Introduction

16 November 2023 16:15

LLMs / Chat / openAI

65 videos

Deep Learning

1. ANN
   - how to improve performance
2. CNN
   - CNN
   - Transfer learning
3. RNN
   - RNN
   - LSTM / GRU
4. Seq2seq models
   - Encoder Dec
   - Attention
   - Transformer
   - Fine-tuning
5. unsupervised deep learning
   - GAN
   - Autoencoders

Sequence Tasks and its types

16 November 2023 16:16

seq2seq → RNN → sequential → sequence → lang/sentence, time series, bioinform → [seq2seq model]

RNN ↓ Types of RNN

4

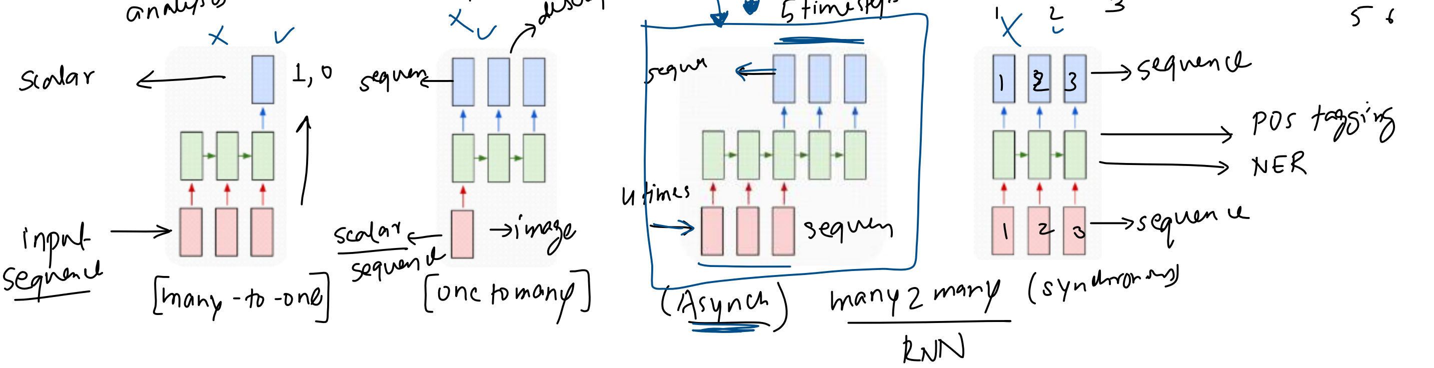
sentiment analysis — text

image caption → google image search

machine translation

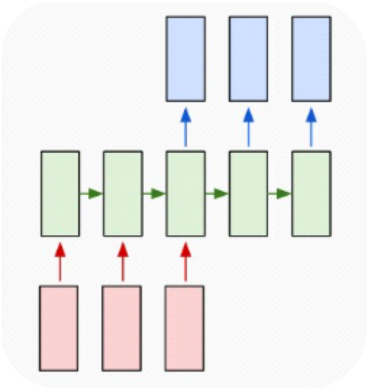
google translate

I love India → मैं भारत से प्यार करता हूं



Seq2Seq Tasks

16 November 2023 16:16



NLP

Seq2seq → machine trans

1) text summari → seq → seq

2) Question answer → seq → seq

knowledge base

3) chatbot → input (text) → output (text)

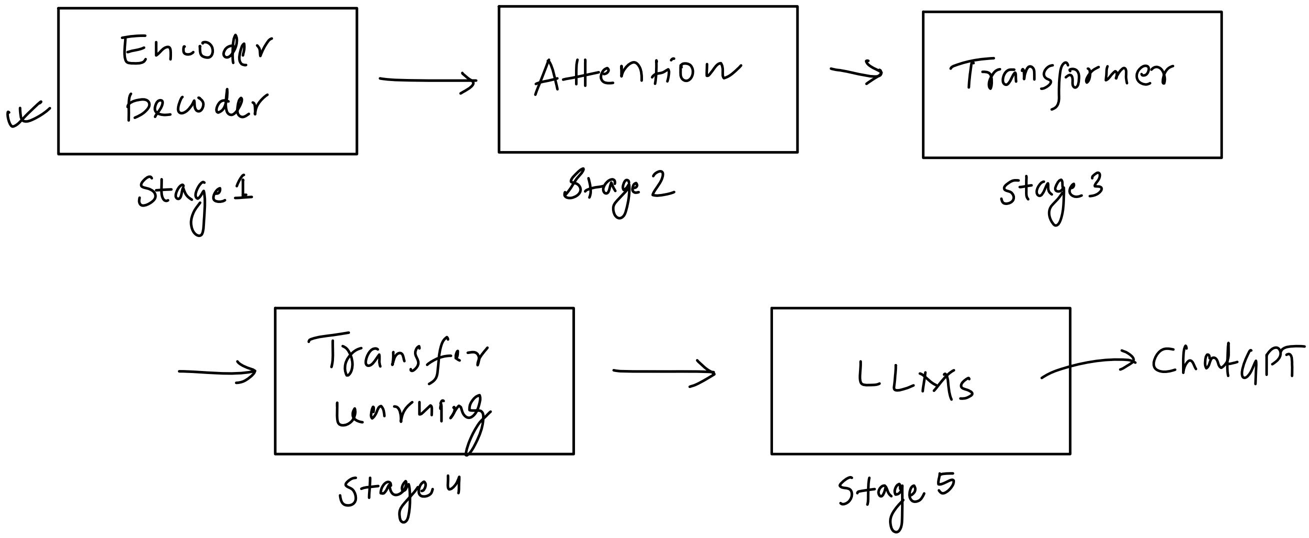
4) speech-to-text → → seq2seq → chatgpt

seq seq

History of Seq2Seq Models

16 November 2023 16:16

ChatGPT



Stage 1 - Encoder Decoder Architecture

18 November 2023 16:16

2014 seminal

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

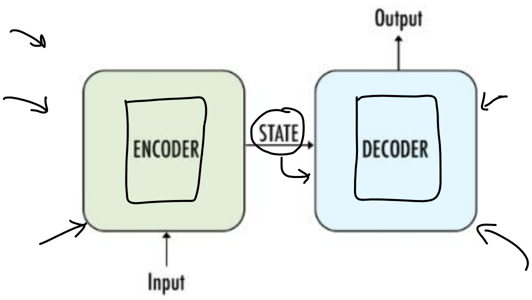
Abstract

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 multilayered 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 on 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.

[Ilya Sutskever] → cofounder OpenAI

seq2seq
↓
diff
↳ encoder decoder

LSTM →



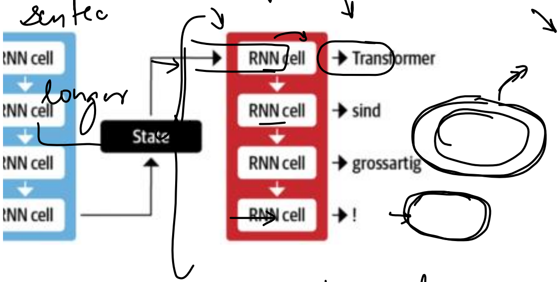
encoder/decoder 100 words

small sentec

longer

→ RNN/LSTM/GRU

context vector



internal C_t / h_t update → summarize

compressed repre

seq2seq → input → out

machine tran

eng → hindi

← compress ← process word by word

I love India → मुझे भारत से प्यार है

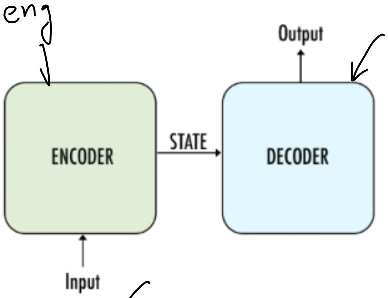
Attention mechanism

Stage 2 - Attention Mechanism

20 November 2023 10:59

eng

hindi



Sadly mistaken, he realized that the job offer was actually an incredible opportunity that would lead to significant personal and professional growth."

