂w_h ∂y ∂o_3 ...

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

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



$$\frac{\partial L}{\partial w_h} = \sum_{j=1}^{n} \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_j} \frac{\partial o_j}{\partial w_h}$$

n = timesteps

for $j=3$

$$\frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial w_h} \rightarrow \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial o_1} \frac{\partial o_1}{\partial w_h}$$

for $j=10$

$$\frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_{10}} \frac{\partial o_{10}}{\partial o_1} \frac{\partial o_1}{\partial w_h}$$

$$\prod_{t=2}^{j} \frac{\partial o_t}{\partial o_{t-1}}$$

$$\prod_{t=2}^{t=3} \frac{\partial o_t}{\partial o_{t-1}} = \frac{\partial o_2}{\partial o_1} \frac{\partial o_3}{\partial o_2}$$

$$o_t = f(x_{it} w_{inp} + o_{t-1} w_h)$$

$$\frac{\partial o_t}{\partial o_{t-1}} = \prod_{t=2}^{j} f'(x_{it} w_{inp} + o_{t-1} w_h) w_h$$

$\downarrow$

$[0-1]$

Problem with RNN

19 December 2022 16:33

RNNs → Sequential data → textual, time series

Suffer (2) major problems → LTSMs (Why)

RNN

→ Problem of long term dependency
→ Unstable gradients Stagnated training

→ Unstable gradients

Applications next word predictor

→ RNN

timesteps

Marathi is spoken in Maha

short

Maha is a beautiful place ± went then last year, but I could not enjoyed proper bcz ± dont understa marathi

long term

RNN

keypad

RNN

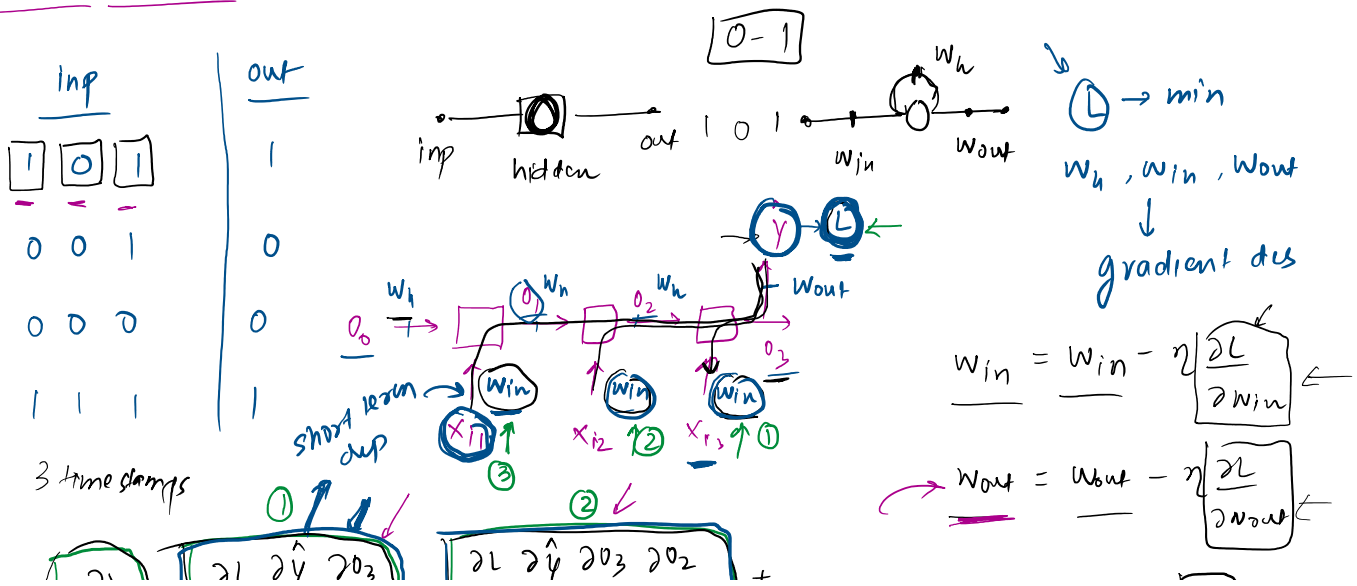
{ Vanishing gradient prob }

{ Exploding gradient problem } → RNN

possible solⁿ

↳ remove

Problem #1 → Problem of long term dependency → Vanishing



$$\frac{\partial L}{\partial w_{in}} = \frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial o_3}\frac{\partial o_3}{\partial w_{in}} + \frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial o_3}\frac{\partial o_3}{\partial o_2}\frac{\partial o_2}{\partial w_{in}} + \frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial o_3}\frac{\partial o_3}{\partial o_2}\frac{\partial o_2}{\partial o_1}\frac{\partial o_1}{\partial w_{in}}$$

(1) (2) (3) → long term dep

inf

$$w_h = w_h - \eta \frac{\partial L}{\partial w_h}$$

→ 3

100 time steps

$$\frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial o_{100}}\frac{\partial o_{100}}{\partial o_{99}} \cdots \cdots \frac{\partial o_2}{\partial o_1}\frac{\partial o_1}{\partial w_{in}} \rightarrow \frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial o_{100}}\prod_{t=2}^{100}\left(\frac{\partial o_t}{\partial o_{t-1}}\right)\frac{\partial o_1}{\partial w_{in}}$$

$$\frac{\partial o_2}{\partial o_1}\frac{\partial o_3}{\partial o_2}\frac{\partial o_4}{\partial o_3} \cdots \frac{\partial o_{100}}{\partial o_{99}}$$

$$o_1 = \tanh(x_{i1} w_{in} + o_0 w_h)$$

$$o_t = \tanh(x_{it} w_{in} + o_{t-1} w_h)$$

$$\frac{\partial o_t}{\partial o_{t-1}} = \tanh'(x_{it} w_{in} + o_{t-1} w_h)\, w_h$$

0 - 1 (relu) ; 0 - 1 (1) ; identity matrix

Vanishing grad

100 or more

$$\frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial o_{100}}\prod_{t=2}^{100}\left(\tanh'(\ )\, w_h\right)\frac{\partial o_1}{\partial w_{in}} \simeq 0 \simeq 0$$

(0-1) (0-1)

Sol(n)

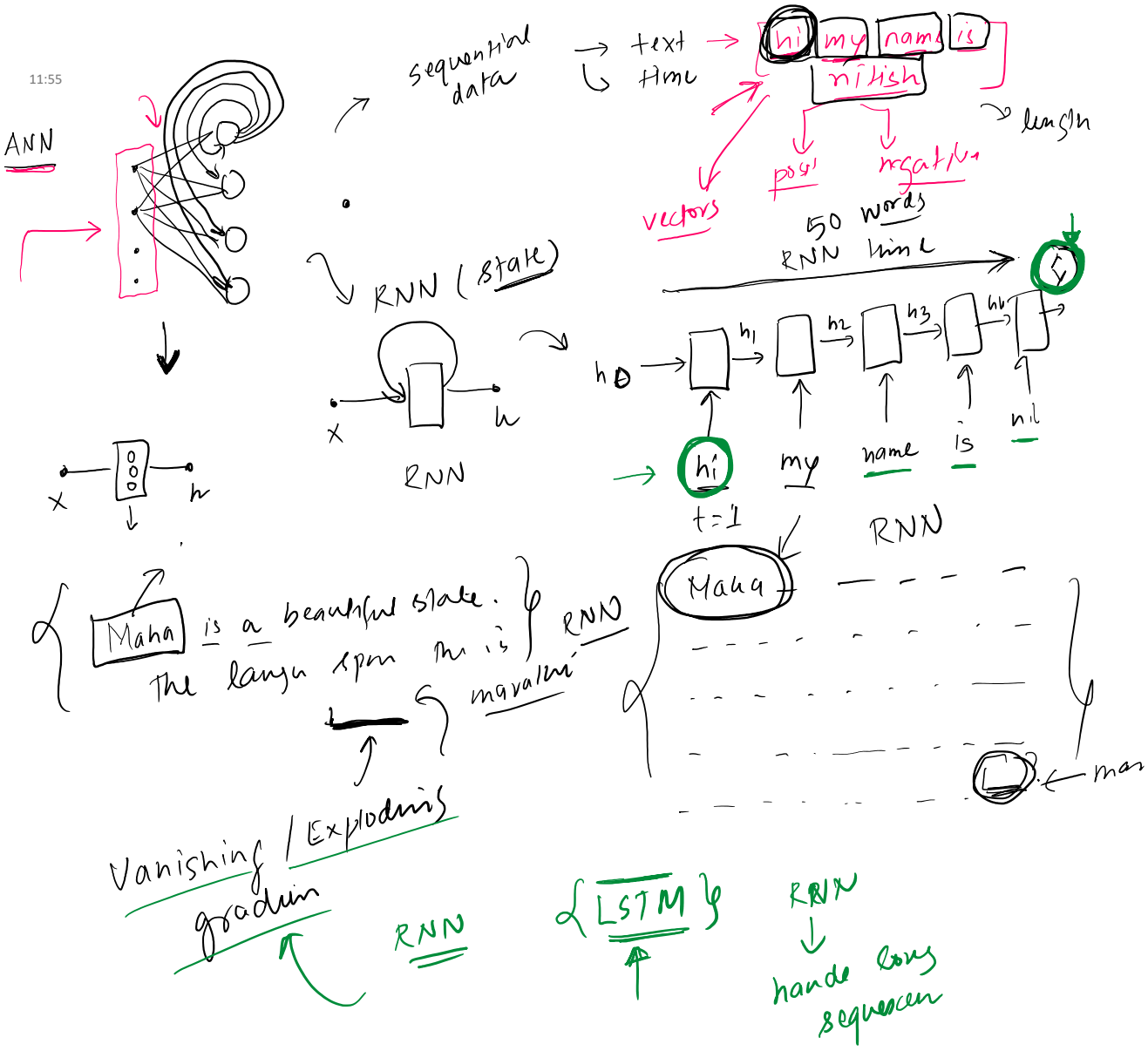
1) Diff activation → relu / leaky relu
2) Better weight init
3) Skip conns
4) LSTM

Problem #2 → Unstable Training (Exploding gradients)

