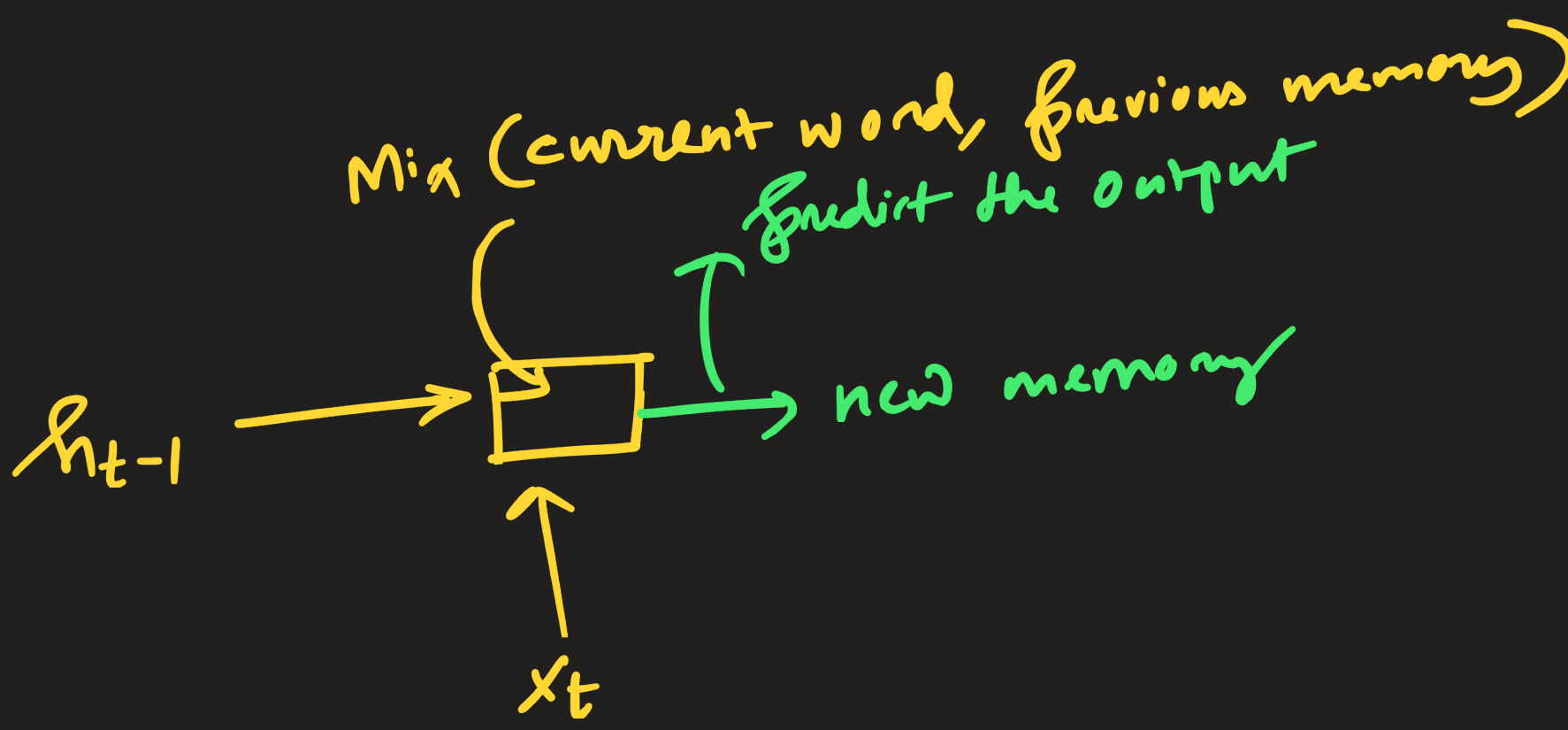
RNNs

Satya Pattnaik

RNN - thinks one word at a time

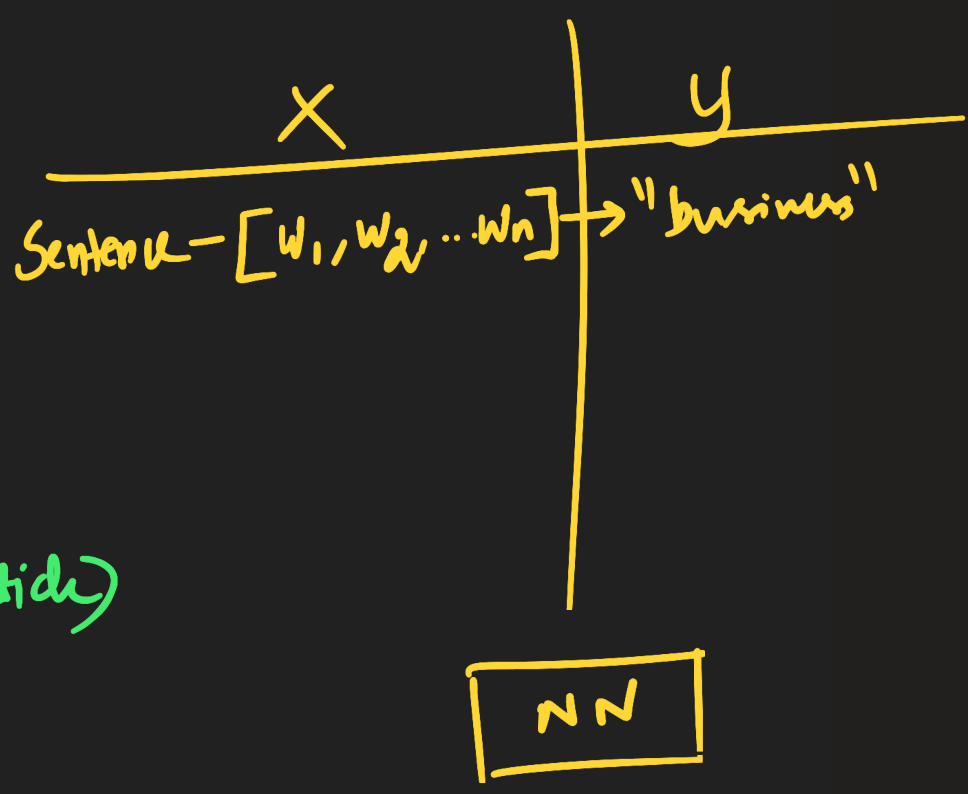


Model

$$y = f(x)$$

y: category

x: text (news article)



Why MLP is not meant for this problem?

S1: dog bites man

S2: man bites dog

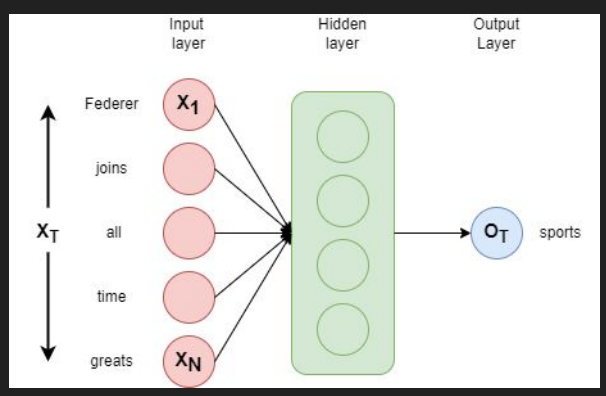
dog bites man

dog
bite
man

100 150 200

200 150 100

What if we use MLP?



All inputs at once

not → -ve

I love food

I love good food

← Sentence of 4 words

I am not a bad singer

This is not a bad movie +ve

100 200 300 +ve ↑

What is missing in this Architecture?

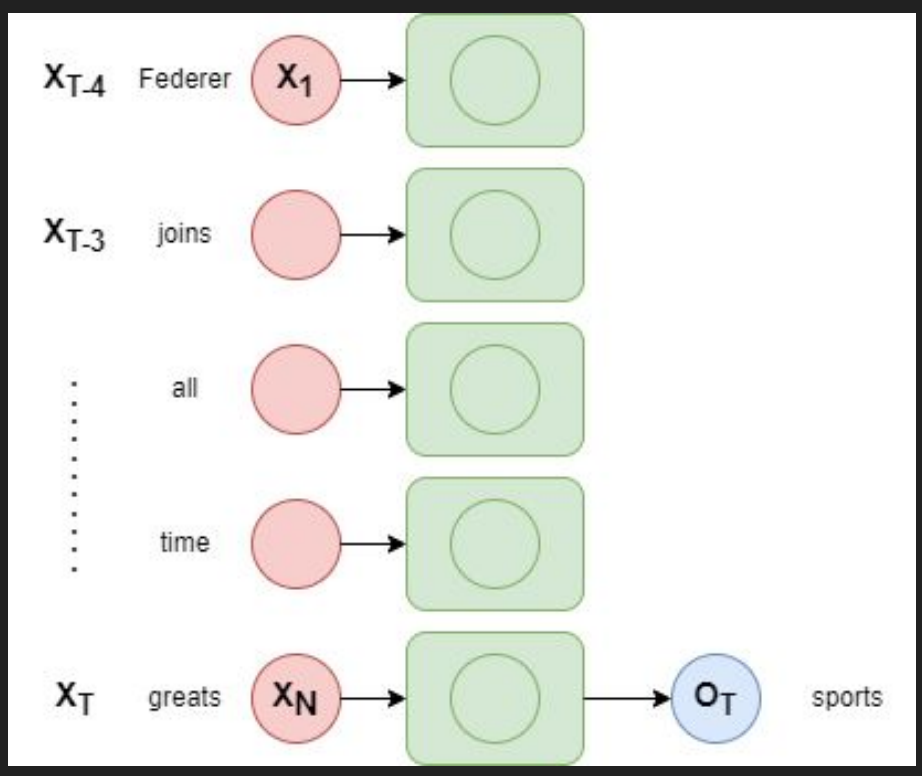
200
300
100

→ no connection with previous input words

→ each word is isolated from each other

→ varying lengths of sentences also will have issues

Can we modify this architecture?

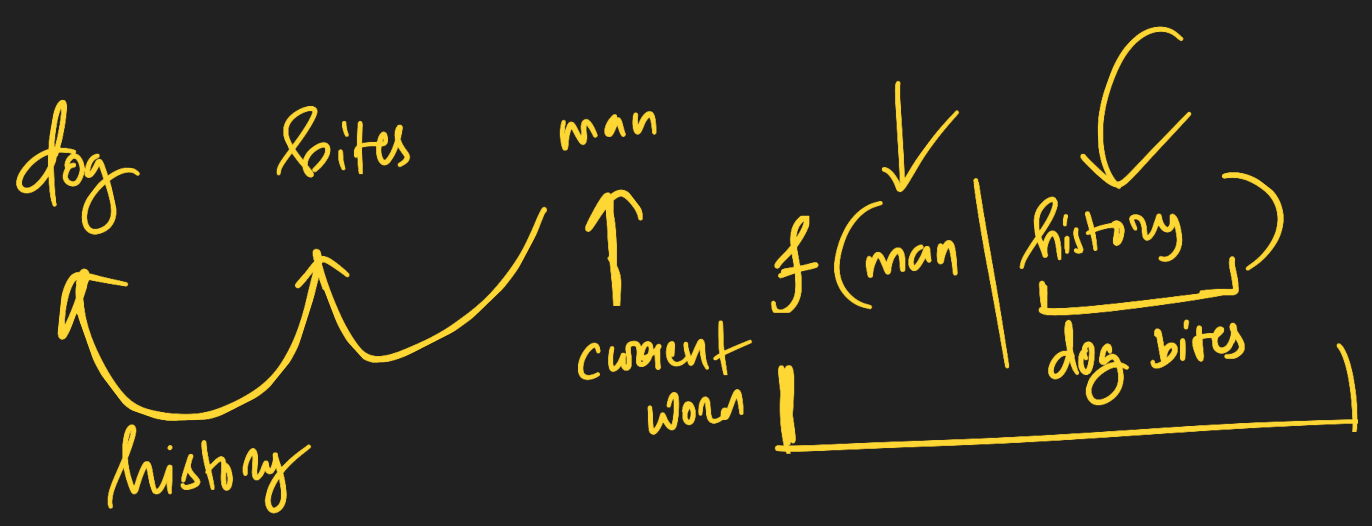


What is missing?