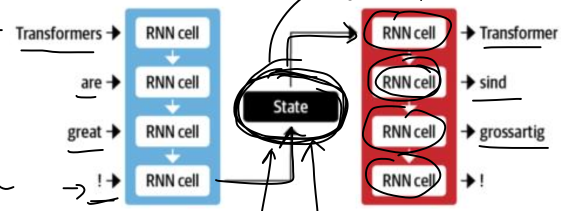
summ

long input → 30 words

context vector

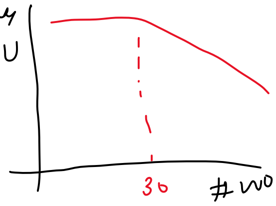
sadly

m



translation quality

BLEU



→ encoder decoder

2015

Attention

→ 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.

1 INTRODUCTION

Neural machine translation is a newly emerging approach to machine translation, recently proposed by Kalchbrenner and Blunsom (2013), Sutskever *et al.* (2014) and Cho *et al.* (2014b). Unlike the traditional phrase-based translation system (see, e.g., Koehn *et al.*, 2003) which consists of many small sub-components that are tuned separately, neural machine translation attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation.

Most of the proposed neural machine translation models belong to a family of *encoder–decoders* (Sutskever *et al.*, 2014; Cho *et al.*, 2014a), with an encoder and a decoder for each language, or involve a language-specific encoder applied to each sentence whose outputs are then compared (Hermann and Blunsom, 2014). An encoder neural network reads and encodes a source sentence into a fixed-length vector. A decoder then outputs a translation from the encoded vector. The whole encoder–decoder system, which consists of the encoder and the decoder for a language pair, is jointly trained to maximize the probability of a correct translation given a source sentence.

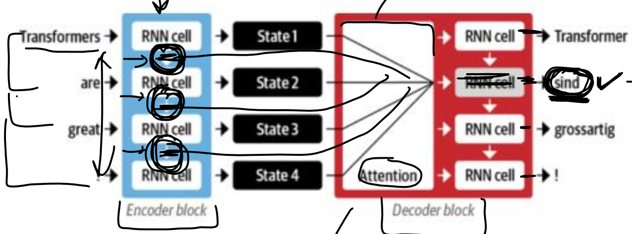
A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector. This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus. Cho *et al.* (2014b) showed that indeed the performance of a basic encoder–decoder deteriorates rapidly as the length of an input sentence increases.

c_t h_t

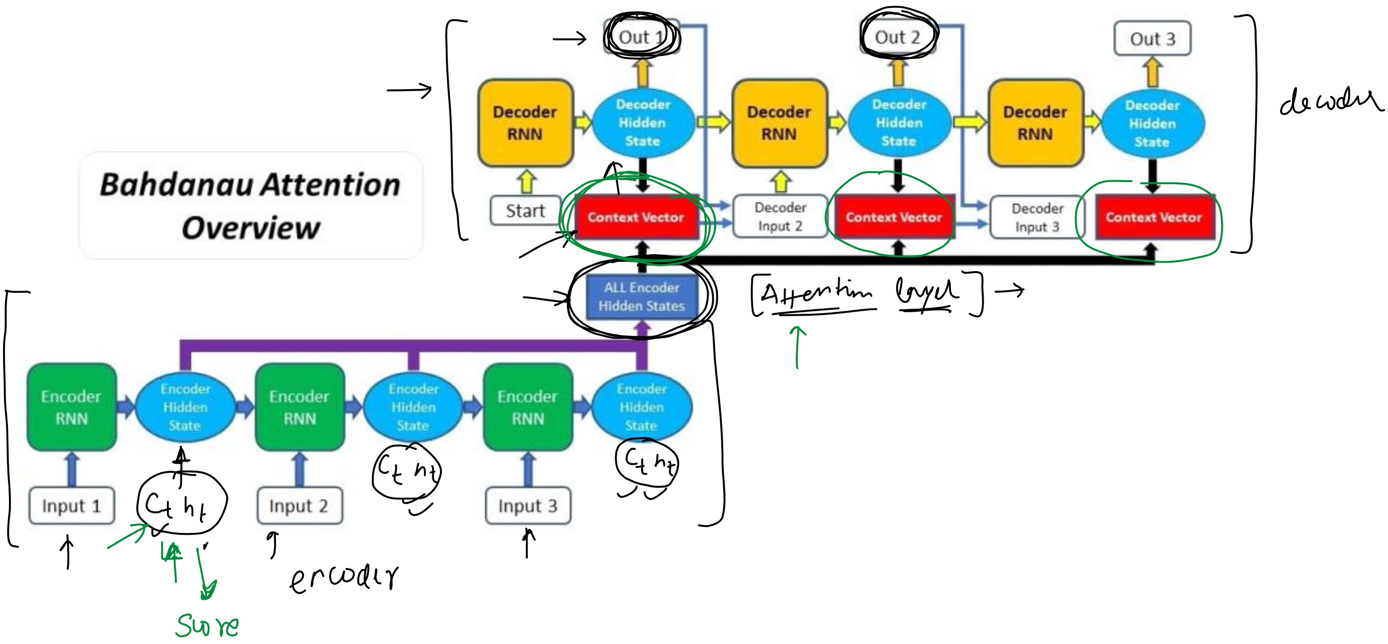
LSTM

context vector

NN →



mechanism


Bahdanau Attention Overview
Out 1
Out 2
Out 3
Decoder RNN
Decoder Hidden State
Start
Context Vector
Decoder Input 2
Decoder Input 3
ALL Encoder Hidden States
Encoder RNN
Encoder Hidden State
Input 1
Input 2
Input 3
decoder
[Attention layer]
encoder
C_t h_t
Score

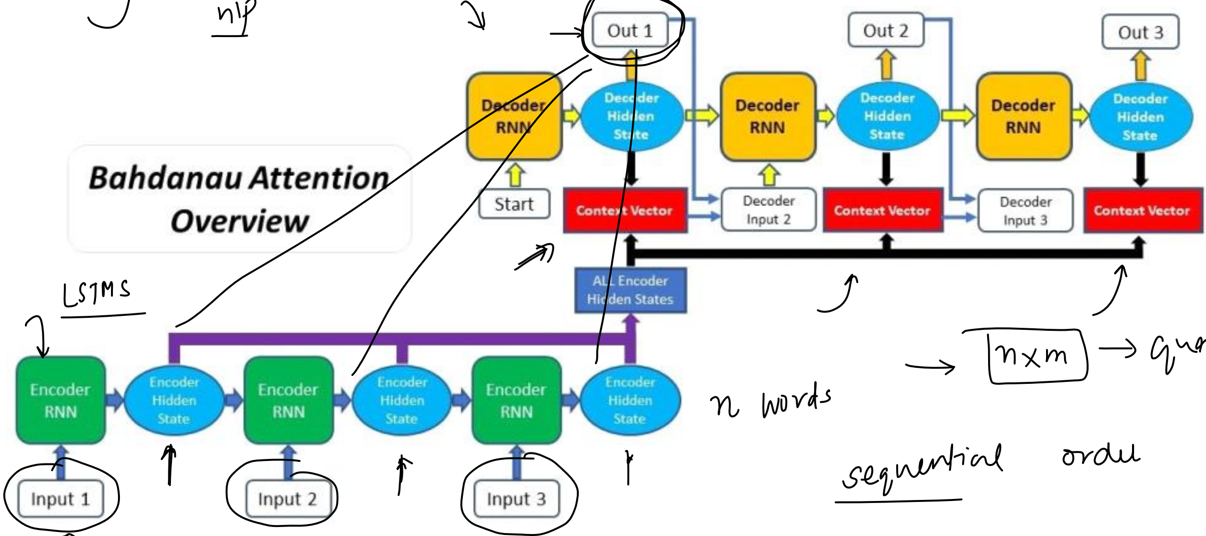
Stage 3 - Transformers
20 November 2023 12:18

nlp

computation complexity

m words

2015-2017



atten

encoder decoder

parallel processing

LSTMs

n words

n x m → quadratic

sequential order

Attention Is All You Need

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
Łukasz 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.

