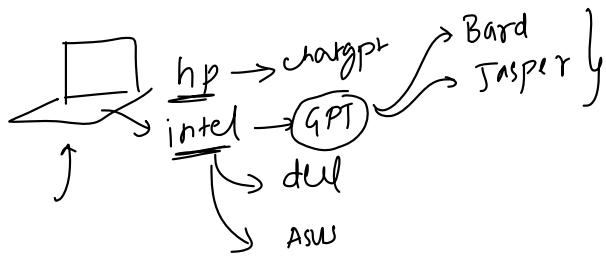
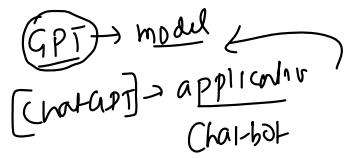
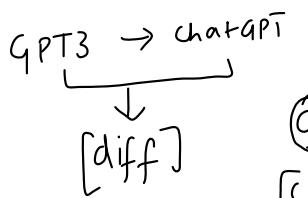


↳ energy consup  
↳ q p 13  
↳ small town  
↳ month



Nov

GPT3 → [ChatGPT]

1) RLHF → Reinforcement learning from human feedback

- + 1 Supervised finetuning → dataset
- + 2 reinforce → prompt production
- + responses
- + human → response bank

===== y labeled  
===== y

2) Incorporate safety and ethical guideline

- + minimize bias

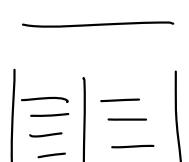
3) Improvement in contextual point

context → maintain context } dialogue  
retain } conversation

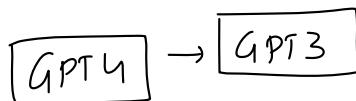
4) Dialogue specific training

- + conversation
- + better understanding → dialogue long → partitions

5) ChatGPT continuous imp → human feedback  
↳ usu



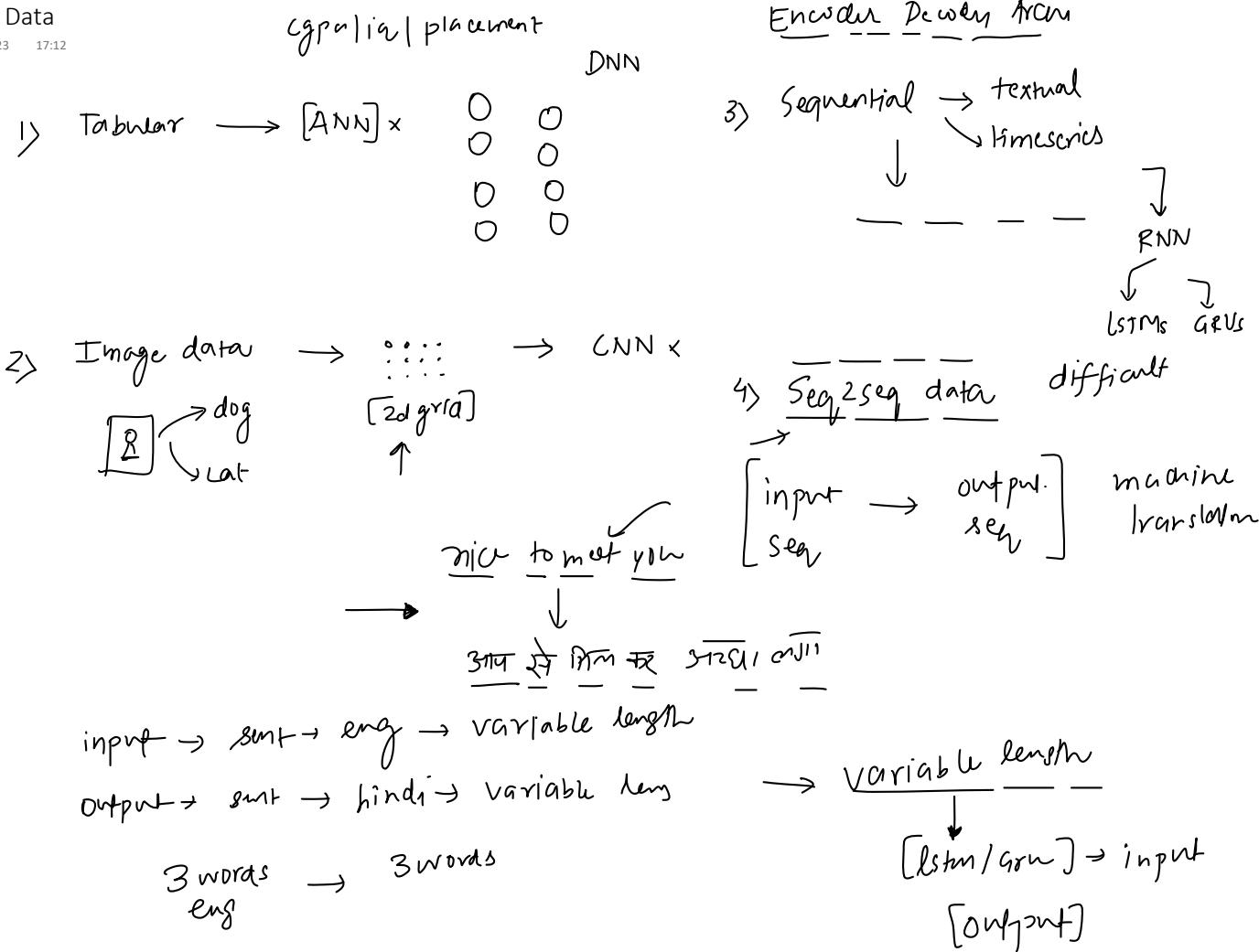
train → refining



$\left| \begin{array}{c} \diagup \\ \diagdown \end{array} \right| = \left| \begin{array}{c} \diagdown \\ \diagup \end{array} \right|$

train  $\rightarrow$   $y^{true}$

$\hookrightarrow$   $\boxed{\text{GPT4}}$   $\rightarrow$   $\boxed{\dots}$



# Before Starting

08 December 2023 19:24

## Prerequisite

- RNN / UTM

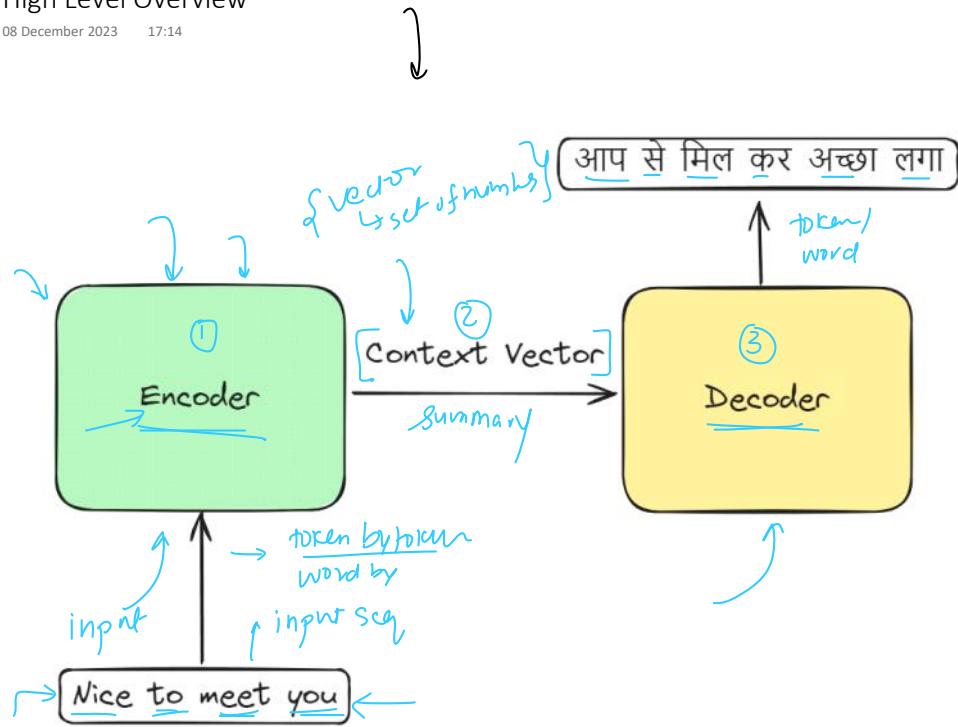
## Plan of attack

- simple version
- deep
- improvements

## High Level Overview

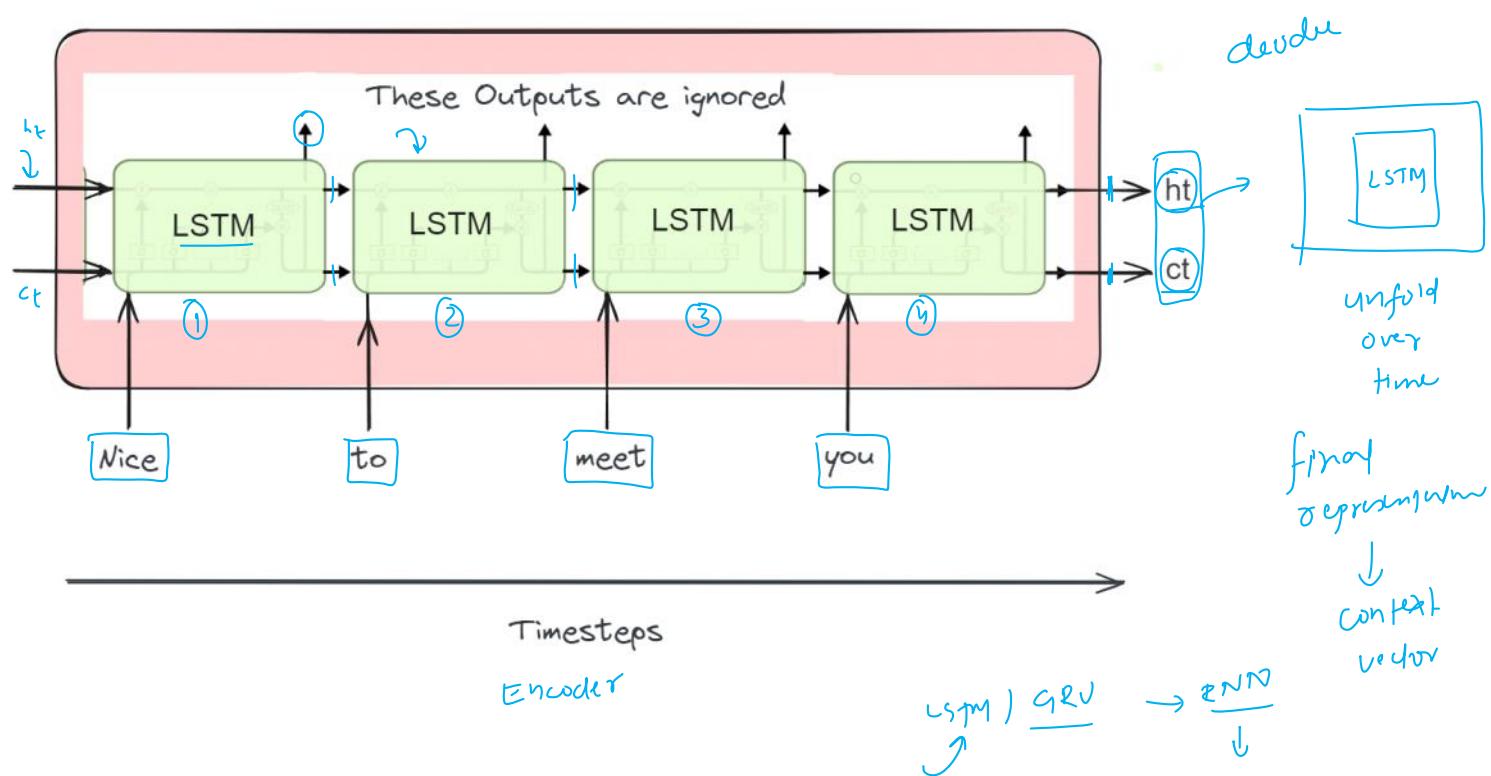
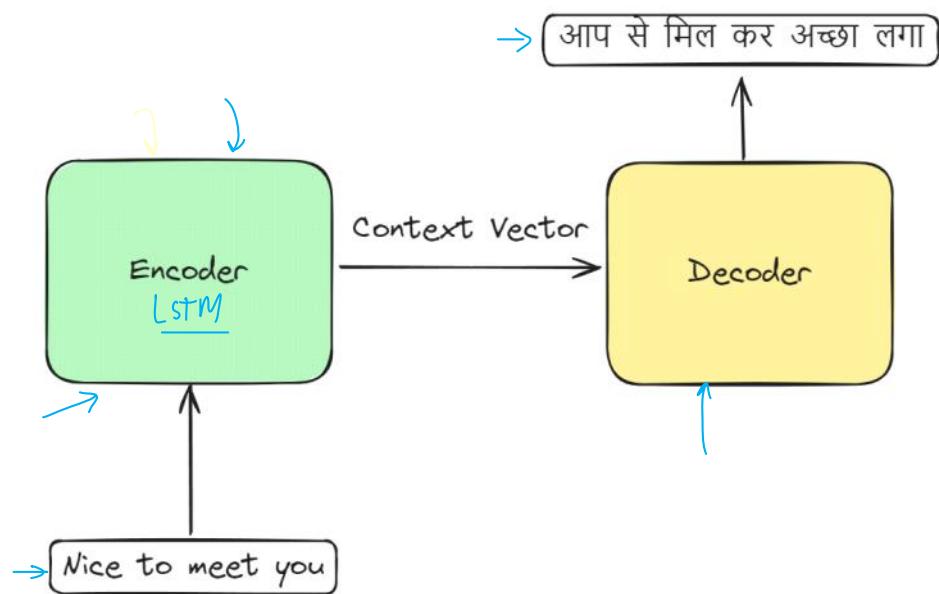
08 December 2023 17:14

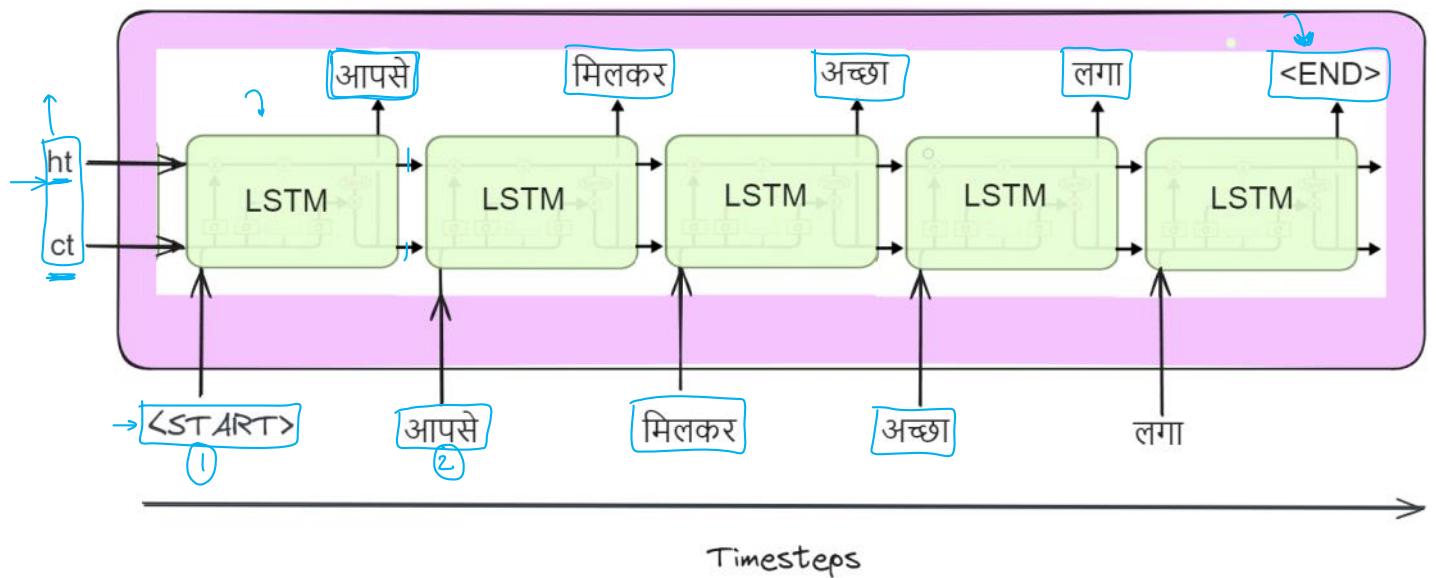
Machine tran → exam



## What's under the hood?

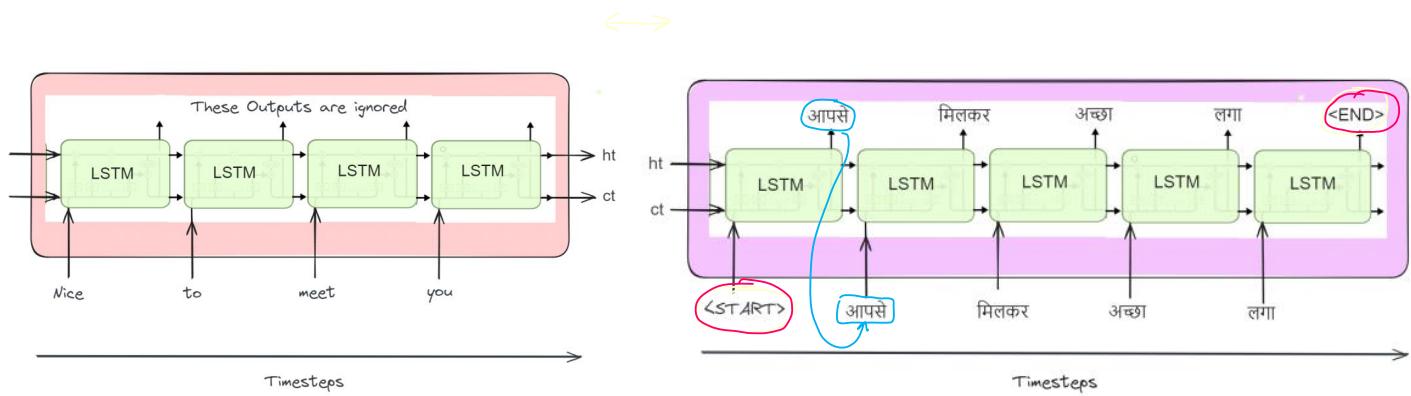
08 December 2023 17:14





## Training the Architecture using Backpropagation

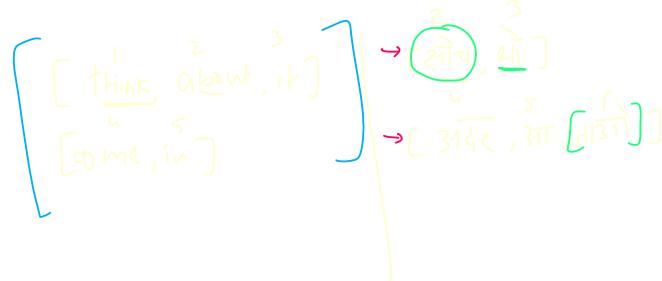
08 December 2023 17:15



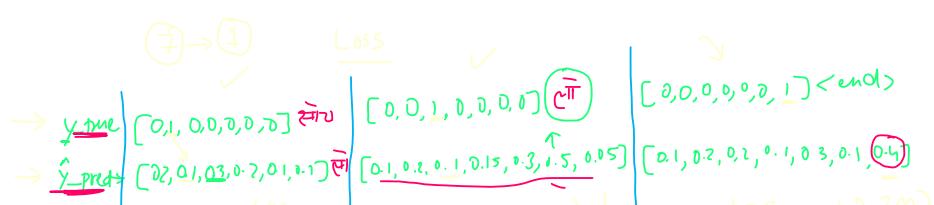
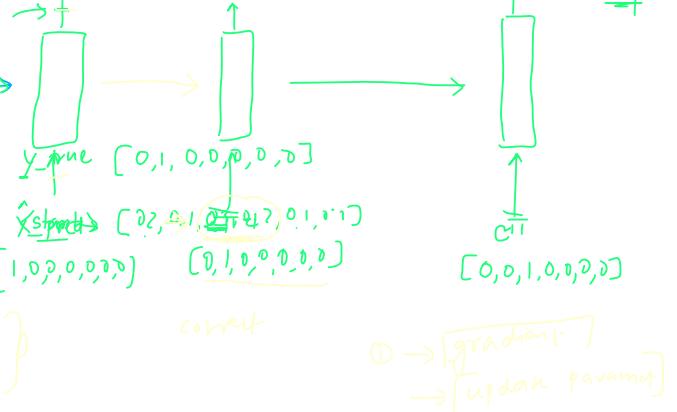
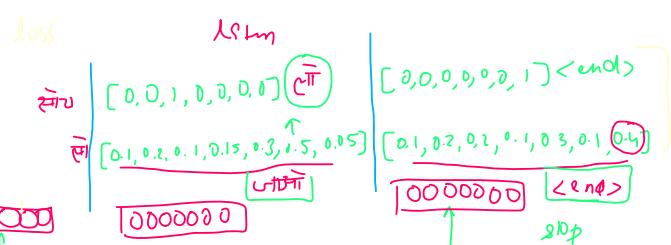
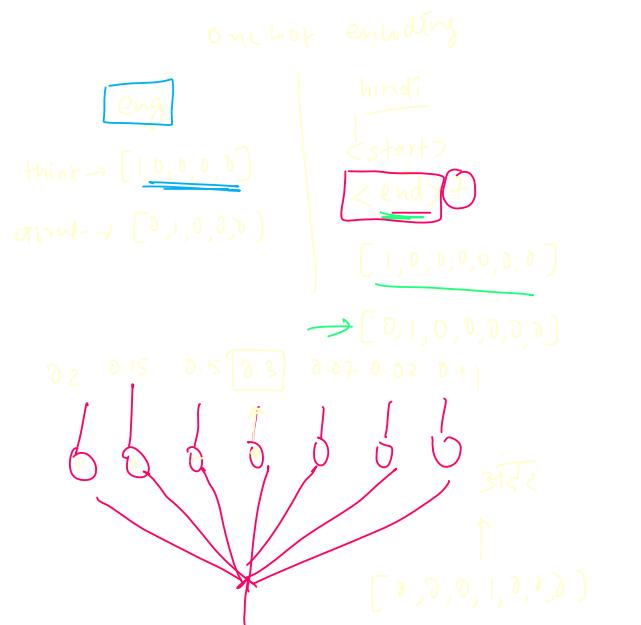
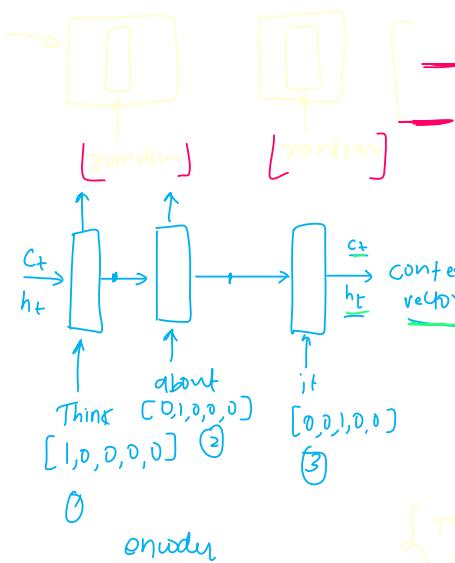
2) Dataset → Machine Translation



<u>eng</u> 1) → Think about it 2) → come in	<u>hindi</u> सोचो लो अंदर आ जाओ	→ numbers <u>NLP</u>
---	---------------------------------------	-------------------------

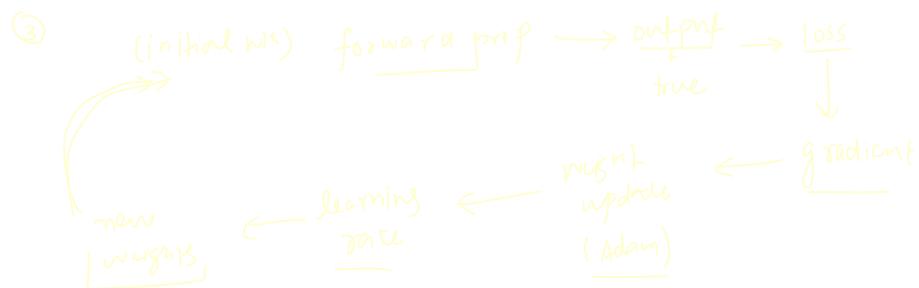
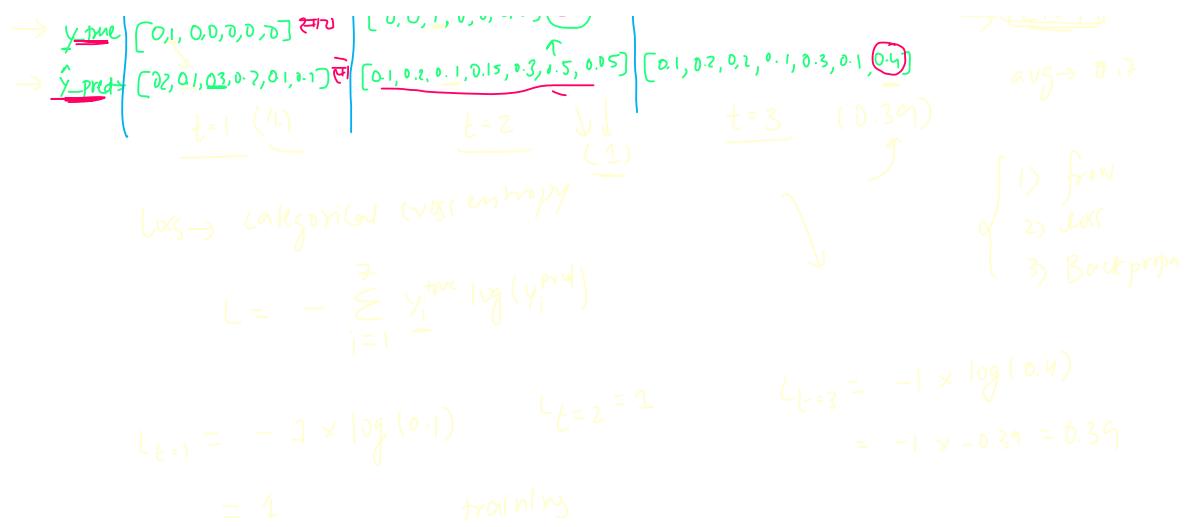


Row 1 → [Think about it] → [सोचो लो] <end>



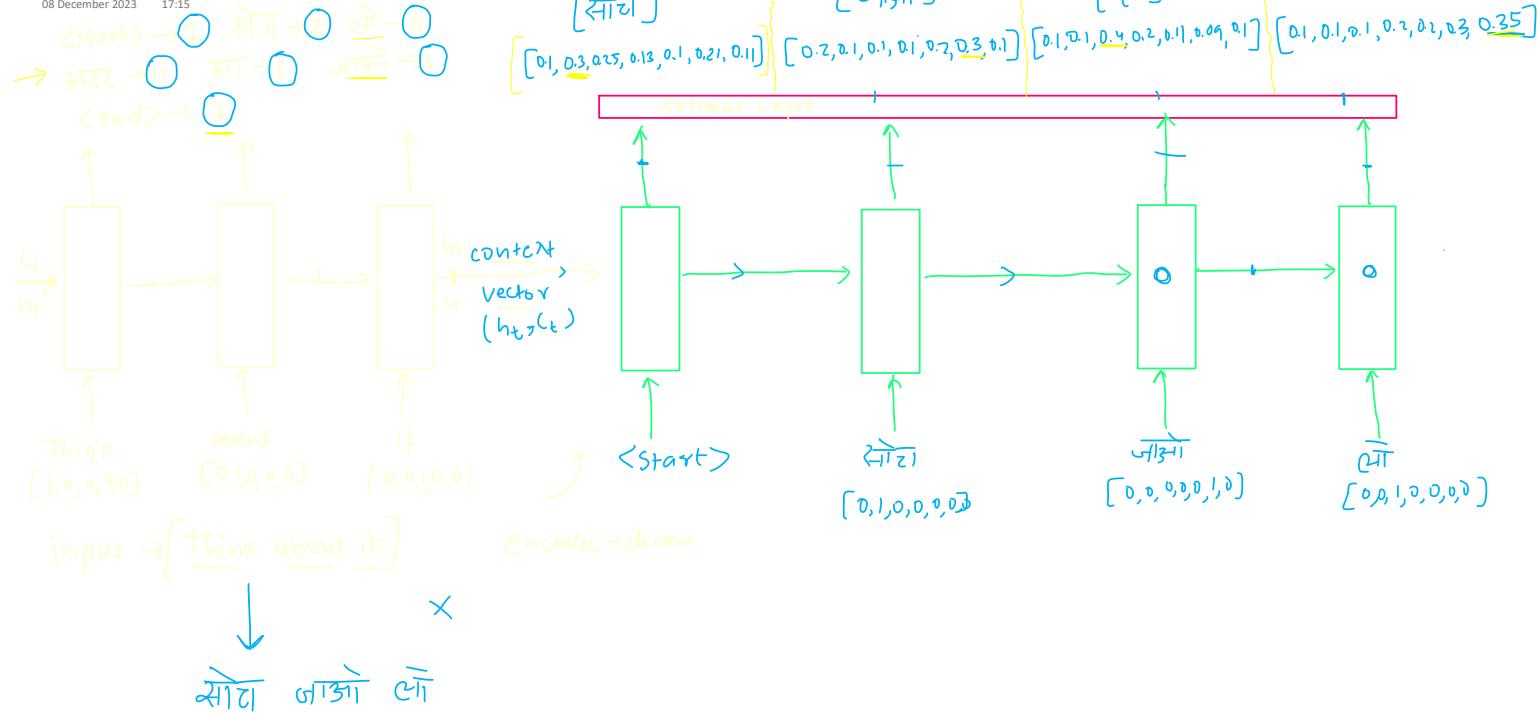
→ 2.39

avg → 0.7



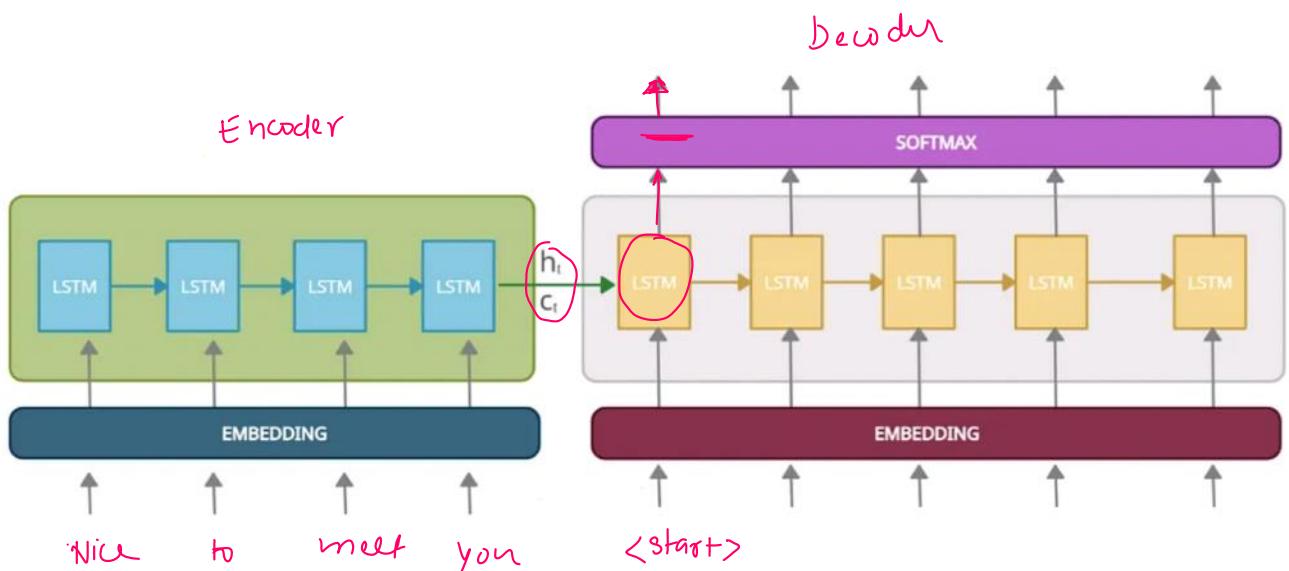
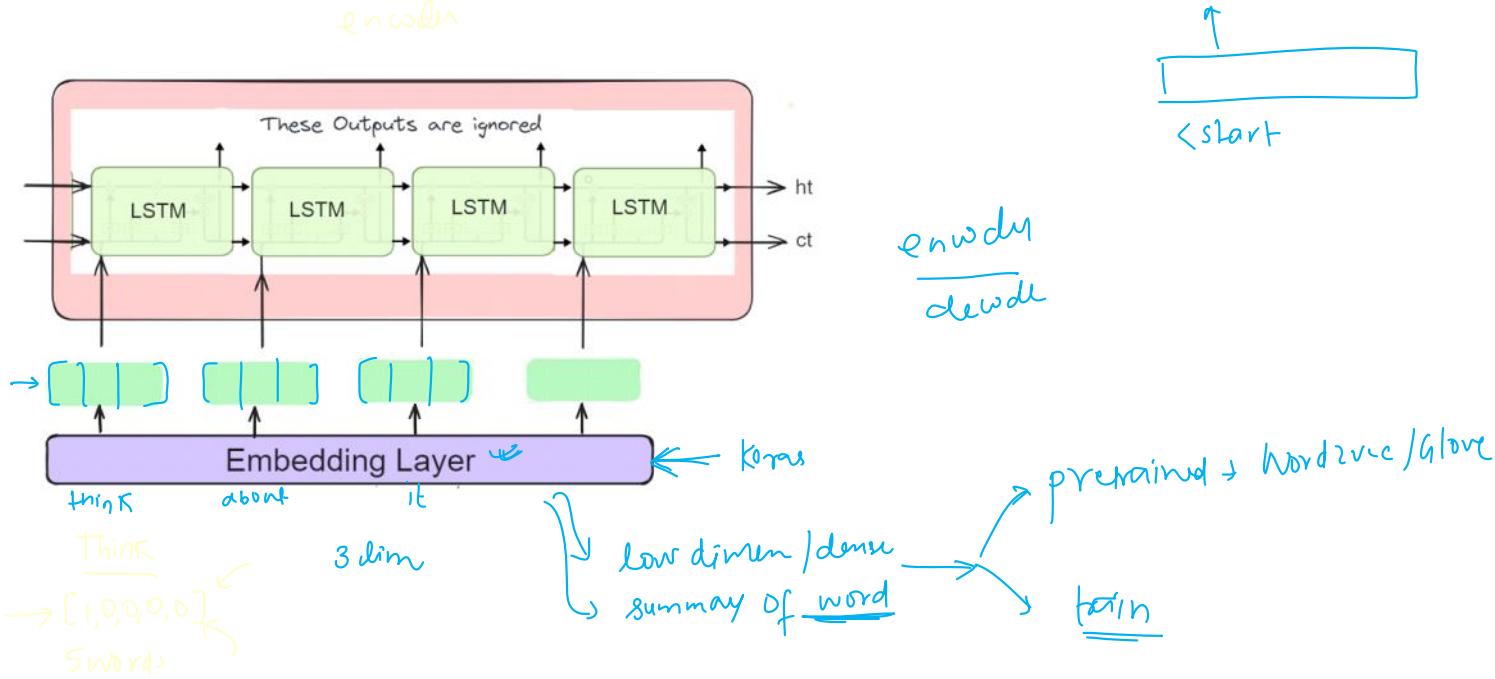
## Prediction

08 December 2023 17:15



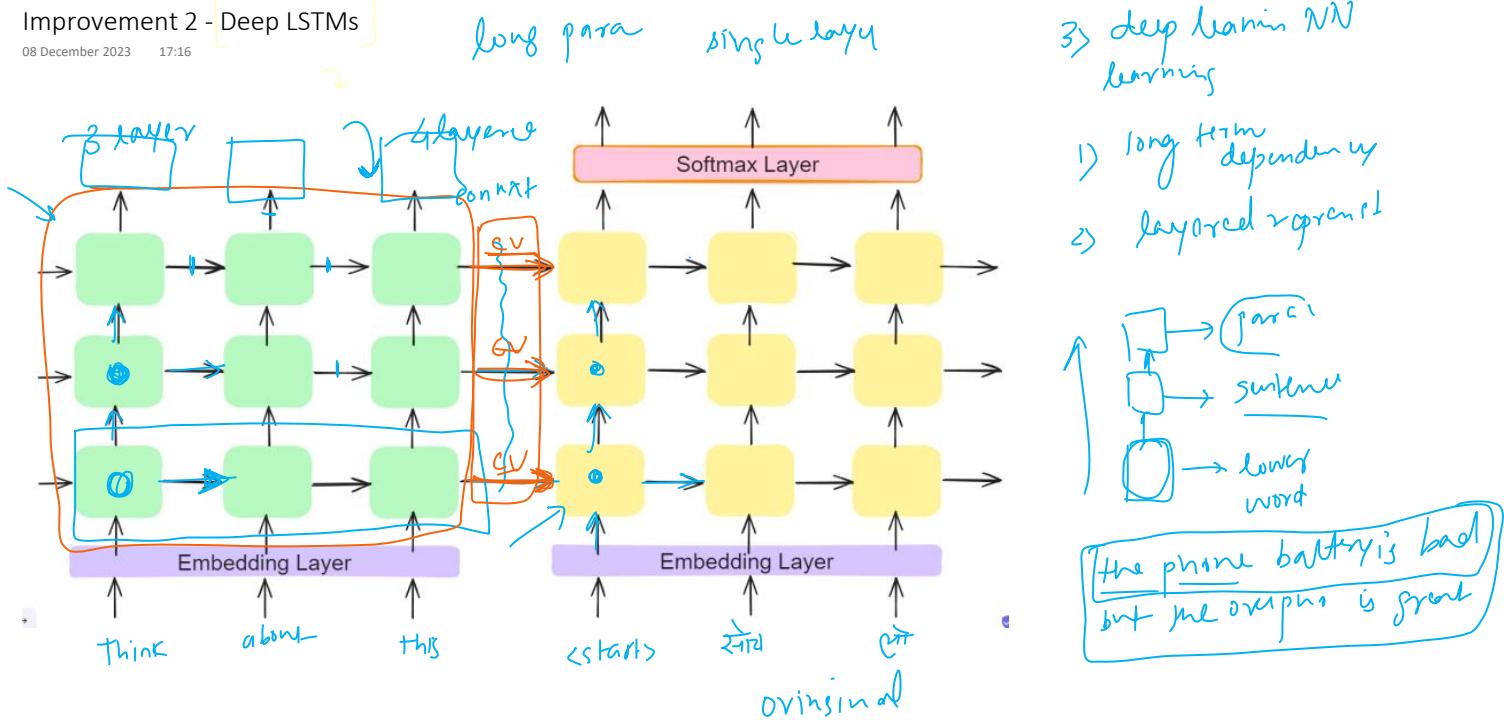
## Improvement 1 - Embeddings

08 December 2023 17:16



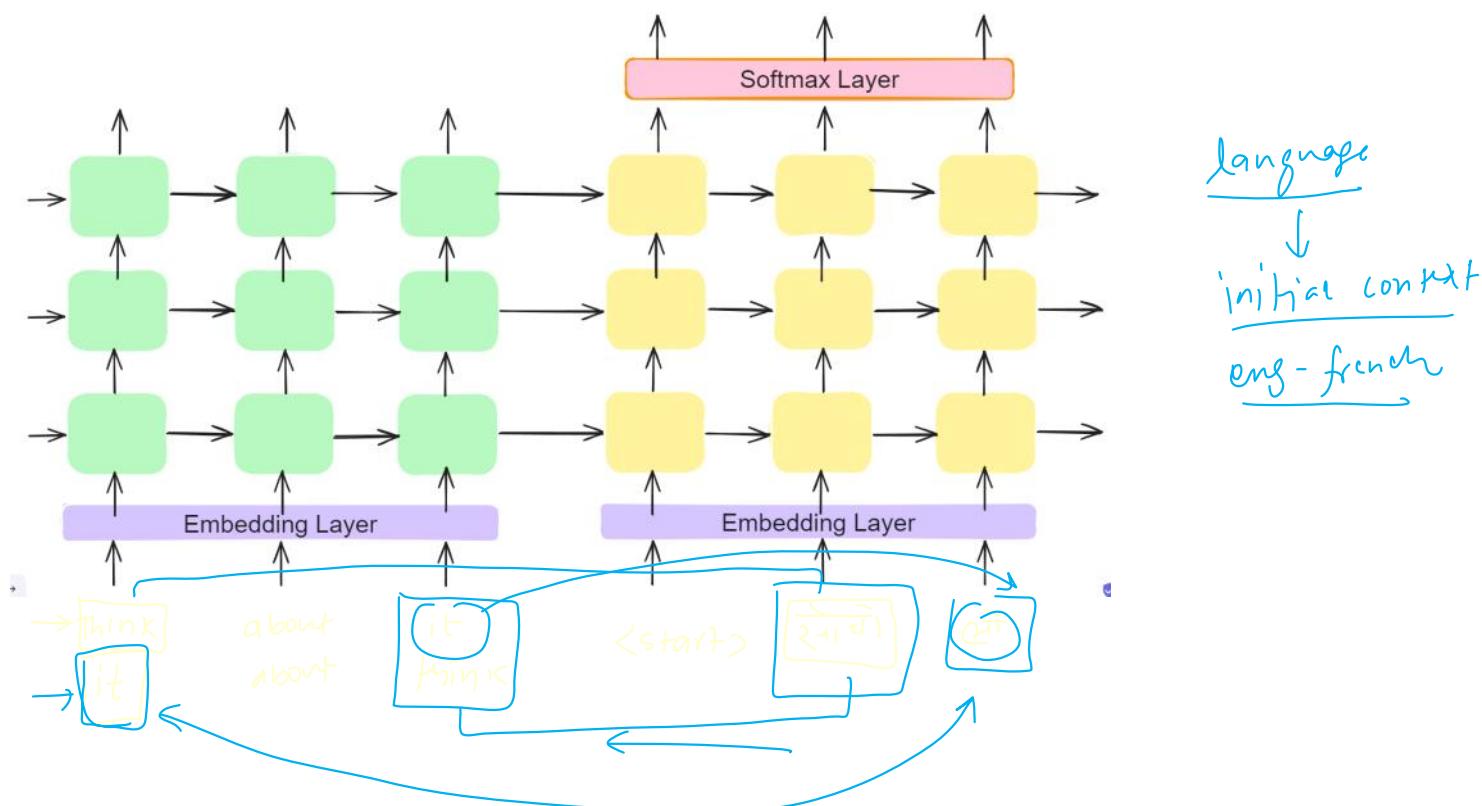
## Improvement 2 - Deep LSTMs

08 December 2023 17:16



## Improvement 3 - Reversing the Input

08 December 2023 17:16



## The Sutskever Architecture

08 December 2023 17:18

<start> <end>

**Application to Translation:** The model focused on translating English to French, demonstrating the effectiveness of sequence-to-sequence learning in neural machine translation.

