Date : 26/7/24
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☑ ☐ ☐

COURSE 5 – SEQUENCE MODELS

Sequence data:

Speech recog., music gen^{r}, sentiment classif; DNA anal; m/c transl; video activity recog; etc; named-entity recogn.

Notation

eg. X: Harry Potter and Hermoine Granger invented a new spell. $T_x = 9$

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<7>}$ $x^{<8>}$ $x^{<9>}$

y: 1 1 0 1 1 0 0 0 0 $T_y = 9$

$y^{<1>}$ $y^{<3>}$

$x^{<t>} \Rightarrow t^{th}$ element o/p $T_x^{(i)}$

$x^{(i)<t>} \Rightarrow t^{th}$ elem. in sequence of i^{th} training example.

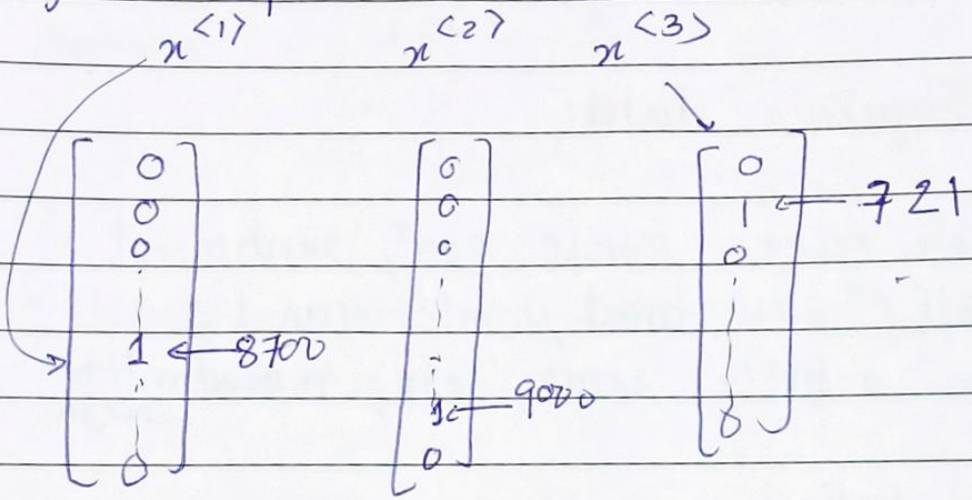
→ A dictionary of for vocabulary of around 50k words is used.

↓

The sequence words are 1-hot encoded.

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

eg. Harry Potter and ...

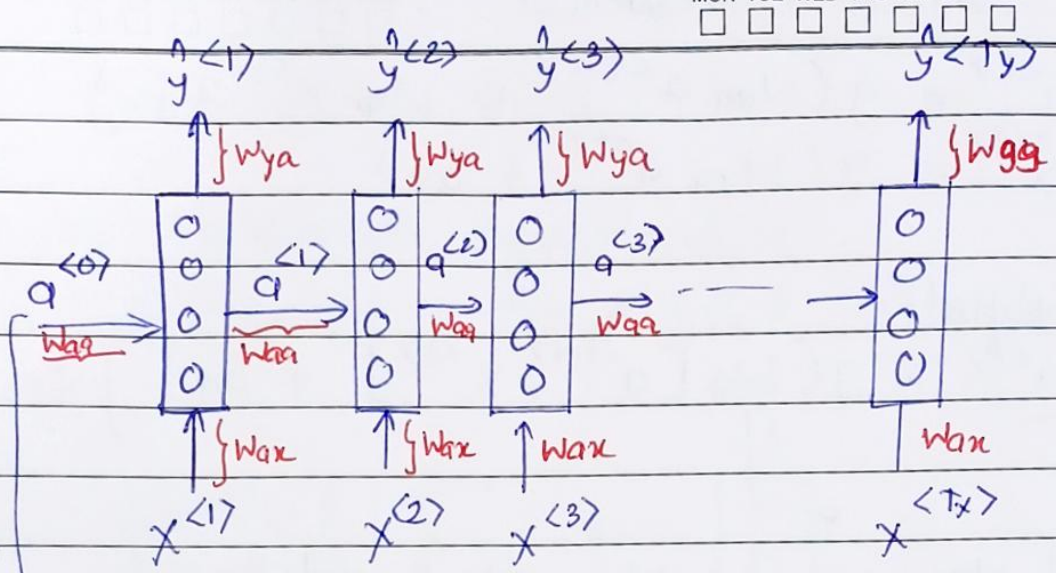


9 such vectors created if $T_x = 9$.

Problem with standard network:
- I/P & O/P can be diff. length in diff eg.
- Doesn't share features learned across diff. positions of text
(If "Harry" comes & is recog. as name then further anywhere "Harry" comes it should recognize it as name.)

Recurrent Neural Networks.
E.

Date :
MON TUE WED THU FRI SAT SUN



Vector of Zeros.

Unidirectional, $T_x = T_y$.

Prediction at a certain time uses info from inputs only EARLIER in sequence. not LATER.

eg. He said, "Teddy Roosevelt is President".
He said, "Teddy bears are cute.".

→ Forward Propagation ←

$$a^{\langle 0 \rangle} = \vec{0}$$

$$a^{\langle 1 \rangle} = g_1\left(W_{aa}\, a^{\langle 0 \rangle} + W_{ax}\, x^{\langle 1 \rangle} + b_a\right)$$

↑ gen^ly tanh / ReLU

$$\hat{y}^{\langle 1 \rangle} = g_2\left(W_{ya}\, a^{\langle 1 \rangle} + b_y\right)$$

↑ gen^ly sigmoid.

Parameters W_a, b_a, W_y, b_y will remain same for all elements of sequences.

Date : .....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

$$a^{<t>} = g(W_{aa} a^{<t-1>} + W_{ax} x^{<t>} + b_a)$$
$$\hat{y}^{<t>} = g(W_{ya} a^{<t>} + b_y)$$

Simplified:

$$a^{<t>} = g(W_a [a^{<t-1>}, x^{<t>}], + b_a)$$

$$[W_{aa} \mid W_{ax}] = W_a \quad \text{(stack side by side)}$$

$$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} \quad \text{(stack on top of other)}$$

$$\hat{y}^{<t>} = g(W_y a^{<t>} + b_y)$$

→ Backprop ← "Backprop through time"

Loss for a single word:

$$\mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -y^{<t>} \log \hat{y}^{<t>} - (1-y^{<t>}) \log (1-\hat{y}^{<t>})$$

Loss for full sequence:

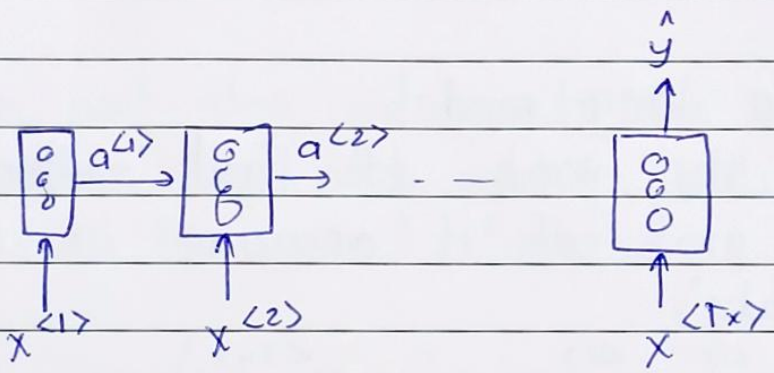
$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_x} \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

$T_x \neq T_y$ RNN

eg. Sentiment classification.

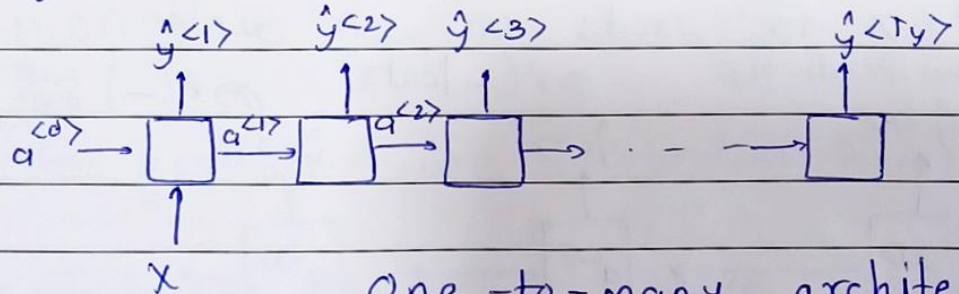
X = text y = ★ 1 2 3 4 5



Many-to-one architecture.

eg. Music generation.

