NLP in Deep learning

Text data → Vector → Numerical Representation

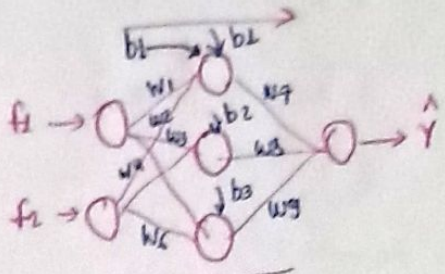
1. BHE
2. Bow
3. Tf-IDf
4. Word2Vec, Avg word2Vec & Sentiment Analysis, Text Classification

- ANN → Artificial Neural N/w → Classification / Regression } - Tabular Data

Eg :- House price Prediction

	f_1	f_2	Y
	House Size	No. of Rooms	Price
→	–	–	–
→	–	–	

f_2 f_1 Y



- Sequence of Data Don't Matter

- CNN (Conv/Convolutional Neural N/w) → Images, Video Frames (Data)

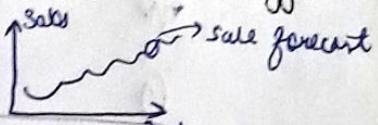
Eg → Image Classification, Object Detection

- Data → Sequential Data.

1. Text Generation → This is a Apple ---- (I/p, Juice)
2. Chatbot Conversation → Q & A (Sentence; Question → Ans)

Sequential Data → The food is good & good the food is ✓ → Meaning may change

3. Language Translation → English, French
4. Auto Suggestion → Linkedin, Gmail → Auto suggestion
5. Sales Data → Date Time



Can we solve this using ANN? ? → Seq Data

NLP in Deep L [Gen AI → LLM, Multimodal]

1. Simple RNN → LSTM / GRU RNN → Bidirectional RNN → Encoder Decoder
2. Transformer ← Self Attention ←

Dataset (Sentiment Analysis)

Text	O/p
The food is good	1
The food is bad	0
The food is not good	0

ANN

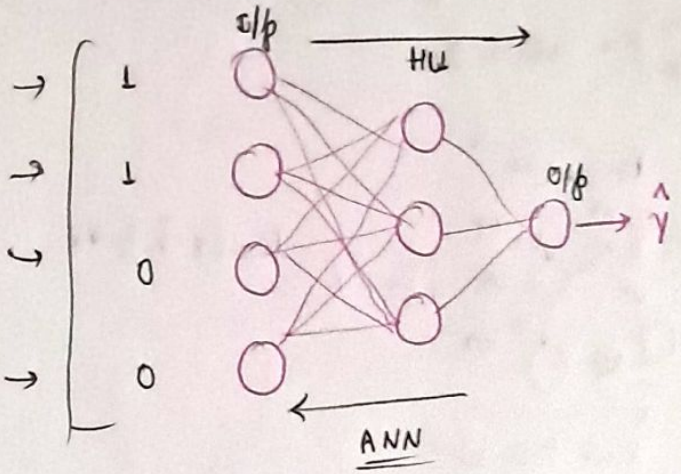
Text preprocessing → Text → Vector

Vocabulary Size → 4

BOW, TF-IDF

Text Data → Sequence Information is Imp
Meaning of the Sentence is less.

	food	good	bad	not
S1	1	1	0	0
S2	1	0	1	0
S3	1	1	0	1



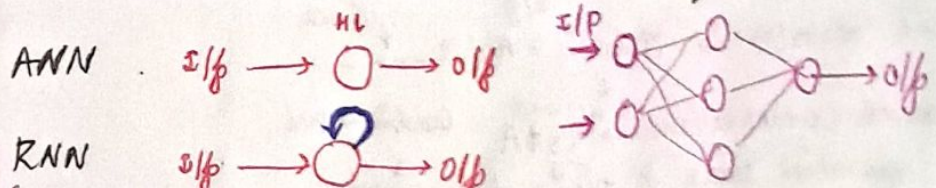
Text Data

Sequence of Information
The food is good
t=1 t=2 t=3 t=4

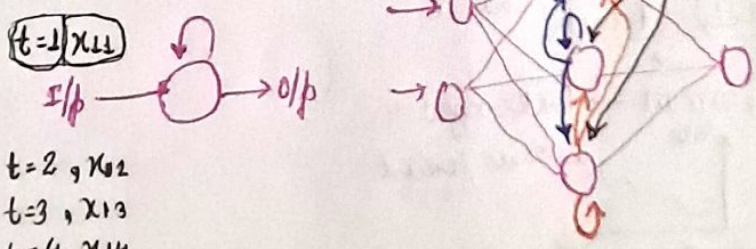
Google Translate → English → French

The - - - ↓ - - -

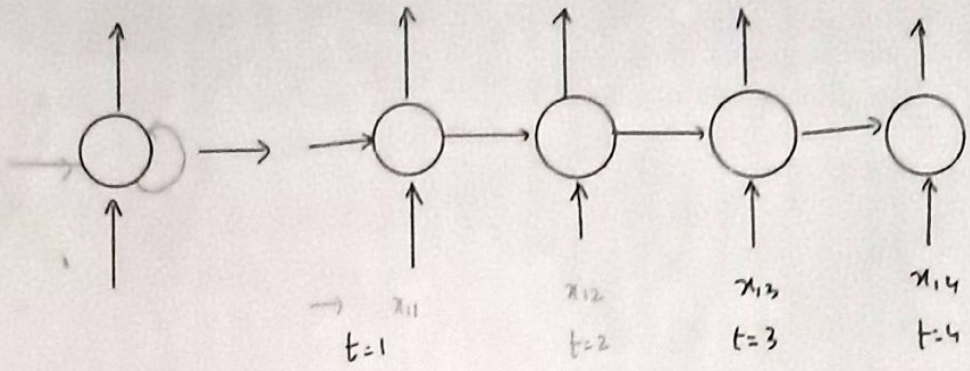
Single RNN → Recurrent Neural N/W → Data → Sequential



x_{11} x_{12} x_{13} x_{14} O/p
The food is good 1



t=2, x_{12}
t=3, x_{13}
t=4, x_{14}



Working Of Simple RNN With Forward Propogation

Dataset

Text

The food is good 1

The food is bad 0

The food is not good 0

Total unique word = 5

Vector of Sentence 1

O/P

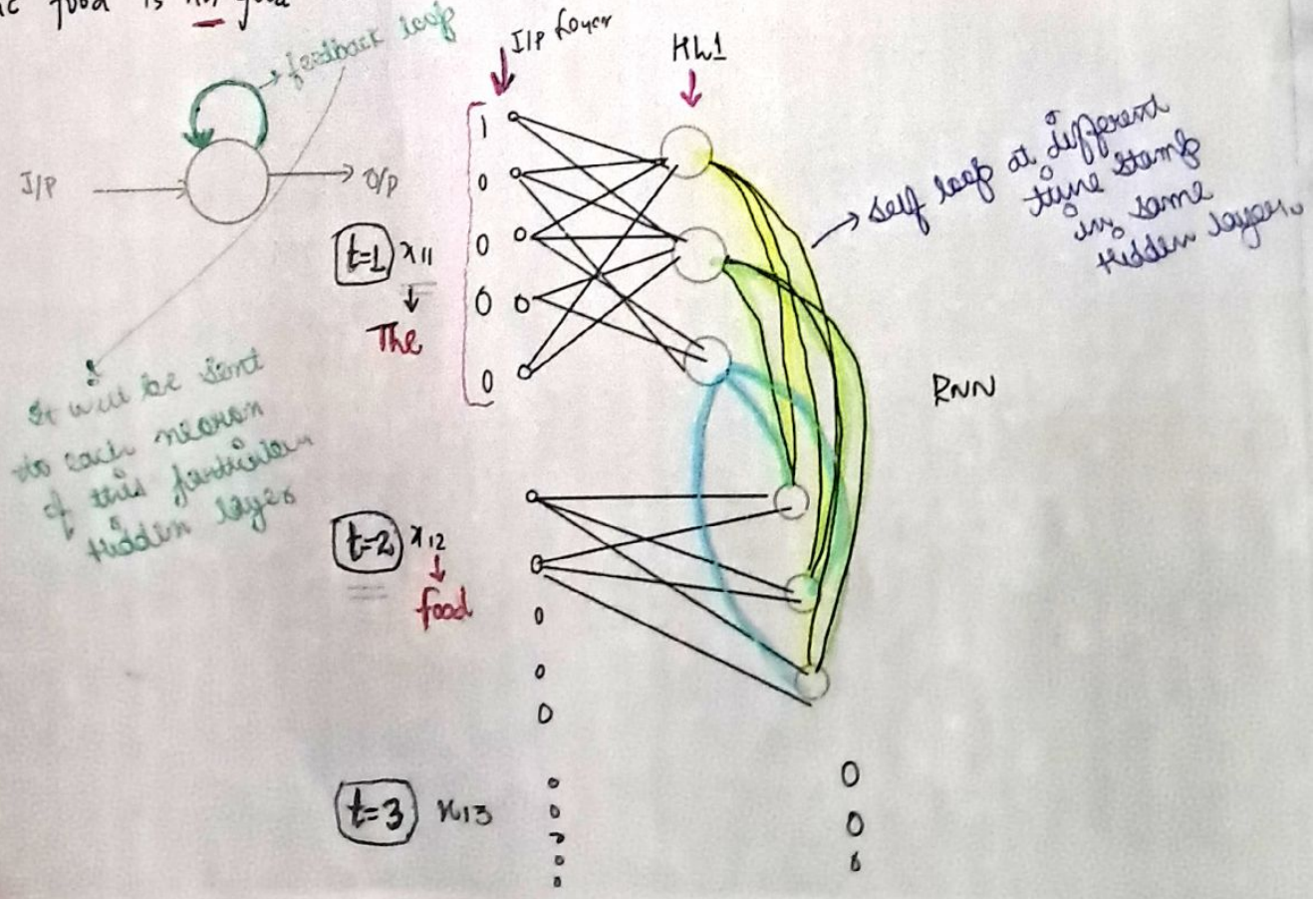
x_{11} x_{12} x_{13}

Words → Vectors

OHE → One hot Encoding

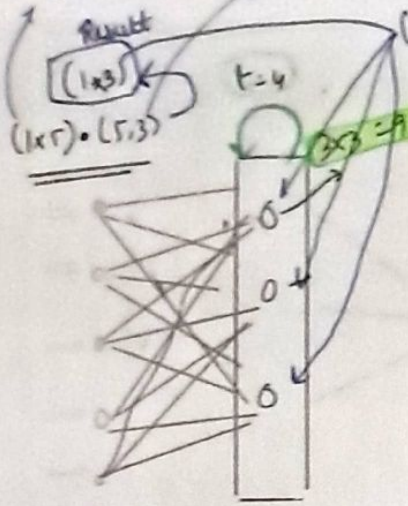
[[1 0 0 0 0] [0 1 0 0 0] [0 0 1 0 0]