**Special End-of-Sentence Symbol:** Each sentence in the dataset was terminated with a unique end-of-sentence symbol ("<EOS>"), enabling the model to recognize the end of a sequence.

→ **Dataset:** The model was trained on a subset of 12 million sentences, comprising 348 million French words and 304 million English words, taken from a publicly available dataset.

**Vocabulary Limitation:** To manage computational complexity, fixed vocabularies for both languages were used, with 160,000 most frequent words for English and 80,000 for French. Words not in these vocabularies were replaced with a special "UNK" token.

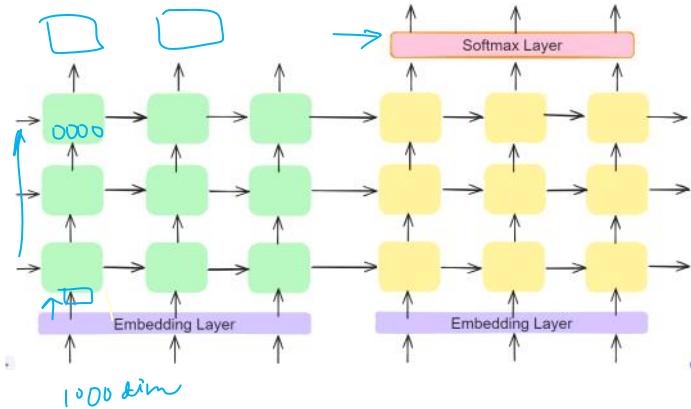
**Reversing Input Sequences:** The input sentences (English) were reversed before feeding them into the model, which was found to significantly improve the model's learning efficiency, especially for longer sentences.

**Word Embeddings:** The model used a 1000-dimensional word embedding layer to represent input words, providing dense, meaningful representations of each word.

**Architecture Details:** Both the input (encoder) and output (decoder) models had 4 layers, with each layer containing 1,000 units, showcasing a deep LSTM-based architecture.

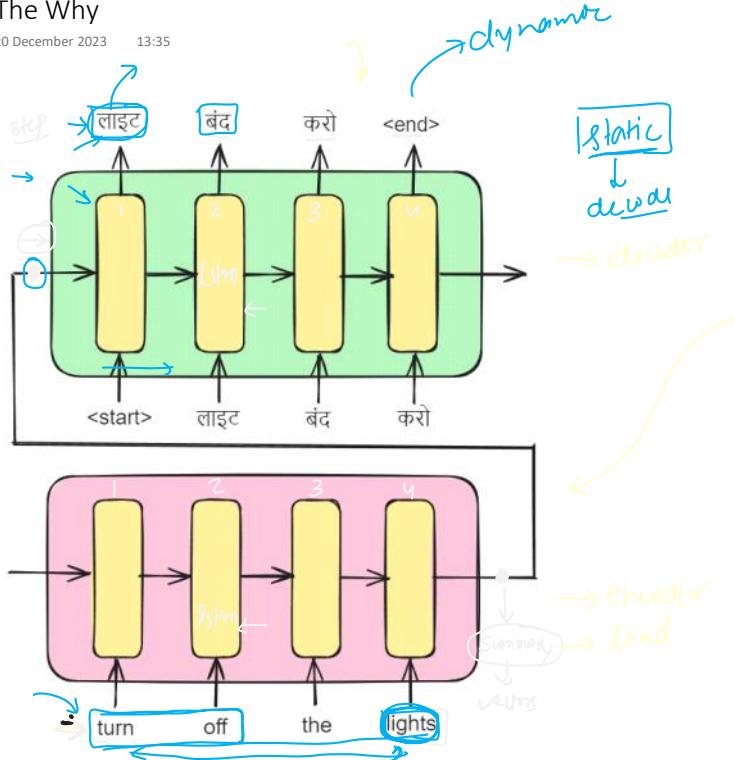
**Output Layer and Training:** The output layer employed a Softmax function to generate the probability distribution over the target vocabulary. The model was trained end-to-end with these settings.

**Performance - BLEU Score:** The model achieved a BLEU score of 34.81, surpassing the baseline Statistical Machine Translation (SMT) system's score of 33.30 on the same dataset, marking a significant advancement in neural machine translation.



## The Why

20 December 2023 13:35



50 words

Encoder

Once upon a time in a small Indian village, a mischievous monkey stole a turban from a sleeping barber, wore it to a wedding, danced with the bewildered guests, accidentally got crowned the 'Banana King' by the local kids, and ended up leading a vibrant, impromptu parade of laughing villagers, cows, and street dogs, all while balancing a stack of mangoes on its head, creating a hilariously unforgettable spectacle and an amusing legend that the village still chuckles about every monsoon season.

Decoder

> 25 words

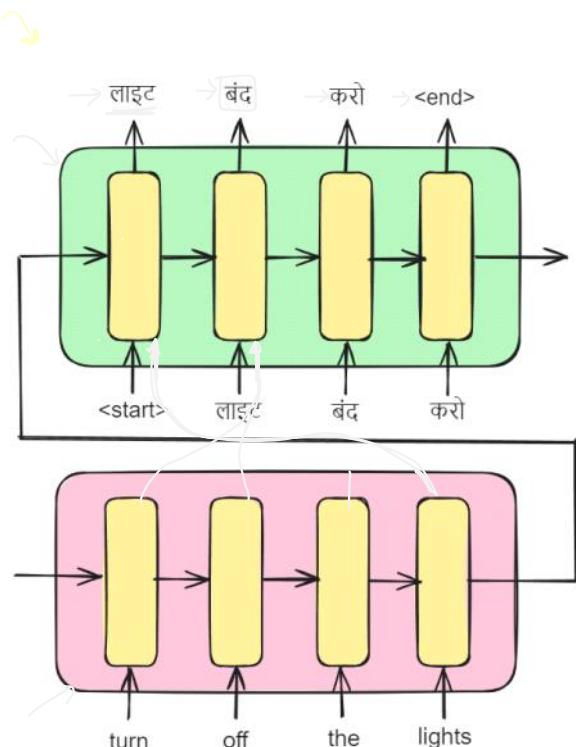
## The Solution

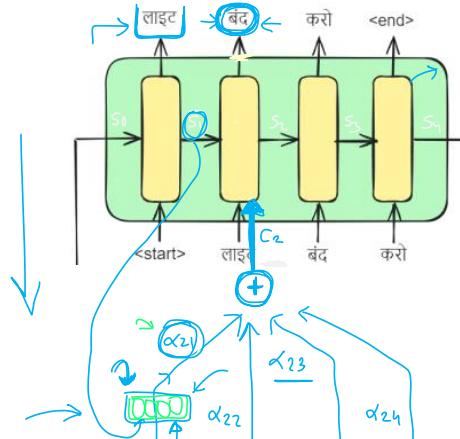
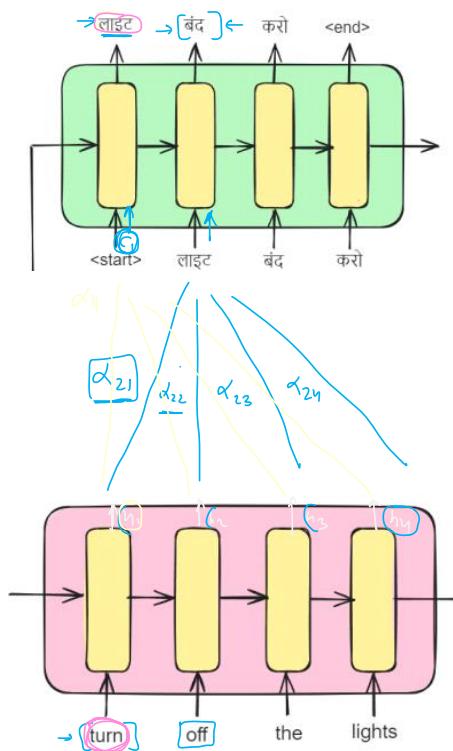
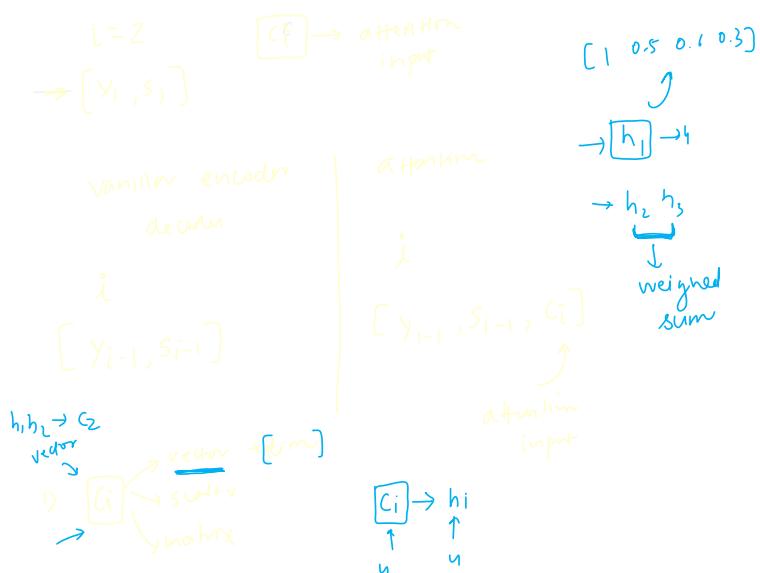
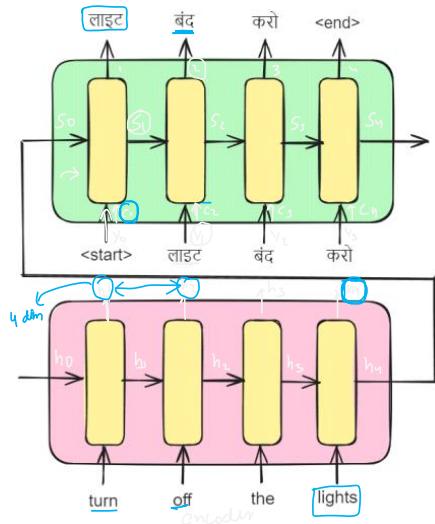
20 December 2023 17:32

Information is valuable to individual companies in determine what information part of Information Security strategy is knowledge based part as general however intellectual and knowledge-based assets

Once upon a time in a small Indian village, a mischievous monkey stole a turban from a sleeping barber, wore it to a wedding, danced with the bewildered guests, accidentally got crowned the 'Banana King' by the local kids, and ended up leading a vibrant, impromptu parade of laughing villagers, cows, and street dogs, all while balancing a stack of mangoes on its head, creating a hilariously unforgettable spectacle and an amusing legend that the village still chuckles about every monsoon season.

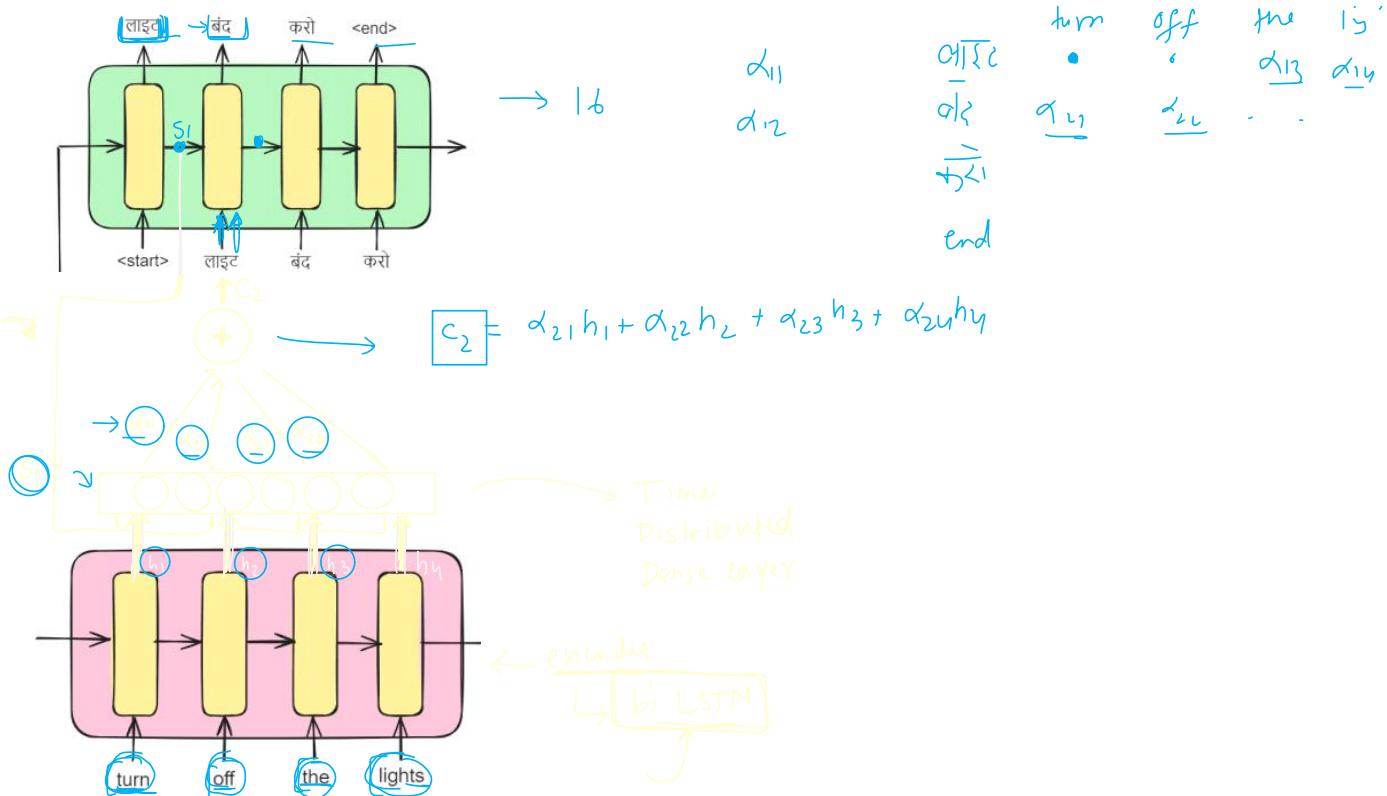
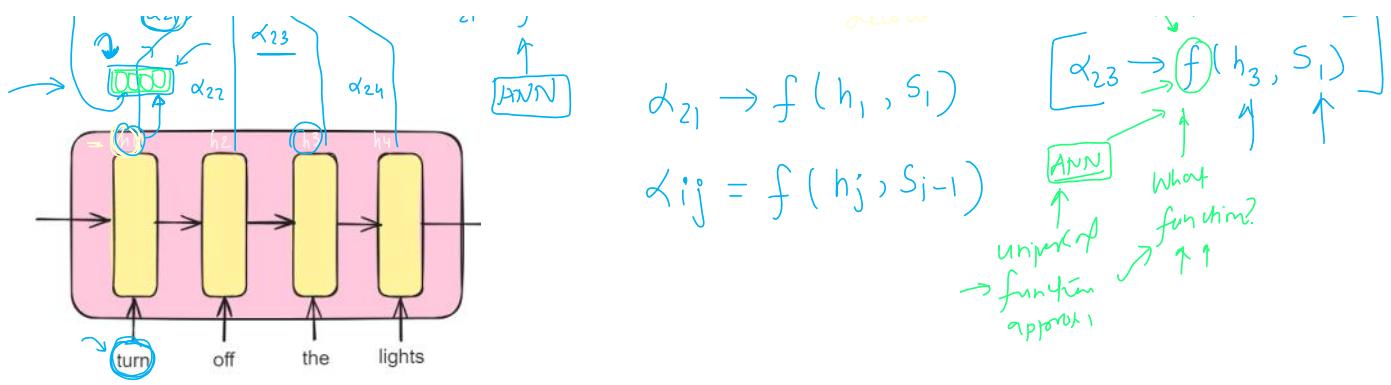
Attention



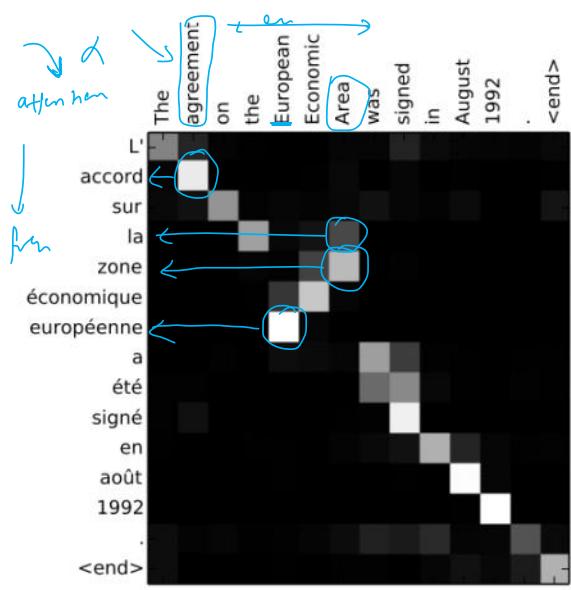
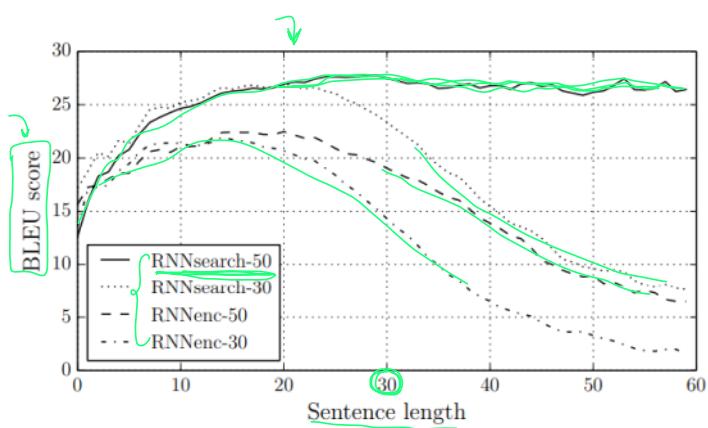


The diagram illustrates an RNN architecture with two main paths: forward and backward.

- Forward Path:** Represented by a sequence of hidden states  $h_1, h_2, \dots$  and inputs  $s_1, s_2, \dots$ . The hidden state  $h_1$  is initialized from an input  $a$  (indicated by a blue arrow). Each hidden state  $h_i$  is generated by a function  $f$  that takes the previous hidden state  $h_{i-1}$  and the current input  $s_i$  as inputs. The output of the forward pass is labeled "given".
- Backward Path:** Represented by a sequence of hidden states  $h_1, h_2, \dots$  and inputs  $s_1, s_2, \dots$ . The hidden state  $h_1$  is initialized from an input  $a$  (indicated by a blue arrow). Each hidden state  $h_i$  is generated by a function  $f$  that takes the previous hidden state  $h_{i+1}$  and the current input  $s_i$  as inputs. The output of the backward pass is labeled "given".
- Contextual State:** A box labeled  $\alpha_{21}$  represents a contextual state. It receives inputs from both the forward and backward hidden states  $h_1$  and  $h_2$ , and from the forward input  $s_1$ . This state is used for alignment and similarity calculations.
- Output:** The final output is determined by the value of  $i = 2$ .

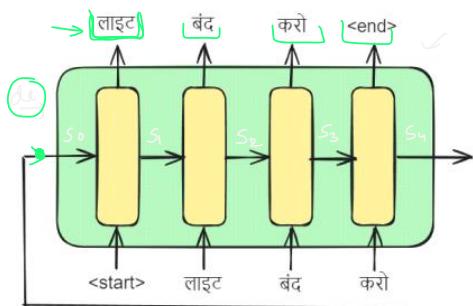


eng-fran



## Recap

16 January 2024 16:10

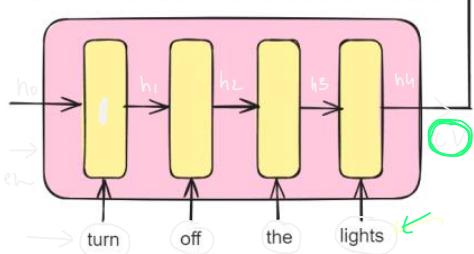


turn off the lights → लाइट बंद करो <end>

[NMT]

Encoder-Decoder

sentence > 30 words  
paragraph  
document

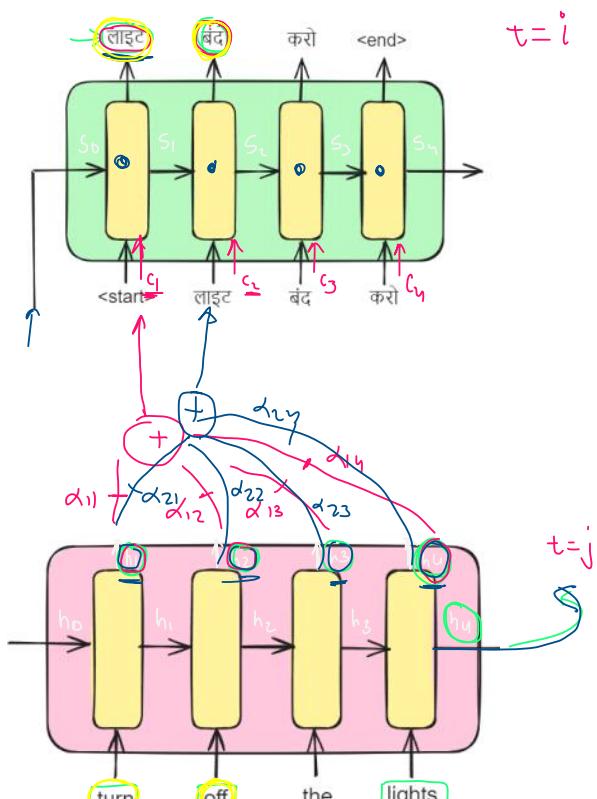


BiLSTM  
Stacked LSTM

softmax

translation

bottleneck → Attention mechanism



$c_1 \approx c_2 \approx c_3 \approx c_4$

$$4 \times 4 = 16$$

Weighted sum

$$c_i^* = \sum_{j=1}^4 \alpha_{ij} h_j$$

$\alpha \rightarrow$  alignment score

$$c_1 = \alpha_{11} h_1 + \alpha_{12} h_2 + \alpha_{13} h_3 + \alpha_{14} h_4$$

$$c_2 = \alpha_{21} h_1 + \alpha_{22} h_2 + \alpha_{23} h_3 + \alpha_{24} h_4$$

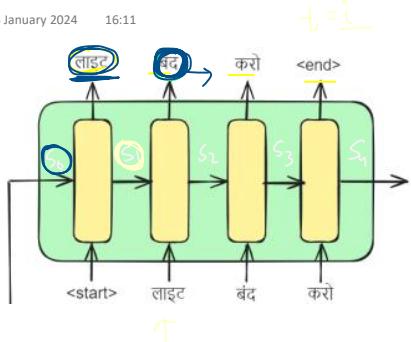
$\alpha \rightarrow$  find out

Bahdanau  
attention  
archite

Luong  
attention

## Bahdanau Attention

16 January 2024 16:11



alignment score

given

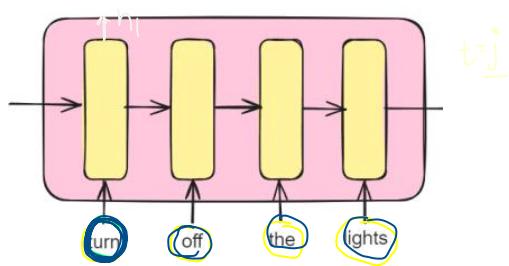
$$g_j = \sum_i \alpha_{ij} h_j$$

alignment

decoded  
→ prev hidden  
state

$$\alpha_{11} \rightarrow \text{लाइट} \rightarrow \text{turn}$$

$$\alpha_{12} \rightarrow \text{बंद} \rightarrow \text{off}$$

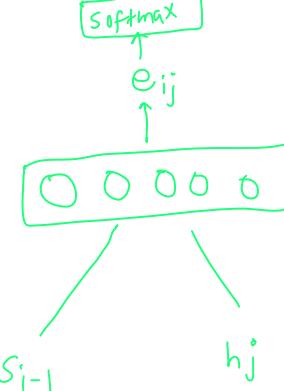


$$\underline{\alpha_{11}} = f(h_1, s_0) \quad \underline{\alpha_{21}} = f(h_2, s_0)$$

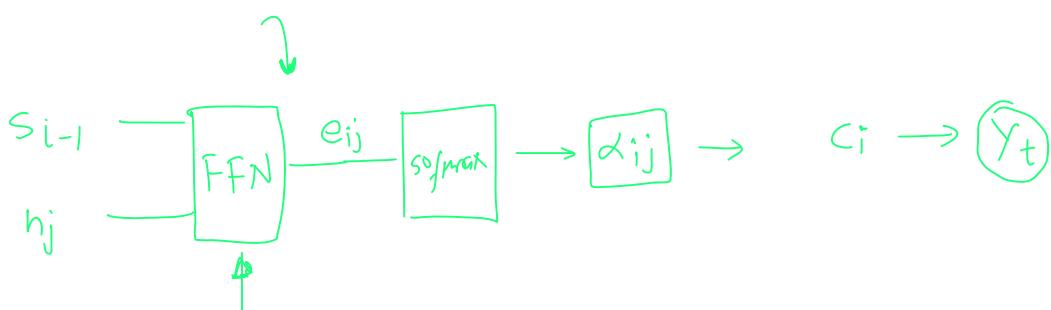
$$\alpha_{ij} = \sum h_j, s_{i-1}$$

approximate

$$FFN \rightarrow ANN \rightarrow \underline{UFA}$$



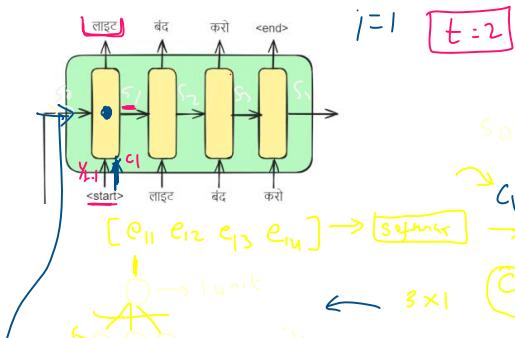
deal  
prev  
hidden  
state



$$s_0, Y_{t+1}, c_1 \rightarrow \text{softmax} \rightarrow Y_t (\text{लाइट}) [S_1]$$

↳ encodes sentence  
↳ h1, h2, h3, hn

(Sj-1) hj



$$s_0 = [e f g h]$$

$$c_1 = \sum \alpha_{ij} h_j$$

$$[\alpha_{11} \alpha_{12} \alpha_{13} \alpha_{14}]$$

$$c_1 = \frac{\alpha_{11} h_1}{\sum} + \frac{\alpha_{12} h_2}{\sum} + \frac{\alpha_{13} h_3}{\sum} + \frac{\alpha_{14} h_4}{\sum}$$

Diagram illustrating a linear combination of hidden states:

$$C_1 = \alpha_{11} h_1 + \alpha_{12} h_2 + \alpha_{13} h_3 + \alpha_{14} h_4$$

where  $\alpha = [e \ f \ g \ h]$

batch operation:  $u \times 8 \rightarrow 8 \times 3 \rightarrow u \times 3$  (using  $\tan(u \times 3)$ )

Diagram illustrating a neural network layer:

Input: turn, off, the, lights

Output:  $h_0 = [a \ b \ c \ d]$  (4dim)

$(4 \times 3) \rightarrow 3 \times 1$

$\left[ e^{e^{11}} + e^{e^{12}} + e^{e^{13}} + e^{e^{14}} \right] \rightarrow 4 \text{ numbers}$

Diagram illustrating scaling of hidden states:

4 rows / 8 cols → 8 cols

$s_{01}$	$s_{02}$	$s_{03}$	$s_{04}$	$h_{11} \ h_{12} \ h_{13} \ h_{14}$
$s_{01}$	$s_{02}$	$s_{03}$	$s_{04}$	$h_{21} \ h_{22} \ h_{23} \ h_{24}$
$s_{01}$	$s_{02}$	$s_{03}$	$s_{04}$	$h_{31} \ h_{32} \ h_{33} \ h_{34}$
$s_{01}$	$s_{02}$	$s_{03}$	$s_{04}$	$h_{41} \ h_{42} \ h_{43} \ h_{44}$

Diagram illustrating a time-distributed fully connected network (FNN):

True label:  $i=2$

Input: turn, off, the, lights

Output:  $C_2 = \alpha_{21} h_1 + \alpha_{22} h_2 + \alpha_{23} h_3 + \alpha_{24} h_4$

Diagram illustrating the attention mechanism:

alignement model:  $e_{ij} = \sum \alpha_{ij} h_j$

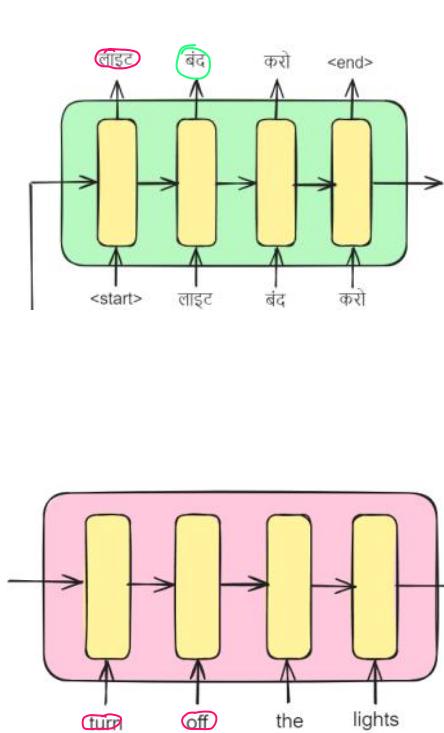
additive attention:  $\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^n \exp(e_{ik})}$

time distributed FNN:

$e_{ij} = V \tanh(W [s_{i-1}; h_j] + b)$

## Luong Attention

17 January 2024 00:09



parameters → slow

$$c_i = \sum \alpha_{ij} h_j \quad \text{FFN} \quad \left\{ \begin{array}{l} \boxed{V \tan(w[s_{i-1}, h_j] + b)} \\ \uparrow \end{array} \right\}$$

$$\alpha_{ij} = f(s_{i-1}, h_j) \times$$

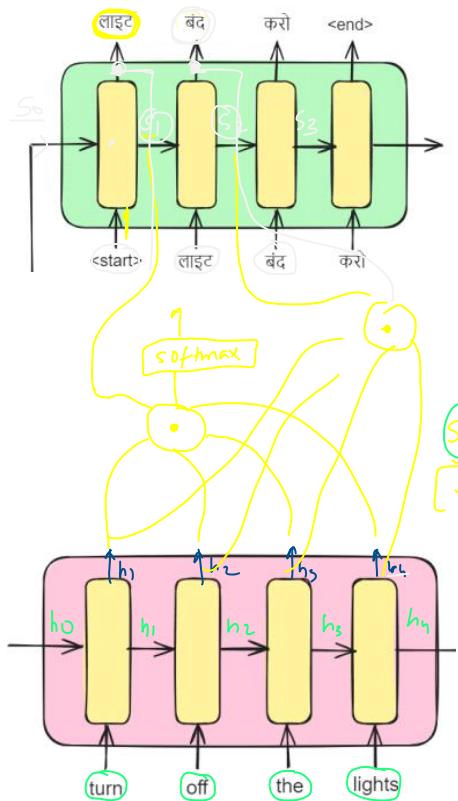
$$\underline{\alpha_{ij} = f(s_i, h_j) \rightarrow [s_i^T \cdot h_j]} \rightarrow \underline{\text{dot product}} \quad \text{fast}$$

updated info  $\xrightarrow{\text{current diff}}$   $s_i = [a \ b \ c \ d]$

$h_j = [e \ f \ g \ h]$

$\alpha_{ij} \uparrow$

$\text{softmax} \leftarrow \boxed{e_{ij}}$   $\left[ \begin{array}{c} a \\ b \\ c \\ d \end{array} \right] \left[ \begin{array}{c} e \\ f \\ g \\ h \end{array} \right] \rightarrow [ae + bf + cg + dh] \rightarrow \text{scalar} \rightarrow \text{attention}$



output  
 $s_1, s_2$

$\boxed{s}$  → softmax

$s_2: e_2 \rightarrow \tilde{e}_2 \rightarrow$  multiplicative after

$$[e_{21} \ e_{22} \ e_{23} \ e_{24}] \text{ softmax} \rightarrow \alpha_{21} \ \alpha_{22} \ \alpha_{23} \ \alpha_{24}$$

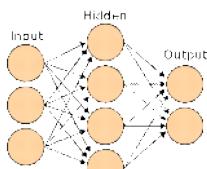
$$\sum \alpha_{ij} h_j$$

$e_{11} \ e_{12} \ e_{13} \ e_{14}$

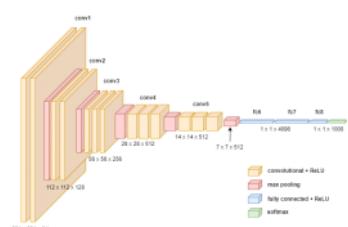
$\alpha_{11} \ \alpha_{12} \ \alpha_{13} \ \alpha_{14} \rightarrow \boxed{c_1}$

# What is Transformer?

27 January 2024 18:41



ANN  
tabular



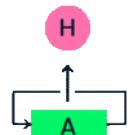
CNN  
Image

→ [Transformer] → seq2seq task

NN arch

- + machine translation
- + question ans
- + text summariz

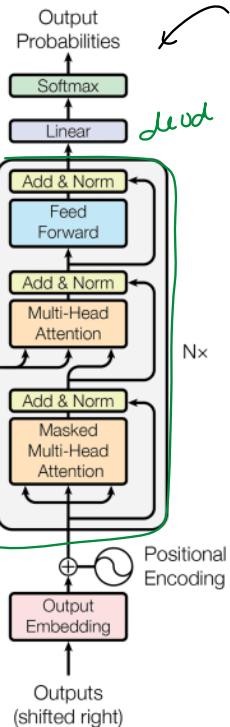
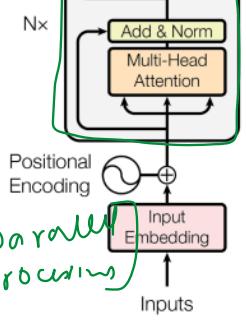
→ Self attention stable



RNN  
Sequence  
→ Text

Arch  
lstm  
enwdy

enwdy



## Attention Is All You Need

2017

→ Google Brain

Deep learnis

Ashish Vaswani\*  
Google Brain  
avaswani@google.com

Noam Shazeer\*  
Google Brain  
noam@google.com

Niki Parmar\*  
Google Research  
nikip@google.com

Jakob Uszkoreit\*  
Google Research  
usz@google.com

Llion Jones\*  
Google Research  
llion@google.com

Aidan N. Gomez\* †  
University of Toronto  
aidan@cs.toronto.edu

Lukasz Kaiser\*  
Google Brain  
lukasz.kaiser@google.com

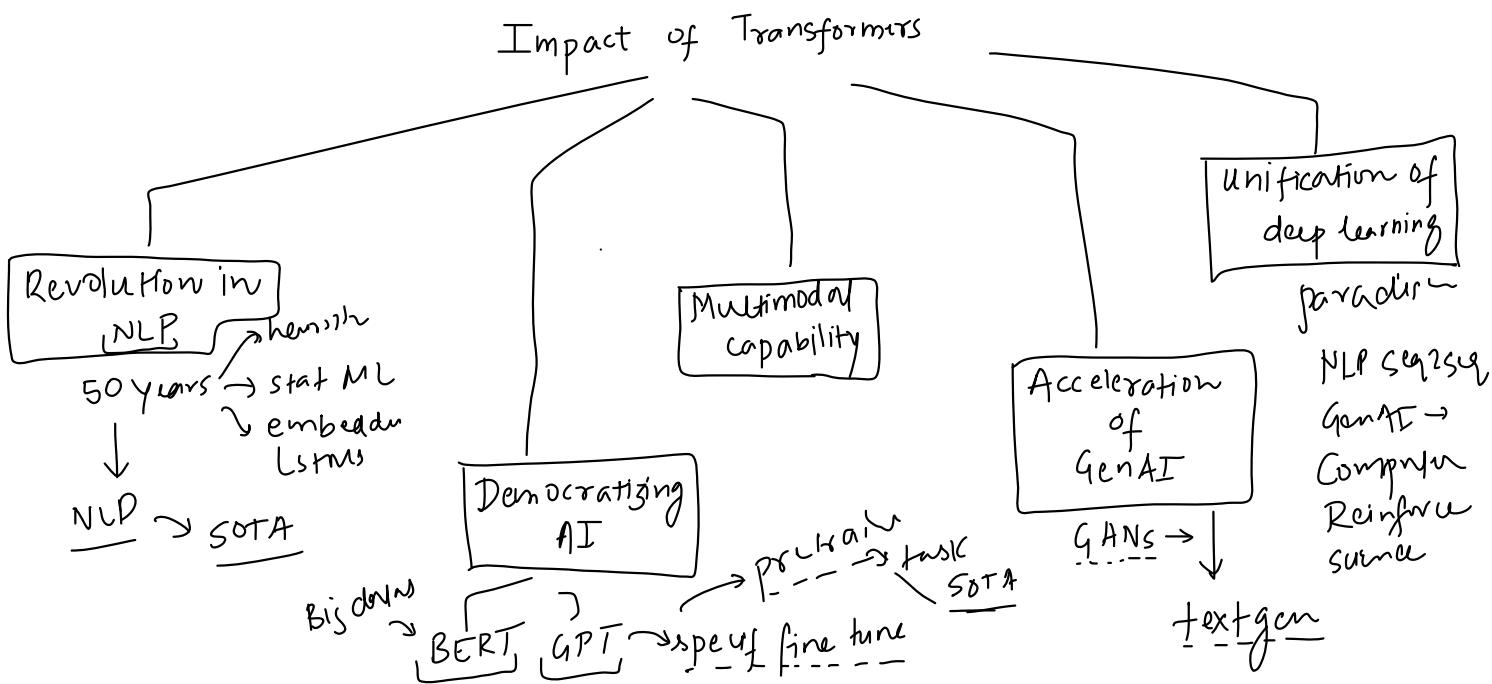
Illia Polosukhin\* ‡  
illia.polosukhin@gmail.com

### Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

# Impact of Transformers

27 January 2024 20:03



# The Origin Story!

27 January 2024 22:38

2014-15 → seq2seq machine

## Sequence to Sequence Learning with Neural Networks

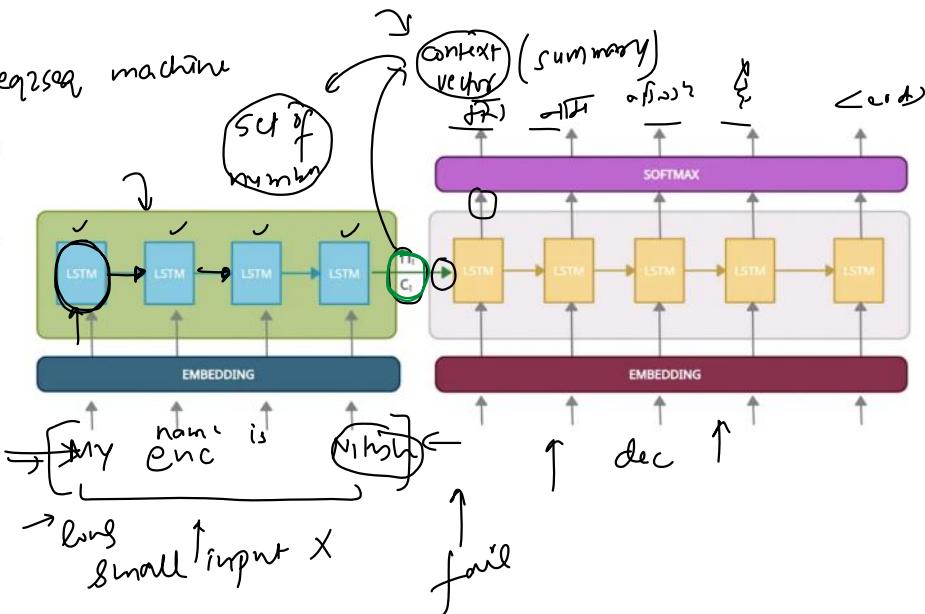
Ilya Sutskever  
Google  
ilyasu@google.com

Oriol Vinyals  
Google  
vinyals@google.com

Quoc V. Le  
Google  
qvl@google.com

### Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multi-layered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that an English to French translation task from the WMT'14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous best result on this task. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice. Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier.



## { NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE }

Dzmitry Bahdanau  
Jacobs University Bremen, Germany

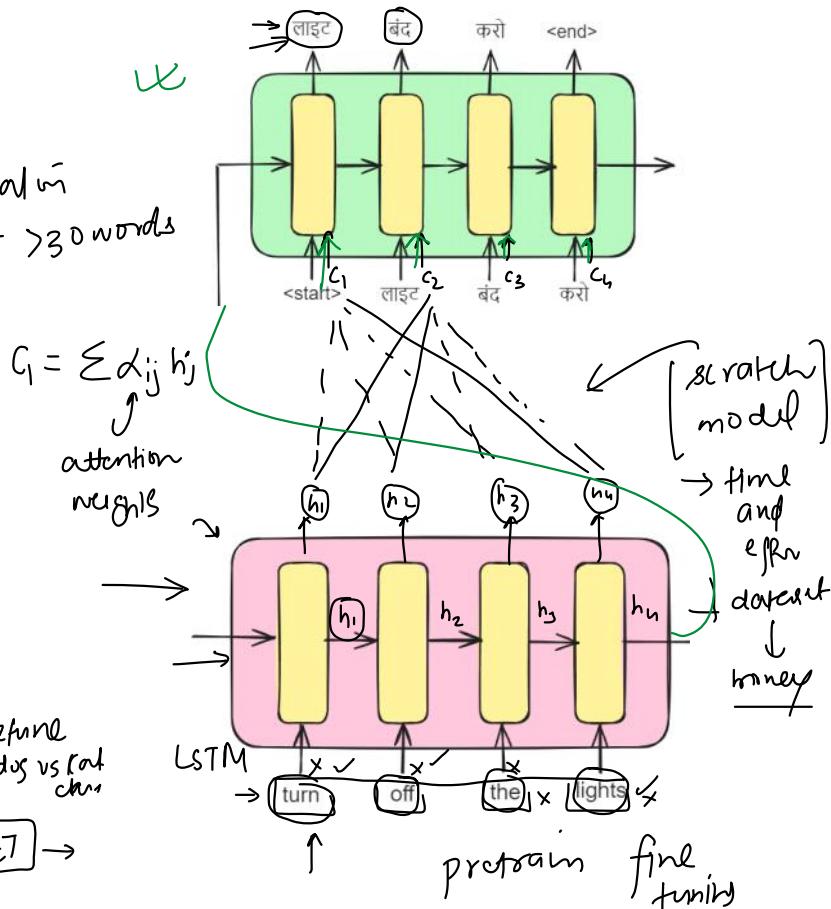
KyungHyun Cho Yoshua Bengio\*  
Université de Montréal

### ABSTRACT

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder-decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder-decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-alignments) found by the model agree well with our intuition.

(sequential) → slow → huge dataset

Transfer learning  
→ CNN → dog vs cat  
→ ImageNet →



research → incremental time travel  
Attention Is All You Need → 2017

→ seq2seq → parallelly  
→ stable  
→ hyperfine  
NLP → BERT → fine tune  
Output Probabilities

→ research → train

[Attention Is All You Need] → 2017



Ashish Vaswani\*  
Google Brain  
avaswani@google.com

Noam Shazeer\*  
Google Brain  
noam@google.com

Niki Parmar\*  
Google Research  
nikip@google.com

Jakob Uszkoreit\*  
Google Research  
usz@google.com

Llion Jones\*  
Google Research  
llion@google.com

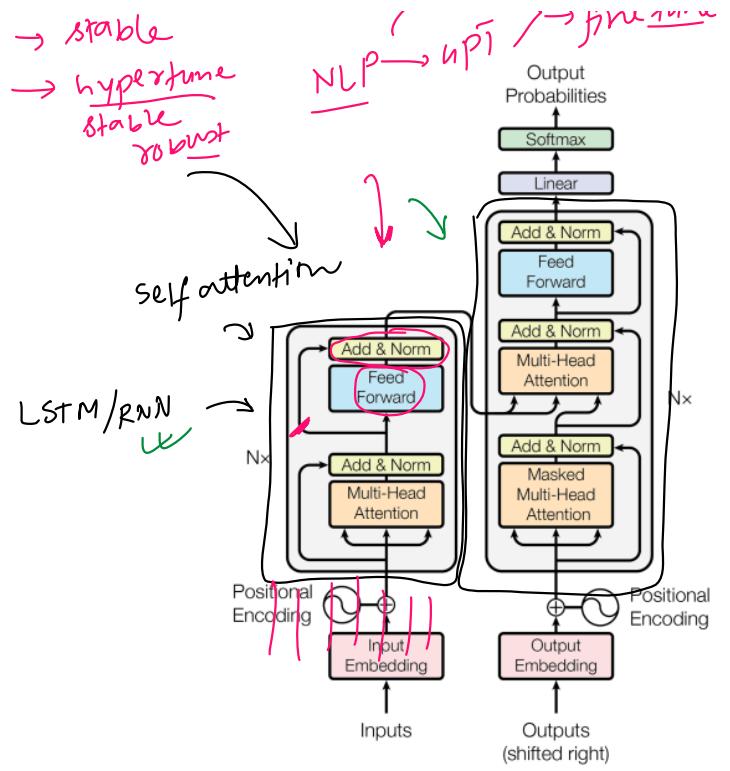
Aidan N. Gomez\* †  
University of Toronto  
aidan@cs.toronto.edu

Lukasz Kaiser\*  
Google Brain  
lukaszkaiser@google.com

Illia Polosukhin\* ‡  
illia.polosukhin@gmail.com

## Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.



# The Timeline

28 January 2024 00:55

2000 - 2014 → RNNs / LSTMs



2014 → Attention



2017 → Transformer

2018 → BERT / GPT (Transfer learning)

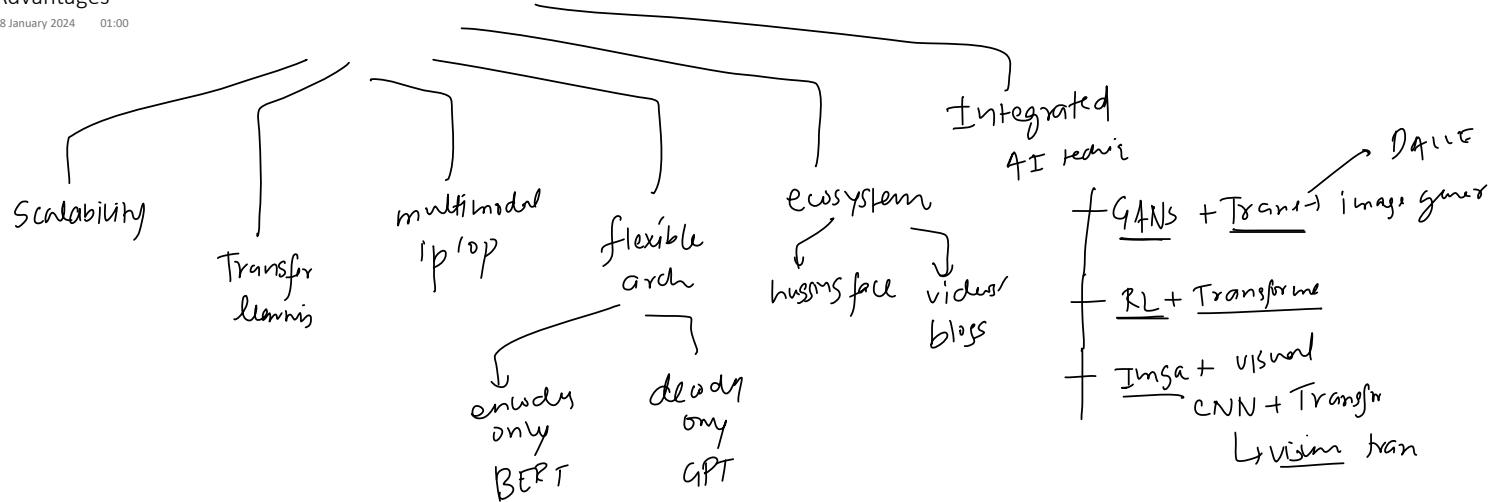
2018 - 2020 → Vision Transfer / AlphaFold-2

2021 → Gen AI

2022 → ChatGPT / Stable Diffusion

## Advantages

28 January 2024 01:00

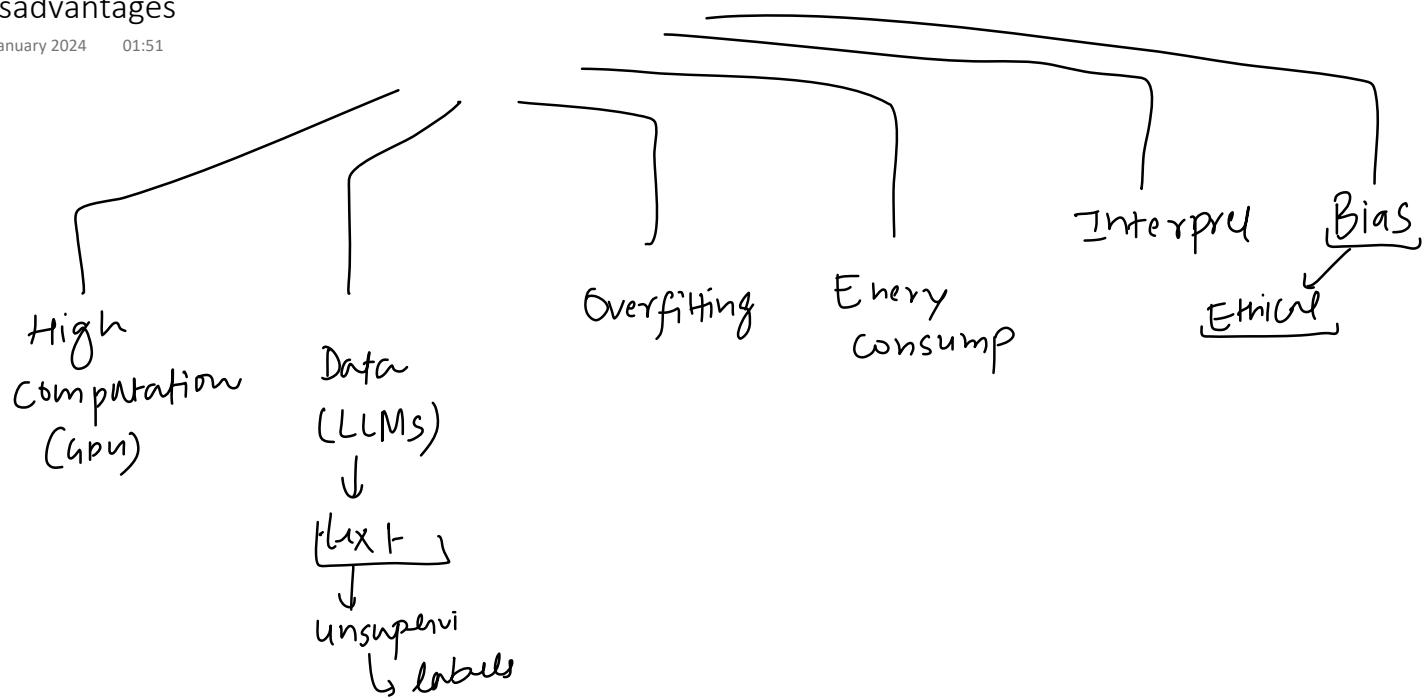


# Famous Applications

28 January 2024 01:17

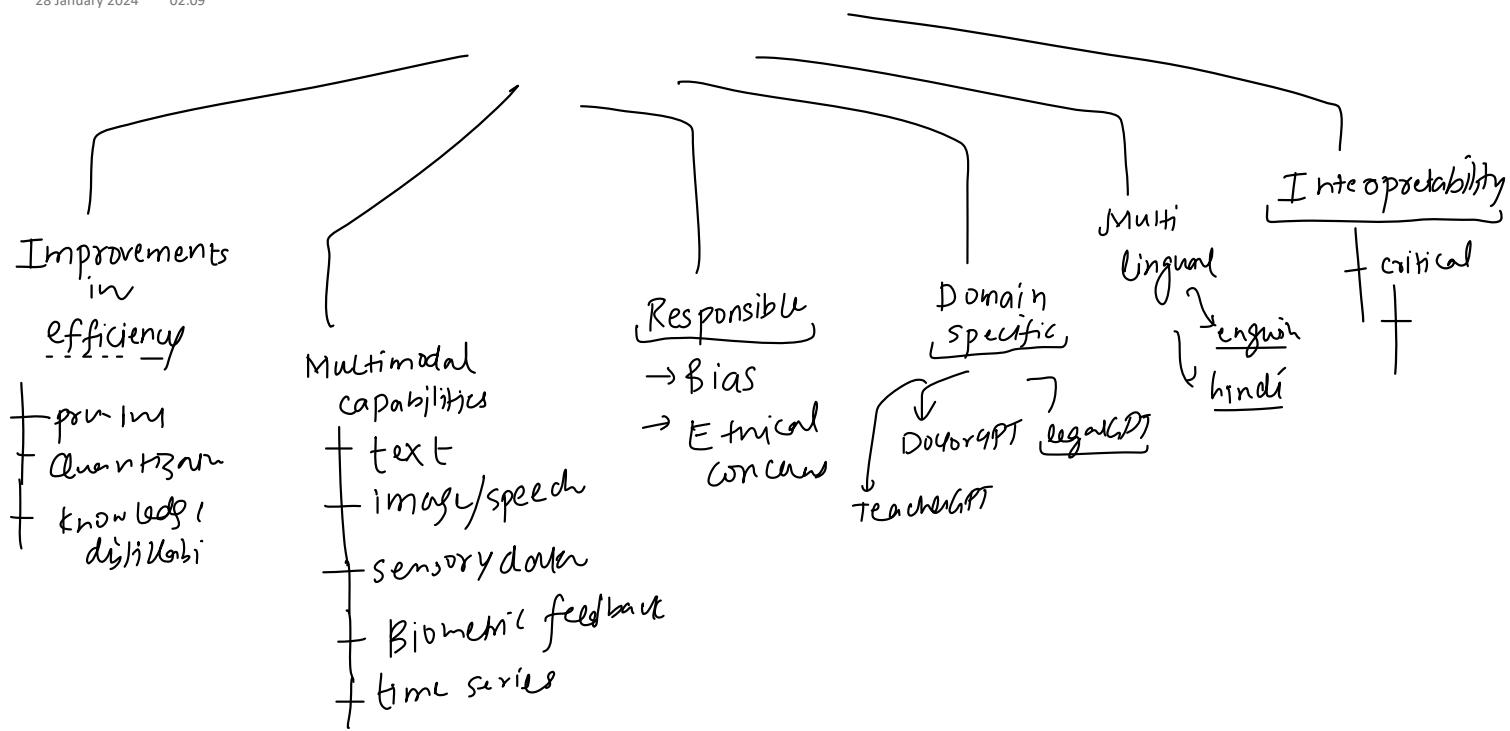
## Disadvantages

28 January 2024 01:51



# Future

28 January 2024 02:09



## The What

### Self attention

computer X

NLP → words → numbers → vectorization

OHE	mat	cat	mat
1)	mat	cat	mat
2)	cat	rat	rat

num → [1 0 0] [0 1 0] [1 0 0]

Bow  
S1  
S2

mat    rat    cat  
[2        0        1]  
→ [0        2        1]

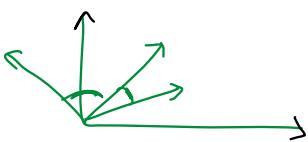
	mat	cat	rat
mat	1	0	0
cat	0	1	0
rat	0	0	1

Tf Idf

word embedding

number

semantic meaning



training with → [0 0 0 0 0] → [0 0 0 0 0] 5 dim

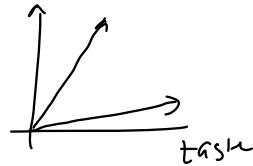
256, 512  
64

cricket

[0.2 0.9 0.9 0.9 0.9] n-dim vector

king → [0.9 0.1 1 0 0.9]  
similar  
queen → [0.9 0.2 0.4 1 0] royal

meaning → vector



Apple → vector

The problem of "Average Meaning"

- 1) An apple a day keeps the doctor away
  - 2) Apple is healthy
  - 3) Apple is better than orange
  - 4) Apple makes great phones
  - ⋮
  - ⋮
- 1000 words
- 1000 phone
- 9000 fruit
- 9000 tech
- 10000 term

dataset



0 0 0 0 → [0 0 0 0] 2 dim

x y  
task technology

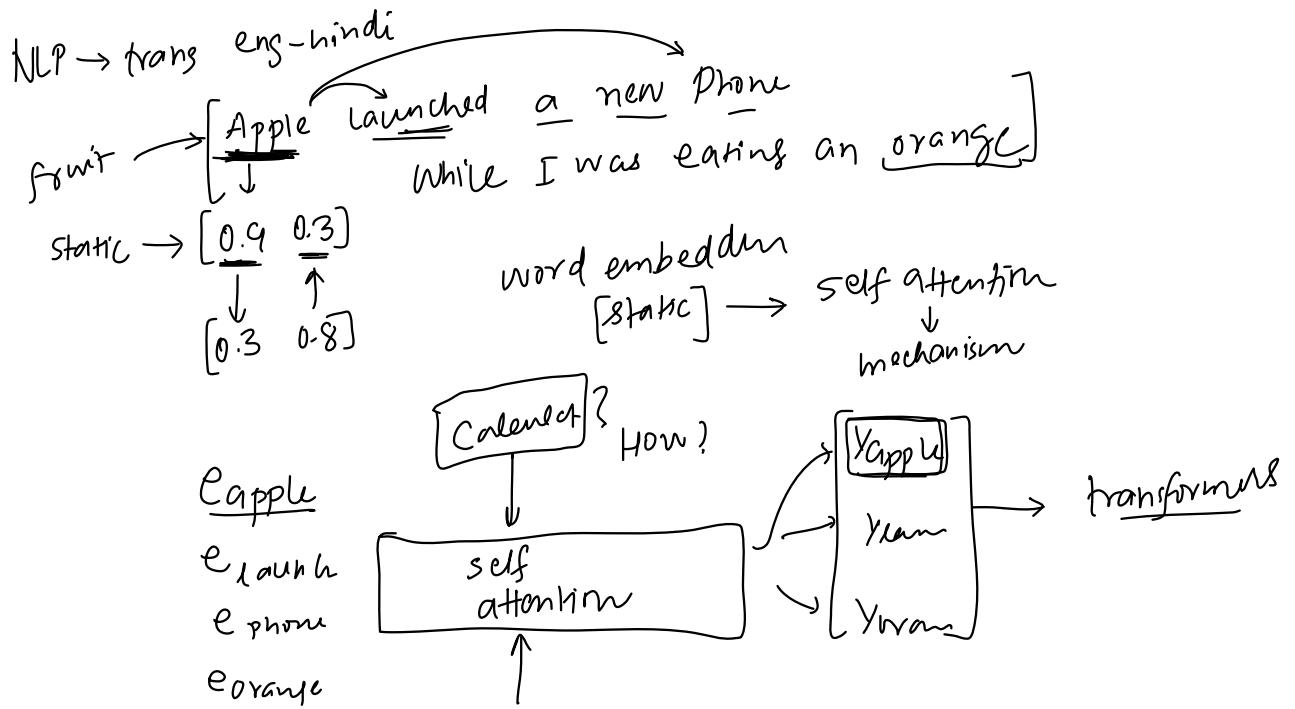
[0.8 0.2] → [0.9 0.3]

word embeddings

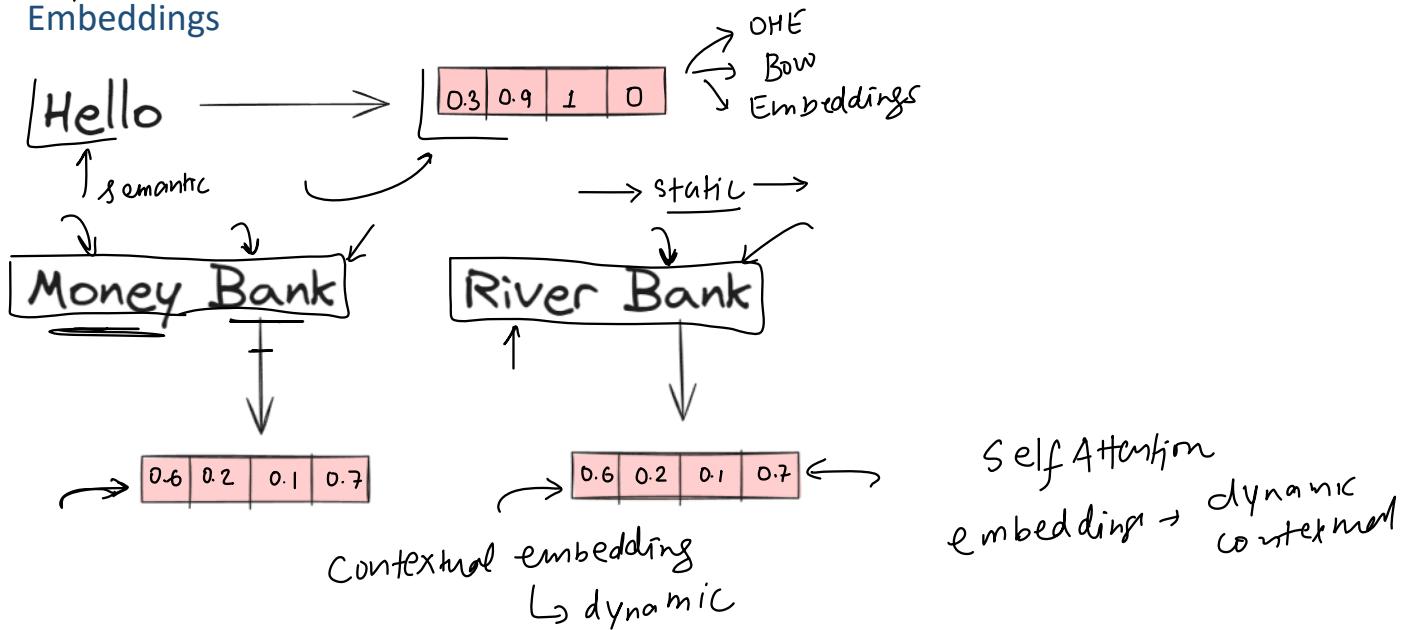
create → use → static

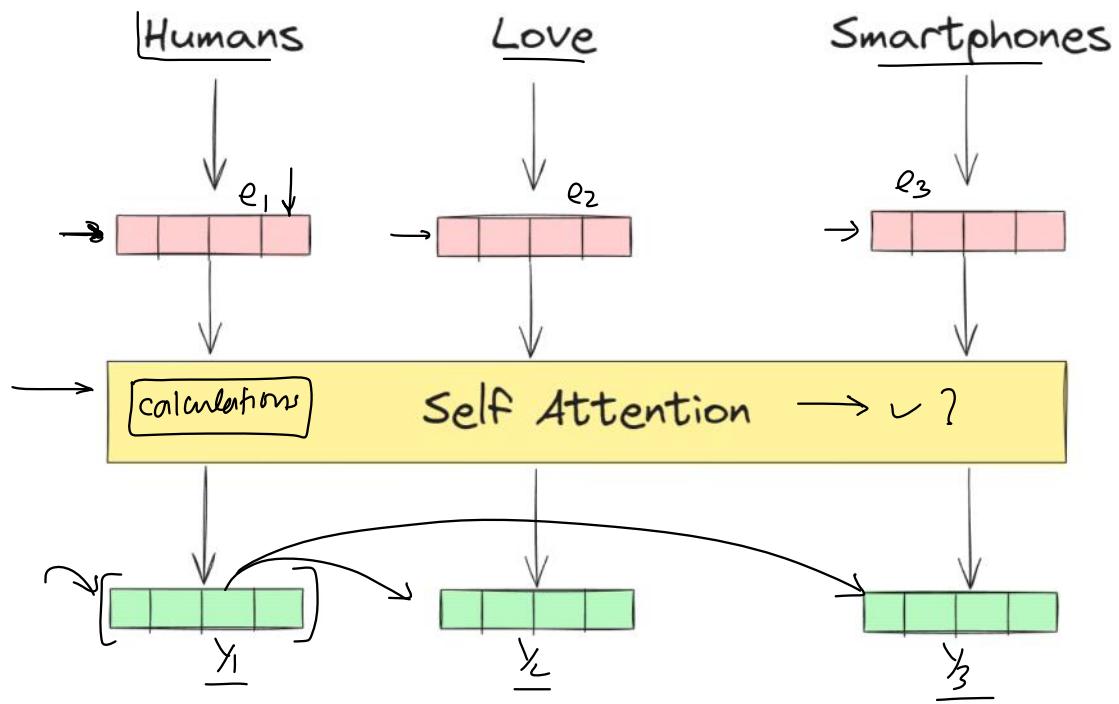
static embedding → [8 mark-  
contextual embedding]

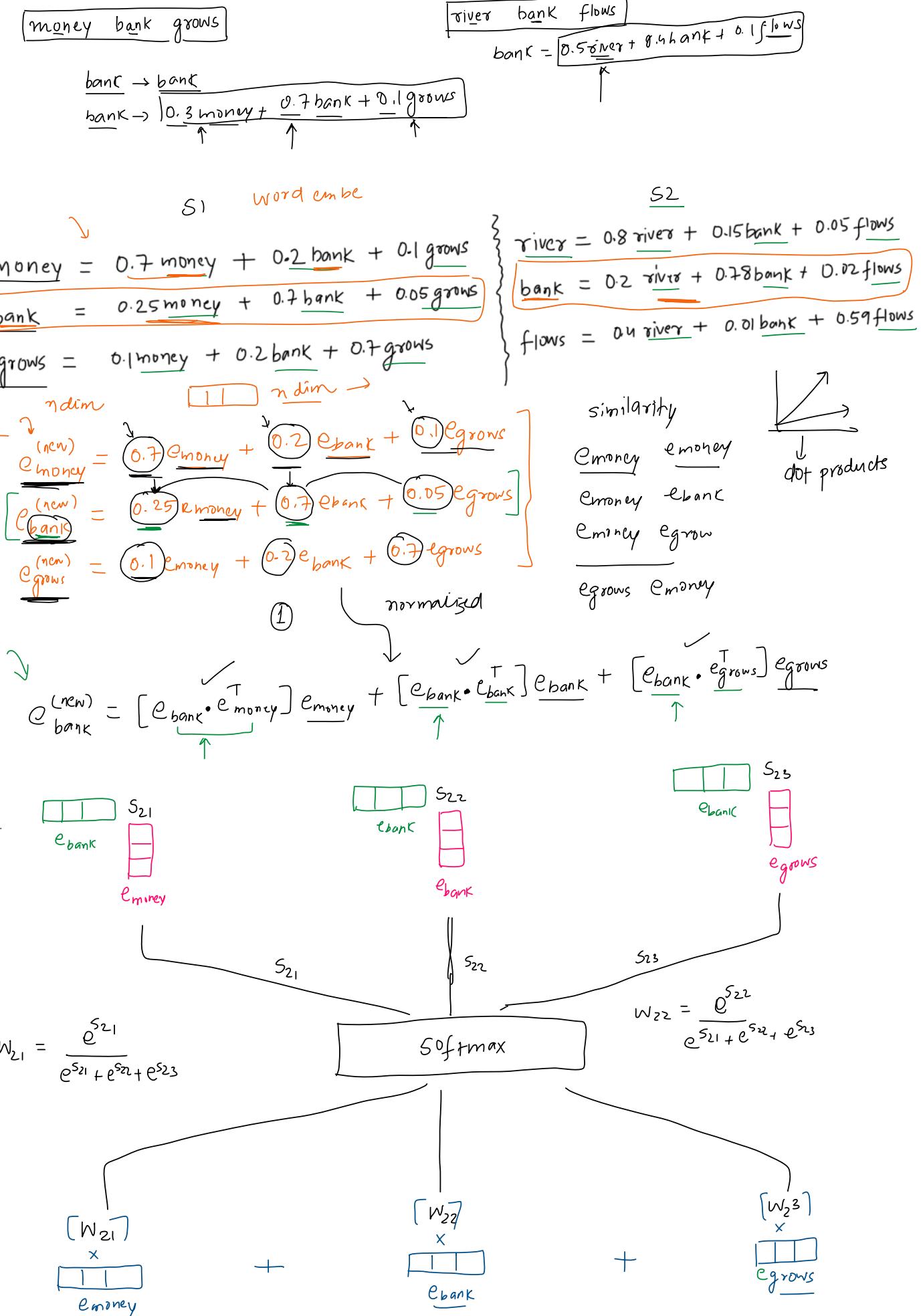
110 - drane eng-hindi

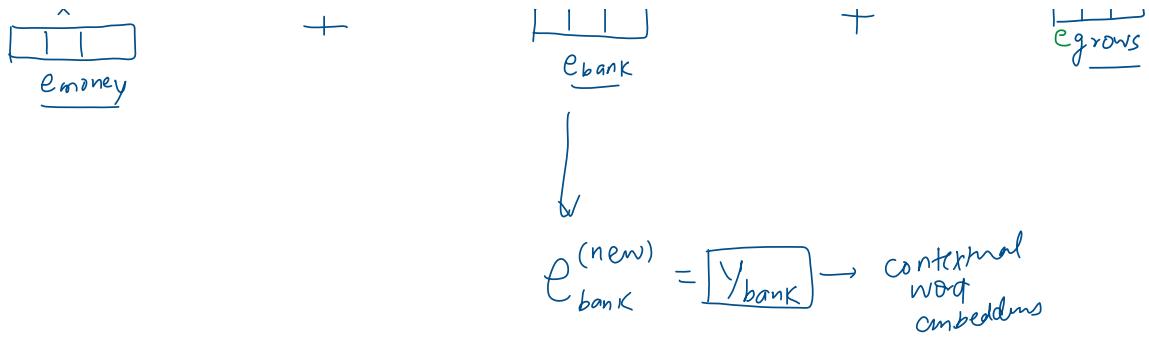


## Embeddings



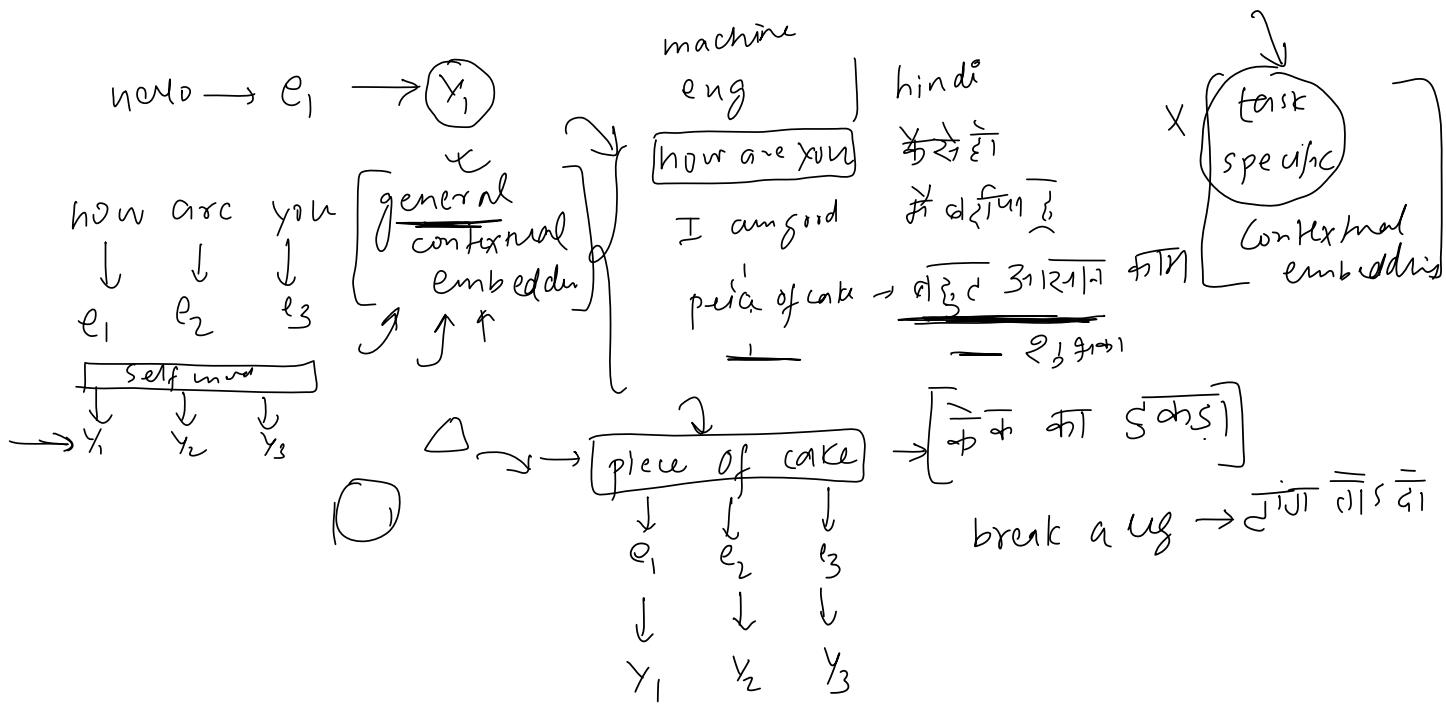
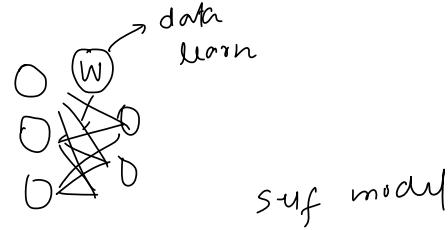