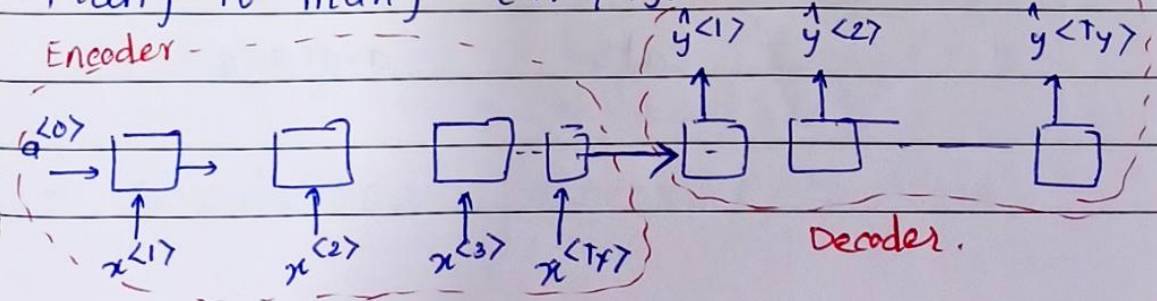
$y \rightarrow y^{<1>}\ y^{<2>} \cdots y^{<T_y>}$



One-to-many architecture.

eg. Previous page's Many-to-many ($T_x = T_y$)

Many-to-many ($T_x \neq T_y$)



eg. Machine Translation.

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

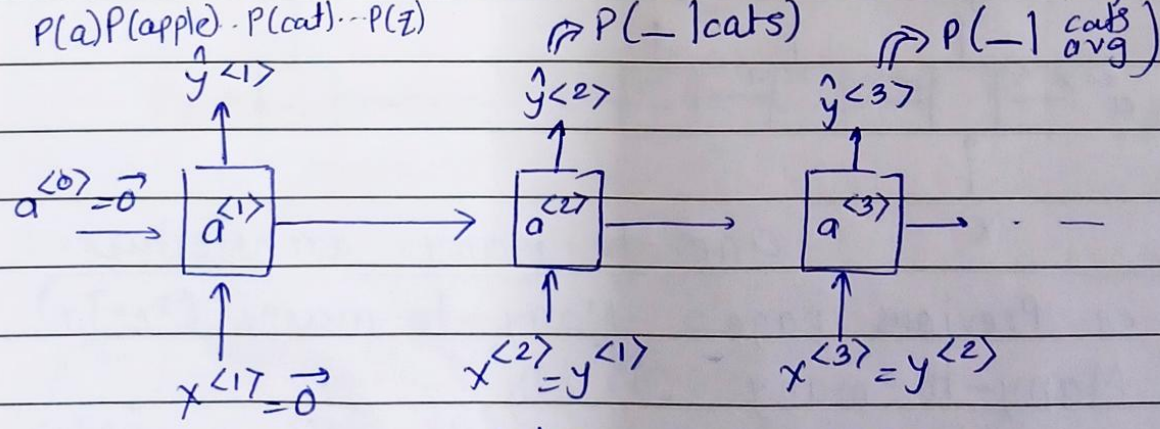
Language Modelling

eg In speech recog:
"The apple and pair salad"
"The apple and pear salad".

Work of a lang. model:
What is the prob. of that sentence (meaning prob of it occuring in some data).
i.e. $P(y^{<1>}, y^{<2>}, \ldots, y^{<T_y>})$

→ Tokenize ←

<EOS> added @ end.
<UNK> for words not in dictionary.

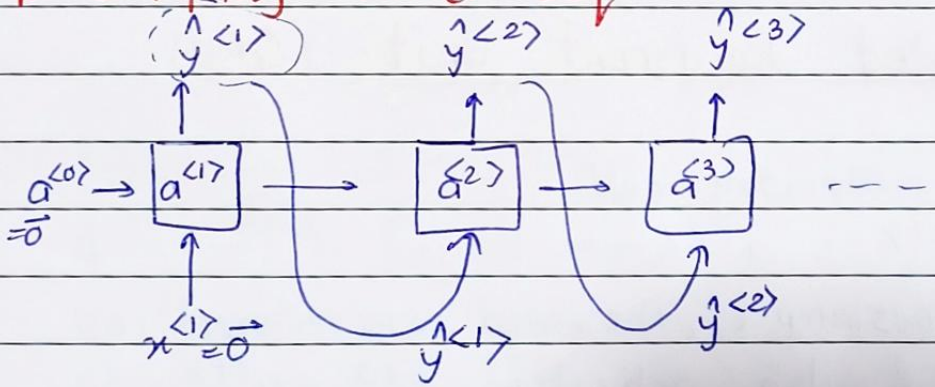


Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

$\hat{y}^{<1>}$ will be softmax prediction of 'what is the chance that first word is some word a--z from dict' or <EOS> or <UNK>'.

In next step, we provide the correct word 'cats' and make prediction $\hat{y}^{<2>}$ given the word 'cats'.

Sampling Novel Seq.



- For first word use np.random.choice.

- when u get <EOS> token, stop.

character-level lang. model

Vocabulary = [a, b, c --, z, ␣, ., ;, i, 0-9, A-Z --]

* Computationally expensive to train.

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

Vanishing & Exploding gradients

eg. The cat , ——— , was sleeping.
The cats , ——— , were sleeping

very long-term dependency

Weakness of a basic RNN Algo: bcz difficult for backprop to update earlier weights caused by last layers.

Gated Recurrent Unit: (GRU)

C = Memory cell.
$c^{<t>} = a^{<t>}$

@ timestamp t, the word was stored in $c^{<t>}$ & also activation o/p $= a^{<t>}$

$\tilde{c}^{<t>}$ = candidate for replacing $c^{<t>}$
$= \tanh\left(W_c \left[c^{<t-1>}, x^{<t>}\right] + b_c\right)$

Γ_u = Update gate

$\Gamma_u = \sigma\left(W_u \left[c^{<t-1>}, x^{<t>}\right] + b_u\right)$
↳ assume always either 0 or 1.
bcz for almost majority values $\sigma = 0$ or 1

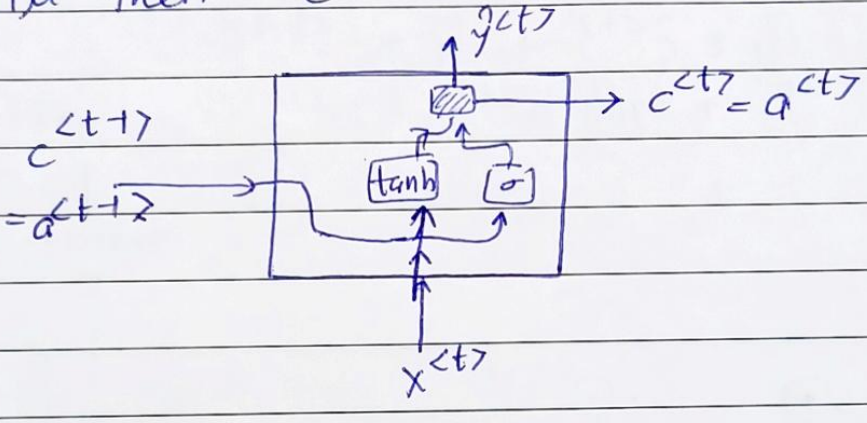
elem-wise

Date : .....
MON TUE WED THU FRI SAT SUN

$$c^{\langle t \rangle} = \Gamma_u * \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle}$$

Job of update gate : to decide whether ~~re~~ update the $c^{\langle t \rangle}$ value with $\tilde{c}^{\langle t \rangle}$ or not.

Till then $c^{\langle t \rangle} = c^{\langle t-1 \rangle}$ (memorize)



$c^{\langle t \rangle}$, $\tilde{c}^{\langle t \rangle}$, Γ_u all same dimensions.