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



1 row with 5 input → each input with 3 hidden layer

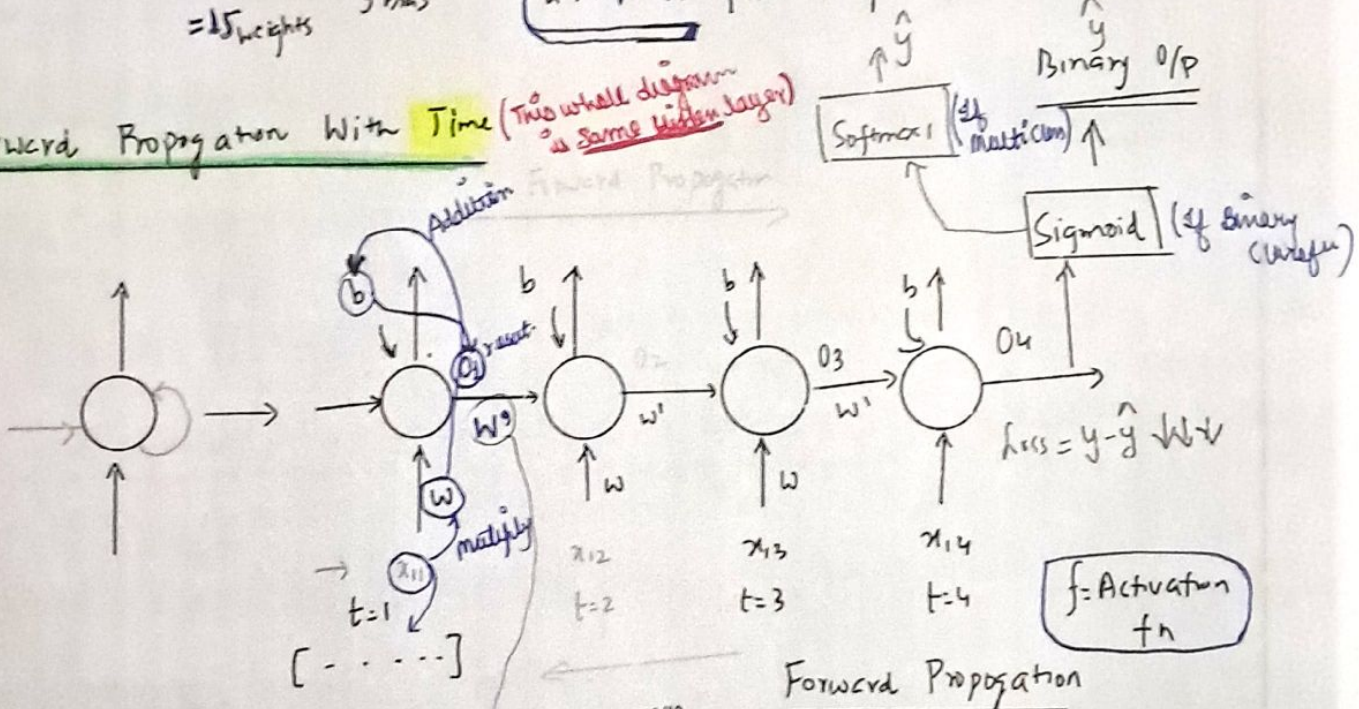
(for each neuron we get (1×3) so total (3×3) for whole hidden layer.



bias

(27+4 = 31 {Trainable param}.

Forward Propagation With Time (This whole diagram is same hidden layer)



Dataset

The food is good

x_{11} x_{12} x_{13} x_{14}

O/P

1

Forward Propagation

$O_1 = f(x_{11} \cdot w + b_1)$

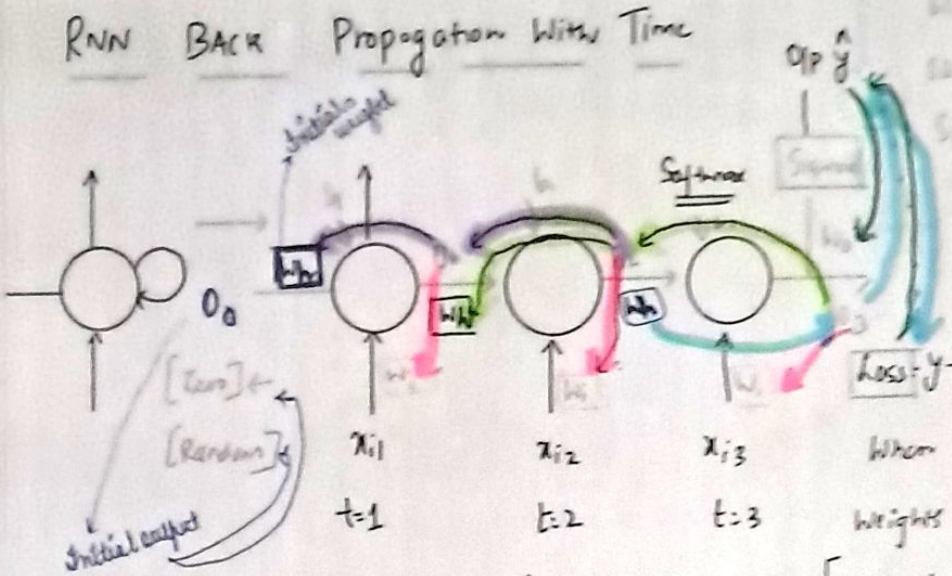
$O_2 = f(x_{12} * w + O_1 * w' + b_1)$

$O_3 = f(x_{13} * w + O_2 * w + b_1)$.

↓ Any activation fun

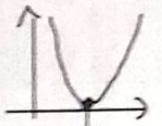
This step happen bcoz of w'' we got from feedback loop

RNN BACK Propogation With Time



$\langle x_{i1}\ x_{i2}\ x_{i3} \rangle$
i = sentence No

When we update all the weights $[w_i, w_h, w_o]$ ← Back Propogation



Forward Propogation

$$O_1 = f(x_{i1} * w_i + O_0 w_h + b_1)$$

$$O_2 = f(x_{i2} * w_i + O_1 * w_h + b)$$

$$O_3 = f(x_{i3} * w_i + O_2 * w_h + b)$$

$$\hat{y} = \sigma(O_3 * w_o)$$

Backward Propogation with Time

↳ Update $[w_i, w_h, w_o]$

Weight updation formula

$$w_{new} = w_{old} - \eta \frac{\partial h}{\partial w_{old}}$$

$\frac{\partial h}{\partial w_{old}}$ → Derivative, slope of GRADIENT DESCENT → to Calculate

① Update w_o

Chain Rule of Derivative

$$\frac{\partial h}{\partial w_{old}} = \frac{\partial h}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_{old}}$$

Solve ONLY w_o update eqn.

$$w_{o_{new}} = w_{o_{old}} - \eta \frac{\partial h}{\partial w_{old}}$$

Because Backpropogation

② Update w_h [Hidden layer weights] → Time Stamps

$$w_{h_{new}} = w_{h_{old}} - \eta \frac{\partial h}{\partial w_{old}}$$

t=1, 2, 3

$$\Rightarrow \frac{\partial L}{\partial w_{h_{old}}} = \left[\frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial o_3} * \frac{\partial o_3}{\partial w_h}\right]_{t=3} + \left[\frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial o_3} * \frac{\partial o_3}{\partial o_2} * \frac{\partial o_2}{\partial w_h}\right]_{t=2}$$

↳ In one Back Propagation all hidden weight(wh) Updated

$$+ \left[\frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial o_3} * \frac{\partial o_3}{\partial o_2} * \frac{\partial o_2}{\partial o_1} * \frac{\partial o_1}{\partial w_h}\right]_{t=1}$$

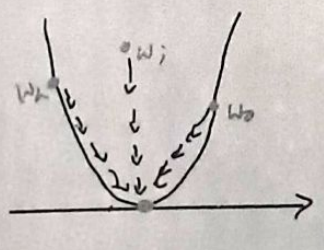
③ Updating Weights W_i → Timestamp

$$W_{inew} = W_{iold} - \eta \boxed{\frac{\partial L}{\partial w_{iold}}}$$

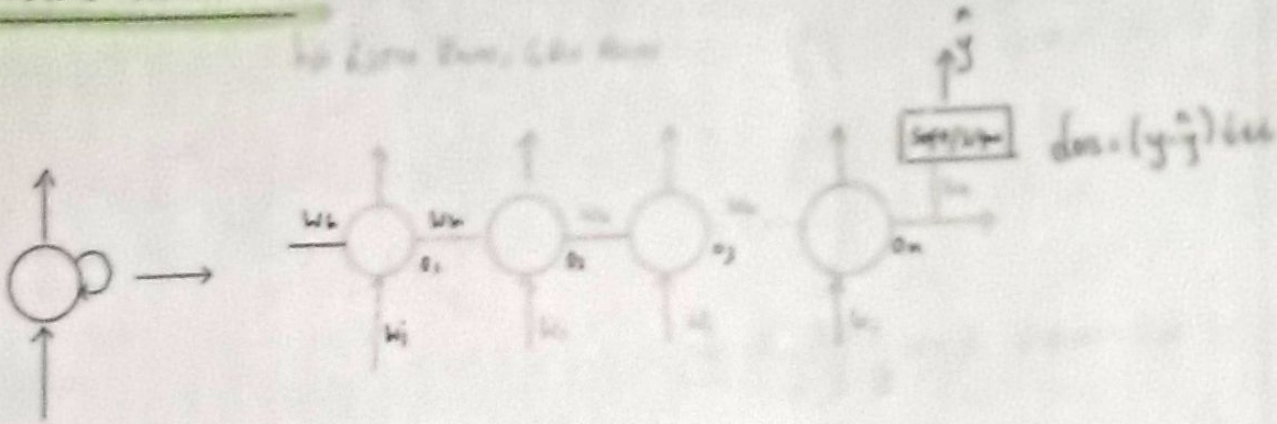
{Update W_i weights}

$$\boxed{\frac{\partial L}{\partial w_{iold}}} = \left[\frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial o_3} * \frac{\partial o_3}{\partial w_{iold}}\right]_{t=3} + \left[\frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial o_3} * \frac{\partial o_3}{\partial o_2} * \frac{\partial o_2}{\partial w_{iold}}\right]_{t=2}$$

$$+ \left[\frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial o_3} * \frac{\partial o_3}{\partial o_2} * \frac{\partial o_2}{\partial o_1} * \frac{\partial o_1}{\partial w_{iold}}\right]_{t=1}$$