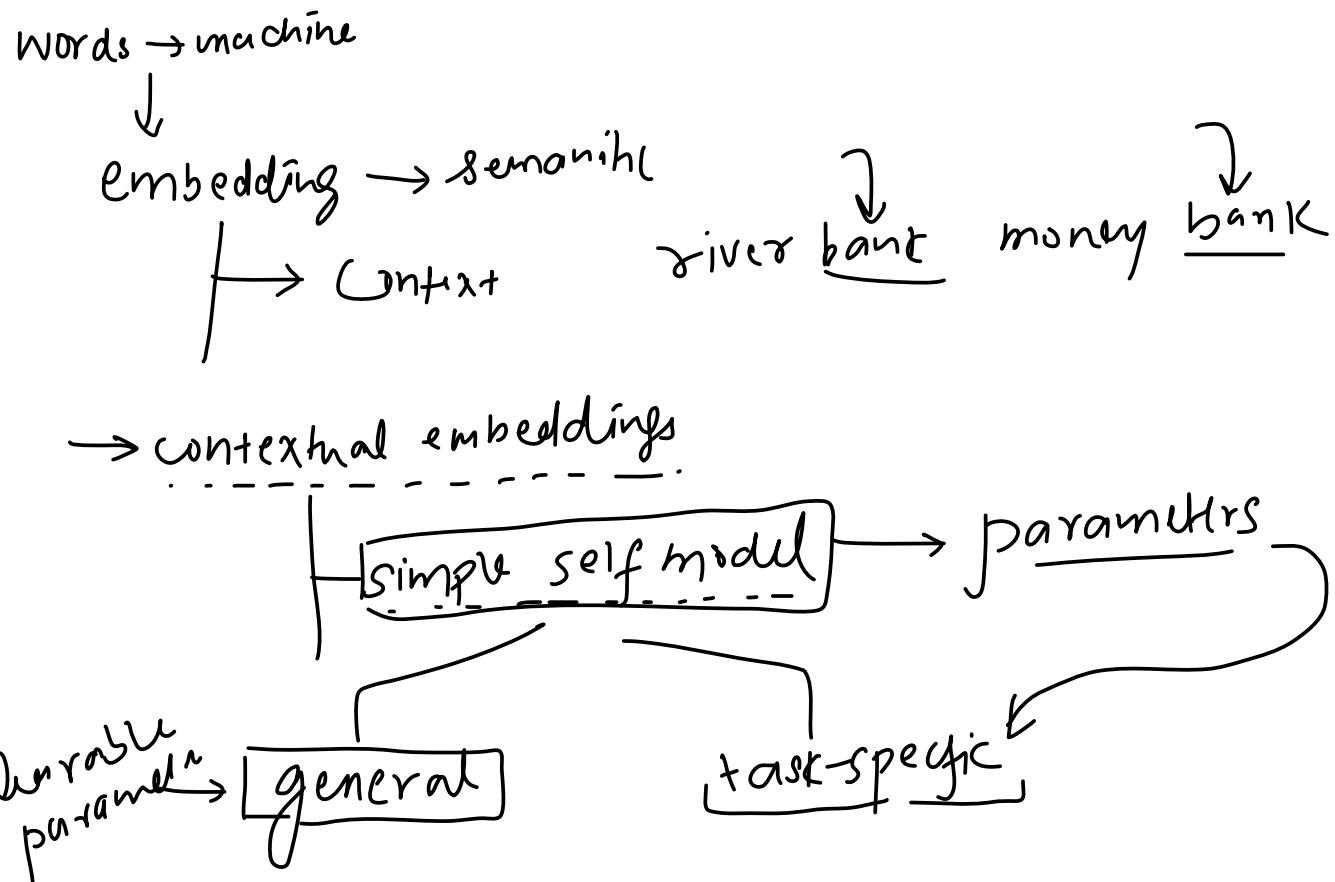
### Points to consider

- This operation is a parallel operation
- There are no parameters involved



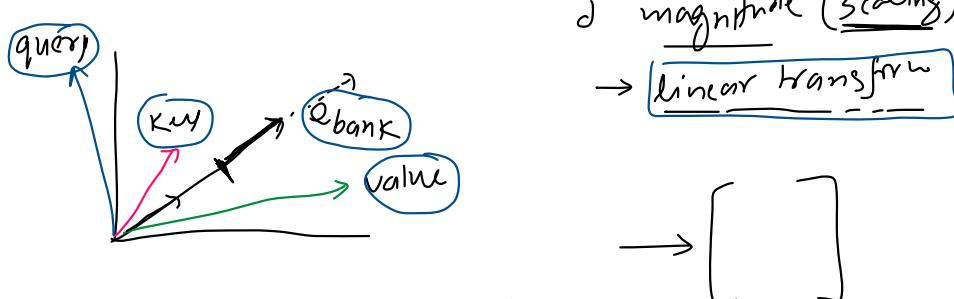
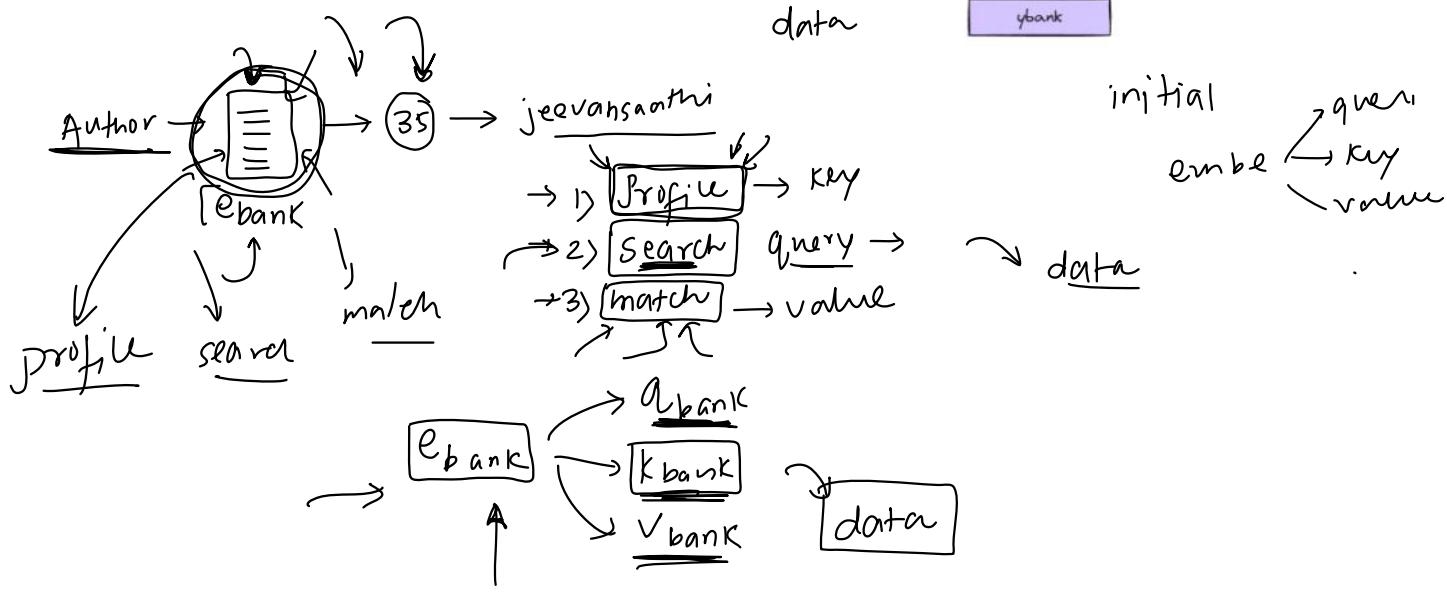
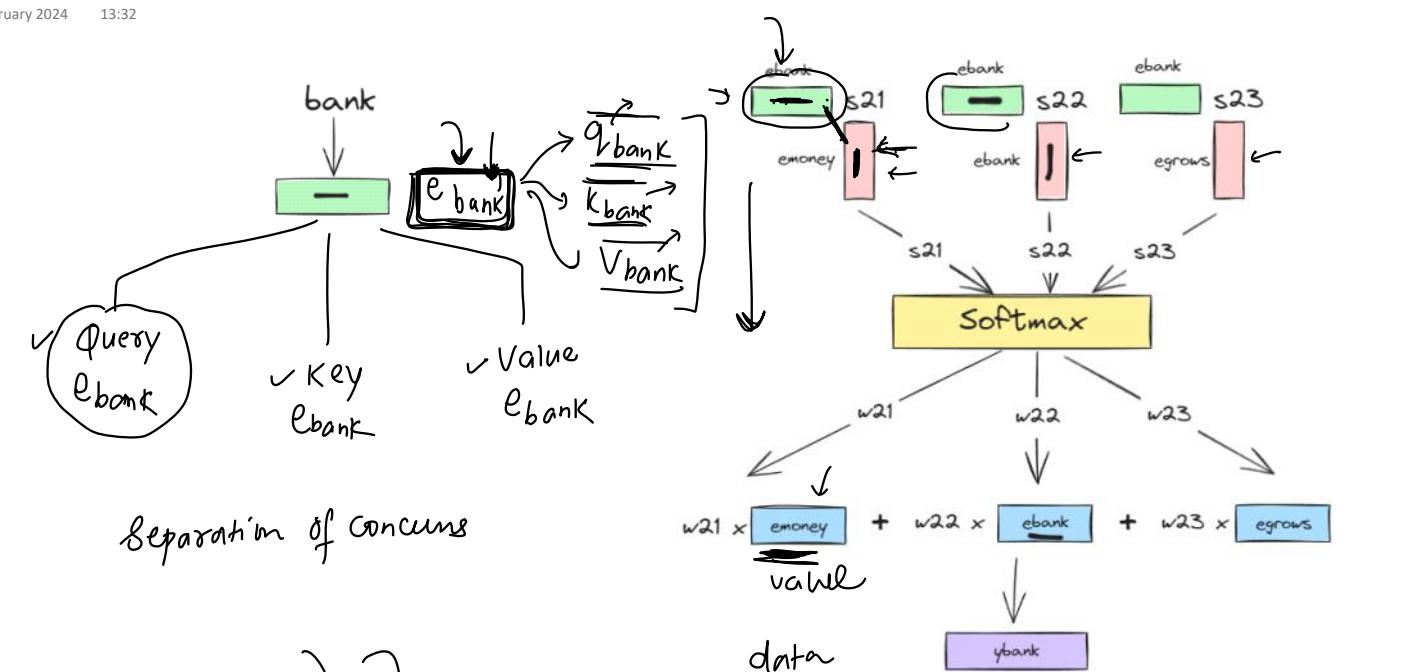
# Progress

06 February 2024 00:42



## Query, Key & Value Vectors

06 February 2024 13:32

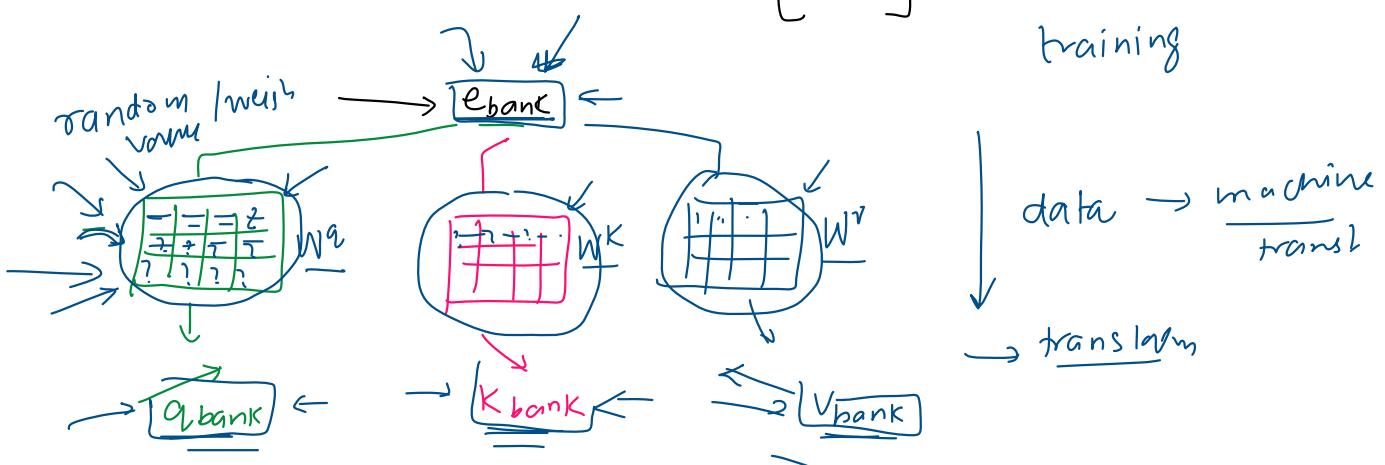


d magnitude (Scaling)

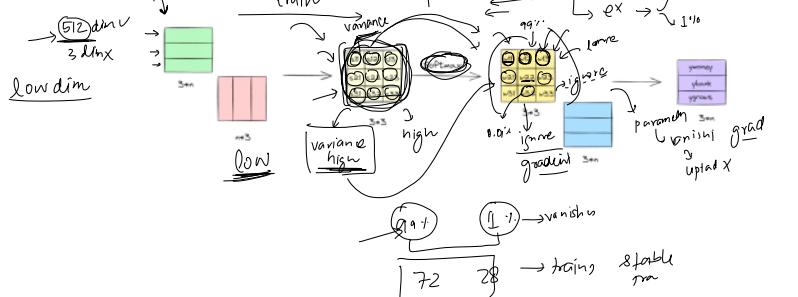
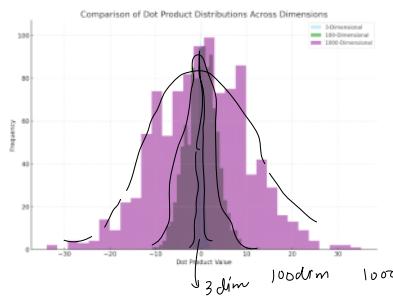
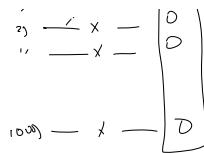
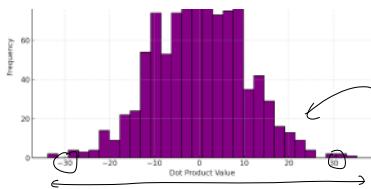
linear transfrm

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

training





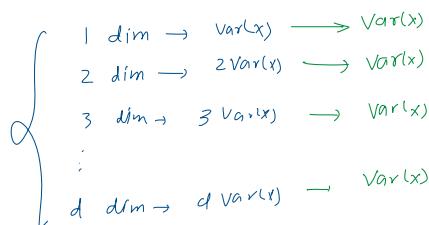
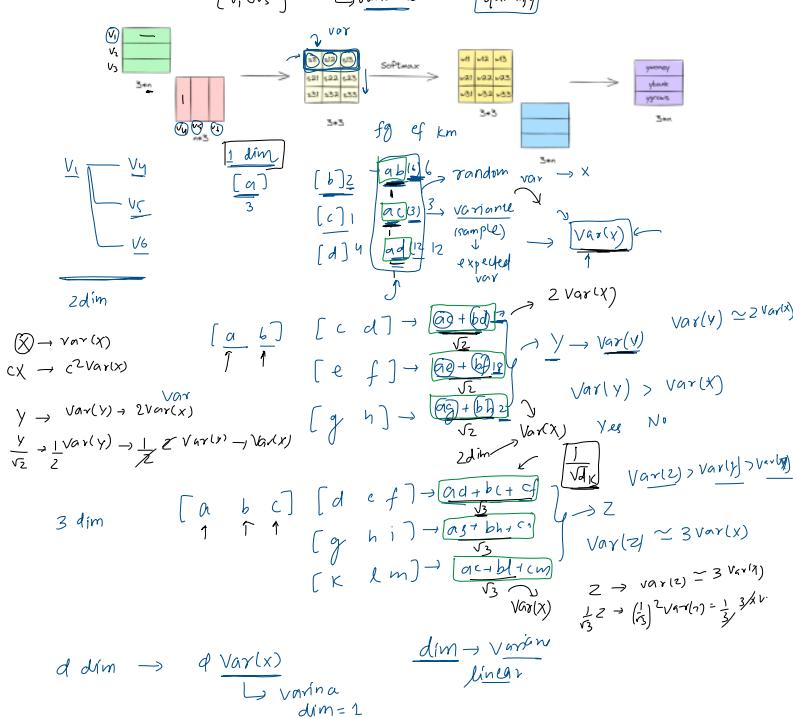


cancel c

$$\begin{cases} X \rightarrow \text{Var}(X) \\ Y \rightarrow cX \rightarrow c^2\text{Var}(X) \end{cases}$$

If you have a random variable  $X$  with a variance of  $\text{Var}(X)$ , and you create a new variable  $Y$  by scaling  $X$  with a constant  $c$ , so that  $Y = cX$ , the variance of  $Y$  ( $\text{Var}(Y)$ ) is related to the variance of  $X$  by the square of the scaling factor  $c$ . Mathematically, this relationship is expressed as:

$$\text{Var}(Y) = c^2\text{Var}(X)$$

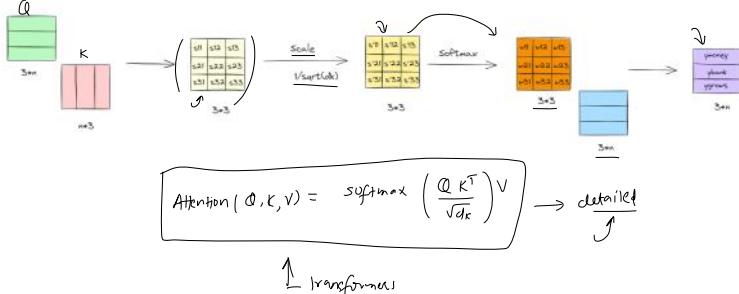


$$[\underbrace{\dots}_{d \text{ dim}}] \rightarrow \underbrace{\text{Var}(x)}_{\text{Varina dim=1}}$$

$\rightarrow$

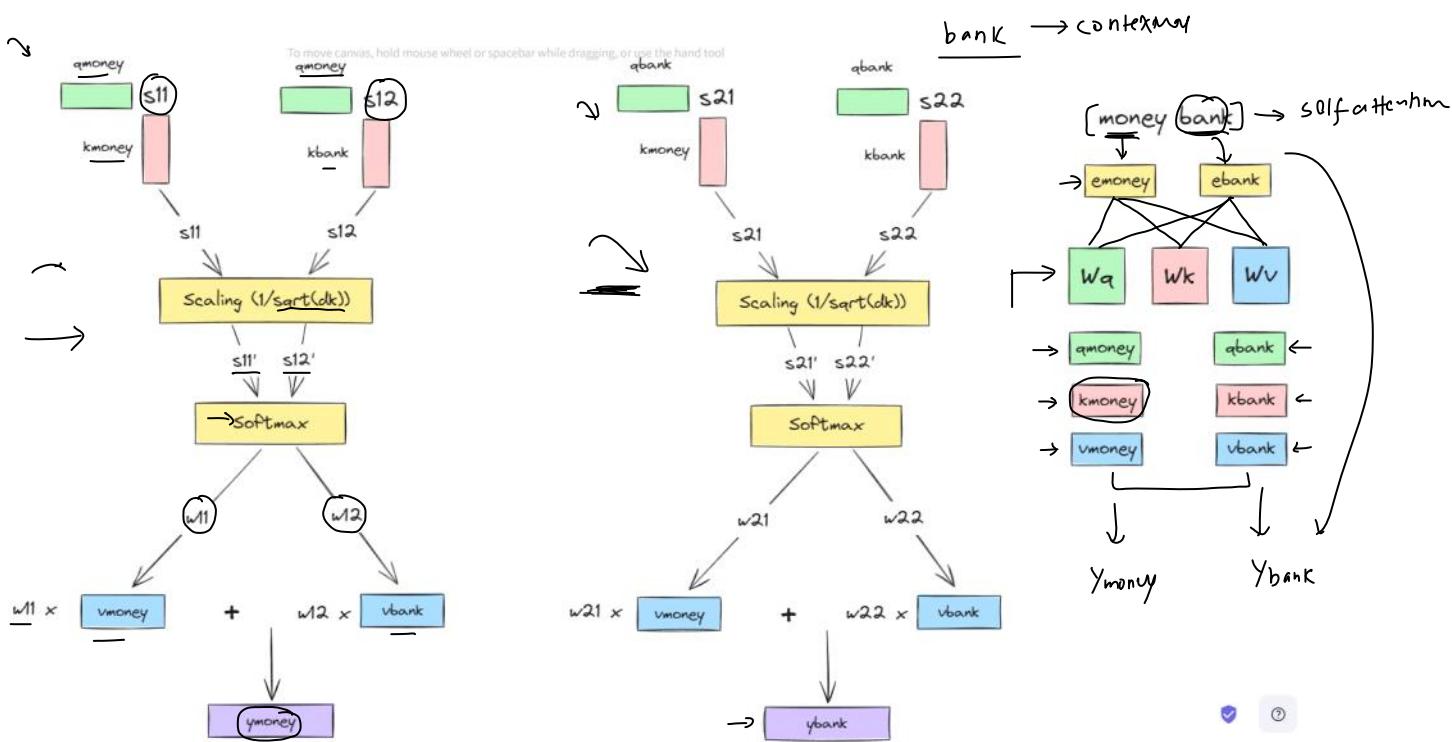
7.

$\sim \dots$



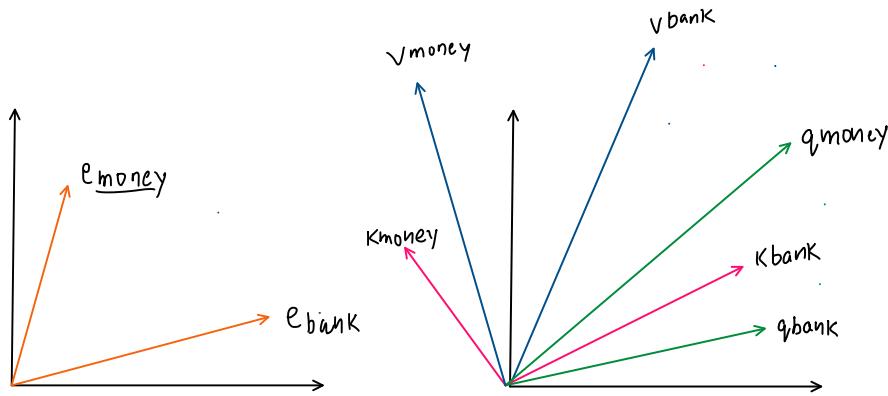
# What is $d_k$

28 February 2024 16:59



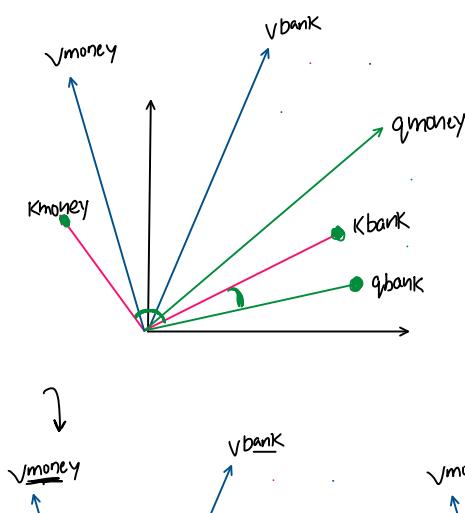
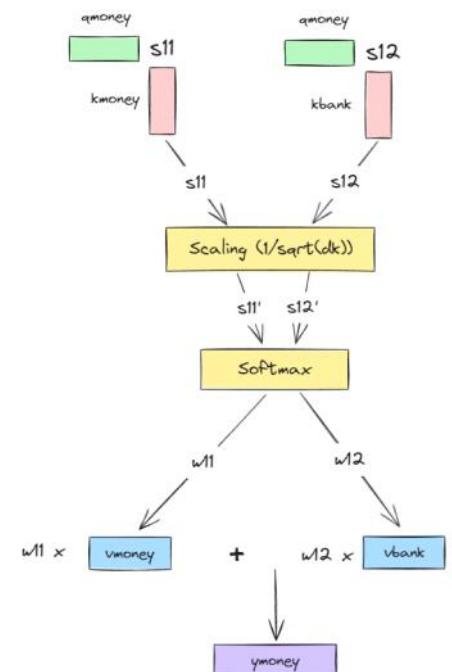
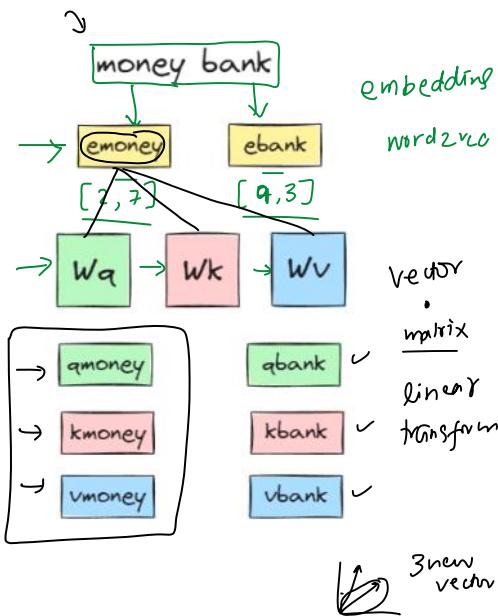
# Geometric Intuition

08 March 2024 15:16



$$Wq = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad Wk = \begin{bmatrix} 3 & 4 \\ 5 & 1 \end{bmatrix} \quad Wv = \begin{bmatrix} 4 & 1 \\ 2 & 1 \end{bmatrix}$$

\* All values are hypothetical



[Dot Product]

$$S_{21} = 10$$

$$S_{22} = \frac{32}{\sqrt{2}}$$

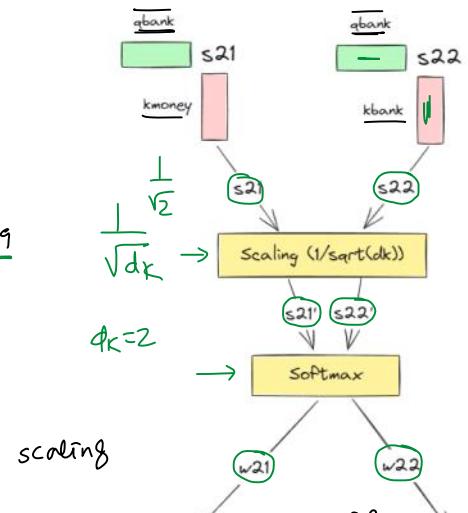
[Scaling]

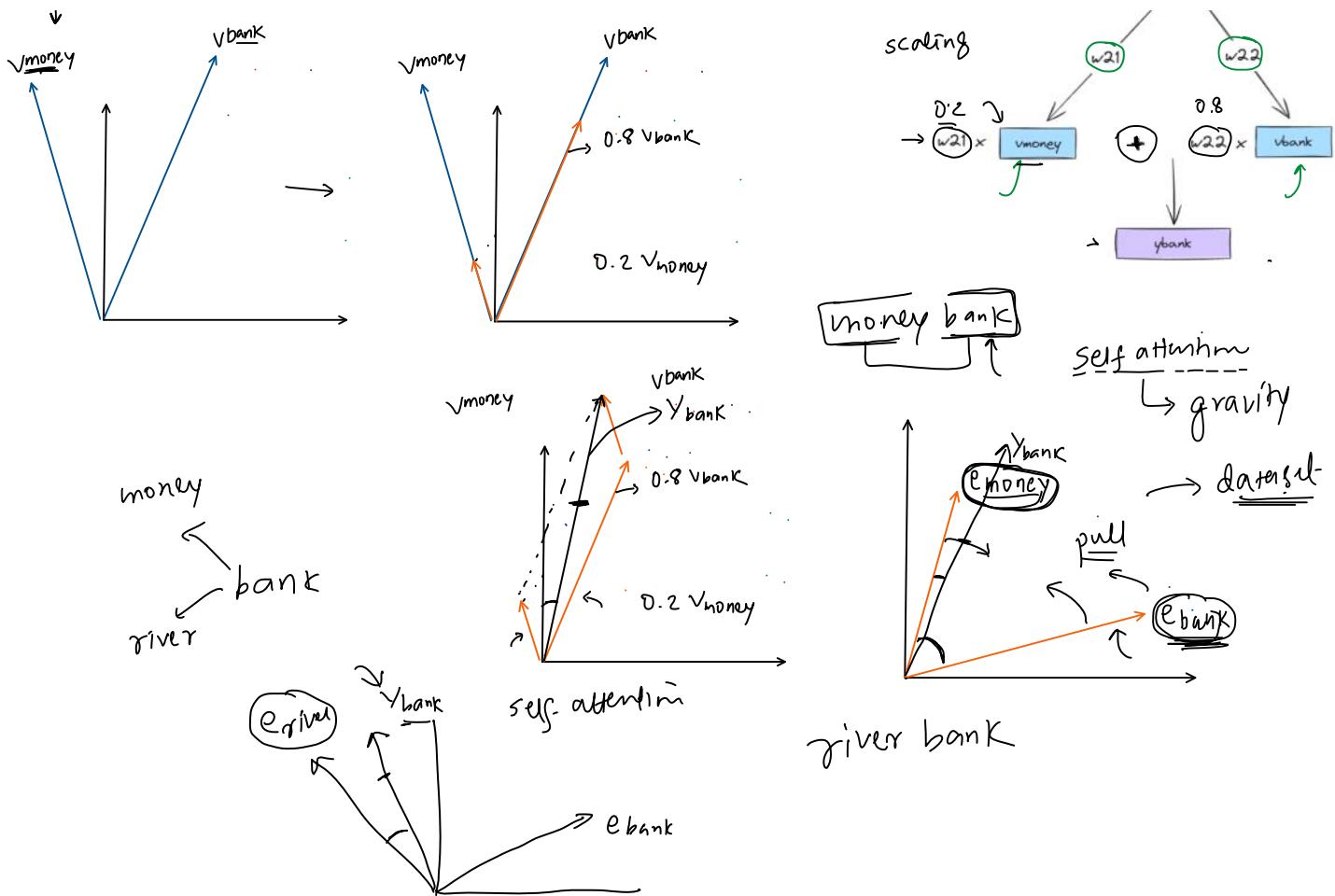
$$S_{21}' = \frac{10}{\sqrt{2}} = 7.09 \quad S_{22}' = 22.69$$

[Softmax]

$$w_{21} = 0.2$$

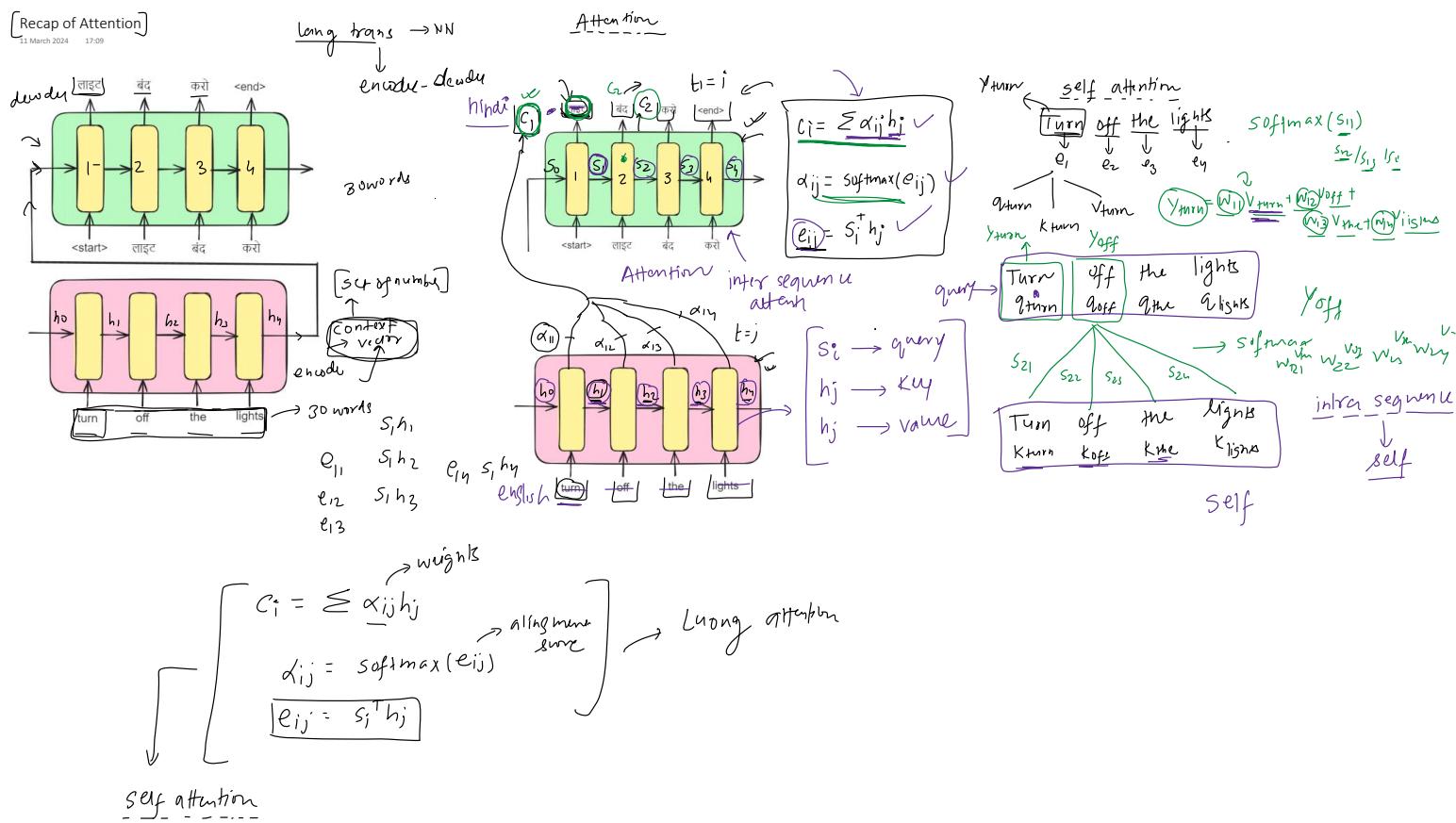
$$w_{22} = 0.8$$





### Recap of Attention

11 March 2024 17:09

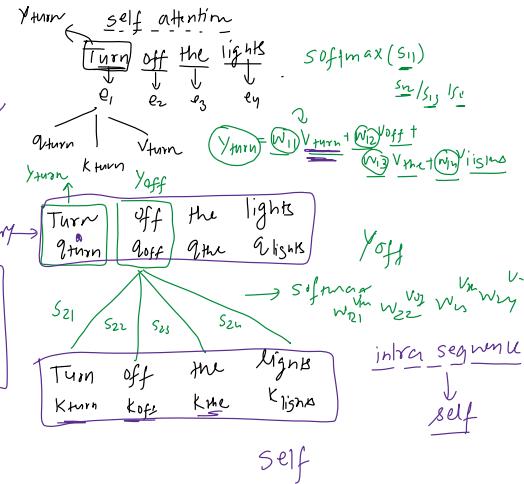
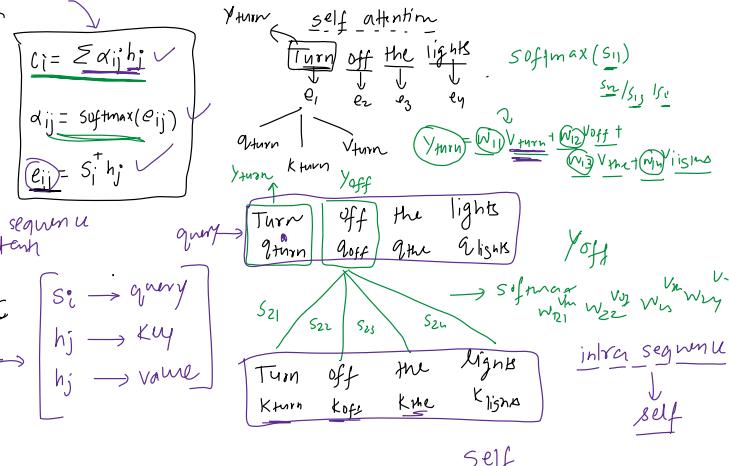


self attention

$$c_i = \sum \alpha_{ij} h_j$$

$$\alpha_{ij} = \text{softmax}(e_{ij})$$

$$e_{ij} = s_i^T h_j$$

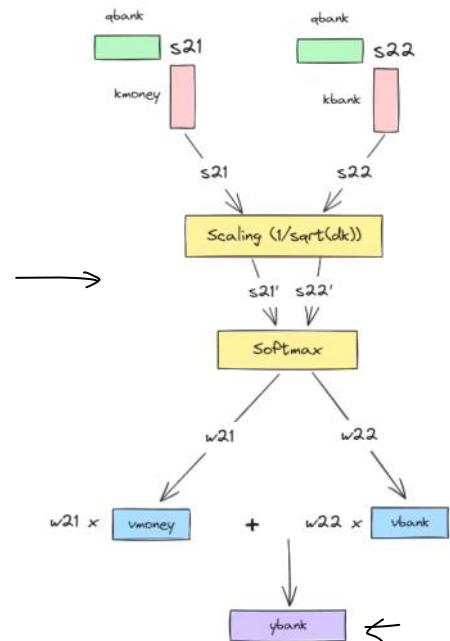
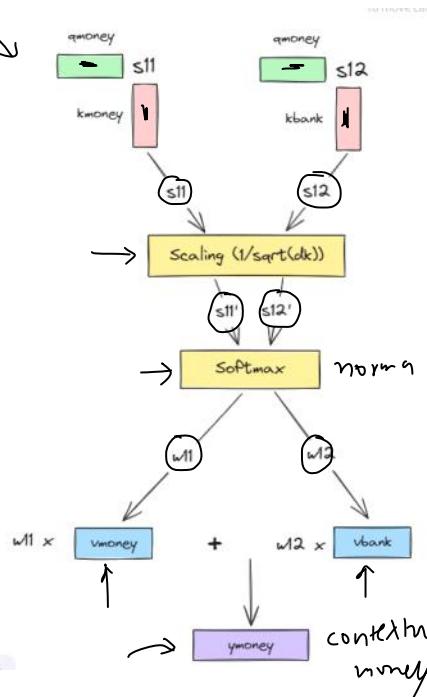
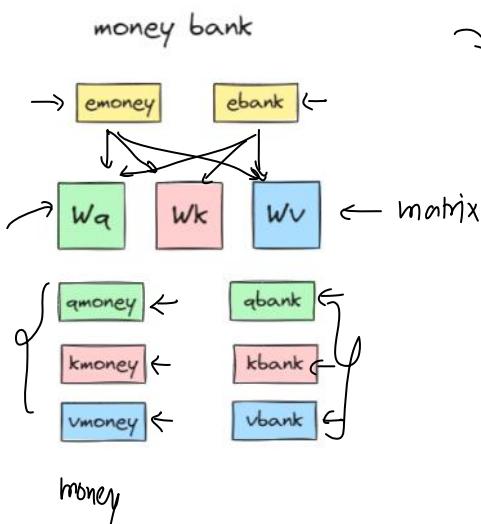
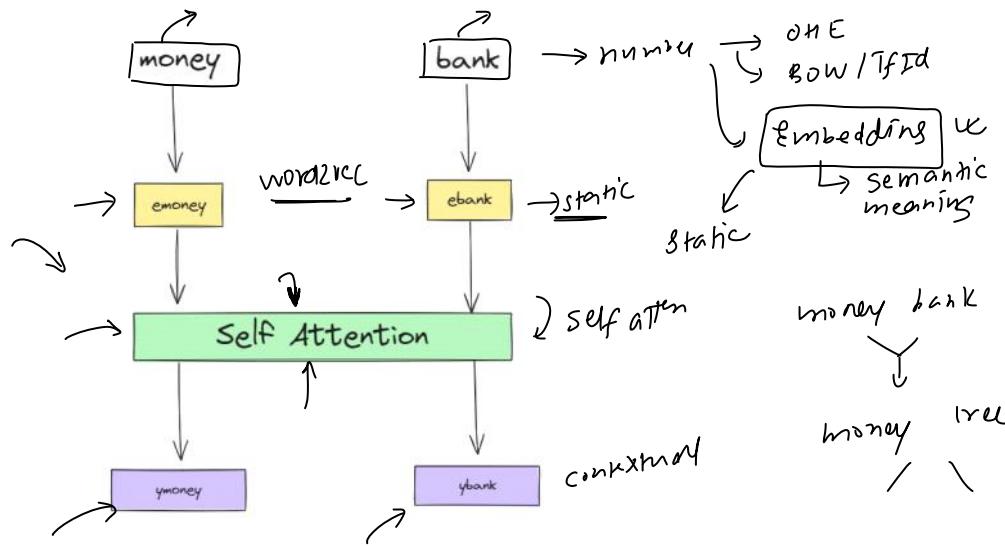
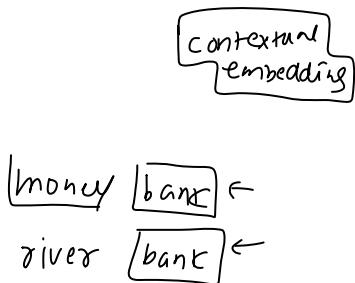