We can assign ~~some~~ a bits for storing singular/plural, food, etc.

$$\tilde{c}^{\langle t \rangle} = \tanh\left(W_c\left[\Gamma_r * c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$$

$$\Gamma_u = \sigma\left(W_u\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_u\right)$$

$$\Gamma_r = \sigma\left(W_r\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_r\right)$$

$$c^{\langle t \rangle} = \Gamma_u * \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle}$$

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

LSTM (Long Short Term Memory)

$$\tilde{c}^{\langle t \rangle} = \tanh\left(W_c \left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$$

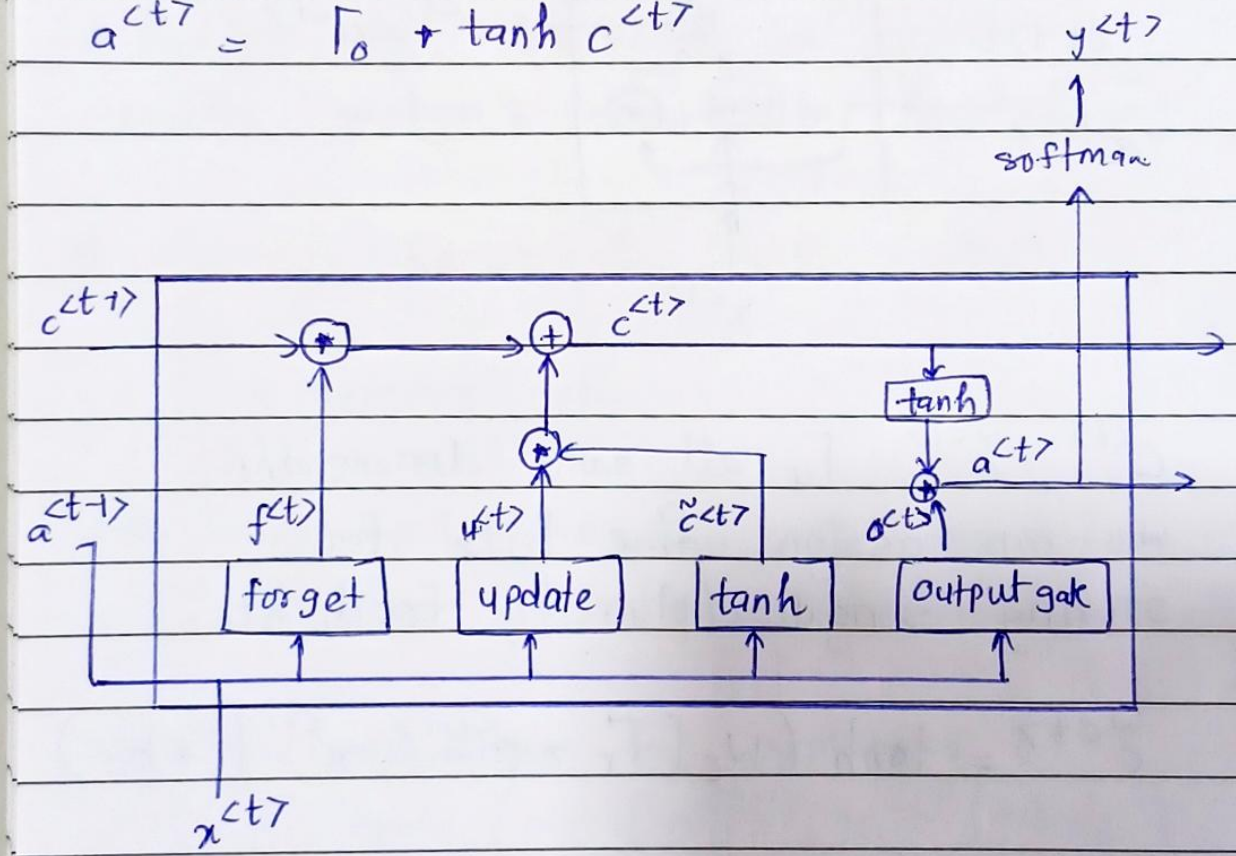
$$\Gamma_u = \sigma\left(W_u \left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_u\right)$$

$$\Gamma_f = \sigma\left(W_f \left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_f\right)$$

$$\Gamma_o = \sigma\left(W_o \left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_o\right)$$

$$c^{\langle t \rangle} = \Gamma_u * \tilde{c}^{\langle t \rangle} + \Gamma_f * c^{\langle t-1 \rangle}$$

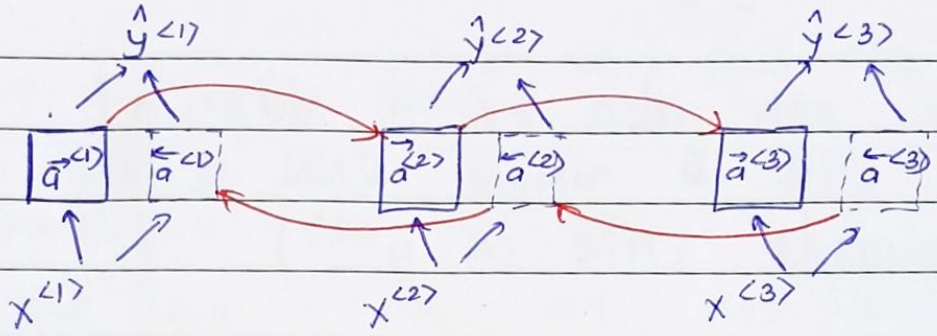
$$a^{\langle t \rangle} = \Gamma_o * \tanh c^{\langle t \rangle}$$



LSTM prev. invented, more computational, but more powerful than GRU

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

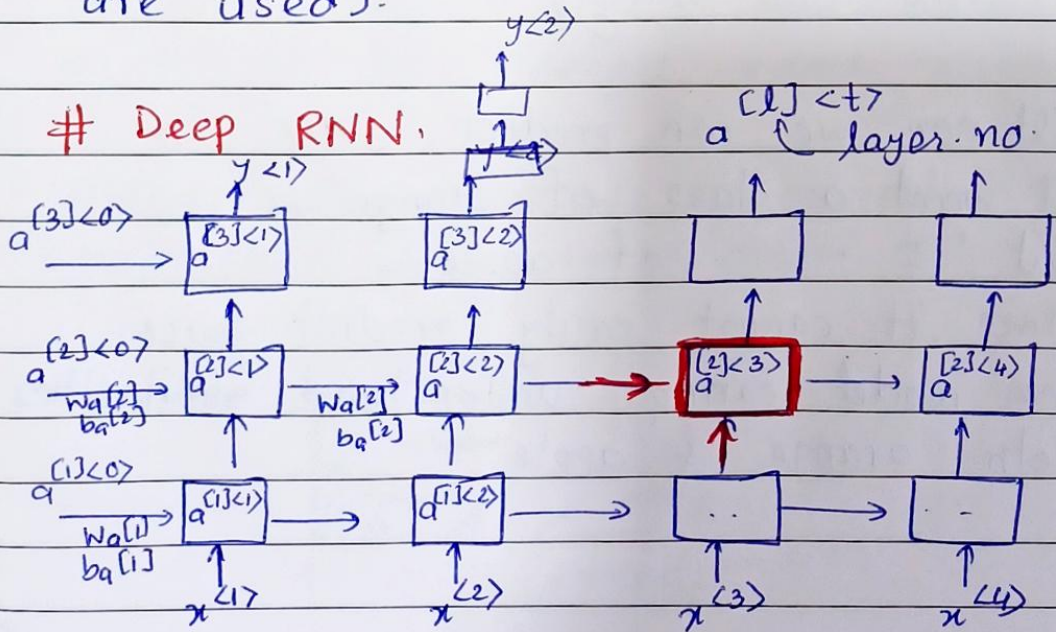
Bidirectional RNN



Acyclic Graph ↖ All this is just for prop.

These indiv. blocks can be LSTM or GRU
Disadv of BRNN: You need the entire sequence of data before making any predictions. eg. In speech recogn, u have to wait till u complete sequence. (Actually, much more complex models are used).

Deep RNN.



Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

$$a^{[2]<3>} = g\left(W_a^{[2]}\left[a^{[2]<2>}, a^{[1]<3>}\right] + b_a^{[2]}\right)$$

There can also be a mix of deep RNN & simple RNN or NN afterwards (like in $y^{<2>}$)

Word Representation.

Word embeddings - lets algo. automatically understand analogies like man-woman, king-queen.

O_{5391} = Vocabulary

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \rightarrow \text{Man}(5391)$$

Till now we can predict 'juice' in 'I want a glass of orange ___'.
But 'I ... ___ of apple ___'
Here, it cannot easily predict juice.
The model cannot understand similarities betn orange & apple.

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

→ Featurized repr – Word embedding –

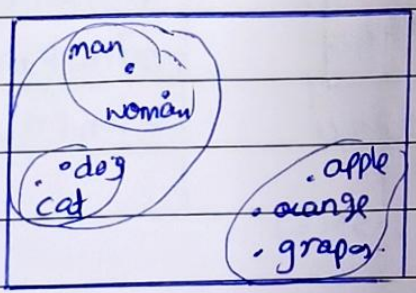
	Man (5391)	Woman (9853)	King	Queen	Apple	Orange
Gendr	−1	1	−0.95	0.97	0	0.01
Royal	0.01	0.02	0.93	0.95	−0.01	0.0
Age						
Food						
Size						
cost						
Verb						
Noun						
etc						

e_{5391} e_{9853}

eg. 300 such features.

So, now Apple & Orange will have much of the features same

Using t-SNE algo, we can convert this 300D vectors to 2D plots.



Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

eg. use in named entity recog.

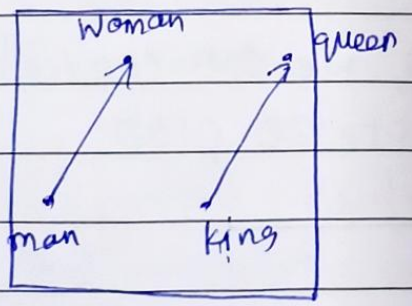
Sally Johnson is an orange farmer.
↳ Train.