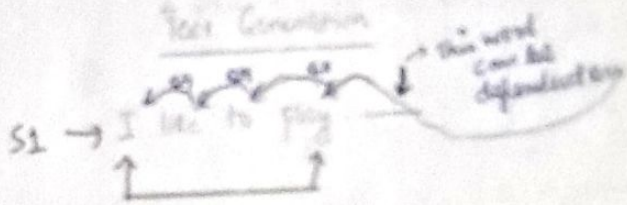
Problems With RNN



ANN → Vanishing Gradient Problem

Text	O/P
The food is good	1
The food is bad	0

Sentences → Small

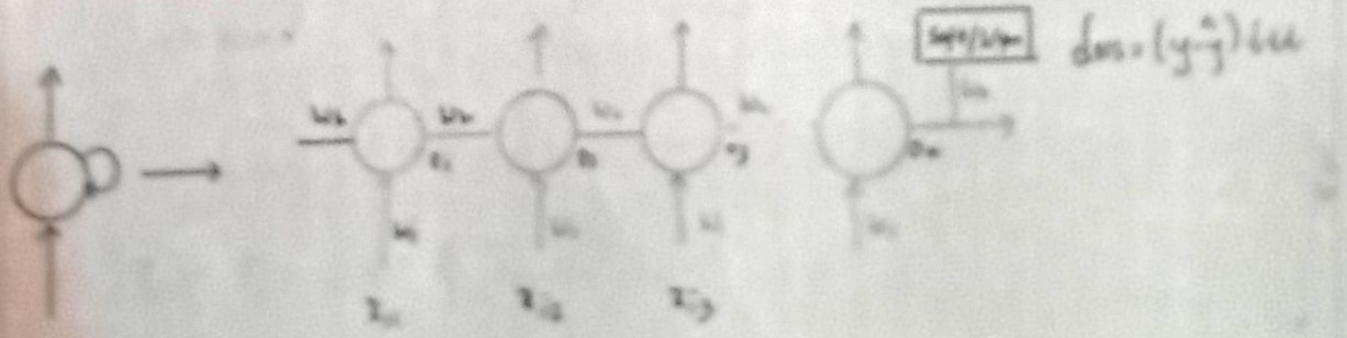


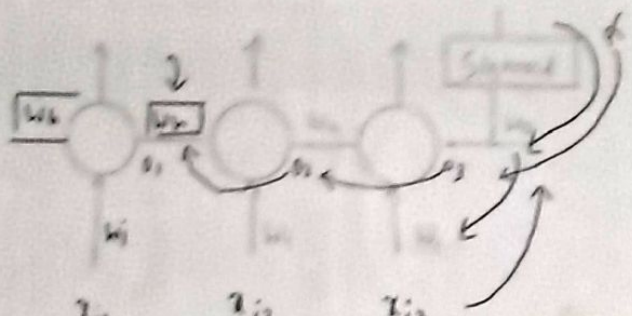
Sentence is long

My Name is KRISH and I like sports like CRICKET, VOLLEYBALL AND also LIKE TO MAKE food Videos

If I have sentence with → 100 words. So at t=100 we will find last word for → to predict

Long term dependency cannot be captured by RNN → Provide Accuracy ↓ Dependency





W_h, W_i, W_o

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W_{old}}$$

$$\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial W_o} \Leftarrow \quad t=2$$

for example if we want it for $t=1$

$$\frac{\partial L}{\partial W_{h_{old}}} = \left[\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_3} \cdot \frac{\partial O_3}{\partial O_2} \cdot \frac{\partial O_2}{\partial W_{h_{old}}} \right] + \cdots$$

↳ for word = only 3

- length of sentence ⇒ 50 words

$t=1$ $\qquad$ $t=2$

$$\frac{\partial L}{\partial W_{h_{old}}} = \left[\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_{50}} \cdot \frac{\partial O_{50}}{\partial O_{49}} \underbrace{\left| \frac{\partial O_{49}}{\partial O_{48}} \right| \cdots}_{0.25 \times 0.02 \times 0.01 \times 0.015} \frac{\partial O_2}{\partial W_{h_{old}}} \right] + \cdots$$

Small Value ≈ 0

$$\frac{\partial O_3}{\partial O_2} = \frac{\partial\, \sigma\left(x_{i3} * W_i + O_2 * W_h + b\right)}{\partial O_2} \Leftarrow O_3.$$

$O_3 = \sigma(x_{i3} * W_i + O_2 * W_h + b)$

(Basic differentiation w.r.t ∂O_2)

constant w.r.t O_2 so zero

$$= \sigma'(1 * W_h) \Rightarrow [0 - 0.25]_4(W_h)$$

Derivative of sigmoid $0 - 0.25$

$t=1$ ⇒ The word is not participating to update the weights value. (Basic weights will change with steps)

↳ Chain Rule is Big

Conclusion

If t=50 wala weight ka gradient update hoga
But t=1 wala weight update nahi hoga
So, Output will more depend on
t=50 wala word to predict
Depending on distance word

t=50

$$\frac{\partial L}{\partial w_{hh}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial o_{50}} + \frac{\partial o_{50}}{\partial w_{hh}}$$

$\neq 0$ $\neq 0$

t=2 ≈ 0 + t-1 ≈ 0

To Solve this

① Relu, Leaky Relu →

② LSTM RNN → long short term Memory RNN } ⇒ Simple RNN

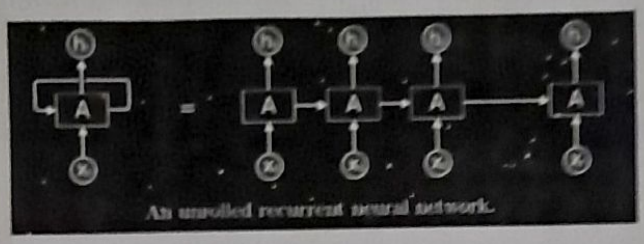
③ GRU RNN →

LSTM RNN [Long Short Term Memory RNN]

RNN ⟶ Long Term Dependencies ⟶ Vanishing Gradient Problem

1. RNN ⟶ Problem ? ✓
2. Why LSTM RNN ? ✓ Basic Representation.
3. How LSTM RNN Works
   - Long Term Memory ✓
   - Short Term Memory ✓
4. LSTM Architecture
5. Working of LSTM RNN

Problems With RNN → Long Term Dependency



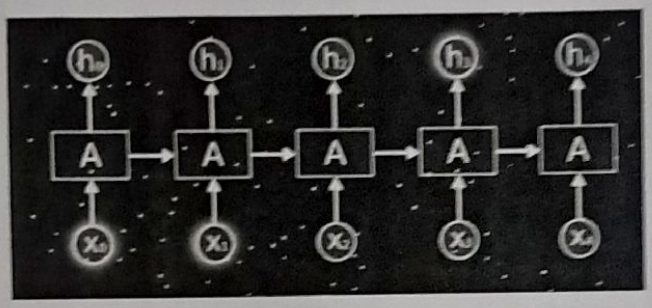
An unrolled recurrent neural network.

Vanishing Gradient Problem

Short term Dependency

Task:
Next word in a Sentence

Working



The color of the sky is blue

further context

Gap is less

Huge gap O/p ← Context

done after Sentence

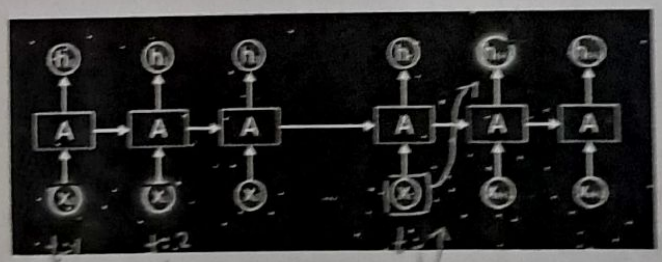
I grew up in India ..... I speak

fluent ____ Context

Name of language

further context → Dependent on India also

Long term Dependency



t=1 t=2 t=3

Huge Gap → Long Term Dependency

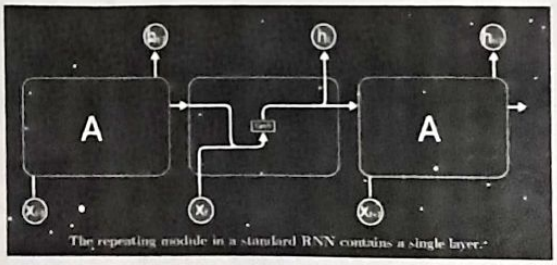
[0-0.25]
[0-1]

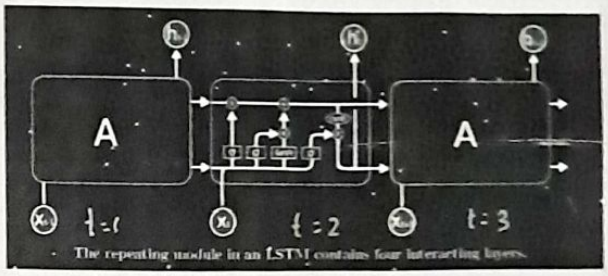
RNN → Long Term Dependency → Vanishing Gradient Problem