2017

LSTM ×

LSTM

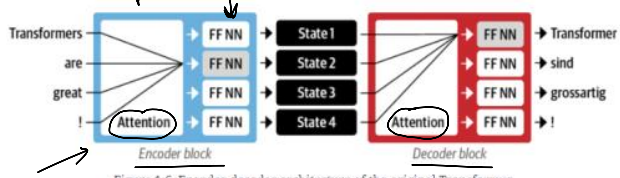


Figure 1-6. Encoder-decoder architecture of the original Transformer

LSTM / RNN cell

Attention

Self-attention

stage

arch

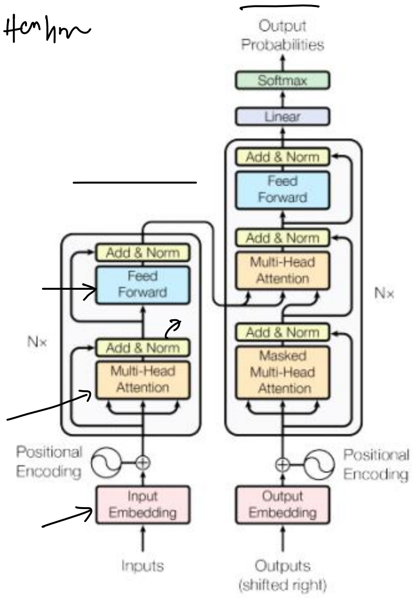


Figure 1: The Transformer - model architecture.

Stage 4 - Transfer Learning

20 November 2023 15:39

1) Hardware → GPUs
2) Time → → $\boxed{100, 1000}$
3) Data → a lot

↓

transformer → Transfer Learning

ULMFiT

Universal Language Model Fine-tuning for Text Classification

Jeremy Howard*
fast.ai
University of San Francisco
j@fast.ai

Sebastian Ruder*
Insight Centre, NUI Galway
Aylien Ltd., Dublin
sebastian@ruder.io

ULMfit

2018 → vision

transfer → nlp tasks

primary pretrain

Abstract

Inductive transfer learning has greatly impacted computer vision, but existing approaches in NLP still require task-specific modifications and training from scratch. We propose Universal Language Model Fine-tuning (ULMFiT), an effective transfer learning method that can be applied to any task in NLP, and introduce techniques that are key for fine-tuning a language model. Our method significantly outperforms the state-of-the-art on six text classification tasks, reducing the error by 18-24% on the majority of datasets. Furthermore, with only 100 labeled examples, it matches the performance of training from scratch on 100× more data. We open-source our pretrained models and code[1].

1 Introduction

Inductive transfer learning has had a large impact on computer vision (CV). Applied CV models (including object detection, classification, and segmentation) are rarely trained from scratch, but instead are fine-tuned from models that have been pretrained on ImageNet, MS-COCO, and other datasets (Sharif Razavian et al., 2014; Long et al., 2015a; He et al., 2016; Huang et al., 2017).

Text classification is a category of Natural Language Processing (NLP) tasks with real-world applications such as spam, fraud, and bot detection (Jindal and Liu, 2007; Ngai et al., 2011; Chu et al., 2012), emergency response (Caragea et al., 2011), and commercial document classification, such as for legal discovery (Roitblat et al., 2010).

While Deep Learning models have achieved state-of-the-art on many NLP tasks, these models are trained from scratch, requiring large datasets, and days to converge. Research in NLP focused mostly on *transductive* transfer (Blitzer et al., 2007). For *inductive* transfer, fine-tuning pretrained word embeddings (Mikolov et al., 2013), a simple transfer technique that only targets a model's first layer, has had a large impact in practice and is used in most state-of-the-art models. Recent approaches that concatenate embeddings derived from other tasks with the input at different layers (Peters et al., 2017; McCann et al., 2017; Peters et al., 2018) still train the main task model from scratch and treat pretrained embeddings as fixed parameters, limiting their usefulness.

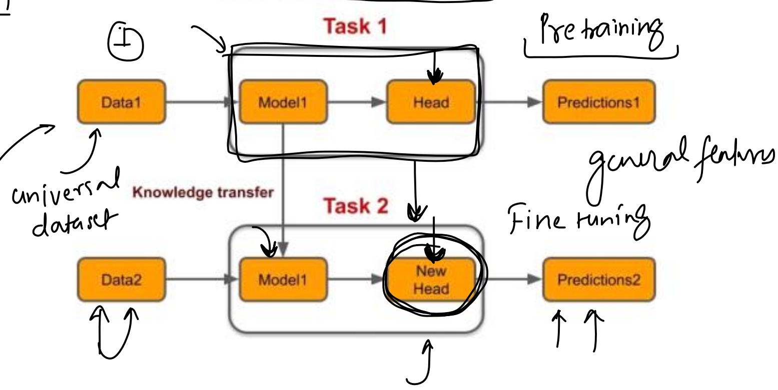
In light of the benefits of pretraining (Erhan et al., 2010), we should be able to do better than *randomly initializing* the remaining parameters of our models. However, inductive transfer via fine-tuning has been unsuccessful for NLP (Mou et al., 2016). Dai and Le (2015) first proposed fine-tuning a language model (LM) but require millions of in-domain documents to achieve good performance, which severely limits its applicability.

We show that not the idea of LM fine-tuning but our lack of knowledge of how to train them effectively has been hindering wider adoption. LMs overfit to small datasets and suffered catastrophic forgetting when fine-tuned with a classifier. Compared to CV, NLP models are typically more shallow and thus require different fine-tuning methods.

We propose a new method, Universal Language Model Fine-tuning (ULMFiT) that addresses these issues and enables robust inductive transfer learning for any NLP task, akin to fine-tuning ImageNet

NLP

Transfer Learning → useful



Transfer learning (TL) is a technique in which knowledge learned from a task is re-used in order to boost performance on a related task.

For example, for image classification, knowledge gained while learning to recognize cars could be applied when trying to recognize trucks.

imagenet → dataset → Millions of
↳ pretrain → ↳ CNN → ResNET
↳ imagen
edges
↳ shapes
↓
fine-tune → cat vs dog classify → 100

2018

1) Task specificity → NLP

Senti, NER, POS, ML, QA, summ

language

NLP → transfer learning

ML

2nd) Lack of data

eng | hin

NLP → transfer learning

Pretraining → Machine transl ~~X~~

↳ language modelling → pretraining → successful → (2)

NLP task → NLP/DL model next word predi

I live in india. and the capital is New Delhi

Language modelling as a Pretraining task

↳ unsupervised pretrain task

1) Rich feature learning

The hotel was exceptionally clean, yet the service was bad / pathetic

→ know trans

↓

text class / quest. | text summ | NER / pos

2) Huge avail of data

pdf → dataset labelling

mt (labw → supervised labeled

eng | hin

→ unsupervised task

fine tuning

[ULMFit]

X transformer

AWD LSTM → wikipedia → [ ] → imdb / yelp / new data → model → test → State of the art

classifier

Unsupervised pretrain Language modelling

Scratch → 10000 rows

100 row → better →

2012 2018

2012 2018

parallely

$\boxed{2018}$

archi

[transformer]

train

[transfer leq] → ?