1) Gradient Clipping
2) Controlled learning rate
3) LSTM

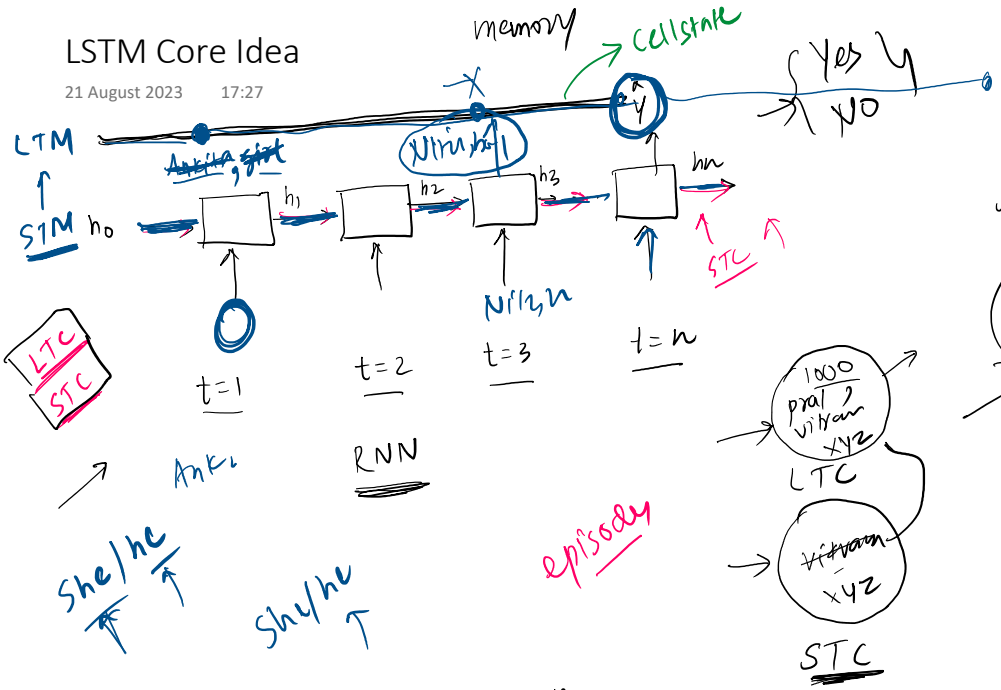
Recap

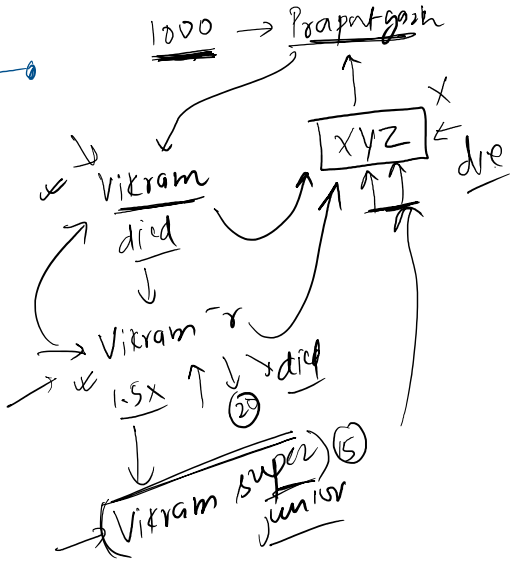
21 August 2023 11:55



LSTM Core Idea

21 August 2023 17:27

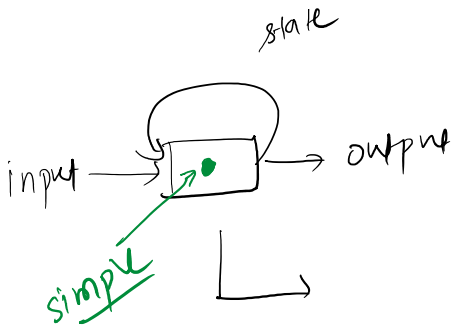


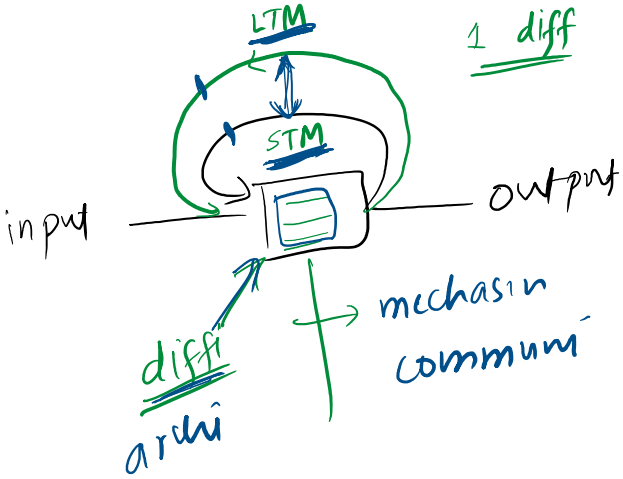


She/he

She/he

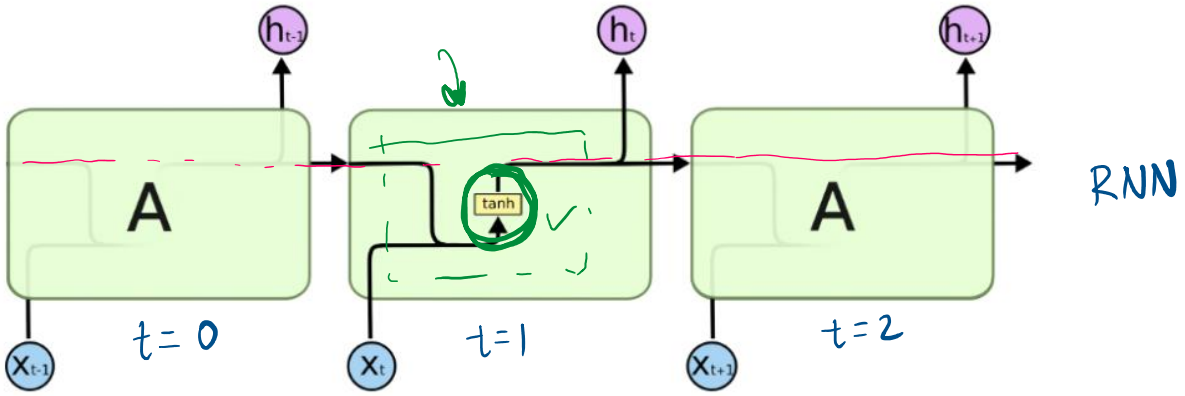
episode

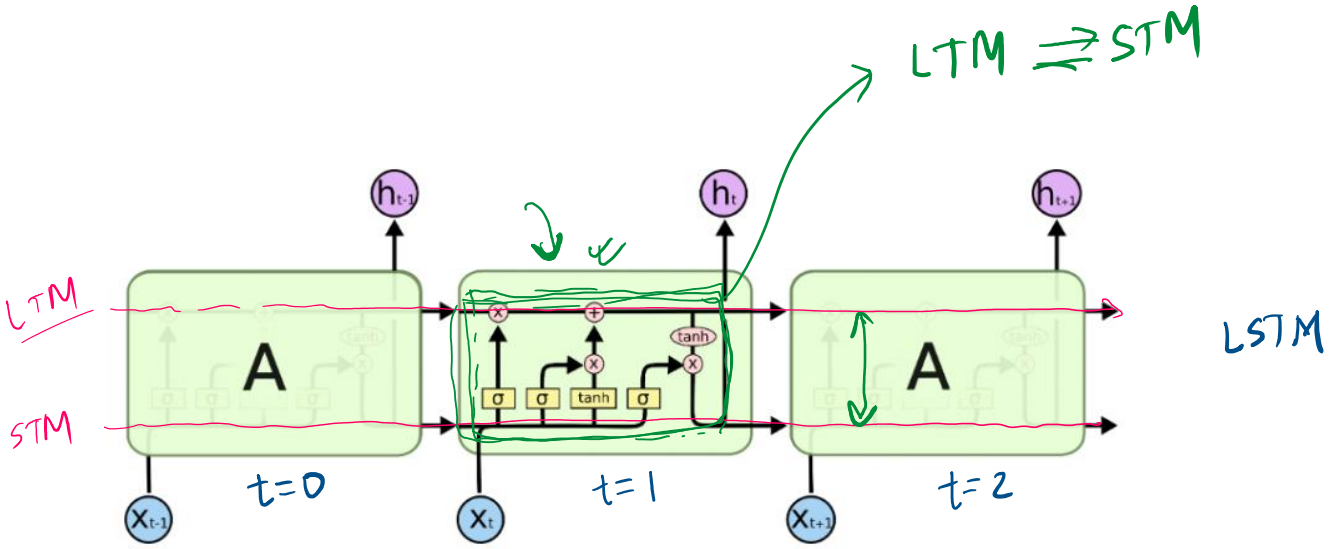


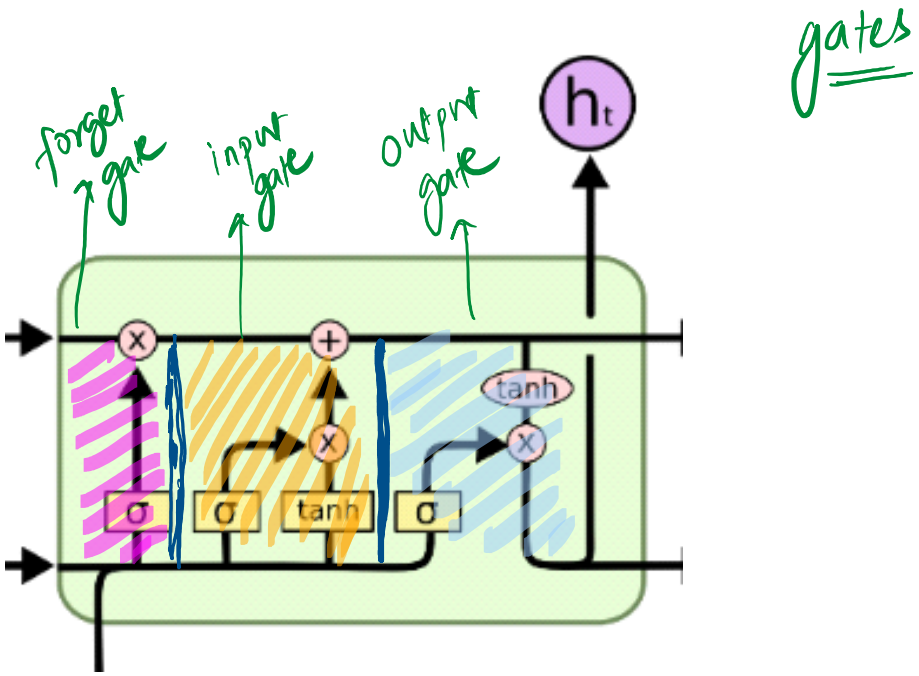


LSTM Architecture

21 August 2023 18:41

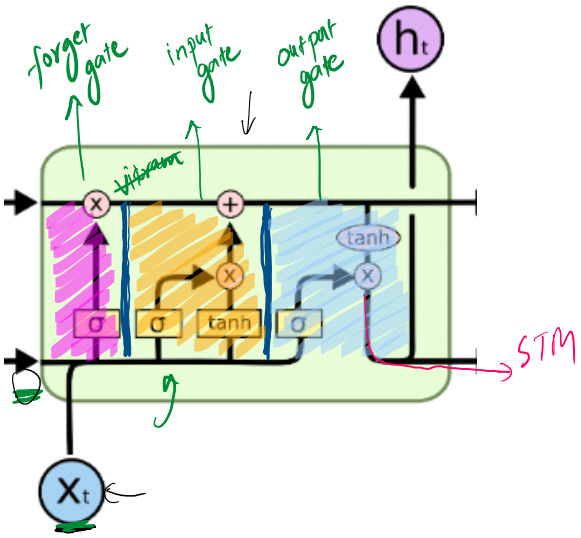






LSTM Gates

21 August 2023 19:06

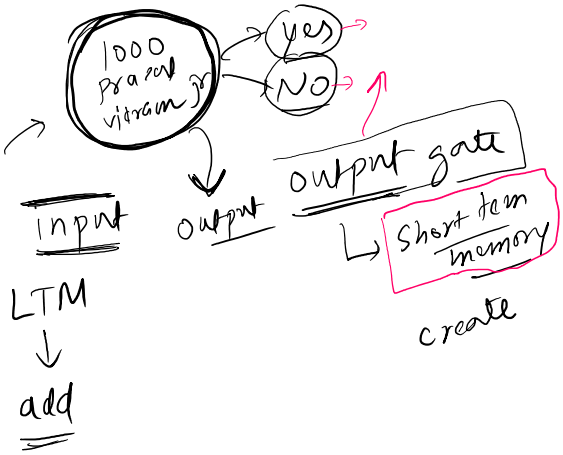


forget

input

LTM

remove



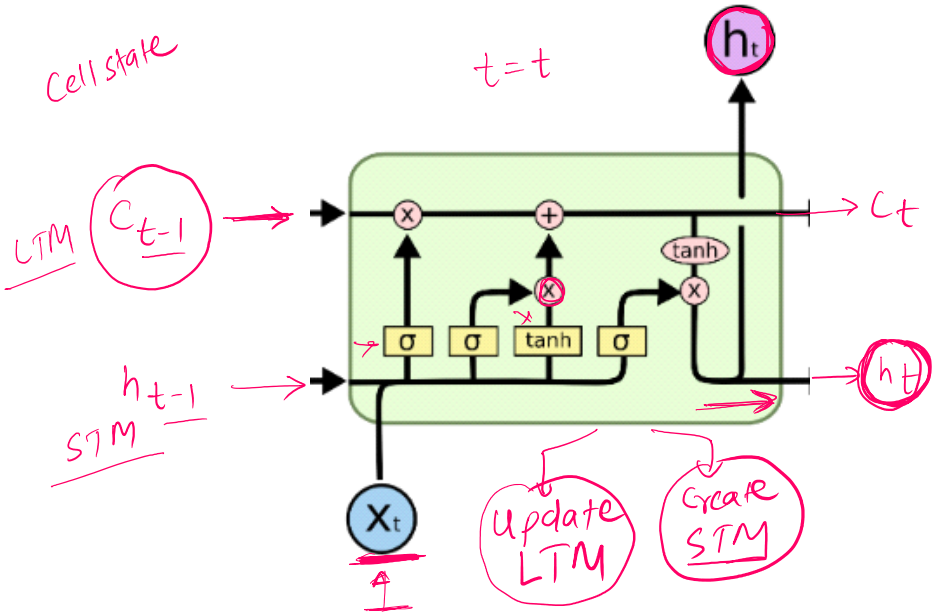
LTM

↓

add

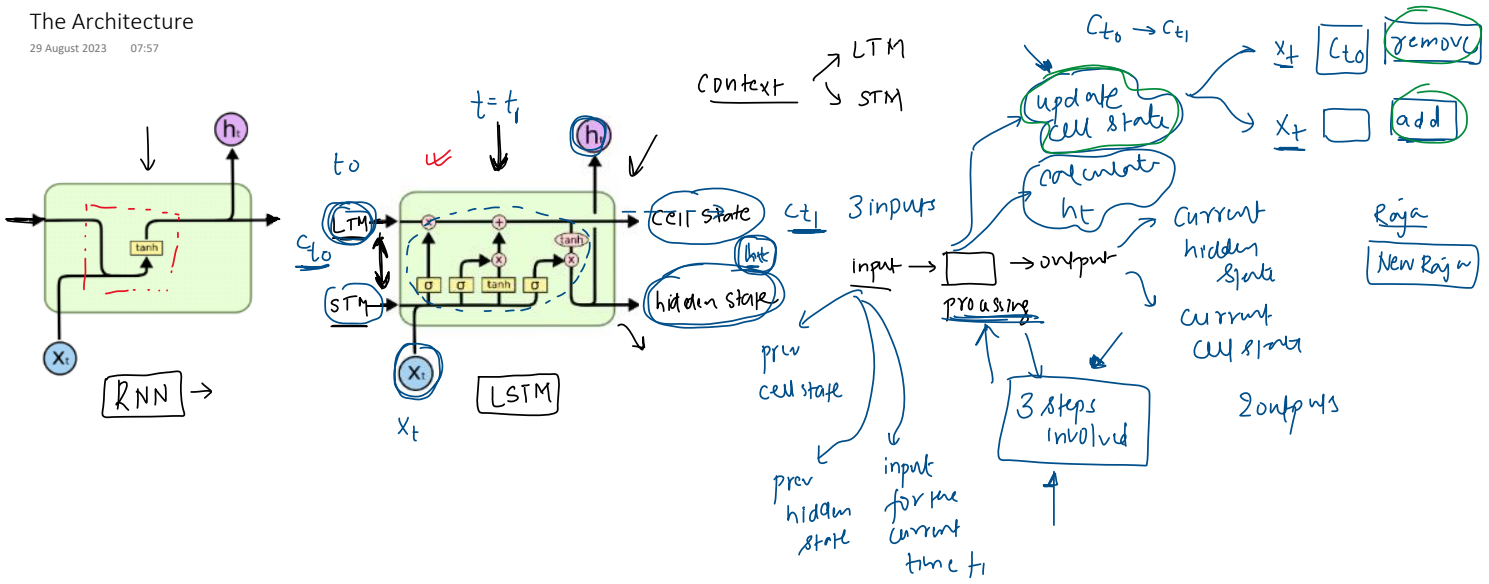
Summary

21 August 2023 19:29



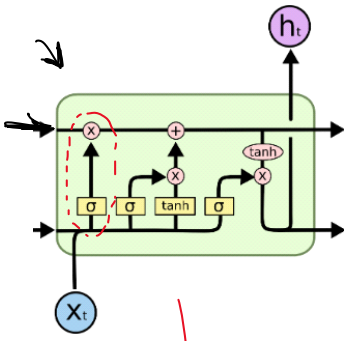
The Architecture

29 August 2023 07:57

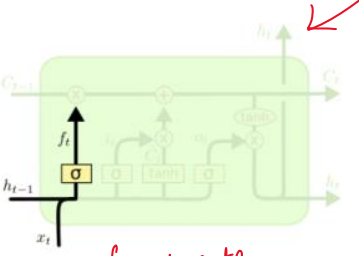


The Gates

29 August 2023 08:09



3 gates

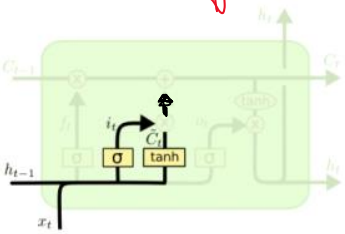


forget gate

to remove something from C_t

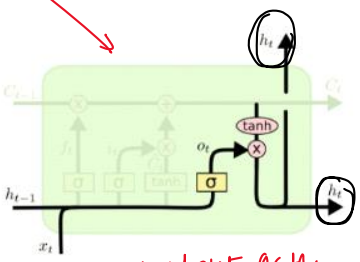
update C_t

+



input gate

to add something to the C_t

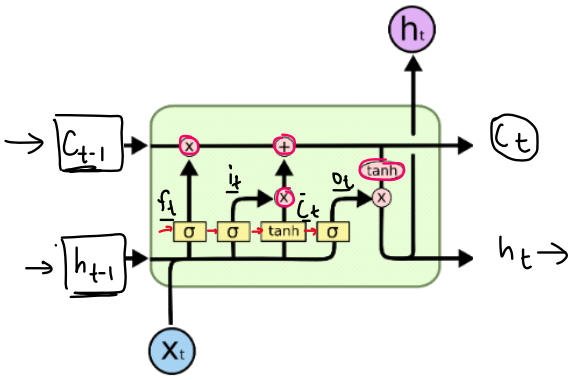


output gate

Calculat hidden state h_t

What are C_t and h_t

29 August 2023 08:08



h_t C_t → [0.1 0.3 0.9]

vectors

3d vector

rule

same

h_t C_t dim equal

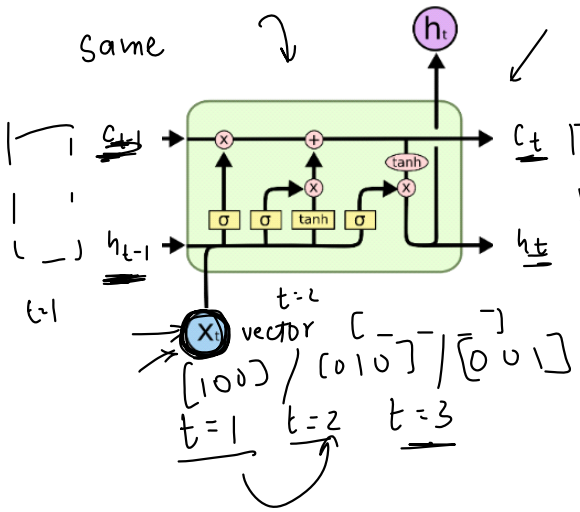
h_t [0.1 0.45 0.6]

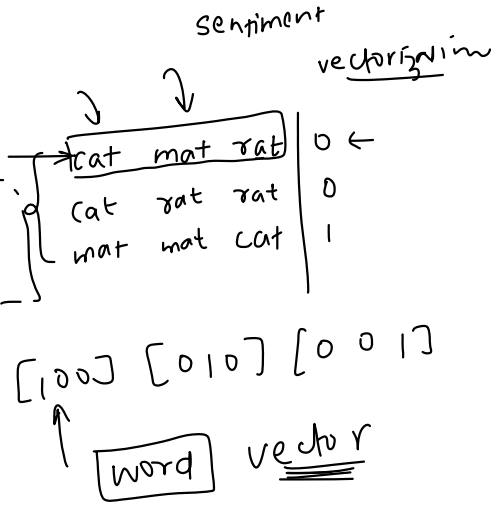
C_t [0.55 0.6 0.0]

What is X_t

29 August 2023 17:40

RNN



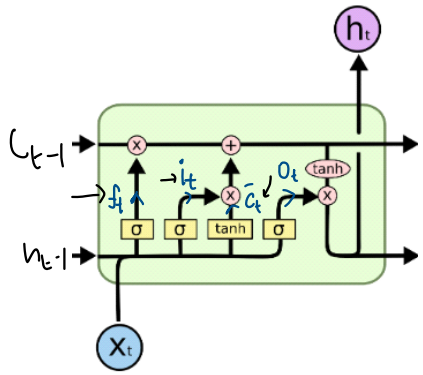


OHE

Cat	mat	rat
1	0	0
0	1	0
0	0	1

What are f_t, i_t, o_t and $\bar{C}t$

29 August 2023 08:09



f_t i_t $\bar{C}_t$ → candidate cell state O_t

f_t forget gate

i_t $\bar{C}_t$ input gate

O_t output gate

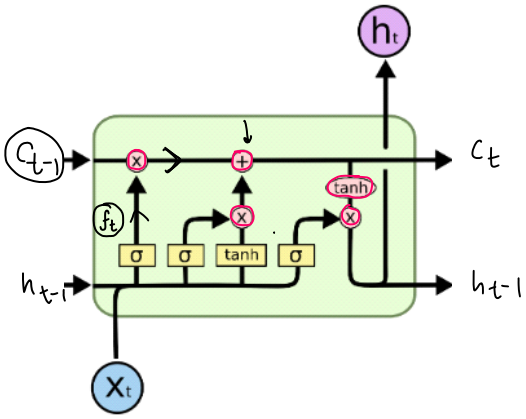
vectors → C_t h_t

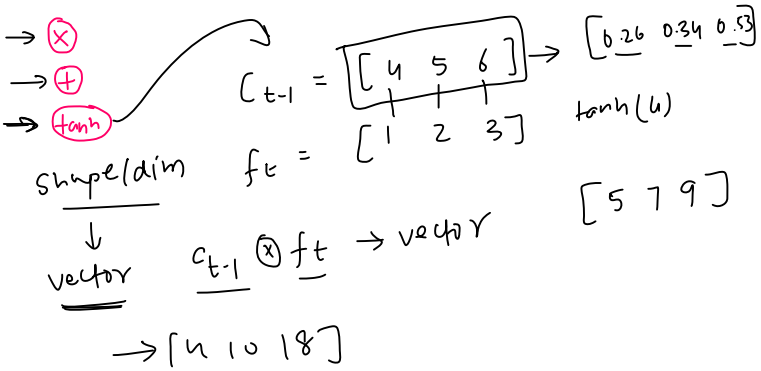
f_t i_t $\bar{C}_t$ O_t

[x y z]
[ ]

Pointwise Operations

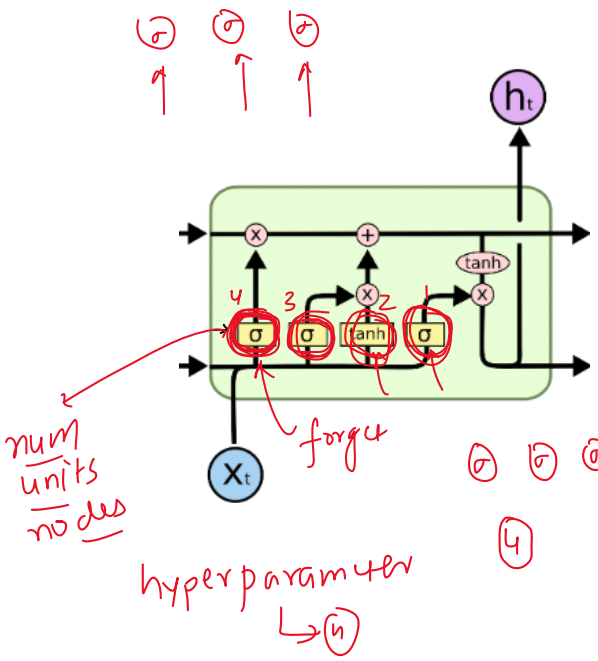
29 August 2023 18:26

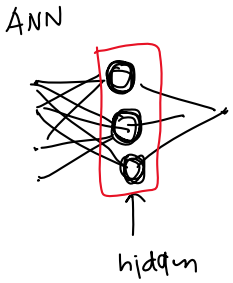


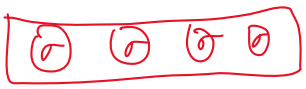