Chain Rule → ≈0

Basic Representation Of RNN And LSTM RNN

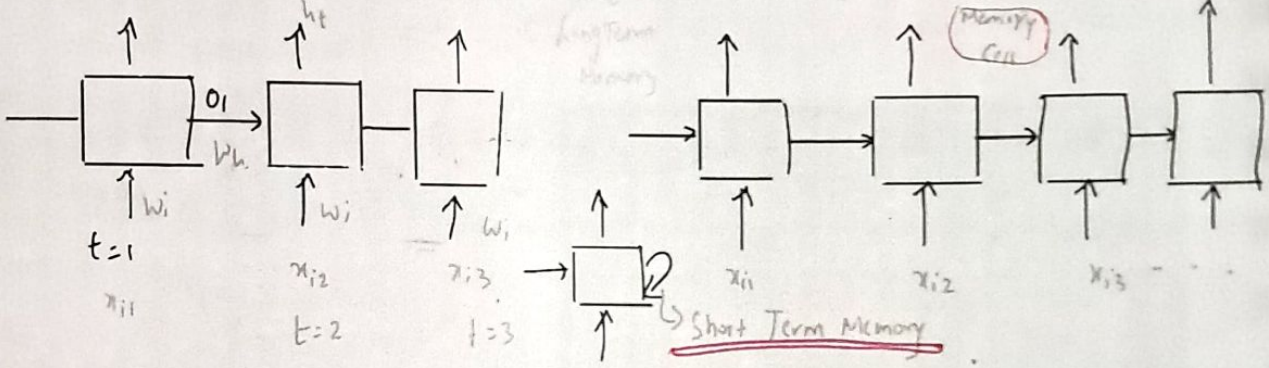
LSTM RNN





t=1 t=2 t=3

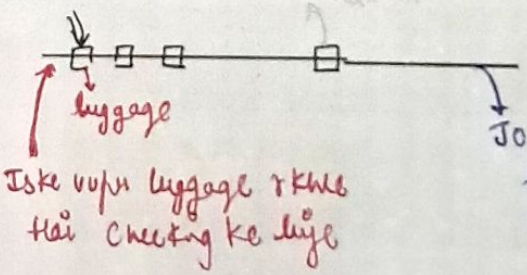
SAME

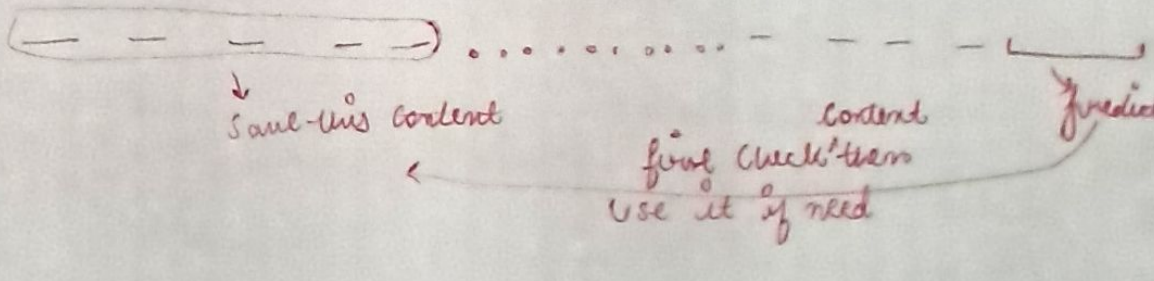


LSTM RNN → Long Term Memory
→ Short Term Memory

Airport

Conveyance Belt : Luggages





memory cell of previous timestamp

C_{t-1}

memory cell

C_t

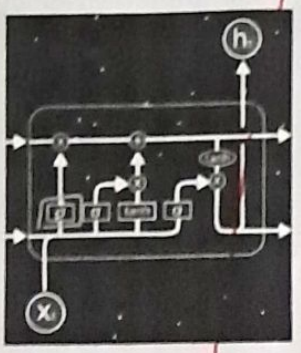
for Long-term Memory

After Removing & Adding Some Content

f_t i_t $\tilde{C}_t$ o_t tanh

h_{t-1} x_t

LSTM Architecture

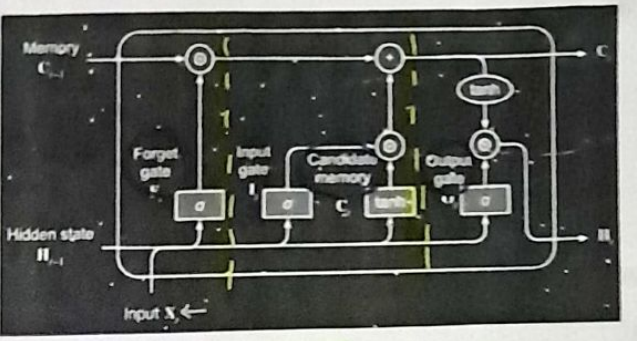


Basic Architecture

LSTM Have Two loop

Long Term

Short Term Memory $[H_{t-1}]$



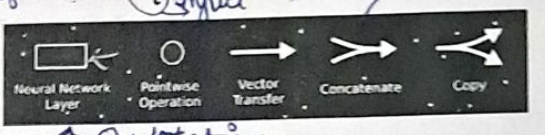
* 3 Gate

{Neural Nw Layer}

h_{t-1} previous Input

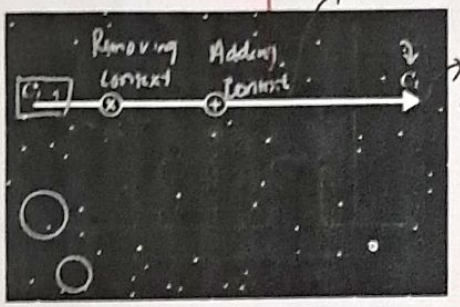
x_t Input

combine two Vector



* Notation

Memory CCII



Long Term Memory

$\times$ $+$ tanh

point wise Operation

V1 $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ ← tanh

V2

$\times \Rightarrow [4 \quad 10 \quad 18]$

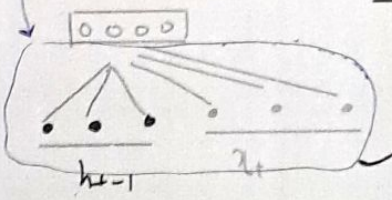
$+ \Rightarrow [5 \quad 7 \quad 9]$

tanh $\Rightarrow [\tanh(1) \quad \tanh(2) \quad \tanh(3)]$

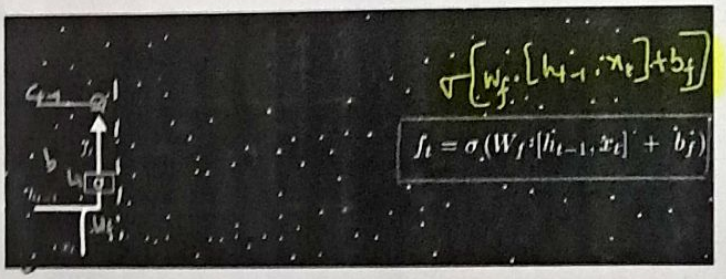
Combining 2 vectors

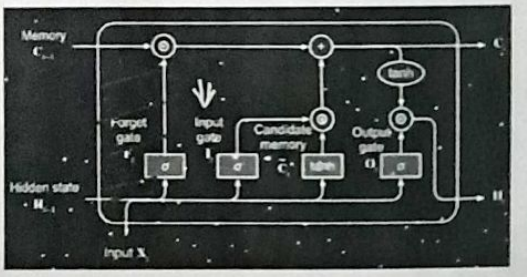
$h_{t-1} = [1 \quad 2 \quad 3]$

$x_t = [2 \quad 3 \quad 4]$



Duplicate Copy





h_{t-1} = Hidden State of previous time stamp

x_t = Word passed as i/p in the current time stamp

Forget Gate

Text — Next Word

x_0 x_1 x_2 x_3 x_4 — y_{15}

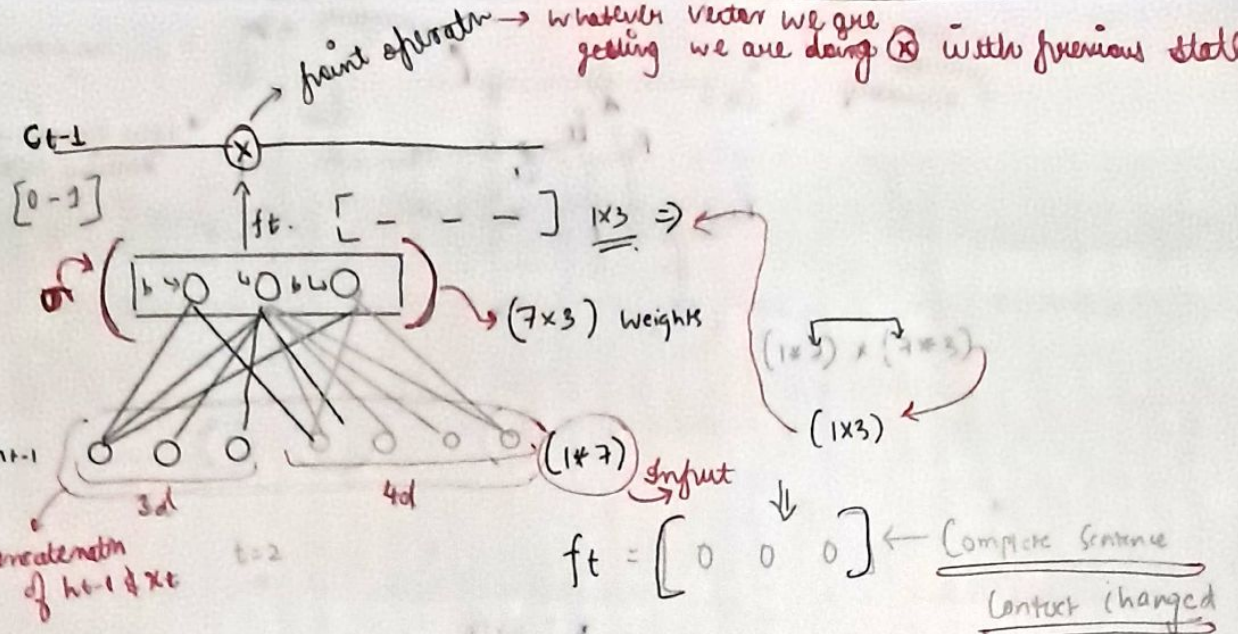
Memory CCU

$[0 \quad 2 \quad 4 \quad 1]$ $[4 \quad 5 \quad 12]$ - - - -

x_t x_{t+1}

h_{t-1} $[1 \quad 2 \quad 4]$ 3d

C_{t-1} $[4 \quad 2 \quad 1]$ 3d



$f_t = [0 \ 0 \ 0]$ ← Complete sentence

Context changed

① $C_{t-1} = [6 \ 8 \ 9] \otimes [0 \ 0 \ 0]$

previous state

$= [0 \ 0 \ 0]$ ← Removing all the previous context

$f_t = [1 \ 1 \ 1]$

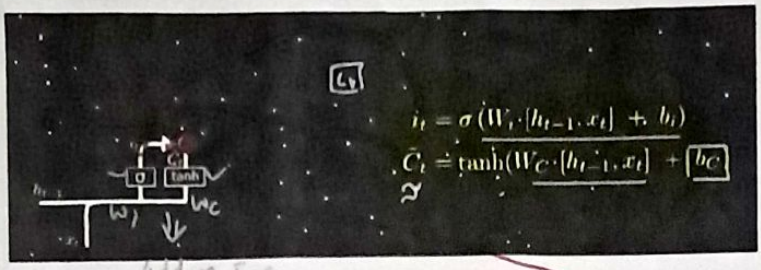
Will not Remove Anything BCOZ context is same

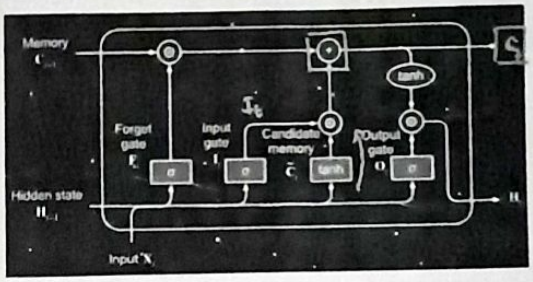
② $C_{t-1} = [6 \ 8 \ 9] \otimes [1 \ 1 \ 1] = [6 \ 8 \ 9]$

③ $C_{t-1} = [\underline{6} \ 8 \ \underline{9}] \otimes [0.5, 1, 0.5] = [3 \ 8 \ 4.5]$

Conclusion : Based on the context → Forget gate will let go some information or will not let go some Info {Forgetting}.

