3rd March 2025:

Sequences:

MLP, CNN } Fixed length data

Language Translation. Length of i/p & o/p not fixed.

→ Images of Varying resolution.

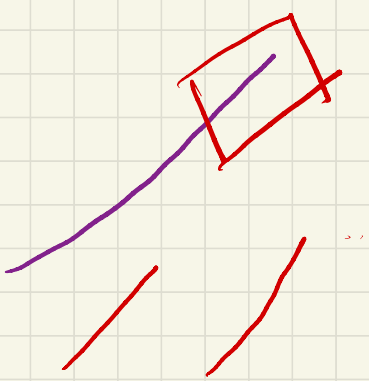
medical records.

Varying length i/p : → fixed length problem.

"fixing the len sequence"

Collection of items in an order : Sequences.

Day	Temp
1	30
2	32
3	33
4	31
5	?

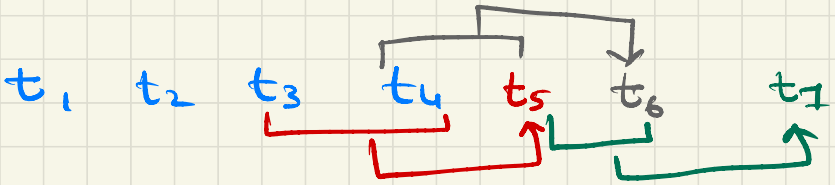


how much to go back?

taking only the past 2 days.

$$t_5 = b + w_1 t_4 + w_2 t_3$$

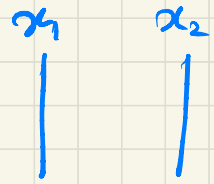
$t_6 = b + w_1 t_5 + w_2 t_4$



Auto Regressive models.

Auto → Self

Regressive → looks back in time.



height	weight
h_1	s_1
h_2	s_2
h_3	⋮
⋮	
h_m	s_m

Same

height	weight
h_1	s_1
h_m	s_m
h_2	s_2
h_3	s_3
⋮	⋮

t_1 t_2 t_3 t_4 t_5

t_2 t_5 t_3 t_4 t_1 ✗

t_0 t_1 t_2 t_3 t_4 t_5 t_6

X			Y
t_0	t_1	t_2	t_3
t_1	t_2	t_3	t_4
t_2	t_3	t_4	t_5
t_3	t_4	t_5	t_6

fix "p"

$$AR(p): \quad x_t = b + \sum_{i=1}^{p} w_i x_{t-i}$$

$p = 1$:

AR (1) :

$$x_t = b + w_1 x_{t-1}$$

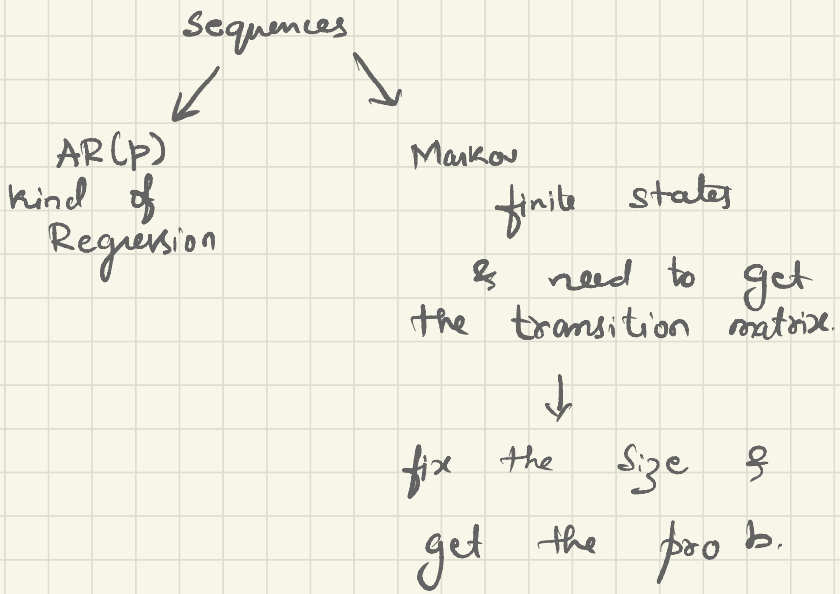
$$x_2 = b + w_1 x_1$$

$$x_3 = b + w_1 x_2$$

Markov:

o/p: next State

how many next states ?



Text to Numbers ?

I	am	enjoying	my	studies.
0	1	2	3	4

(821)

I	love	my	country.
0	1	2	3

(242)

Oxford English Dict.

1 world

2

3

⋮

242 : country.

enjoy : 821

Coffee is great

Car battery is dead, oh great

each word some number based on the index of constructed dictionary

I am from Mysore
My name is Raghava
I like eating Dosa.

- Tokenisation:

Vocabulary

6th Mar 2025:

fixed length i/p → fixed length o/p

→ Variable length i/p → fixed length o/p

Named Entity Representation: NER

XX Harry potter won the Triwizard Cup. X X

1 1 0 0 1 0

x_1 x_2 x_3 x_4 x_5 x_6

x_t : t^{th} feature

$x_{i,j}$: j^{th} feature in i^{th} example.

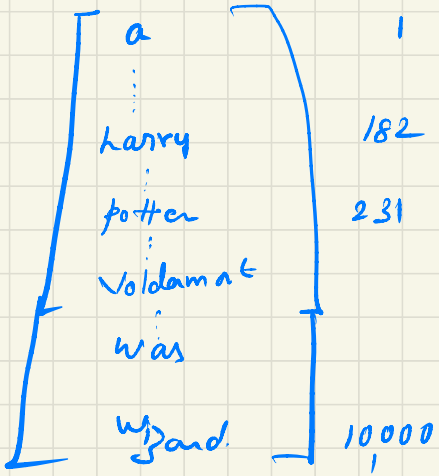
In this specific case of NER.

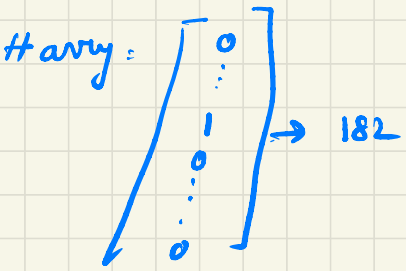
$y_1, y_2 \cdots, y_6$

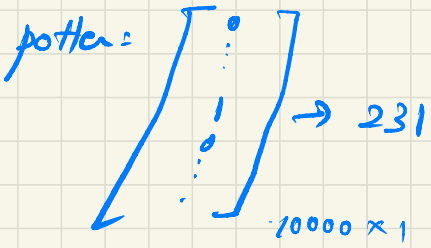
$y_{i,j}$: j^{th} output of i^{th} feature

Construct Vocabulary:

num of unique tokens (Size of vocabulary) is 10,000







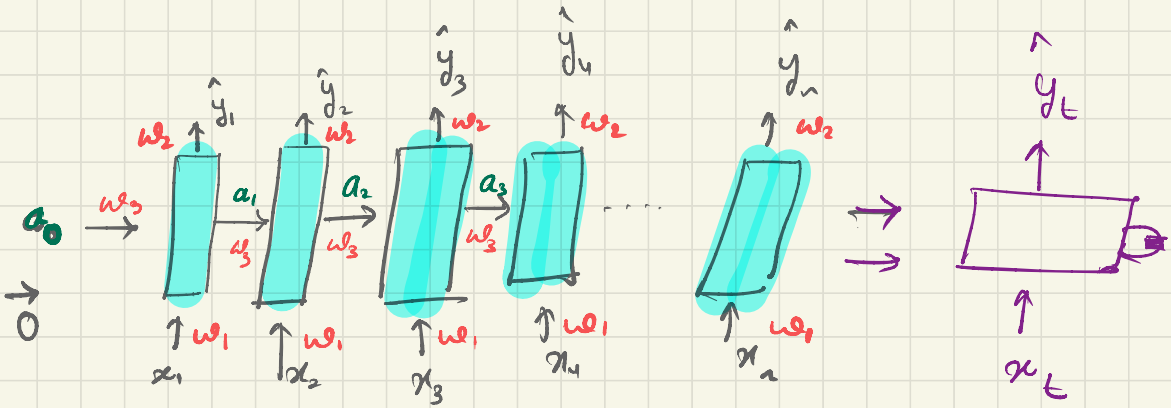
One hot encoding

model to map from X → Y.

word not in Vocabulary <unk>

→ we can perform zero padding which is in-efficient.

RNNs: Recurrent Neural Networks:



input is given from left to right

$$a_0 = \vec{0}$$

$$a_1 = g_1(w_1 \cdot x_1 + w_3 \cdot a_0 + b_1)$$

$$\hat{y}_1 = g_2(w_2 \cdot a_1 + b_2)$$

$$a_2 = g_1(w_1 \cdot x_2 + w_3 \cdot a_1 + b_1)$$

$$\hat{y}_2 = g_2(w_2 \cdot a_2 + b_2)$$

tanh / Relu : Activation.

$$a_t = g_1(w_1 \cdot x_t + w_3 \cdot a_{t-1} + b_1)$$

$$\hat{y}_t = g_2(w_2 \cdot a_t + b_2)$$

10th march 2025:

Problem with RNNs

→ Teddy Roosevelt was the president of USA.