Robert Lin is an apple farmer.
↳ Test

1. Learn word emb. from large text corpus.
2. Transfer emb to new task with smaller training set.
3. Continue to finetune w.f with new data.

$e_{man} - e_{woman} \approx \begin{bmatrix} -1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$

$e_{king} - e_{queen}$



300D space points
(not on t-SNE)

Find word w such that maximize similarity
$sim(e_w, e_{king} - e_{man} + e_{woman})$

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

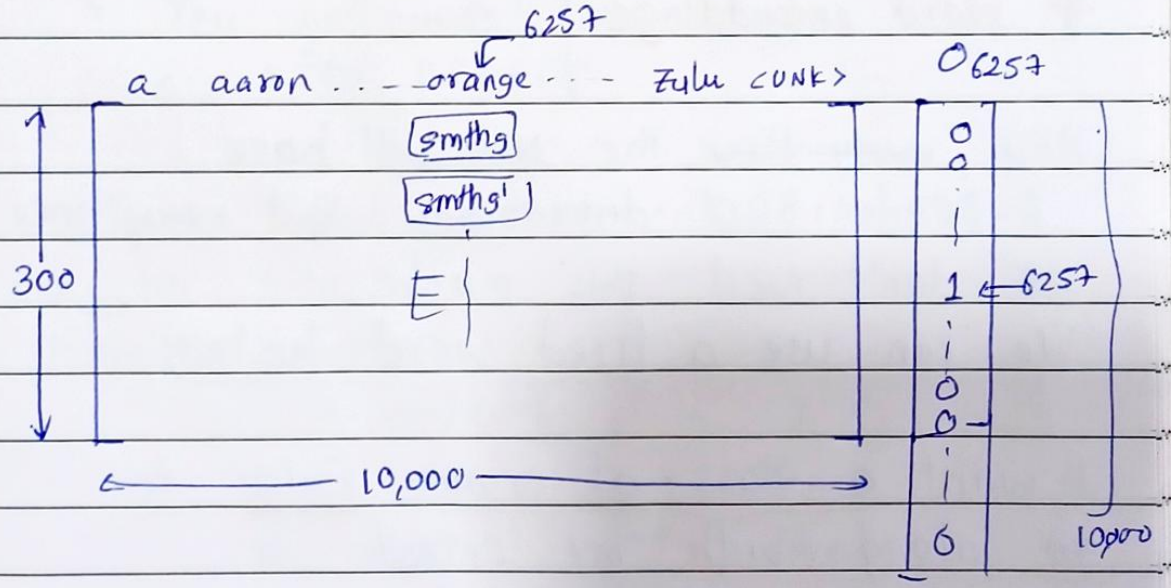
→ Cosine similarity ←

↳ commonly used.

$$sim(e_w, e_{king} - e_{man} + e_{woman})$$

$$sim(u, v) = \frac{u^T \cdot v}{\|u\|_2 \|v\|_2}$$

→ Embedding Matrix. ←

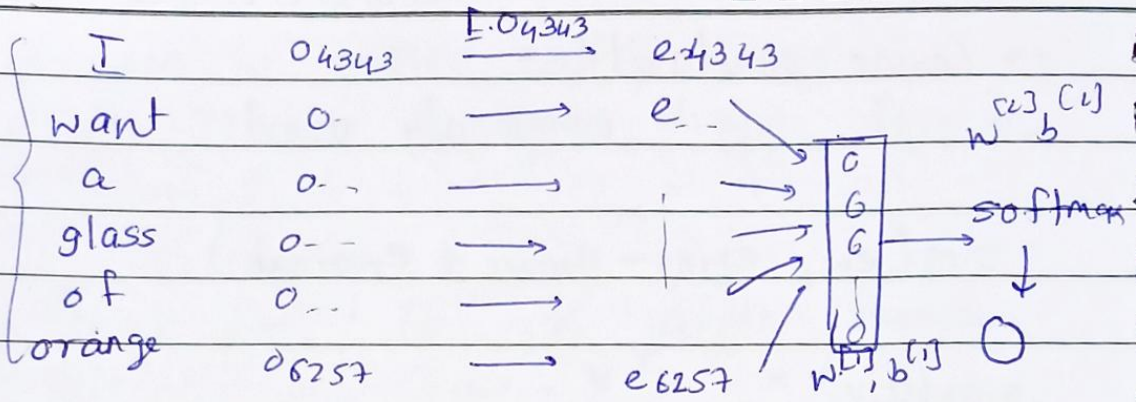


$E \cdot O_{6257} = e_{6257}$

$(300, 10K) \rightarrow (10K, 1)$

$\begin{bmatrix} smthg \\ smthg' \\ \vdots \end{bmatrix}$ $(300, 1)$

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐



10k possibilities for softmax's O/P

Word embeddings

Here, ~~every time~~ the NN will have $6 \times 300 = 1800$ dimensions. But every time we don't need this much.
We can use a fixed word history

I want a glass of orange juice to go along with my cereal.

eg. of context: Last 4 words, 4 words on L & R, Last 1 word, nearby 1 word

Date : 28/7/24
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☑

Word2Vec :

* Embeddings (feature vectors for each & every word) are not hand crafted, instead they are learnt using neural network training.

* 1. Take a fake problem → like find missing word
  2. Solve it using NN.
  3. You get word embeddings as a SIDE EFFECT

* There lived a king called Ashoka.

eg. for step 2.

Training samples:

X (context)		y
lived, a	→	There
a, king	→	lived
king, called	→	a
⋮		⋮

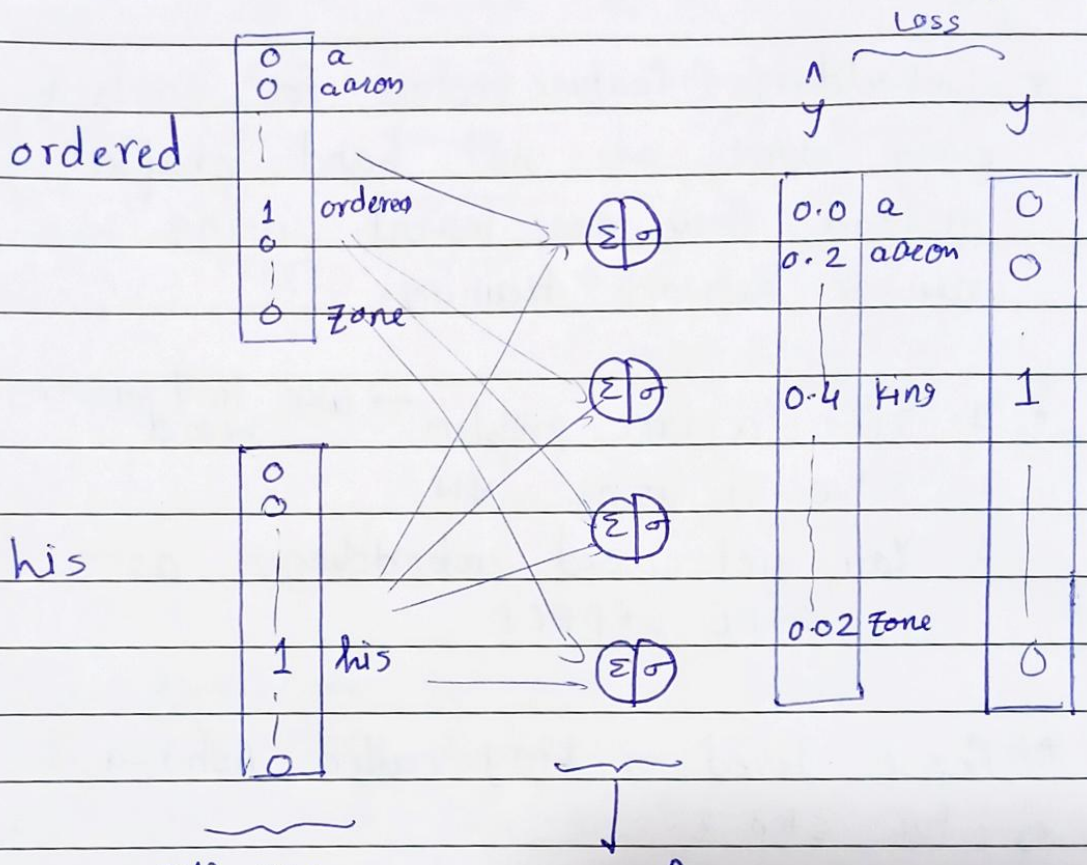
This is called semi-supervised learning

CBOW: Continuous Bag of Words

target — context

Date : .....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

king ordered his ← Backprop.