Stage 5 - LLMs

20 November 2023 18:51

2018 → 10 ULMFit Jan Oct

Transformer

[language]

Google

BERT [Encoder only]

OpenAI [decoder only model]

→ GPT

transformer

dataset → huge

→ Transfer learning → txt summ, ques, sentim, ner, pos

download

limited dataset

finetune

SOTA

GPT → GPT2 → GPT3 → language model → LLM → LLMs

→ parameter

→ dataset

[LM → LLMs]

Large scale LLMs

Qualities of LLMs

1) Data → billions → GPT3 → 45 TBs

→ book, websites, internet

→ diversity → bias

2) Hardware → Cluster of GPU → GPT3 → Supercomputer → 1000s NVIDIA GPU

3) Training → days to weeks

4) Cost → hardware + electc + infra + experts → individual

→ millions

companies

govt

Institutes

4) energy consump

gpt3 →

↳ energy consump

↳ gpt3 → small town

↳ month

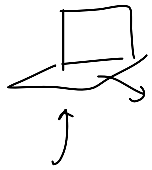
The Grand Finale - ChatGPT

22 November 2023 10:13

GPT3 → chatGPT
↓
[diff]

GPT → model
[ChatGPT] → application
Chatbot

Nov



hp → chatgpt
intel → GPT → Bard, Jasper
dell
ASUS

GPT3 → [ChatGPT]

1) RLHF → Reinfo lear from human feed
   - 1 Supervised Fine tuning → dataset
   - 2 Reinforce → — prompt produces
     - responses
     - human → respone rank

labelled

2) Incorporate safety and ethical guideline
   - minize biases

3) improvent in contextual point
   context retain → maintain context } dialogu conver

4) Dialogue specific training
   - conversation
   - better under → dialogue lang → patterns

5) ChatGpt continous imp → human feedbac
   ↳ user

train → refining

[GPT4] → [GPT3]

Seq2Seq Data

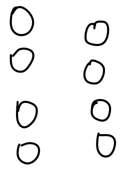
08 December 2023 17:12

cgpa | iq | placement

Encoder Decoder Archi

DNN

1) Tabular → [ANN] ×



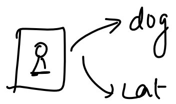
3) Sequential → textual
→ timeseries

↓

_ _ _ _

→ RNN
→ LSTMs
→ GRUs

2) Image data → [2d grid] → CNN ×



4) Seq2seq data — difficult

[input seq → output seq] — machine translation

nice to meet you

↓

आप से मिल कर अच्छा लगा

input → sent → eng → variable length

output → sent → hindi → variable leng

→ variable length

↓

[lstm / gru] → input

[output]