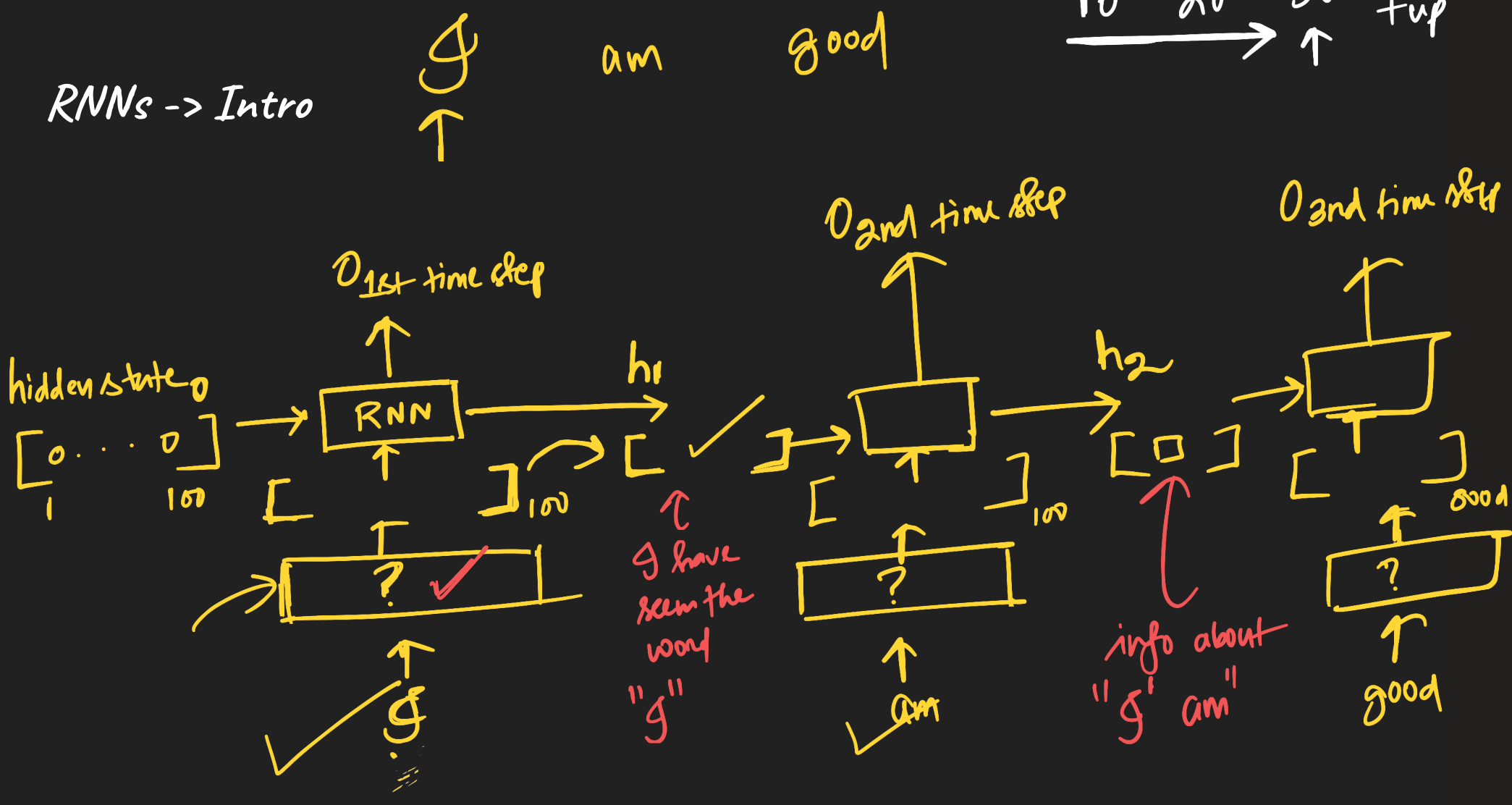
Summarize the issues

{ log bites man
{ man bites dog

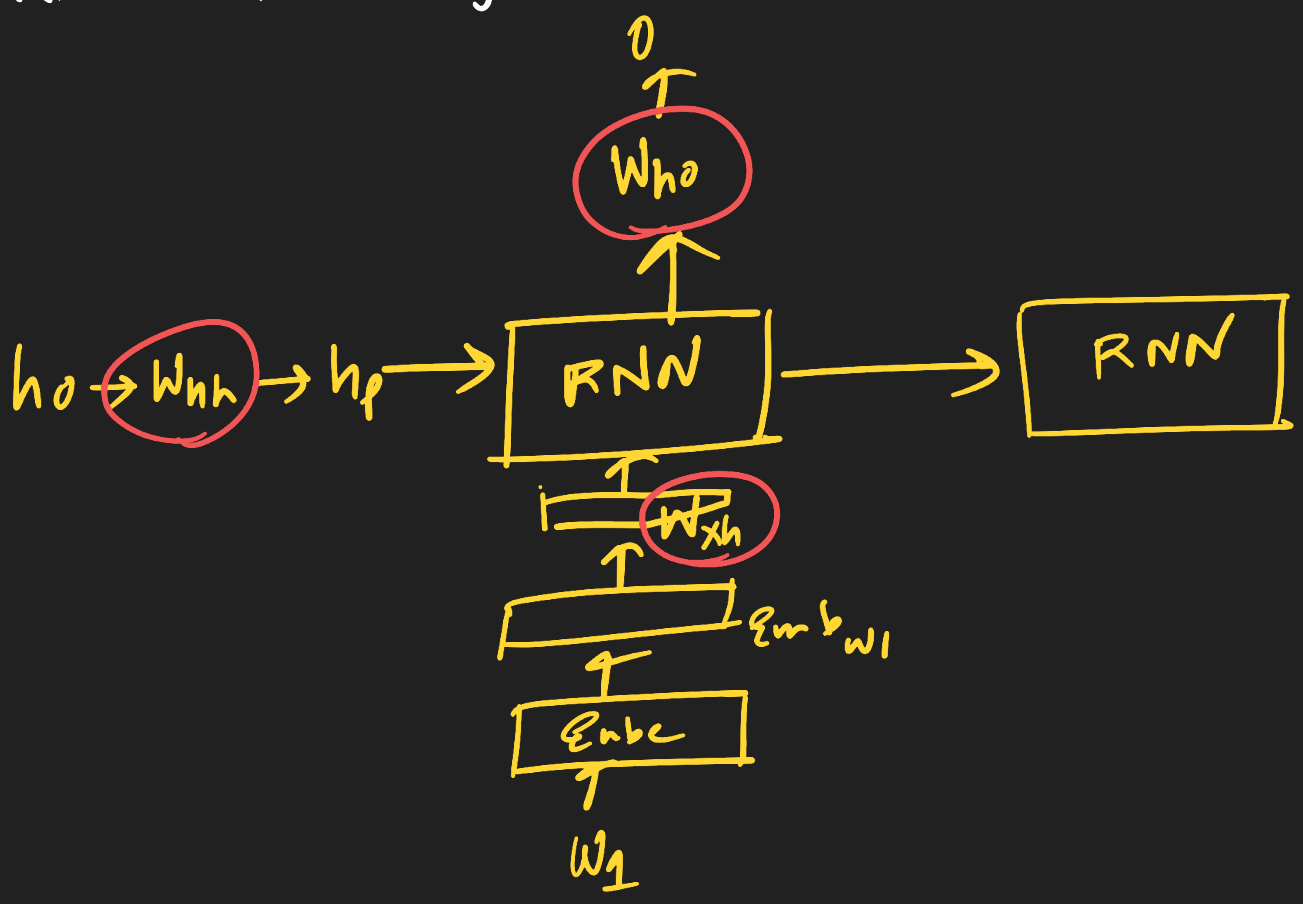
NN
$o_t = f(\text{words})$

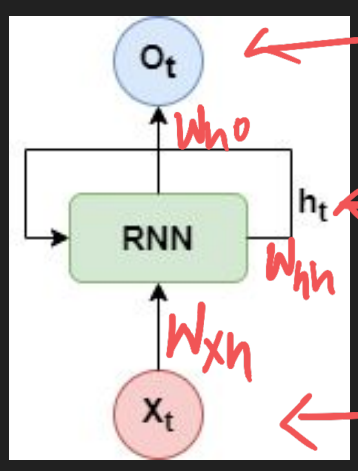


RNNs -> Intro


I
am
good
10
20
30
stock
up
0 1st time step
0 2nd time step
0 3rd time step
hidden state 0
[0 . . . 0]
1
100
RNN
h1
h2
100
?
I have seen the word "I"
I am
info about "I am"
good

RNNs -> Advantages





output

hidden state
(captures history)

input word

How is it unravelled over time?

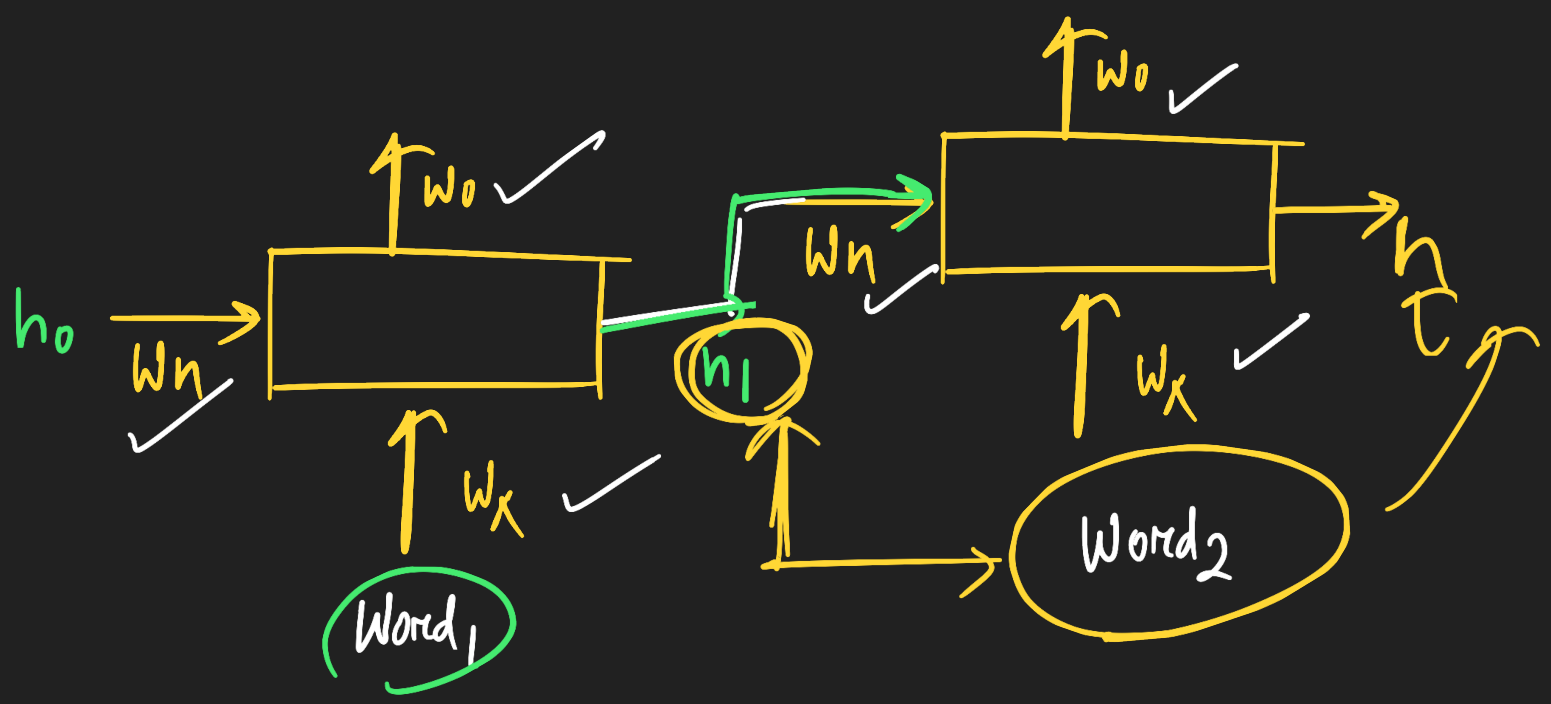
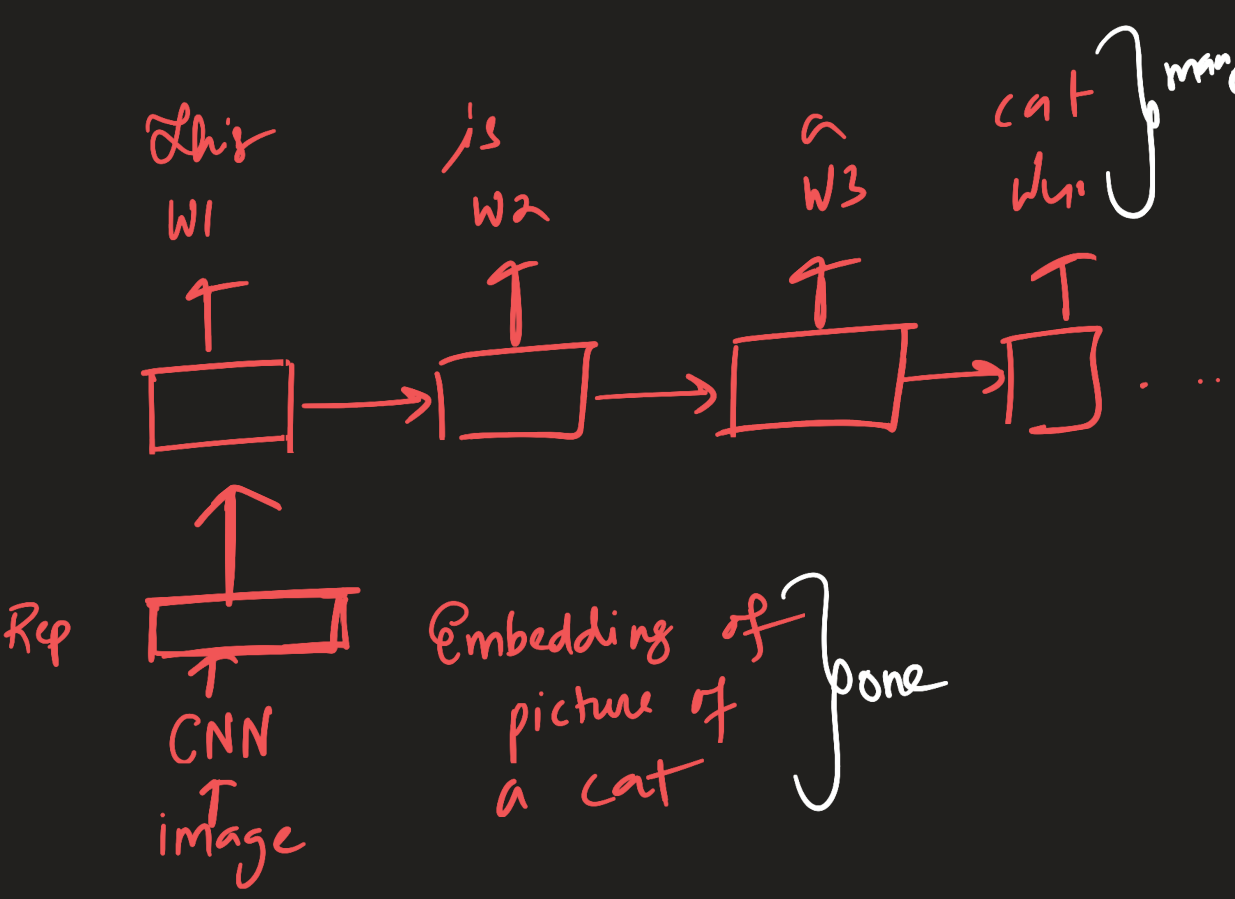
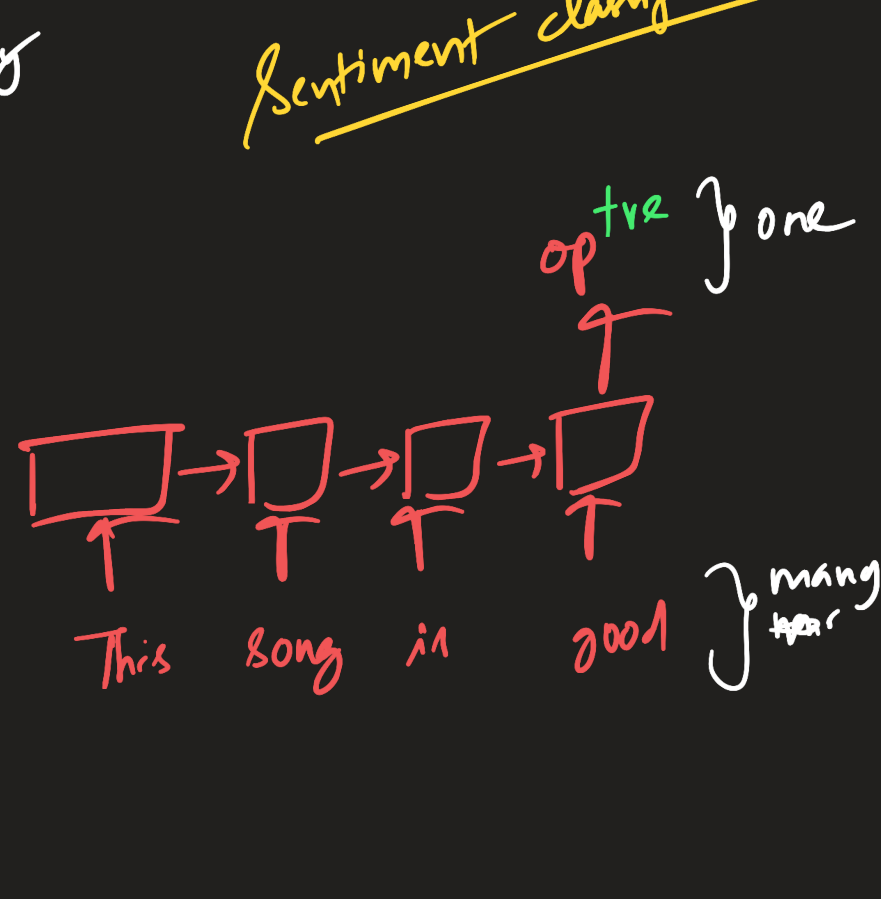
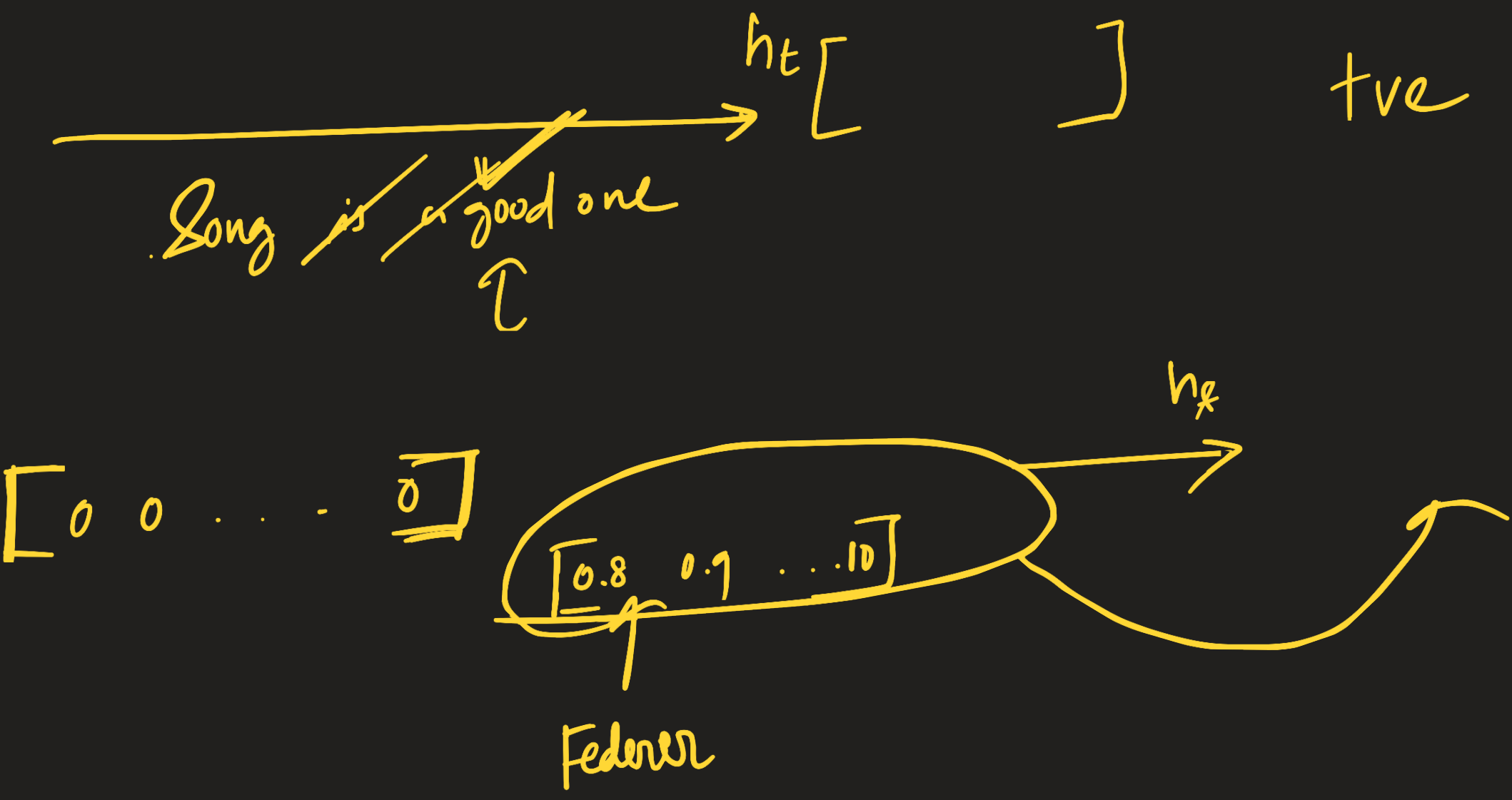