1 1 0 0 0 0 1

→ Teddy Bears are for Sale in Big bazaar.

0 0 0 0 0 0 1

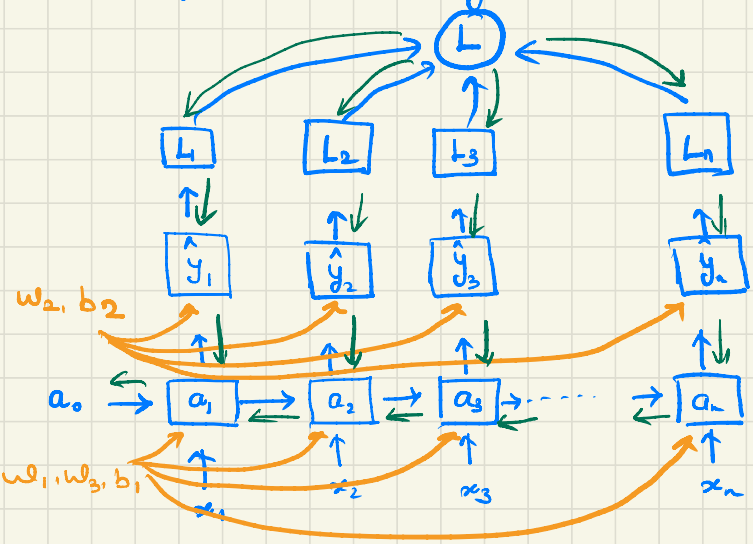
Note: to decide the NER we need to look forward for future words, which is not possible in RNNs.

Bi directional RNNs.

How to update the weights.

Computational graph

Back Propagation Through Time



Binary cross entropy loss :

$$L_1(\hat{y}_1, y_1) = -y_1 \log \hat{y}_1 - (1-y_1) \log (1-\hat{y}_1)$$

$$L(\hat{y}, y) = \sum_{i=1} L_i(\hat{y}_i, y_i)$$

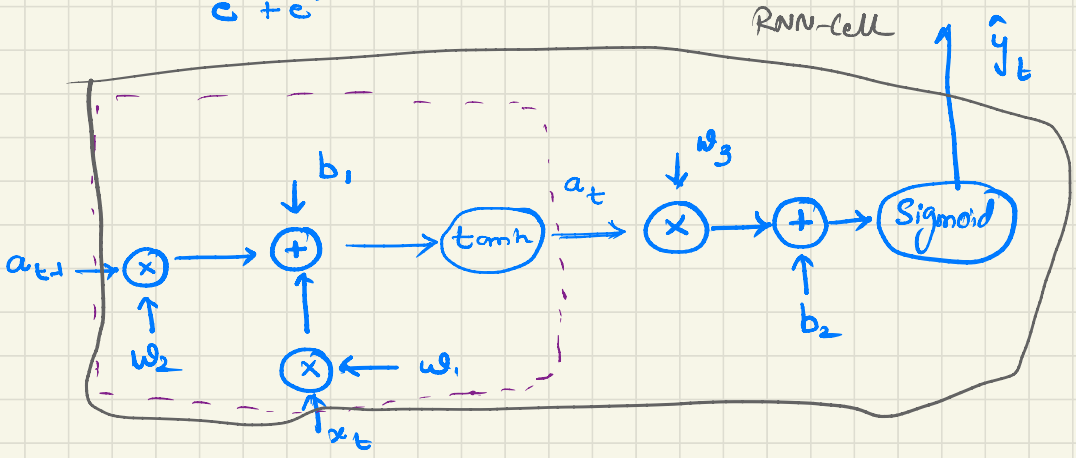
$$\begin{bmatrix} y_1 \\ 1-y_1 \end{bmatrix} \quad \begin{bmatrix} \hat{y}_1 \\ 1-\hat{y}_1 \end{bmatrix}$$

11th March 2025:

$$a_t = \tanh(w_1 x_t + w_2 a_{t-1} + b_1)$$

$$\hat{y} = \text{Sigmoid}(w_3 a_t + b_2)$$

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$



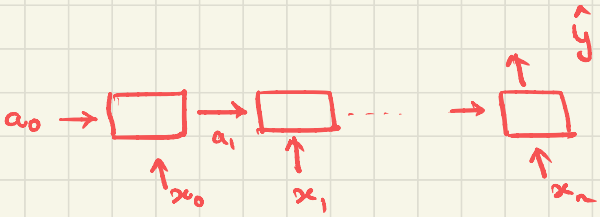
NER model we are looking is a Many to Many model. & size of input & output were same.

→ Movie Ratings

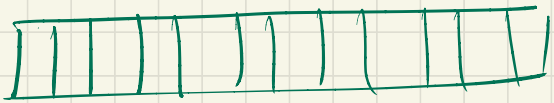
input: Review

output: number of stars $\{1, 2, 3, 4, 5\}$

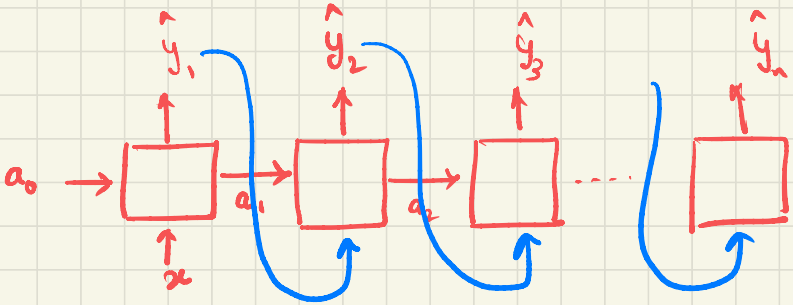
Many to One Mapping.



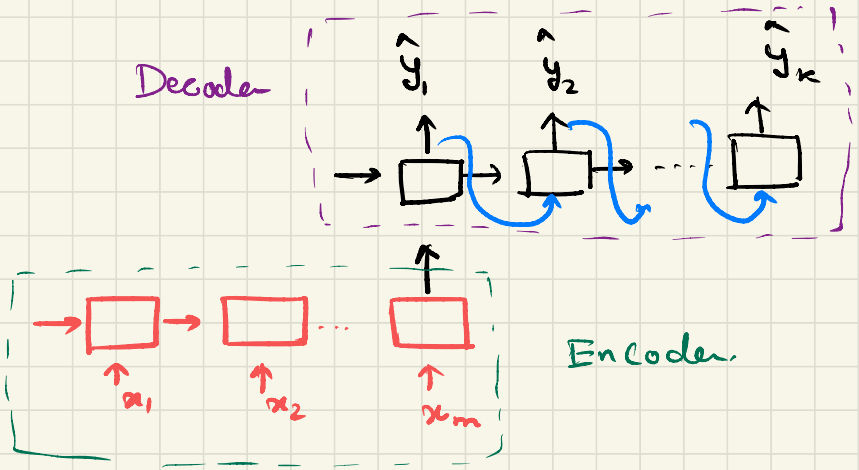
→ given the base node, play Some Raga.



One to Many Problem.



Machine translation : Many to Many. The length of i/p & o/p are diff



Consider Speech to Text

1: The apple & Pair Salad was good

2: The apple & pear Salad was good

Occurance (S_2) > Occurance (S_1)

I have total of "N" sentences in my corpus.

$$\frac{O(S_2)}{N} > \frac{O(S_1)}{N}$$

$$P[S_2] > P[S_1]$$

I like eating Pongal
x_1 x_2 x_3 x_4

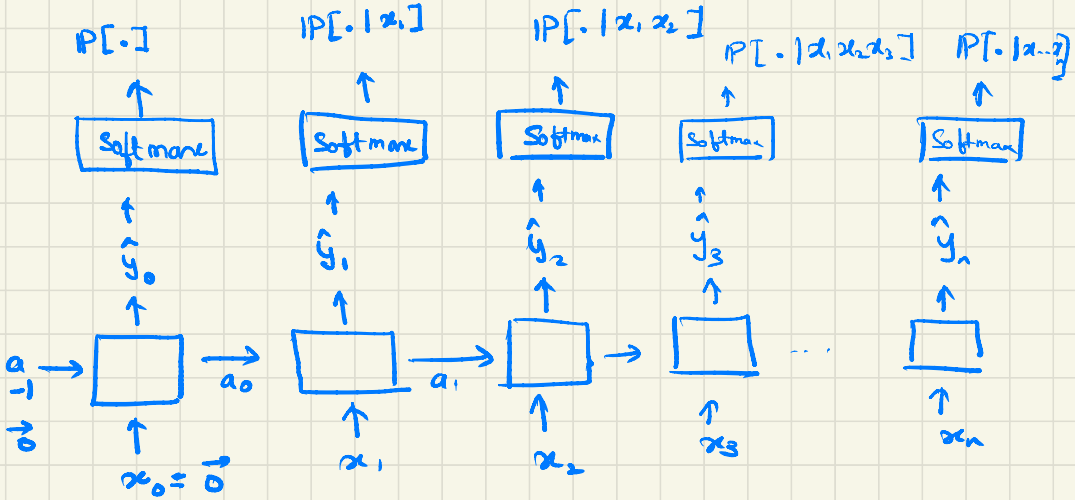
word Translation problem.

length of i/p & o/p is same.

output should choose one word out of all the words in the vocabulary.