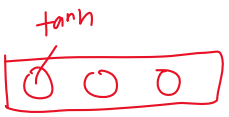
→Neural Network Layers

29 August 2023 18:34





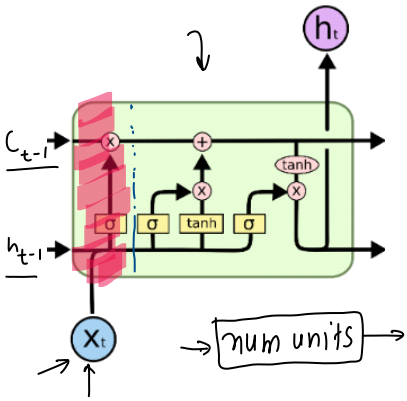




hyperparameter → 4

The Forget Gate

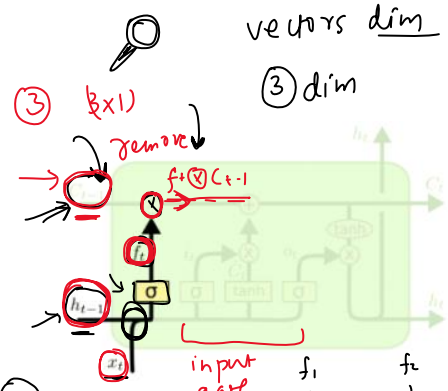
29 August 2023 19:58



→ num units → 3

4 dim vector

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



vectors dim

3 dim

3 (3×1) remove

$f_t \otimes C_{t-1}$

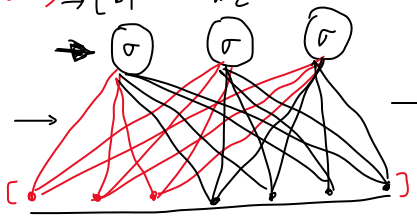
input gate (add)

f_1 f_2 f_3 → $[f_1\ f_2\ f_3]$ → f_t (3)

b_1 b_2 b_3 (3×1)

(3×7) (7×1)

→ (3×1) + (3×1) = (3×1)



→ $W_{3\times7}$ (21 weights) 21 wts

biases 4 biases

h_{t-1} (3) x_t (4) → 7×1

input → output → Cell state → remove

(3×1) ⊗ (3×1)

σ σ σ

→ 1) Calculate f_t

2) $C_{t-1} \otimes f_t$

forget gate ← 0% 50% 100%

remove

$[h_{t-1}, x_t]$

f_t

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

(3×1) (3×7) (7×1) (3×1) + (3×1) → (3×1) → (3×1)

long term

$[4\ 5\ 6]$

C_{t-1} f_t $[0\ 0\ 0]$ $\left[\frac{1}{2}\ \frac{1}{2}\ \frac{1}{2}\right]$

$[1\ 1\ 1]$

$[2\ 2.5\ 3]$

$[4\ 5\ 6]$

$[0\ 0\ 0]$ → 100% remove

removal

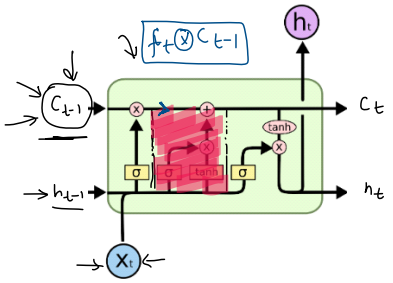
$f_t \otimes C_{t-1}$ → (3×1)

(C_{t-1}) → remove

The Input Gate

30 August 2023 04:38

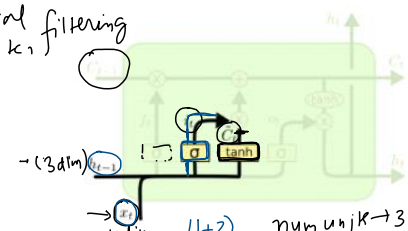
add some new imp info to c_t



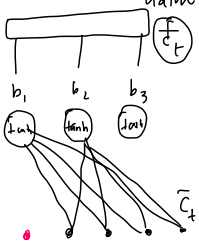
Stages

1) $\tilde{C}_t$ candidate cell state
2) i_t
3) C_t → current cell state

potential filtering important info



(3dim) (1dim) (1+2) num units → 3



$\tilde{C}_t$ = (3x1)

$[\ ]$

→ W_C (3x7)

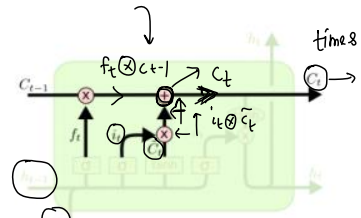
$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$

(3x7) (7x1) (3x1)

(3x1) (3x1)

(3x1)

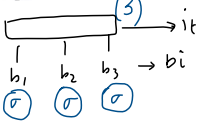
h_{t-1} x_t



timestep

pointwise $i_t \otimes \tilde{C}_t \rightarrow \tilde{C}_t^*$ (filtered candidate cell state)

(3x1) (3x1)



→ $[\ ]$ W_i

$[0.5\ 0.5\ 0.5]$ $[4\ 5\ 6]$

1 1 1 ↓ 50% 0% 100%

$[2, 2.5\ 3]$

$C_t = f_t \otimes C_{t-1} \oplus i_t \otimes \tilde{C}_t$

h_{t-1} x_t

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

(3x7) (7x1) (3x1)

(3x1) (3x1)