② Input Gate And Candidate Memory





Adding Info

$I_t = [\ 2\ \ 4\ \ 6\ ]$ — (*) — C_t $[\ 0\ \ 2\ \ 0\ ]$ I/p Gate

New Info to add

$[\ 0\ \ 8\ \ 0\ ]$

New Info added

I_t

b → $[\ 0\ \ 0\ \ 0\ ]$

W_i

h_{t-1} ← 0 0 0 0 0 0 0 → x_t

tanh

(1×3)

b →

(7×3)

W_C

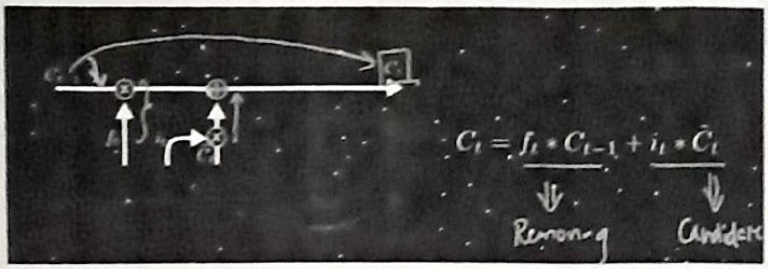
0 0 0 0 0 0 0

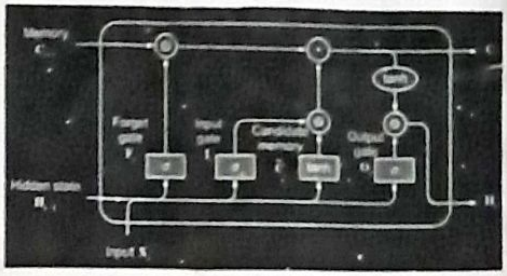
(1×7)

Context = If any information needed to be added it the memory

C_{t-1} → The information will be added

⑪ Combine forget & Input Gate





I stay in India - - - - -

And I speak ——→

English | Hindi

OR

Forgetting Some info

Memory

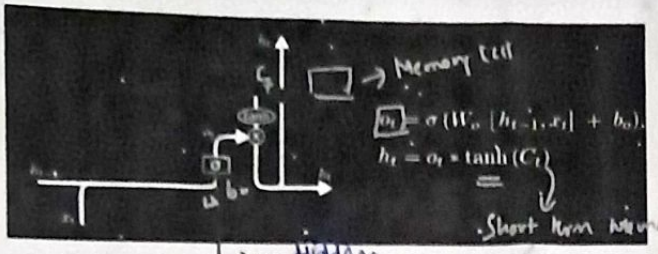
There → Forget Gate

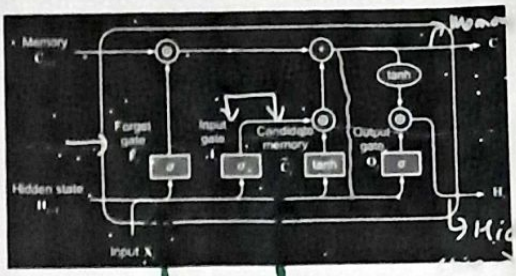
I/p Gate ⊗ Candidate memory

+

C_{t-1} ⇒ C_t

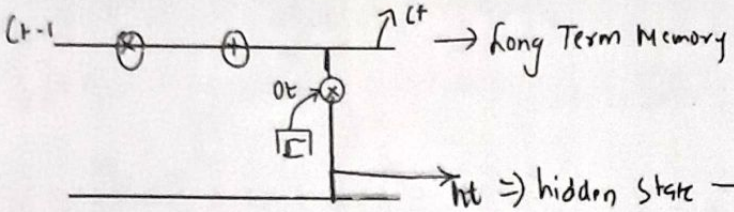
Output gate LSTM RNN





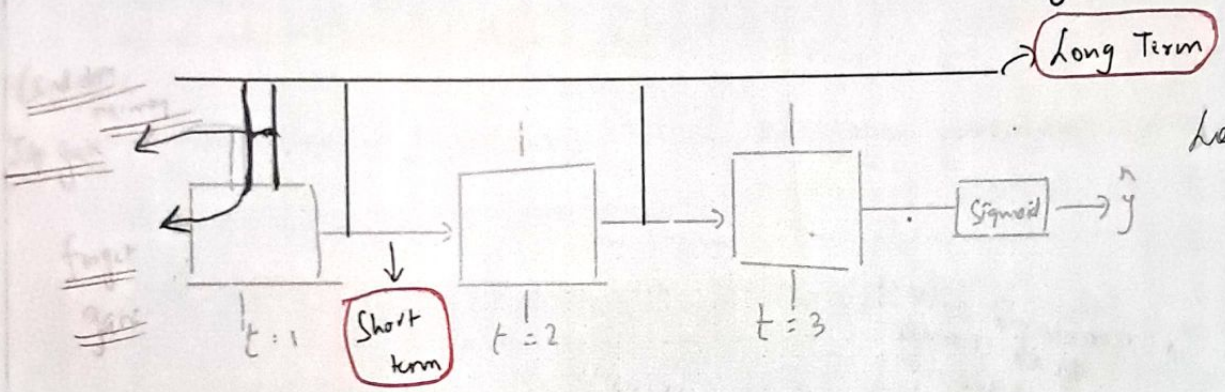
Forget → Some Info → context → Memory Cell

Add Info → context



→ Long Term Memory

Ht ⇒ hidden state → Short term Memory

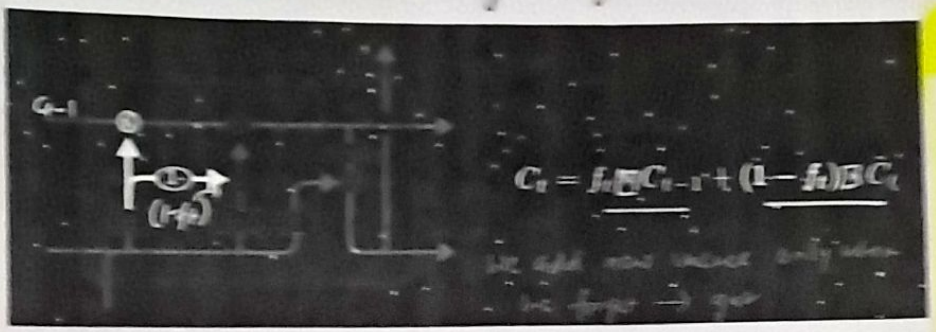


Less Wht

[Wi, Wc, Wo] → Updating ← Back Propogation

GRU RNN ⇒ LSTM Variant

Another variation → Coupling forget and i/p Gates



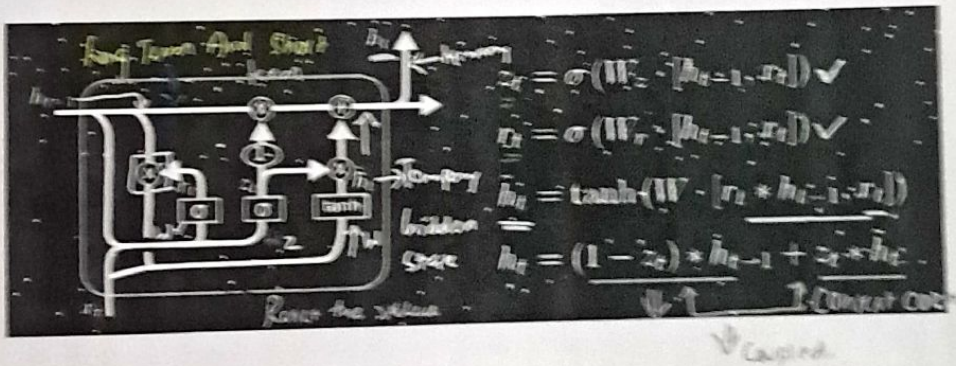
Gru : We only forget when we're going to i/p something in its place. We only i/p new values to the state when we forget something older.

Instead of separately deciding what to forget and what we should add new info, we make these decisions together.

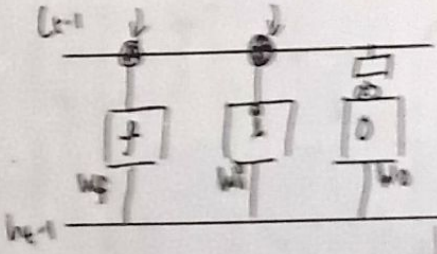
Another Variant

GRU → Gated Recurrent Unit [Cho, et al [2014]]

1990 → LSTM
2000 - Variants
2014 → GRU



z_t ⇒ Update Gate ←
r_t ⇒ Reset Gate ←
$\tilde{h}_t$ ⇒ Temporary hidden state



Problem

LSTM : Complex Architecture

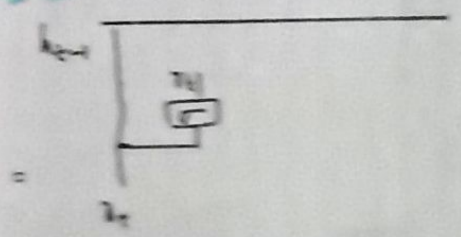
bias
$[W_f, W_i, W_o]$

+ Cand. date memory

O/P

→ Short Term Memory

Training Time ↑↑



Reset gate = r_t =

→ Reading some info from h_{t-1} ⇒ Memory → LSTM & GRU

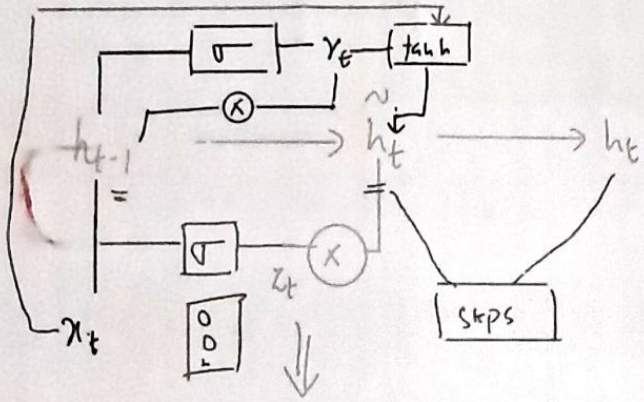
h_{t-1} [0.6 0.5 0.3 0.9]