$\therefore$ need to use Softmax.

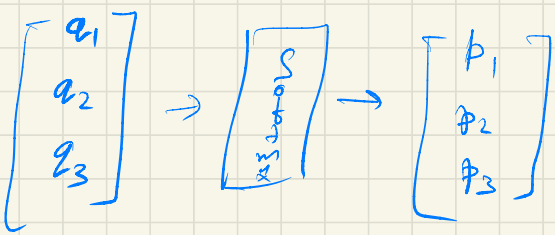
Consider there are 10 words in vocabulary then we should get a probability vector of length 10.



[Eat, the, life, is, going, on, well, orange, is, good]

The ___

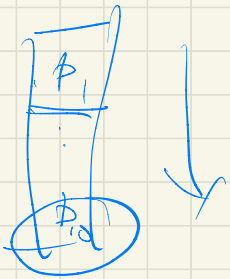
The orange ___



$$p_1 = \frac{e^{q_1}}{e^{q_1} + e^{q_2} + e^{q_3}}$$

$$p_2 = \frac{e^{q_2}}{e^{q_1} + e^{q_2} + e^{q_3}}$$

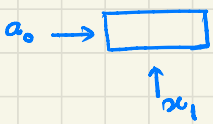
$$p_3 = \frac{e^{q_3}}{e^{q_1} + e^{q_2} + e^{q_3}}$$



12th March 2025:

Size of vocab: 5000

S_1 : I like eating dosa
S_2 : I am from mysore
S_3 : we are not in mars
S_4 : I like teaching
S_5 : Once there was a lion



→ Size of Vocabulary : 5000

each word is a vector in 5000 dim $x_1 \in \mathbb{R}^{5000}$

→ Consider the batch of size "n"

if we want to give the batch as input

then what will be the size 5000×5

input @ every time step :

→ what will be the size considering all time stamps together.

max time stamp : 10

then @ max the whole input is $5000 \times 5 \times 10$

x_1 x_2 ... x_{10}

$$a_t = (w_1 x_t + w_2 a_{t-1} + b_1)$$

$$x_t \in \mathbb{R}^{5000 \times 5}$$

$$w_1 \in ? \quad \text{(231)} \quad \mathbb{R}^{10}$$

$$? \times 5000$$

$$w_1 \in \mathbb{R}$$

→ Size of w_1 & w_2 are dependent on Size of a_t

→ this can be fixed by user : 231

that means for one word a_t generated is a Vector of Size 231

∴ for a batch 231 × 5

∴ for all the time stamps 231 × 5 × 10

$$w_2 \cdot a_{t-1}$$

↓ ↓

231 × 231 231 × 5

$$w_1 \in \mathbb{R}^{231 \times 5000}$$

$$w_2 \in \mathbb{R}^{231 \times 231}$$

$$a_t = w_1 * x_t + w_2 * a_{t-1} + b$$

$$= (231 * 5000)(5000 * 5) + (231 * 231)(231 * 5)$$

$$a_t = (231 * 5) + (231 * 5) + \left[\ \right]_{231*1}$$

$$\hat{y} = (w_3 * a_t + b_2)$$

$$a_t \in \mathbb{R}^{231*5}$$

$$w_3 \in \mathbb{R}^{?*231}$$

w_3 a_t

$(? \times 231)$ (231×5)

what should be the size of output

$$w_3 \in \mathbb{R}^{5000 \times 5}$$

$$b_2 \in \mathbb{R}^{5000 \times 1}$$

$$w_1 \in \mathbb{R}^{231 \times 5000} \qquad b_1 \in \mathbb{R}^{231 \times 1}$$

$$w_2 \in \mathbb{R}^{231 \times 231}$$

$$w_3 \in \mathbb{R}^{5000 \times 231} \qquad b_2 \in \mathbb{R}^{5000 \times 1}$$

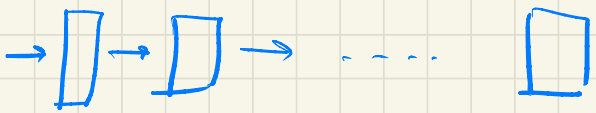
Vanishing Gradients:

next word prediction:

The man, who was . . . . . is . . . .

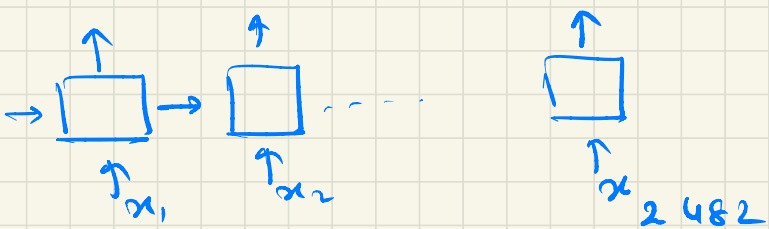
The men, who were . . . . . are . . . .

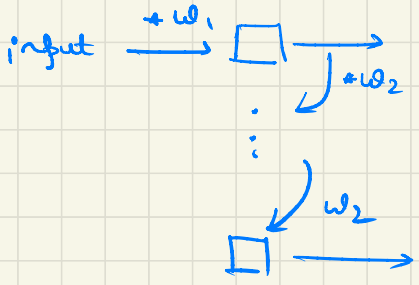
→ long term dependencies.



200 layer MLP.

Vanishing gradients. (NaN)





$n=50$

$input * w_1 * w_2^n$

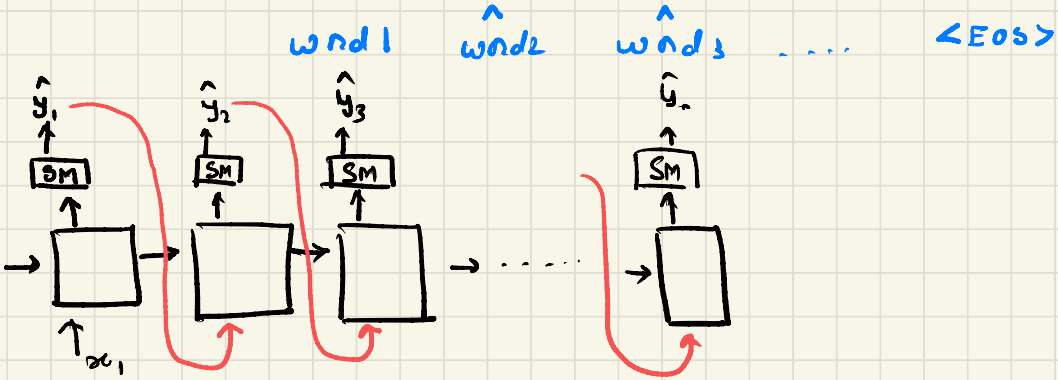
$\frac{\partial}{\partial w_1} \approx input * w_2^n$

if $w_2 = 2 \rightarrow \frac{\partial}{\partial w_1} = input * 2^{50}$

$w_2 = 0.5 \Rightarrow \frac{\partial}{\partial w_1} = input * (0.5)^{50}$

13th March 2025.

Sequence generation: Next word Prediction.



GRU : Gated Recurrent Unit.

What to remember. Problem : Sequence generation.

The Cat . . . . . is . . .

we need to remember that we have used Singular earlier & generate the accordingly.

we introduce an idea of remember entity "C_t"

$C_t = a_t$ is same in GRU.

$$\tilde{C}_t = \tanh\left(w_c [C_{t-1}, x_t] + b_c \right)$$

Previous Content

$$w_c = w_c^1 C_{t-1} + w_c^2 x_t$$

$$= w_c \cdot [C_{t-1}, x_t]$$

$$\Gamma_u = \sigma(W_u[c_{t-1}, x_t] + b_u)$$

how much to remember from previous.

$$c_t = \Gamma_u * \tilde{c}_t + (1 - \Gamma_u) * c_{t-1}$$

$\Gamma_u = 1$

The Cat . . . . . is

$\Gamma_u = 0$ $\Gamma_u = 0$ $\Gamma_u = 0$

→ wkt value of c_t is pretty much retained so there is no problem of vanishing gradients.