σ(3x1) → (3x1)

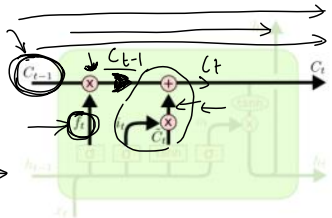
i_t - filter

$\tilde{C}_t \rightarrow C_t$

10%

$[4\ 5\ 6]$ → $[4\ 5\ 6]$

$t = t-1$ $t = t$



$f_t = [1\ 1\ 1]$

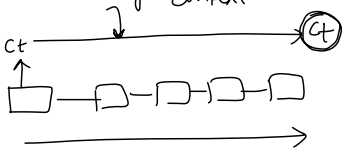
$i_t \otimes \tilde{C}_t = [0\ 0\ 0]$

$[4\ 5\ 6]$ $[0\ 0\ 0]$

↓

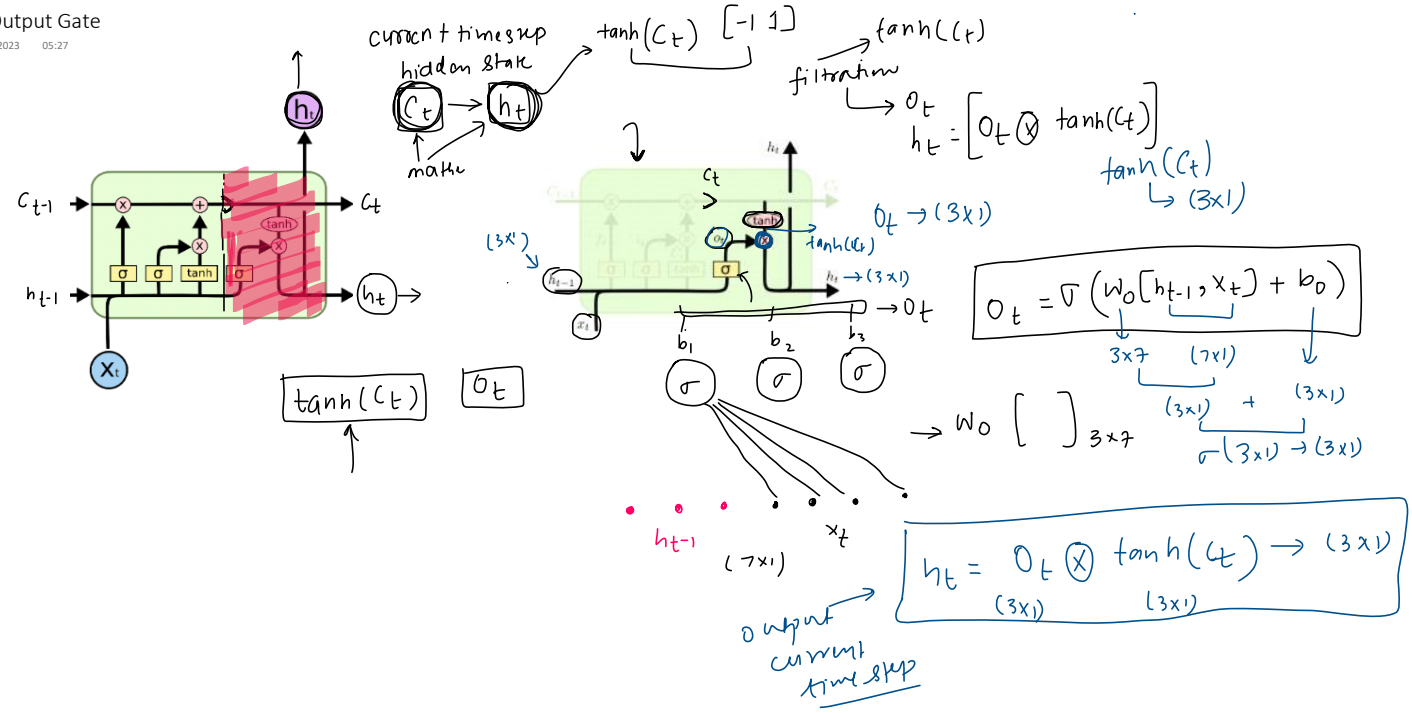
$C_t = [4\ 5\ 6]$

long term context



The Output Gate

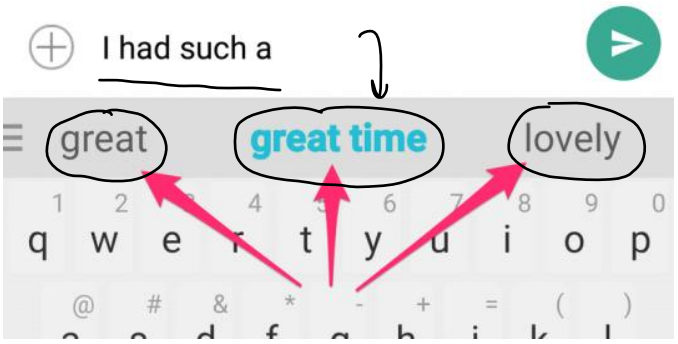
30 August 2023 05:27

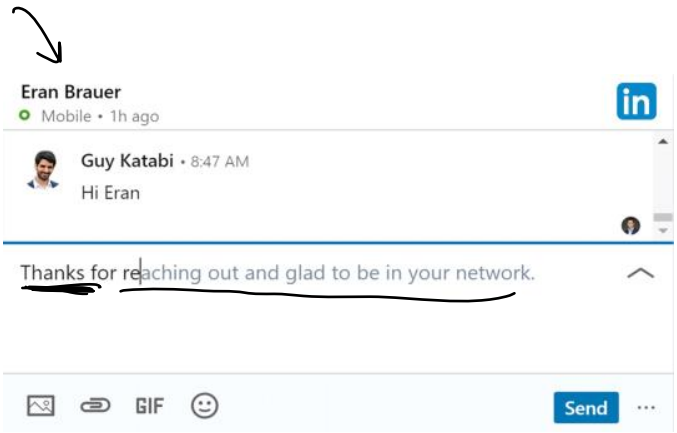


What is a Next Word Predictor

08 September 2023 08:50

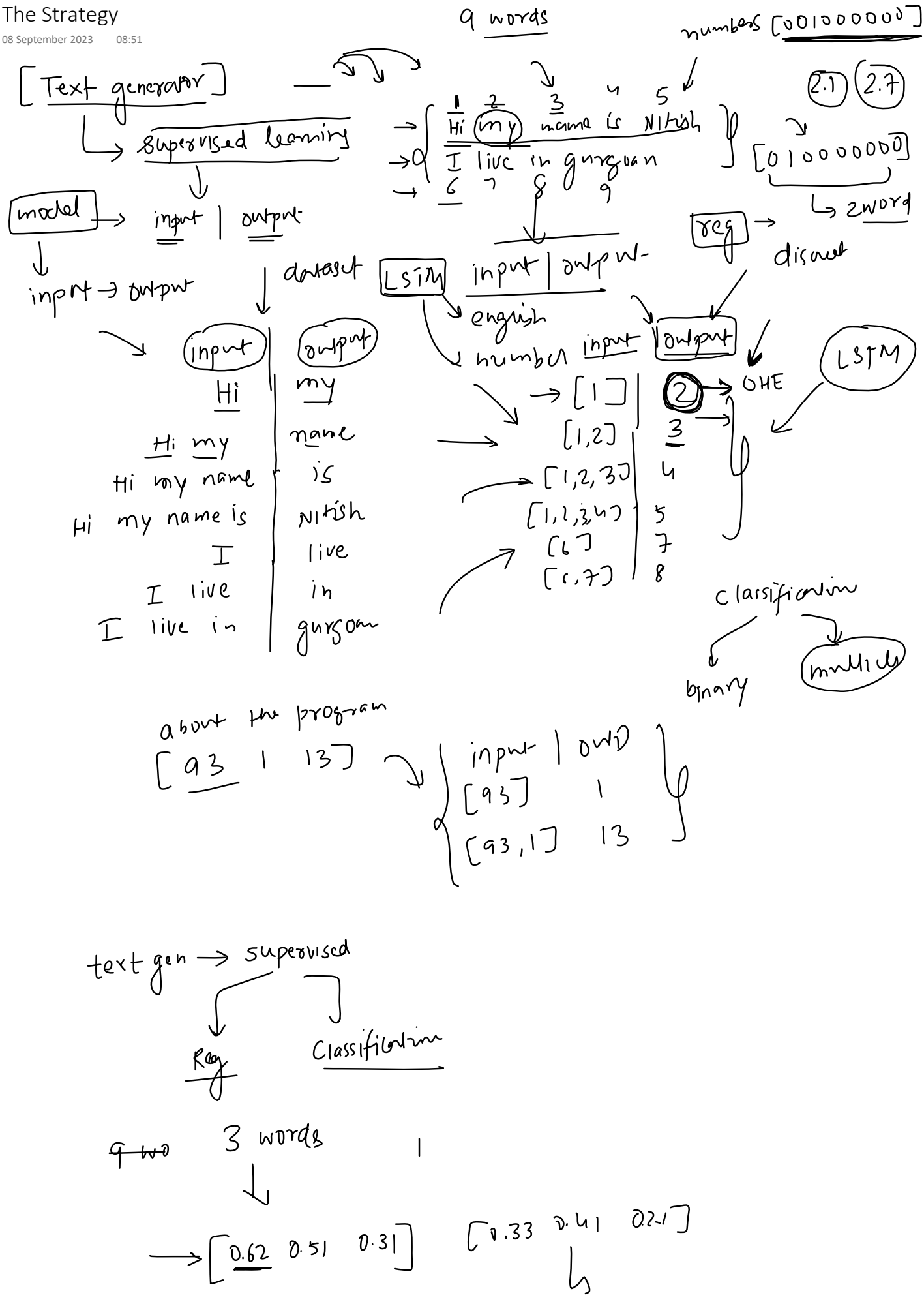
code

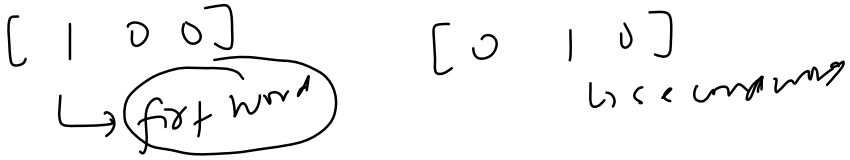




The Strategy

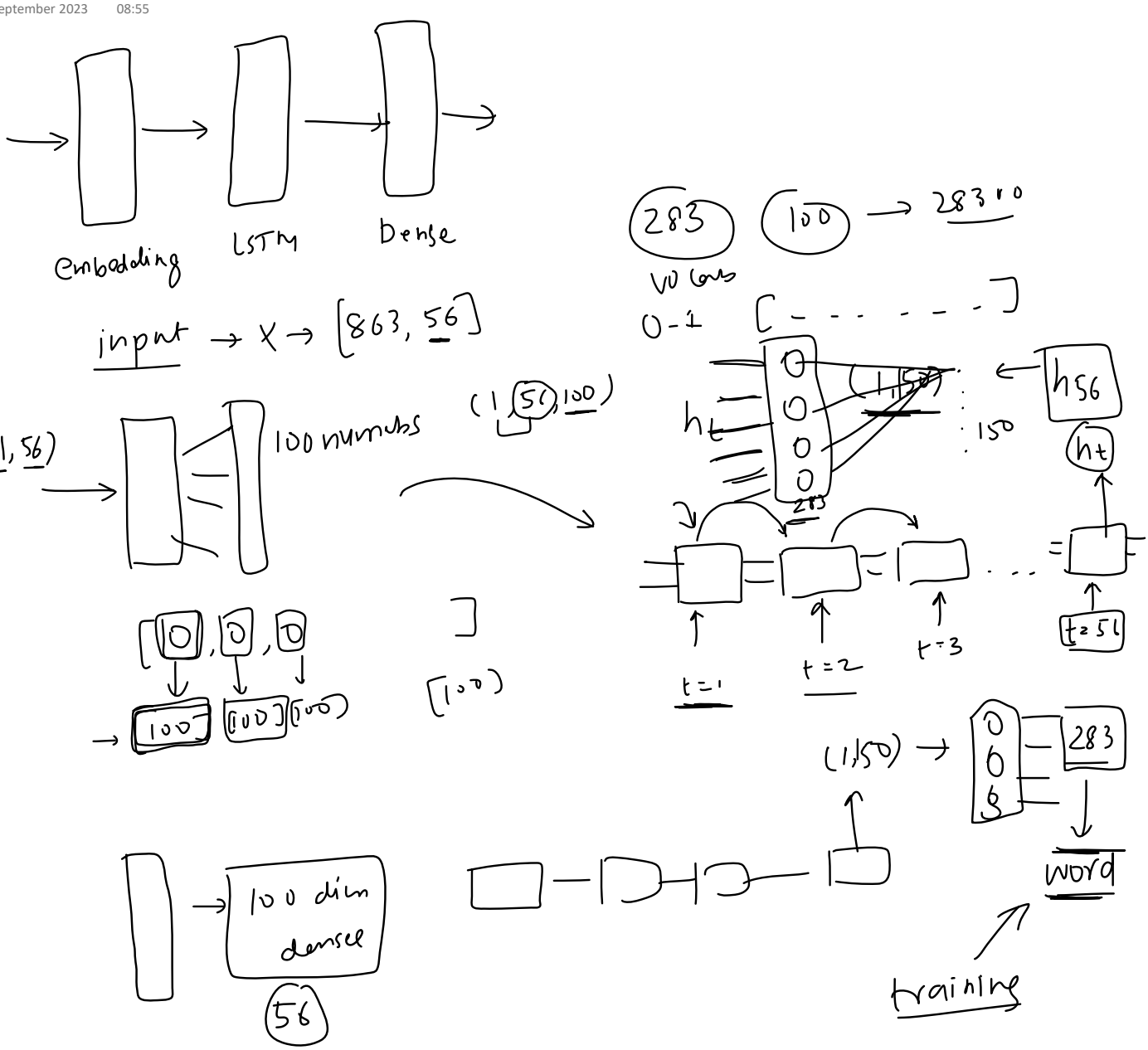
08 September 2023 08:51




[1 0 0]
first word
[0 1 0]