3 words eng → 3 words

Before Starting

08 December 2023 19:24

Prerequisite

- RNN / LSTM

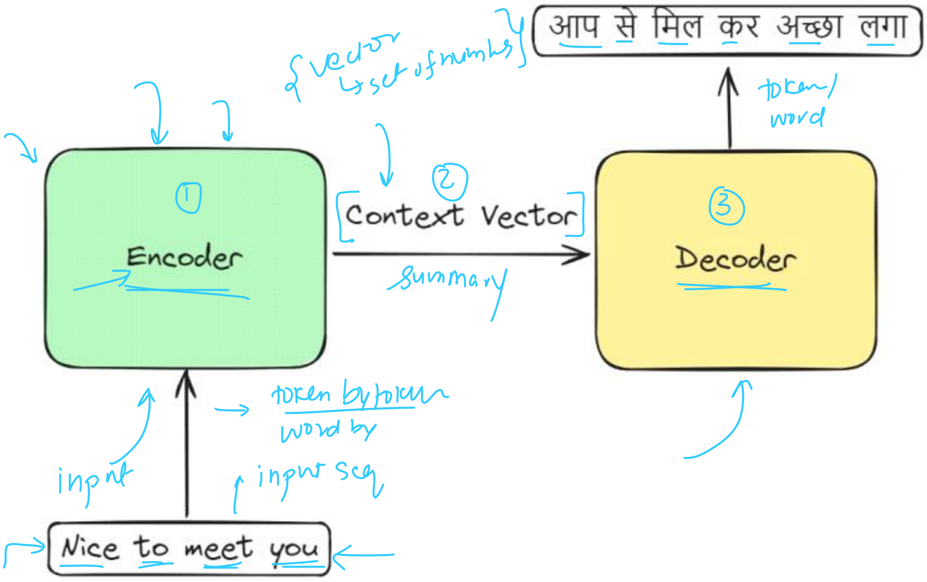
Plan of attack

→ simple version
→ deep
→ improvements

High Level Overview

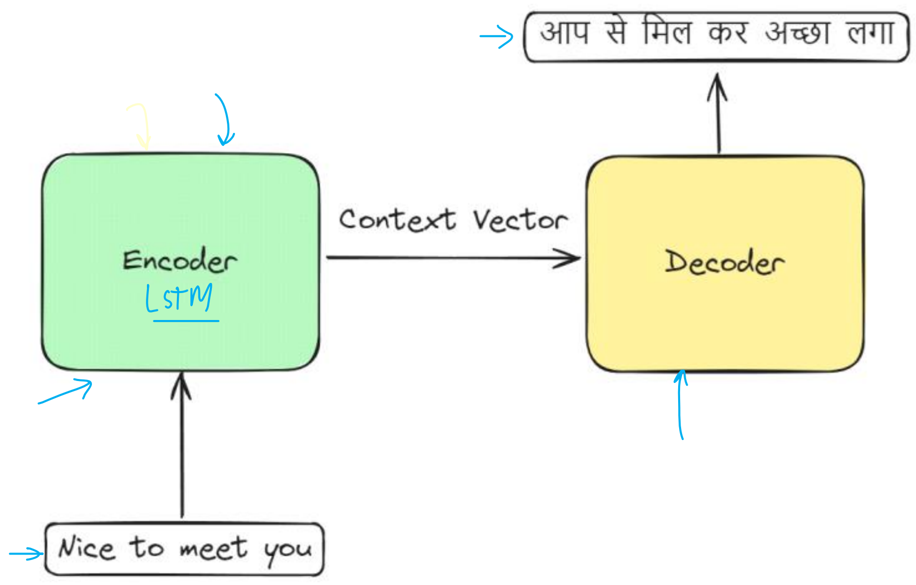
08 December 2023 17:14

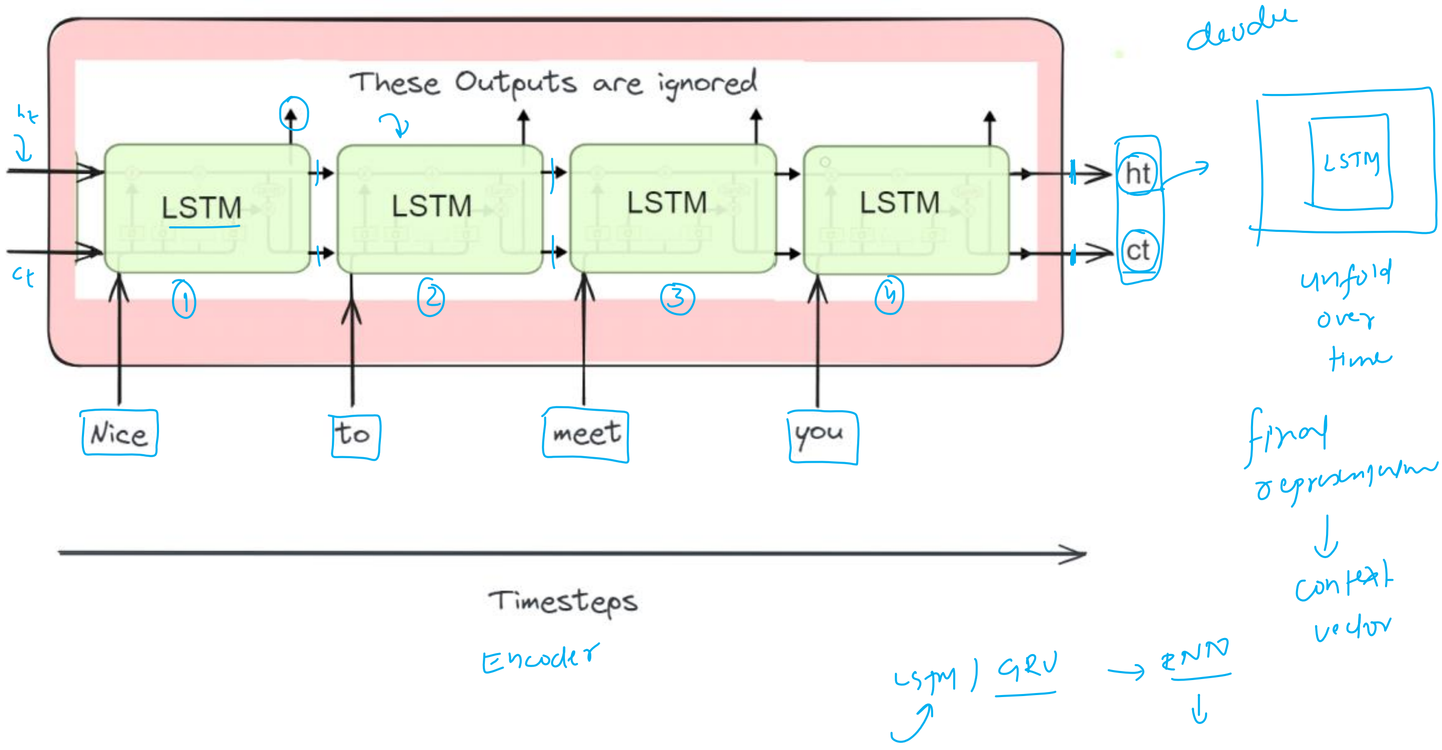
Machine tran → exam

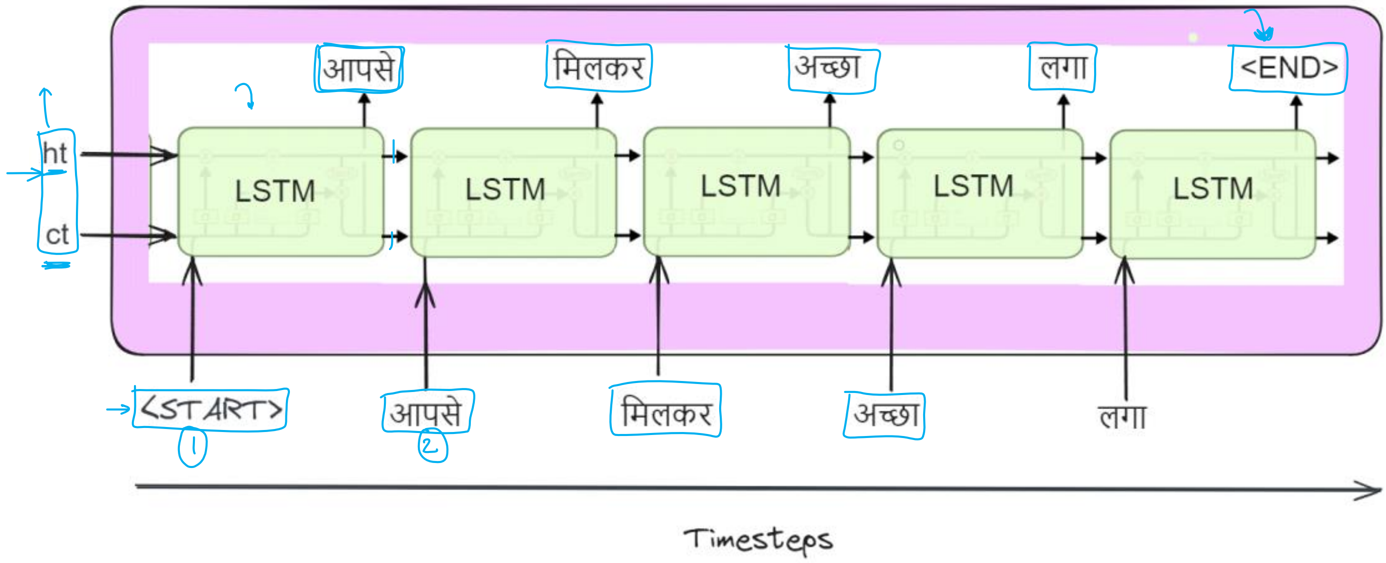


What's under the hood?

08 December 2023 17:14



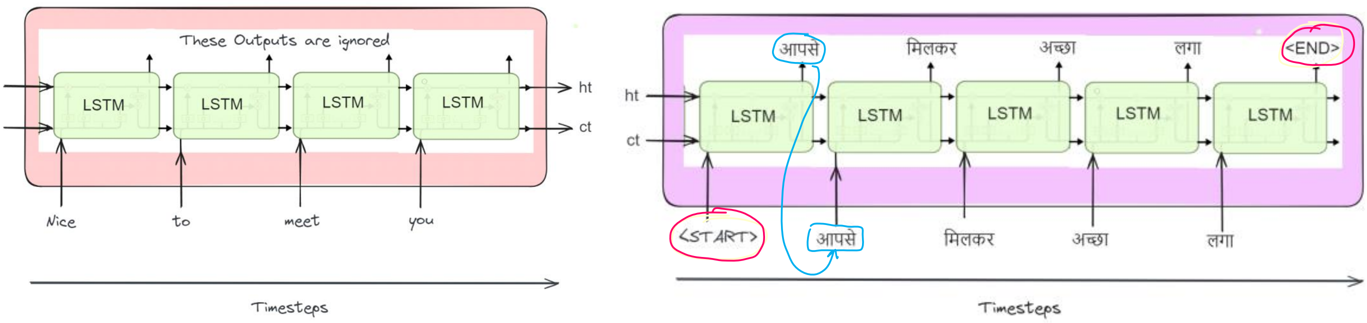


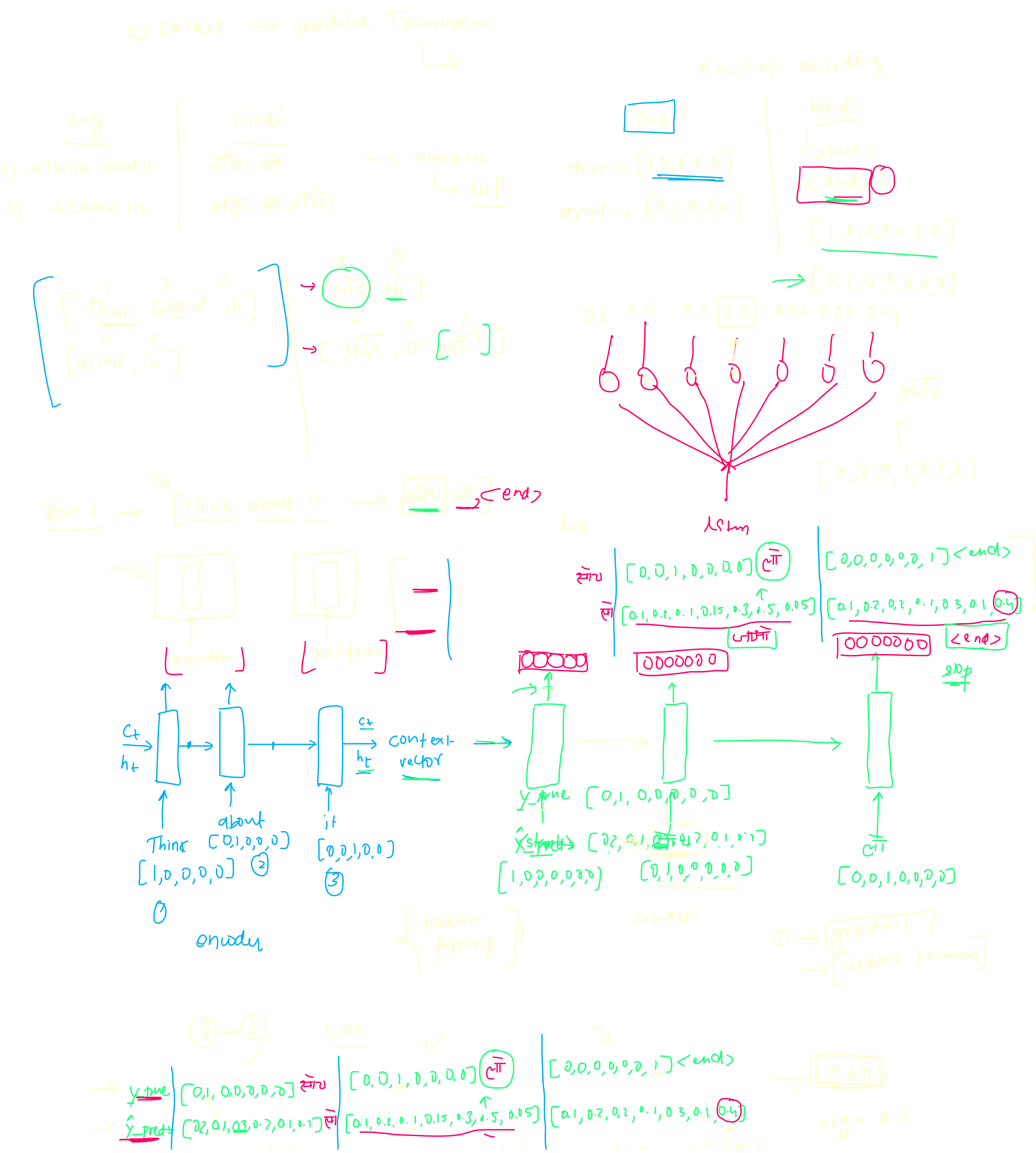


आपसे
मिलकर
अच्छा
लगा
<END>
ht
ct
LSTM
LSTM
LSTM
LSTM
LSTM
<START>
1
आपसे
2
मिलकर
अच्छा
लगा
Timesteps