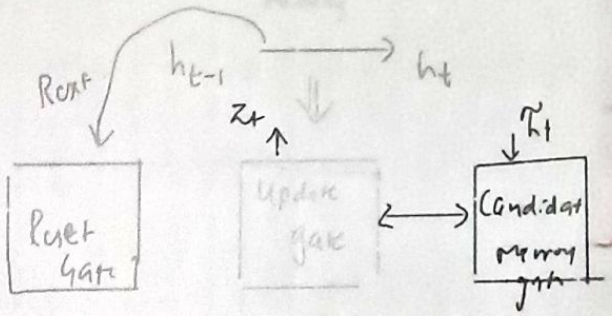
⊗ ←

r_t [

↓ ↓ ↓ ↓

x_t → [0.12 0.20 0.24 0.18] ← Resetting → Context





What context Info needs to be Added

⇓

Candidate hidden state · [current Context]

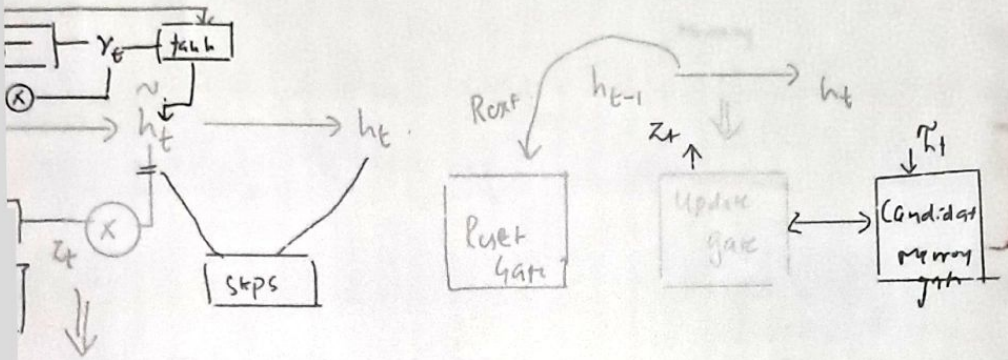
⇓

Imp → Add Info

z_t ⇒ Imp Info

0.5 0.3 0.9]

0.20 0.24 0.18] ← Reacting → Context



What context Info needs to be Added

⇓

Candidate hidden state · [current Context]

⇓

Imp → Add Info

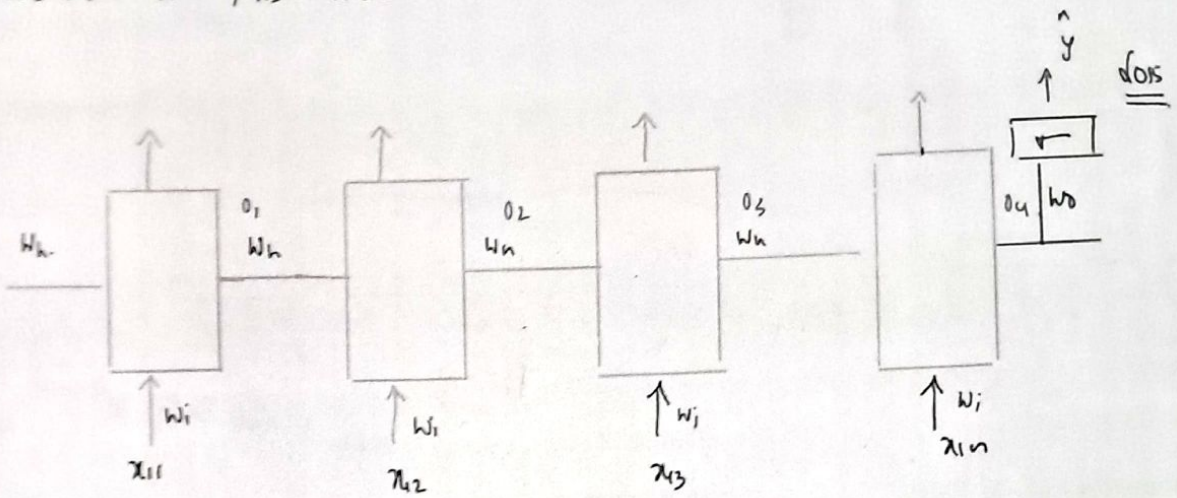
z_t ⇒ Less Info

Bidirectional RNN

① Simple RNN → practical Implement → Embedding layers

② LSTM, GRU variants RNN → Practical Implement

③ Bidirectional RNN / LSTM RNN



t=1

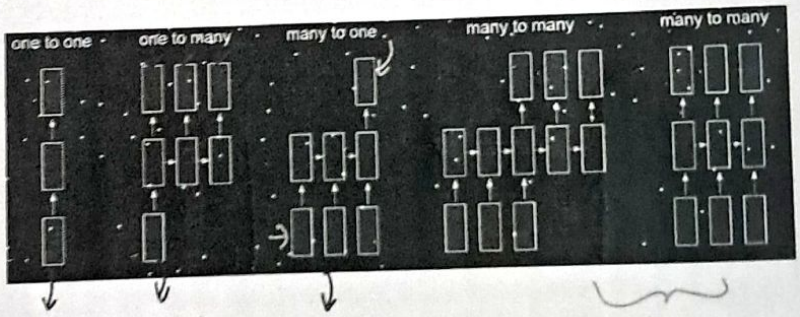
Types of RNN

① One to Many RNN

② Many to One RNN

③ Many to Many RNN

④ One to One RNN



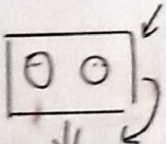
One i/p
One o/p

One i/p
o/p many

Image scanch

I/p many
o/p one

Language Translation
many I/p
many o/p

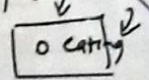


There is a dog and cat

Eg:

Eg: Image Captioning

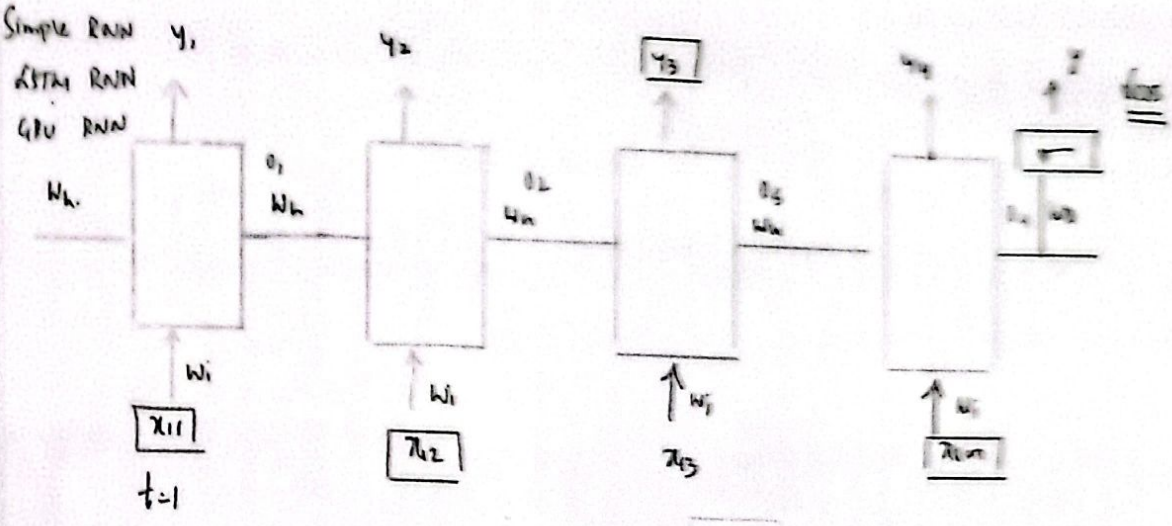
A cat eating food



Text Input on google

O/p > picture

Simple RNN
LSTM RNN
GRU RNN

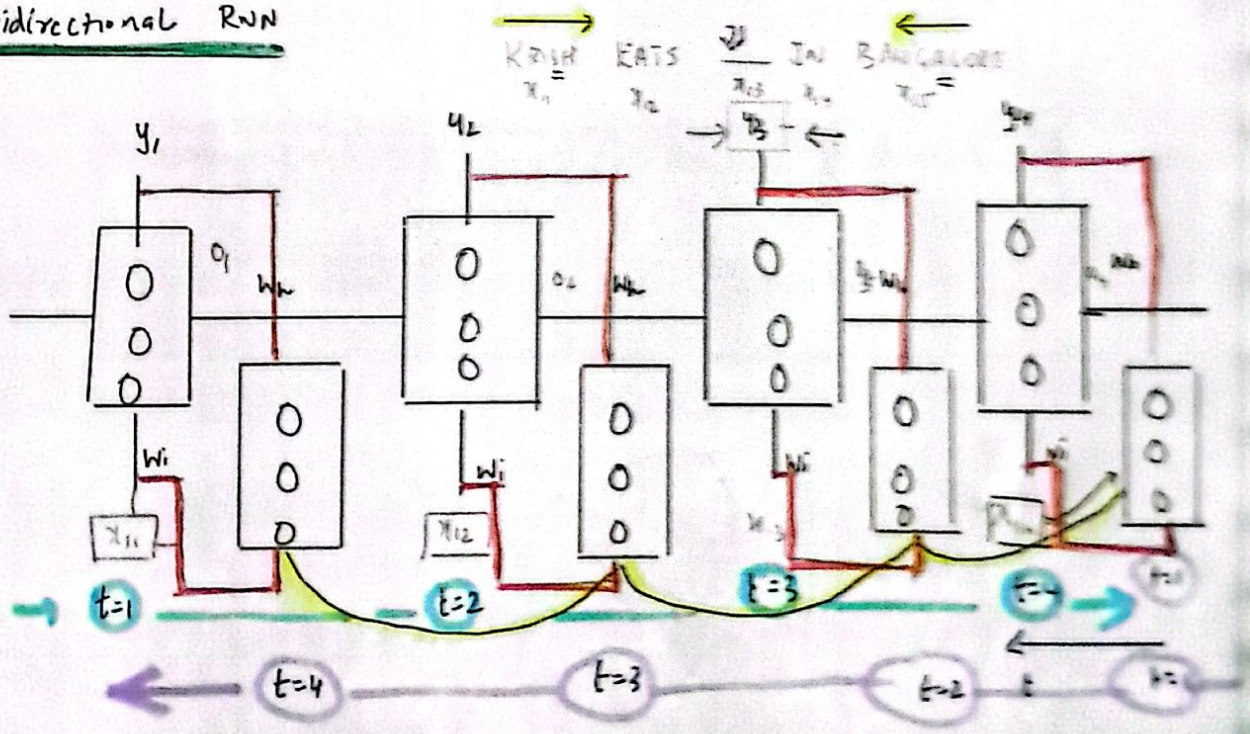


Text Example = KRISH EATS DOSA IN BANGALORE

KRISH EATS PIZZA IN PARIS

Defined one better previous & forward words.