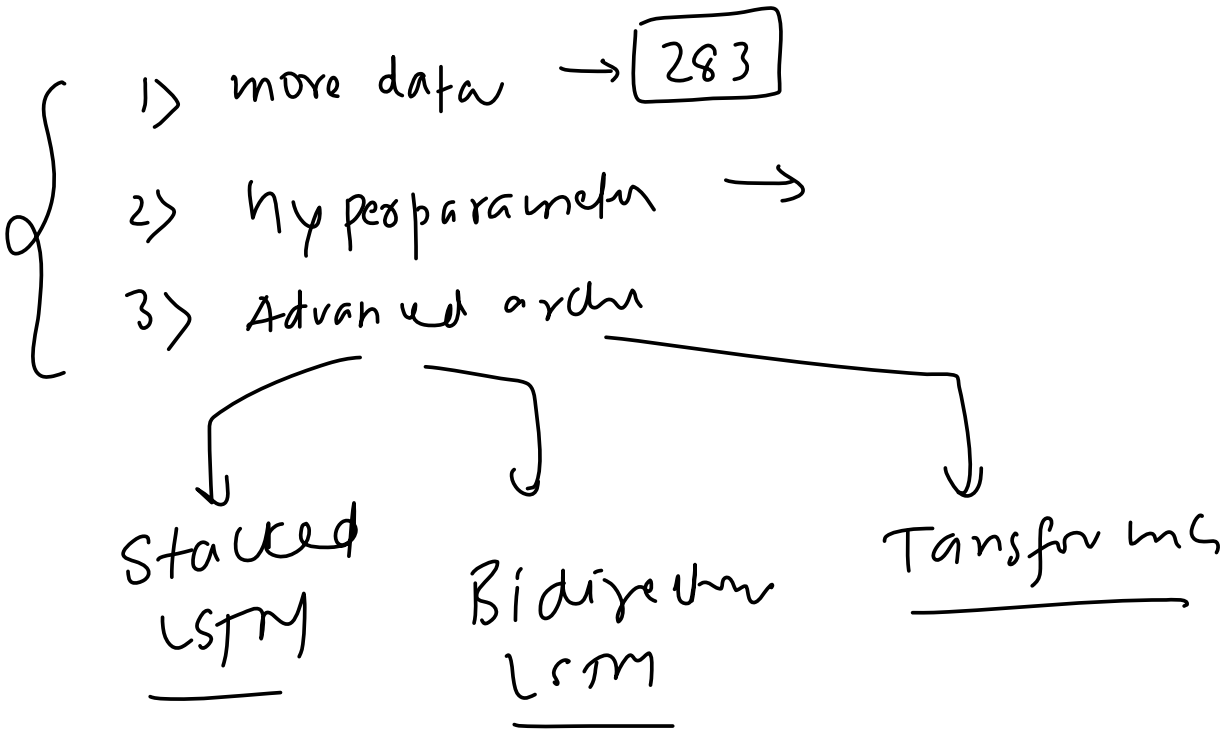
The Architecture

08 September 2023 08:55



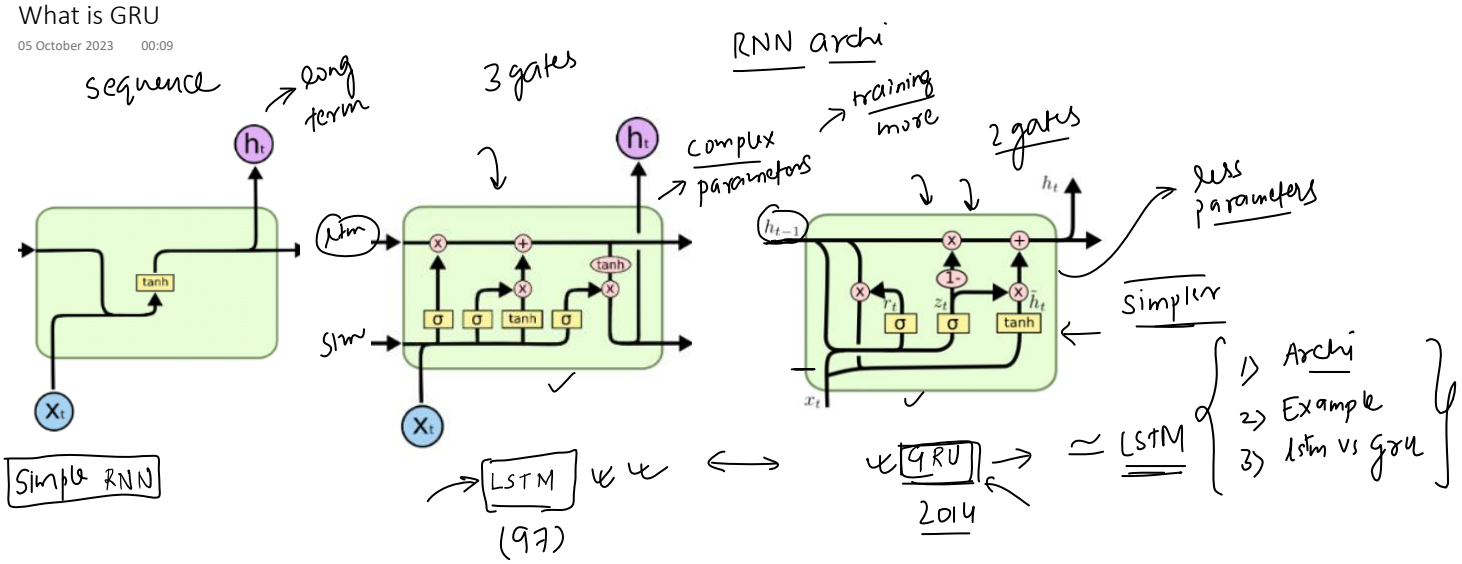
How to improve performance?

08 September 2023 08:51



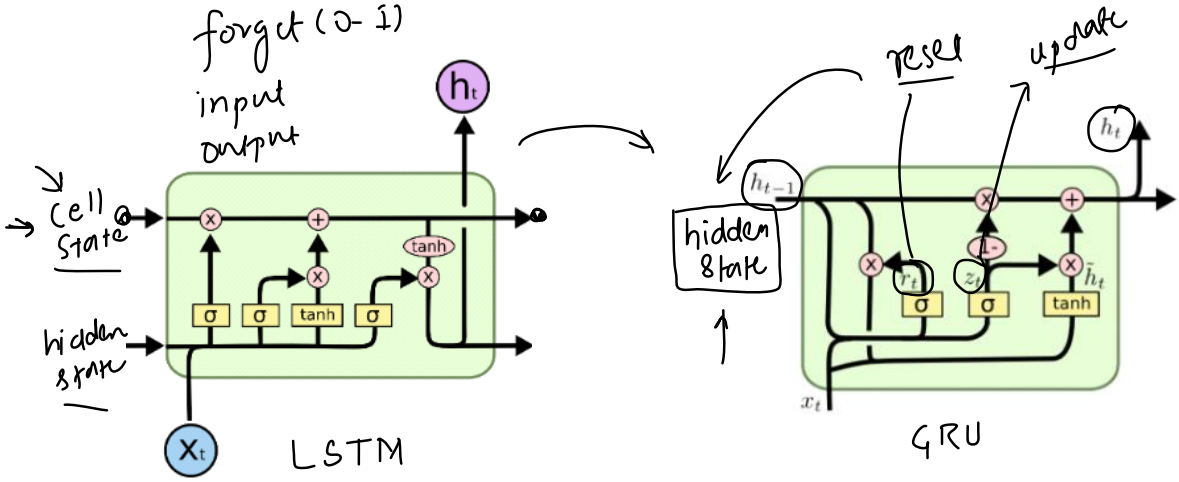
What is GRU

05 October 2023 00:09



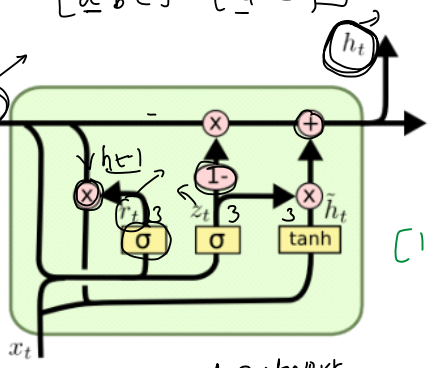
The Big Idea Behind GRU

05 October 2023 00:47



The Setup

05 October 2023 01:07



[a b c] [d e f]

[ad be cf]

[a+d b+e c+f]

neural network layers (FC layers)

→ Advise → LSTM/GRU → confusing

goal → t

prev hidden state | curr hidden state | input

h_{t-1}, x_t → h_t

vectors

h_{t-1}, h_t, x_t, r_t, z_t, $\tilde{h}_t$? → candidate hidden state

σ tanh ?

reset gate

update gate

[1 2 3 4] X 0 + 1- ?

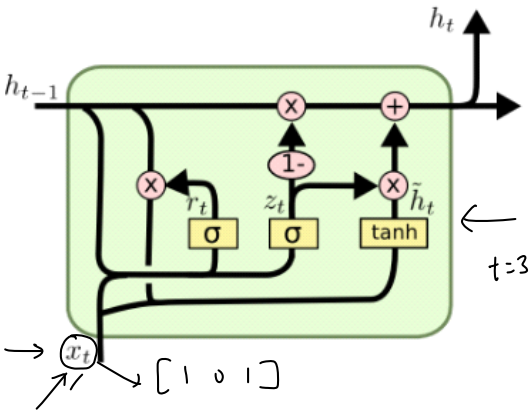
[1 2 3 4] update same

number of 10 = 5 6

σ → σ σ σ σ σ

The Input Xt

05 October 2023 01:52



t input → vector

Sentiment

OHE / BOW / Word2Vec

text	review
cat mat rat	1
cat rat rat	0
→ mat rat mat	1

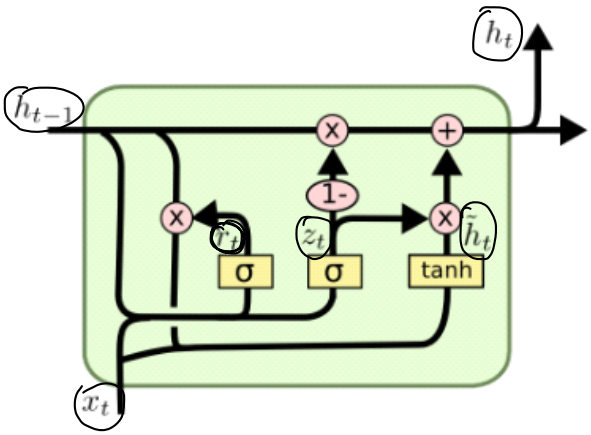
S1 [1 0 0] [0 1 0] [0 0 1]

S2 [1 0 0] [0 0 1] [0 0 1]

S3 [0 1 0] [0 0 1] [0 1 0]

Architecture

05 October 2023 02:10



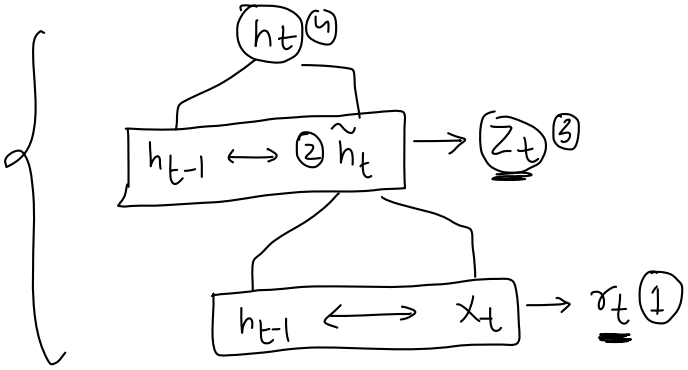
h_{t-1} , x_t ⟶ h_t

→ 1) Calculate r_t (reset gate)

→ 2) Calculate $\tilde{h}_t$ (candidate hidden state)