Training the Architecture using Backpropagation

08 December 2023 17:15





$\rightarrow$ y_true [0, 1, 0, 0, 0, 0, 0]

$\rightarrow$ $\hat{y}$_pred $\rightarrow$ [0.2, 0.1, 0.3, 0.2, 0.1, 0.1] | [0.1, 0.2, 0.1, 0.15, 0.3, 0.5, 0.05] | [0.1, 0.2, 0.2, 0.1, 0.3, 0.1, 0.4]

t=1 (1) | t=2 (1) | t=3 (0.39)

avg → 0.7

1) fwd
2) loss
3) Backprop

Loss → categorical cross entropy

$$L = -\sum_{i=1}^{7} y_i^{true} \log(y_i^{pred})$$

$$L_{t=1} = -1 \times \log(0.1) = 1$$

$$L_{t=2} = 1$$

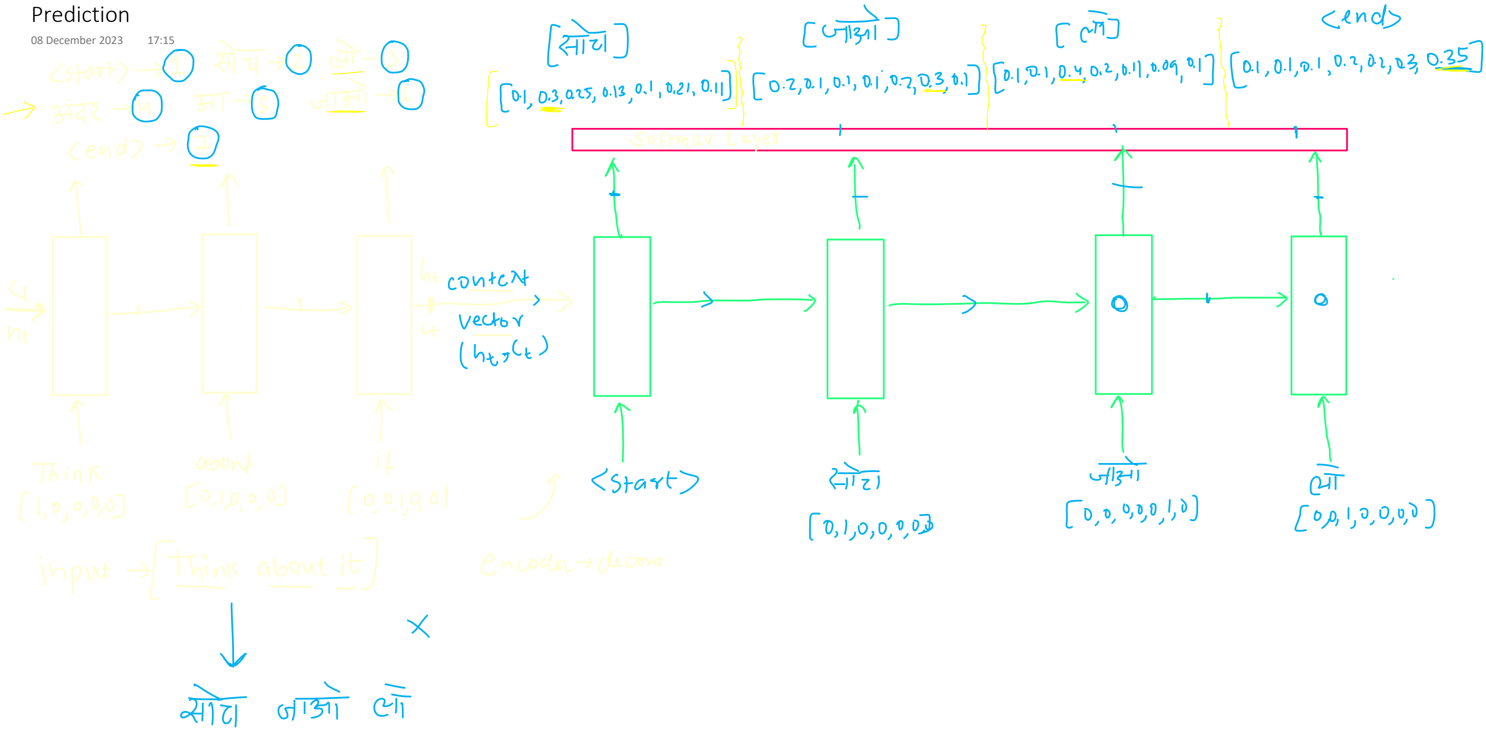
$$L_{t=3} = -1 \times \log(0.4) = -1 \times -0.39 = 0.39$$

training

③ (initial wts) forward prop → output / true → loss → gradient → weight update (Adam) → learning rate → new weights → (initial wts)