Singular / Plural

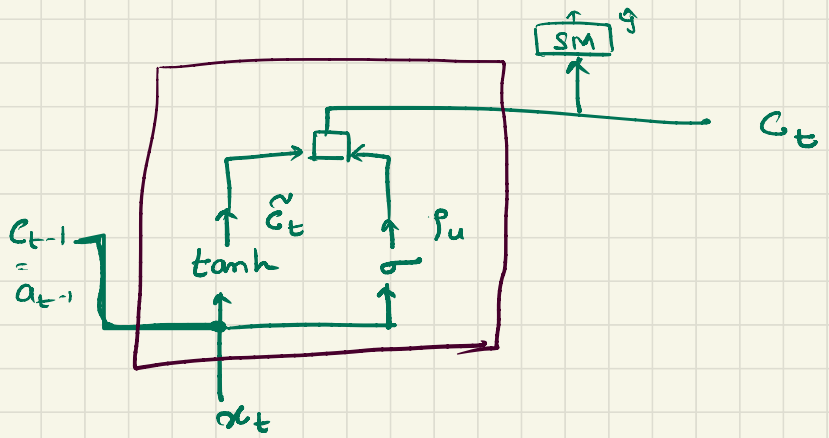
Past / Present

$$\begin{bmatrix} \cdot \\ \vdots \end{bmatrix}$$

c_t is a vector.

Then Γ_u will also be a vector.

→ Γ_u will say which terms (bits) to update.



LSTM : Long Short term Memory

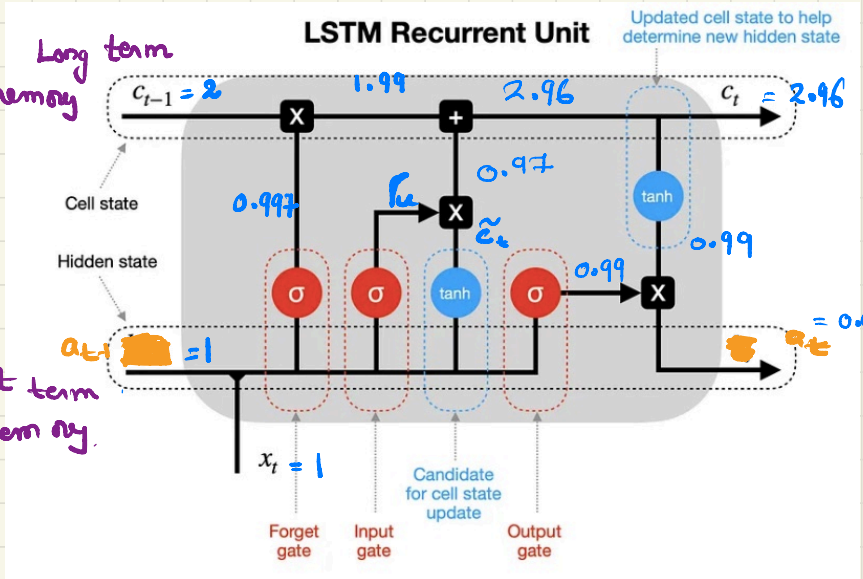
→ Better than GRUs to get long term dependencies

→ (1997)

→ a_t & c_t are different

$$\tilde{c}_t = \tanh\left(w_c \left[a_{t-1}, x_t \right] + b_c \right]$$

$$\Gamma_u = \sigma\left(w_u \left[a_{t-1}, x_t \right] + b_u \right]$$

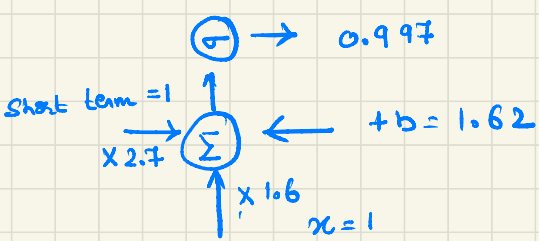


+ 2 separate paths to make predictions
  * long term memory
  * short term memory.

* Upper path for long term it's modified by multiplication & addition ⇒ No weights & Bias.

Forget Gate:

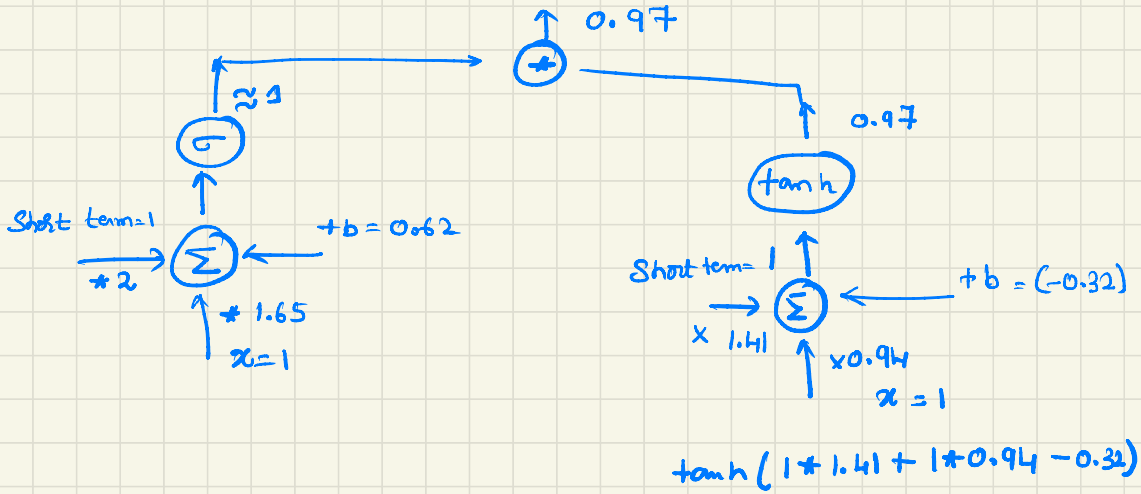
what % of long term memory do we need to remember



$\sigma(1 + 2.7 + 1 * 1.6 + 1.62)$

=

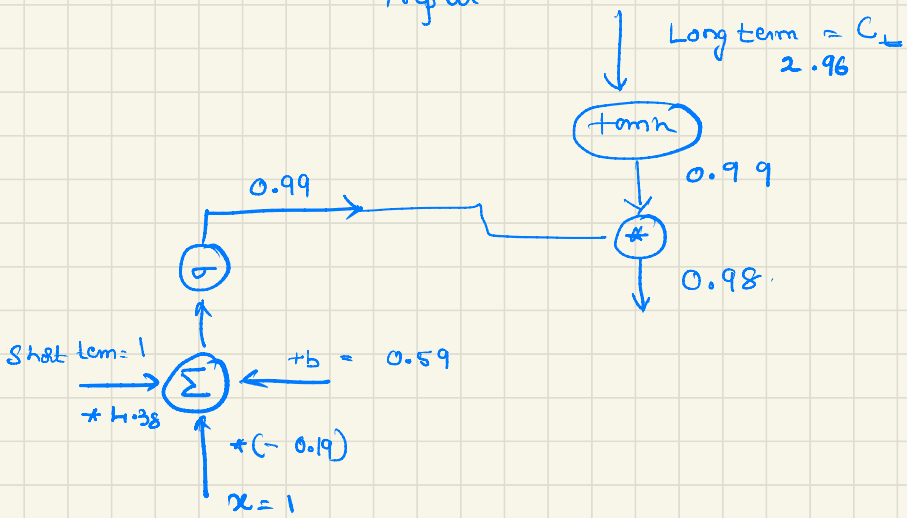
Input Gate: Potential memory need to added to the long term based on short term & input.



Output gate:

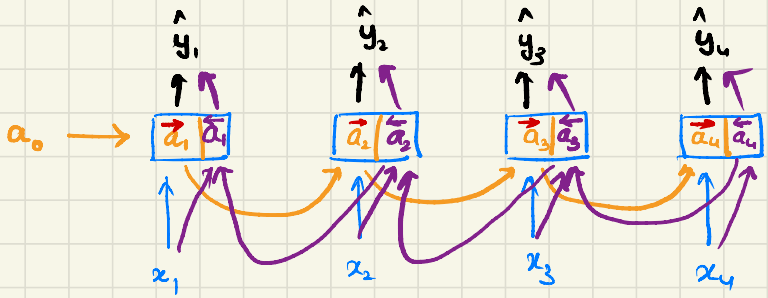
Potential short term memory obtained by modulating.