Image captioning



Sentiment classification



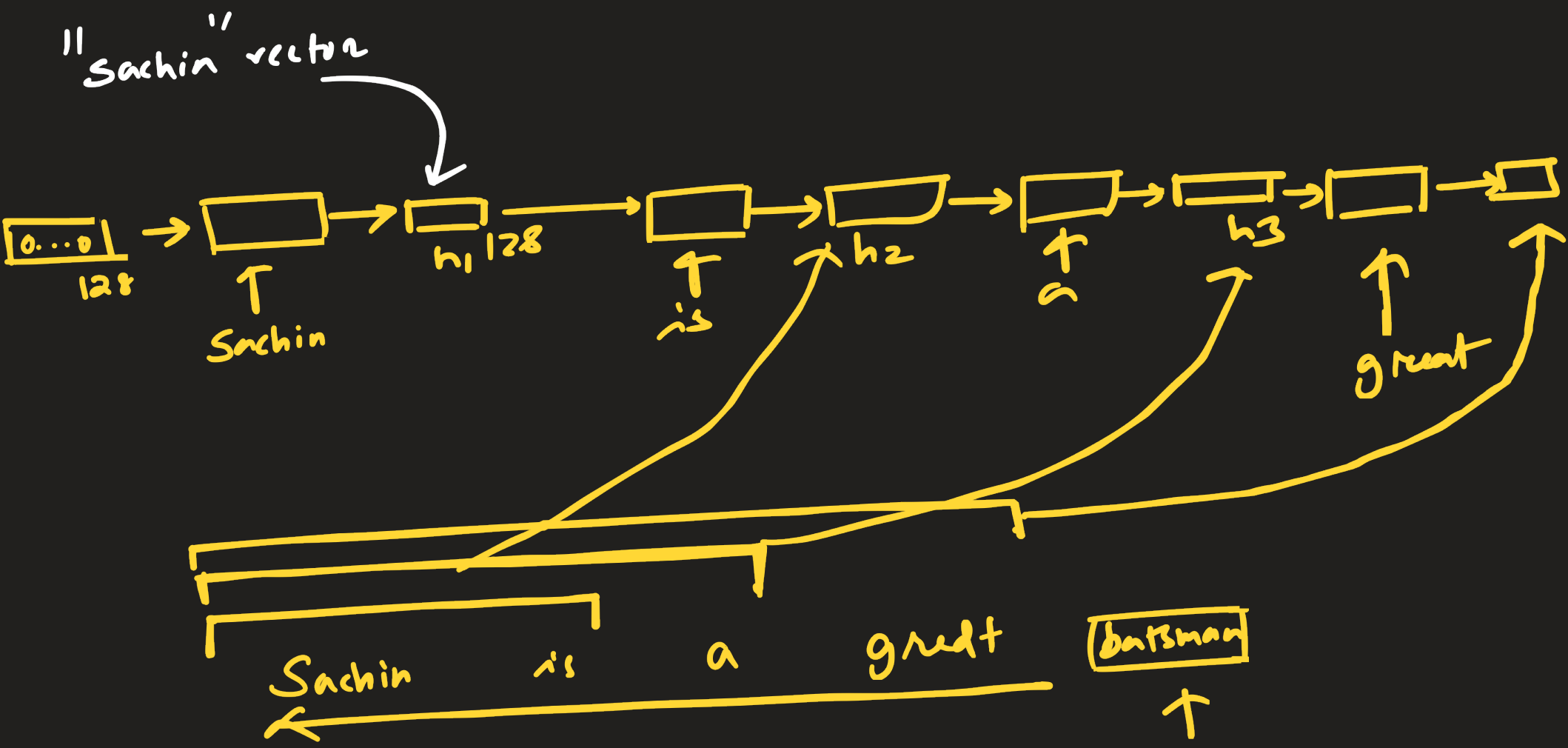


$$\longrightarrow h_2 = f(emb_1, emb_2)$$

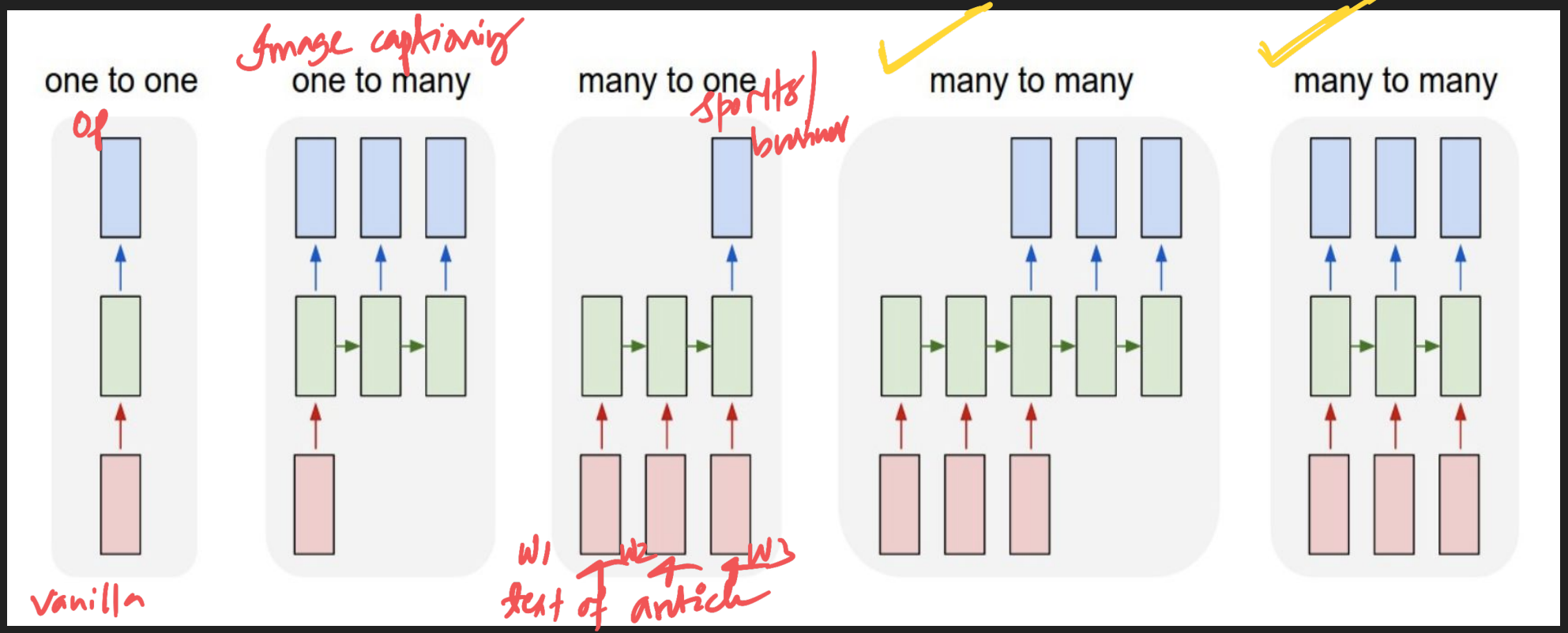
W_1 W_2

great food

great game


"Sachin" vector
0...0
128
Sachin
h1 128
is
h2
a
h3
great
Sachin
is
a
great
batsman

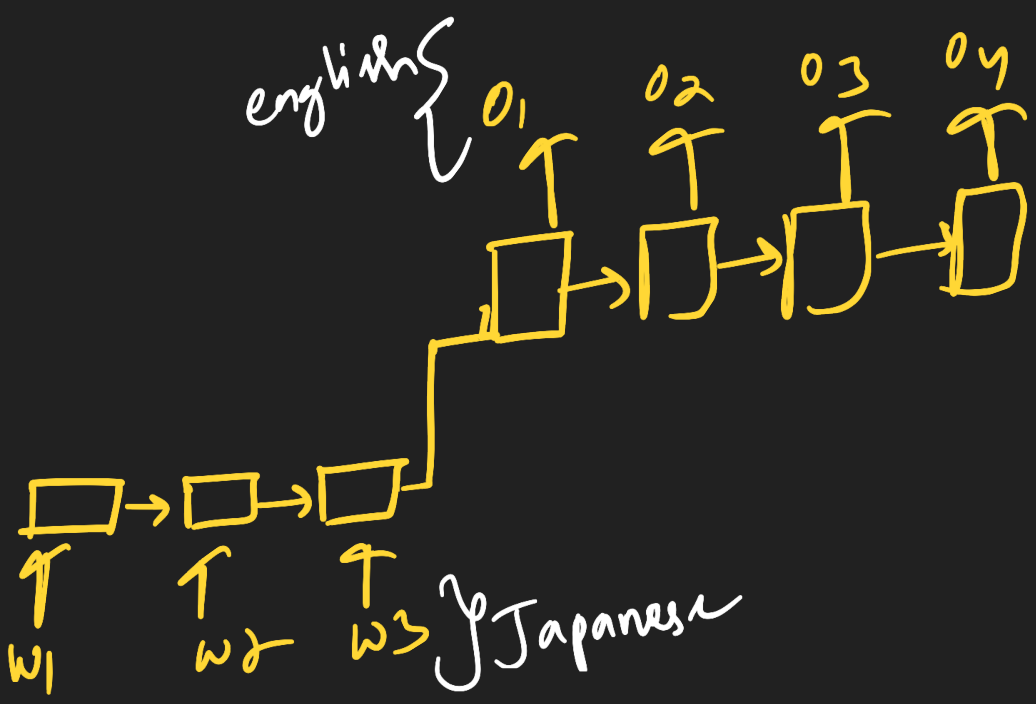
Types of RNN



Types of RNN

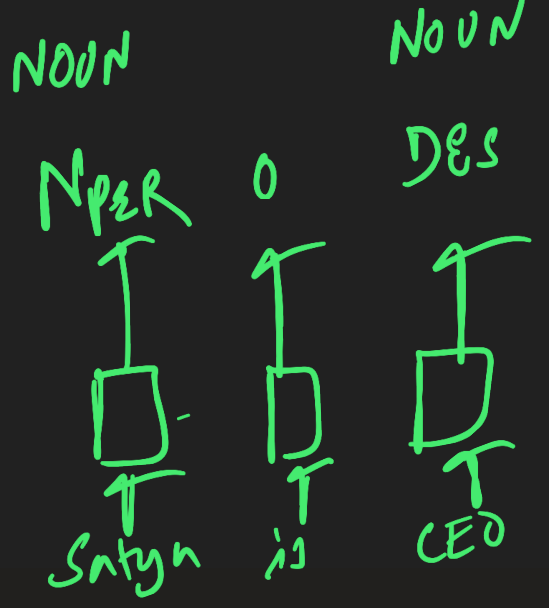
Many to Many

no. of ip ≠ no. of op



Many to Many

no. of ip = no of op

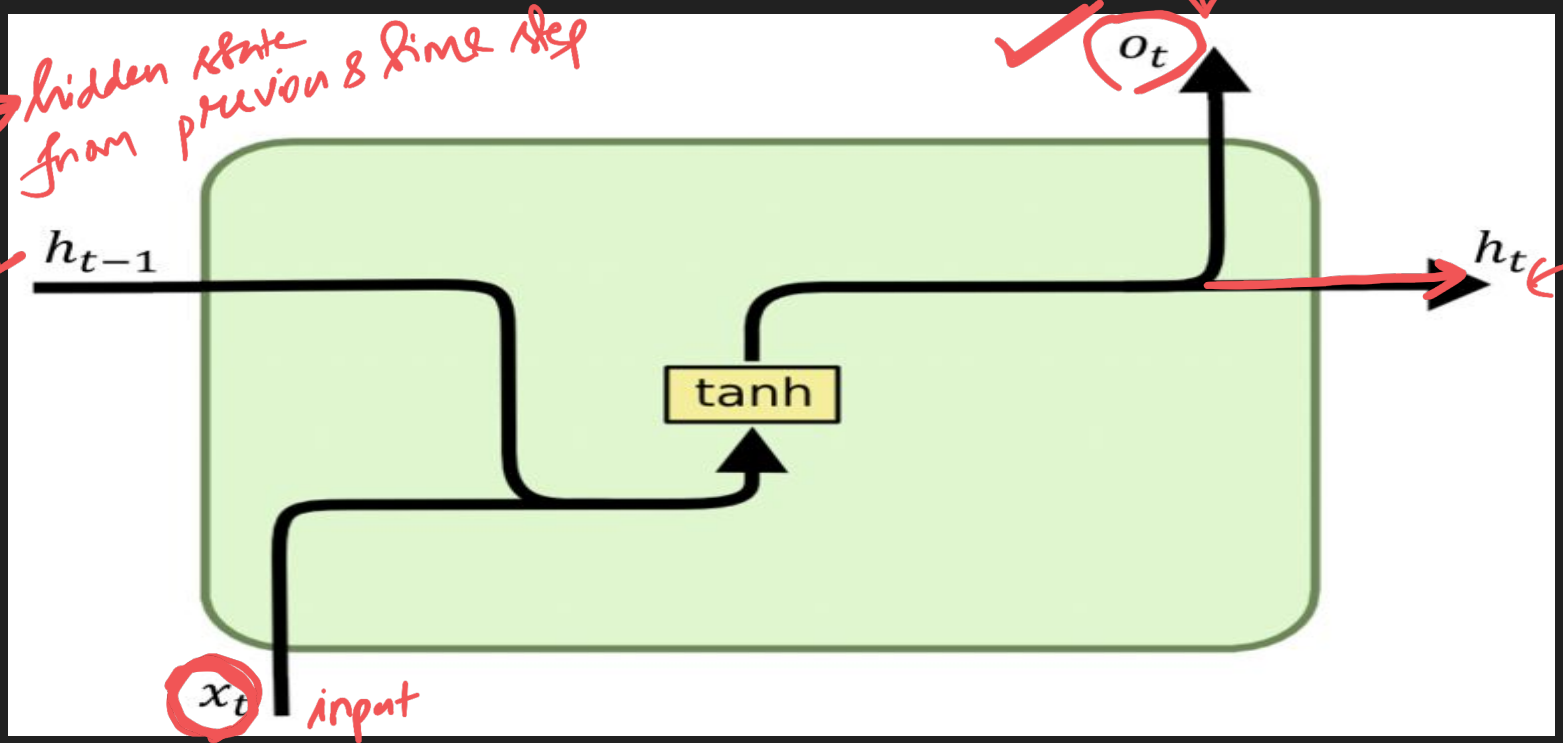


h_{t-1} 100 200 → x_t 300 } o_t, h_t

Different Components of RNN

output = $Z(h_t)$

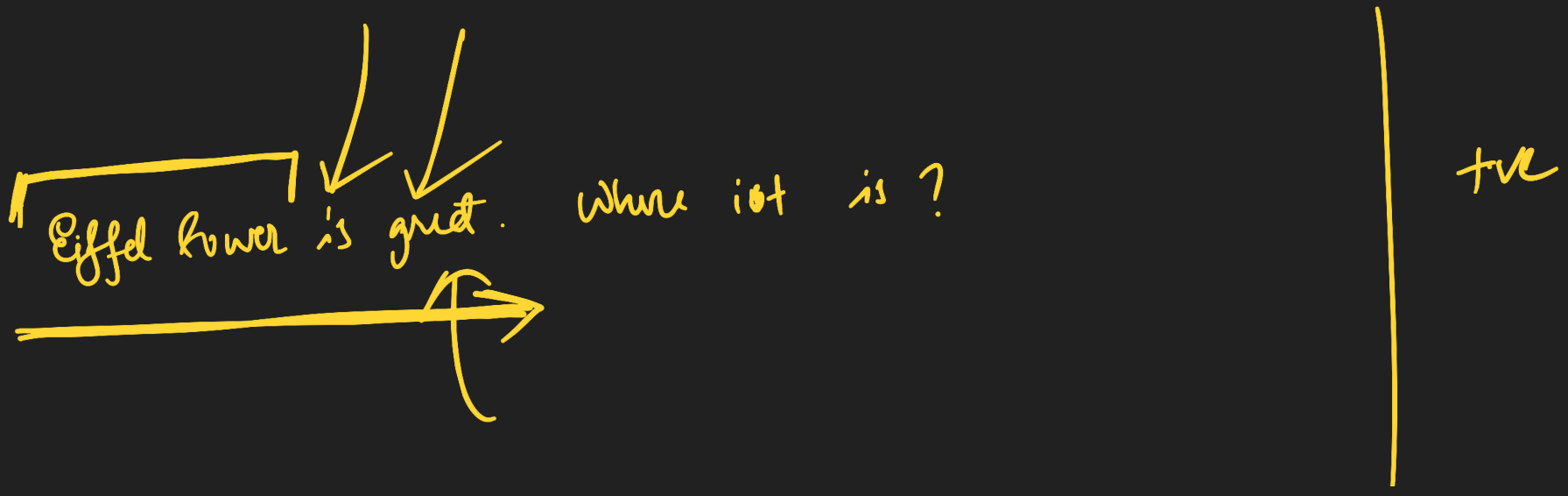
$h_t = F(x_t, h_{t-1})$



1 | 10

Word

But do we pass the words as it is to RNN cell?

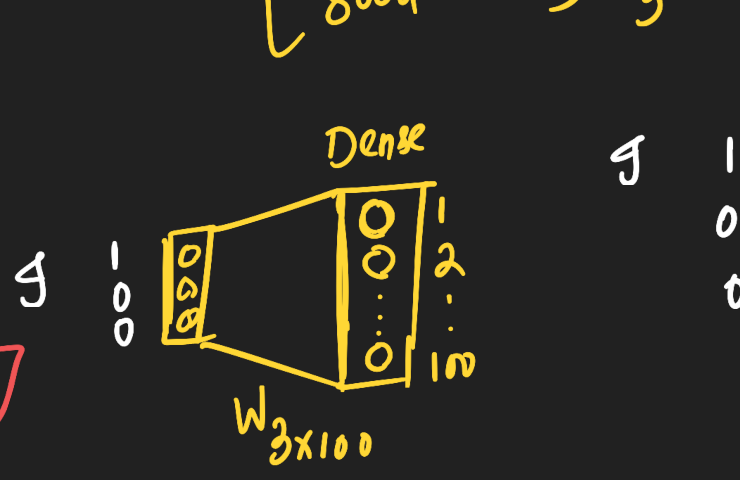


But do we pass the words as it is to RNN cell?