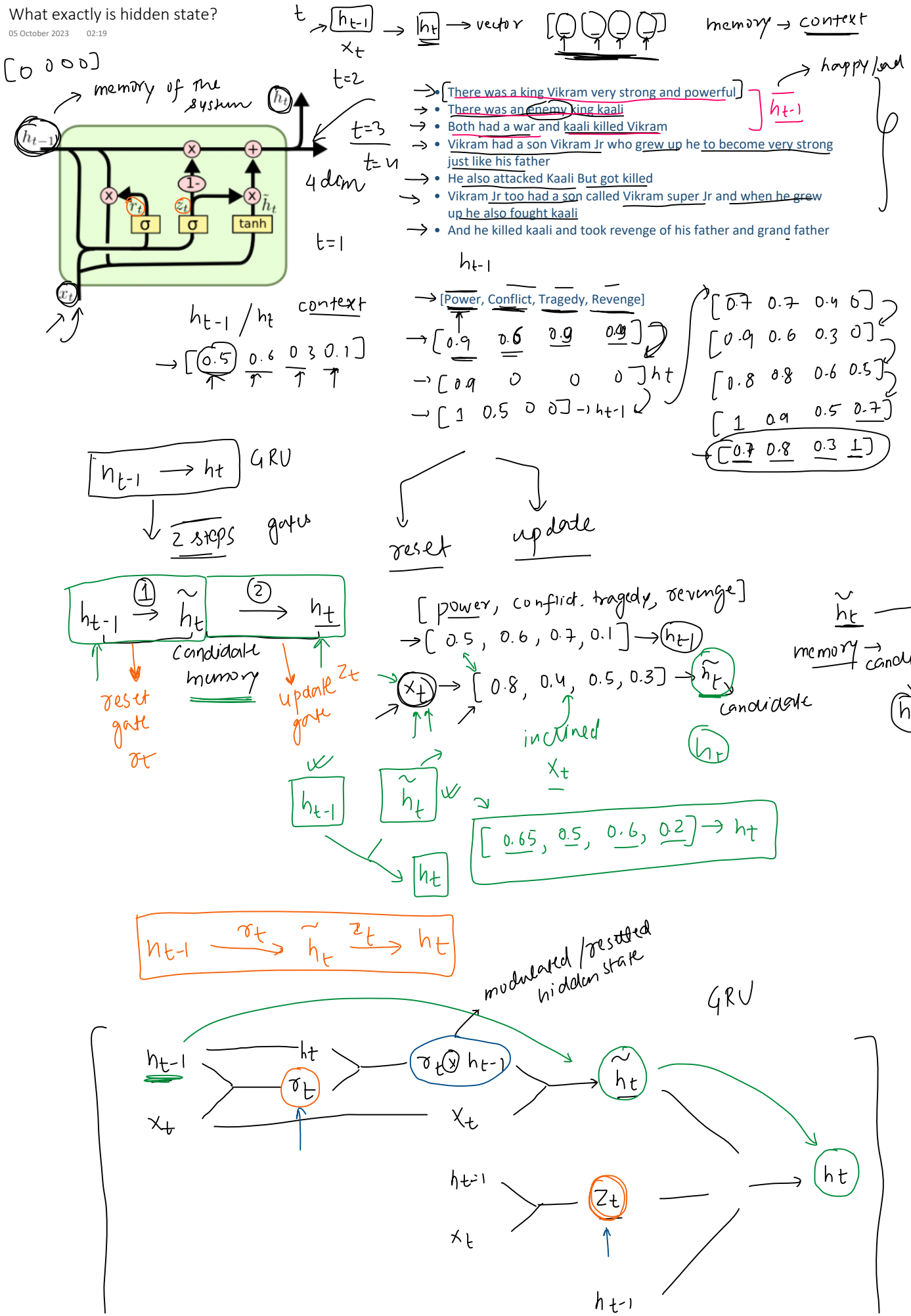
→ 3) Calculate z_t (update gate)

→ 4) Calculate h_t (current hidden state)



What exactly is hidden state?

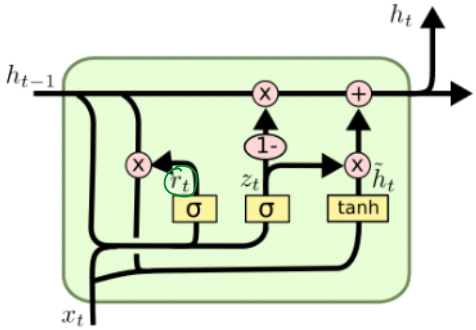
05 October 2023 02:19




h_{t-1}

Calculating the reset gate

05 October 2023 14:24



reset

r_t → vector → h_{t-1}

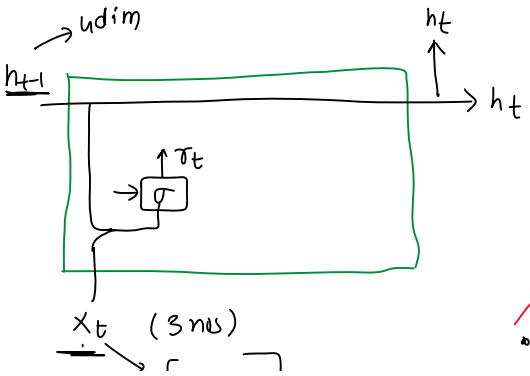
↳ gate

$h_{t-1} = [0.6, 0.6, 0.7, 0.1]$ lower lower higher

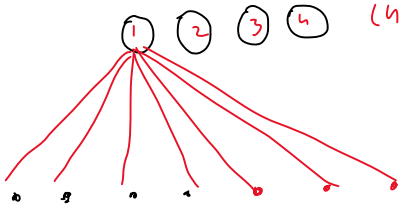
$r_t = [0.8, 0.2, 0.1, 0.9]$

gate — 80% percent — 20% — 90%

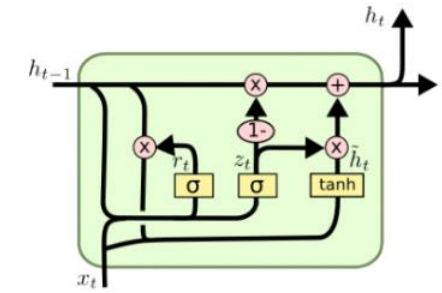
reset

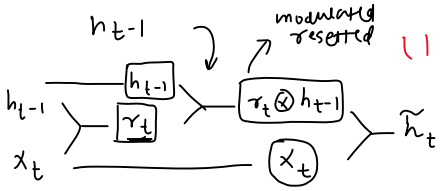


$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r)$



← (7x4) = 28 + 4





(1x7) (7x4)

(1x4) + (1x4) → ((1x4))

h_t

(1x4) → r_t

$\tilde{h}_t = [0.7\ 0.2\ 0.1\ 0.2]$

x_t

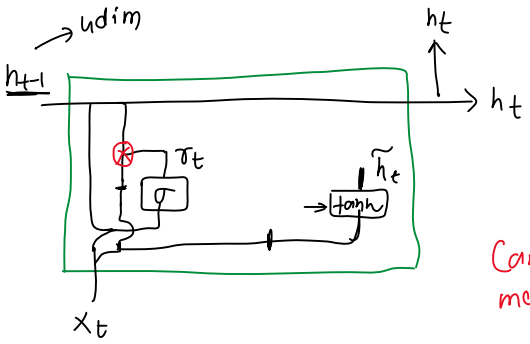
$h_{t-1} = [0.6, 0.6, 0.7, 0.1]$

$r_t = [0.8, 0.2, 0.1, 0.9]$ 90%

power conflict tragedy revenge

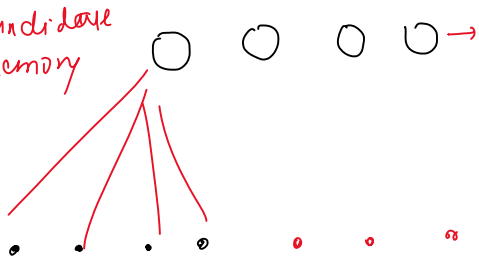
$h_{t-1} \otimes r_t = [0.48, 0.12, 0.07, 0.09]$ → x_t

h_{t-1} / $\tilde{h}_t$ → z_t



$\tilde{h}_t = \tanh(W_c[h_{t-1} \otimes r_t, x_t] + b_c)$

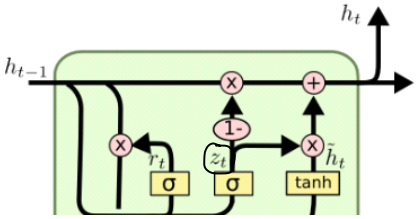
Candidate memory

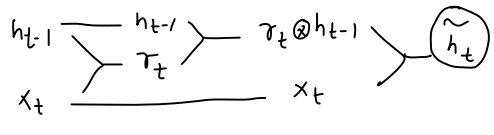


→ b_c

→ W_c (7x4)

$(h_{t-1} \otimes r_t)$

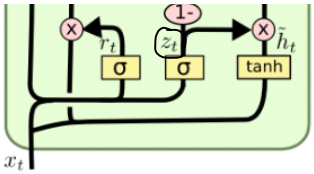


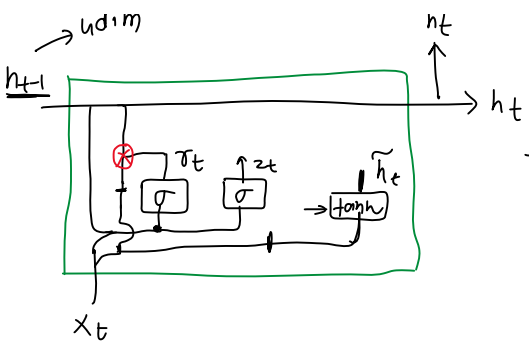


$\tilde{h}_t = [0.7\ 0.2\ 0.1\ 0.2]$

$h_{t-1} = [0.6\ 0.6\ 0.7\ 0.1]$

h_{t-1} z_t





h_{t-1} x_t → z_t

$[[- - - -]]$ (0-1)

h_{t-1} and $\tilde{h}_t$ (h_t)

gate

$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z)$

x_t → important

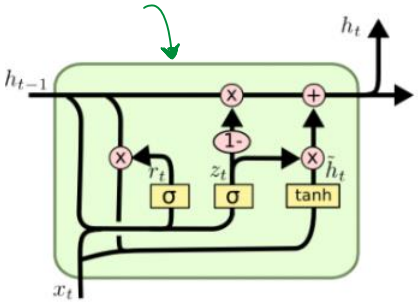
x_t → not imp

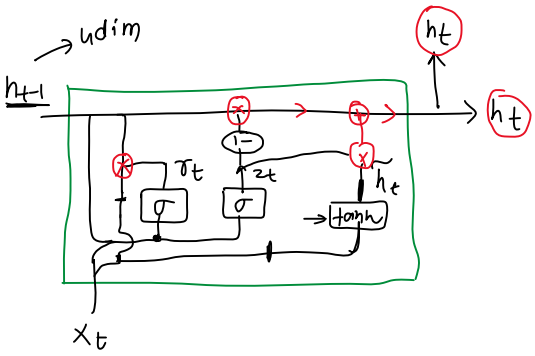
(1×4) h_{t-1} $\tilde{h}_t$ h_t

○ ○ ○ ○ → bias(4)

→ (7×4)

h_{t-1}





$\tilde{h}_t$ = [0.7 0.2 0.1 0.2]

h_{t-1} = [0.6 0.6 0.7 0.1]

z_t = [0.1 0.7 0.8 0.2]

h_t [0.07 0.14 0.08 0.04]

0.1 0.1

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

$(1-z_t)$ → 0.9

$(1-z_t)$ [0.9 0.3 0.2 0.8] [0.1 0.6 0.7 0.1]

[0.54 0.18 0.14 0.08]

h_t [0.61 0.32 0.22 0.12]

power conflict tragedy revenge

h_{t-1} x_t → h_{t-1} r_t → $r_t \otimes h_{t-1}$ x_t → $\tilde{h}_t$

h_{t-1} x_t → z_t

h_{t-1} z_t $\tilde{h}_t$ → h_t

LSTM vs GRU

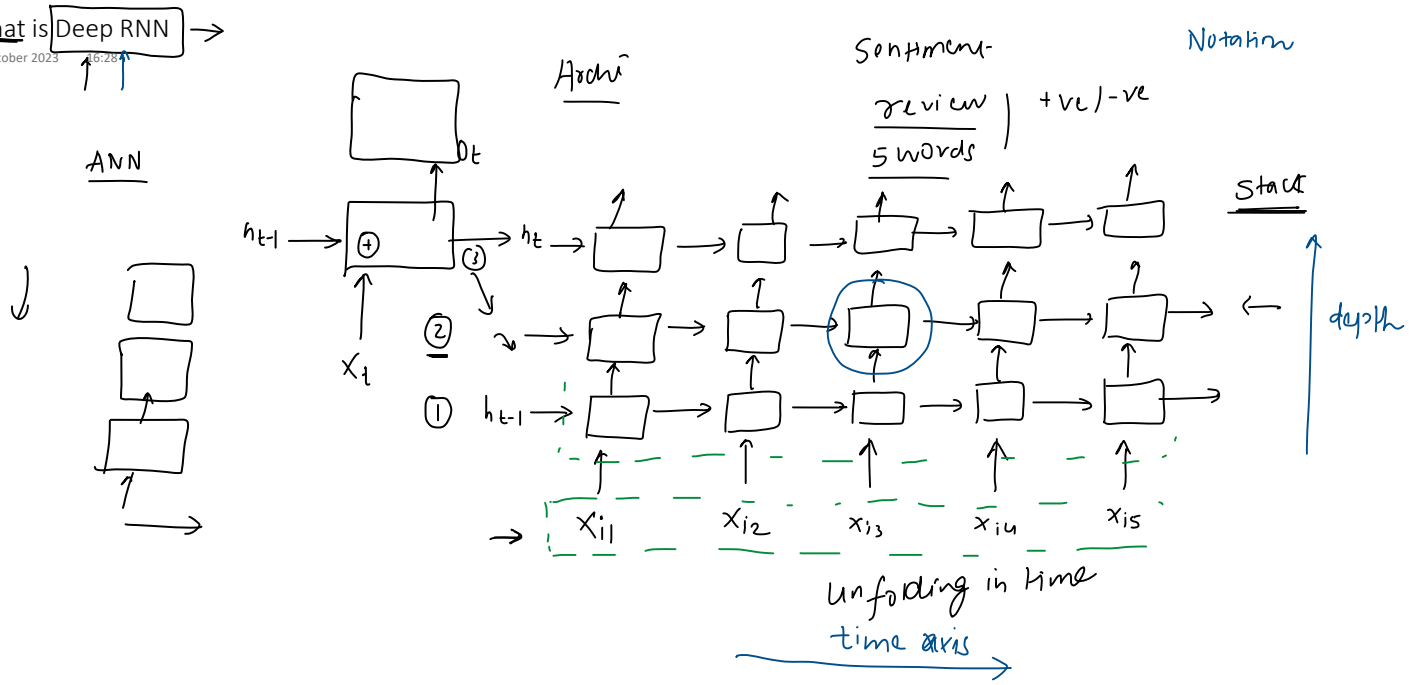
05 October 2023 16:45

Here are the main differences between LSTM and GRU:

1. Number of Gates:
   - LSTM: Has three gates — input (or update) gate, forget gate, and output gate.
   - GRU: Has two gates — reset gate and update gate.
2. Memory Units:
   - LSTM: Uses two separate states - the cell state (ct) and the hidden state (ht). The cell state acts as an "internal memory" and is crucial for carrying long-term dependencies.
   - GRU: Simplifies this by using a single hidden state (ht) to both capture and output the memory.
3. Parameter Count:
   - LSTM: Generally has more parameters than a GRU because of its additional gate and separate cell state. For an input size of d and a hidden size of h, the LSTM has 4 ×((d×h)+(h×h)+h)4×((d×h)+(h×h)+h)parameters.
   - GRU: Has fewer parameters. For the same sizes, the GRU has 3×((d×h)+(h×h)+h)3 ×((d×h)+(h×h)+h)parameters.
4. Computational Complexity:
   - LSTM: Due to the extra gate and cell state, LSTMs are typically more computationally intensive than GRUs.
   - GRU: Is simpler and can be faster to compute, especially on smaller datasets or when computational resources are limited.
5. Empirical Performance:
   - LSTM: In many tasks, especially more complex ones, LSTMs have been observed to perform slightly better than GRUs.
   - GRU: Can perform comparably to LSTMs on certain tasks, especially when data is limited or tasks are simpler. They can also train faster due to fewer parameters.
6. Choice in Practice:
   - The choice between LSTM and GRU often comes down to empirical testing. Depending on the dataset and task, one might outperform the other. However, GRUs, due to their simplicity, are often the first choice when starting out.