Long term
Short term
input



RNNs
GRUs
LSTMs
} ⇒ Parsing i/p from left to Right

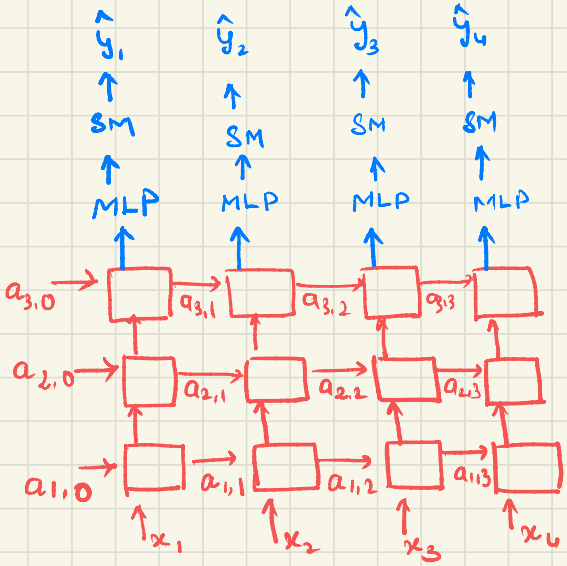
Bi-Directional RNNs

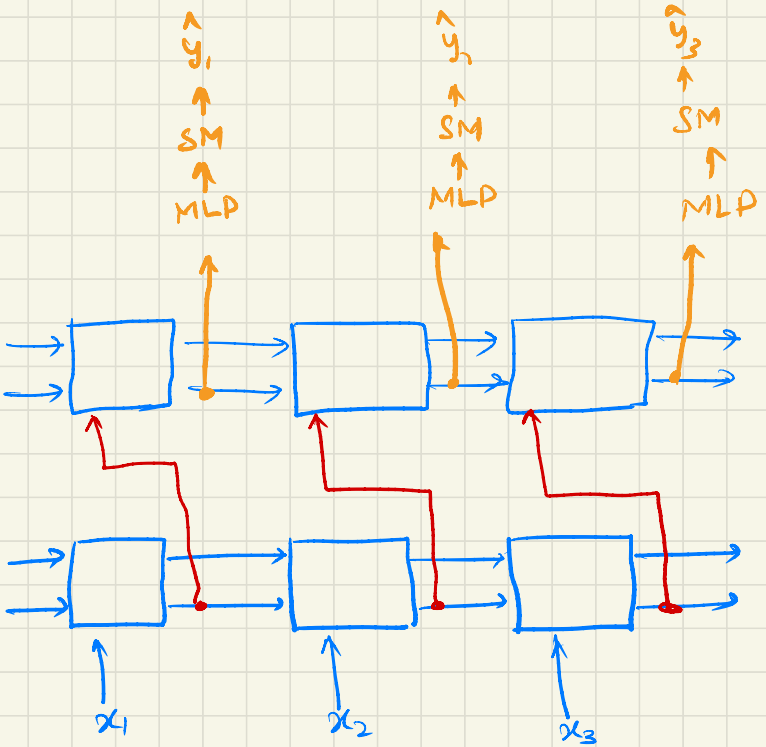


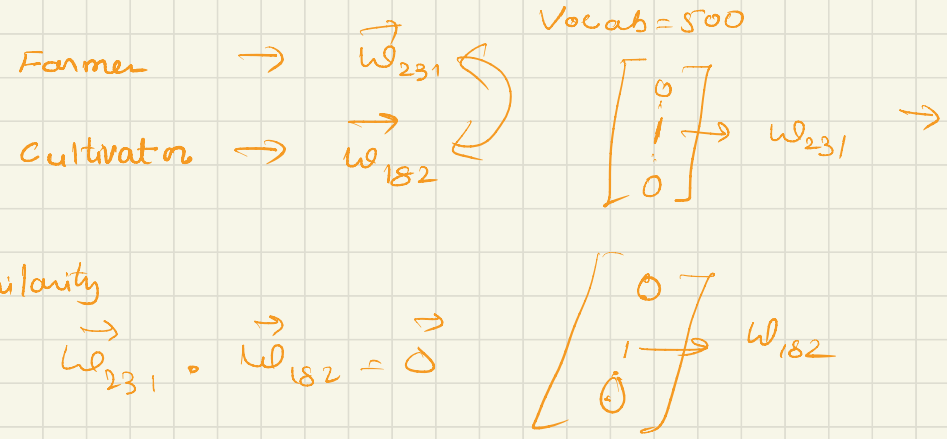
$$g\left(w_1 \vec{a}_1 + w_2 \overleftarrow{a}_1 + b\right)$$

bi-directional GRUs
bi-directional LSTMs

Deep RNNs :


$\hat{y}_1$
$\hat{y}_2$
$\hat{y}_3$
$\hat{y}_4$
SM
SM
SM
SM
MLP
MLP
MLP
MLP
$a_{3,0}$
$a_{3,1}$
$a_{3,2}$
$a_{3,3}$
$a_{2,0}$
$a_{2,1}$
$a_{2,2}$
$a_{2,3}$
$a_{1,0}$
$a_{1,1}$
$a_{1,2}$
$a_{1,3}$
x_1
x_2
x_3
x_4


$\hat{y}_1$
$\hat{y}_2$
$\hat{y}_3$
SM
SM
SM
MLP
MLP
MLP
x_1
x_2
x_3



Word Embeddings: Vocab Size: 500

Man	Women	King	Queen
w_{142}	w_{420}	w_{111}	w_{341}

I want a glass of orange ____

I want a glass of apple ____

* relation b/w orange & apple is not easy generalize.
* Dot product of orange & apple is zero.

⇒ there is no relation, which is not true.

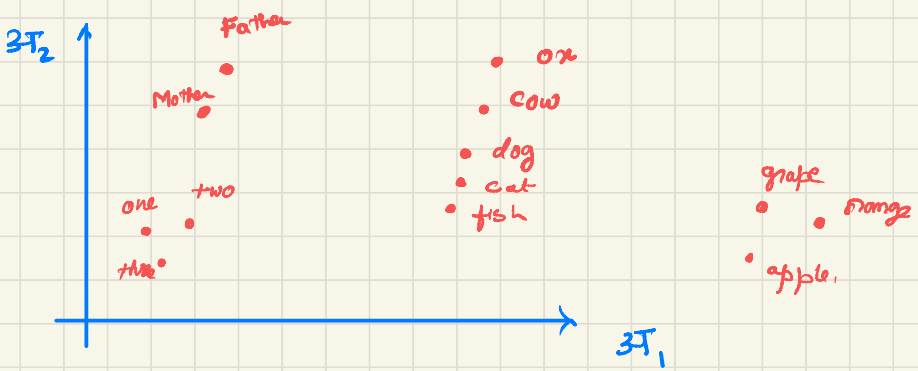
	Cow	Ox	Mother	Father	Apple	Orange
animal	0.99	0.99	0.01	0.01	0.01	0.01
Food	0.01	0.01	0.01	0.01	0.99	0.99
Relation	0.01	0.01	0.99	0.99	0.01	0.01
Gender (+1, -1)	-1	+1	-1	+1	0.01	0.01
⋮						

e_i : feature vector of i^{th} word.

* we consider some 182 feature

* now the dot product b/w the feature vectors tell us how they are related.

* t-SNE Plot : 2D Visualizing word embedding.



* huge Corpus of text

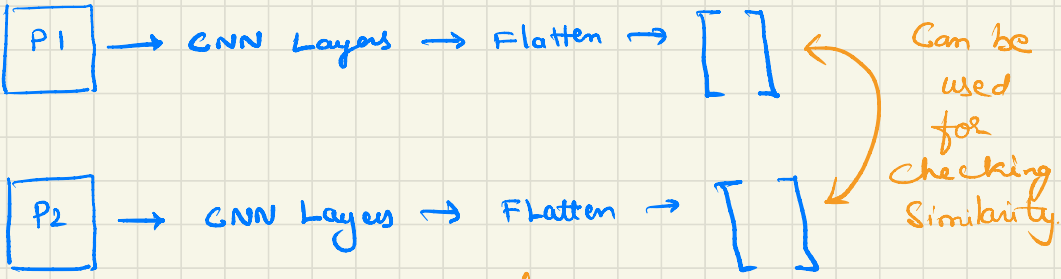
1 Billion words → (learn) → word embeddings

we can take these word embeddings which are learnt on huge Corpus and use for our task.

"Transfer Learning"

* Fine tuning: train on your data for a very small time.

*

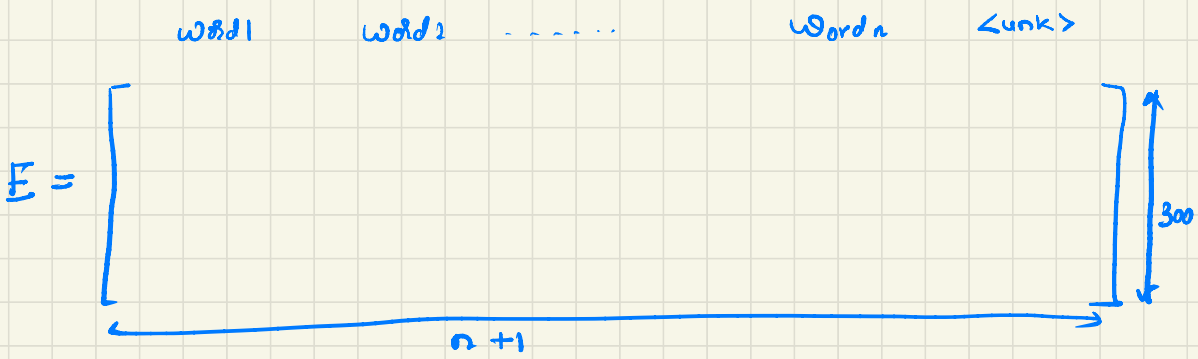


$$Cow = \begin{bmatrix} 0.99 \\ 0.01 \\ 0.01 \\ -1 \end{bmatrix} \qquad \overrightarrow{Cow} - \overrightarrow{ox} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2 \end{bmatrix} \qquad ox = \begin{bmatrix} 0.99 \\ 0.01 \\ 0.01 \\ +1 \end{bmatrix}$$

$$Mother = \begin{bmatrix} 0.01 \\ 0.01 \\ 0.99 \\ -1 \end{bmatrix} \qquad \overrightarrow{Mother} - \overrightarrow{Father} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2 \end{bmatrix} \qquad Father = \begin{bmatrix} 0.01 \\ 0.01 \\ 0.99 \\ +1 \end{bmatrix}$$

$e_{cow} - e_{ox} \approx e_{mother} - e_{father}$

$\approx e_{daughter} - e_{son}$



Embedding Matrix.