1] Pretrained embedding

2] Trainable embedding layer

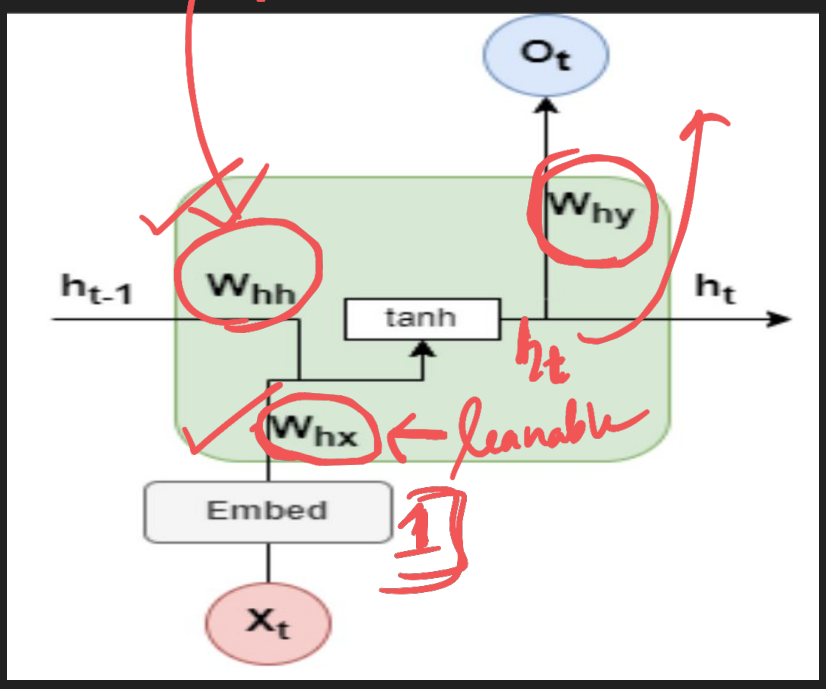
vocab { I → 1, am → 2, good → 3

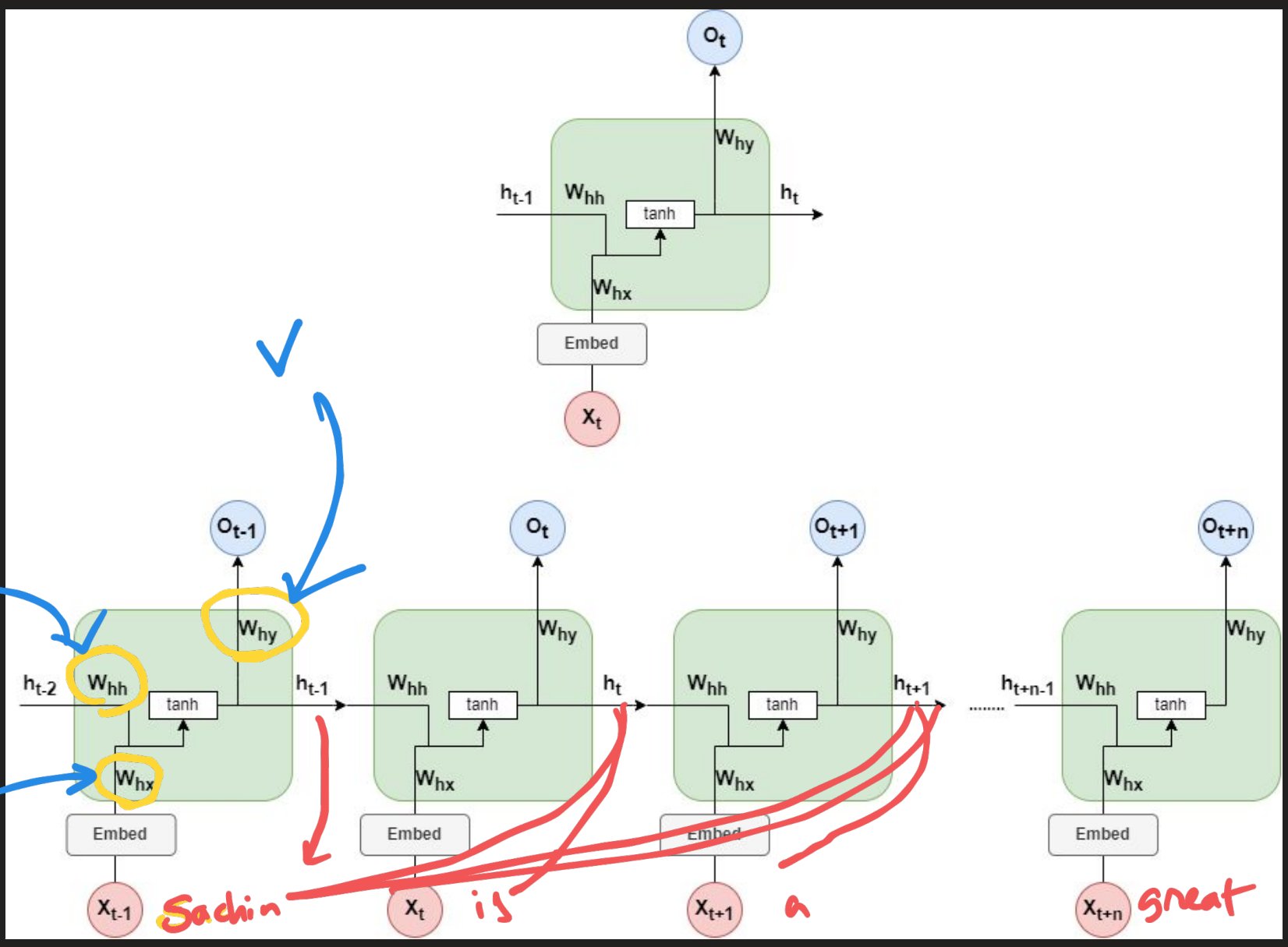


I 1 am 0 good 0
0 1 0
0 0 1

↑ Weight is trainable

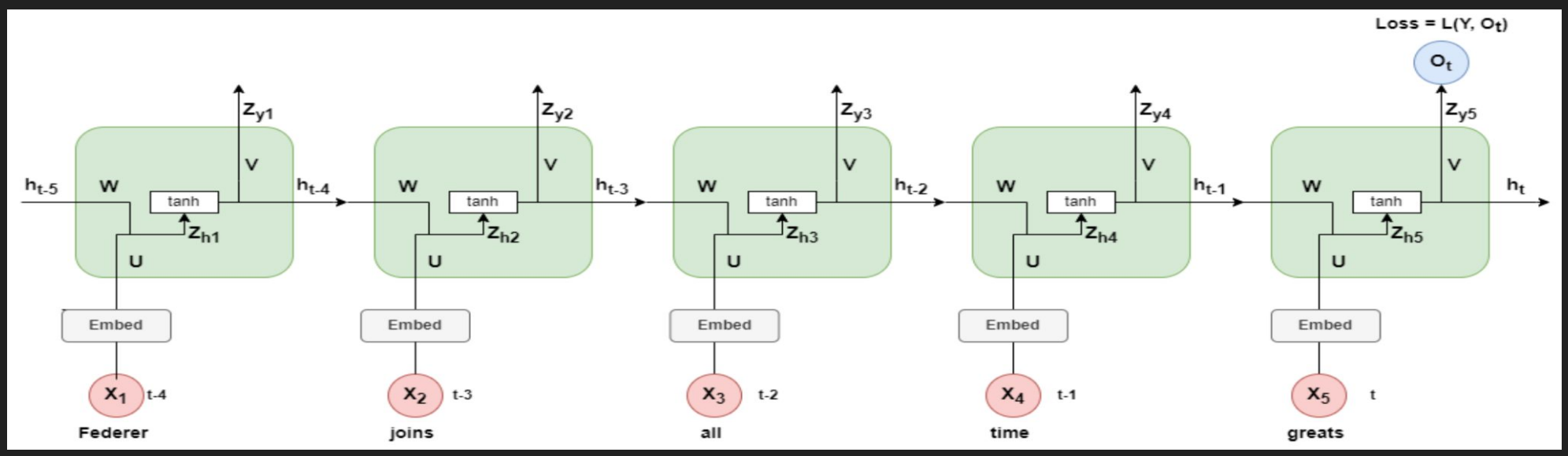
What is being learnt?

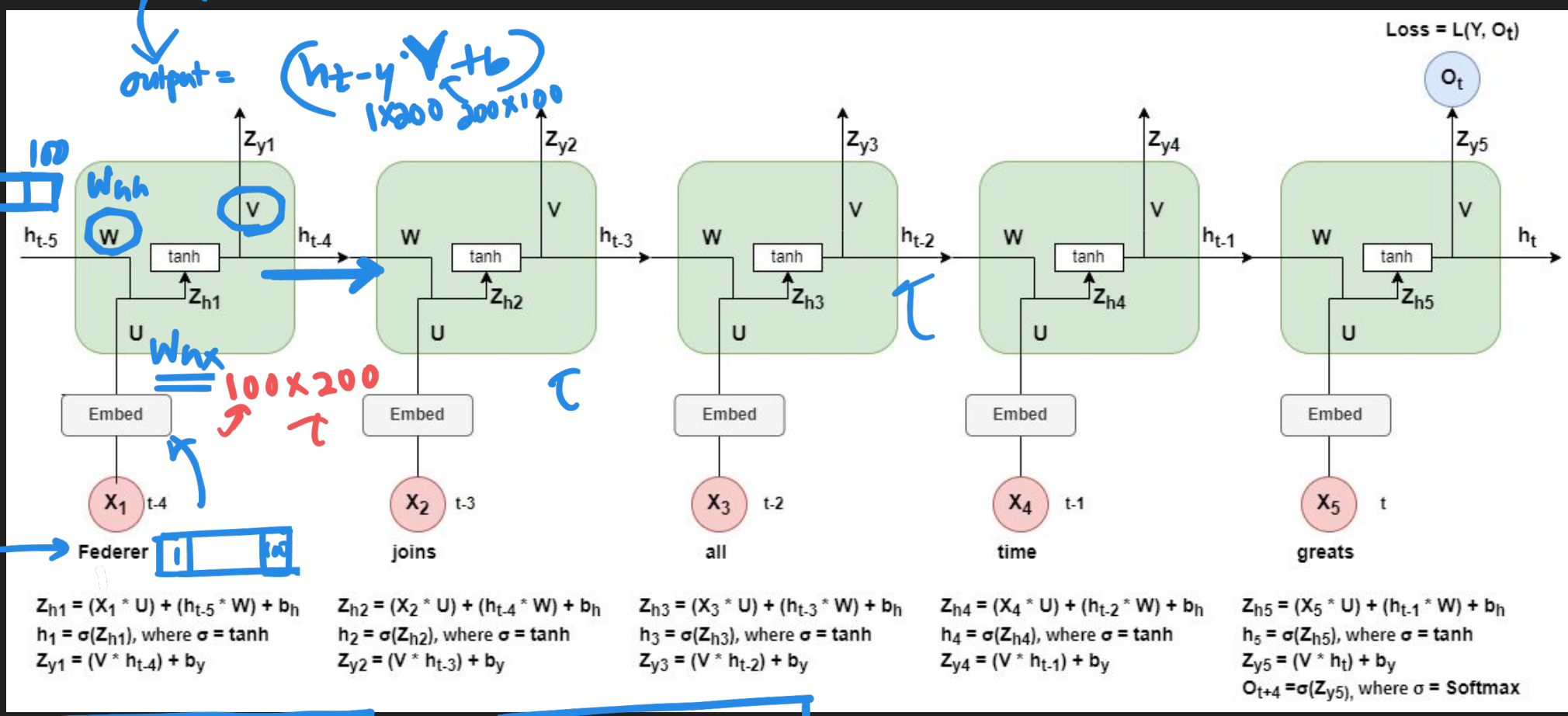


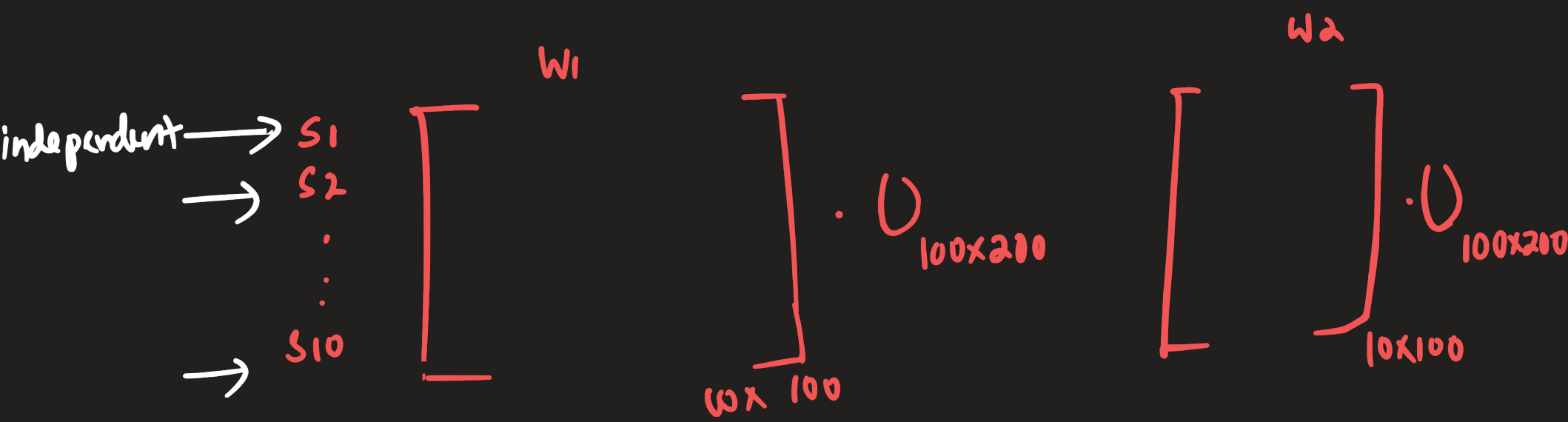

O_t
W_{hy}
h_{t-1}
W_{hh}
tanh
h_t
W_{hx}
Embed
x_t
O_{t-1}
O_t
O_{t+1}
O_{t+n}
W
U
h_{t-2}
h_{t+1}
h_{t+n-1}
x_{t-1}
x_{t+1}
x_{t+n}
Sachin
is
a
great

RNN Forward Pass

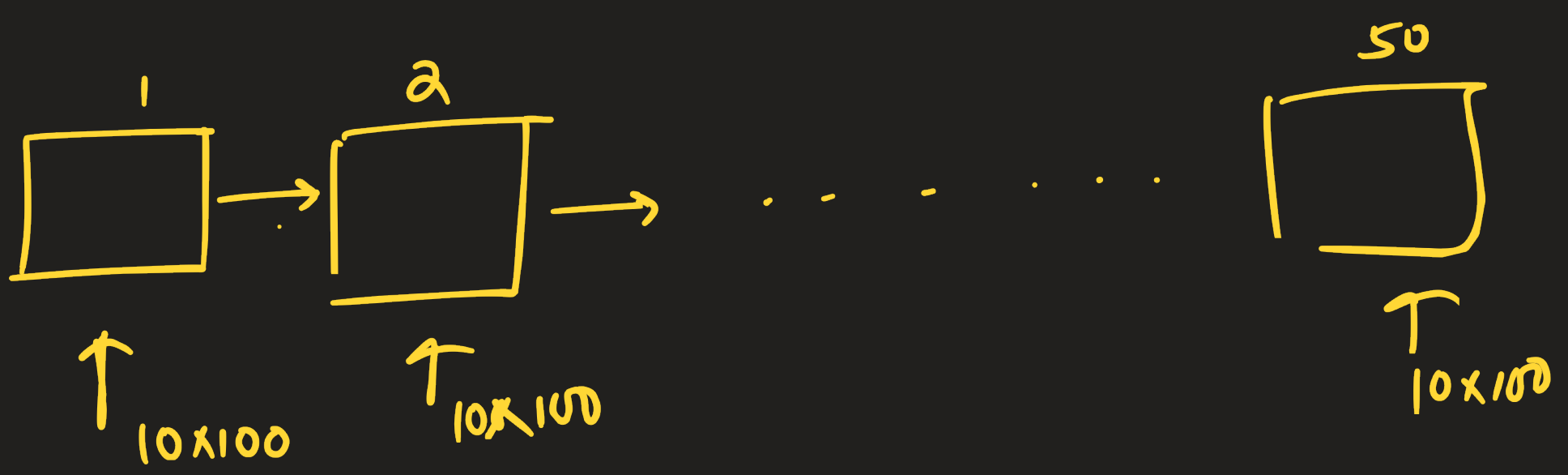
RNN Forward Pass



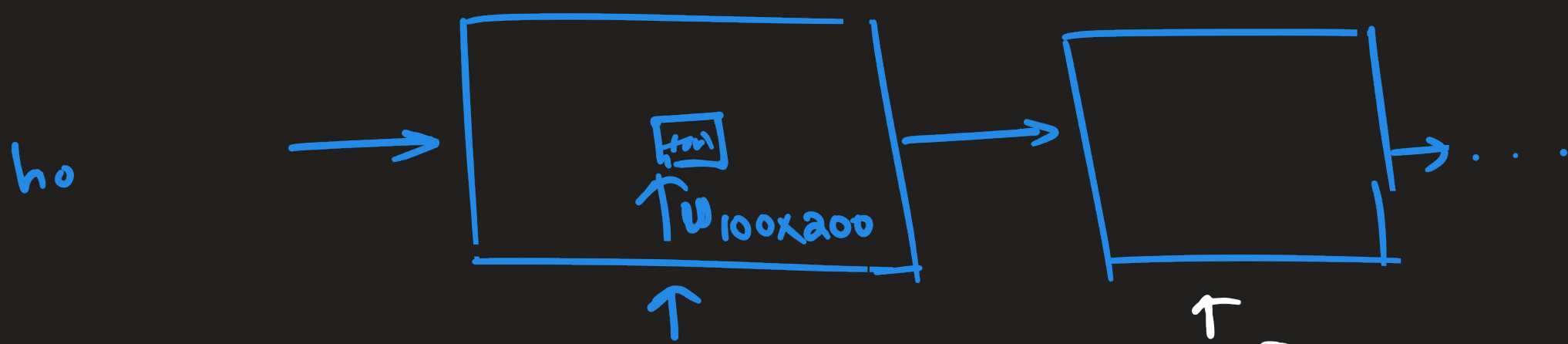

Loss = L(Y, Ot)
Ot
Zy1
Zy2
Zy3
Zy4
Zy5
V
W
U
tanh
Zh1
Zh2
Zh3
Zh4
Zh5
ht-5
ht-4
ht-3
ht-2
ht-1
ht
Embed
X1 t-4
X2 t-3
X3 t-2
X4 t-1
X5 t
Federer
joins
all
time
greats
Zh1 = (X1 * U) + (ht-5 * W) + bh
h1 = σ(Zh1), where σ = tanh
Zy1 = (V * ht-4) + by
Zh2 = (X2 * U) + (ht-4 * W) + bh
h2 = σ(Zh2), where σ = tanh
Zy2 = (V * ht-3) + by
Zh3 = (X3 * U) + (ht-3 * W) + bh
h3 = σ(Zh3), where σ = tanh
Zy3 = (V * ht-2) + by
Zh4 = (X4 * U) + (ht-2 * W) + bh
h4 = σ(Zh4), where σ = tanh
Zy4 = (V * ht-1) + by
Zh5 = (X5 * U) + (ht-1 * W) + bh
h5 = σ(Zh5), where σ = tanh
Zy5 = (V * ht) + by
Ot+4 =σ(Zy5), where σ = Softmax