self

intra-sequence  
self

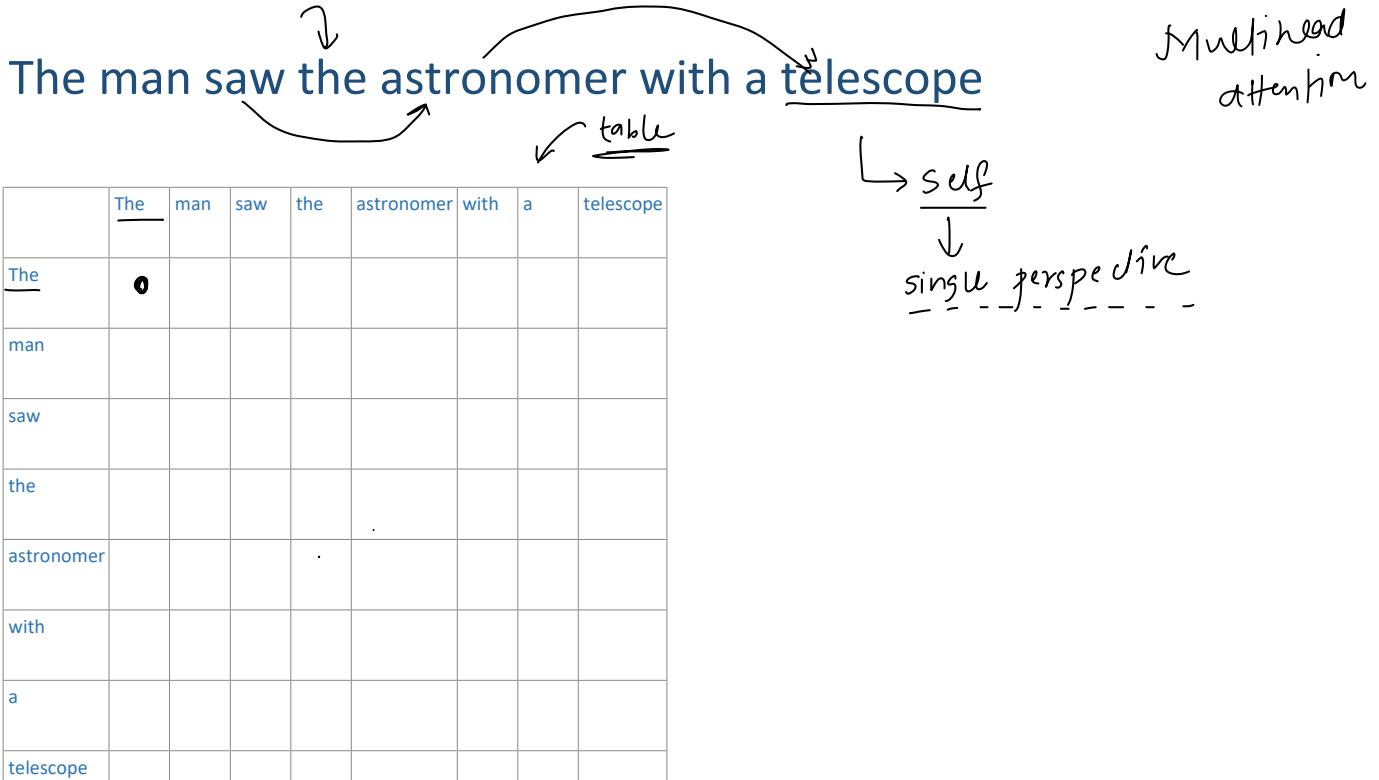
## Recap of Self Attention

08 April 2024 14:46



## Problem with Self Attention

08 April 2024 14:47



doc summarization

Self attention

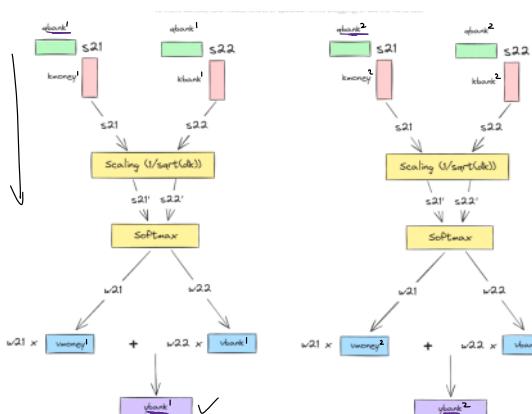
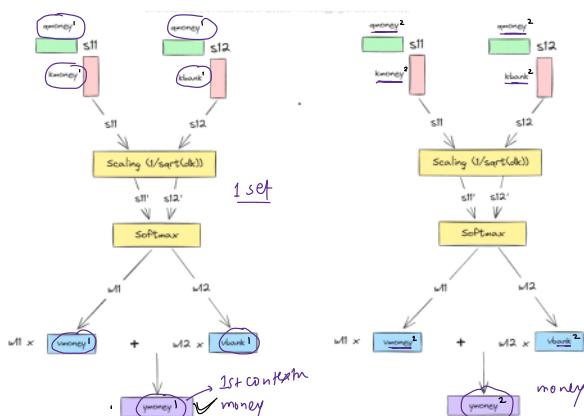
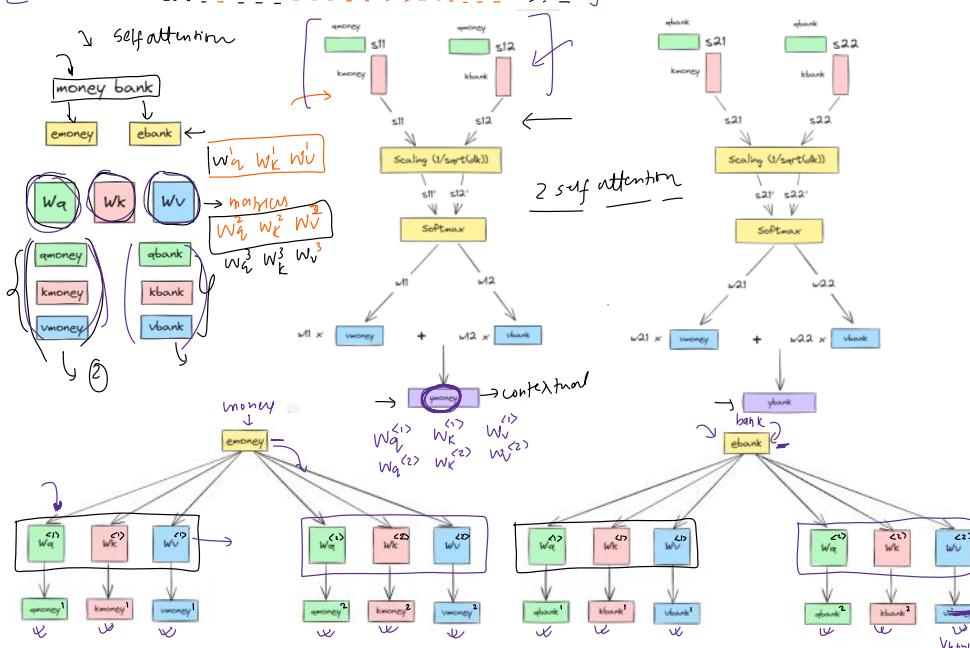
The future of AI in India presents a dynamic and promising landscape, marked by rapid advancements and a burgeoning ecosystem of innovation. With a robust talent pool of engineers and IT professionals, India is poised to become a significant player in the global AI arena. The government's proactive stance on AI, exemplified by initiatives like the National Strategy for Artificial Intelligence, aims to harness AI's potential across various sectors, including healthcare, education, agriculture, and urban infrastructure. Indian startups and tech giants are increasingly incorporating AI to solve complex societal challenges, improve efficiency, and enhance service delivery. Moreover, India's focus on ethical AI and data security aims to create a sustainable and responsible growth trajectory. As AI becomes more integrated into daily life and industry, India's unique blend of technological prowess, entrepreneurial spirit, and societal needs will likely shape a distinctive path in the AI domain, fostering innovation that is not only technologically advanced but also socially inclusive and impactful.

India is poised to become a key player in the AI domain, leveraging its skilled workforce and government initiatives to apply AI across various sectors like healthcare and education. With a focus on innovation, ethical AI, and data security, India aims to integrate AI to address societal challenges, enhance efficiency, and promote inclusive growth. This approach positions India to uniquely contribute to global AI advancements while ensuring sustainable and responsible development within its own borders.

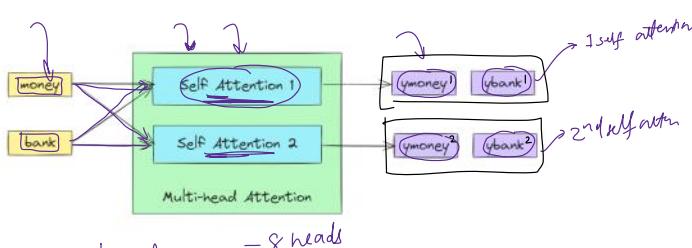
India's AI future holds promise as it harnesses a burgeoning talent pool and government initiatives to pioneer AI-driven innovation. With a focus on sectors like healthcare and education, India aims to leverage AI for societal development. The emphasis on ethical AI and data security underscores India's commitment to responsible technological advancement. This approach positions India not only as a global AI hub but also as a trailblazer in addressing societal challenges through cutting-edge technology.

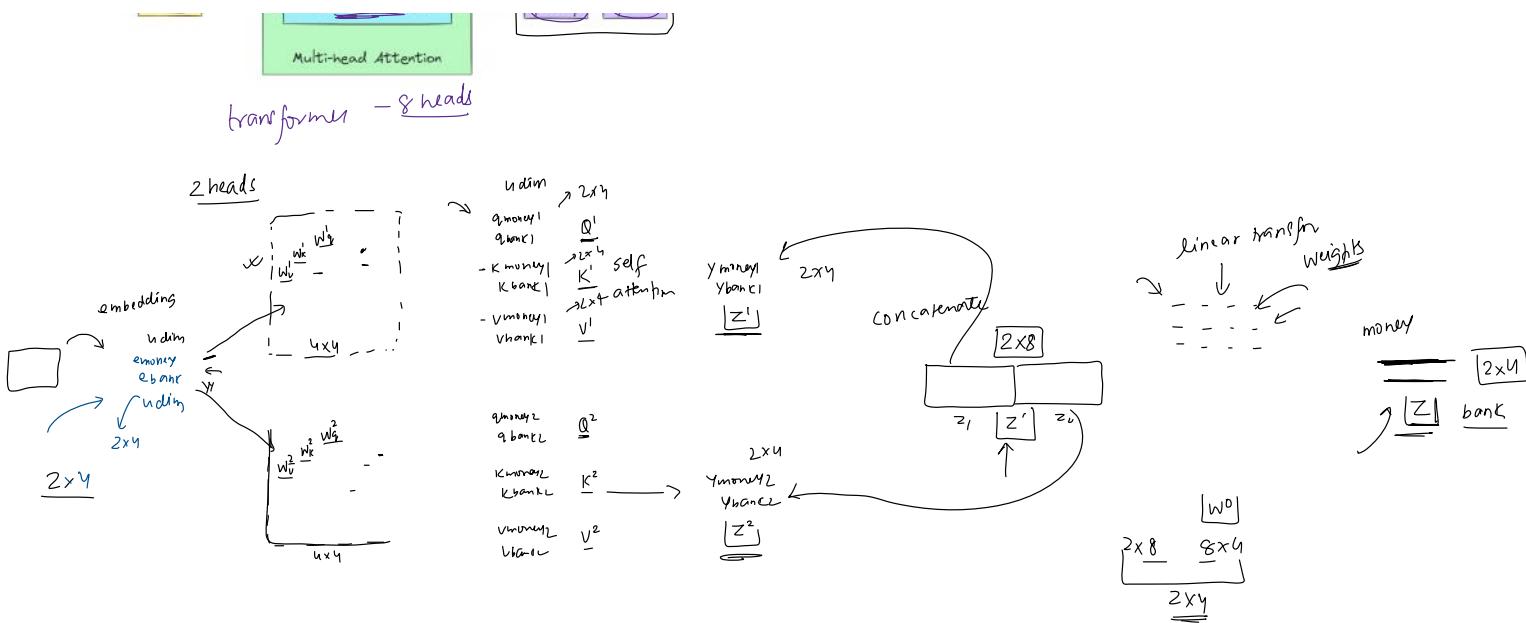
## The man saw the astronomer with a telescope

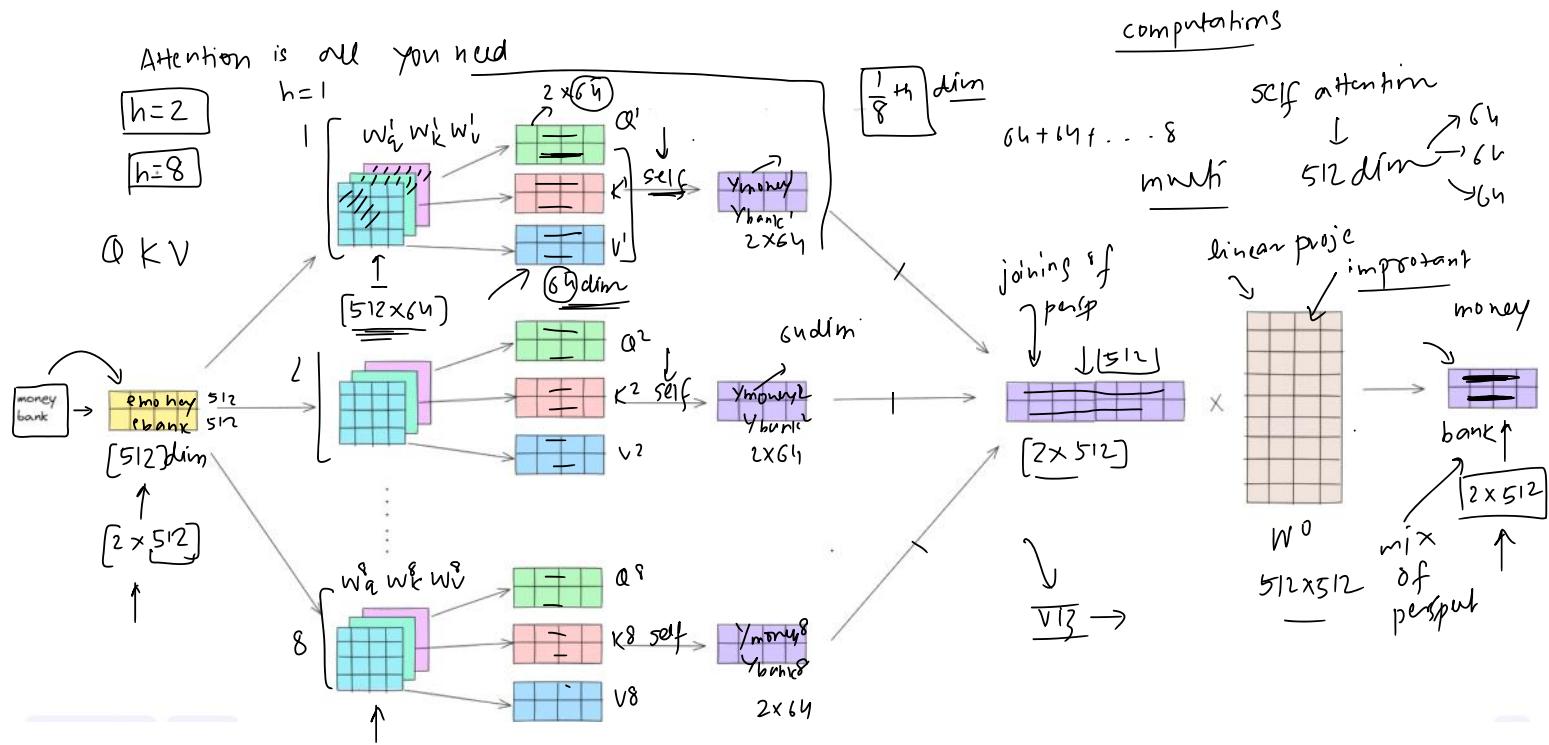
self attention → ②



mult



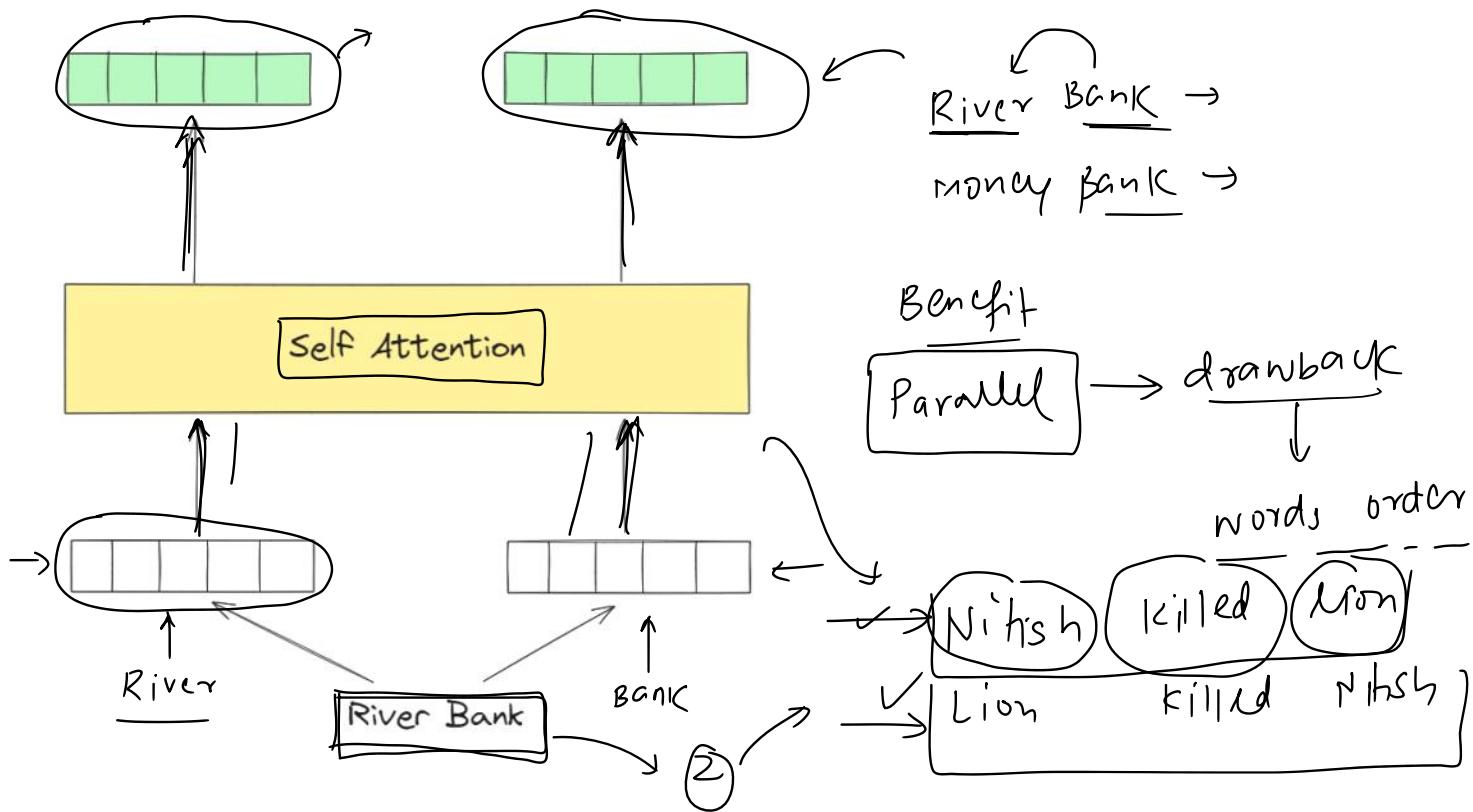




# The Why

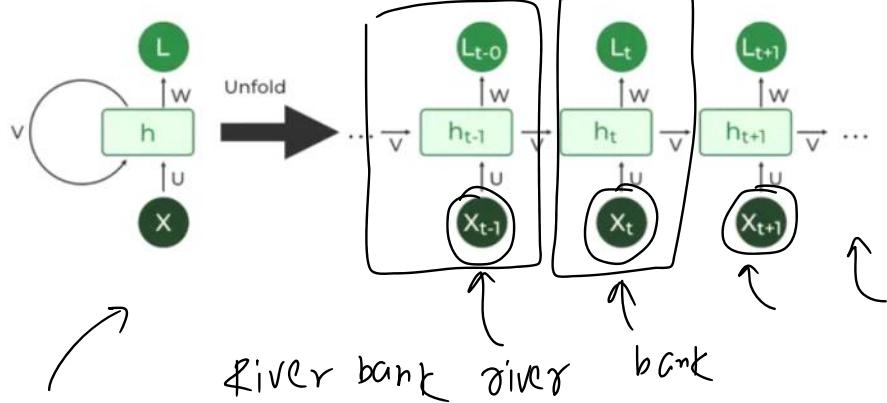
23 May 2024 14:33

## Self-Attention



RNN → sequential

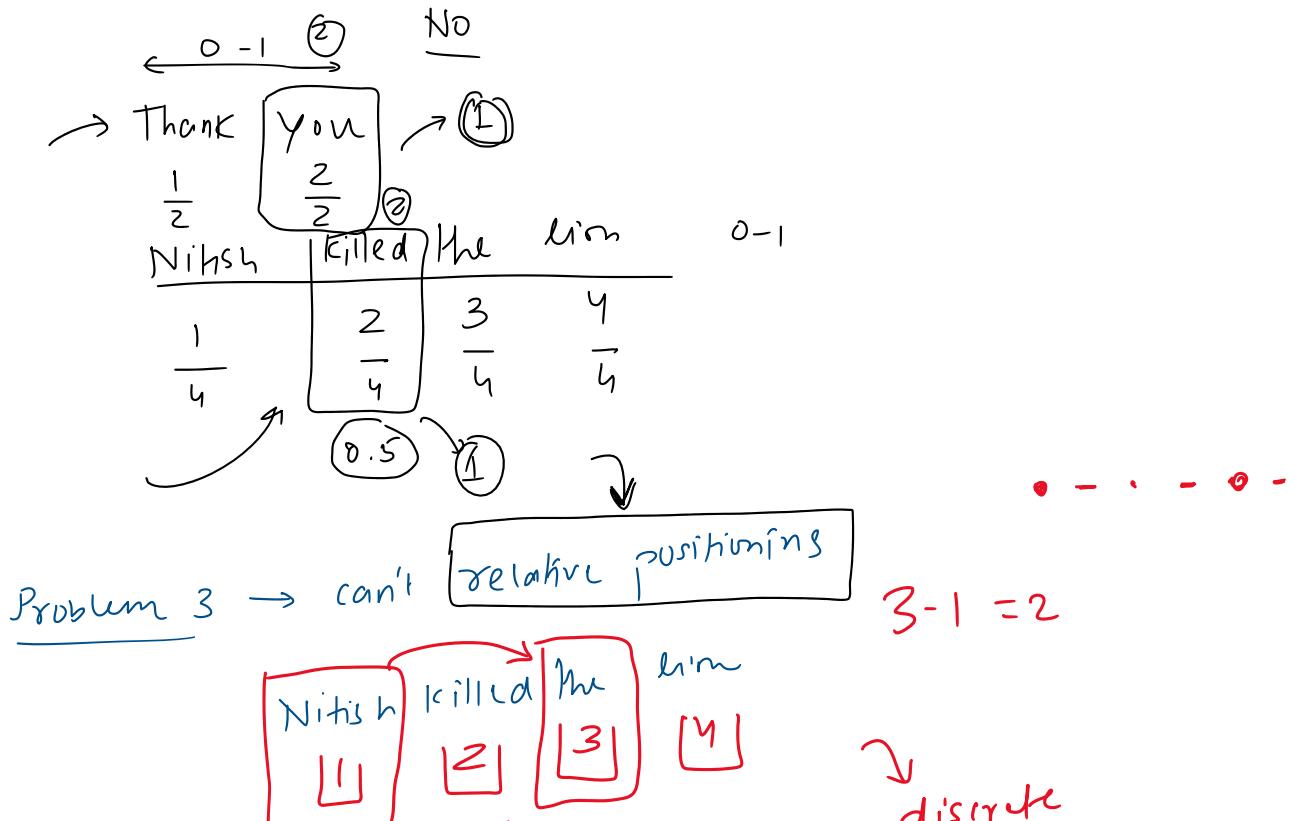
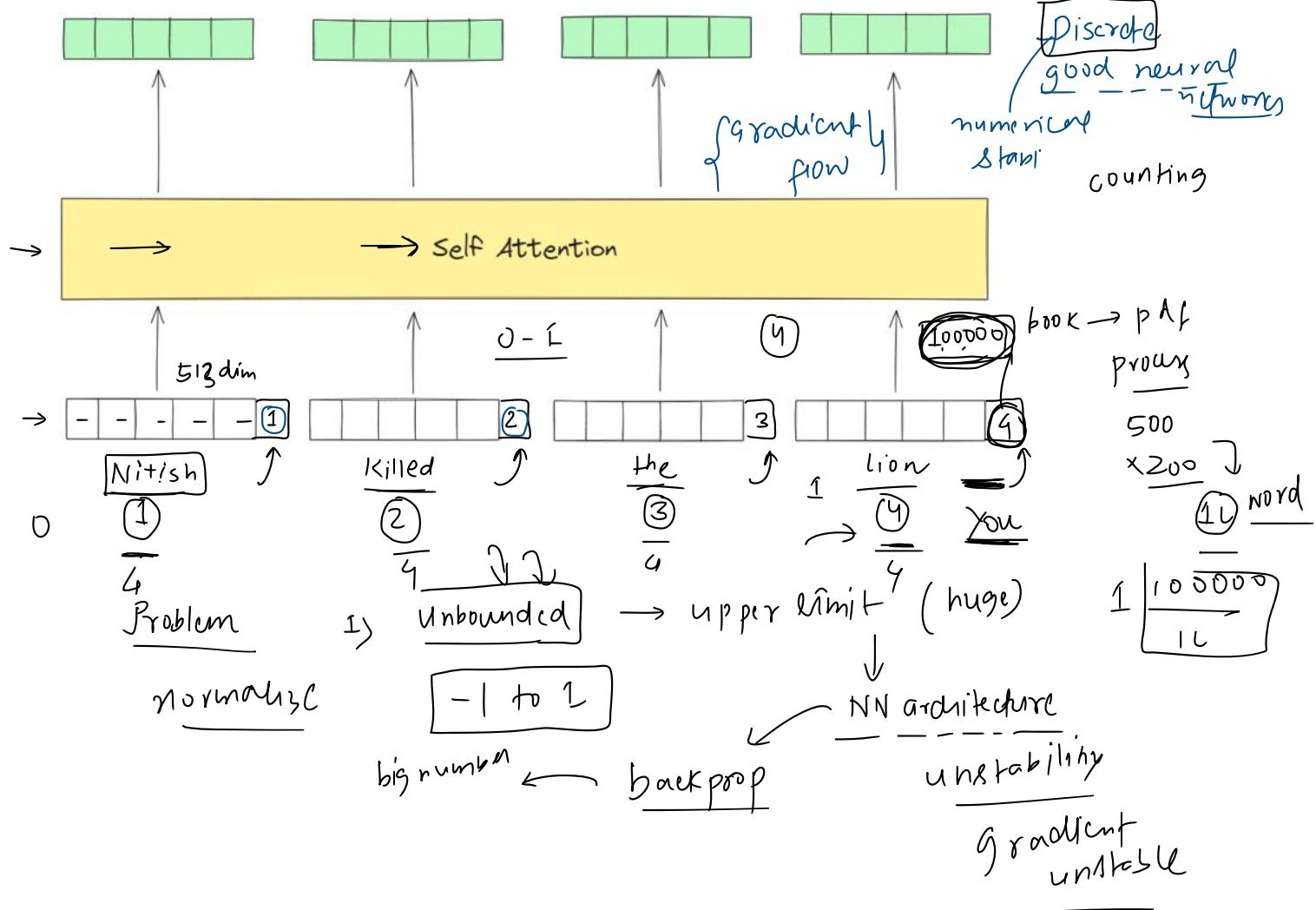
Nithish killed him

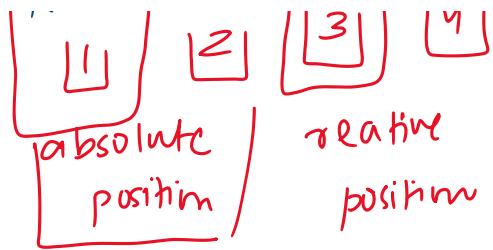


## Proposing a simple solution

23 May 2024 15:37

Smooth transition, continue



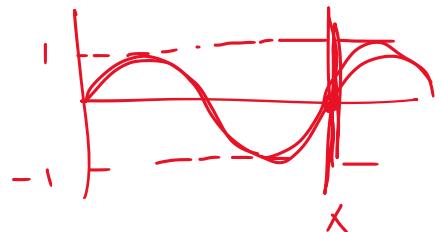


discrete  
↓  
periodic  
relative positioning

Problem

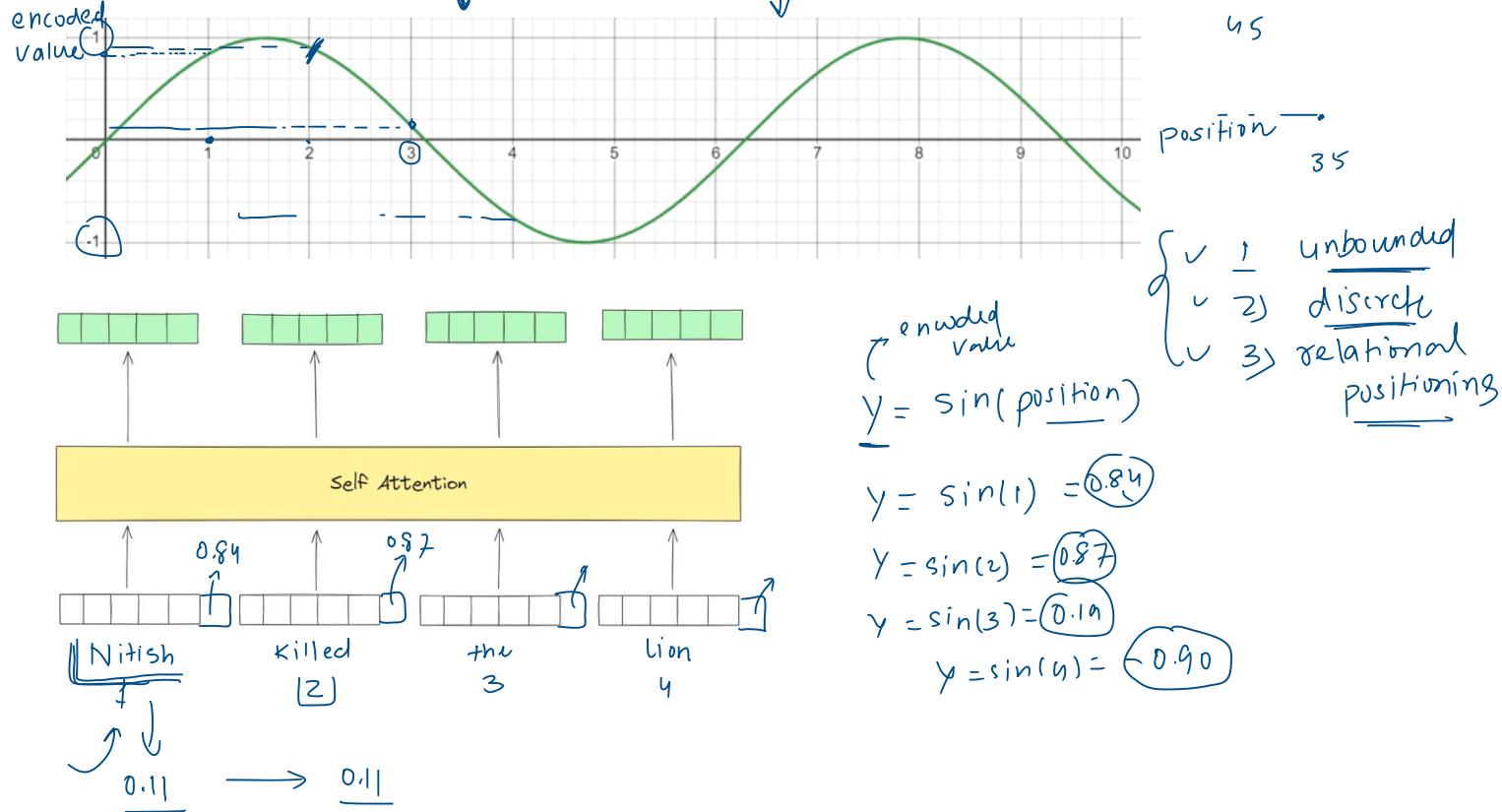
- unbounded ( bounded ) ✓✓
- discrete ( continuous ) → function →
- relative ( periodic ) ✓

positional encoder → sine ↓  
better solution

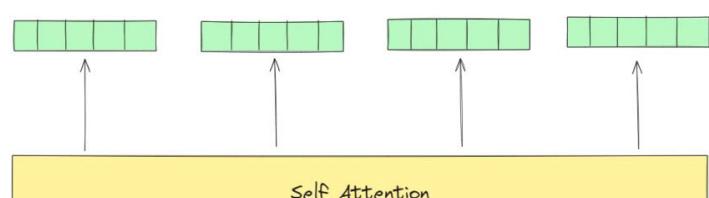
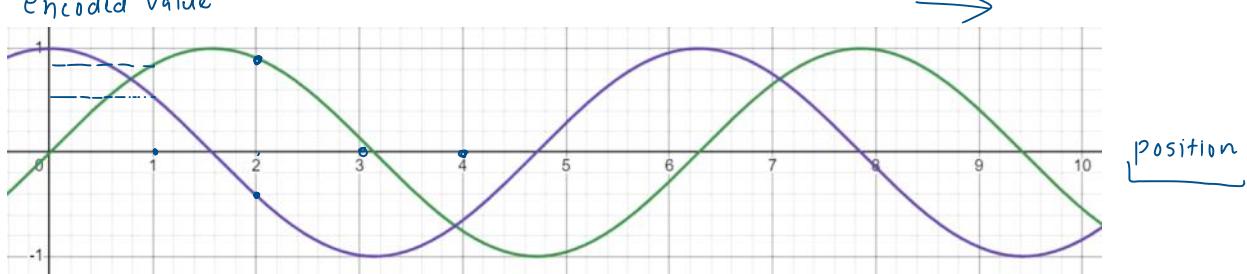


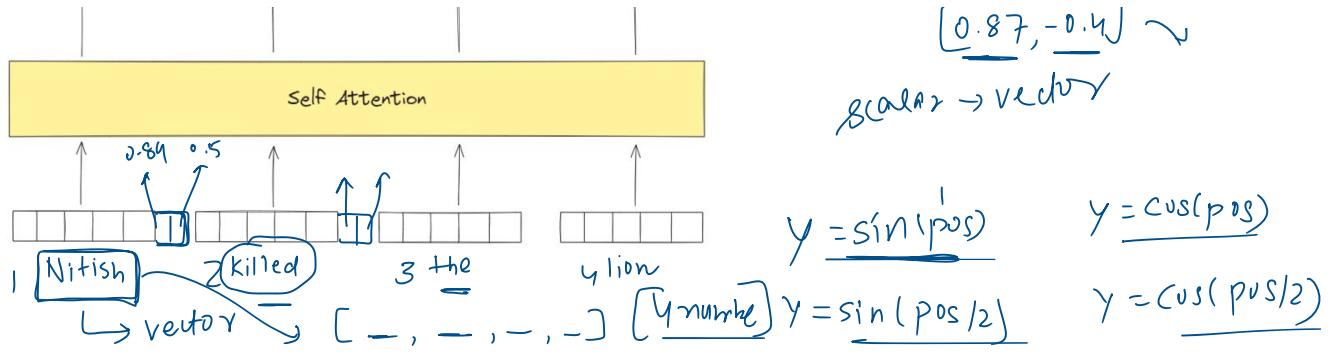
## The sine function as a solution

23 May 2024 17:17



Encoded value





$$[0.87, -0.4] \rightsquigarrow$$

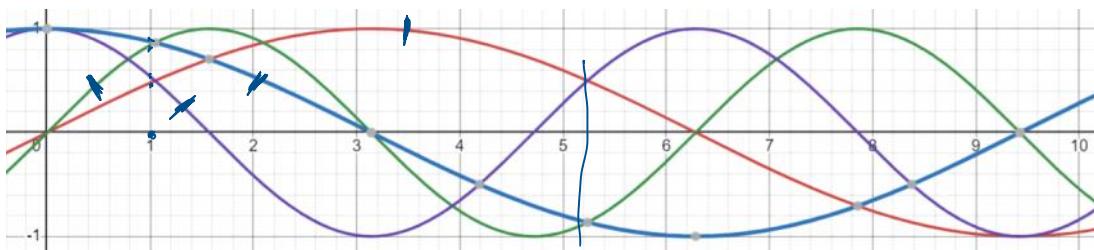
scalar → vector

$$y = \sin(\frac{1}{pos})$$

$$y = \cos(\frac{1}{pos})$$

$$y = \sin(\frac{pos}{2})$$

$$y = \cos(\frac{pos}{2})$$



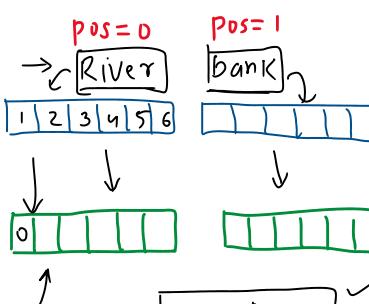
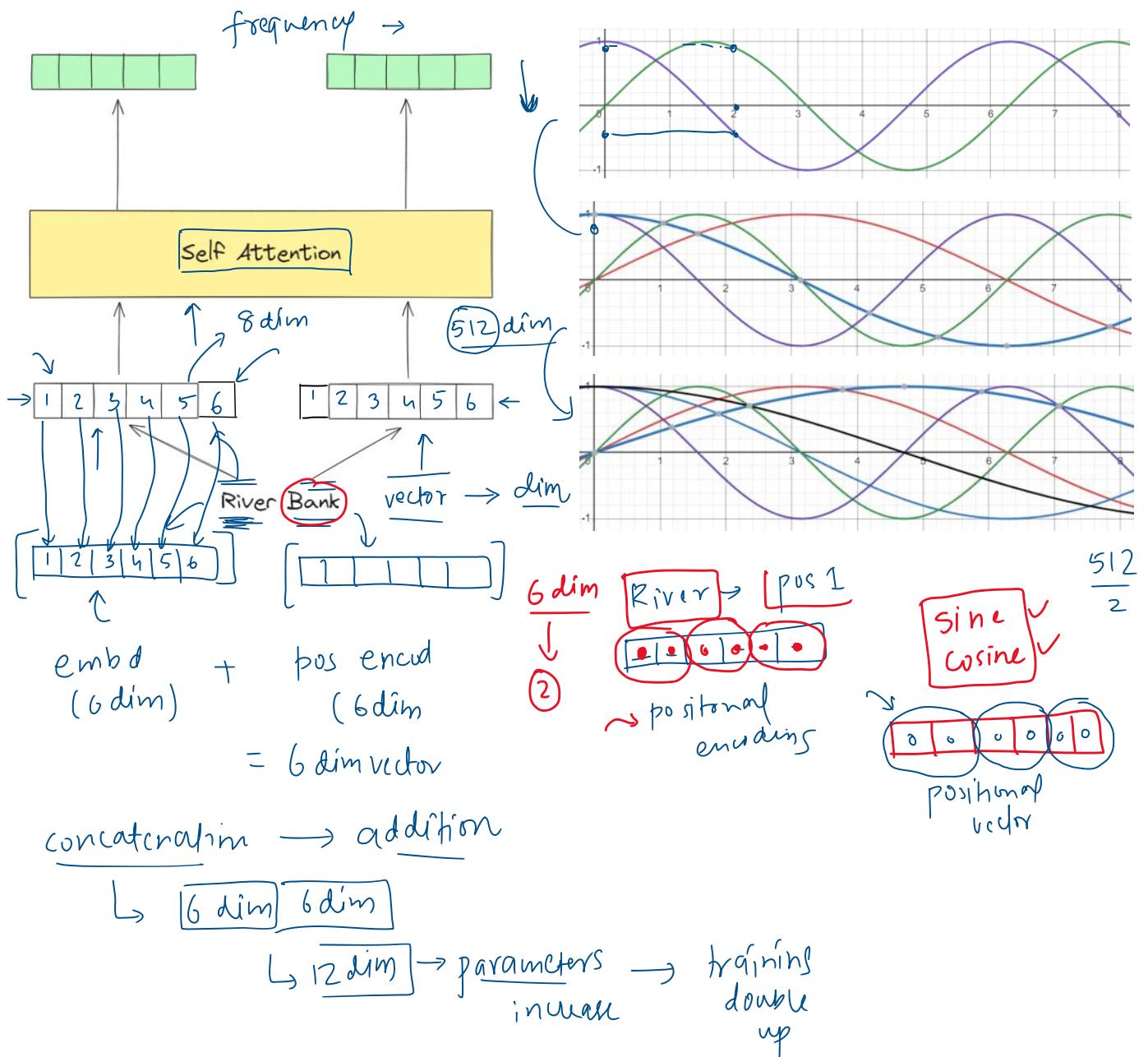
position number  
vector

$$y = \sin(pos) \quad y = \cos(pos)$$

$$y = \sin(pos/2)$$

## Positional Encoding

23 May 2024 20:35



pos → position  
pos = 0 pos = 1  
 $d_{model} \rightarrow \text{dim of embedding}$   
 $d_{model} = 6$

$$d_{model} = 6$$

for  $i = 0$  → pos = 0

$PE(0, 0) = \sin(0 / 10000^{\circ}) = 0$

$PE(0, 1) = \cos(0 / 10000^{\circ}) = 1$

2i+1

$$PE_{(pos, 2i)} = \sin(pos / 10000^{2i/d_{model}})$$

$$PE_{(pos, 2i+1)} = \cos(pos / 10000^{2i/d_{model}})$$

$i = 0 - [d_{model}/2] + 1$

$0 - [6/2] = 0 - 3$

$i = 0 - 3$

for  $i = 0$

$$PE(1, 0) = \sin(1 / 10000^0) = 0.84$$

$$PE(1, 1) = \cos(1 / 10000^0) = 0.54$$

for  $i = 1$

for  $i = 1$

for  $i=1$  ✓

$$PE(0, 2) = \sin\left(\frac{0}{10000^{1/3}}\right) = 0$$

$$PE(0, 3) = \cos\left(\frac{0}{10000^{1/3}}\right) = 1$$

for  $i=1$

$$PE(1, 2) = \sin\left(\frac{1}{10000^{1/3}}\right) = 0.04$$

$$PE(1, 3) = \cos\left(\frac{1}{100000^{1/3}}\right) = 0.99$$

for  $i=2$  ✓

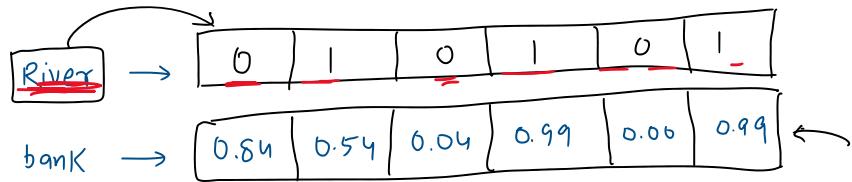
$$PE(0, 4) = \sin\left(\frac{0}{10000^{2/3}}\right) = 0$$

$$PE(0, 5) = \cos\left(\frac{0}{10000^{2/3}}\right) = 1$$

for  $i=2$

$$PE(1, 4) = \sin\left(\frac{1}{10000^{2/3}}\right) = 0.00$$

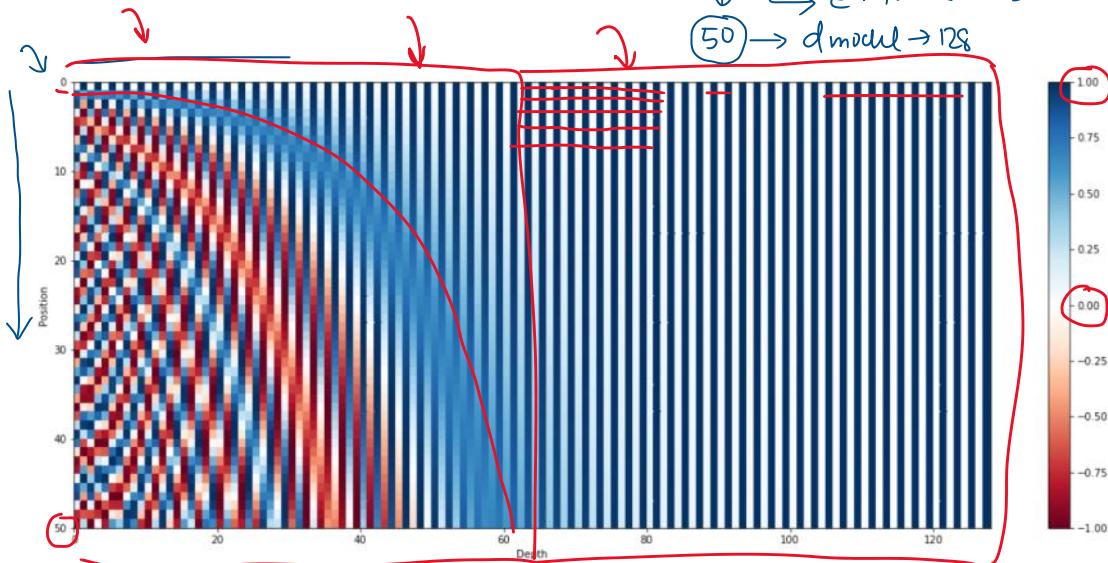
$$PE(1, 5) = \cos\left(\frac{1}{10000^{2/3}}\right) = 0.99$$