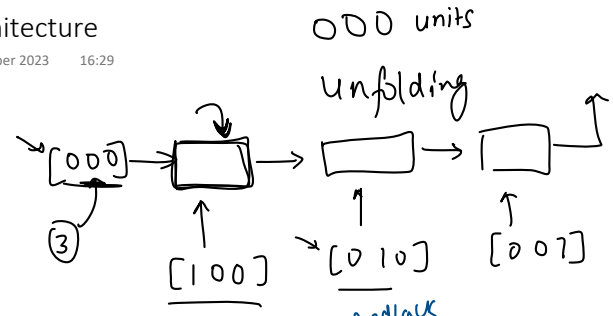
What is Deep RNN →

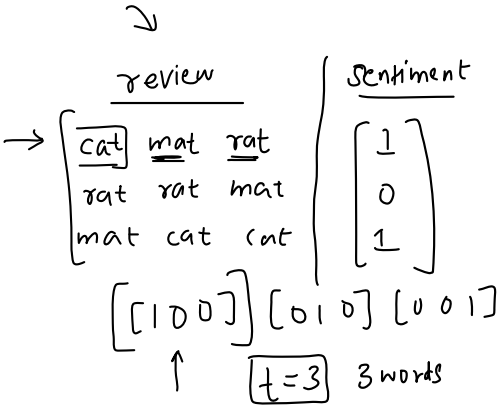
17 October 2023 16:28

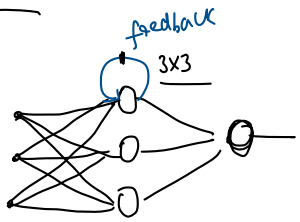


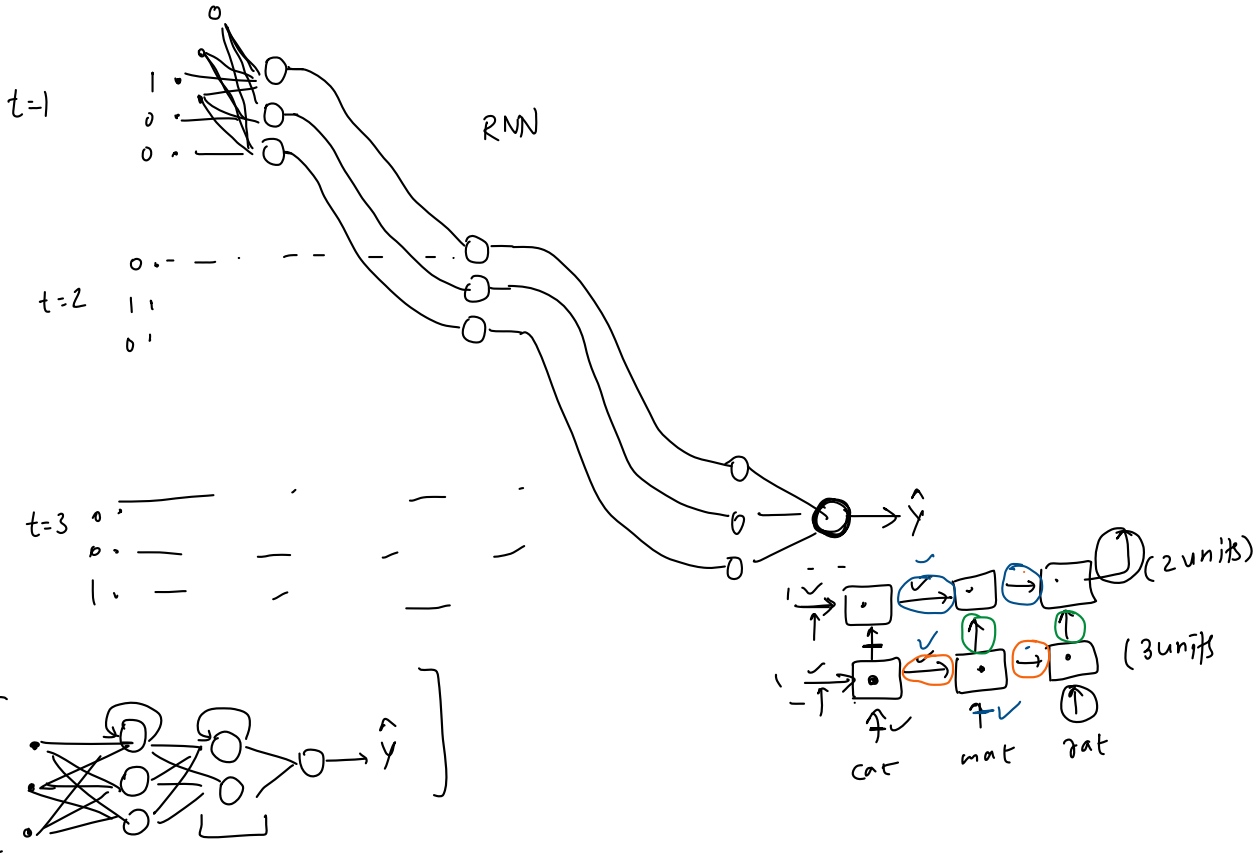
Architecture

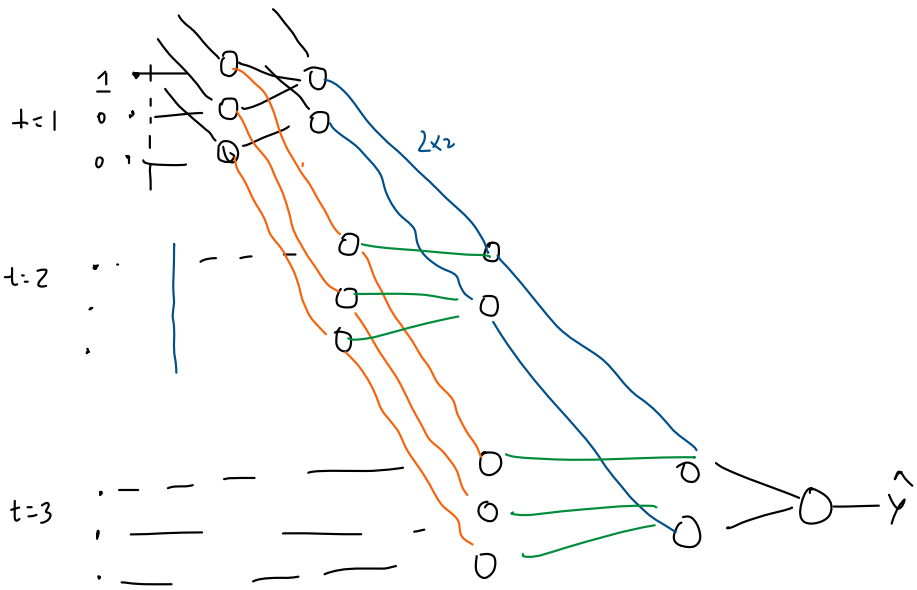
17 October 2023 16:29

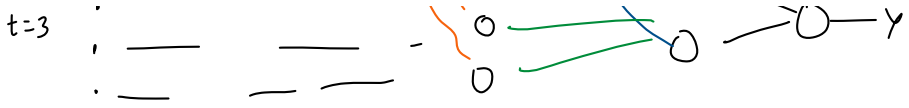






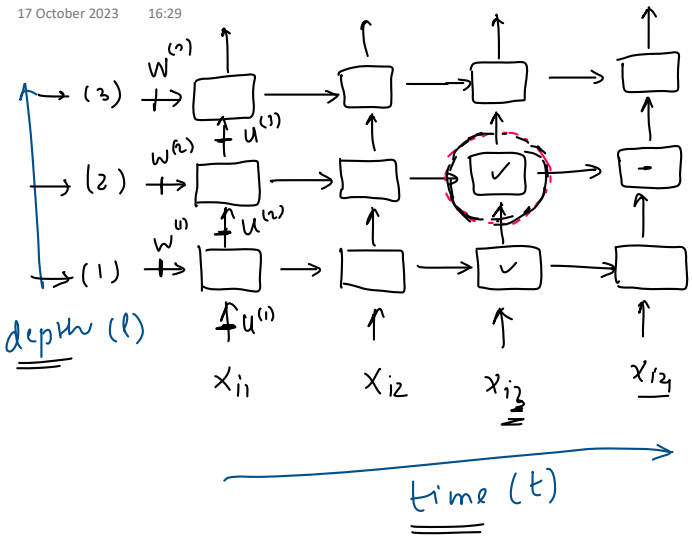


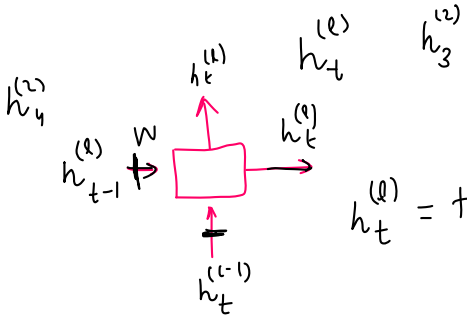



t=3
y

Notation

17 October 2023 16:29





$$h_t^{(l)} = \tanh\left(W^{(l)} h_{t-1}^{(l)} + U^{(l)} h_t^{(l-1)} + b^{(l)}\right)$$

Why and When to use?

17 October 2023 16:29

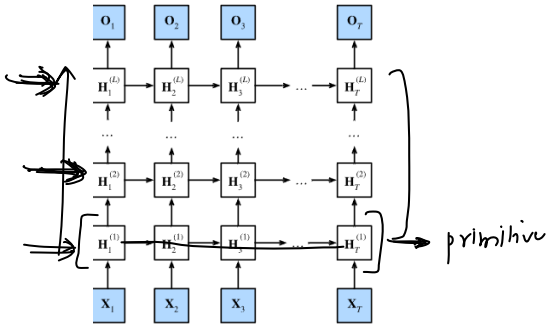
deep RNN

stack

1. Hierarchical Representation ✓
2. Customization for Advanced Tasks

product

sentence



encoder-decoder

machine

deep RNNs

love hate amazing terrible

deep RNN

Audio is bad

Display is great

Overall I am happy

→ sentiment

When to use Deep RNNs?