$E \cdot O_{182} = e_{182}$

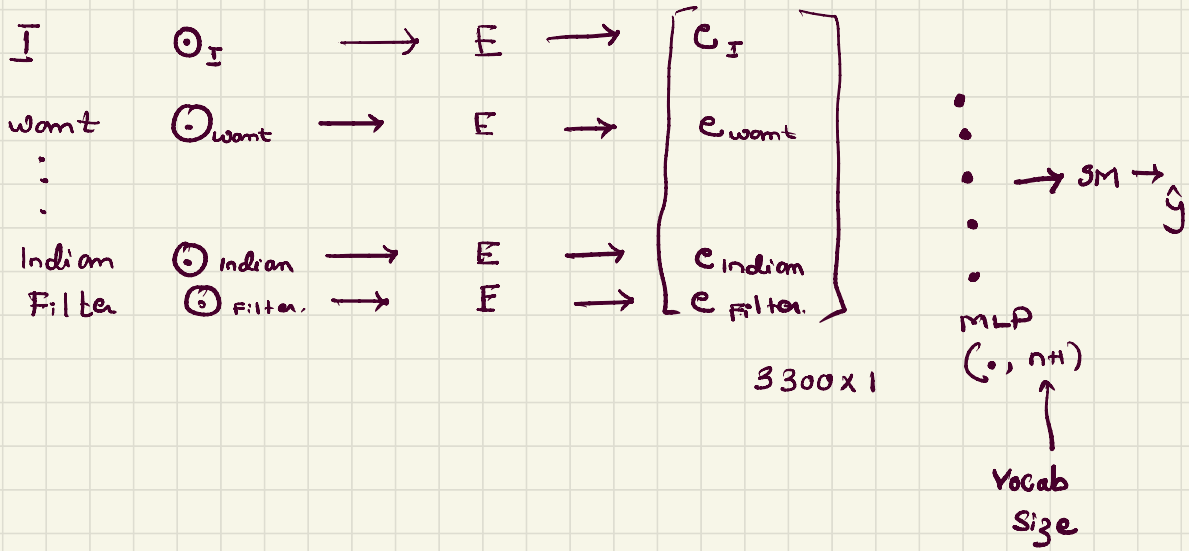
$$\begin{bmatrix} E \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} e_{182} \end{bmatrix}$$

Learning Word Embedding : Neural Language Models

: Next word prediction.

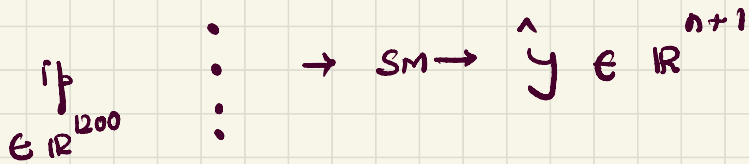
I want to drink a hot cup of South

Indian Filter _____



→ No need to take 11 vectors always
Fix a window previous 4 words

input $\in \mathbb{R}^{1200}$



→ Context ⟶ target

↳ words as i/p.

→ you can also modify the context

↳ words from right, ↳ words from left.

→ Simple context : One word context.

$$\begin{bmatrix} 0.1 \\ 0.2 \\ 0.4 \\ 0.1 \\ 0.2 \end{bmatrix}$$

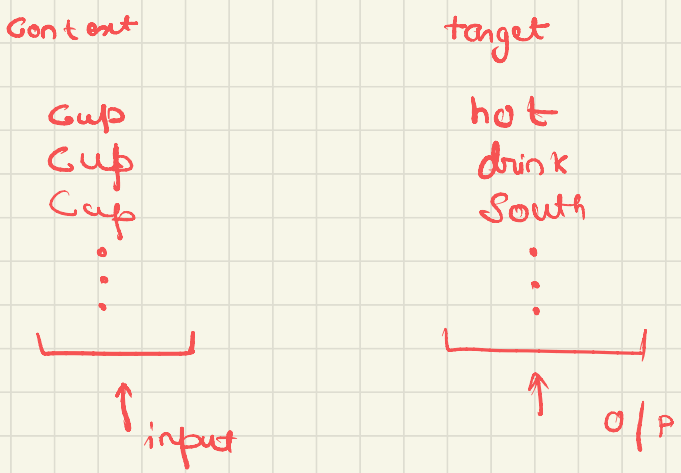
Word 2 Vec:

→ skip grams

I want to drink a hot cup of South Indian Filter Coffee, with Idli Sambar.

→ Choose a word randomly as context.

→ Target is Choosen in a window.



model :

map Context "c" → target "t".

$$\odot_c \rightarrow E \rightarrow e_c \rightarrow MLP \rightarrow SM \rightarrow \hat{y}$$

$$\hat{y} = IP[t \mid c]$$

Prob of target given the context.

this will be a prob vector of size (n+1)

$$= \frac{e^{\theta_t^T \cdot e_c}}{\sum_{j=1}^{n+1} e^{\theta_j^T \cdot e_c}}$$

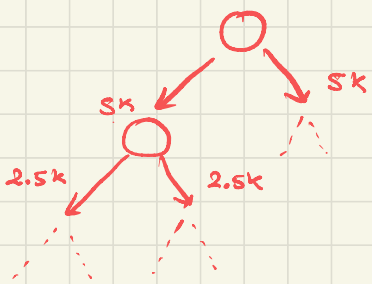
$$L(\hat{y}, y) = -\sum_{i=1}^{n+1} y_i \log \hat{y}_i$$

CCE Loss.

Hierarical Softmax

Vocab Size : 10,000

binary Classifier: Logistic regre



→ words of vocabulary is the leaf nodes.
it has a unique path from the root.

→ you will have around $\log_2 n$

→ The prob is computed as a product of binary probabilities along the path from root to the word.

→ Skip gram O (size of vocab)

Hierarical O ($\log_2$ (size of vocab))

Glove Algorithm:

Global Vectors for word representations.

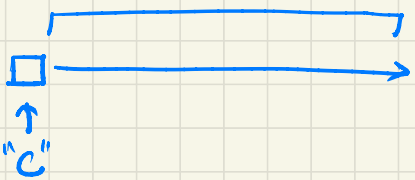
→ from the data Come up with the Co-occurance matrix.

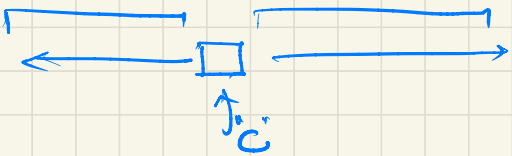
	i's	the	apple	fruit	Computer.
apple	5	20	100	50	5
fruit	3	15	50	100	1
Computer	1	10	5	1	100

→ window & then Compute this.

x_{ij} : # times "i" appears in the Context of "j"
"t" "c"

x_i : # times "i" appey in the Context





$$p_{ij} = \frac{x_{ij}}{x_i}$$

p_{ij} is greater in magnitude :

i & j are strongly related.

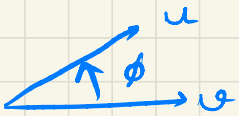
* when we have target appearing ± 10 words from the target

$$x_{ij} = x_{ji}$$

but x_i may not be equal to x_j

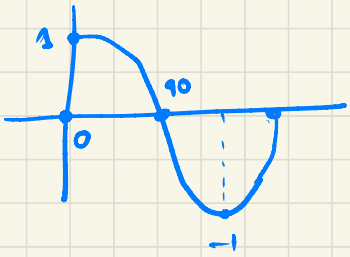
∴ p_{ij} may not be equal to p_{ji}

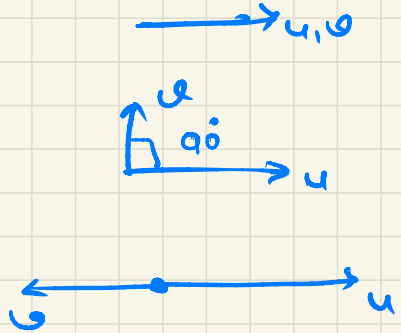
* $$\sum_{i=1}^{10k} \sum_{j=1}^{10k} \left(\underbrace{\theta_i^T e_j}_{\text{how is the similarity b/w i \& j}} + b_i + b_j - \log(x_{ij}) \right)^2$$

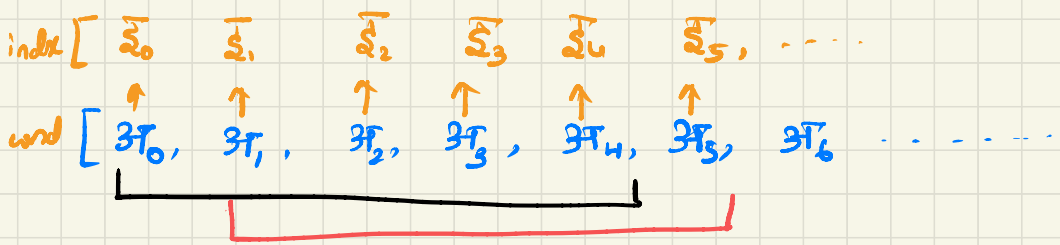


$$\text{Sim}(u, v) = \frac{u^T v}{\|u\|_2 \, \|v\|_2}$$

$$\|u\|_2 = \sqrt{u_1^2 + u_2^2 + \cdots + u_k^2}$$







i/p [[s0, s1, s2, s3, s4], [s1, s2, s3, s4, s4], ...

o/p [[s5], [s6] . . . .