10 Sentences × 50 words × 100 size of embedding



how many such timesteps = 5



no. of sentences

10 sentences = 10 x 50 x 100

size of each embedding

each sentence 100 words

X . U → 10 x 200

10x100 100x200

input repn of 10 sentences but only the 1st words

Steps

U x current word — What does this word mean

W x previous memory — What I was thinking so far

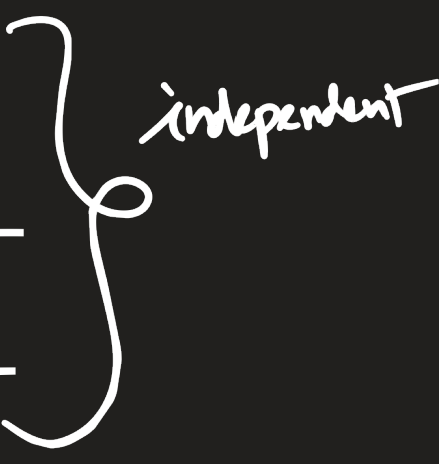
V x memory — Based on my understanding so far what should I answer

9pm

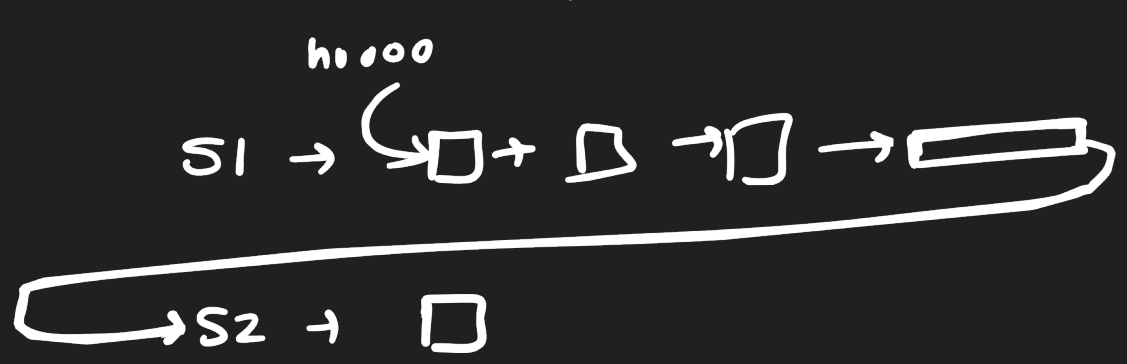
S1 W1 W2 W3 <PAD>

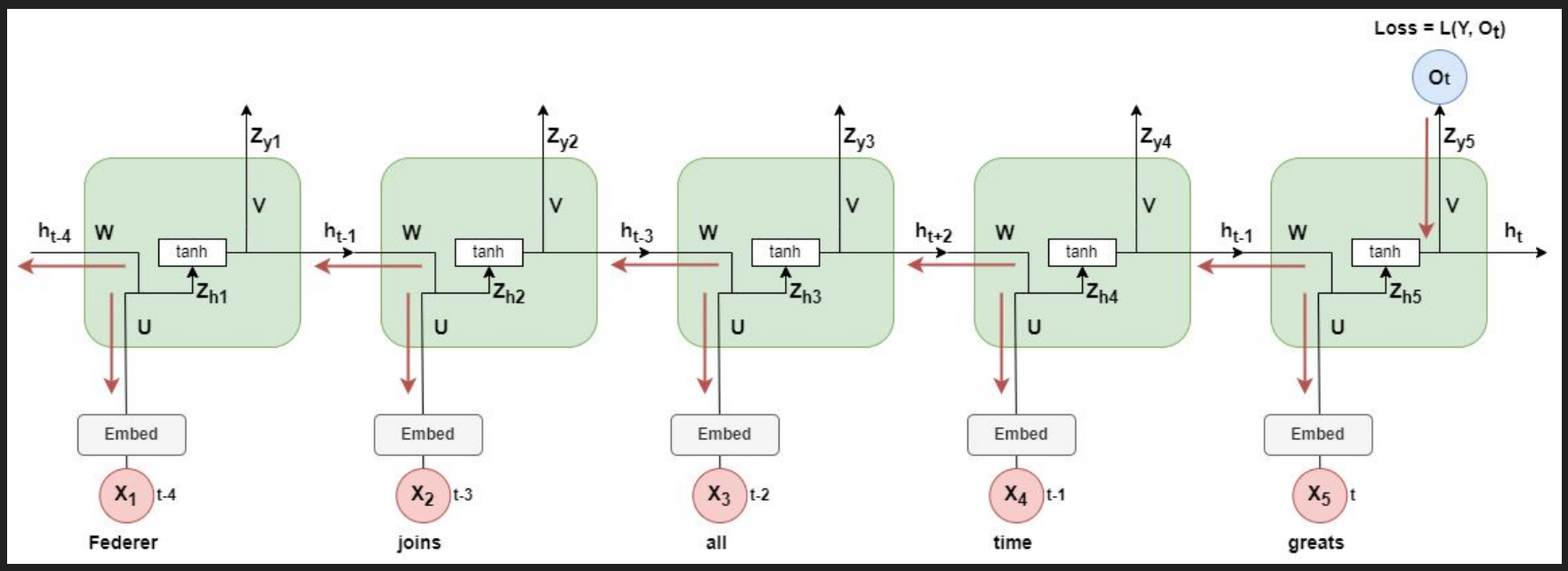
S2 W1 W2 W3 W4

S3 W1 W2 <PAD> <PAD>



1 1 0 0

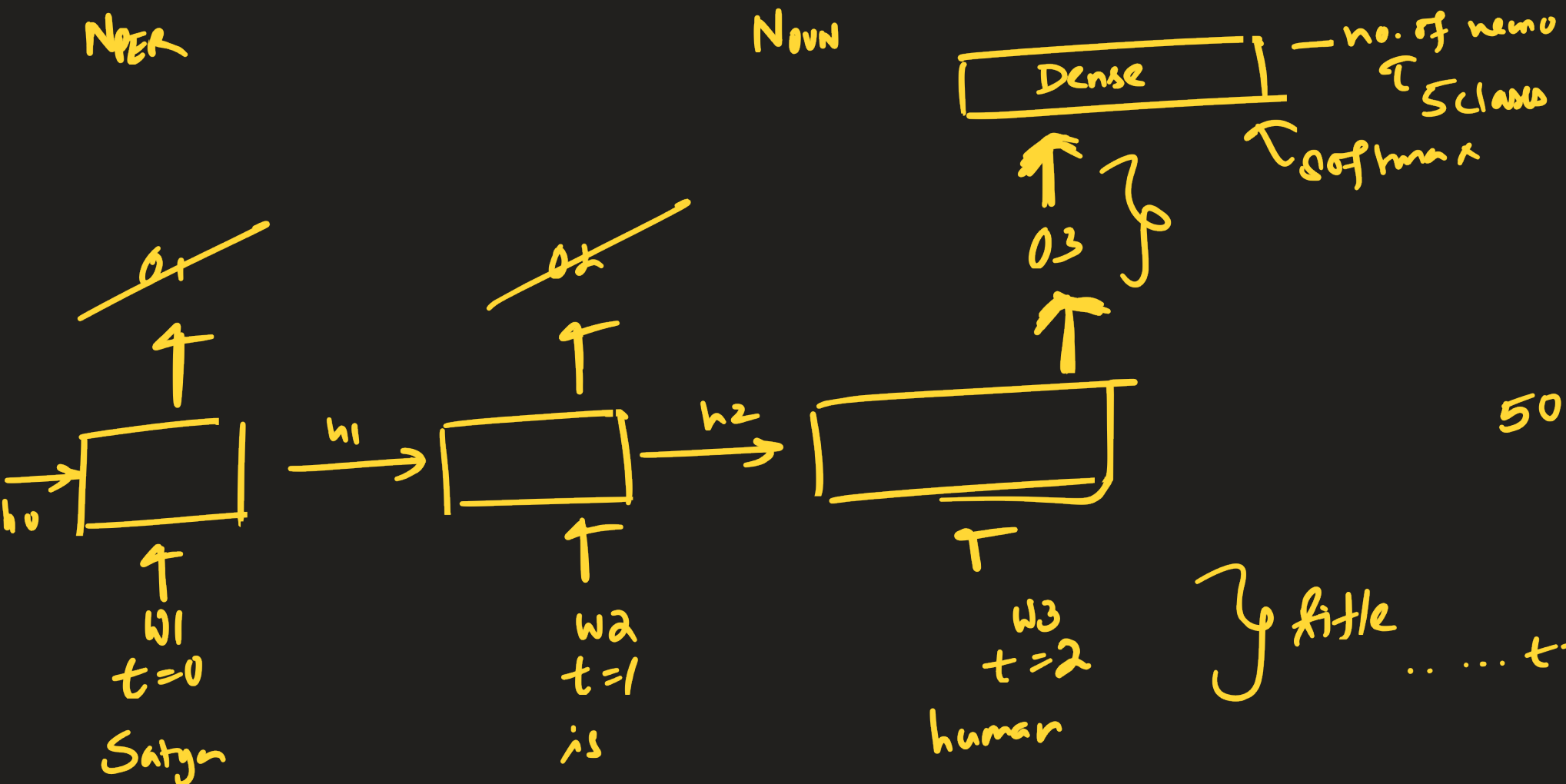



Loss = L(Y, Ot)
Ot
Zy1
Zy2
Zy3
Zy4
Zy5
V
W
tanh
Zh1
Zh2
Zh3
Zh4
Zh5
U
ht-4
ht-1
ht-3
ht+2
ht-1
ht
Embed
X1 t-4
X2 t-3
X3 t-2
X4 t-1
X5 t
Federer
joins
all
time
greats

Great god is

{god is great}

Weight Updates

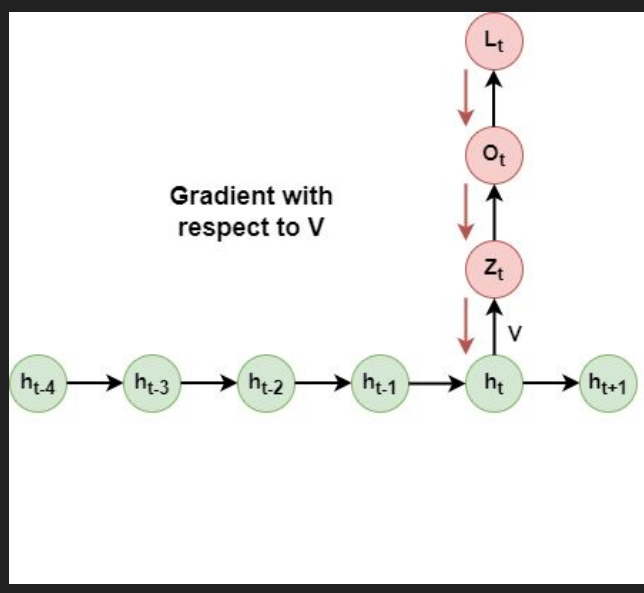


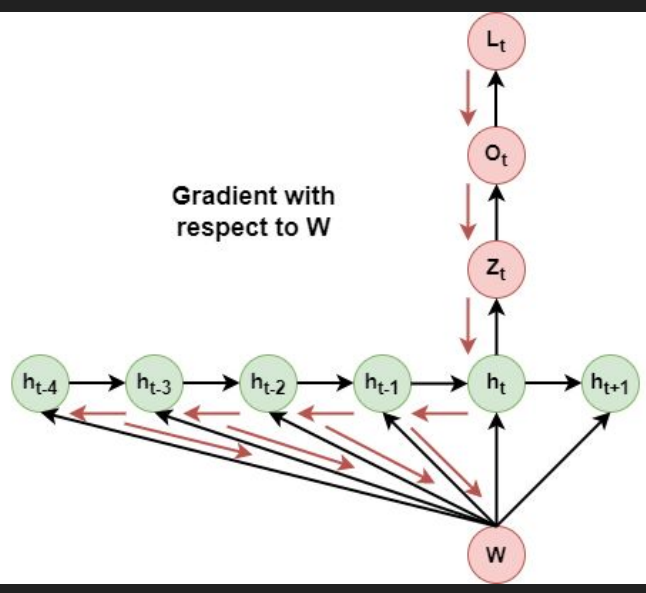
50

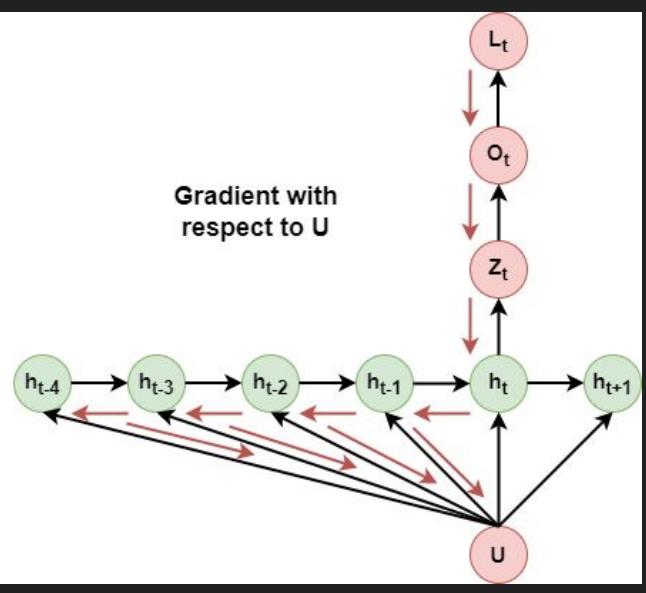
1AM x
2AM y
3AM z

g am good
t=0 t=1 t=2

Q.10


Gradient with
respect to V
L_t
O_t
Z_t
V
h_{t-4}
h_{t-3}
h_{t-2}
h_{t-1}
h_t
h_{t+1}


Gradient with
respect to W
L_t
O_t
Z_t
h_{t-4}
h_{t-3}
h_{t-2}
h_{t-1}
h_t
h_{t+1}
W


Gradient with
respect to U
L_t
O_t
Z_t
h_{t-4}
h_{t-3}
h_{t-2}
h_{t-1}
h_t
h_{t+1}
U

Loss

$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial O_t}\frac{\partial O_t}{\partial Z_t}\frac{\partial Z_t}{\partial h_t}\frac{\partial h_t}{\partial W} +$$

$$\frac{\partial L_t}{\partial O_t}\frac{\partial O_t}{\partial Z_t}\frac{\partial Z_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial W} +$$

$$\frac{\partial L_t}{\partial O_t}\frac{\partial O_t}{\partial Z_t}\frac{\partial Z_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial h_{t-2}}\frac{\partial h_{t-2}}{\partial W} +$$

$$\frac{\partial L_t}{\partial O_t}\frac{\partial O_t}{\partial Z_t}\frac{\partial Z_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial h_{t-2}}\frac{\partial h_{t-2}}{\partial h_{t-3}}\frac{\partial h_{t-3}}{\partial W} +$$

$$\frac{\partial L_t}{\partial O_t}\frac{\partial O_t}{\partial Z_t}\frac{\partial Z_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial h_{t-2}}\frac{\partial h_{t-2}}{\partial h_{t-3}}\frac{\partial h_{t-3}}{\partial h_{t-4}}\frac{\partial h_{t-4}}{\partial W} +$$

$$\frac{\partial L_t}{\partial O_t}\frac{\partial O_t}{\partial Z_t}\frac{\partial Z_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial h_{t-2}}\frac{\partial h_{t-2}}{\partial h_{t-3}}\frac{\partial h_{t-3}}{\partial h_{t-4}}\frac{\partial h_{t-4}}{\partial h_{t-5}}\frac{\partial h_{t-5}}{\partial W}$$

Let's generalize this to a single formula

$$\frac{\partial \mathbf{L_t}}{\partial \mathbf{W}} = \frac{\partial \mathbf{L_t}}{\partial \mathbf{O_t}} \frac{\partial \mathbf{O_t}}{\partial \mathbf{Z_t}} \frac{\partial \mathbf{Z_t}}{\partial \mathbf{h_t}} \sum_{r=1}^{t} \frac{\partial \mathbf{h_t}}{\partial \mathbf{h_r}} \frac{\partial \mathbf{h_r}}{\partial \mathbf{W}}$$