## Interesting Observations

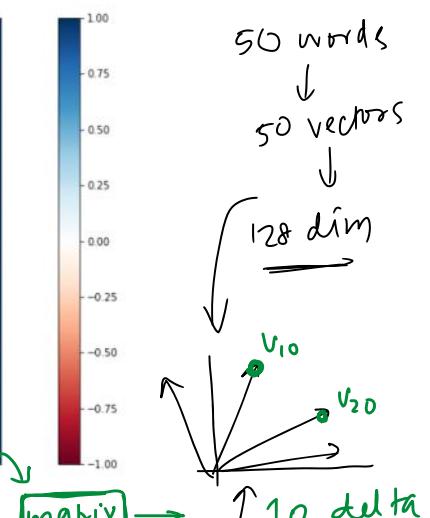
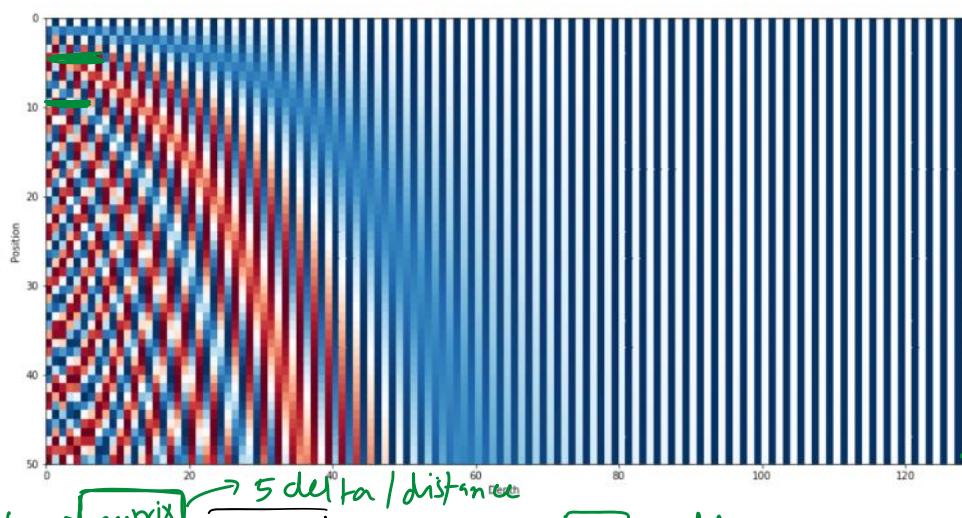
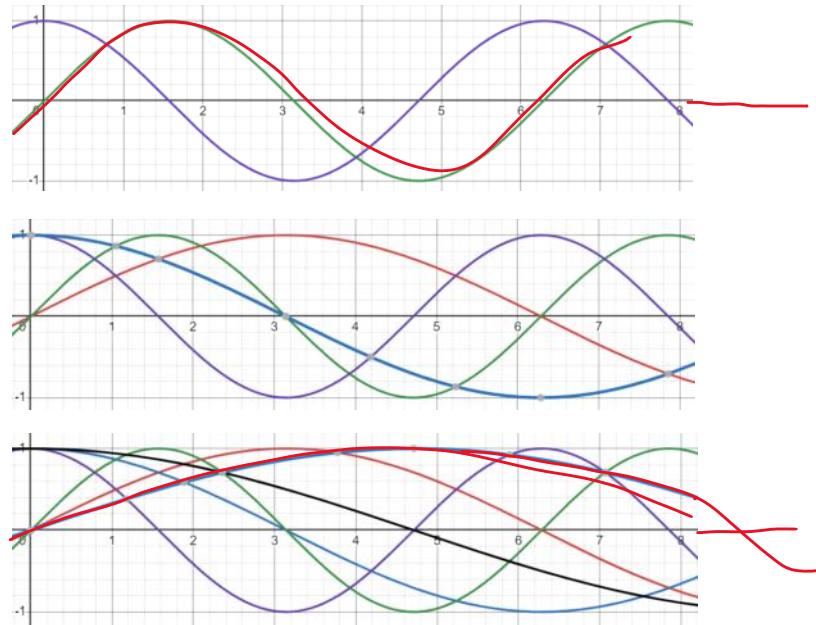
24 May 2024 02:06

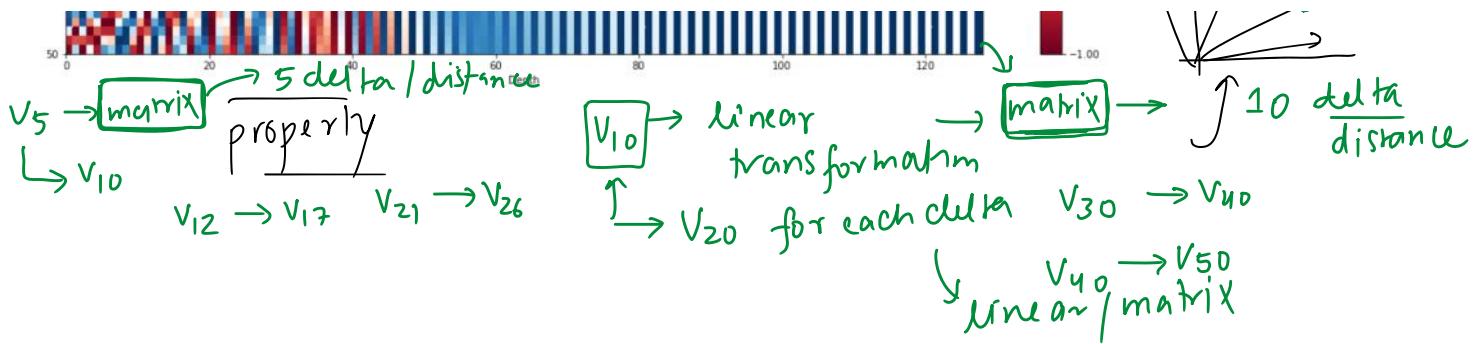
Sentence  $\rightarrow$  50 words  $\downarrow$  embedding  $\rightarrow$  128  $d_{\text{model}} = 128$



100 binary encoding 4bit 8bit

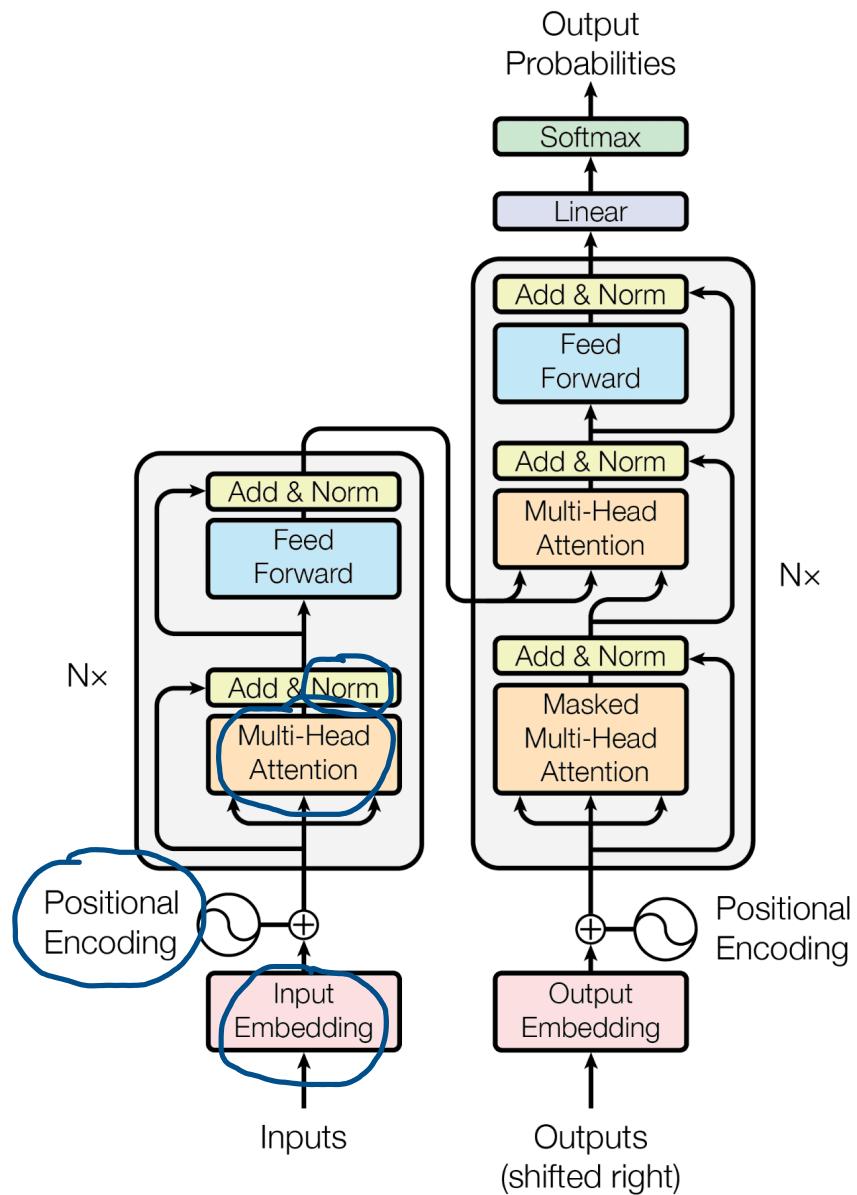
0 :	0 0 0 0	8 :	1 0 0 0
1 :	0 0 0 1	9 :	1 0 0 1
2 :	0 0 1 0	10 :	1 0 1 0
3 :	0 0 1 1	11 :	1 0 1 1
4 :	0 1 0 0	12 :	1 1 0 0
5 :	0 1 0 1	13 :	1 1 0 1
6 :	0 1 1 0	14 :	1 1 1 0
7 :	0 1 1 1	15 :	1 1 1 1





# Agenda

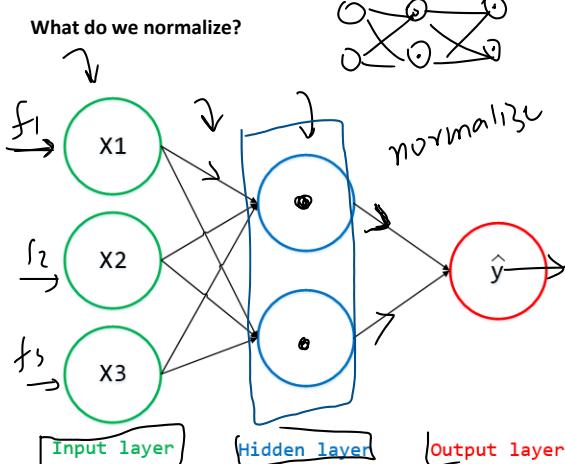
07 June 2024 02:03



## What is Normalization

05 June 2024 10:32

Normalization in deep learning refers to the process of transforming data or model outputs to have specific statistical properties, typically a mean of zero and a variance of one.



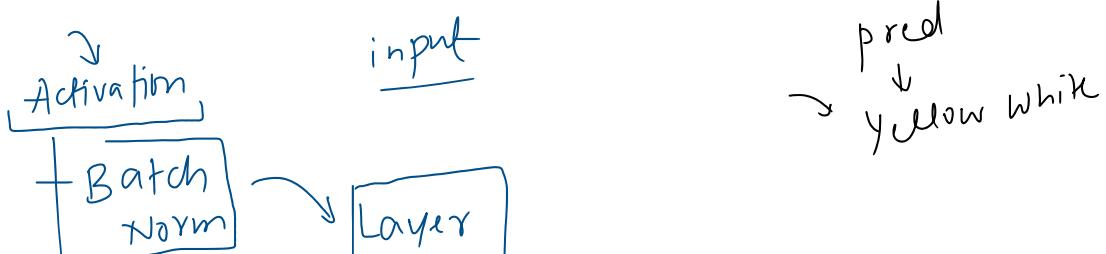
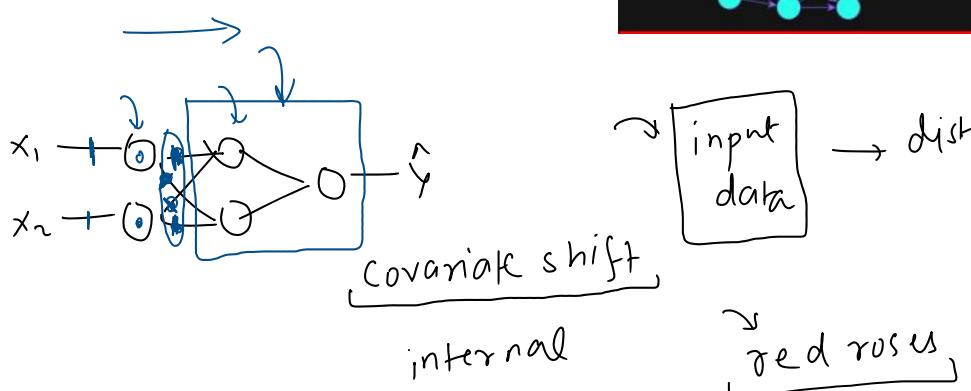
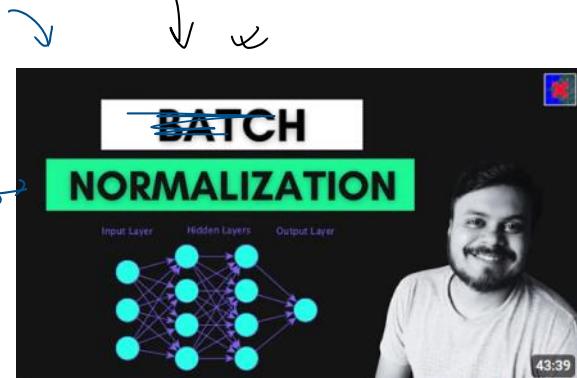
$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{array} \quad \text{min-max}$$

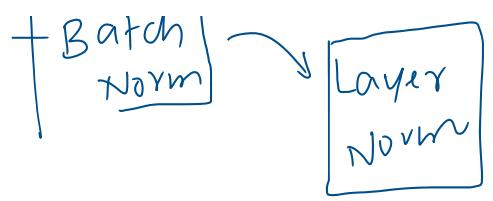
$$\frac{x_i - M}{\sigma} \quad \mu = 0 \quad \sigma = 1$$

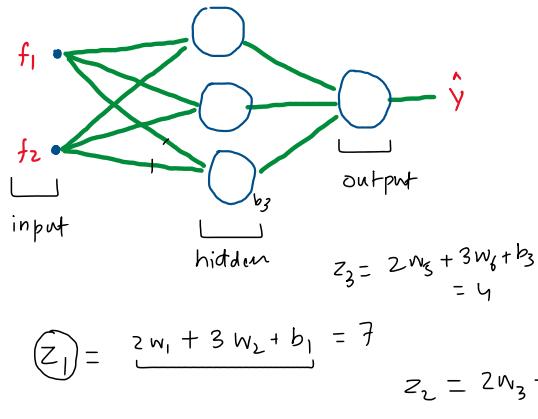
$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{array}$$

### Benefits of Normalization in Deep Learning

- Improved Training Stability:
  - Normalization helps to stabilize and accelerate the training process by reducing the likelihood of extreme values that can cause gradients to explode or vanish.
- Faster Convergence:
  - By normalizing inputs or activations, models can converge more quickly because the gradients have more consistent magnitudes. This allows for more stable updates during backpropagation.
- Mitigating Internal Covariate Shift:
  - Internal covariate shift refers to the change in the distribution of layer inputs during training. Normalization techniques, like batch normalization, help to reduce this shift, making the training process more robust.
- Regularization Effect:
  - Some normalization techniques, like batch normalization, introduce a slight regularizing effect by adding noise to the mini-batches during training. This can help to reduce overfitting.







$f_1$	$f_2$	$z_1$	$z_2$	$z_3$
2	3	7	5	4
1	1	2	3	4
5	4	1	2	3
6	1	7	5	6
7	1	3	3	4

$$\frac{7 - \mu_1}{\sigma_1} = \frac{0.36}{(1)} \gamma_1 + \beta_1 = 0.36$$

$$\frac{2 - \mu_1}{\sigma_1} = 0.71 \gamma_1 + \beta_1 = 0.71$$

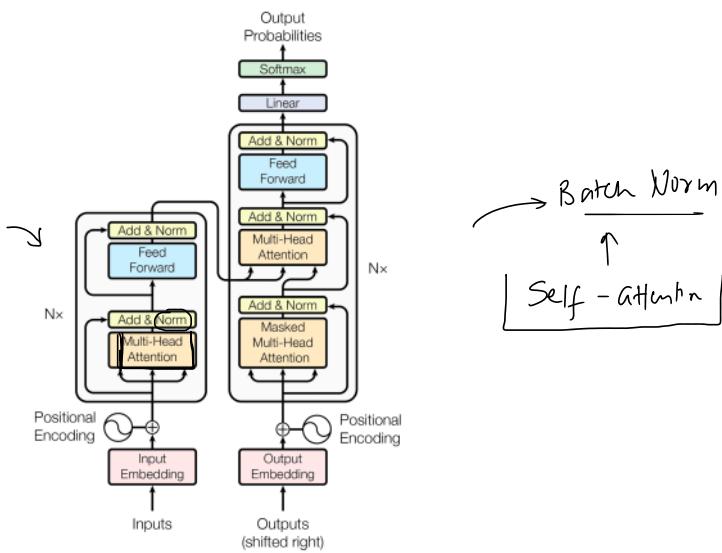
$$\frac{5 - \mu_2}{\sigma_2} = -0.21 \gamma_2 + \beta_2 = -0.21$$

$$\frac{4 - \mu_3}{\sigma_3} = 0.12 \gamma_3 + \beta_3 = 0.12$$

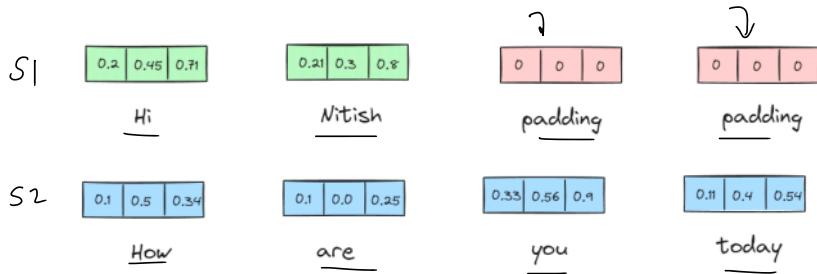
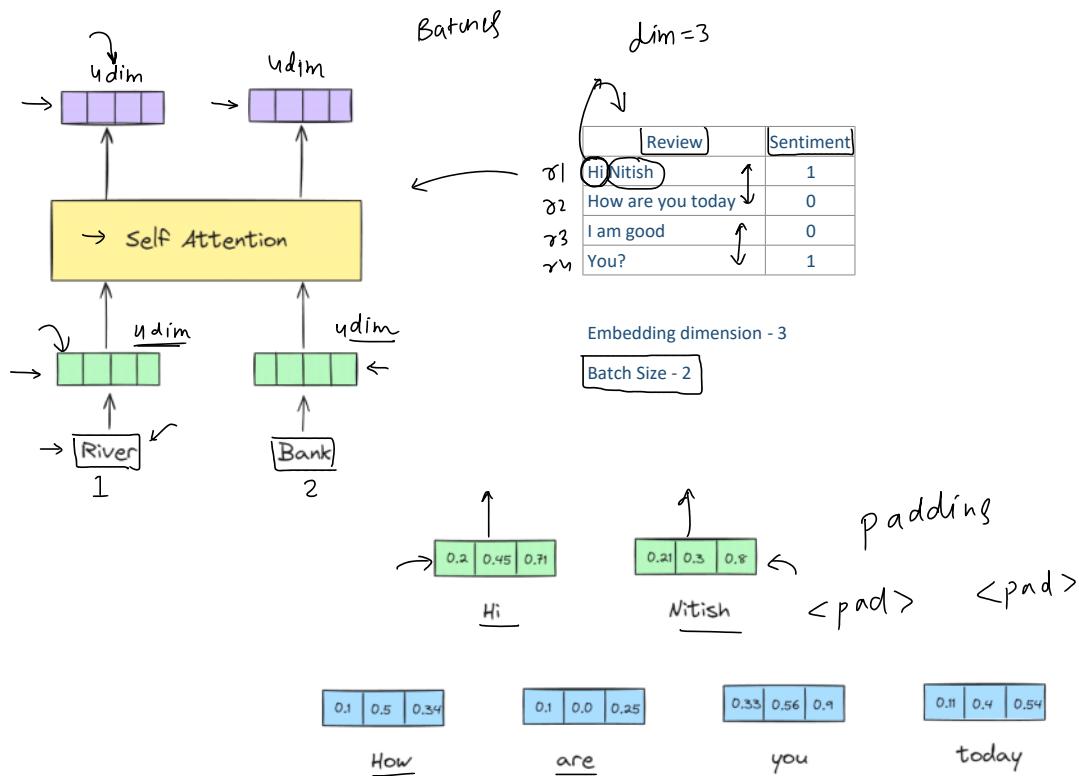
Why don't we use Batch Norm in Transformers?

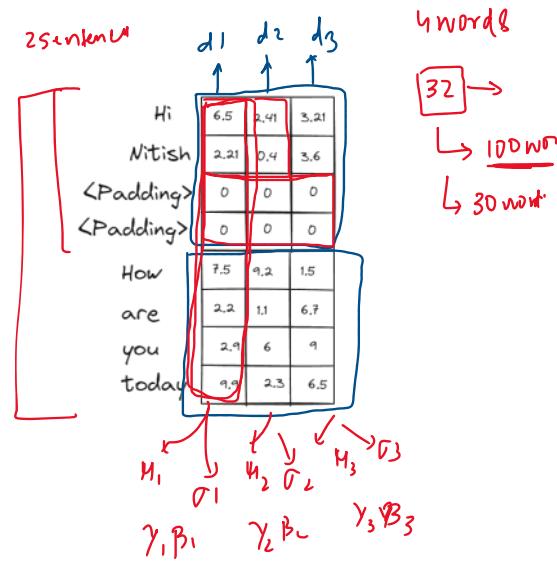
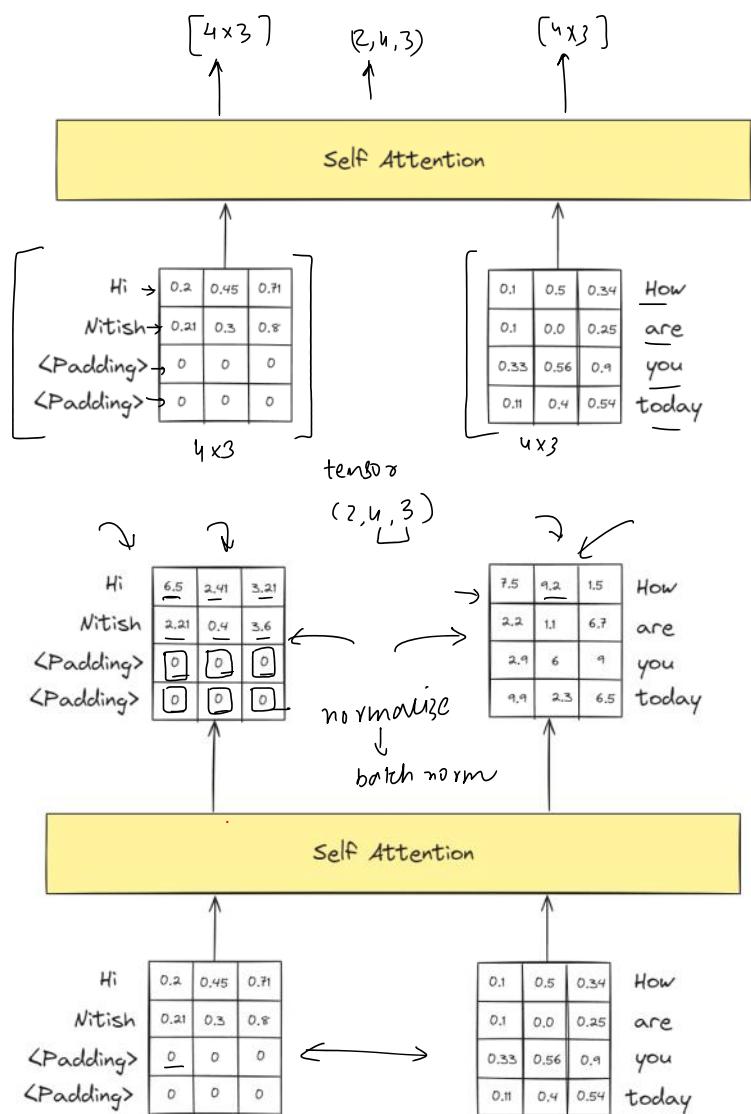
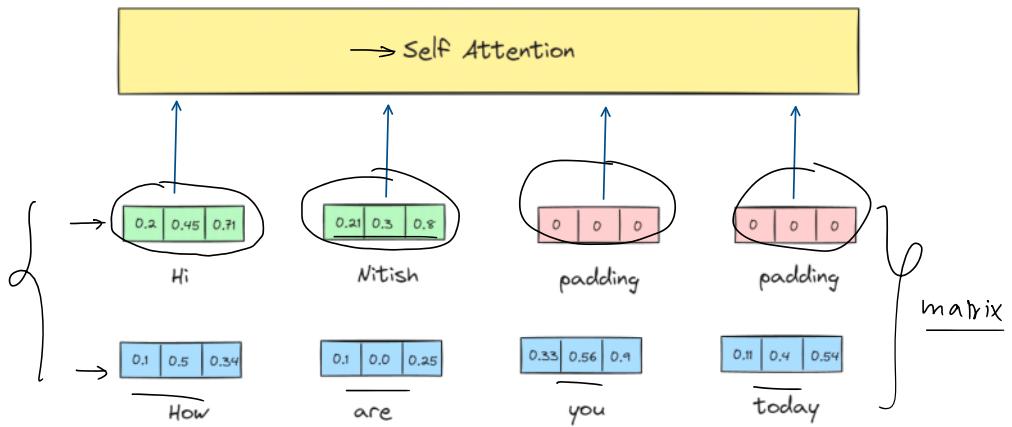
05 June 2024 10:40

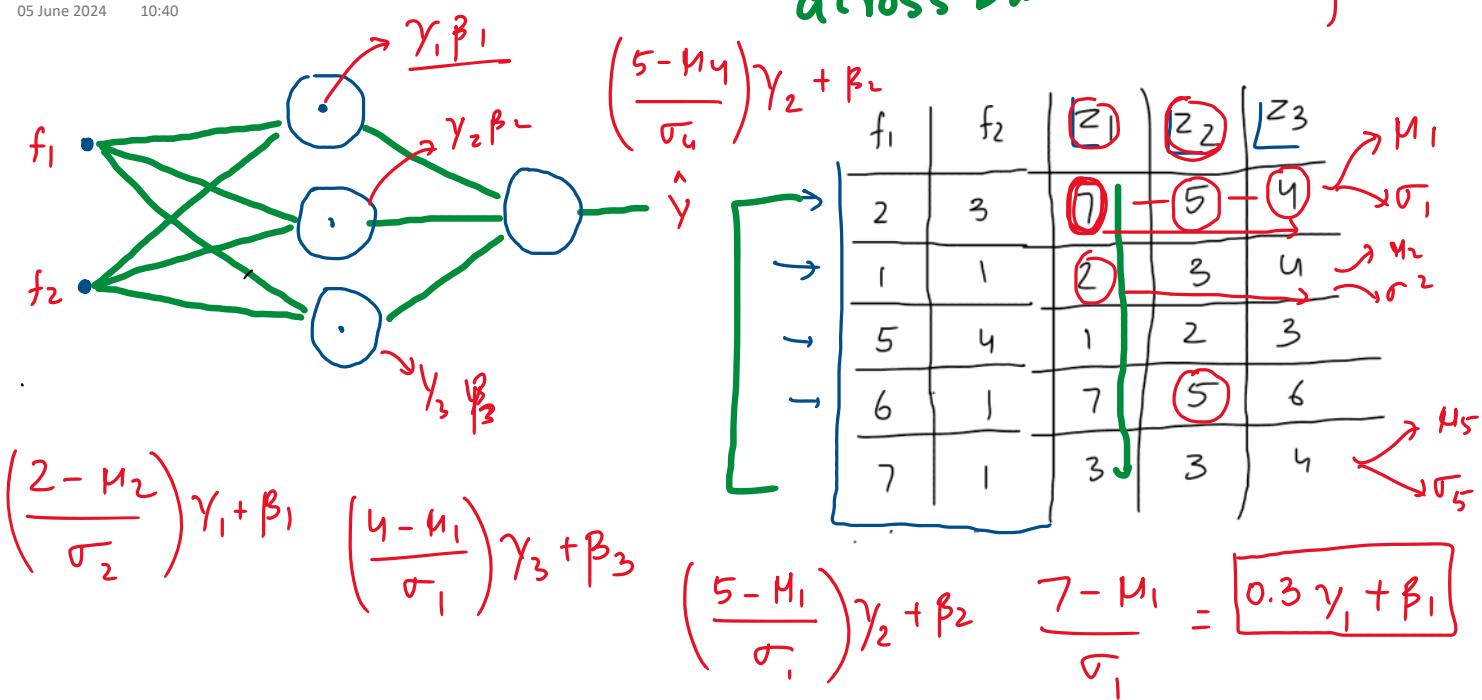
↳ self attention



Batch Norm  
↑  
Self - attention

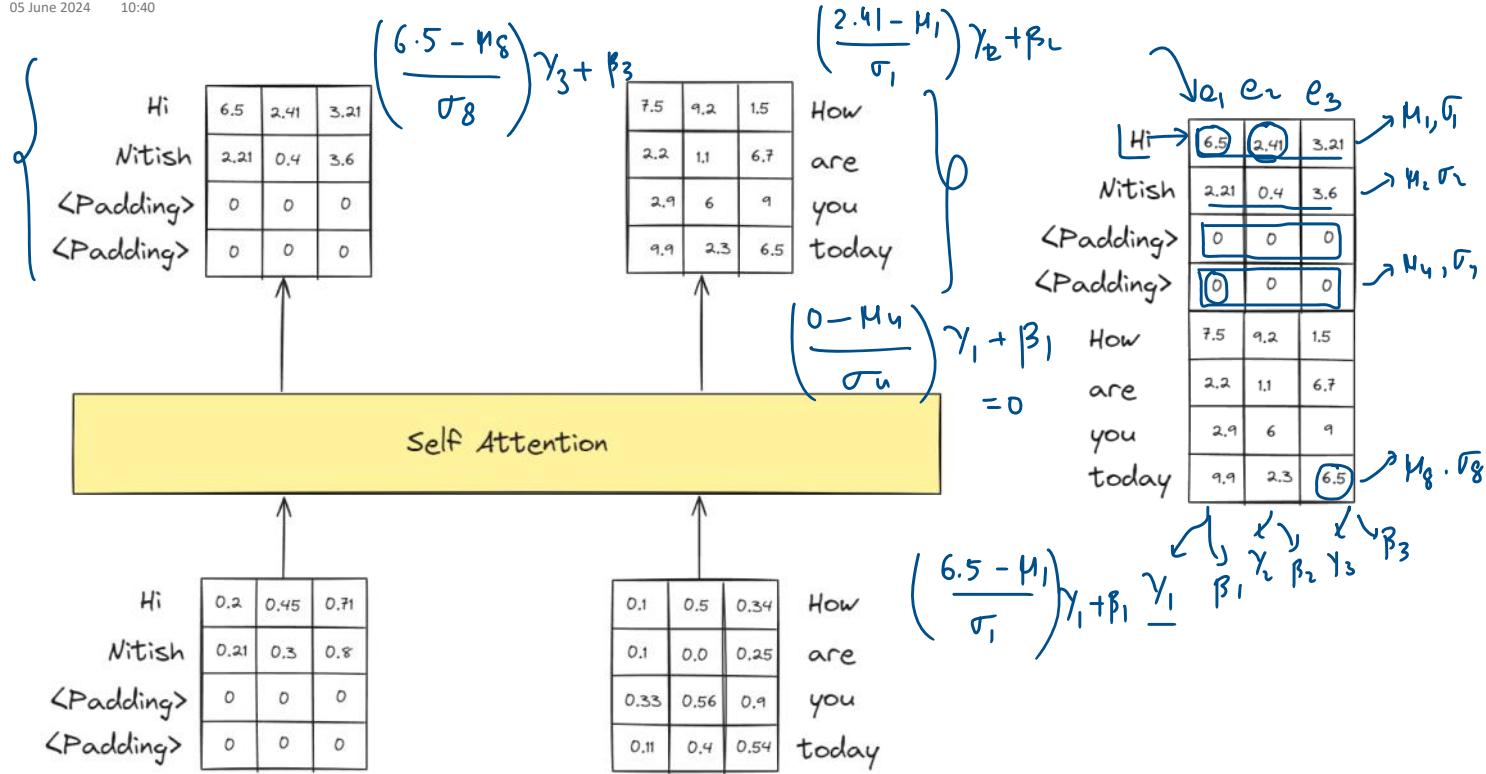






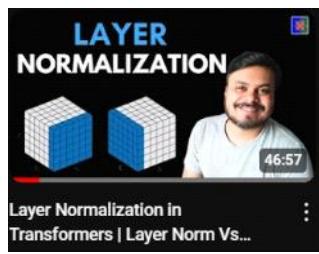
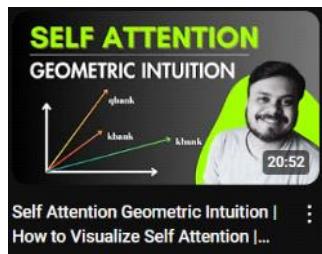
## Layer Norm in Transformers