I/P one hot vectors.

no. of features we want (hyperparameter)

After training the model over all examples, the o/p vector for 'king' would be very similar to that of 'emperor'.

Skip-Gram: king (context) ordered his (target)

↓ just reverse the above NN.

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

For skip-gram:

$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10k} e^{\theta_t^T e_c}}$$

→ Negative Sampling ←

Is there any relation bet^n context word & another word.

	c (x)	word(t) (x)	y? (y)
	orange	juice	1
k	orange	king	0
	orange	emperor	0

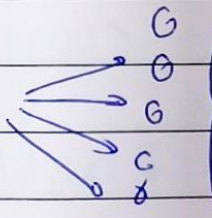
k = 5-20 smaller dataset

k = 2-5 larger dataset

$$p(y=1|c,t) = \sigma(\theta_t^T \cdot e_c)$$

~~We~~

$O_{6257} \rightarrow E \rightarrow e_{6257}$



vocabulary. 10K

Now instead ~~of~~ outputting full vocabulary predictions, we make it to predict on the +ve sample & k -ve samples deliberately.

⇒ like a 10,k binary classification prblm. split into 4-5 binary class everytime.

Date :..............................
MON TUE WED THU FRI SAT SUN

?? How to choose -ve samples ??

1> sample most freq used words.

or

2> sample uniformly from vocabulary

or

3> $P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10k} f(w_j)^{3/4}}$

$f(w_i)$ = observed freq. of a particular word in English.

GLoVe (Global vectors for word repr)

X_{ij} = # times j appears in context of i.

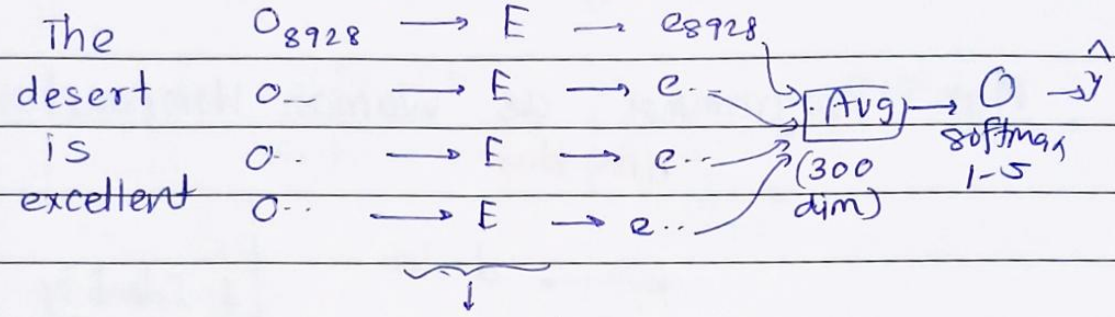
(i ← c, j ← t)

⇒ i & j kitne baar agal-bagal mein or within some ± x words ke range me aatein hai. ⇒ CLOSE PROXIMITY

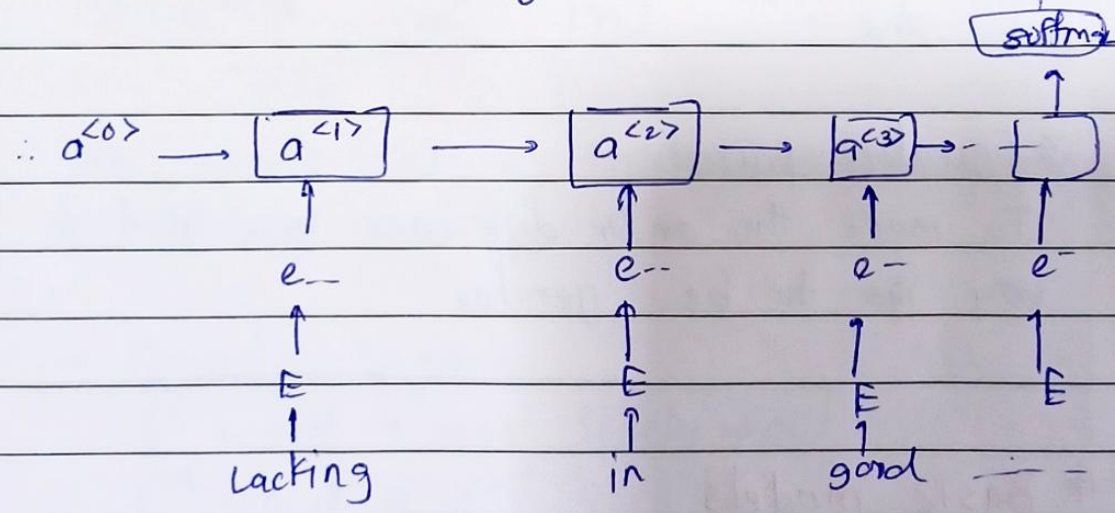
Minimize: $\sum_{i=1}^{10k} \sum_{j=1}^{10k} f(x_{ij}) \left(\theta_i^T e_j + b_i + b_j' - \log x_{ij} \right)^2$

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

Sentiment Classification



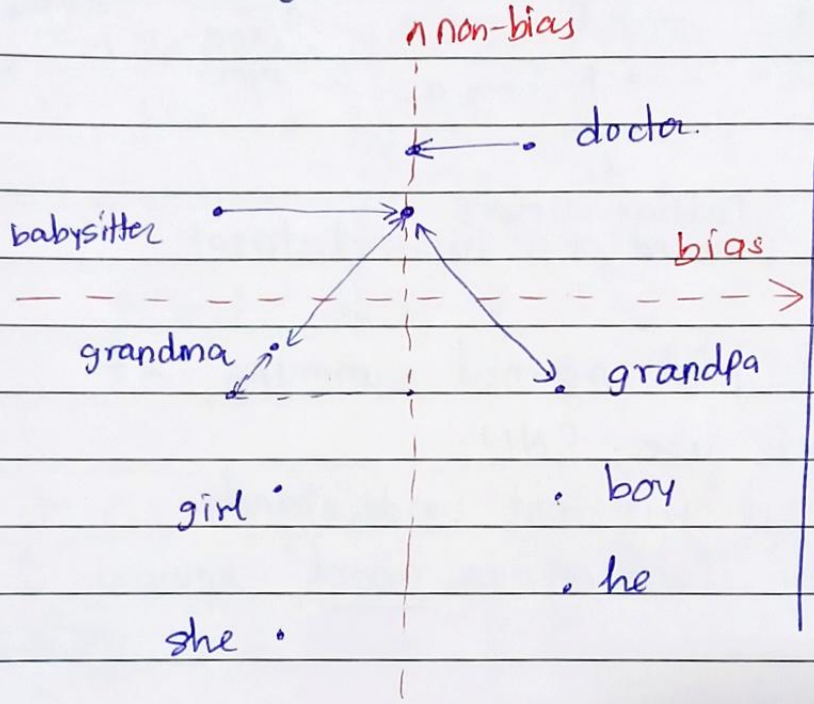
Instead of just normal summing of vectors, we use RNN.
Bcz o/w it will not understand:
"Completely lacking in good" review.



Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

Problem of bias

Man : Programmer as Woman : Homemaker



1- Identify bias direction

2. Neutralize

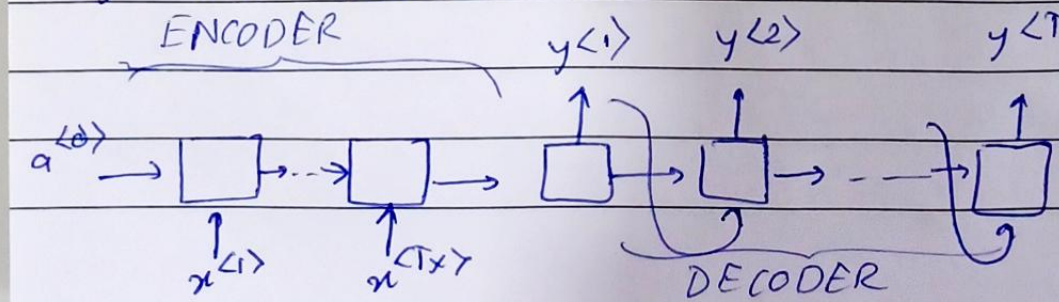
Project to get rid of bias.

3. Equalize pairs

To make the only difference b/w girl & boy ~~as~~ to be 'gender'.

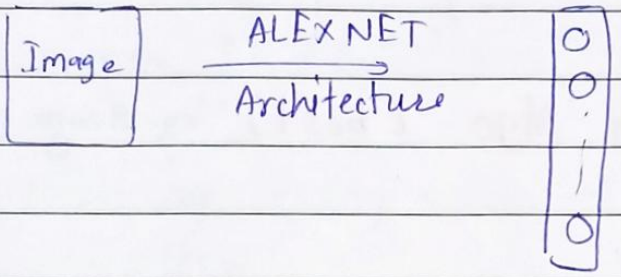
Basic models

eg. M/c translation.



Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

eg. Image captioning



→ Finding Most Likely ←

- Jane is visiting Africa in september.
- Jane is going to be visiting ... —
- In sept, Jane will visit Africa

$$P(y^{<1>}, \ldots, y^{<T_y>} \mid x)$$

$y^{<1>}, \ldots, y^{<T_y>}$: Eng

x : French.

We want the y that maximizes conditional prob

→ Greedy Search:

1st word → most likely word

2nd word → 2nd " "

But this approach might output

Jane is going to be - —

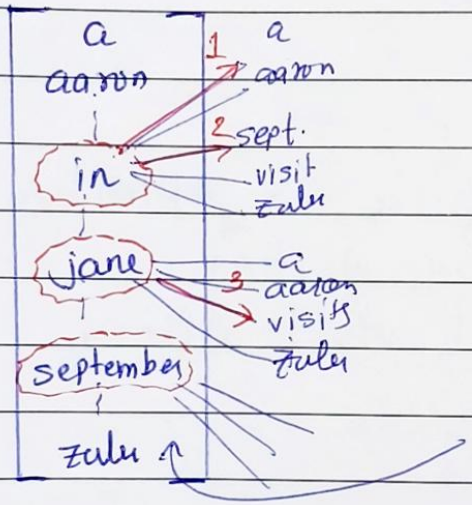
Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

→ Approximate Search:

$$\arg\max P(\hat{y}^{<1>} \dots \hat{y}^{<T_y>} | x)$$

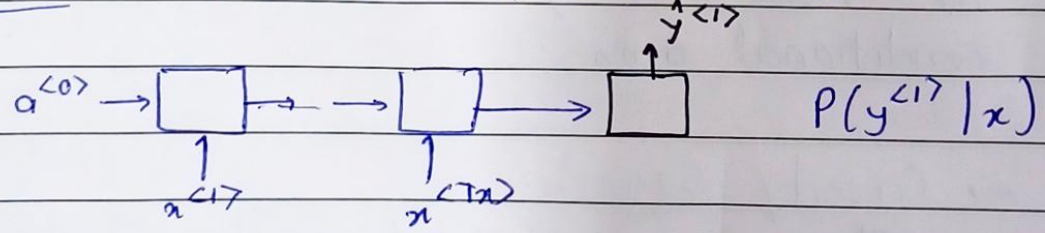
→ Beam Search Algo (best) eg. Beam width B → 3

~~Step I~~



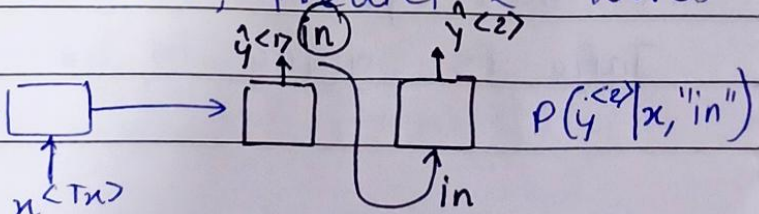
No top 3 ~~in~~ from september, so it is dropped.

Step-I: Try to evaluate prob. of 1st word.



B=3 ⇒ will predict top 3 such words.

Step-II: By hardwiring each of the previous 3 words as 1st word, predict 2nd word.



Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

$$P(y^{<1>}, y^{<2>} | x) = P(y^{<1>} | x) \times P(y^{<2>} | x, y^{<1>})$$

29/7/24

$$\arg\max_{y} \prod_{t=1}^{T_y} P(y^{<t>} | x, y^{<1>} \dots y^{<t-1>})$$

multiplying many such <1 nos. = very low num.
⇒ underflow

$$\therefore \arg\max_{y} \sum_{y=1}^{T_y} \log P(y^{<t>} | x, y^{<1>} \dots y^{<t-1>})$$

↑

But still these probs will be negative overall. ⇒ Algo will try for shorter translations.

$$\therefore \left[\frac{1}{T_y^{\alpha}} \sum_{t=1}^{T_y} \log P(y^{<t>} | x, y^{<1>} \dots y^{<t-1>}) \right]$$

$0 < \alpha < 1$ Hyperparam

Bleu Score:

Bilingual Evaluation Understudy

Use:

If there are multiple great answers.

Date : 29/7/24

MON TUE WED THU FRI SAT SUN

☑ ☐ ☐ ☐ ☐ ☐ ☐

Ref 1: The cat is on the mat

Ref 2: There is a cat on the mat

MT O/P: The cat the cat on the mat

Bigrams	Count	Clip count
① the cat	2	1
② cat the	0	0
③ cat on	1	1
④ on the	1	1
⑤ the mat	1	1

$\frac{4}{6}$

$$P_n = \frac{\sum_{n\text{-grams} \in \hat{y}} \text{Count clip (n-gram)}}{\sum_{n\text{-grams } \hat{y}} \text{Count (n-gram)}}$$

P_n → n-grams

Combined BLEU Score $= BP \cdot \exp\left(\frac{1}{4} \sum_{n=1}^{4} P_n\right)$

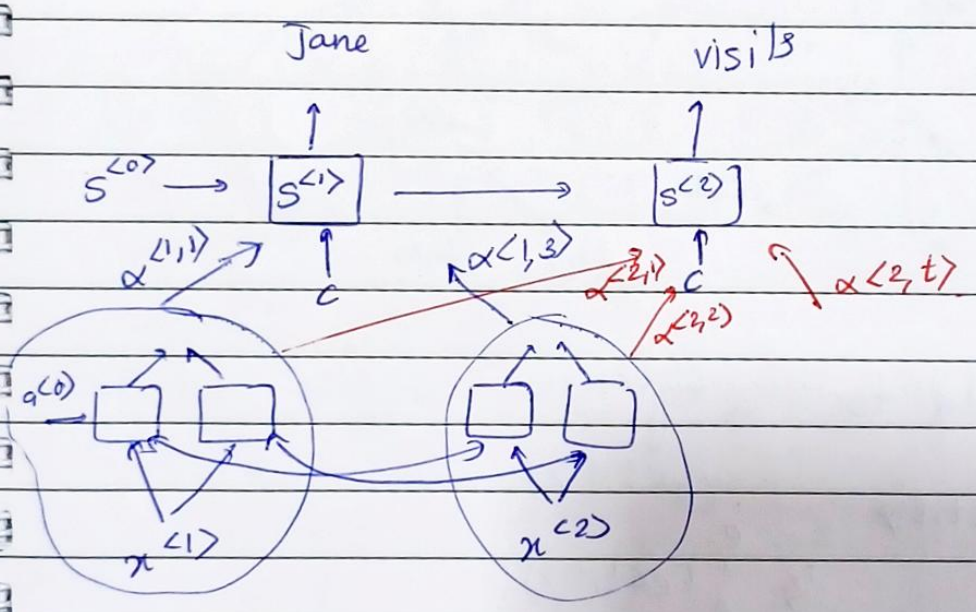
Brevity penalty $= \begin{cases} 1 & \text{if MT O/P length} > \text{ref. O/P length} \\ \exp\left(1 - \frac{\text{ref. O/P length}}{\text{MT O/P length}}\right) & \text{otherwise} \end{cases}$

Brevity penalty → penalizes shorter translations

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

Attention

The Encoder - Decoder architecture cannot handle long sentence.
Encoder cannot generate a vector summarizing the full sentence @ a time
∴ We need an attention region.



α <,> are weights / attn wts.

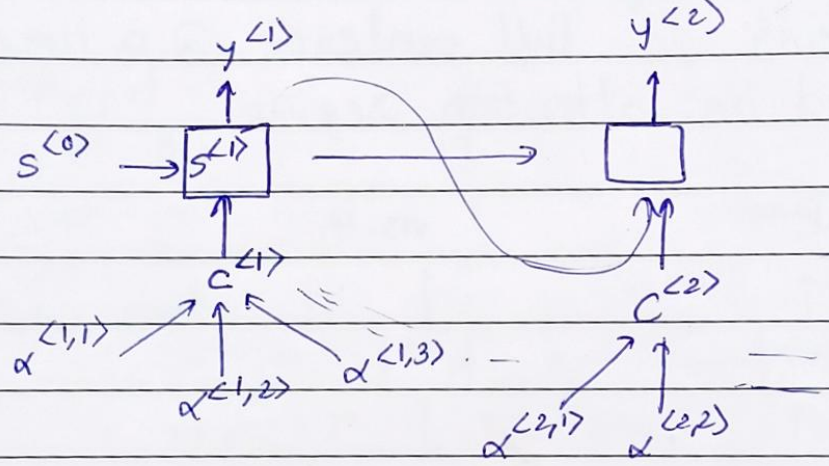
Model: $a^{<t'>} = (\overrightarrow{a}^{<t'>}, \overleftarrow{a}^{<t'>})$
Concatenation of both.