Update the weight matrix W

$$W = W - \alpha \frac{\partial \mathbf{L_t}}{\partial \mathbf{W}}$$

Sachin Tendulkar is a Cricketer

Sachin Tendulkar is a MP of Rajya Sabha

$$\overrightarrow{h_t} = f(U x_t + W \overrightarrow{h_{t-1}})$$

$$\overleftarrow{h_t} = f'(U' x_t + W' \overleftarrow{h_{t+1}})$$

$$o_t = \underset{avg}{concat} / [\overrightarrow{h_t}, \overleftarrow{h_t}]$$

tve/-ve

I am good

GOV

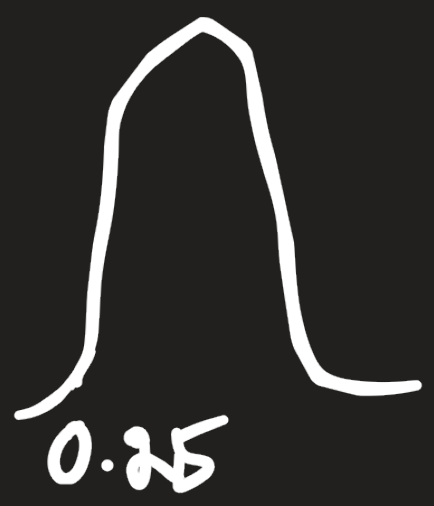
PER DES ORG

Sachin r/c MP from Rajyasabha-

batsman

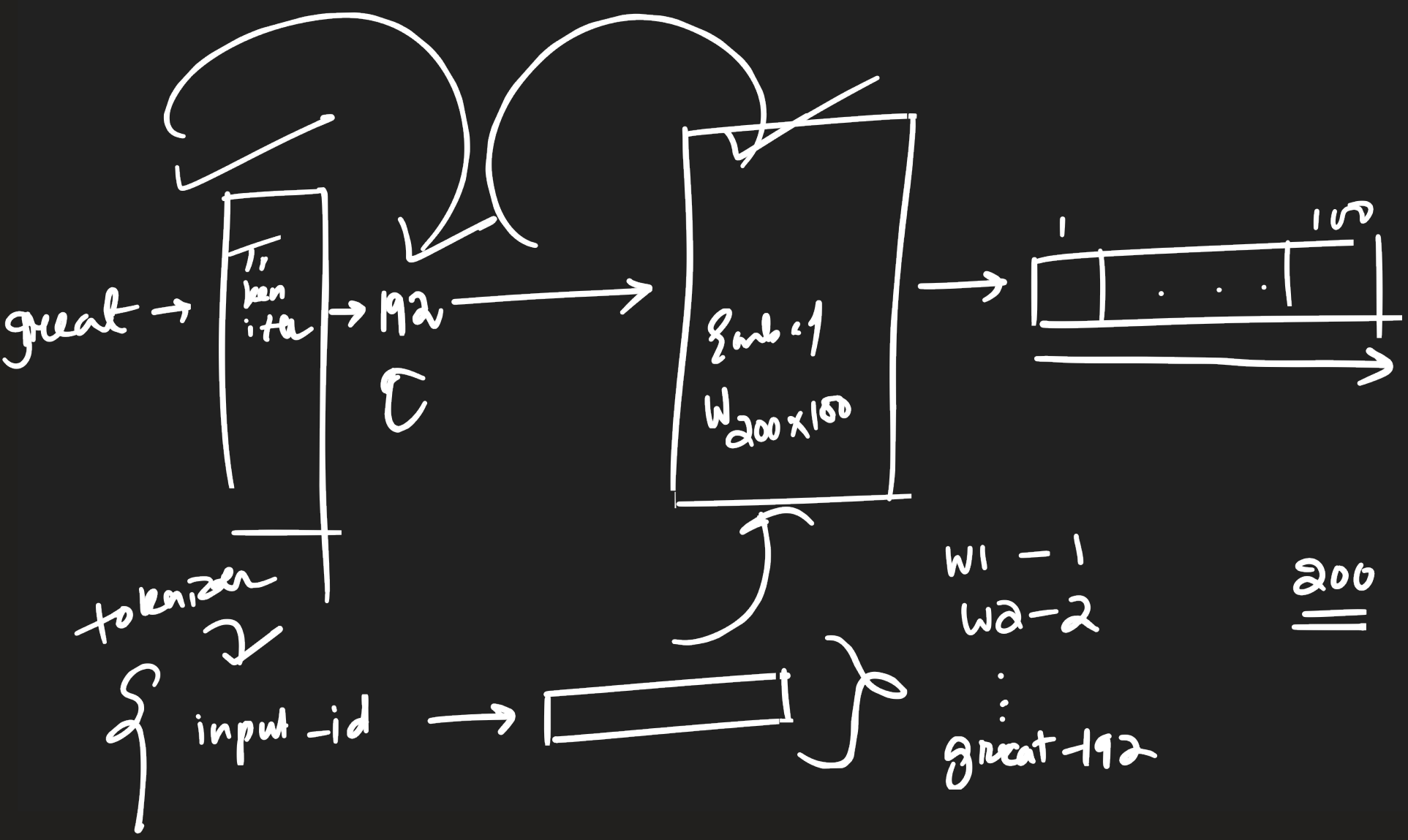
PLAYER

tan'h = 1



(0.25 * 1.25) ***** ≅ 0

Sequence words


great →
Tokenizer
→ 192
C
Embedding
W200×100
1
100
tokenizer
2
input-id
W1 - 1
W2 - 2
great - 192
200

10

500

(None, None, 100)

1

(None, 50)

1 2 . . . 500

S1

S10

each worker

1 10

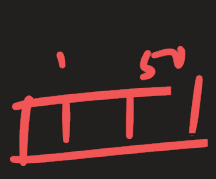
1 50

t=1 W1

t=2 W2

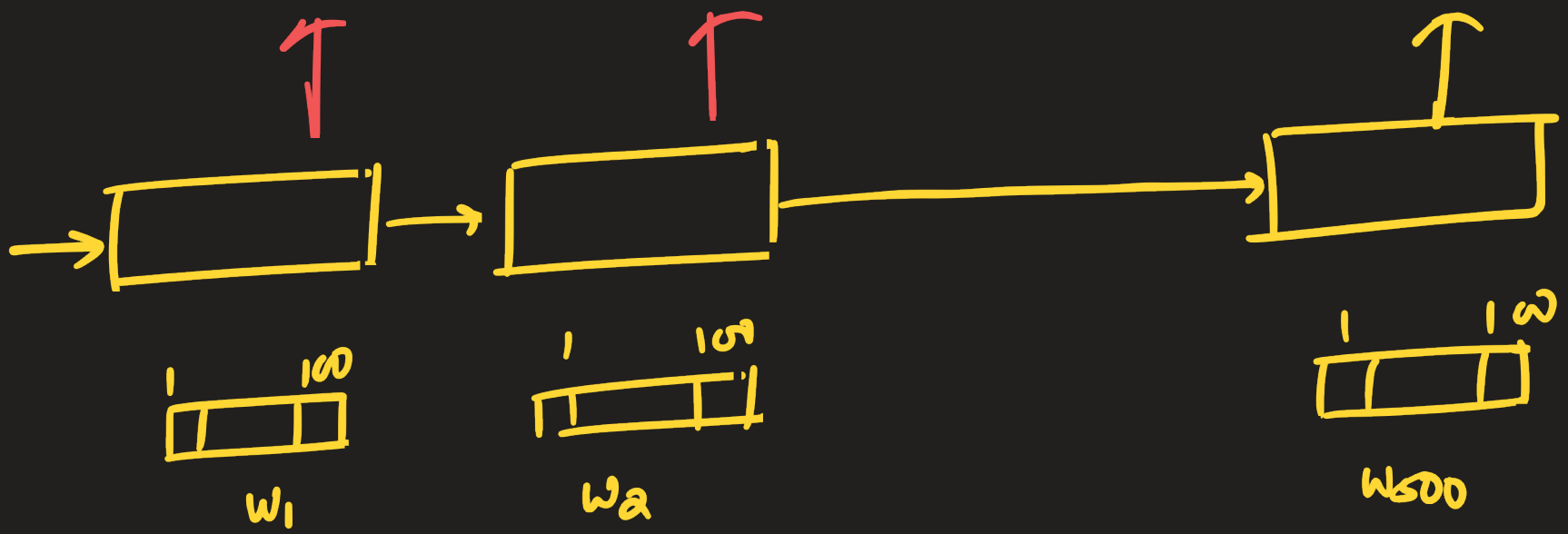
t=500 W500

Return Seq = True

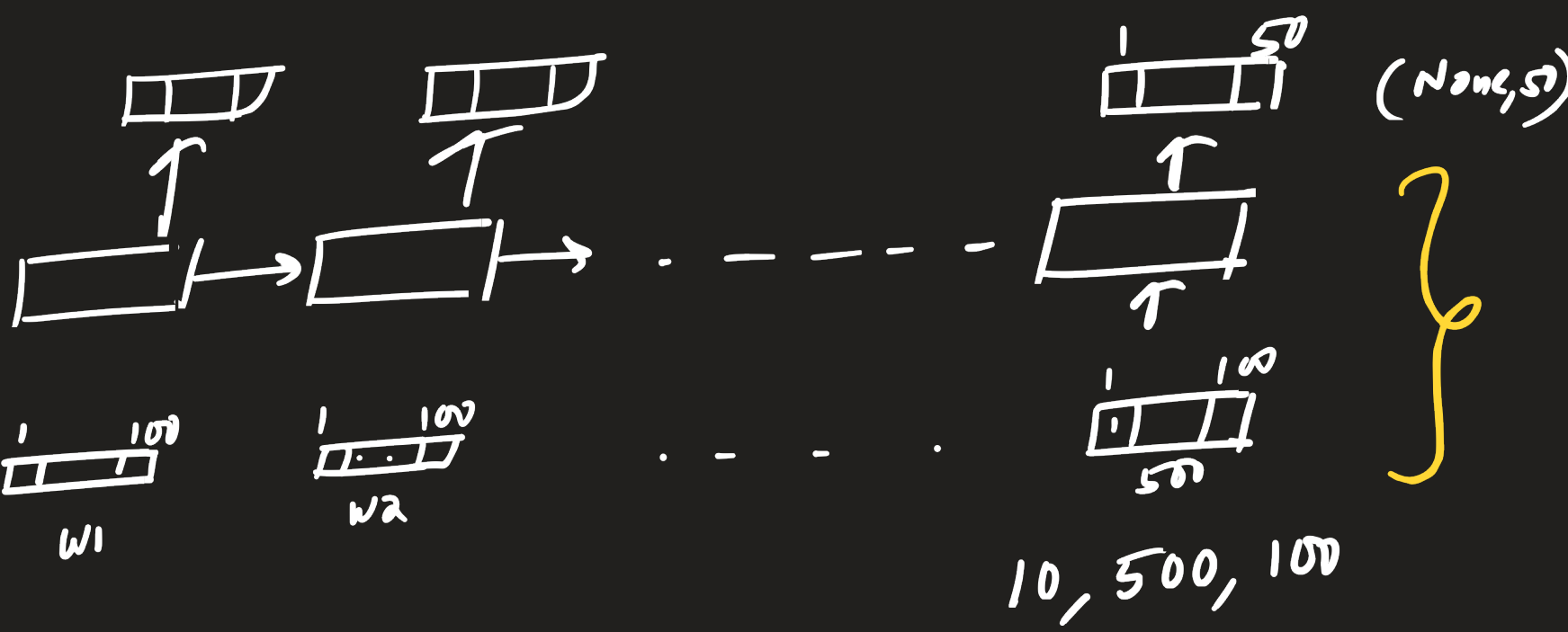


Return Seq = False

SN Softmax



10

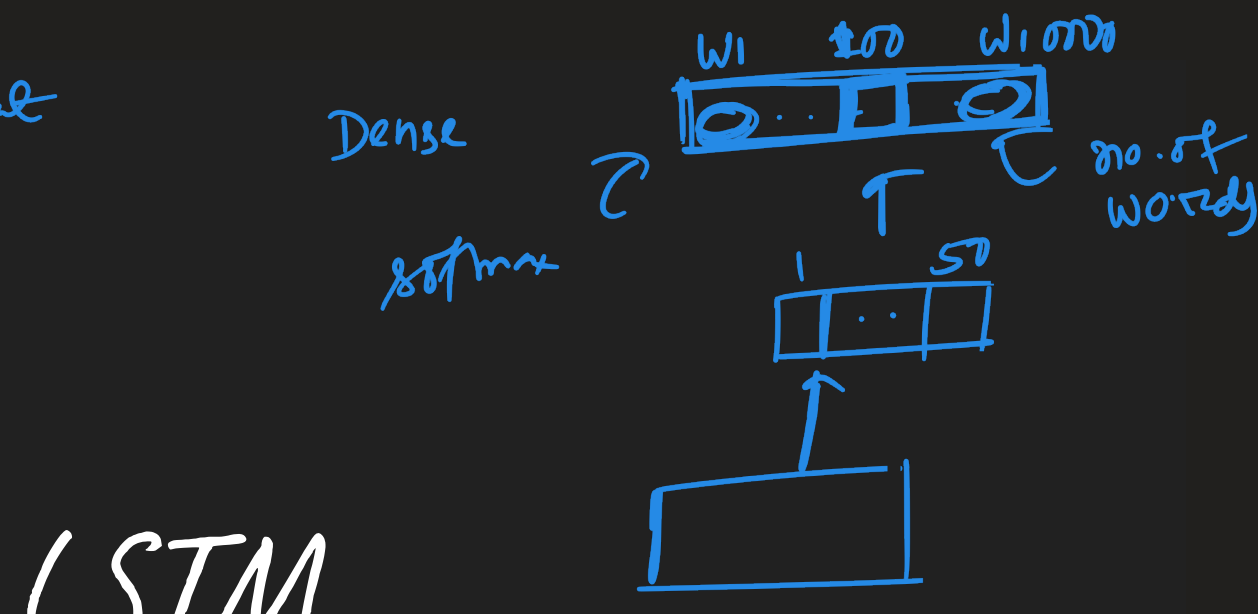


no. of sentence, each sentence of 500

None, None, 100 ← word dim = 100

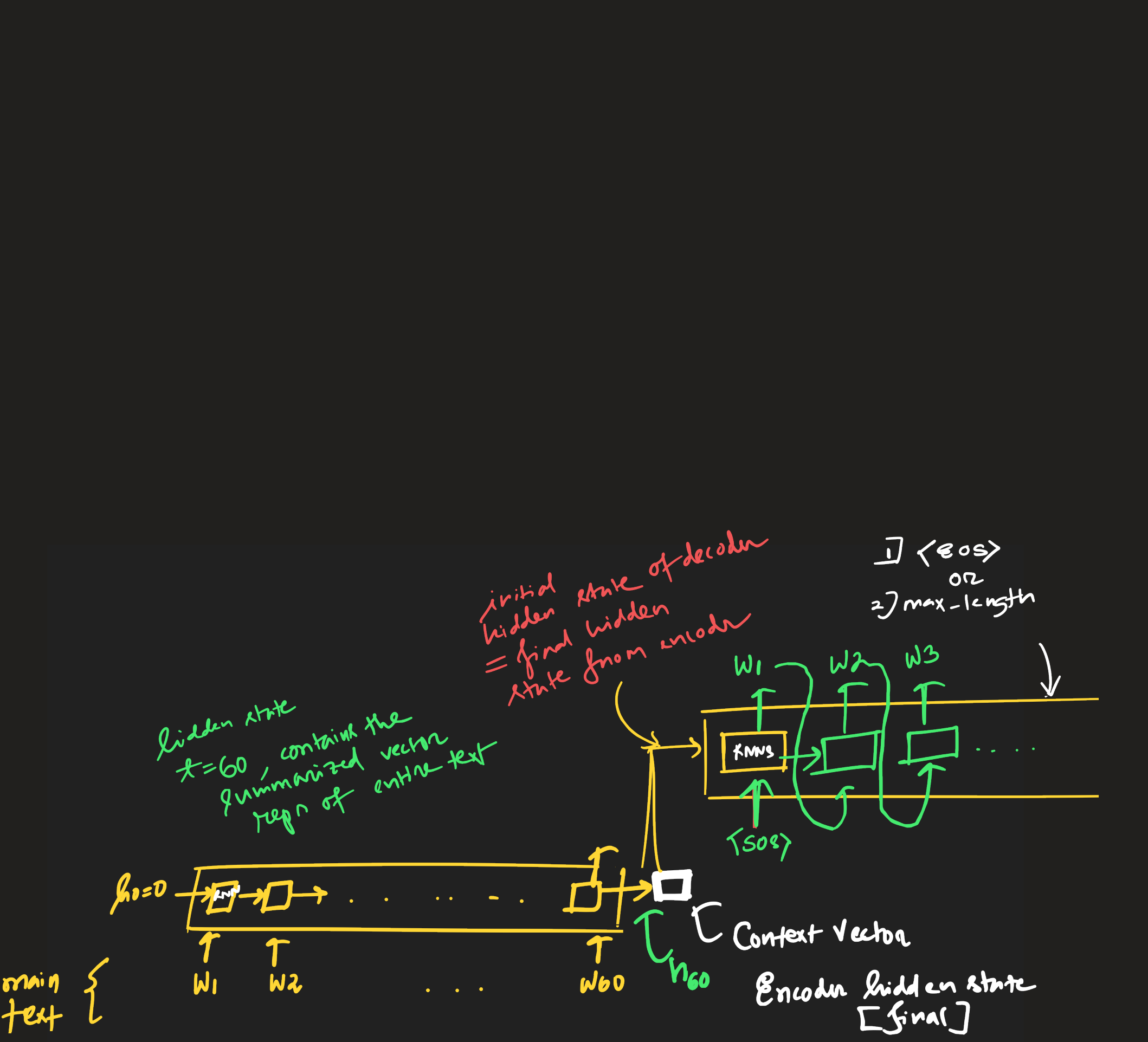
None, 1 , 50:

100 → great



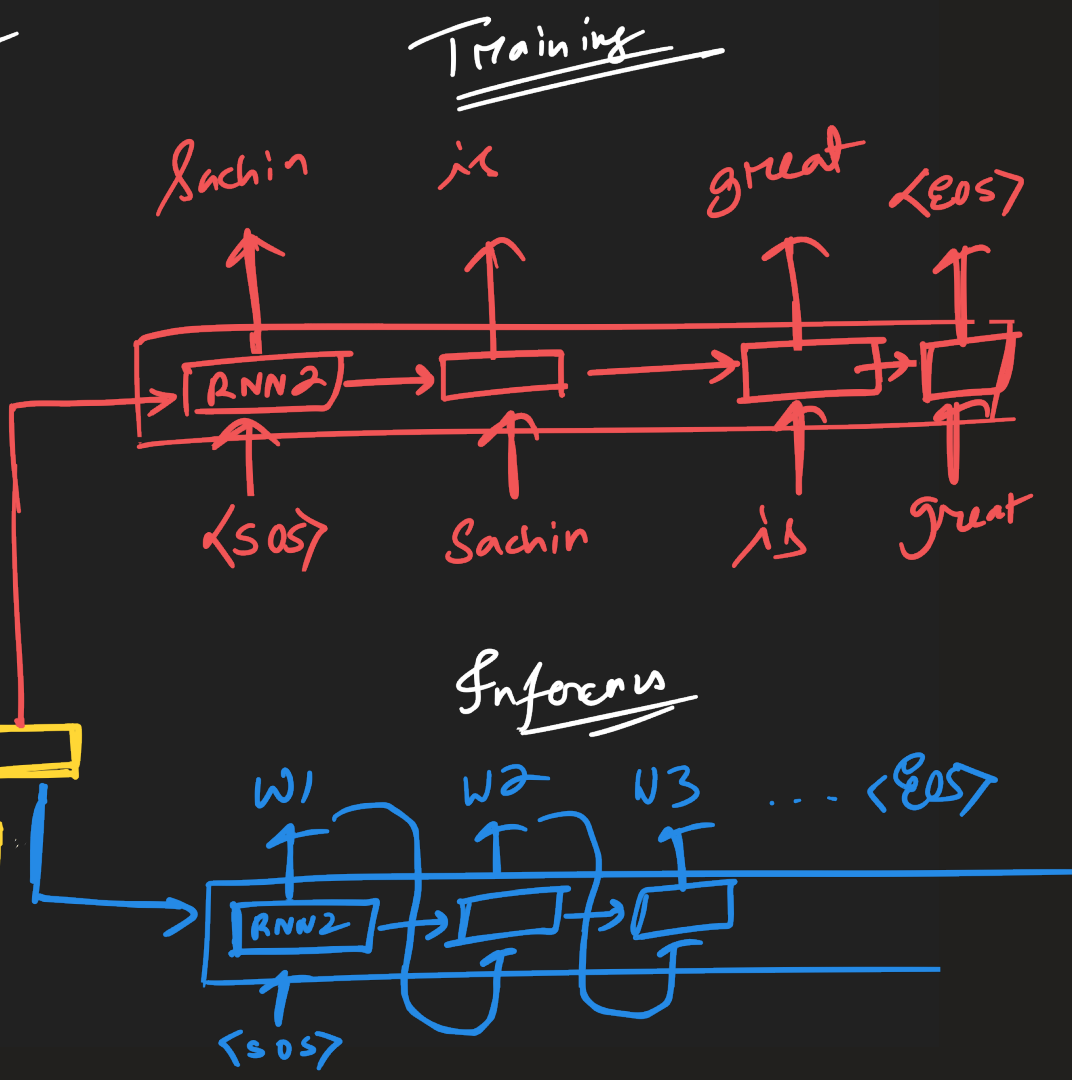
LSTM

Satya Pattnaik


initial hidden state of decoder = final hidden state from encoder
1] <EOS> or 2] max_length
W1
W2
W3
RNN
<SOS>
hidden state t=60, contains the summarized vector repn of entire text
h0=0
W1
W2
W60
h60
Context Vector
Encoder hidden state [final]
main text

X	y
Sachin is a good cricketer, He is from Maharashtri	Sachin is great

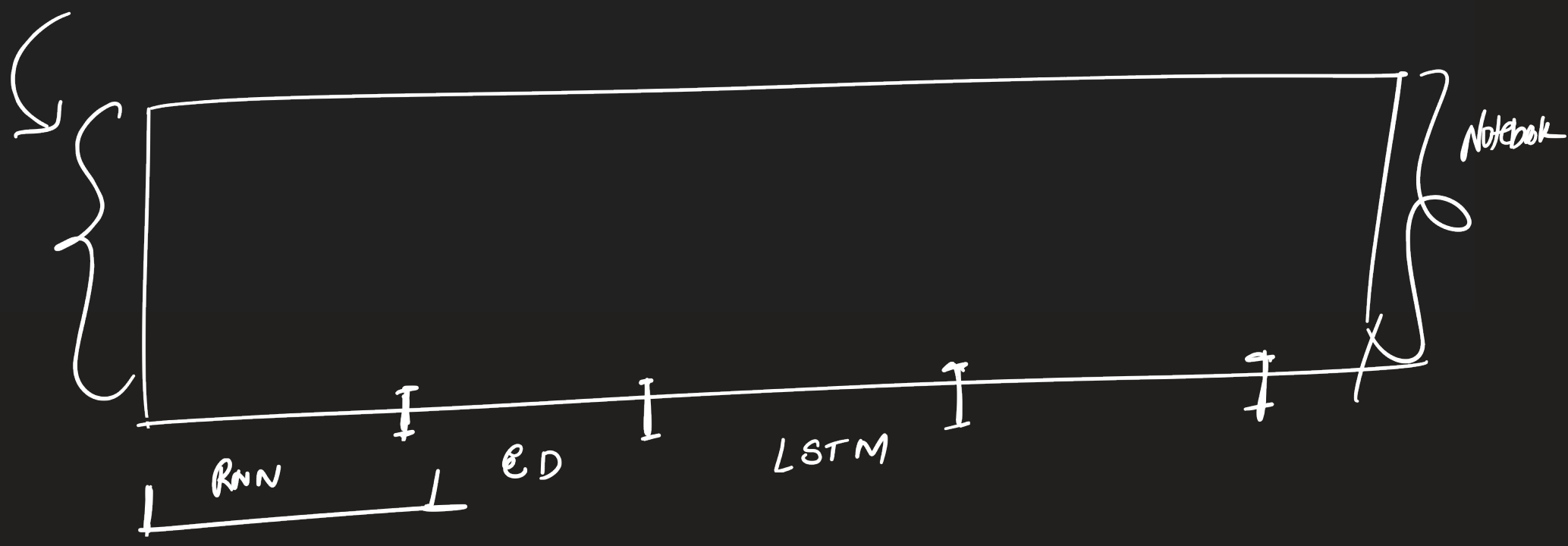
q

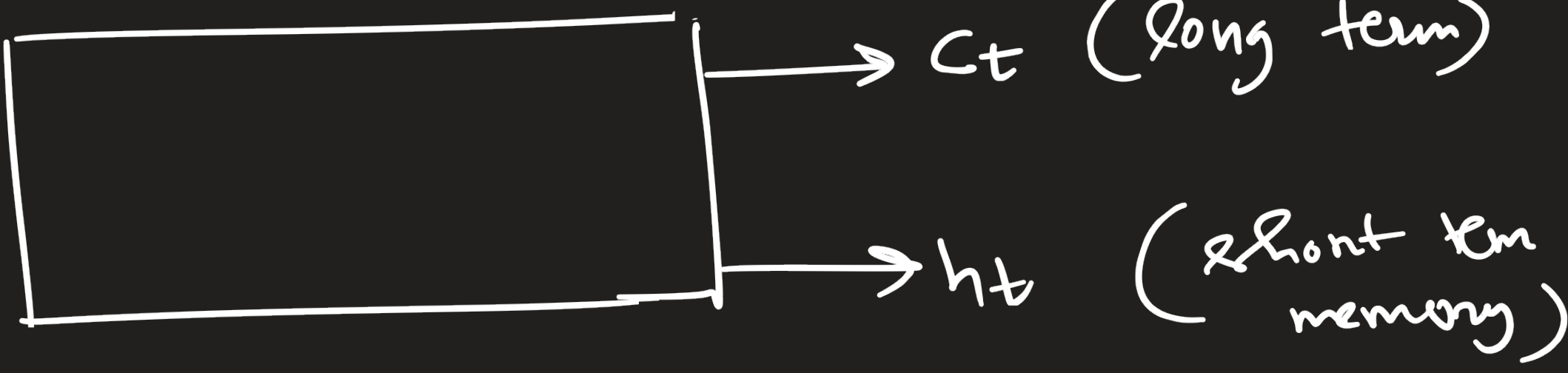


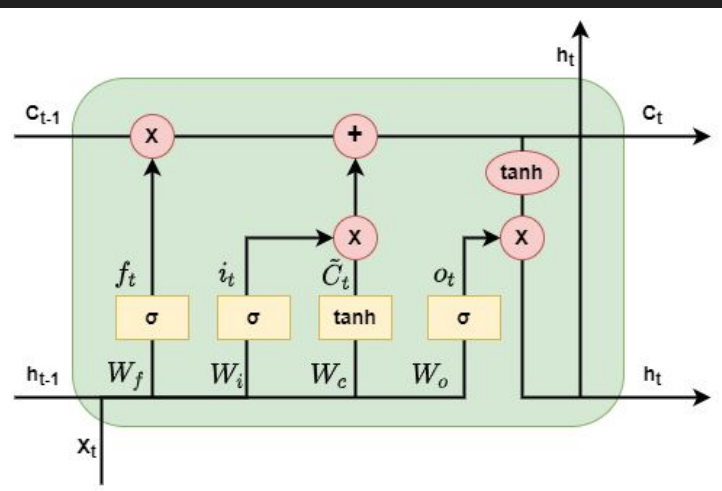
W1 . . . - W00

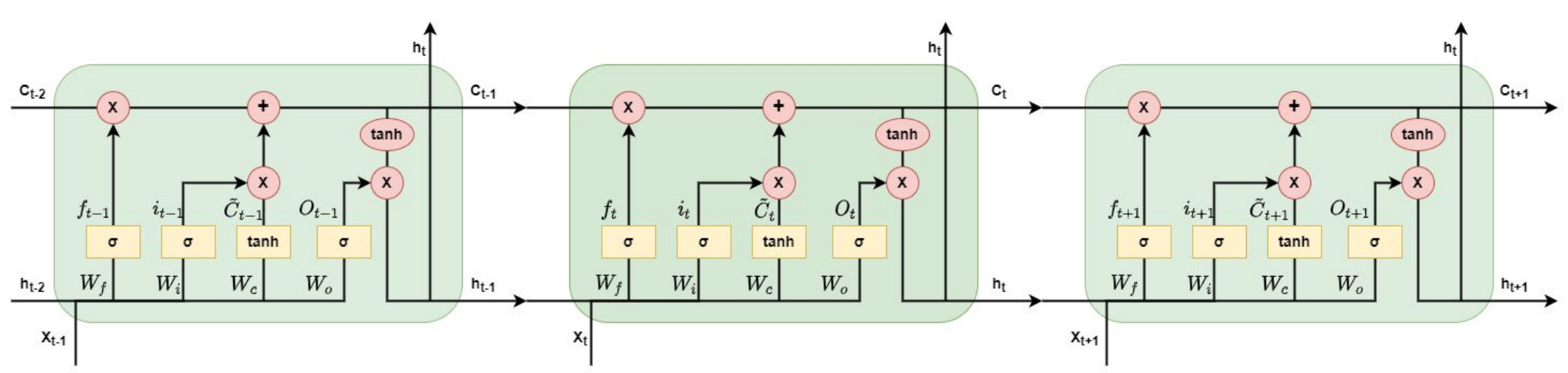
Long term dependency not taken into acount

Bottleneck Problem —

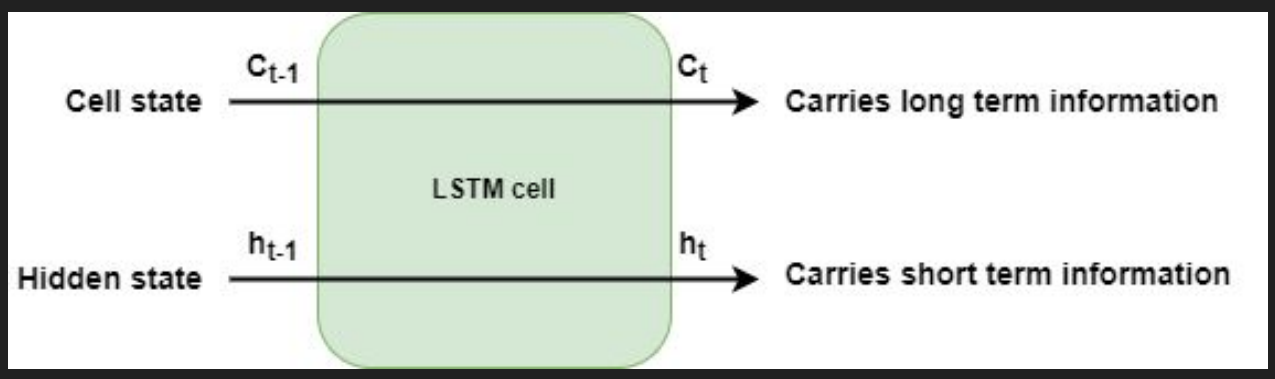



Ct (long term)
ht (short term memory)


h_t
C_{t-1}
C_t
x
+
tanh
x
x
f_t
i_t
$\tilde{C}_t$
o_t
σ
σ
tanh
σ
h_{t-1}
W_f
W_i
W_c
W_o
h_t
X_t


h_t
C_{t-2}
C_{t-1}
x
+
tanh
x
x
f_{t-1}
i_{t-1}
$\tilde{C}_{t-1}$
O_{t-1}
σ
σ
tanh
σ
h_{t-2}
W_f
W_i
W_c
W_o
h_{t-1}
X_{t-1}
h_t
C_t
f_t
i_t
$\tilde{C}_t$
O_t
h_t
X_t
h_t
C_{t+1}
f_{t+1}
i_{t+1}
$\tilde{C}_{t+1}$
O_{t+1}
h_{t+1}
X_{t+1}

How does LSTM overcome short term memory problem

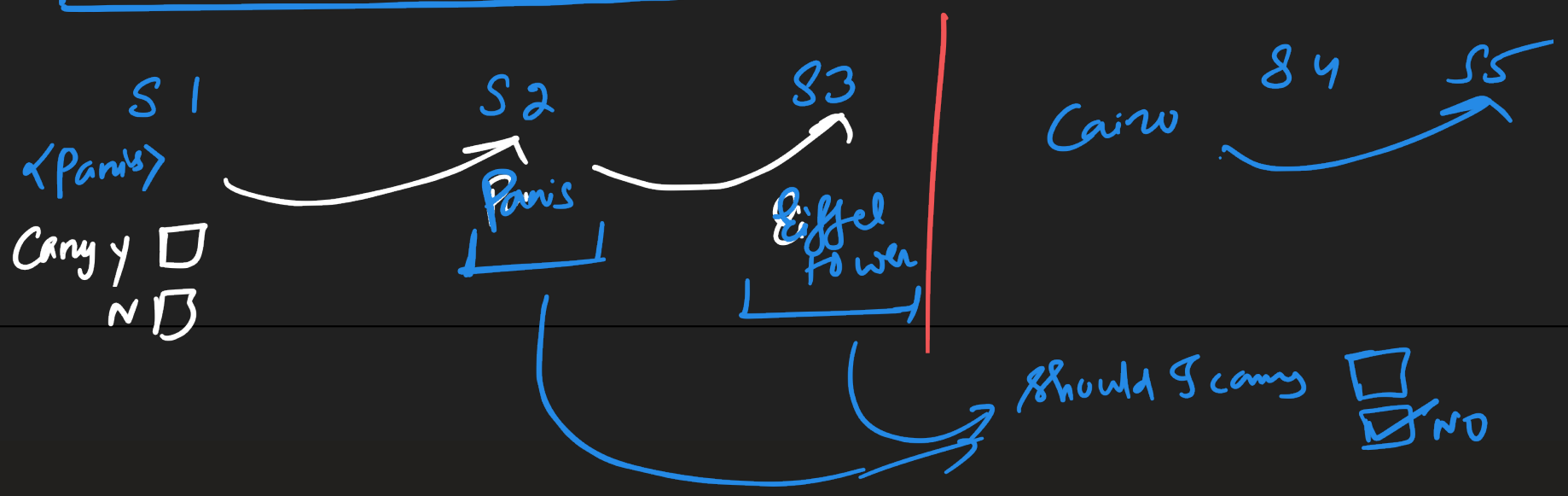


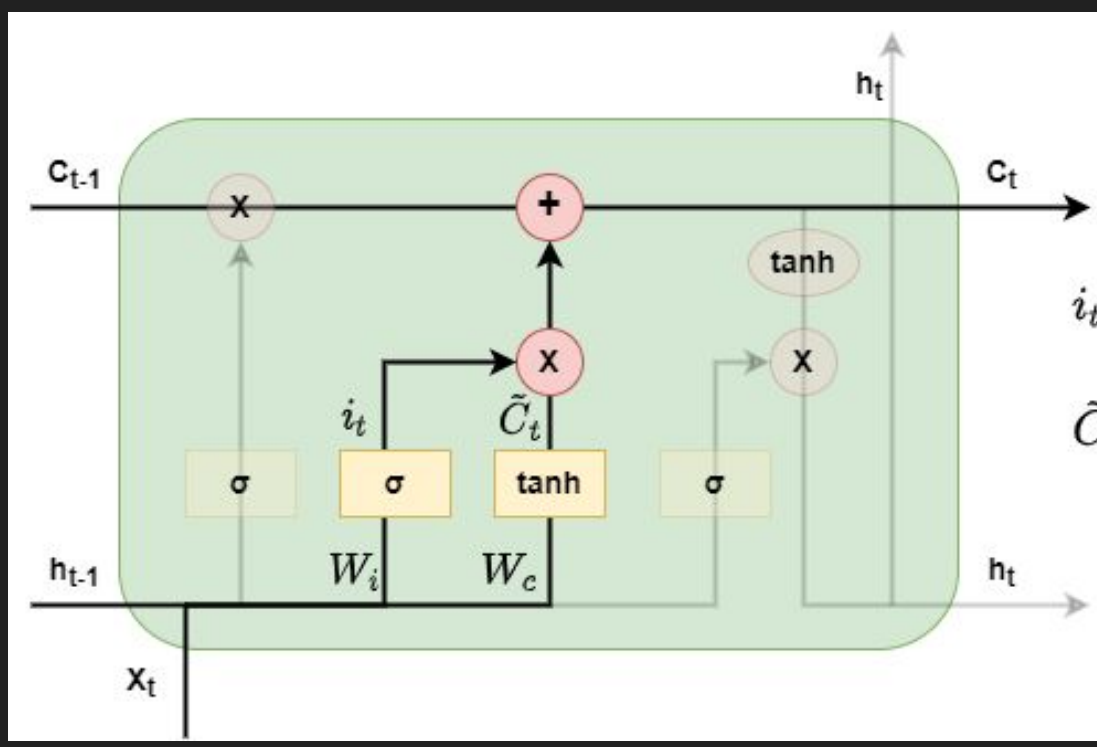
Let's understand it with an example sentence

Sentence: I like to travel and I visited **Paris** last year as a part of my business trip. It was winter season and too cold for someone from the sub-continent. I stayed at Hilton hotel and I was fortunate enough to get a room where from my window I could see the **Eiffel tower**. I took my family on vacation to **Cairo**, Egypt last month. They were quite a lot of market area for shopping to buy souvenirs and my kids enjoyed seeing the great **pyramids** for the first time.