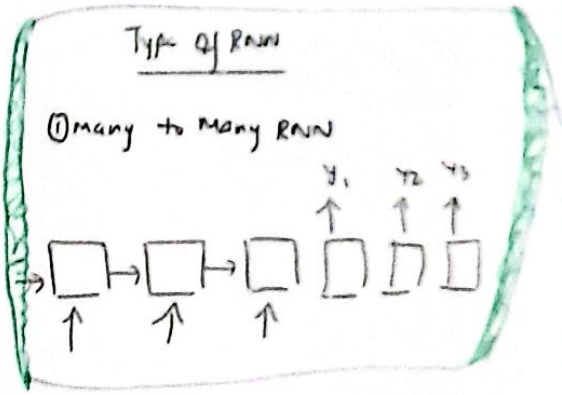
Bidirectional RNN



① Forward propagation → Equation ←

Encoder And Decoder

1. Simple RNN → Vanishing Gradient Problem
2. LSTM RNN → Long Short Term Memory.
3. GRU RNN → Long Short Term Memory.
4. Bidirectional RNN ← If we have Context with further Words



Encoder And Decoder }

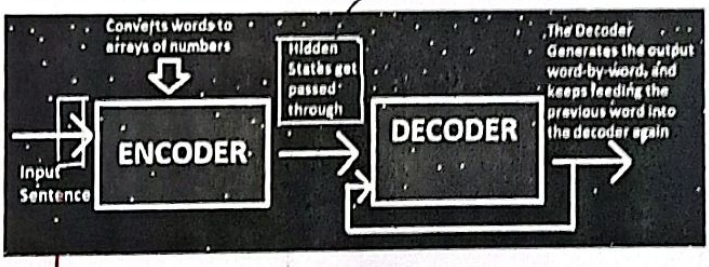
Sequences I/p — O/p Sequence Of Words

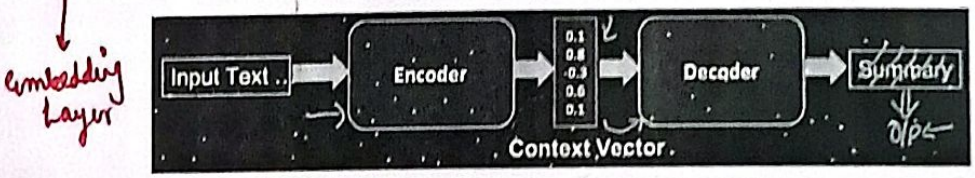
Eg:- One Language To other

English → French

Eg:- Linked that → Hi, How are you?

Simple Working





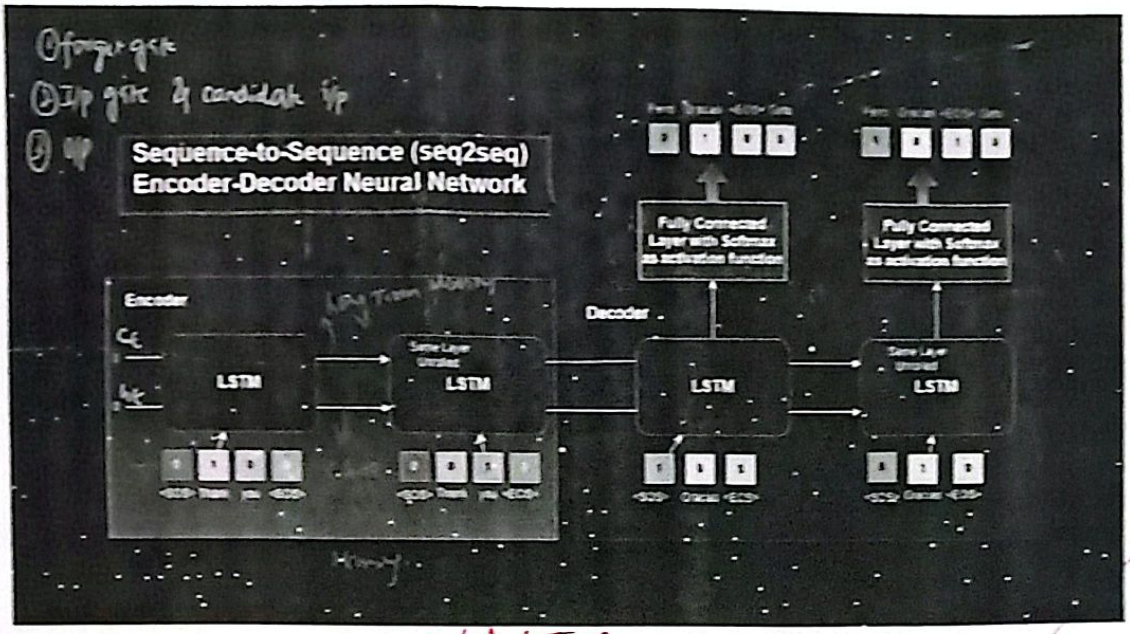
Convert

1. Encoder ⇒ I/p ⇒ Context Vector ← Vectors
2. Decoder ⇒ ⇒ O/p

Usecase

1. Language Translation
2. Text Generation
3. Text Suggestion

Recurrent vanishing Gradient problem



Dataset

English | French

<SOS> Thank you <EOS> | Gracias → <SOS> GRACIAS

end of sentence

True o/p | y truth

{Training}

Encoder

Context Vectors

Long Term

Short Term

Softmax

Softmax

Decoder

OHE ⇒ <SOS> Thank You <EOS>. Memory

t=1 t=2 t=3 t=4

Context of this sentence.

Loss $(y-\hat{y})^2$

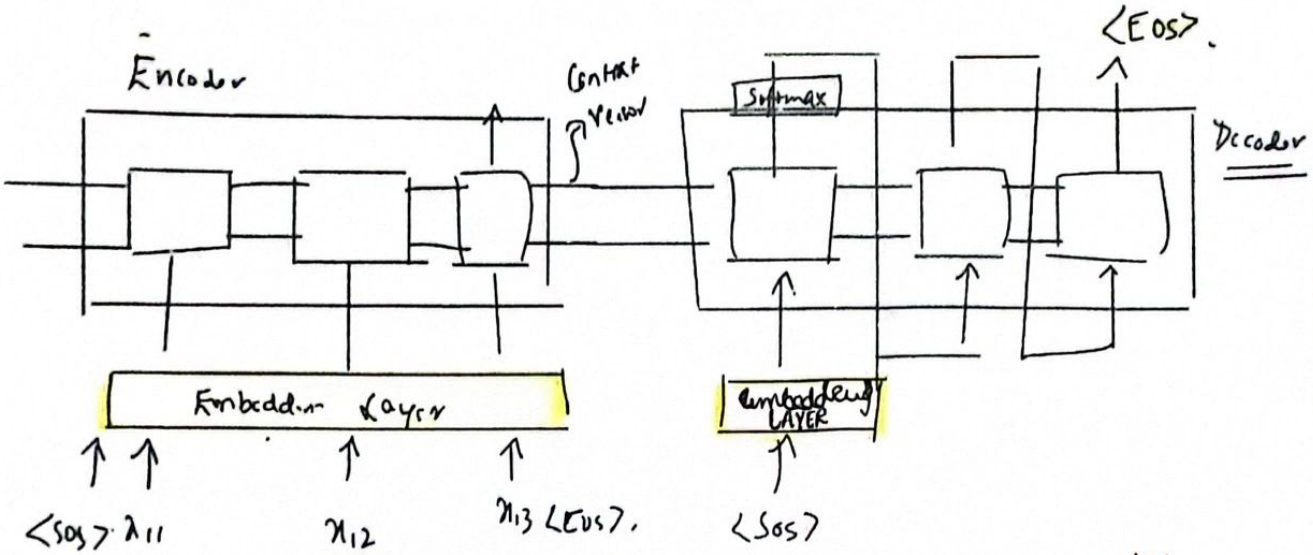
Optimizer

Update weights

<EOS> GRACIAS <EOS>

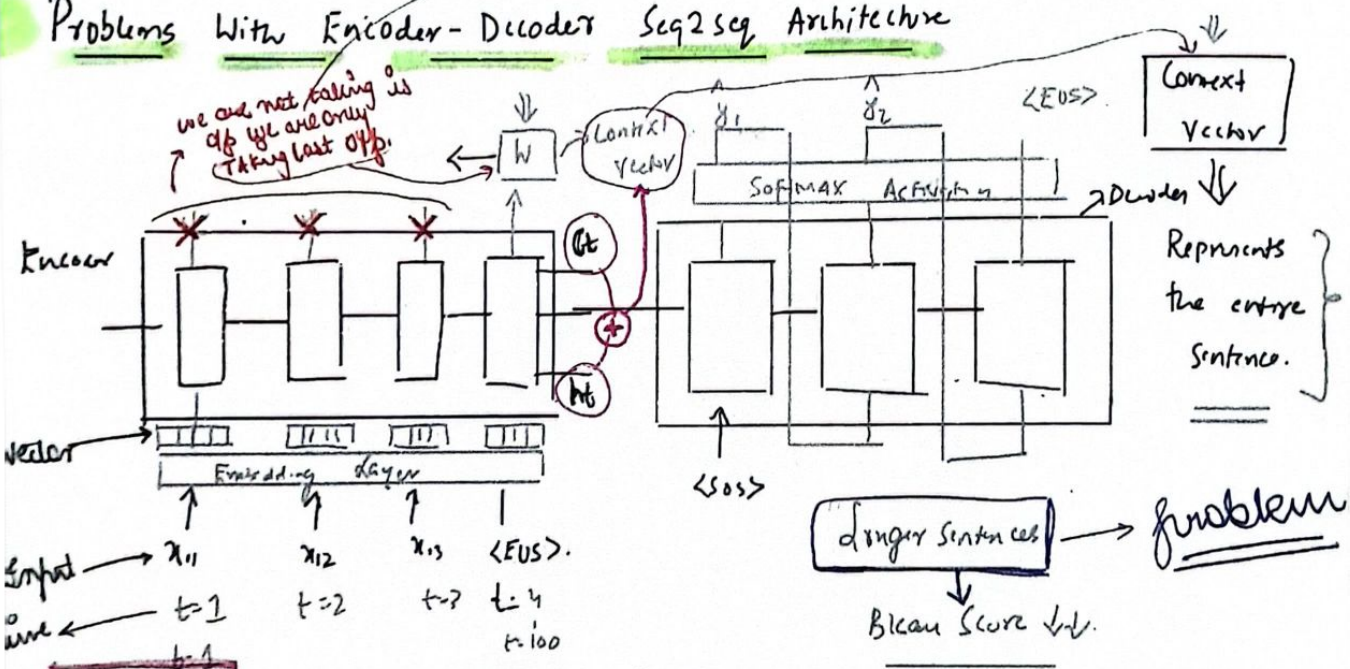
Since 0.6 is largest among all values we get previous as o/p

<EOS>.