Complex tasks

Speech recog

Machine translation

Large datasets

Overfitting

Computational

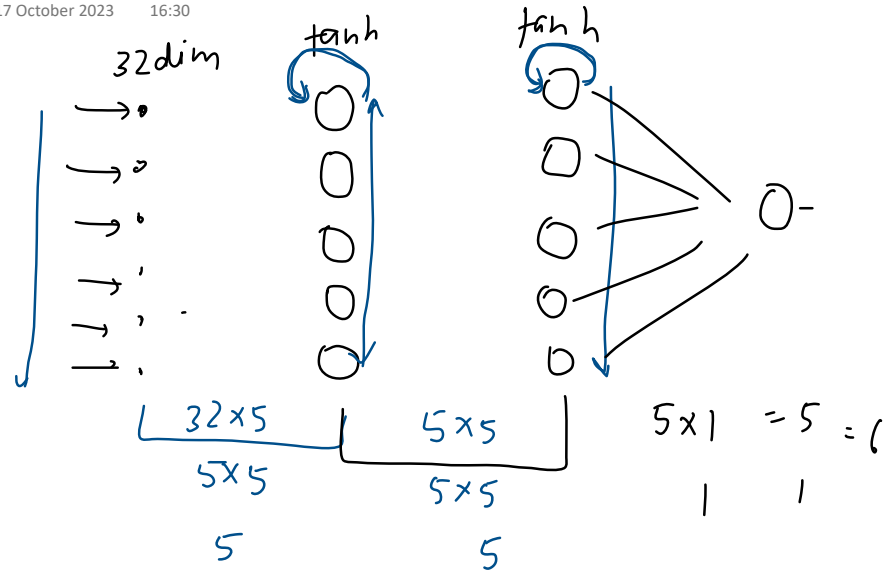
Simpler Models

↓

Deep RNN

Code Example

17 October 2023 16:30



→ embedded

10000 x 32 = 3,20,000

SimpleRNN 1

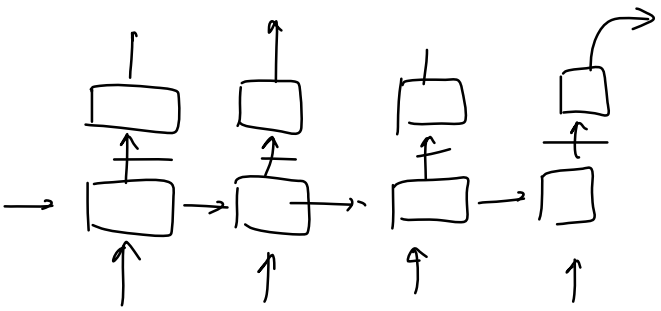
190

SimpleRNN 2

55

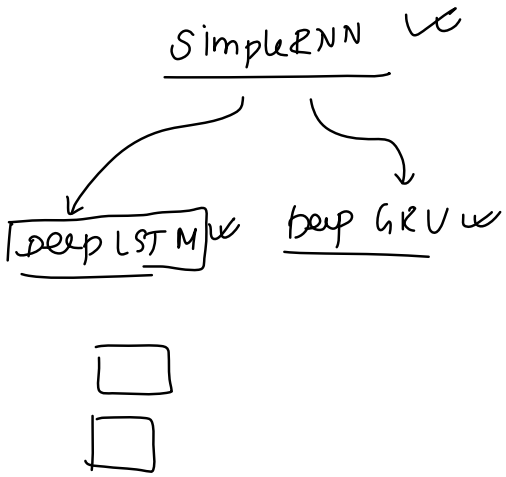
Dense

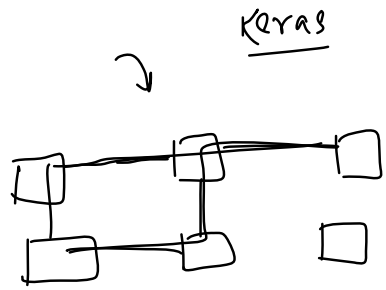
6



Variants

17 October 2023 16:30



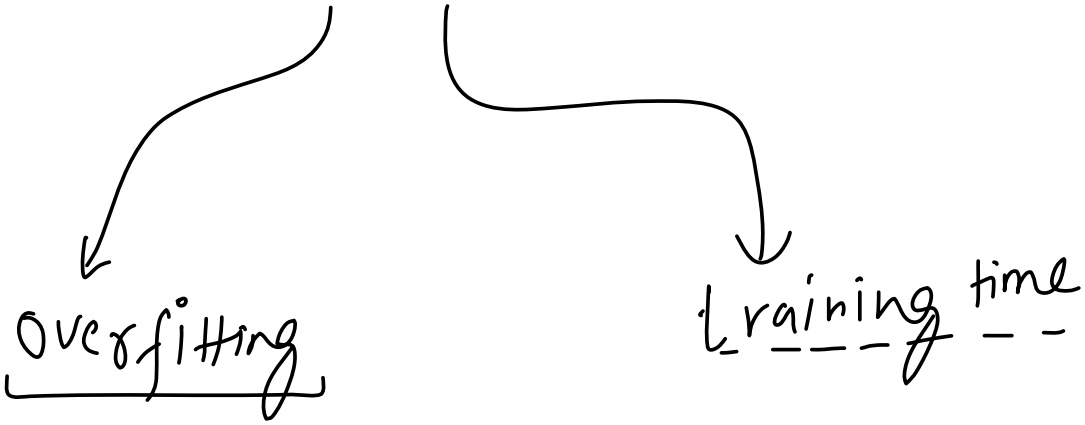


[Computation graph]

Pytorch

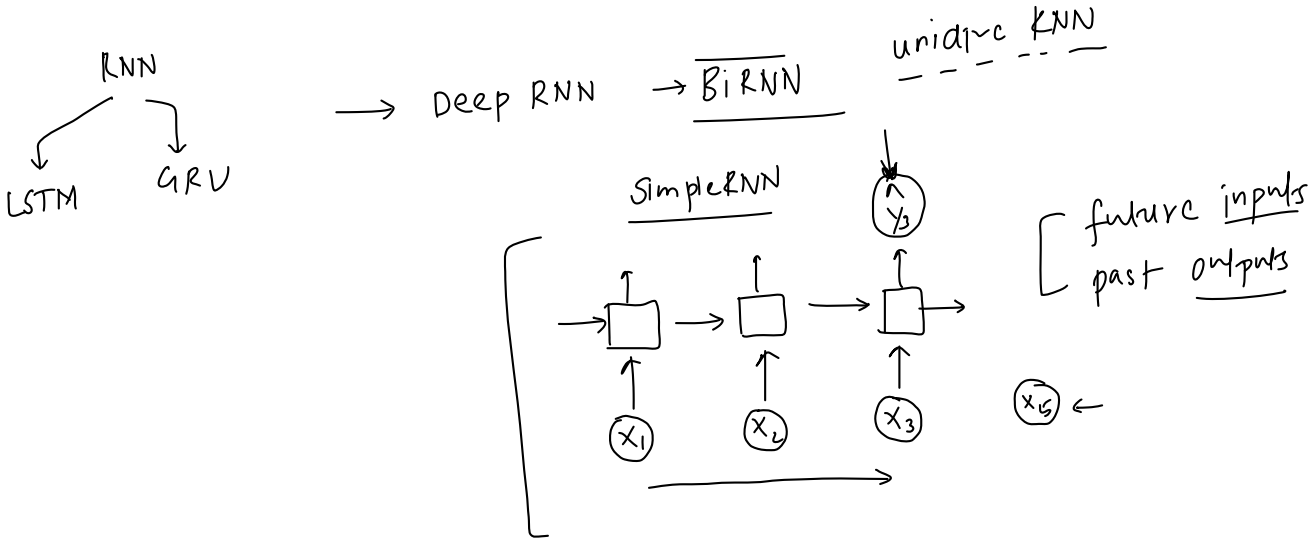
Disadvantages

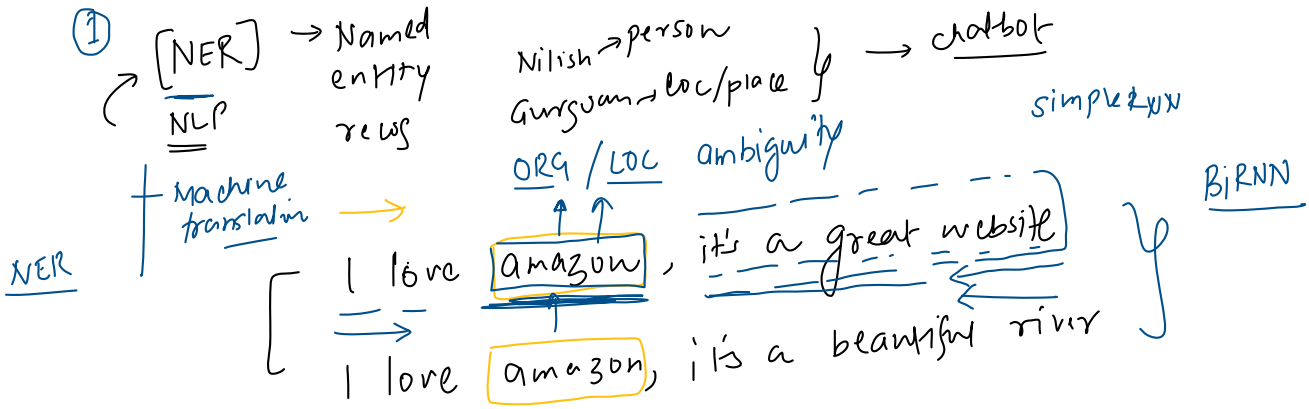
17 October 2023 16:30



The Why?
26 October 2023 15:19

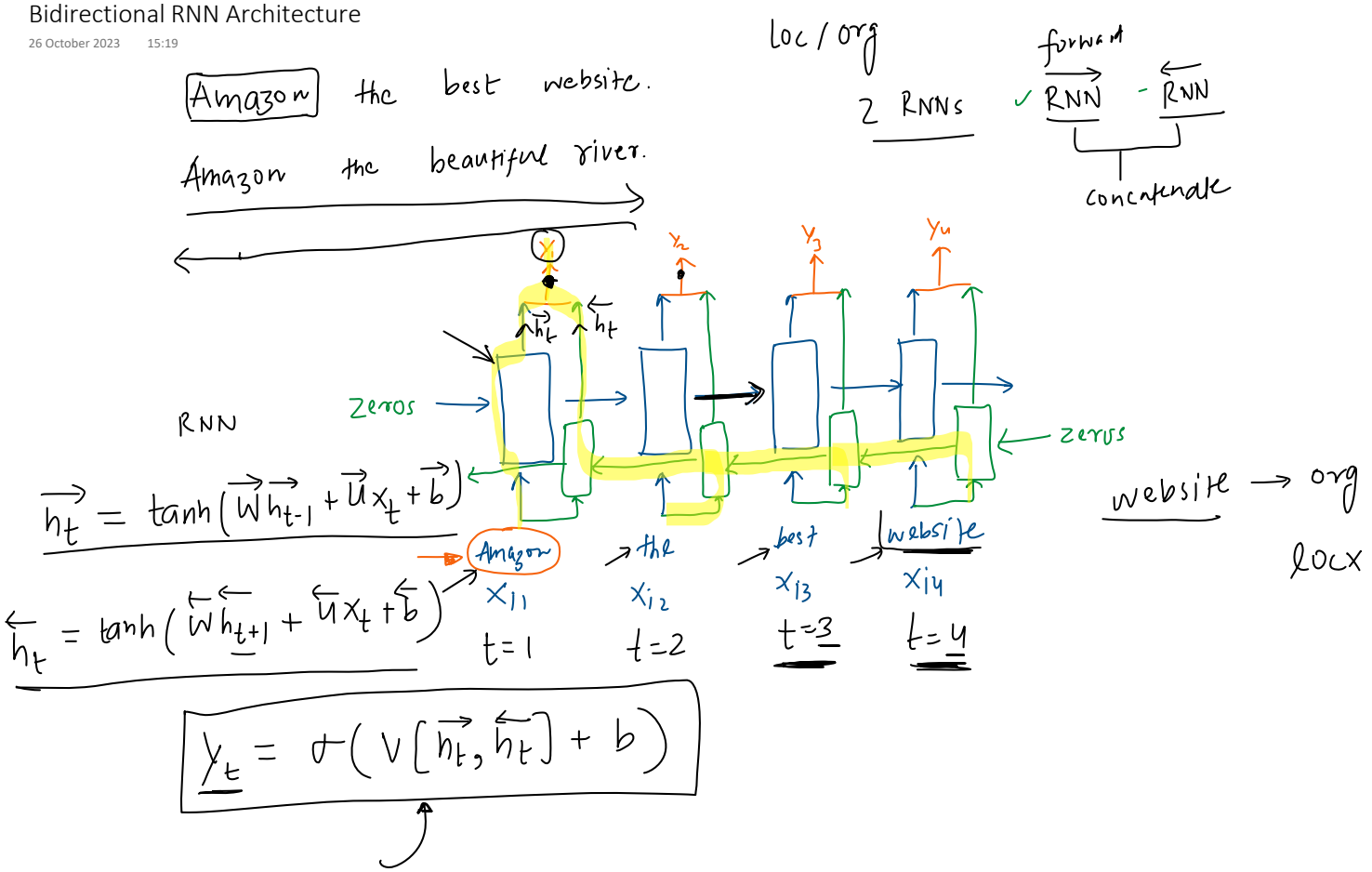
BiRNN





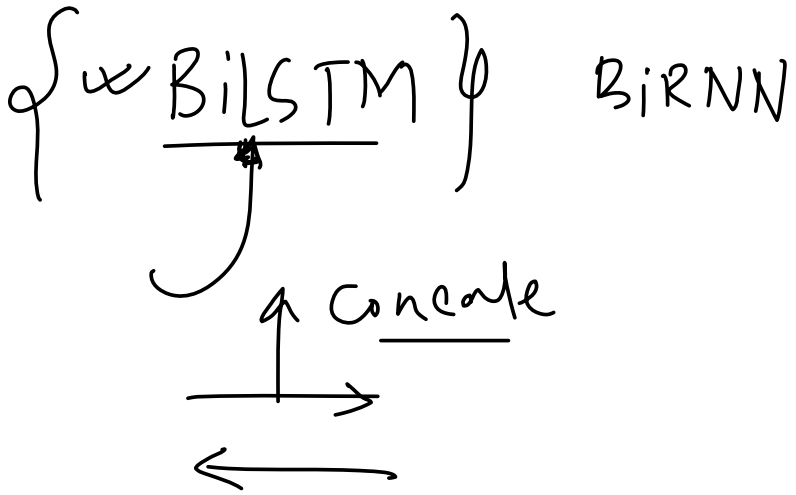
Bidirectional RNN Architecture

26 October 2023 15:19



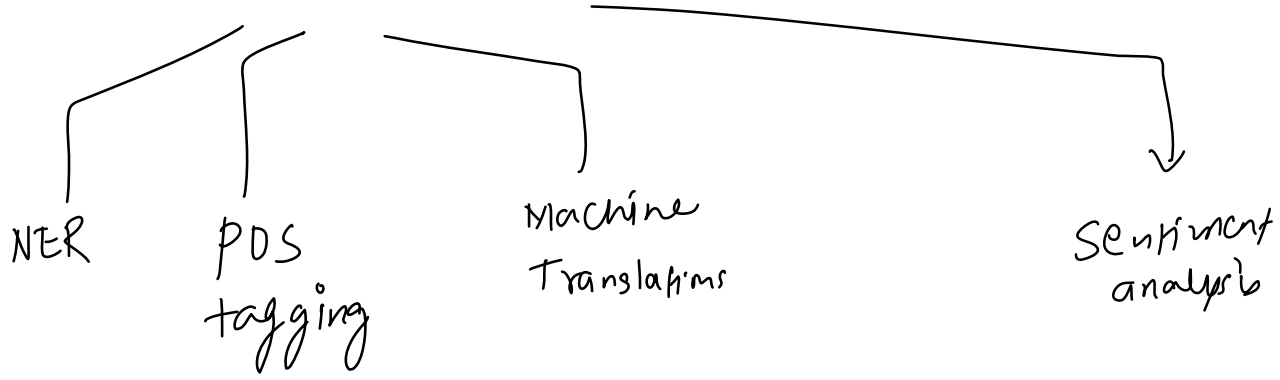
Code

26 October 2023 15:21



Applications and Drawbacks

26 October 2023 15:21



[Time series forecasting]

→ ←

→ Complexity → 190 → 380 → training → overfitting
double

→ → ← [Speech recog] → bi rnn

↳ latency → slow