$\alpha^{<t,t'>}$ = amt of attention $y^{<t>}$ should pay to $a^{<t'>}$

Date :
MON TUE WED THU FRI SAT SUN

$\sum_{t'} \alpha^{\langle 1,t' \rangle} = 1$ Sum of all attentions for a particular $y^{\langle t \rangle} = 1$

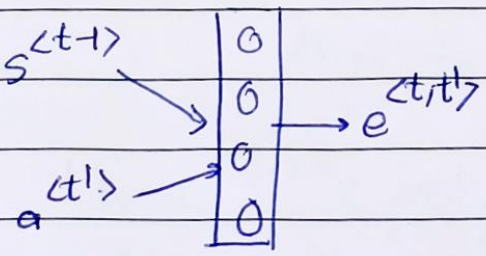
$c^{\langle 1 \rangle} = \sum_{t'} \alpha^{\langle 1,t' \rangle} a^{\langle t' \rangle}$



Formula for $\alpha^{\langle t,t' \rangle}$

$$\alpha^{\langle t,t' \rangle} = \frac{e^{(e^{\langle t,t' \rangle})}}{\sum_{t'=1}^{T_x} e^{(e^{\langle t,t' \rangle})}}$$

To compute $e^{\langle t,t' \rangle}$:



Train a very small NN
→ Takes quadratic time $T_x \times T_y$ but acceptable

previously, phonemes were used for sp-recog.

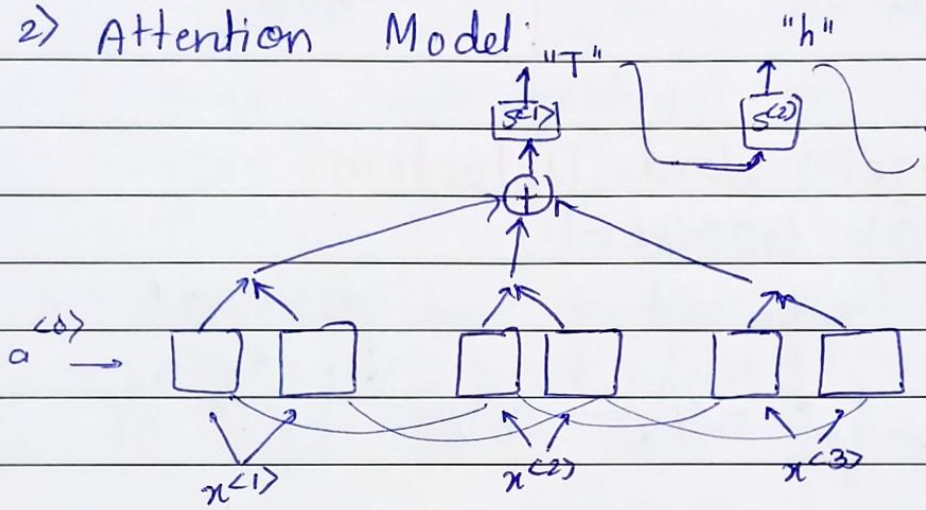
Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

Speech Recognition:

1) CTC cost for speech recogn.
Connectionist Temporal Classif.

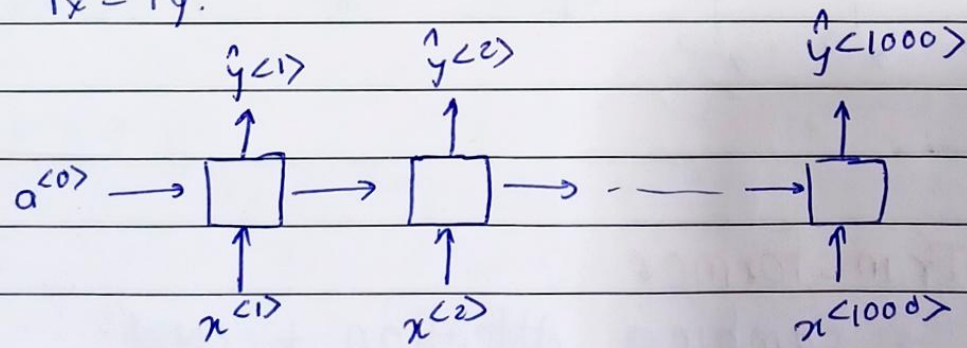
(OR)

2) Attention Model:



1) CTC: Simple RNN with GRU/LSTM.

$T_x = T_y$.



10 sec. audio
Feature rate = 100Hz
⇒ Total of 1000 inputs.

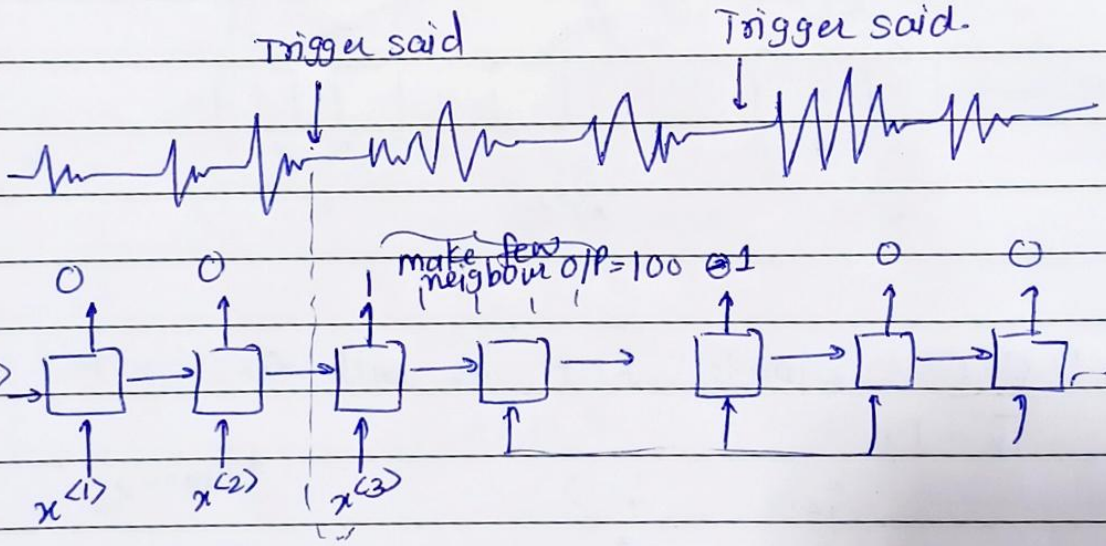
Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

CTC generates o/p like:

ttt _ _ _ h _ eee _ _ _ ␣ _ _ _ qqq _ _

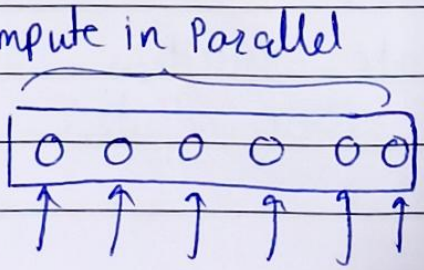
blank ↑ space ↑

Trigger Word Detection.
(OK GOOGLE!)



Transformer

↳ Combines Attention + CNN style of processing



Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

→ Self-attention ←

Compute Attn based representatn for each I/P word:

$A(q, k, V)$

eg.	$x^{<1>}$	$x^{<2>}$	$x^{<3>}$	$x^{<4>}$	$x^{<5>}$
	Jane	visite	l'Afrique	en	septembre
	$A^{<1>}$	$A^{<2>}$	$A^{<3>}$	$A^{<4>}$	$A^{<5>}$

Depending on how we are thinking of Africa → historical interest?, holiday dest.?, etc.