05 June 2024 10:40

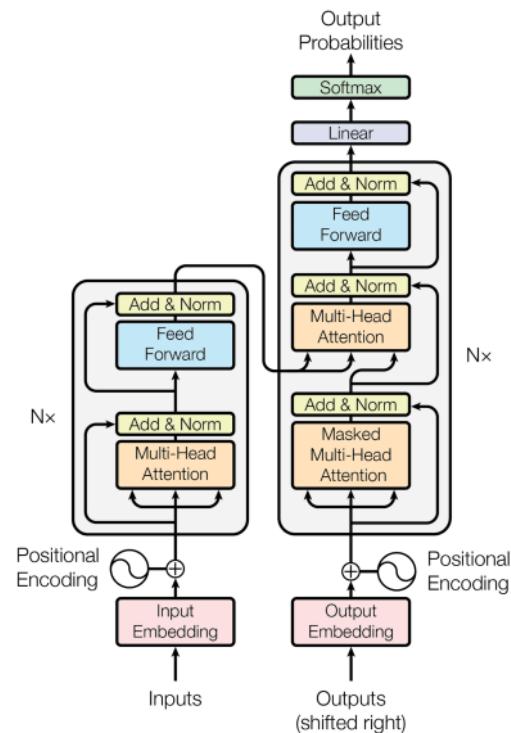


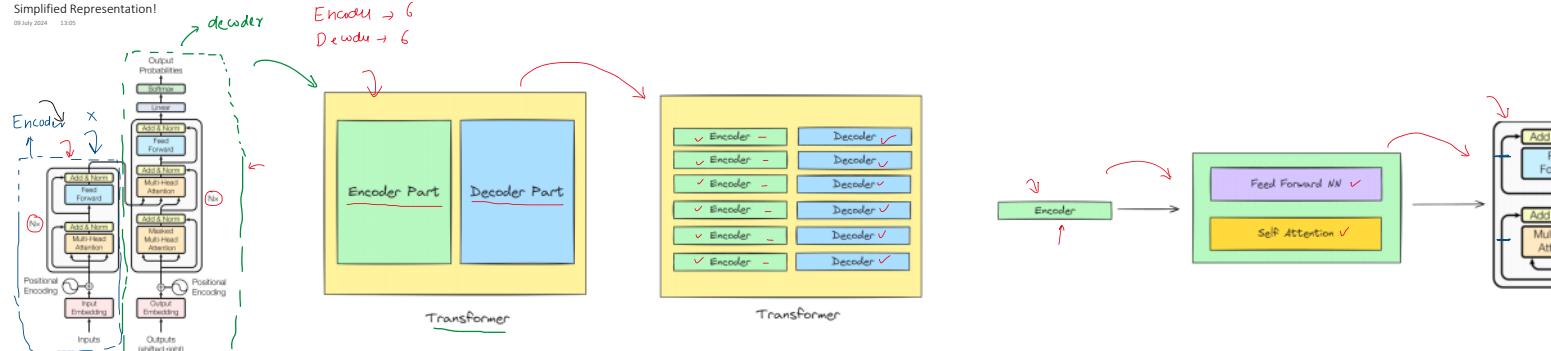
## Recap

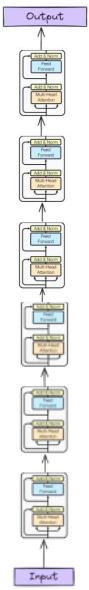
09 July 2024 08:47



7 hrs

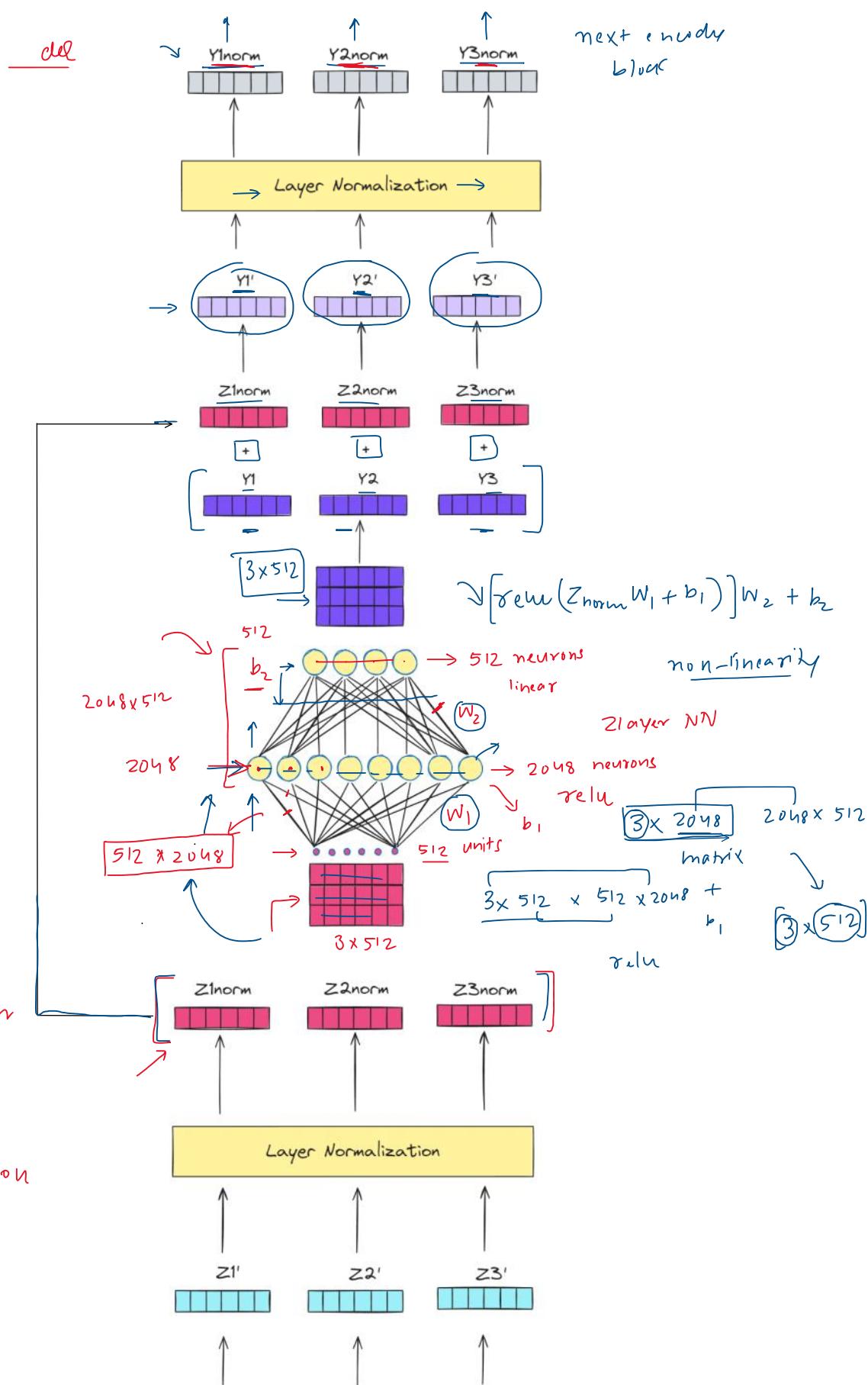
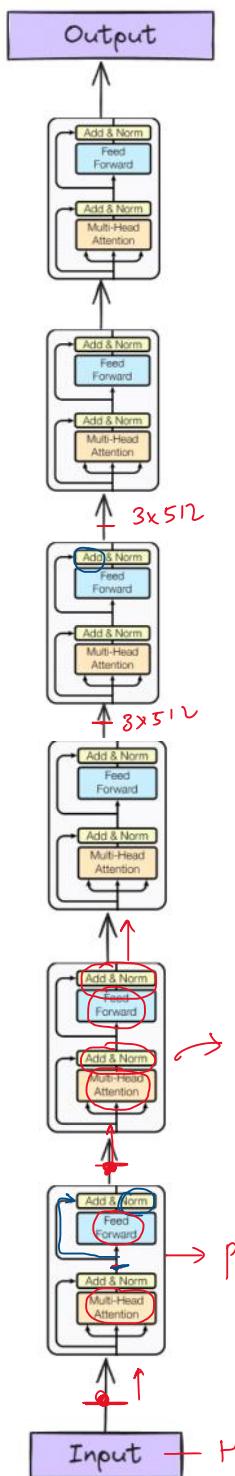


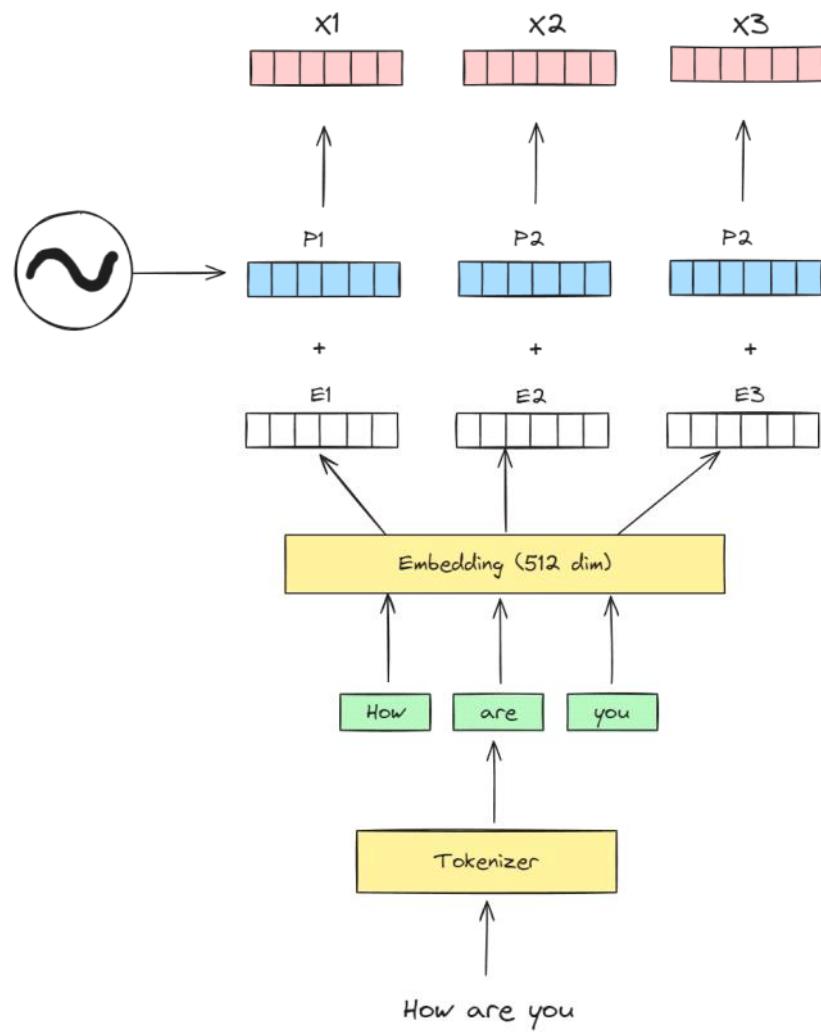
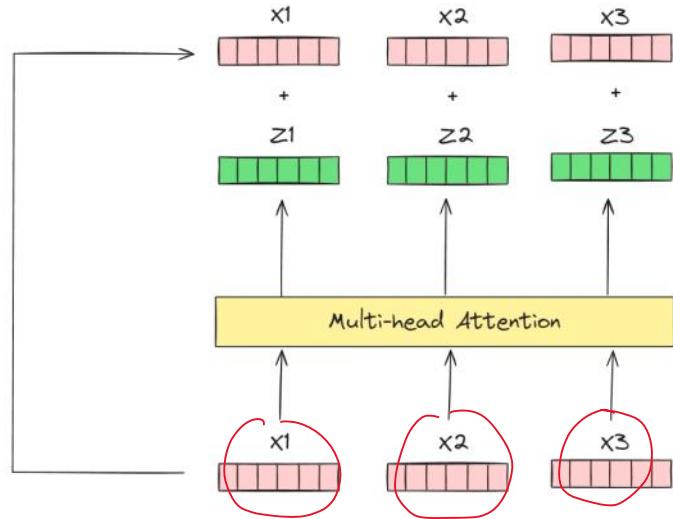




## Encoder Architecture

09 July 2024 16:31





## Some questions

09 July 2024 20:48

- { 1. Why use residual connections? → Kaggle → comment
- 2. Why use a FFNN?
- 3. Why use 6 encoder blocks?

language  
↓  
complex

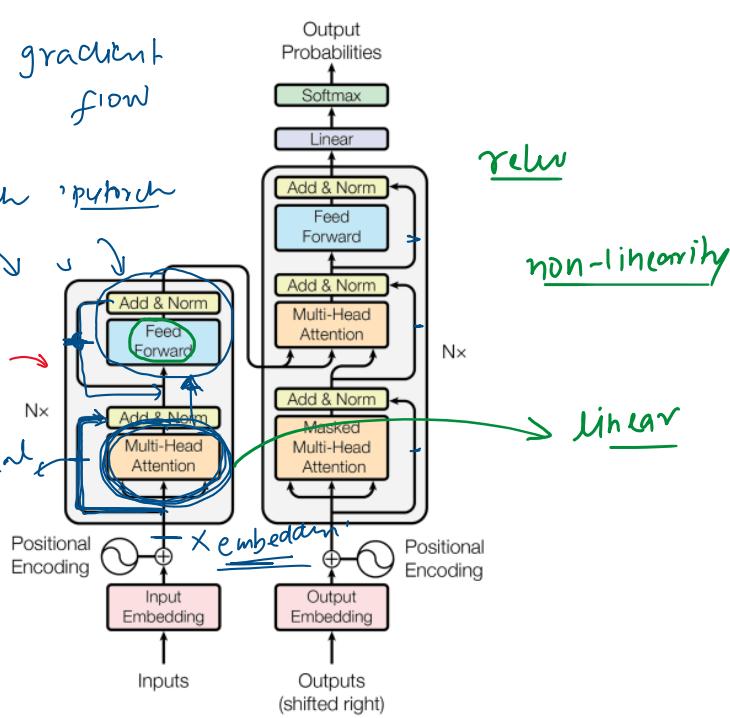
▷ Stable training → deep NN  
ResNet  
vanishing grad

2)

alternate

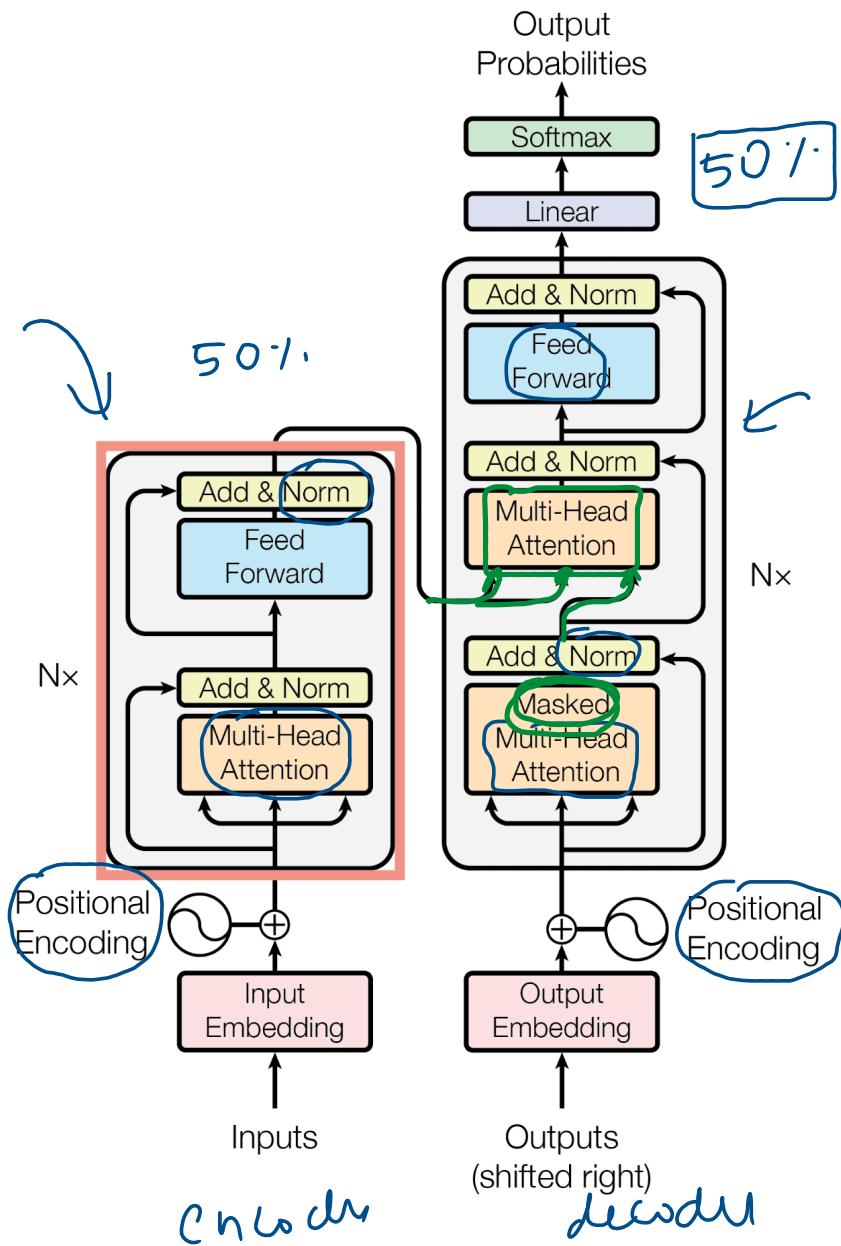
gradient  
flow

scratch → pytorch



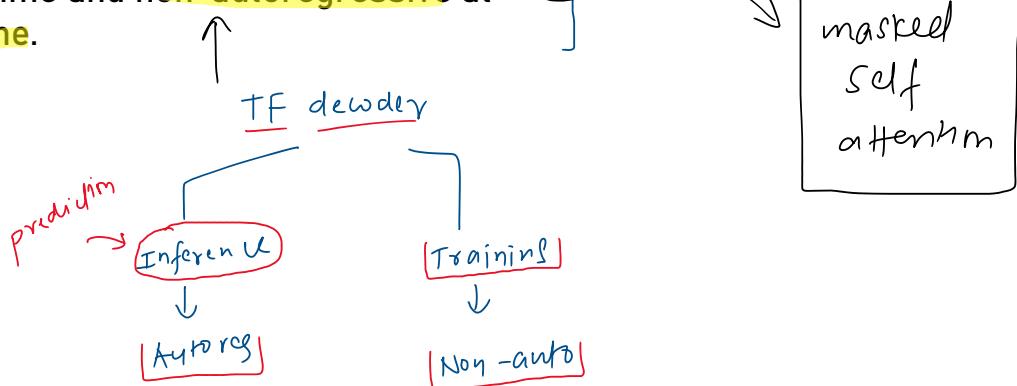
# Recap

23 July 2024 17:09

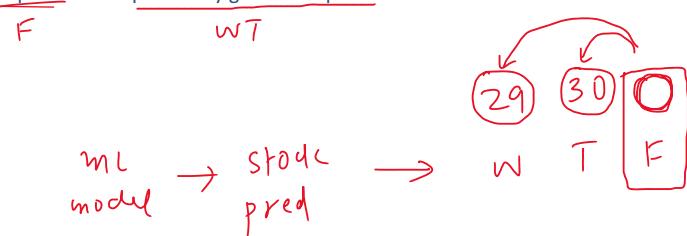


economics → Time series

The Transformer decoder is **autoregressive at inference time** and **non-autoregressive at training time**.



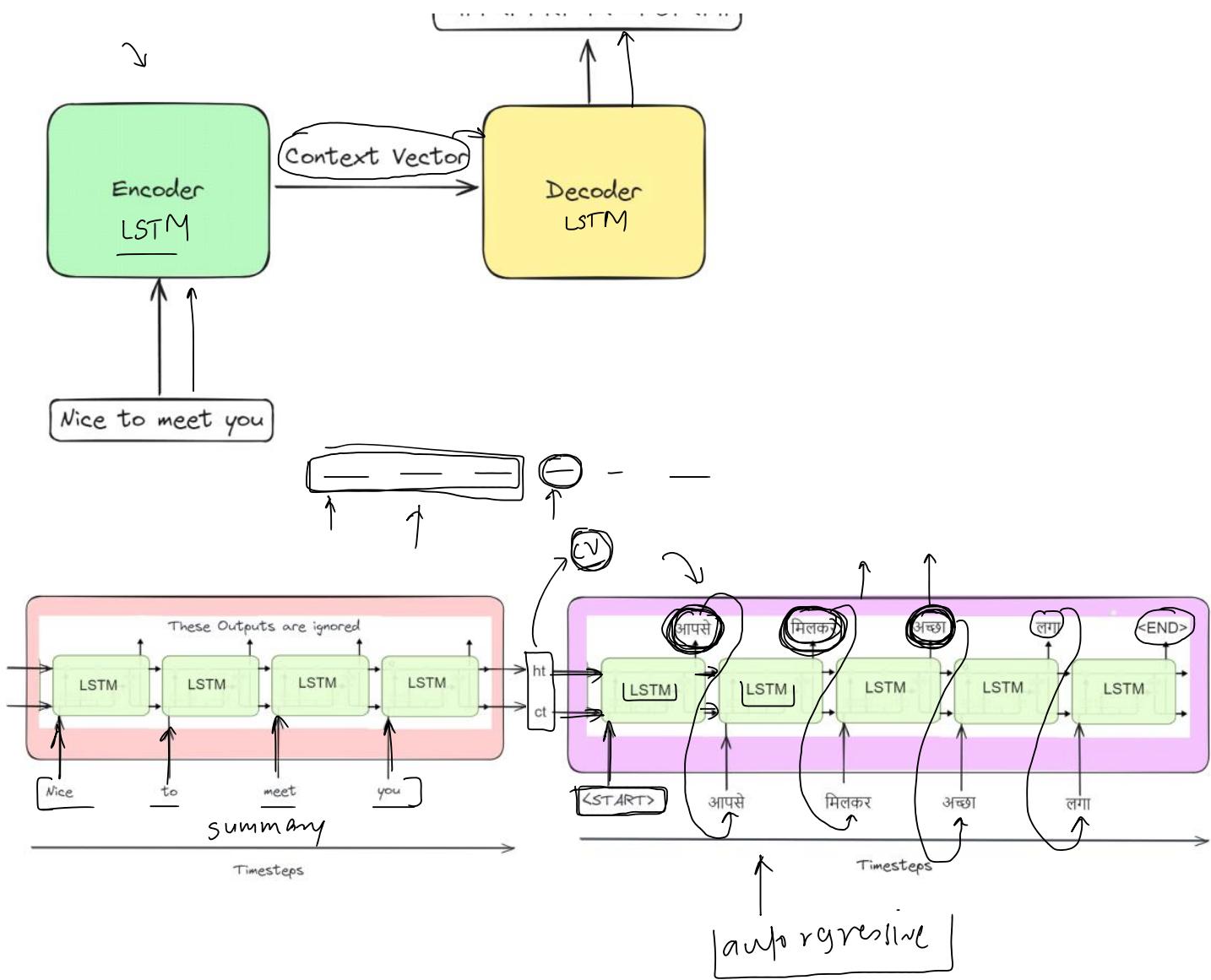
→ In the context of deep learning, autoregressive models are a class of models that generate data points in a sequence by conditioning each new point on the previously generated points.



→ Encoder - Decoder → 11ya

आप से मिल कर अच्छा लगा





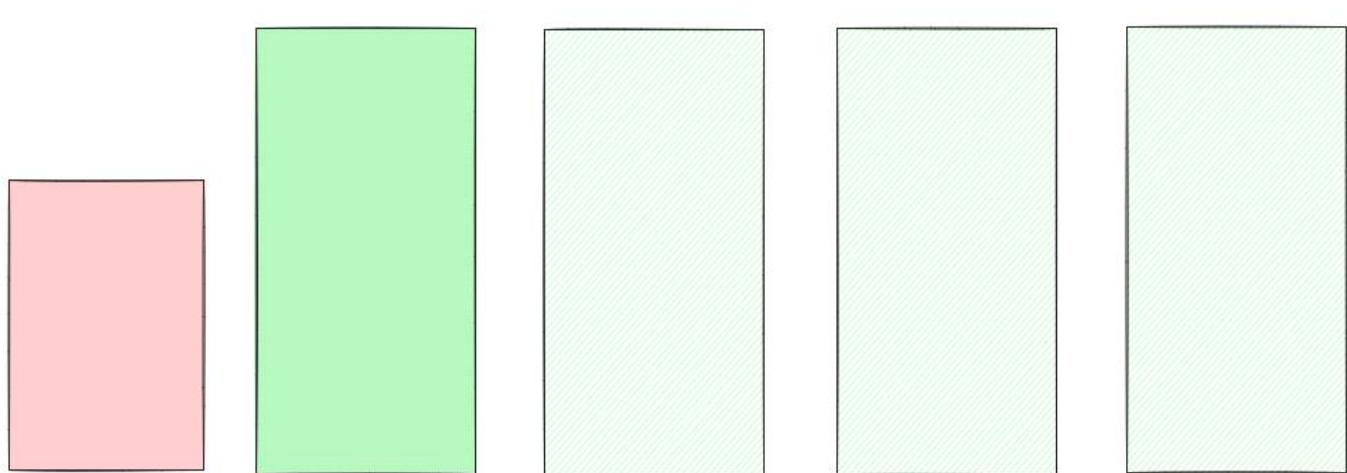
# Transformer as an Autoregressive Model

23 July 2024 23:16

The Transformer decoder is autoregressive at inference time and non-autoregressive at training time.

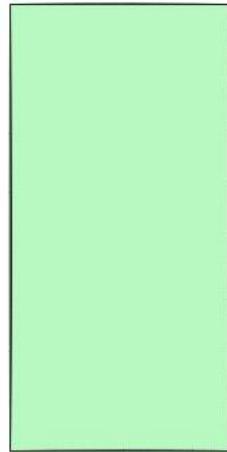
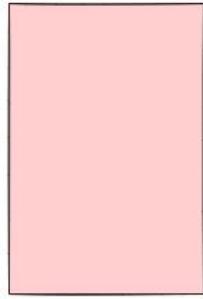
Inference

Query Sentence -> I am fine



## Training

S.No	English Sentence	Hindi Sentence
1	How are you?	आप कैसे हैं
2	Congratulations	बधाई हो
3	Thank you	धन्यवाद

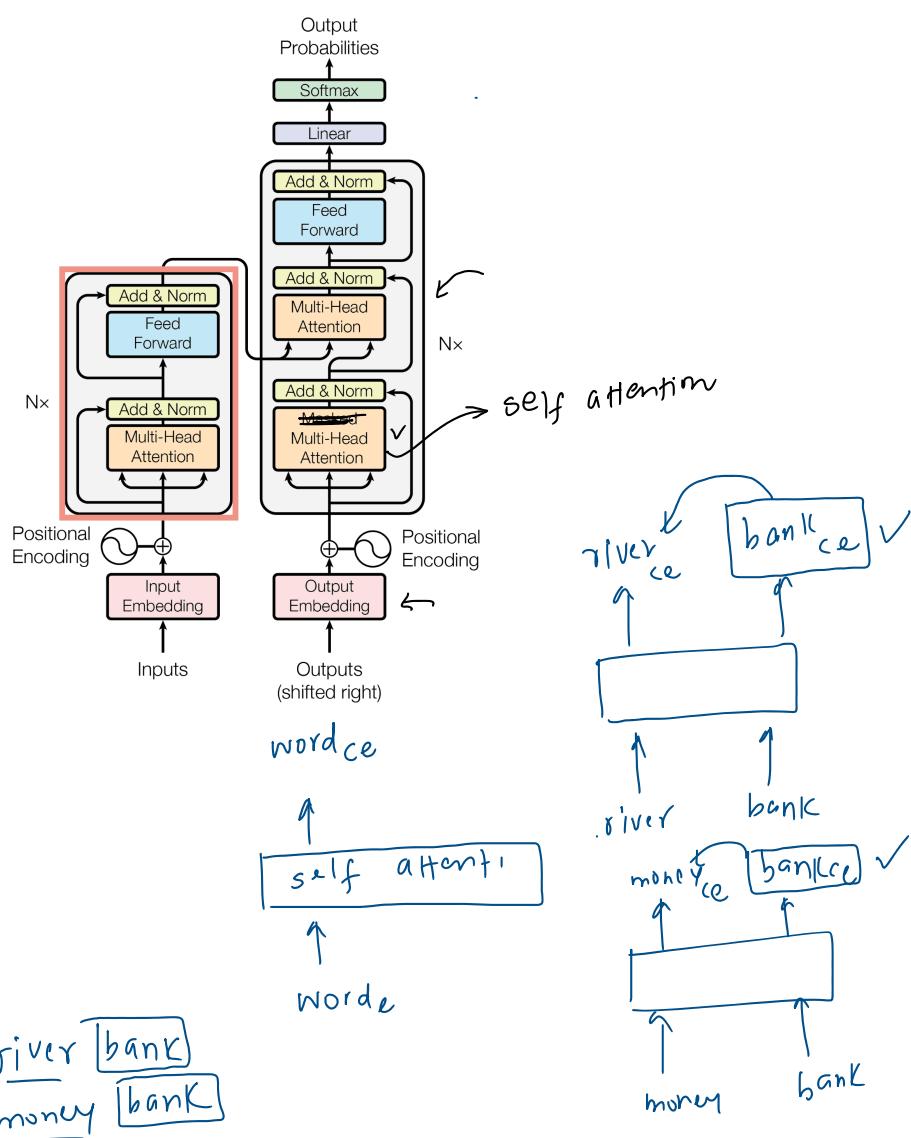
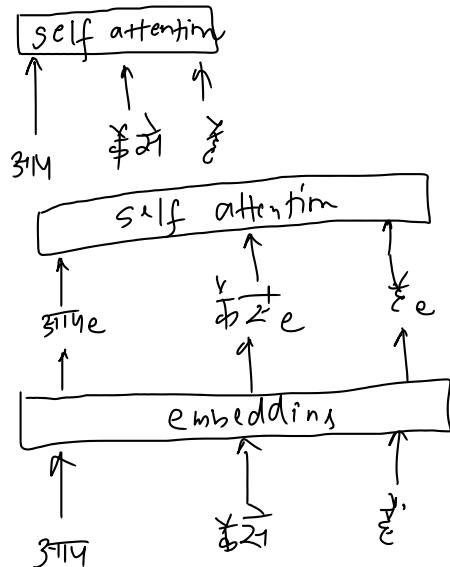


## The problem in parallelizing

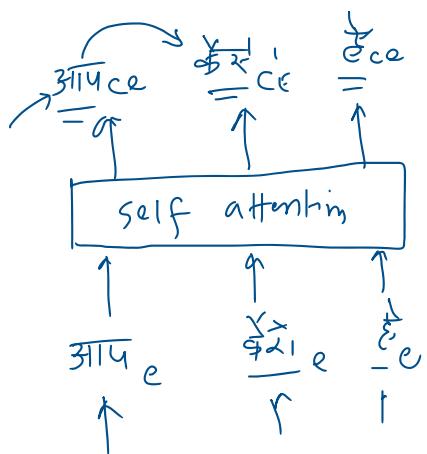
25 July 2024 22:55

S.No	English Sentence	Hindi Sentence
1	How are you?	आप कैसे हैं
2	Congratulations	बधाई हो
3	Thank you	धन्यवाद

→



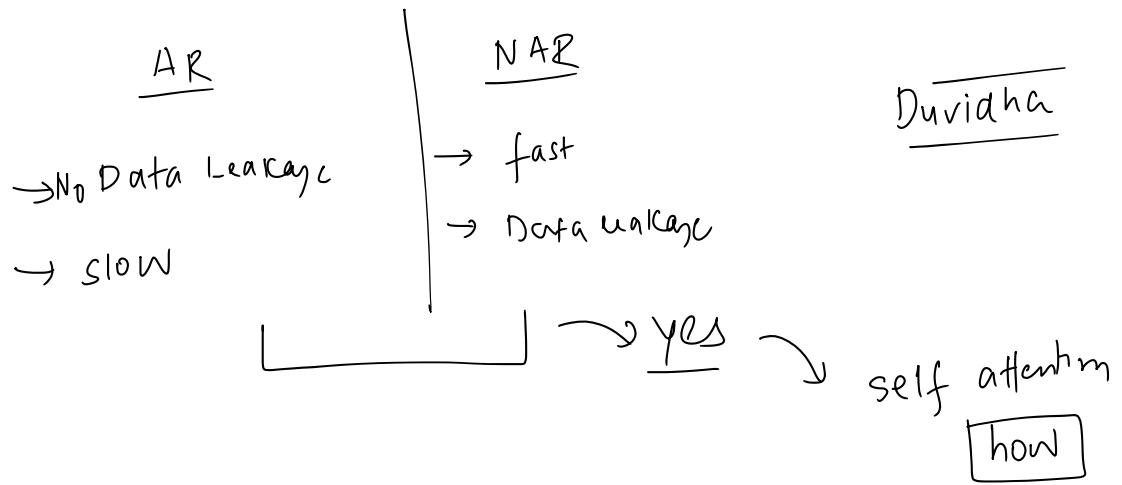
Data Leakage



Cheating

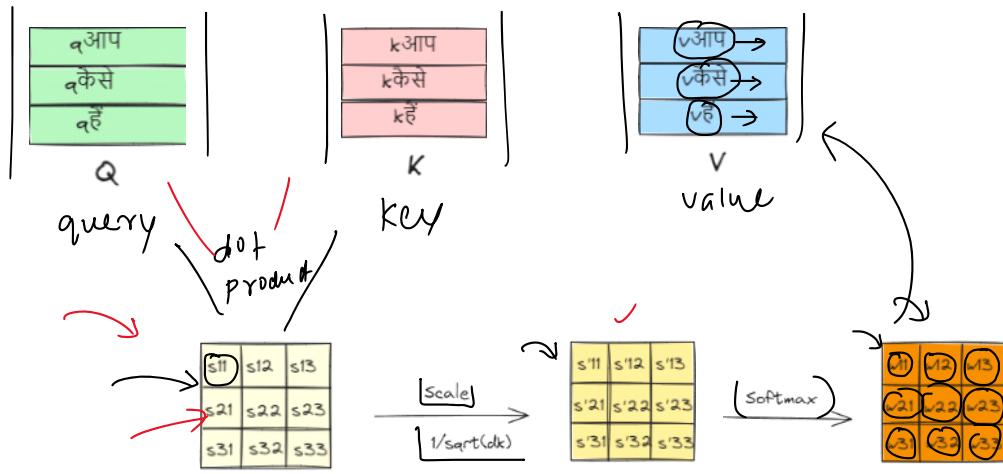
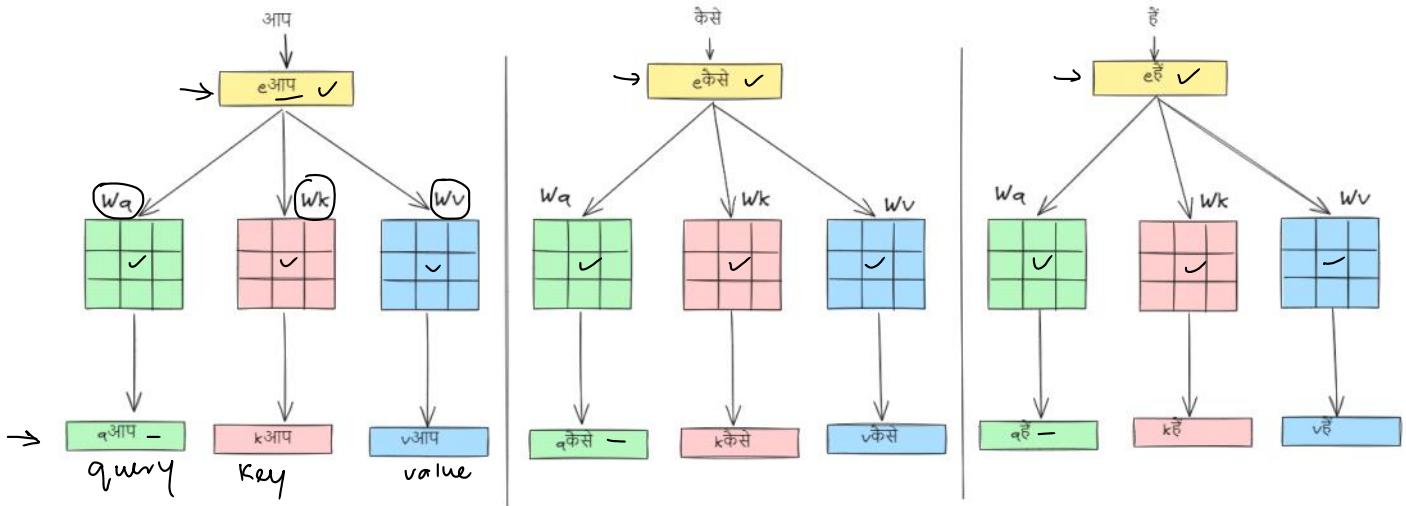
$$\begin{aligned}
 314ce &= 0.8 \underline{314e} + 0.1 \underline{कैसे} + 0.1 \underline{हैं} \\
 \underline{\underline{कैसे}} &= 0.15 \underline{314e} + 0.75 \underline{\underline{कैसे}} + 0.1 \underline{हैं} \\
 \underline{\underline{हैं}} &= 0.1 \underline{314e} + 0.2 \underline{\underline{कैसे}} + 0.7 \underline{\underline{हैं}}
 \end{aligned}$$

current token value  
↓  
future token value



## Finding the answer

26 July 2024 00:21



Equations for calculating weights:

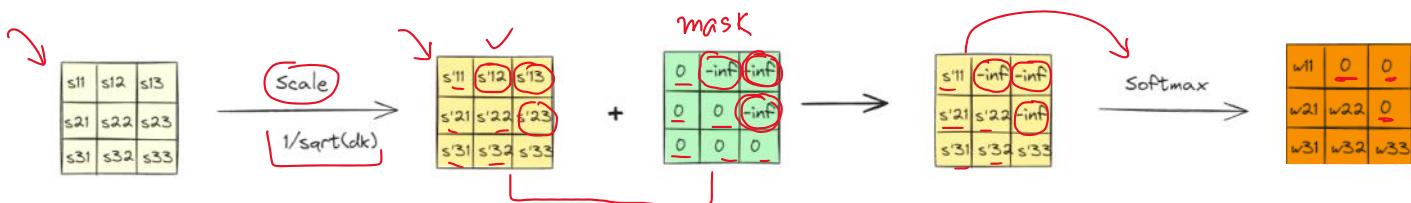
$$\begin{aligned} w_{12} &= w_{11} * s_{12} \\ w_{13} &= w_{11} * s_{13} \\ w_{23} &= w_{21} * s_{23} \end{aligned}$$

$$\begin{aligned} w_{12} &= w_{11} * s_{12} \\ w_{13} &= w_{11} * s_{13} \\ w_{23} &= w_{21} * s_{23} \end{aligned}$$

$$\begin{aligned} w_{12} &= w_{11} * s_{12} \\ w_{13} &= w_{11} * s_{13} \\ w_{23} &= w_{21} * s_{23} \end{aligned}$$

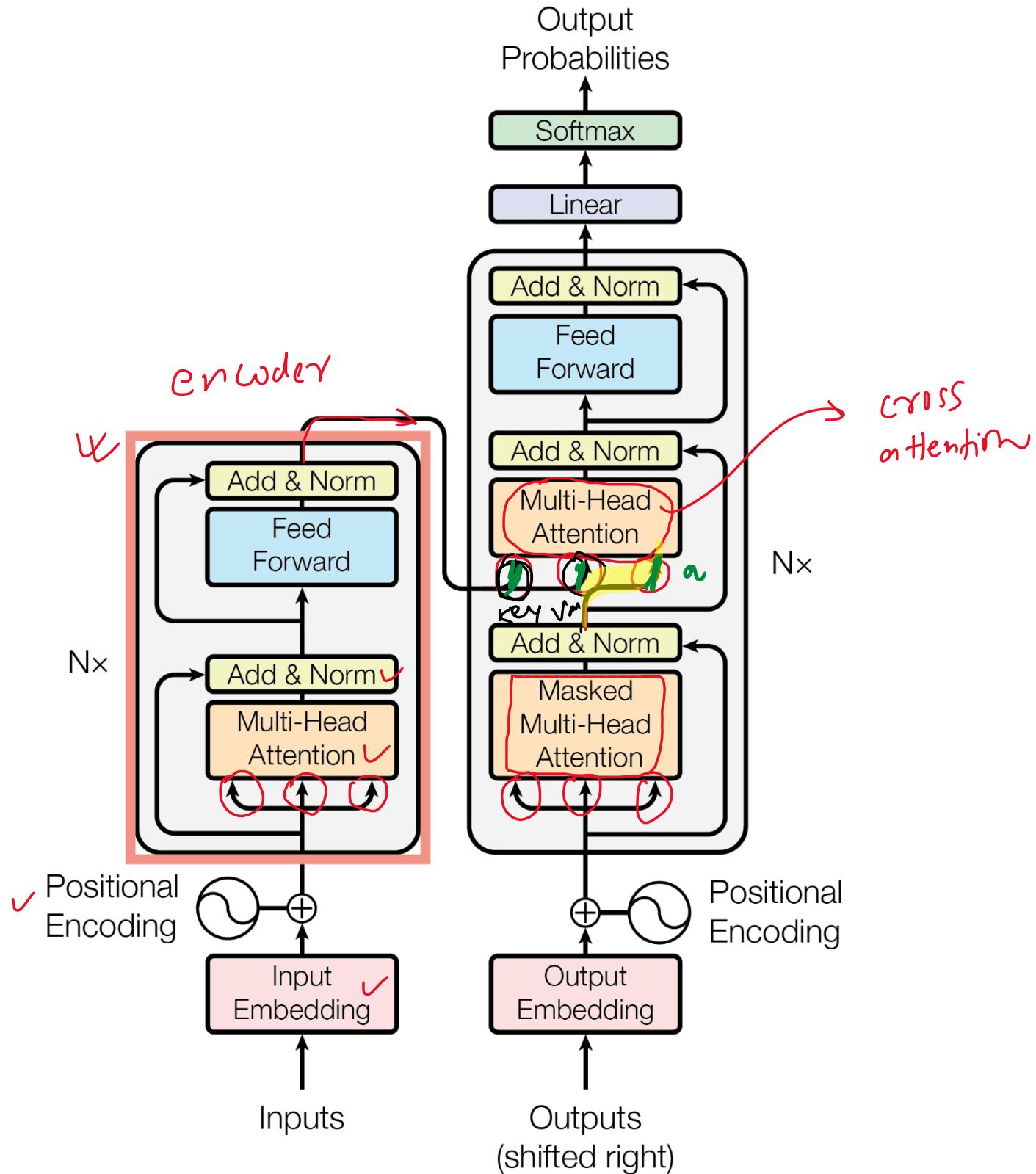
$$w_{12} = w_{13} = w_{23} = 0$$

$$\text{softmax}(-\infty) = 0$$



# Plan of Action

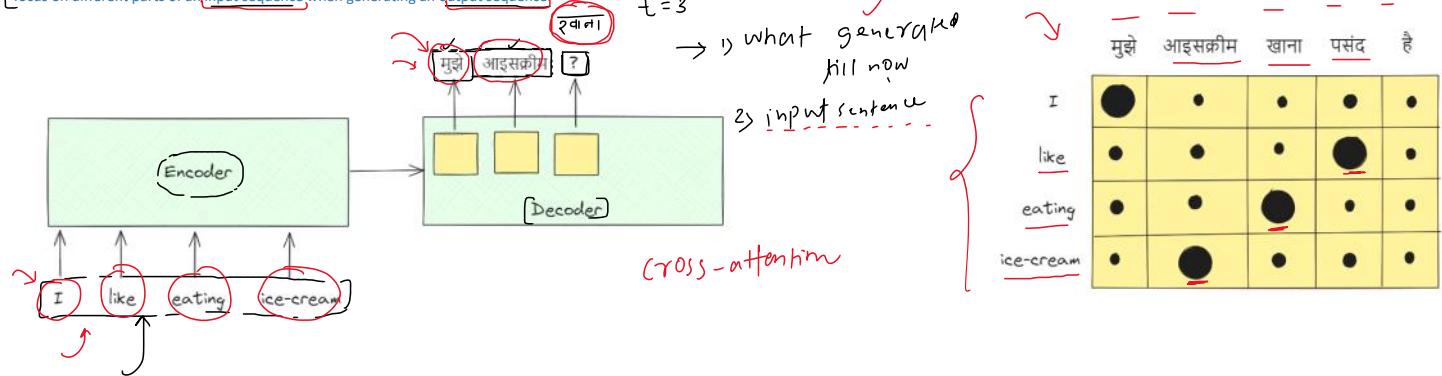
12 August 2024 17:53



## What is Cross Attention

12 August 2024 17:53

Cross-attention is a mechanism used in transformer architectures, particularly in tasks involving sequence-to-sequence data like translation or summarization. It allows a model to focus on different parts of an input sequence when generating an output sequence.