Sentiment Analysis:

i/p tweet

o/p : very offensive - 0

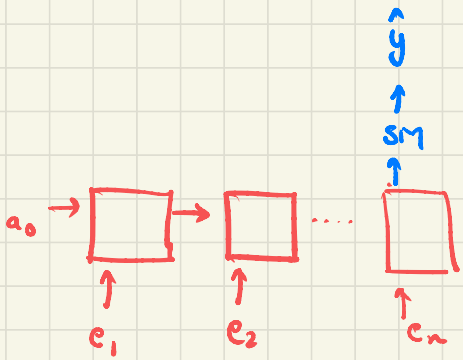
offensive - 1

normal - 2

good - 3

very good. - 4

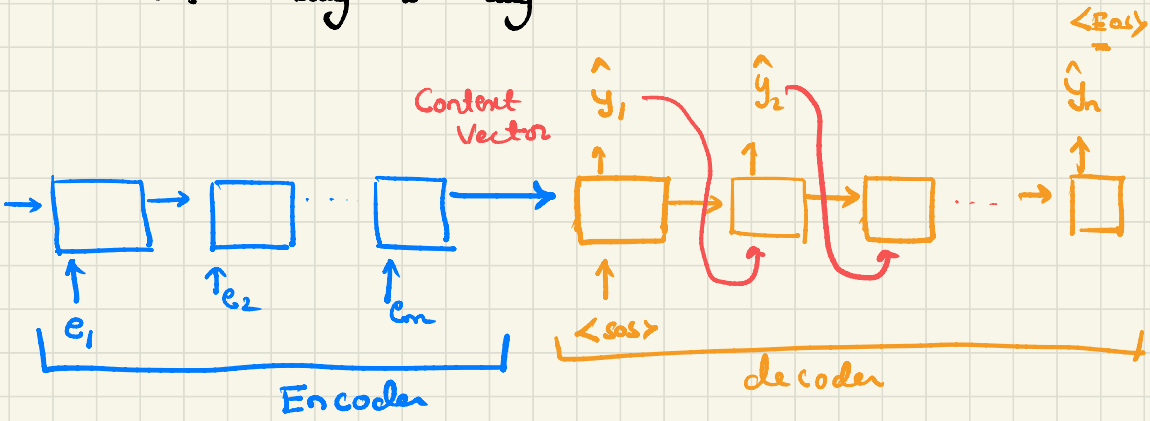
Arch : many to One.



Google Translate: Sequence to Sequence Model.

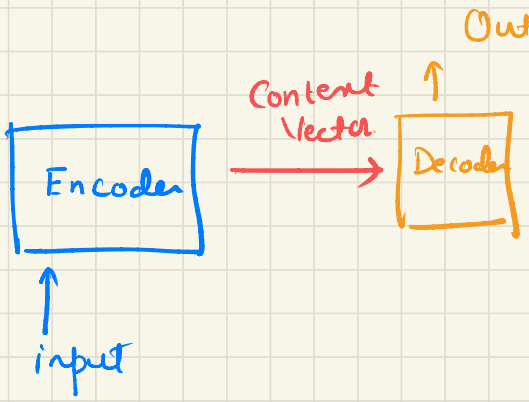
French to English.

Arch : Many to Many



<SOS> : Start of Sentence.

<EOS> : End of Sentence



BLEU : Bilingual Evaluation Understudy

i/p : बिल्ली चटाई पर है

Machine Translation (Candidate) : The cat is on the mat

(Reference) Human Translation :

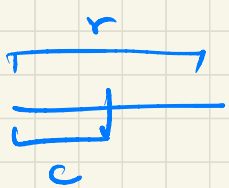
There is a Cat on the mat

Candidate : [the, Cat, is, On, the, mat]

Reference : [There, is, a, Cat, On, the, mat]

Candidate unigrams :

	Candidate	Reference
the	2	1
Cat	1	1
is	1	1
On	1	1
mat	1	1



$$p_1 = \frac{\text{matched unigrams}}{\text{Total candidate unigrams}} = \frac{5}{6} \approx 0.83$$

Candidate Bi-grams.

	Candidate	Reference
the cat	1	0
cat is	1	0
is on	1	0
on the	1	1
the mat	1	1

$$p_2 = \frac{\text{Matched bigrams}}{\text{Total Candidate bigrams}} = \frac{2}{5} = 0.4$$

* Decide on how many n-grams you need to take

normally people go till p_4

$$\frac{1}{4}\log p_1 + \frac{1}{4}\log p_2 + \frac{1}{4}\log p_3 + \frac{1}{4}\log p_4$$

in our case we are stopping @ bi gram

$$\frac{1}{2}\log 0.83 + \frac{1}{2}\log 0.4$$

$$= 3\text{T}$$

Brevity Penalty (BP)

Candidate length (c): 6

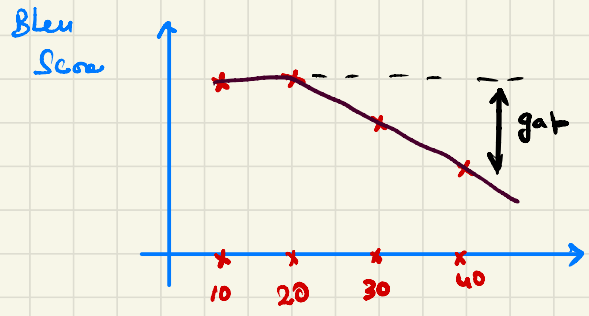
reference length (r): 7

$c > r$: $BP = 1$

else: $\exp\{1 - \frac{r}{c}\}$

$= \exp\{1 - \frac{7}{6}\} = e^{-1/6} = BP$

Bleu Score: $BP * \exp\left(\frac{1}{2}\sum_{i=1}^{2} \log p_i\right)$



Observation: as the sequence length increases Bleu score start decreasing.

Problem : Machine Translation.

$x_1, \ldots, x_m$ input sequence.

$\hat{y}_1, \ldots, \hat{y}_n$ Output sequence.

$n \neq m$

$\alpha_{1,1}$: how much attention should we give for 1st i/p word to generate 1st o/p word.

$\alpha_{2,1}$: for 1st i/p word to generate 2nd o/p word.

$\alpha_{i,j}$

to generate — i/p word.

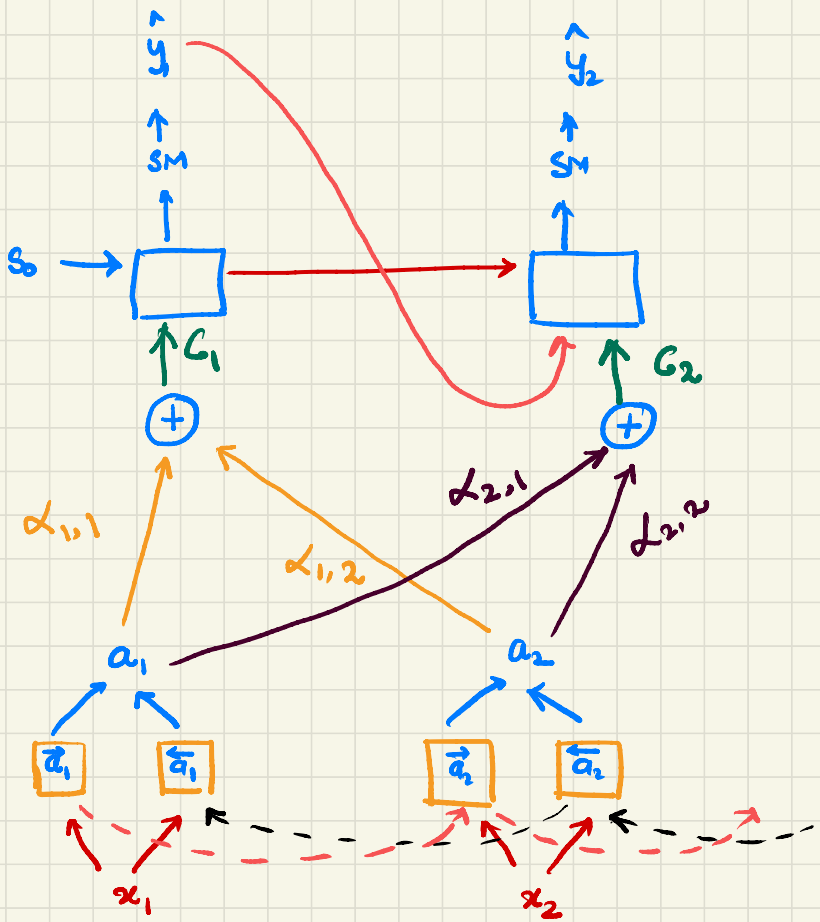
$a_t = (\overrightarrow{a_t}, \overleftarrow{a_t})$