S1 S2 S3 S4 S5

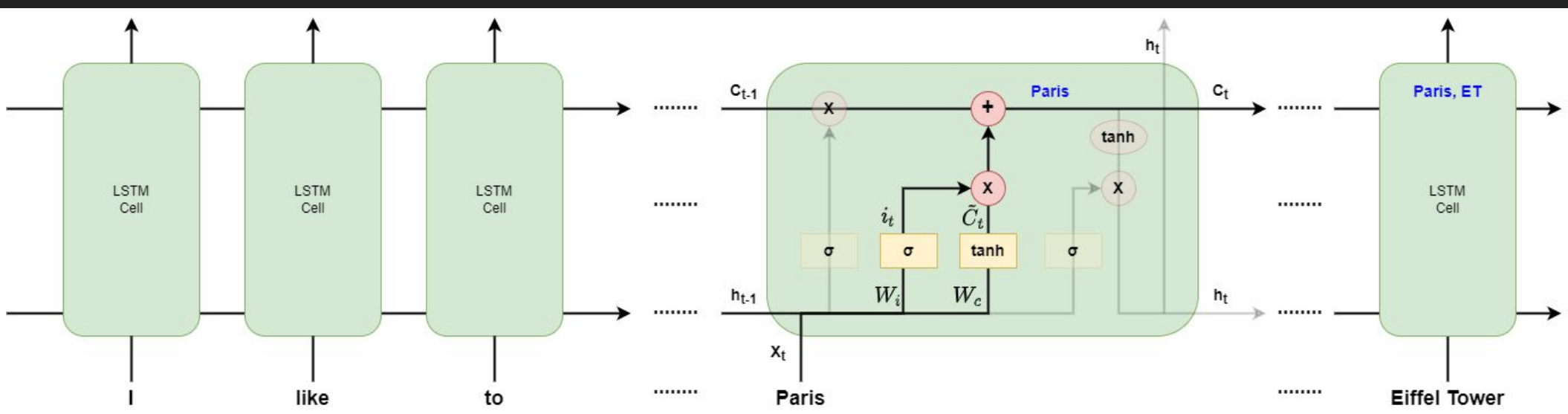
Summary: I saw Eiffel tower when I was in Paris and during my vacation I visited the Pyramids in Cairo.

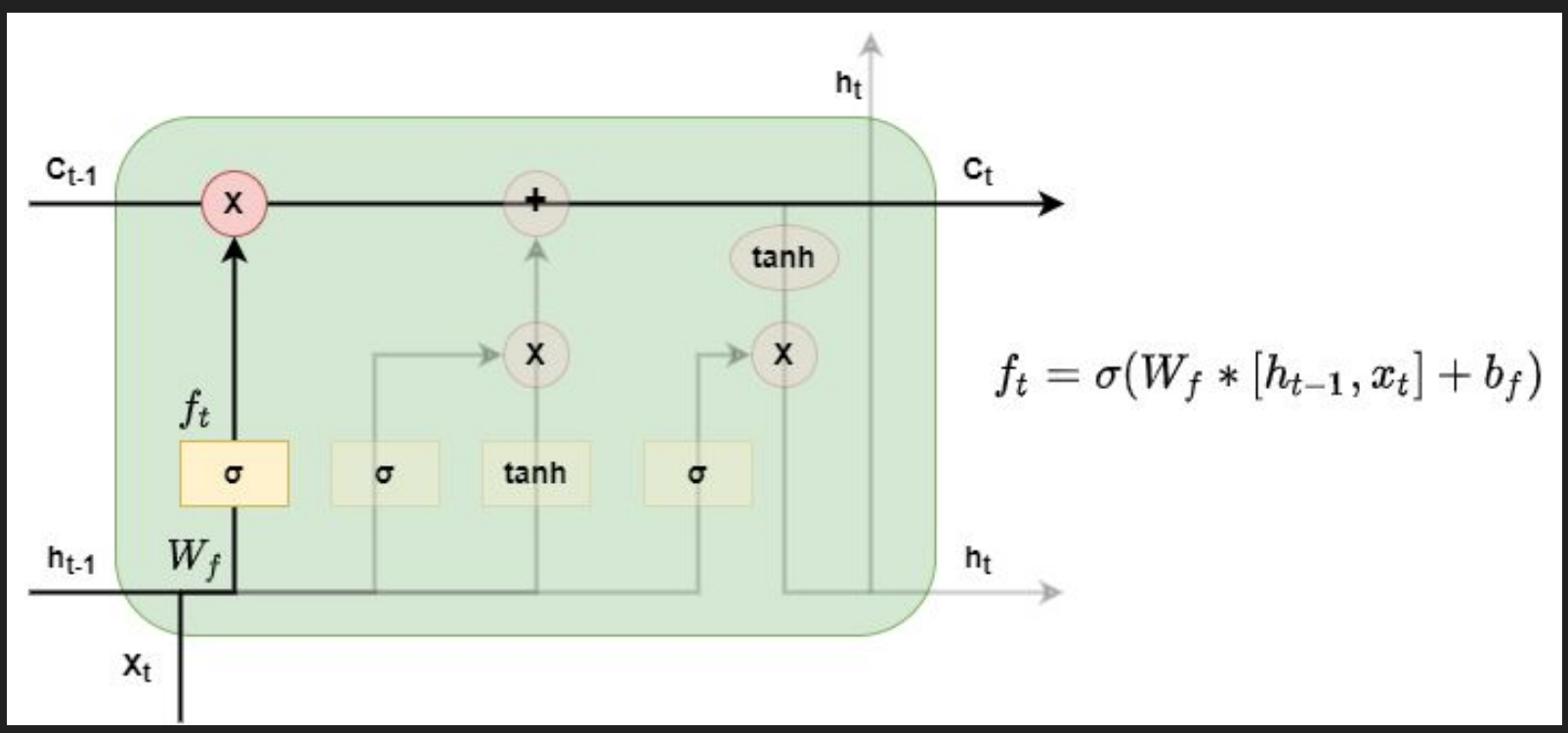




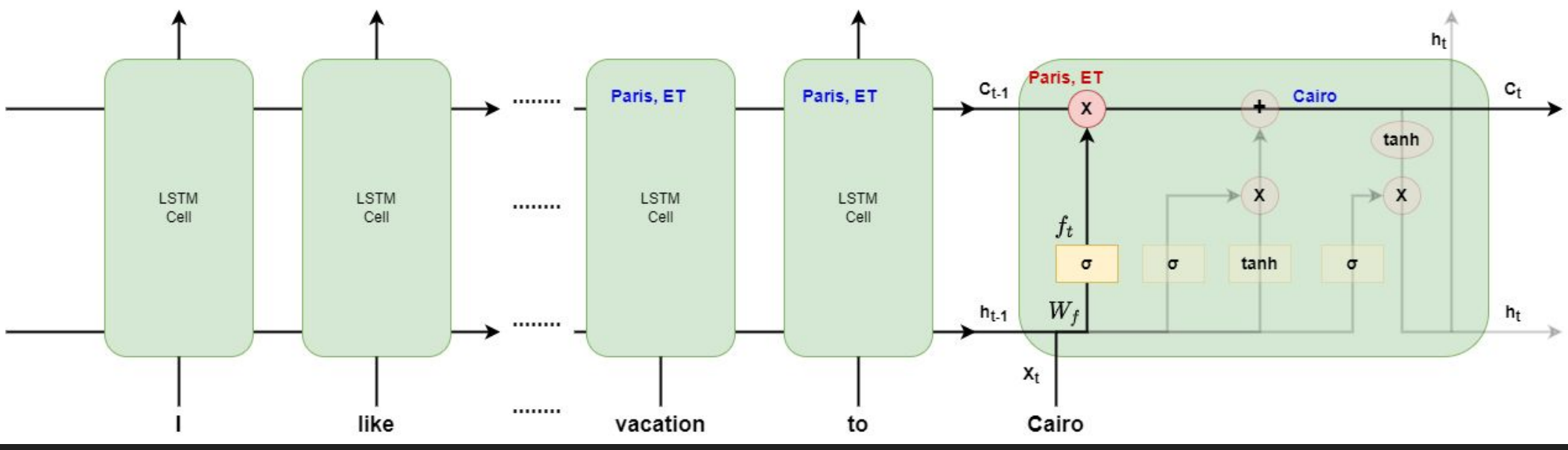
$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$$

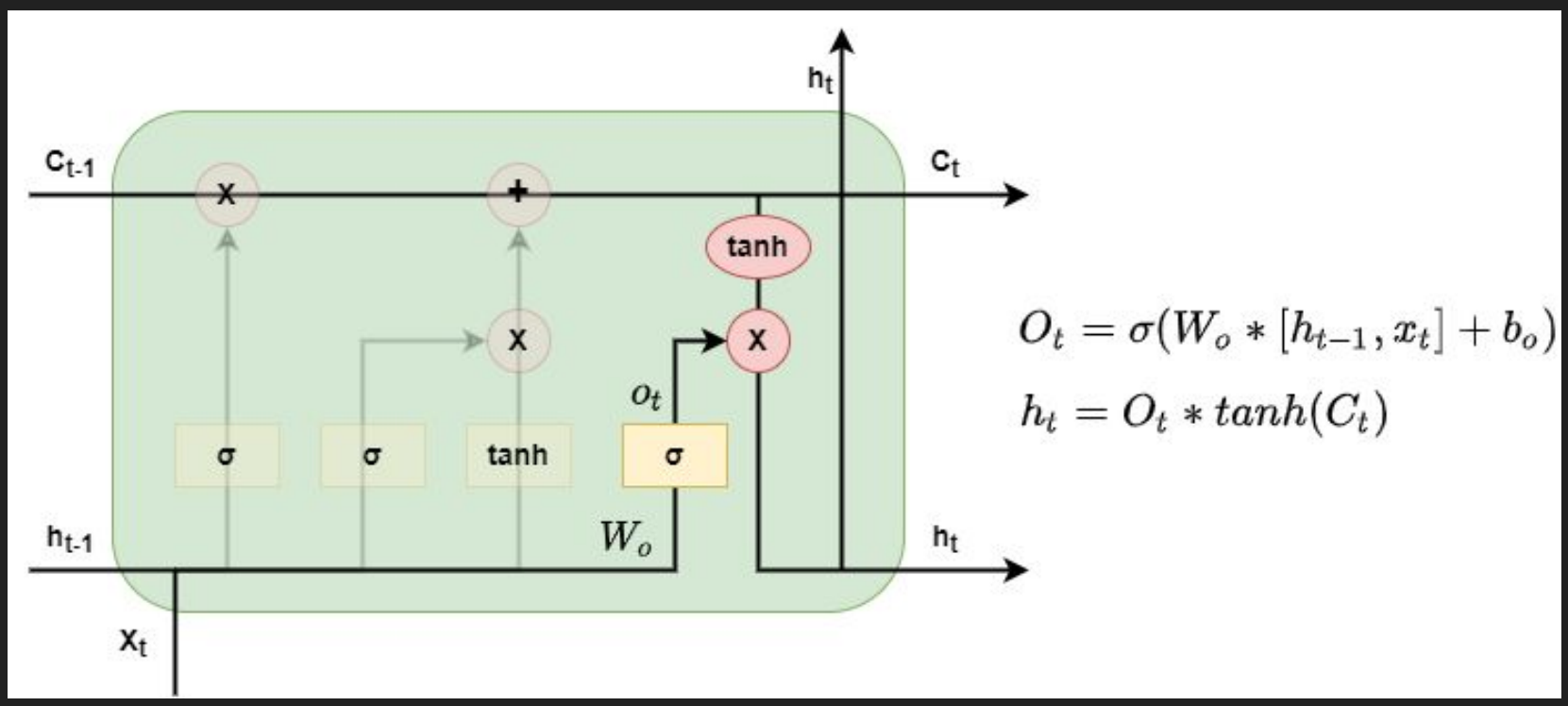
$$\tilde{C}_t = tanh(W_C * [h_{t-1}, x_t] + b_c)$$


LSTM Cell
LSTM Cell
LSTM Cell
LSTM Cell
I
like
to
Paris
Paris, ET
C_{t-1}
C_t
h_t
h_{t-1}
x_t
i_t
$\tilde{C}_t$
W_i
W_c
σ
σ
tanh
σ
tanh
X
X
X
+



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


LSTM Cell
LSTM Cell
.....
.....
LSTM Cell
LSTM Cell
Paris, ET
Paris, ET
C_{t-1}
Paris, ET
X
+
Cairo
C_t
h_t
tanh
X
X
f_t
σ
σ
tanh
σ
h_{t-1}
W_f
h_t
x_t
I
like
vacation
to
Cairo



$$O_t = \sigma(W_o * [h_{t-1}, x_t] + b_o)$$

$$h_t = O_t * tanh(C_t)$$

LSTM(Code)

$$C_t = (i_t * \tilde{C_t}) + (f_t * C_{t-1})$$

$$\frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial C_t}{\partial f_t}\frac{\partial f_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial i_t}\frac{\partial i_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial \tilde{C}_t}\frac{\partial C_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial C_{t-1}}$$

$$C_t = (i_t * \tilde{C_t}) + (f_t * C_{t-1})$$

$$\frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial C_t}{\partial f_t}\frac{\partial f_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial i_t}\frac{\partial i_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial \tilde{C}_t}\frac{\partial C_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial C_{t-1}}$$

What is learnt

You can calculate the **ROUGE-1** score like this:

$$\text{ROUGE-1}_{\text{recall}} = \frac{\text{unigram cand.} \cap \text{unigram ref.}}{|\text{unigram ref.}|}$$

$$\text{ROUGE-1}_{\text{precision}} = \frac{\text{unigram cand.} \cap \text{unigram ref.}}{|\text{unigram cand.}|}$$

$$\text{ROUGE-1}_{\text{F1}} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}$$

$$\text{ROUGE-L}_{\text{recall}} = \frac{\text{LCS(cand., ref.)}}{\#\text{words in ref.}}$$

$$\text{ROUGE-L}_{\text{precision}} = \frac{\text{LCS(cand., ref.)}}{\#\text{ words in cand.}}$$

$$\text{ROUGE-L}_{\text{F1}} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}$$

	Candidate	Reference
Text	I really loved reading the Hunger Games.	I loved reading the Hunger Games.
Unigrams	[('I',), ('really',),('loved',), ('reading',), ('the',), ('Hunger',), ('Games.',)]	[('I',), ('loved',), ('reading',), ('the',), ('Hunger',), ('Games.',)]
Bigrams	[('I', 'really'), ('really', 'loved'), ('loved', 'reading'), ('reading', 'the'), ('the', 'Hunger'), ('Hunger', 'Games.')]	[('I', 'loved'), ('loved', 'reading'), ('reading', 'the'), ('the', 'Hunger'), ('Hunger', 'Games.')]
LCS	I loved reading the Hunger Games.	

The ROUGE-1 calculation is performed as

$$\text{ROUGE-1}_{\text{recall}} = \frac{\text{unigram cand.} \cap \text{unigram ref.}}{|\text{unigram ref.}|} = \frac{6}{6} = 1$$

$$\text{ROUGE-1}_{\text{precision}} = \frac{\text{unigram cand.} \cap \text{unigram ref.}}{|\text{unigram cand.}|} = \frac{6}{7}$$

$$\text{ROUGE-1}_{\text{F1}} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} = \frac{12}{13} \approx 0.923$$

Similarly, the ROUGE-L is calculated as

$$\text{ROUGE-L}_{\text{recall}} = \frac{\text{LCS(cand., ref.)}}{\#\text{words in ref.}} = \frac{6}{6} = 1$$

$$\text{ROUGE-L}_{\text{precision}} = \frac{\text{LCS(cand., ref.)}}{\#\text{ words in cand.}} = \frac{6}{7}$$

$$\text{ROUGE-L}_{\text{F1}} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} = \frac{12}{13} \approx 0.923$$