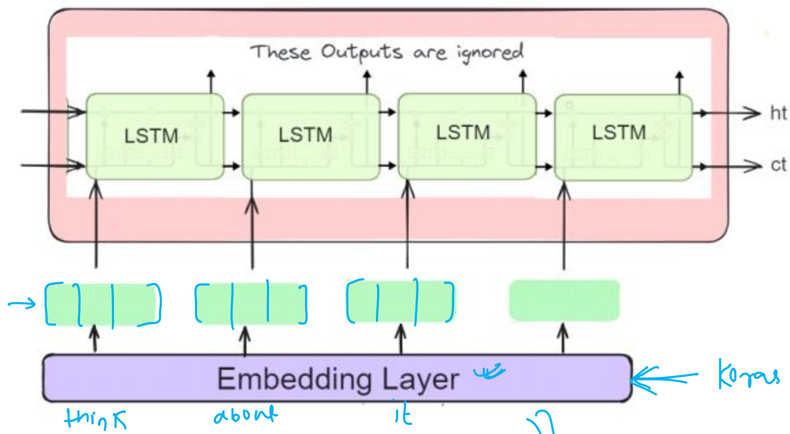
Prediction

08 December 2023 17:15



Improvement 1 - Embeddings

08 December 2023 17:16



<start

encoder

decoder

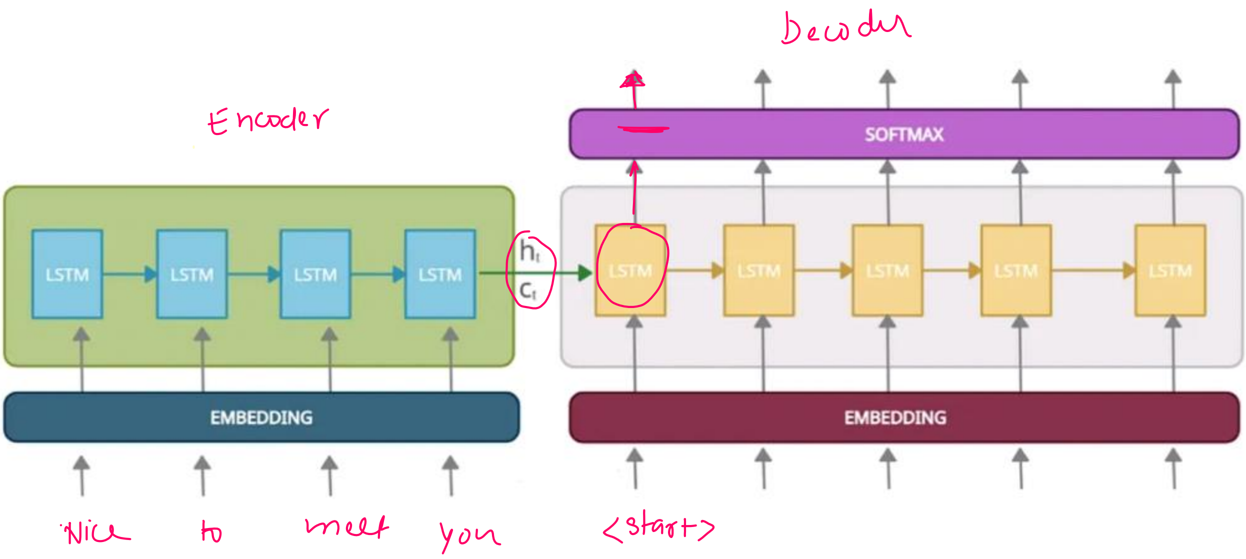
3 dim

low dimen / dense

summary of word

pretrained + word2vec / Glove

train

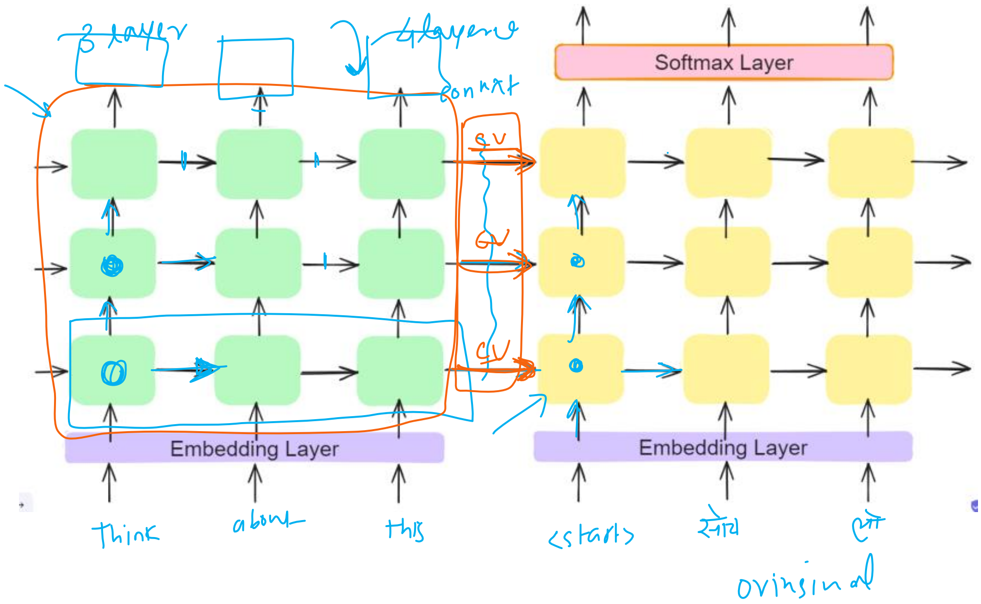


Improvement 2 - Deep LSTMs

08 December 2023 17:16

long para

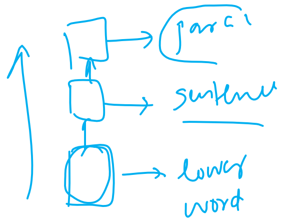
single layer



3) deep learning NN learning

1) long term dependency

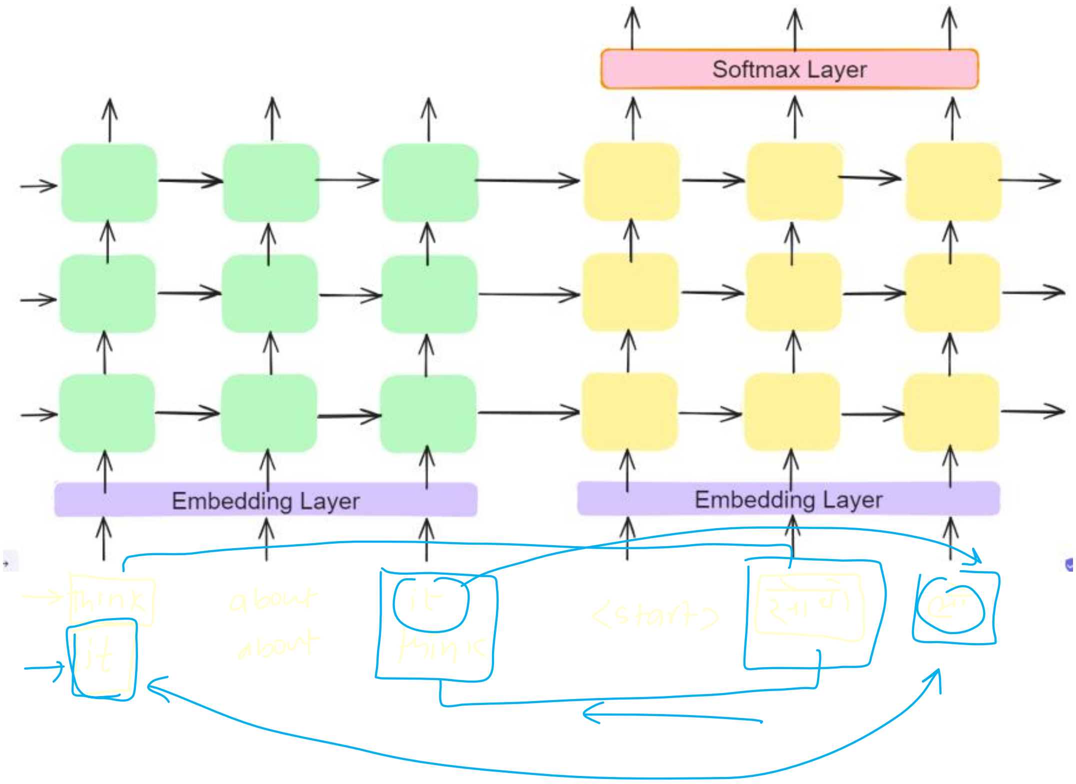
2) layered representation



the phone battery is bad but the overall phone is great

Improvement 3 - Reversing the Input

08 December 2023 17:16



language

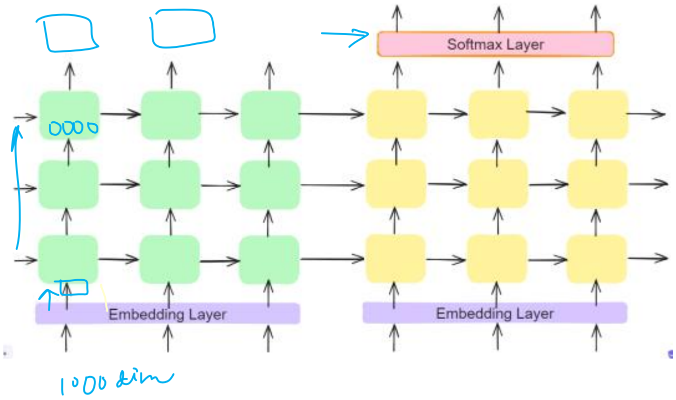
initial context

eng - french

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.