input length : T_x

$$\sum_{t=1}^{T_x} \alpha_{1,t} = 1$$

$$c_1 = \sum_{t=1}^{T_x} \alpha_{1,t}\, a_t$$

$$= \alpha_{1,1} * a_1 + \alpha_{1,2} * a_2 \cdots$$



$\alpha_{1,2}$: amount of attention $\hat{y}_1$ should pay for x_2

$\alpha_{1,2}$:

a_2

s_1

$\alpha_{t,t'}$: amout of attention $\hat{y}_t$ should pay to the input $x_{t'}$

$s_{t'-1}$ → MLP → $e_{t,t'}$ (pre SM)

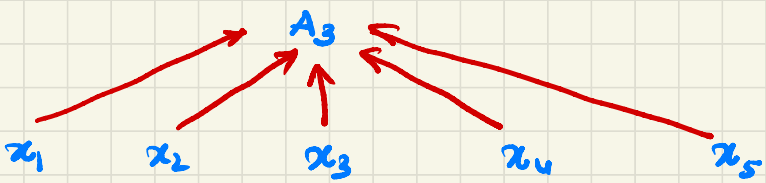
$a_{t'}$ →

$$\alpha_{t,t'} = \frac{\exp\{e_{t,t'}\}}{\sum_{t=1}^{Tx} \exp\{e_{t,t'}\}}$$

Transformer n/w architecture:

x_1, x_2, x_3, x_4, x_5 → input in word embedding.

A_1, A_2, A_3, A_4, A_5 → Attention.



		Query (Q)	Key (K)	Value (V)
x_1	A_1	q_1	k_1	v_1
x_2	A_2	q_2	k_2	v_2
x_3	A_3	q_3	k_3	v_3
x_4	A_4	q_4	k_4	v_4
x_5	A_5	q_5	k_5	v_5

q_i: is like a question

$q_3 \cdot k_1$: how good is 1st word for 3rd word's question.

$q_i \cdot k_j$: how good is the j^{th} word for i^{th} question.

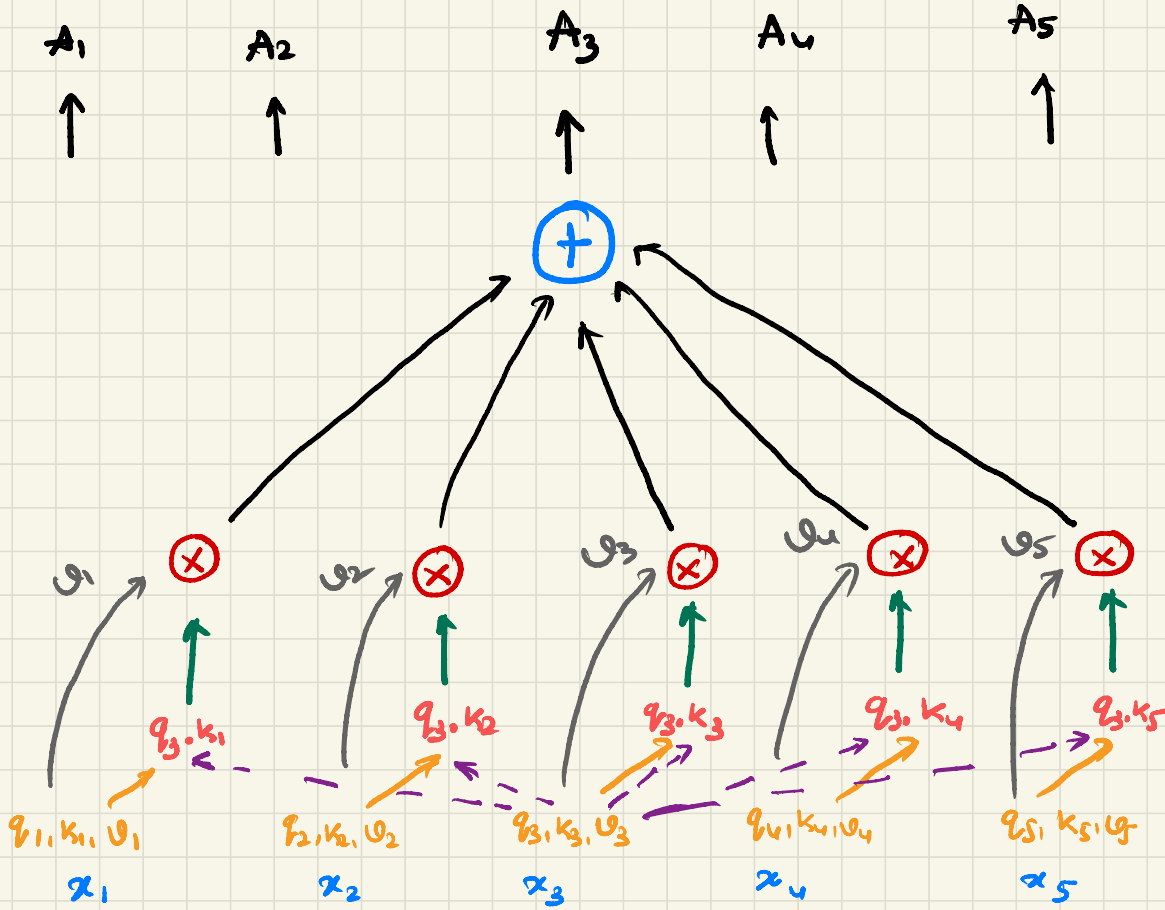
→ the aim of this operation is to get more info needed to help to get most useful info

$$A(q, K, V) = \sum_i \frac{\exp\{q \cdot k_i\}}{\sum_j \exp\{q \cdot k_j\}} \cdot v_i$$

Softmax

$q_1 \Rightarrow$

$$\frac{\exp\{q_1, k_1\}}{\sum_j \exp\{q \cdot k_j\}} \cdot v_1 + \frac{\exp\{q_1, k_2\}}{\sum_j \exp\{q_1 \cdot k_j\}} \cdot v_2 + \cdots$$


A1
A2
A3
A4
A5
v1
v2
v3
v4
v5
q3·k1
q3·k2
q3·k3
q3·k4
q3·k5
q1, k1, v1
q2, k2, v2
q3, k3, v3
q4, k4, v4
q5, k5, v5
x1
x2
x3
x4
x5

$q_i \in \mathbb{R}^{d_k}$

$k_i \in \mathbb{R}^{d_k}$

$v_i \in \mathbb{R}^{d_v}$

$x \in \mathbb{R}^{n \times d_{model}}$

n: num of example
d_{model}: dim of embedding

$$Q = X W_Q$$
$$(4 \times 70)\ (70 \times 50)$$

$$V = X W_V$$
$$(4 \times 70)\ (70 \times 60)$$

$$K = X W_K$$
$$(4 \times 70)(70 \times 50)$$

$d_k = 50$
$d_v = 60$

$n = 4$
$d_{model} = 70$

$x \in \mathbb{R}^{4 \times 70}$

$$\begin{bmatrix} -x_1- \\ -x_2- \\ -x_3- \\ -x_4- \end{bmatrix} = X \quad 4 \times 70$$

$$\begin{bmatrix} -q_1- \\ -q_2- \\ -q_3- \\ -q_4- \end{bmatrix} = Q \quad 4 \times 50$$

$$\begin{bmatrix} | & | & | & | \\ k_1 & k_2 & k_3 & k_4 \\ | & | & | & | \end{bmatrix}$$

K^T

$$\begin{bmatrix} -q_1- \\ -q_2- \\ -q_3- \\ -q_4- \end{bmatrix} \begin{bmatrix} | & | & | & | \\ k_1 & k_2 & k_3 & k_4 \\ | & | & | & | \end{bmatrix}$$

$$= \begin{bmatrix} q_1 k_1 & q_1 k_2 & q_1 k_3 & q_1 k_4 \\ q_2 k_1 & q_2 k_2 & q_2 k_3 & q_2 k_4 \\ q_3 k_1 & q_3 k_2 & q_3 k_3 & q_3 k_4 \\ q_4 k_1 & q_4 k_2 & q_4 k_3 & q_4 k_4 \end{bmatrix} = Q \cdot K^T$$

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{Q k^T}{\sqrt{d_k}}\right) V$$

Scaled dot Product attention