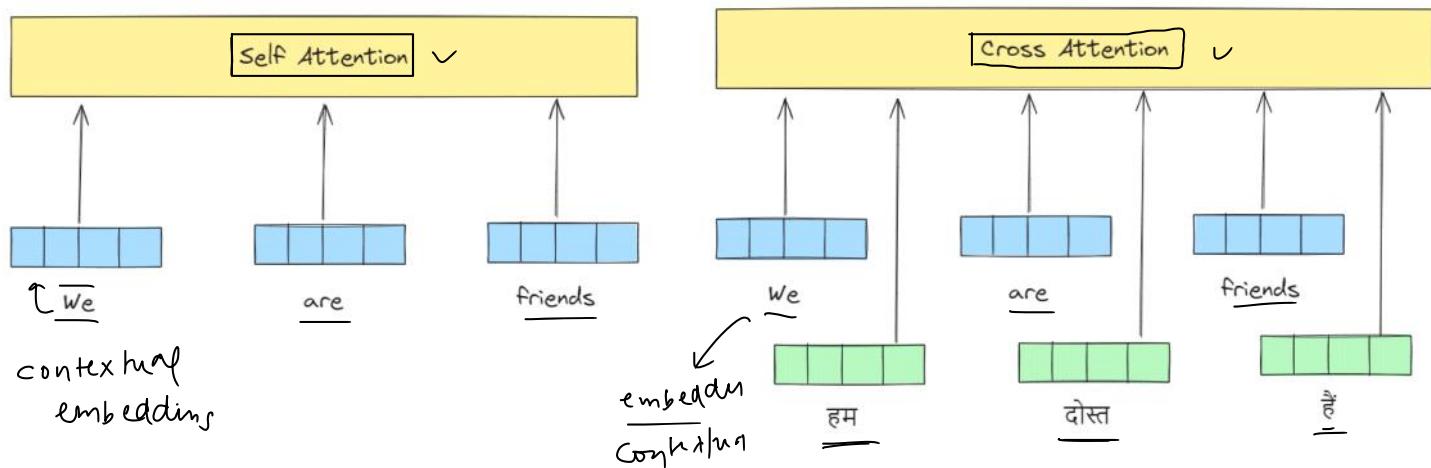
Cross Attention is conceptually very similar to Self-Attention

Self-Attention Vs Cross Attention

1. The input ✓
2. The processing ✓
3. The output ✓

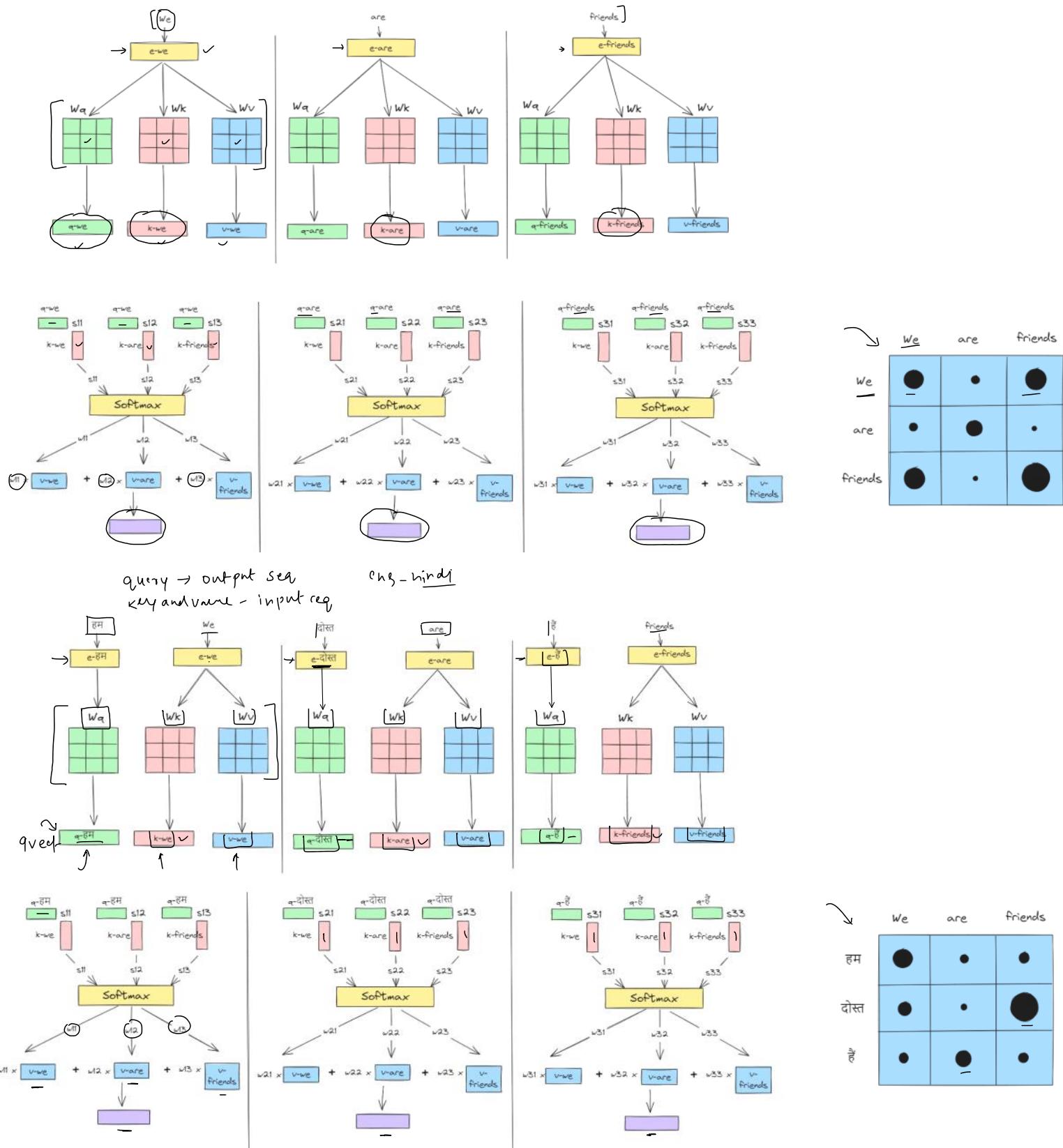
## Self-Attention Vs Cross Attention (Input)

13 August 2024 08:22



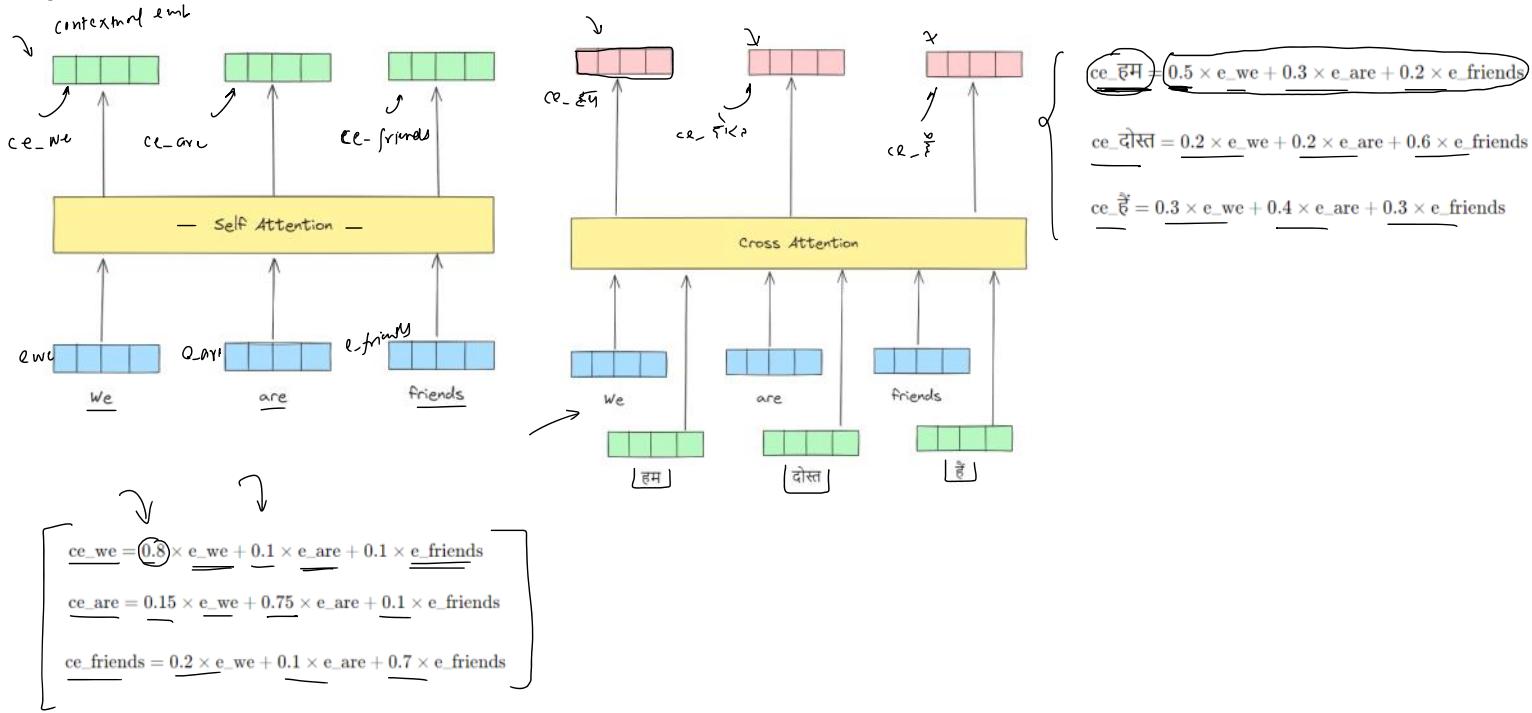
## Self-Attention Vs Cross Attention (Processing)

12 August 2024 17:54



### Self-Attention Vs Cross Attention [Output]

12 August 2024 18:05



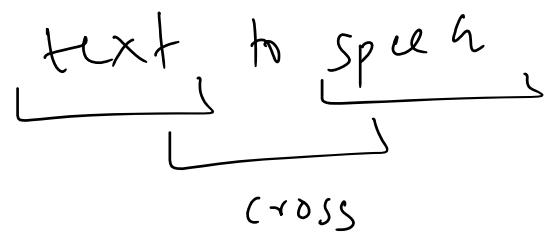
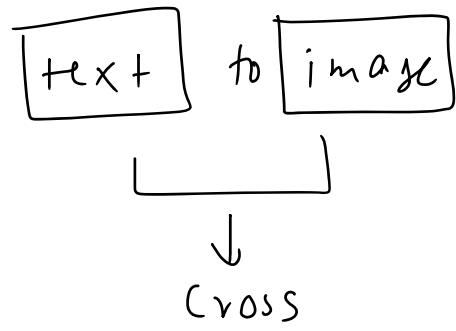
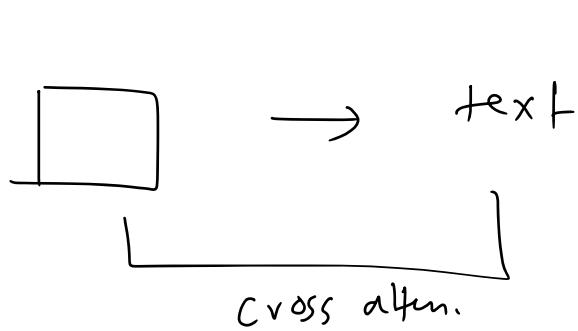
# Cross Attention Vs Bahdanau/Luong Attention

12 August 2024 18:06

## Use-cases

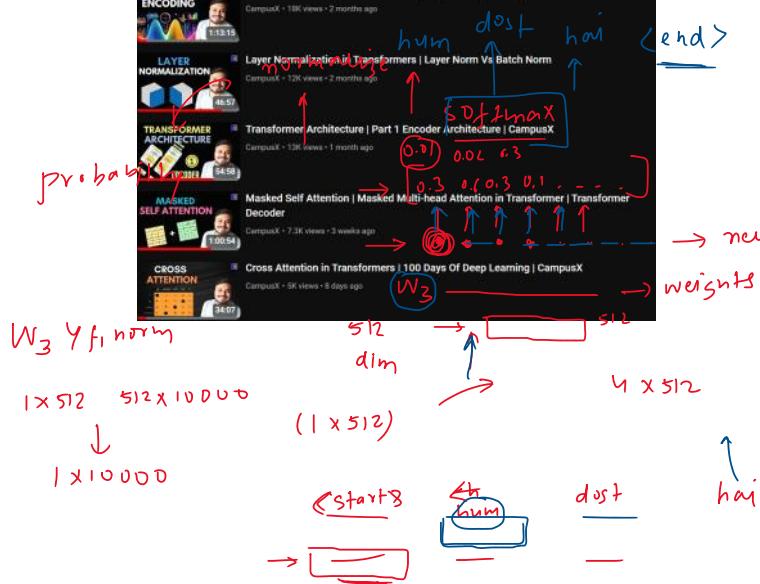
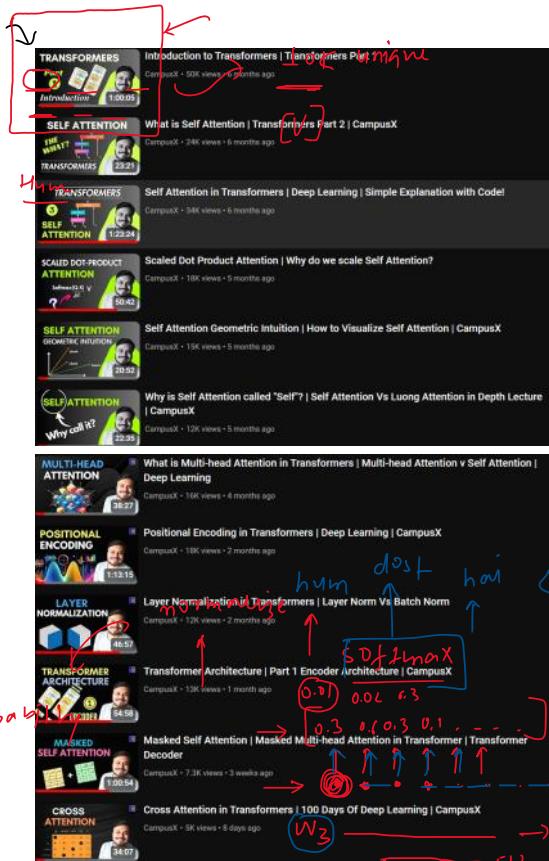
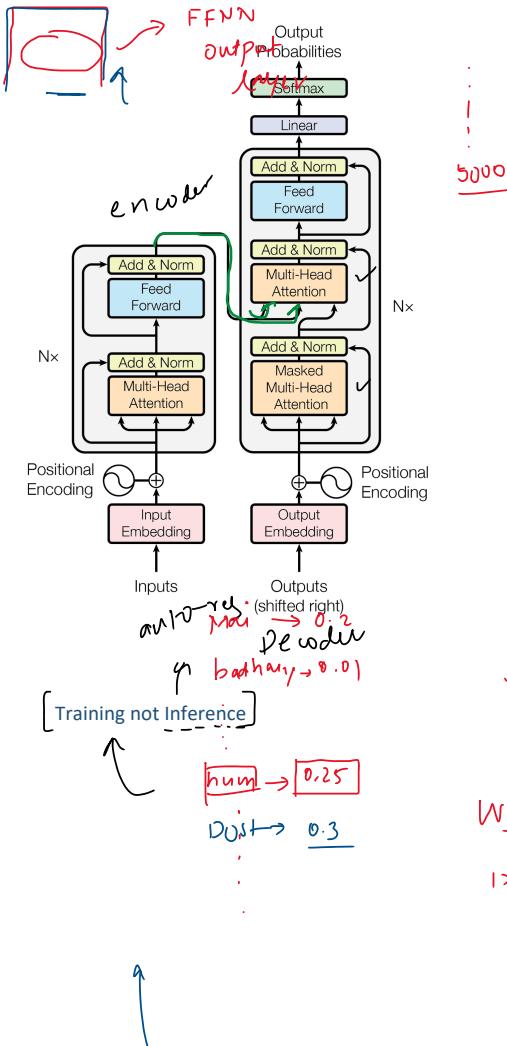
12 August 2024 18:07

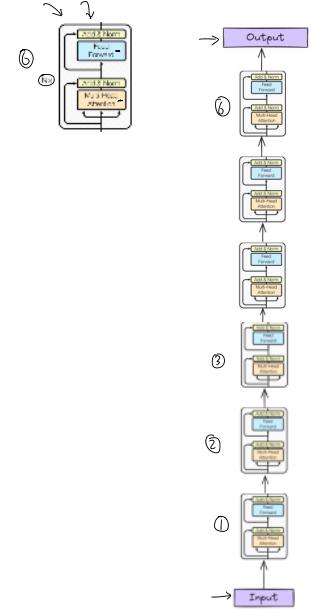
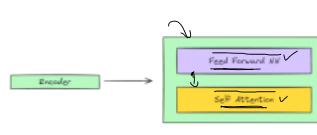
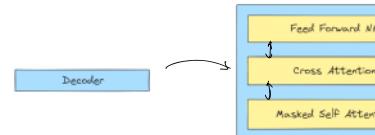
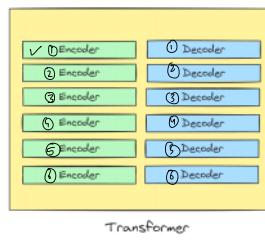
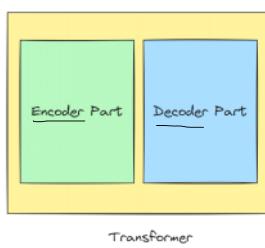
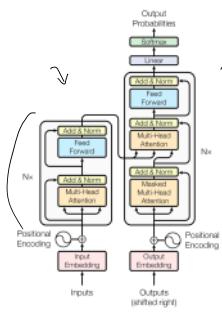
image caption

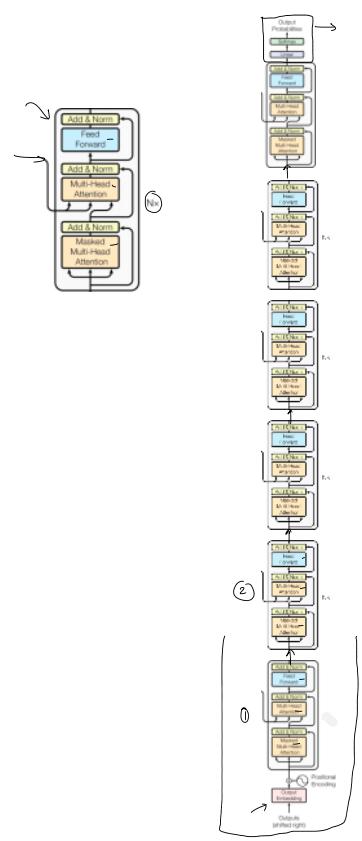


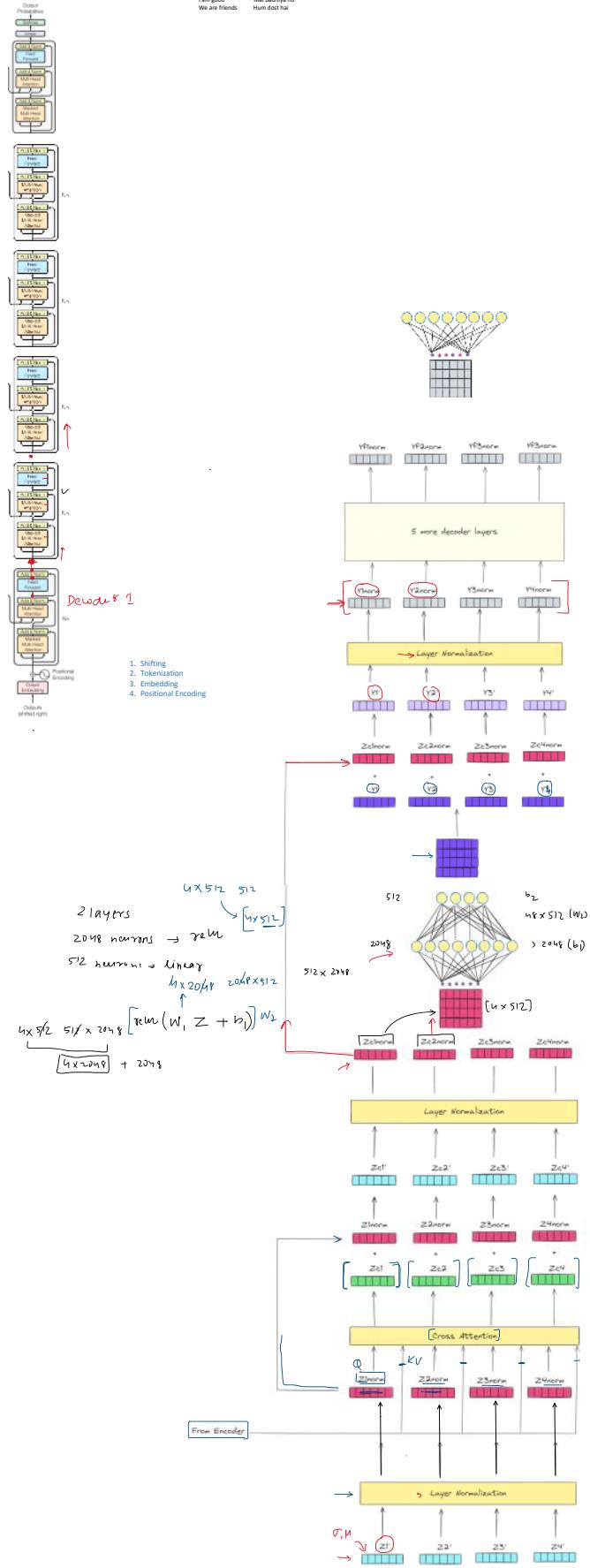
## Plan of Attack

22 August 2024 15:36

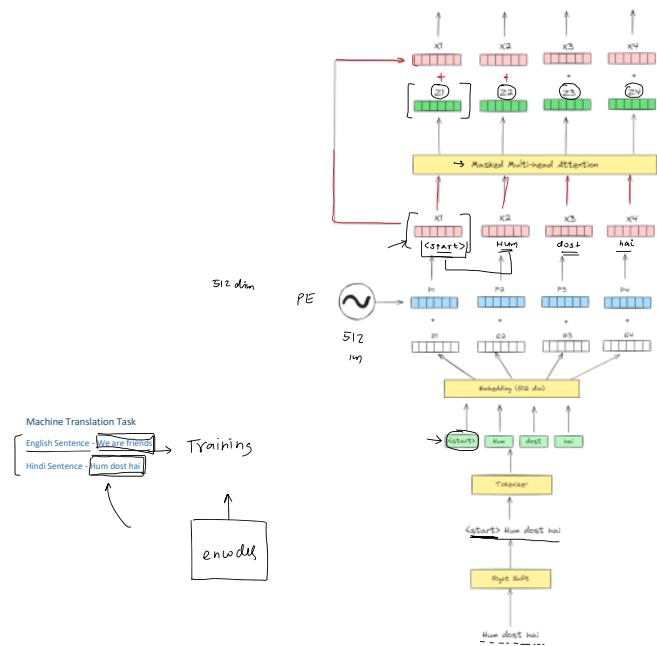










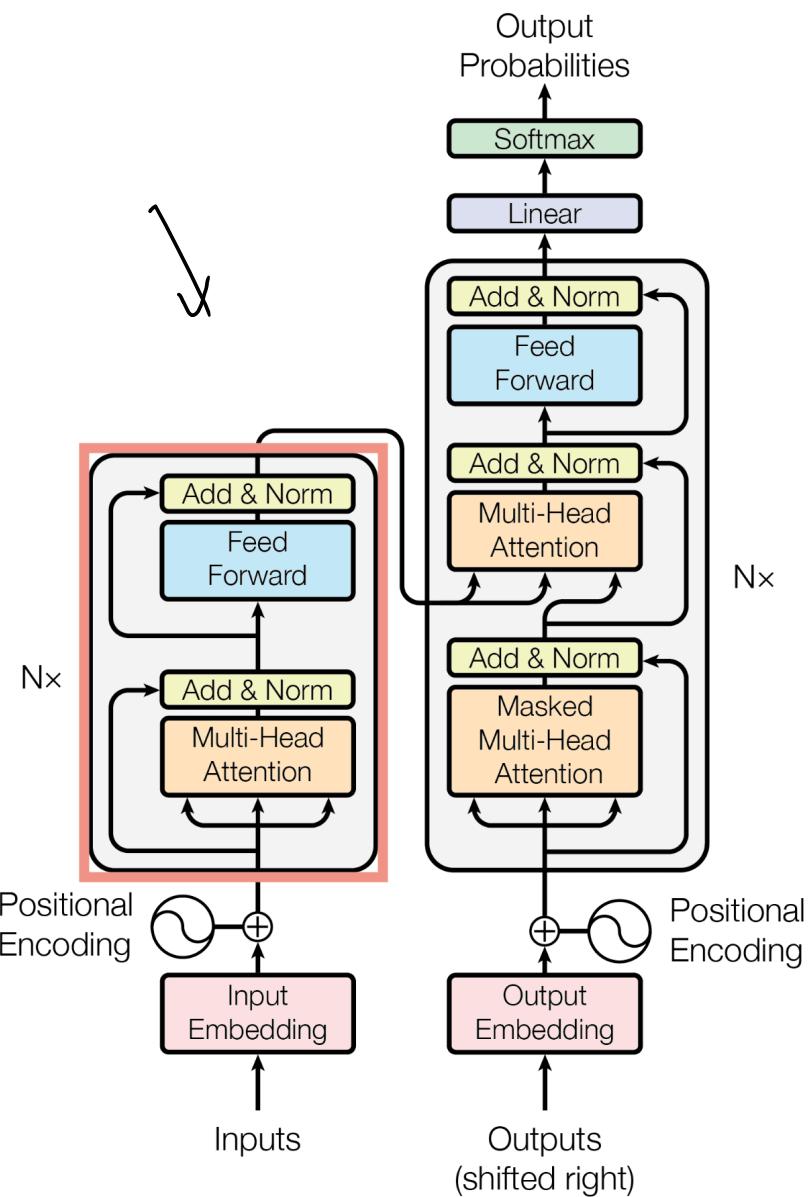




# Plan of Attack

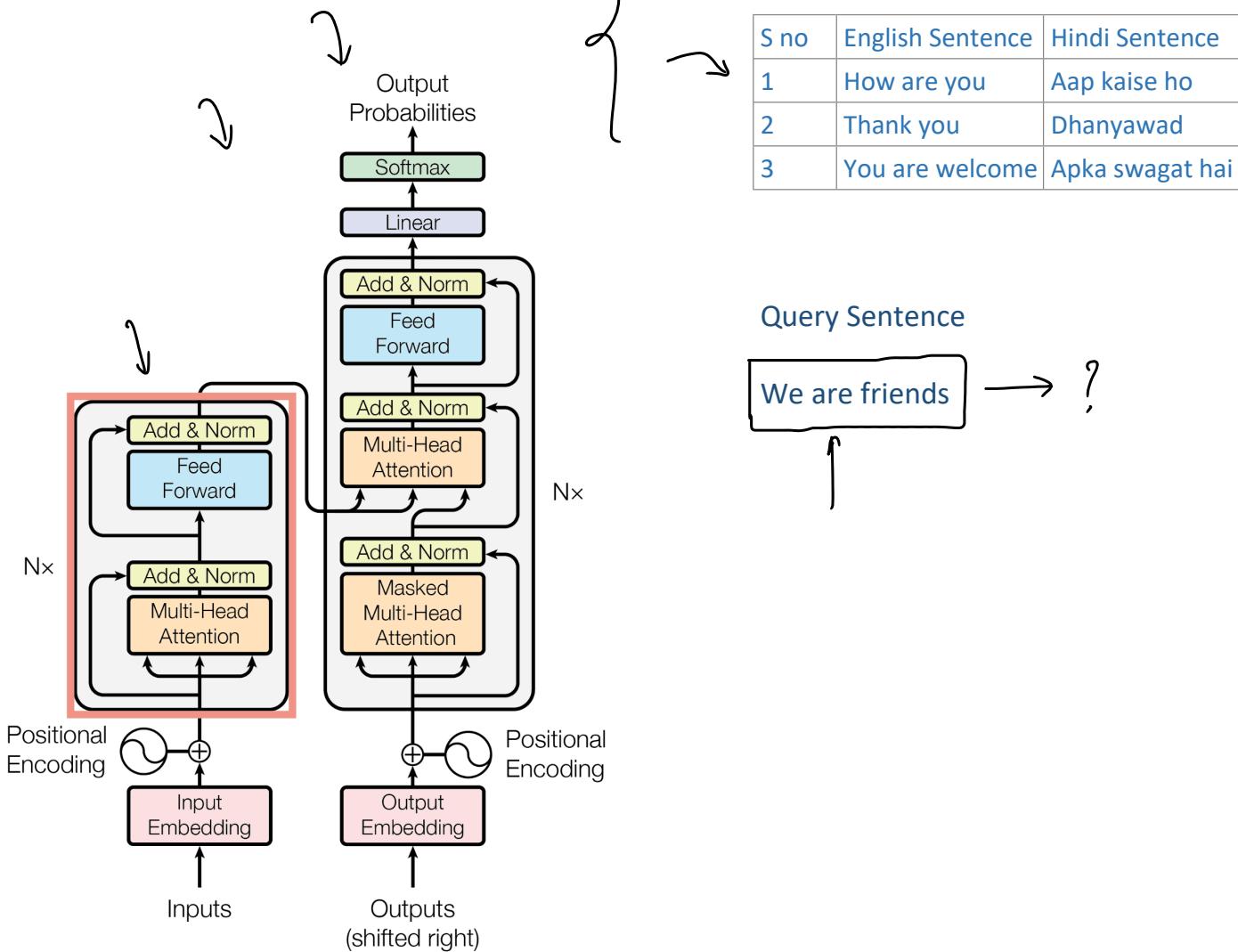
03 September 2024 00:37

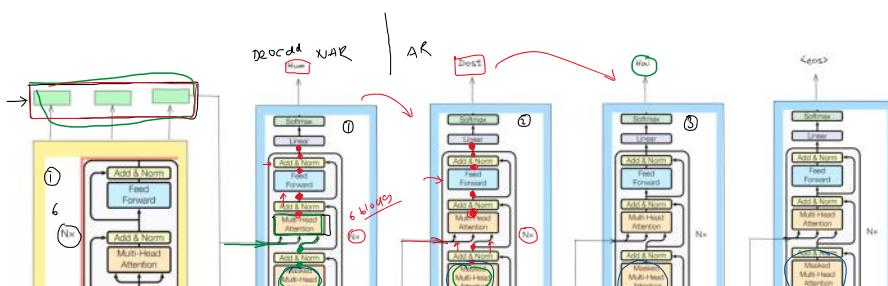
trainings

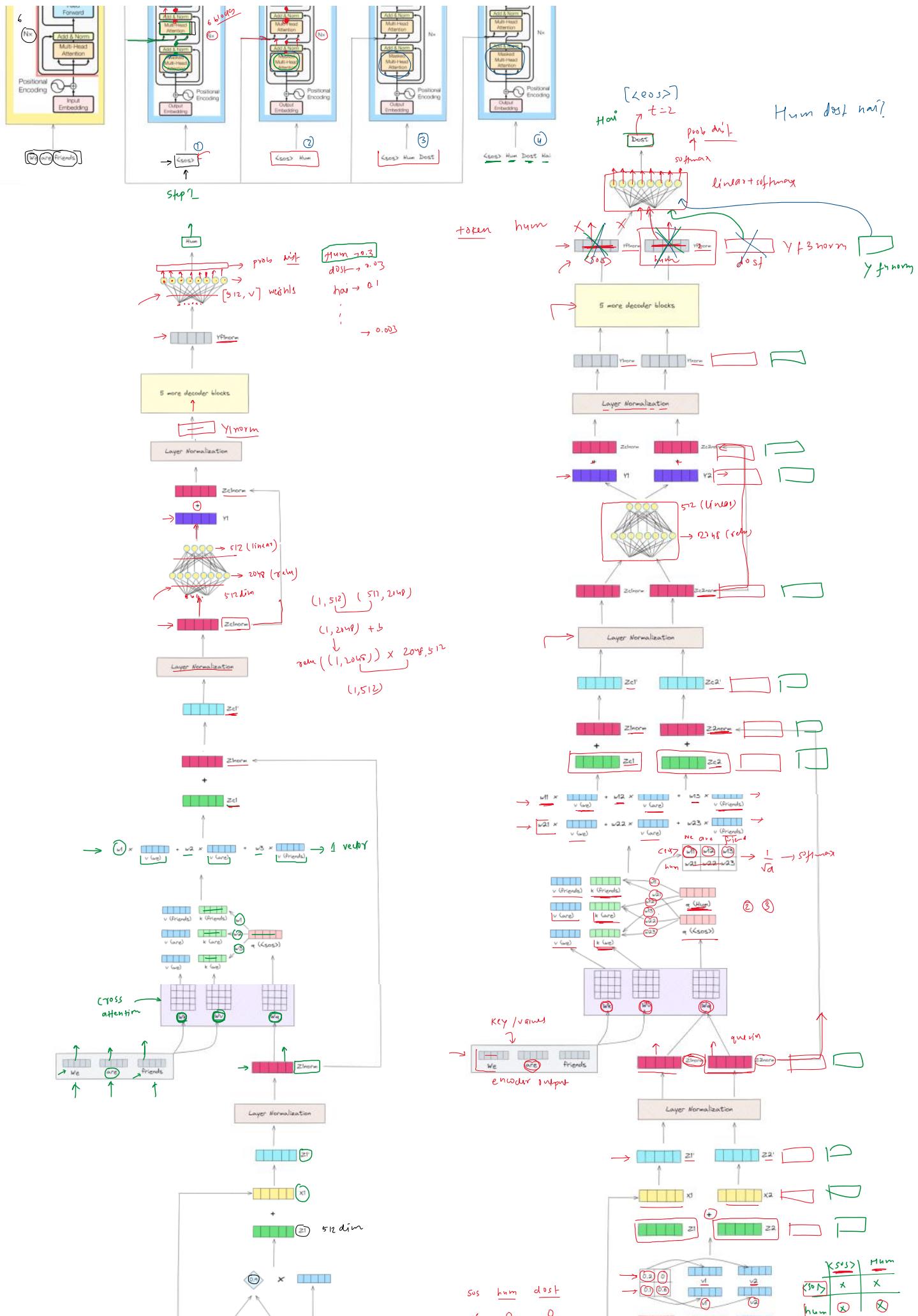


# The Setup

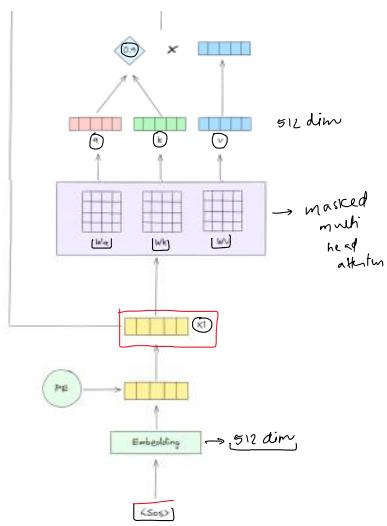
03 September 2024 00:39







**FCNN**  
**CROSS**  
**msce atr.**  
 $\uparrow$   
 $x_1$



<u>sos</u>	<u>hum</u>	<u>dost</u>	
<u>sos</u>	X	0	0
<u>hum</u>	X	X	0
<u>dost</u>	X	X	X