↓

we will choose to represent it differently

$A^{<3>}$ will look at the surrounding words to get context.

$$A(q, K, V) = \sum_i \frac{\exp(q \cdot k^{<i>})}{\sum_j \exp(q \cdot k^{<j>})} v^{<i>}$$

softmax-like

Each input has $q^{<t>}$, $k^{<t>}$ & $v^{<t>}$.

$$q^{<3>} = W^q \cdot x^{<3>}$$
$$k^{<3>} = W^k \cdot x^{<3>}$$
$$V^{<3>} = W^V \cdot x^{<3>}$$

↳ PARAMETER MATRICES

$x^{<3>}$ = word emb. of $x^{<3>}$ l'Afrique

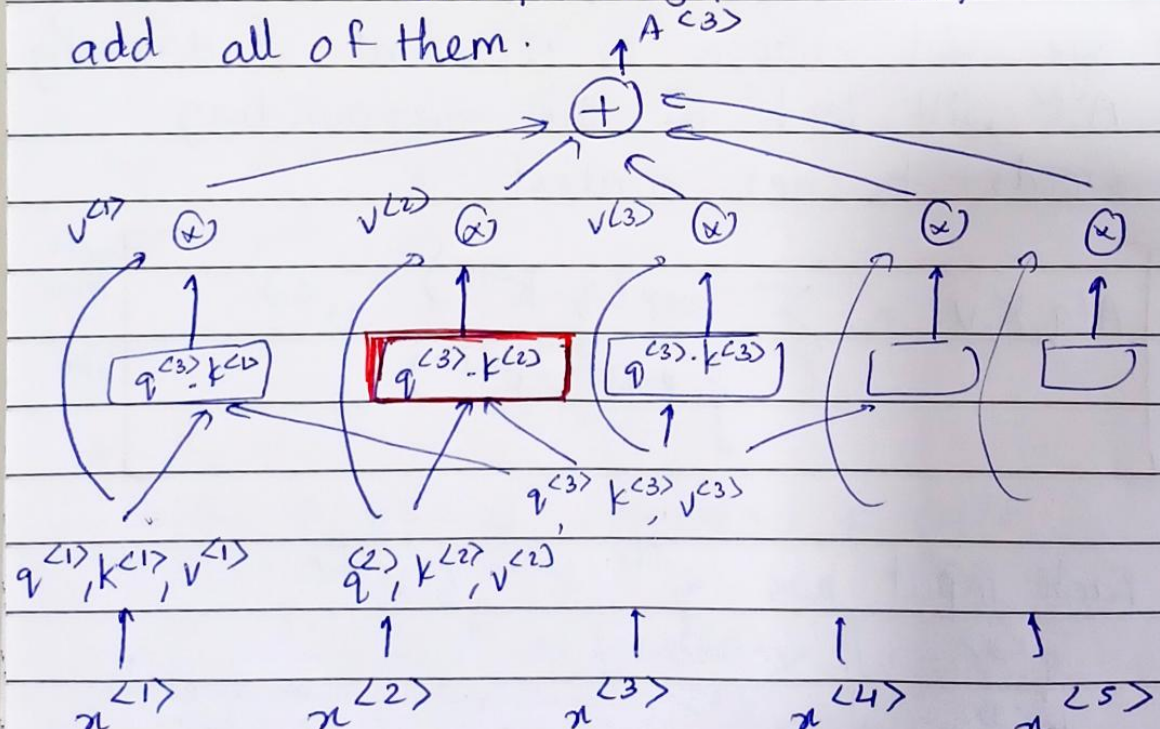
Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

$q^{<3>}$ is a question you can ask to l'Afrique like : "what's happening there?"

↳ Answers are $q^{<3>} \cdot k^{<1>}$, $q^{<3>} \cdot k^{<2>}$
$q^{<3>} \cdot k^{<3>}$ . . . . dot prod

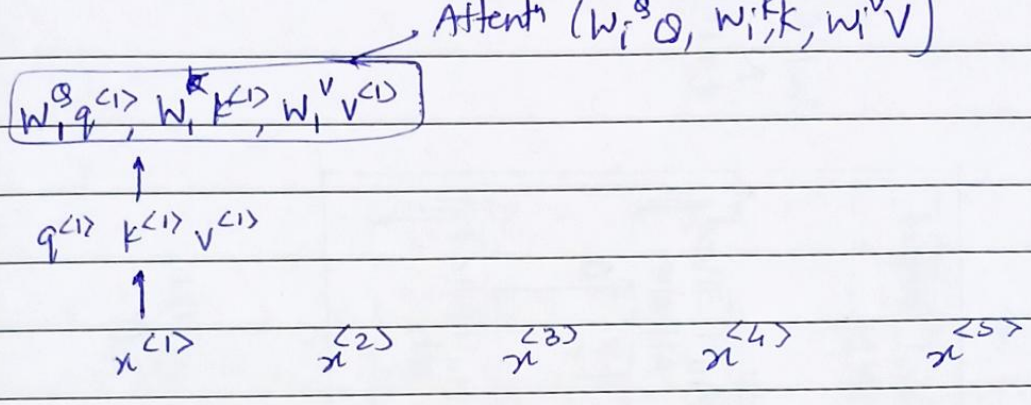
If $k^{<1>}$ was repr. a person & $k^{<2>}$ an action, then $q^{<3>} \cdot k^{<2>} > q^{<3>} \cdot k^{<1>}$

Take softmax of all $q^{<>} \cdot k^{<>}$ & multiply them with values correspondingly. Finally add all of them.



Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

→ Multi-Head Attention ←



eg. for head = 1 → we know whats happening.

For head = 2 → we know when happening?

For head = 3 → who did?

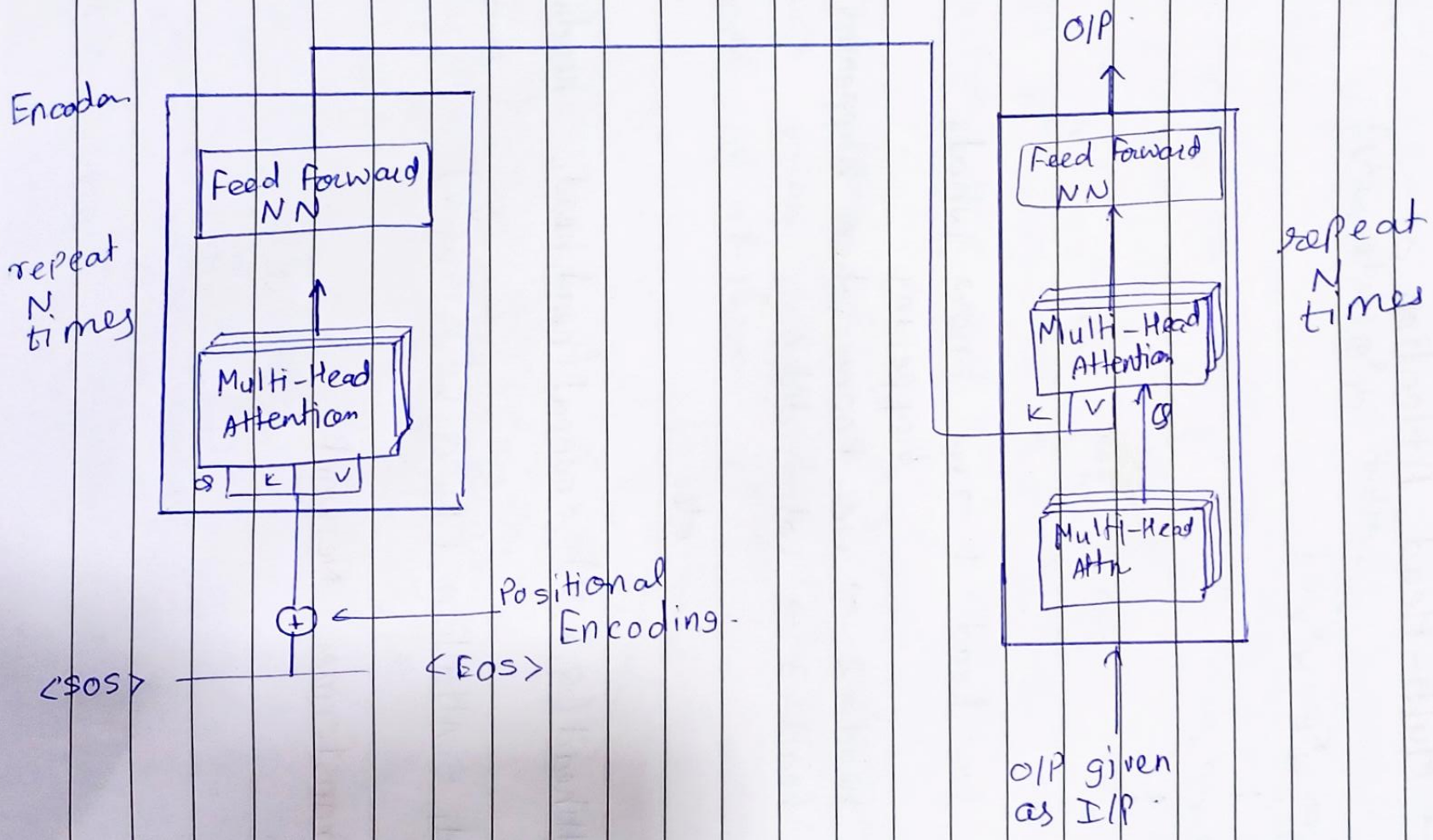
etc.

$$\text{Multihead}(Q, K, V) = \text{concat}(\text{head}_1, \text{head}_2 \text{ -- head}_h) W^O$$

zeros.

$$\text{head}_i = \text{Attention}(W_i^Q Q, W_i^K K, W_i^V V)$$

Transformer Network



Date :

MON TUE WED THU FRI SAT SUN