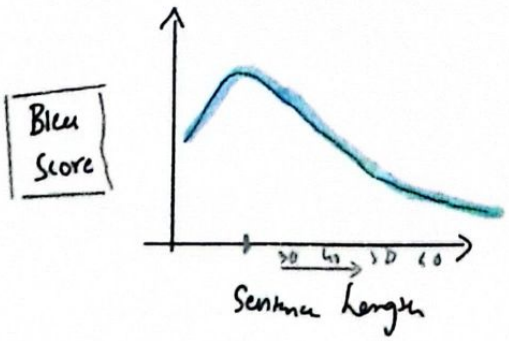


Sentence will have more info about last words

Problems With Encoder-Decoder Seq2seq Architecture



Rucarchurs : Sentences of varying length

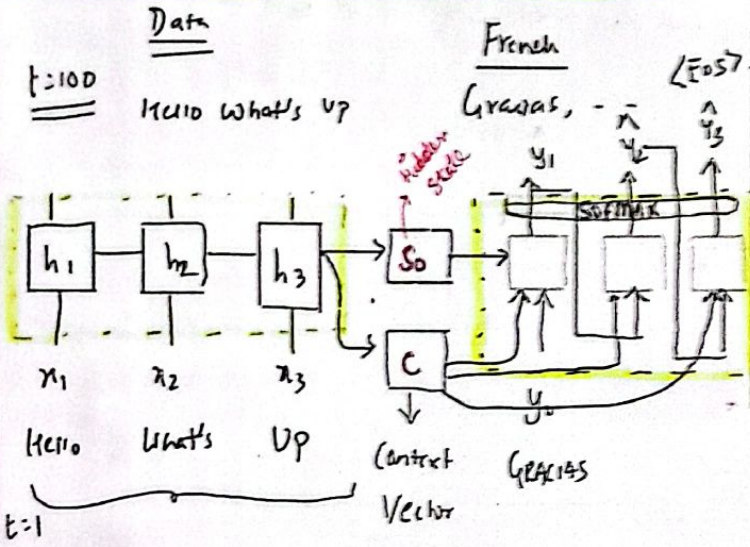


⇒ Seq to Seq Data

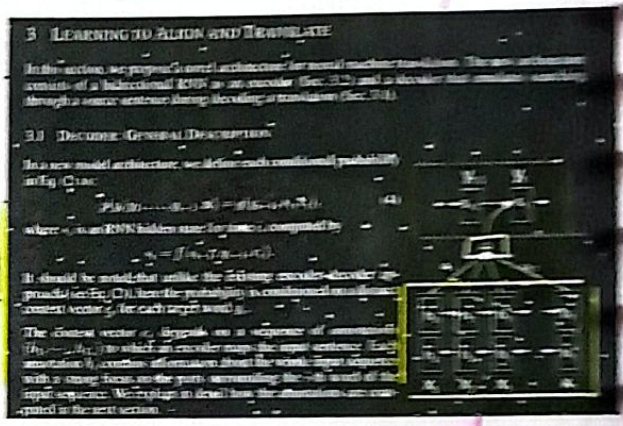
⑧ Attention Mechanism → Seq2Seq Network

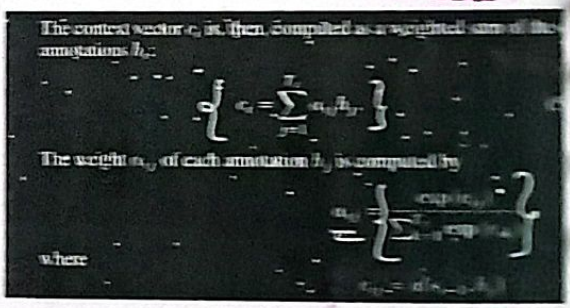
longer paragraph → {context vector}
+
{Context}

Attention Mechanism / Seq2Seq Networks



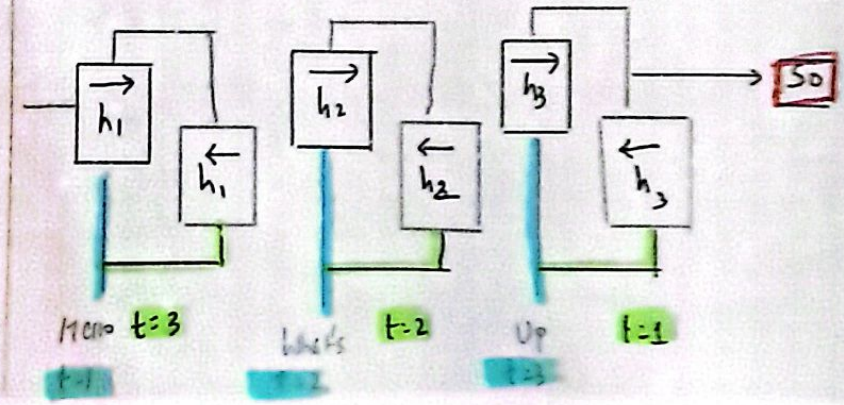
Encoder Decoder Architecture

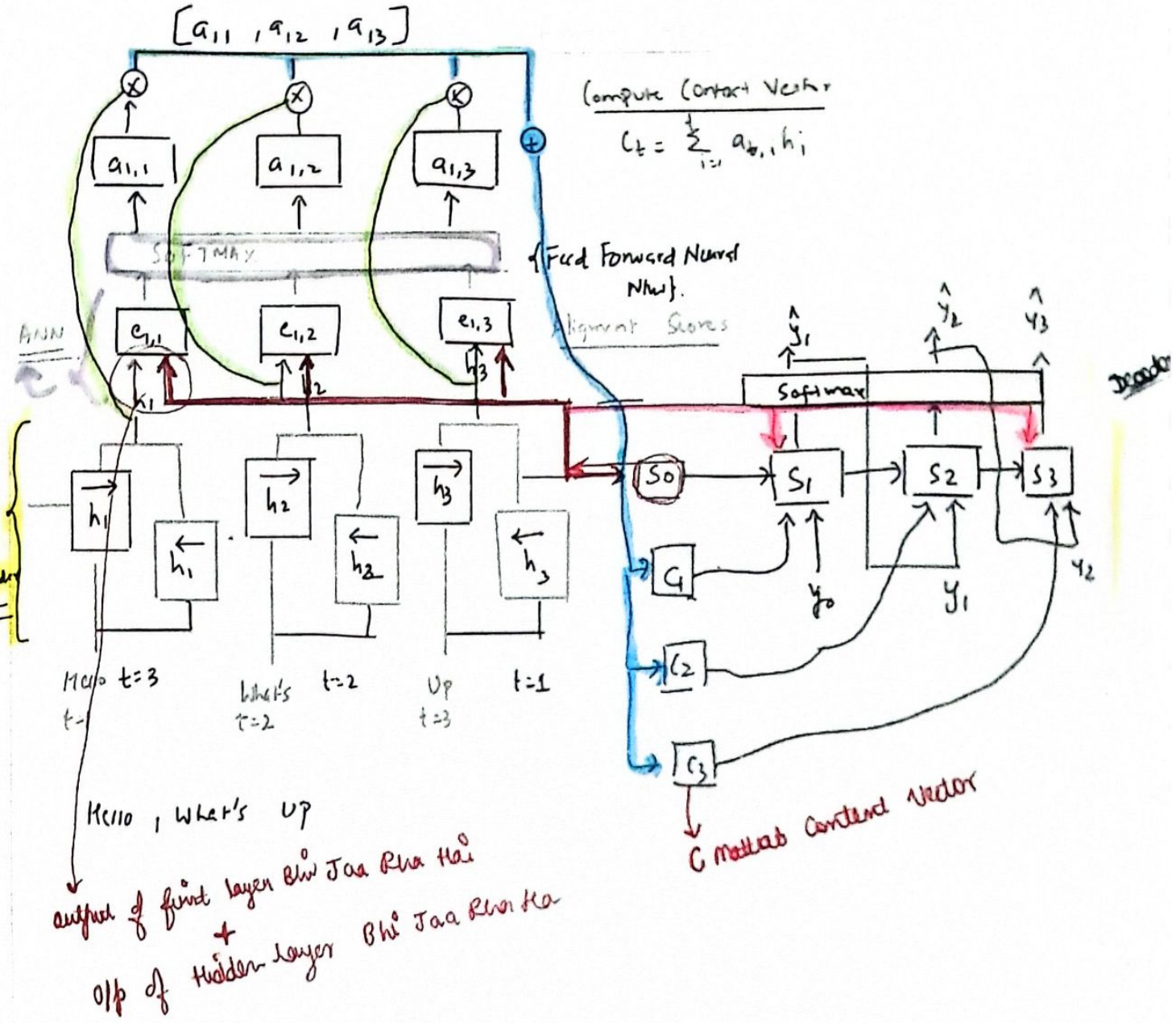




Attention Mechanism

https://erdem.pl/2021/05/introduction-to-attention-mechanism




[a11 , a12 , a13]
Compute Context Vector
$c_t = \sum_{i=1} a_{t,i} h_i$
SOFTMAX
{Feed Forward Neural Nwt}.
Alignment Scores
Softmax
Decoder
Encoder
Hello t=3
What's t=2
Up t=1
Hello , What's Up
output of first layer Bhi Jaa Rha Hai
+
O/p of Hidden layer Bhi Jaa Rha Ha
C matlab Context Vector