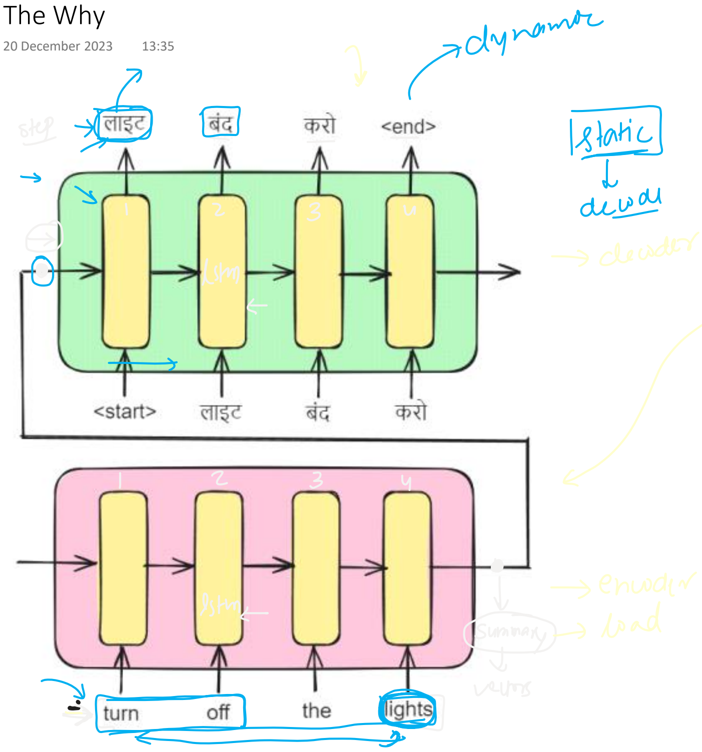
Architecture Details: Both the input (encoder) and output (decoder) models had 4 layers, with each layer containing 1,000 units, showcasing a deep LSTM-based architecture.

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



Encoder

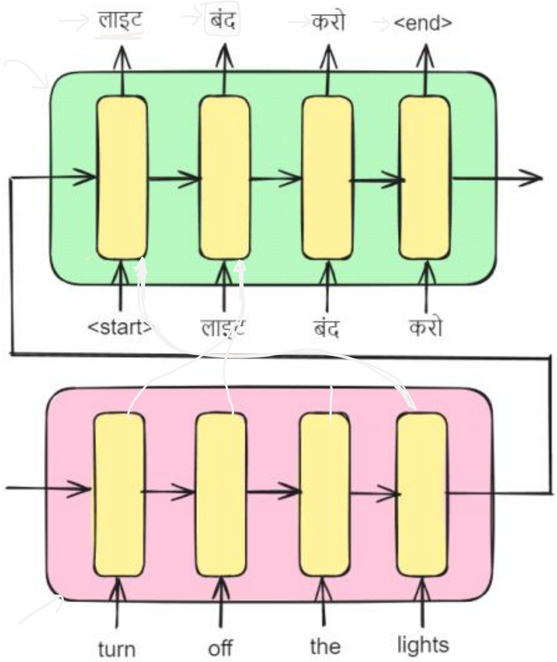
Once upon a time in a small Indian village, a mischievous monkey stole a turban from a sleeping barber, wore it to a wedding, danced with the bewildered guests, accidentally got crowned the 'Banana King' by the local kids, and ended up leading a vibrant, impromptu parade of laughing villagers, cows, 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

The Solution

20 December 2023 17:32

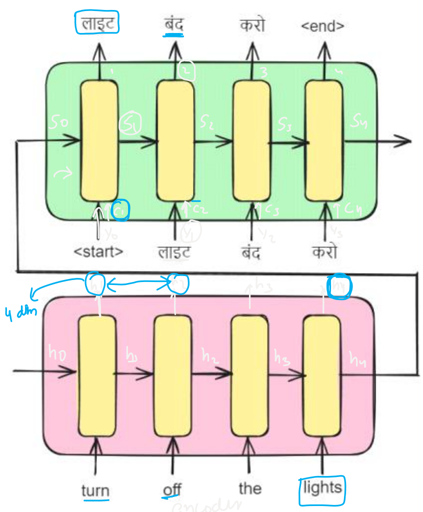
Attention



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.

The What

21 December 2023 06:04



$[1\ 0.5\ 0.1\ 0.3]$

$\to \boxed{h_1} \to 4$

$\to h_2\ h_3$ → weighed sum

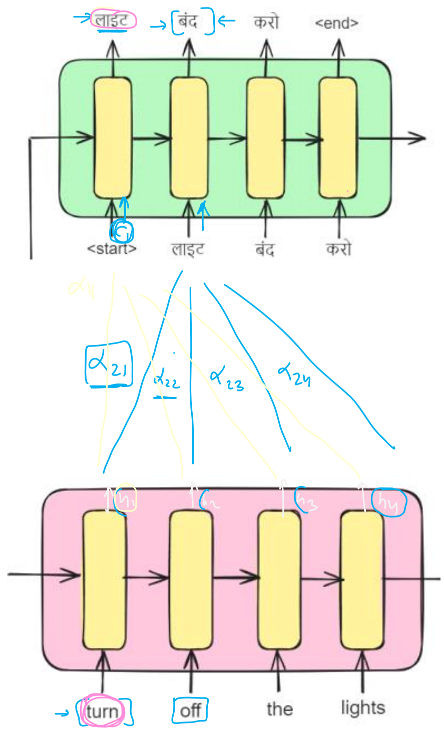
$h_1 h_2 \to c_2$

vector

vector [dim]

$\boxed{c_i} \to h_i$

4 4



decoder

α_{11}

α_{21}

4 dim

$c_i \to$ 4 dim

scalar

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

vectors

$c_3 =$

c_4

$\alpha_{11} = 0.6$

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

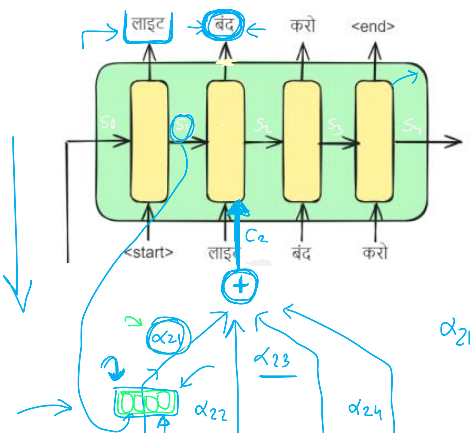
$i \times j = 16 \to$

encoder

j

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

$c_1 =$

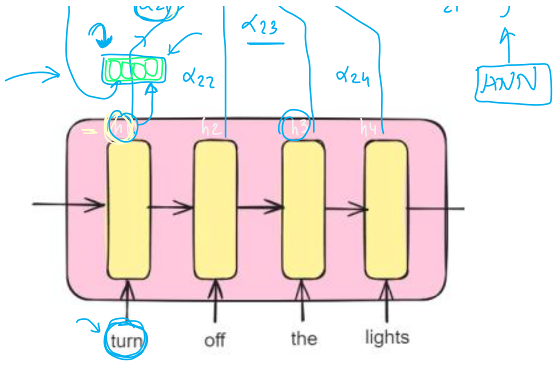


$\alpha_{21} = f(h_1, s_1)$

ANN

$\alpha_{21} \to f(h_1, s_1)$

$\alpha_{23} \to f(h_3, s_1)$



$\alpha_{21} \to f(h_1, s_1)$

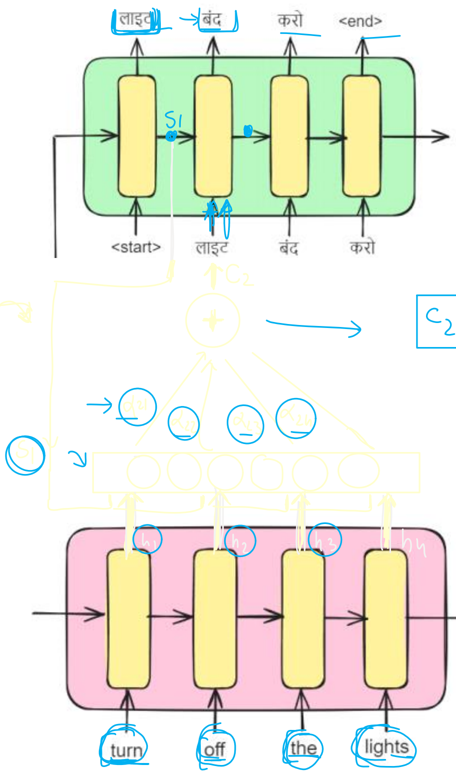
$\alpha_{ij} = f(h_j, s_{i-1})$

$\boxed{\alpha_{23} \to f(h_3, s_1)}$

ANN

universal function approx

what function?



→ 16

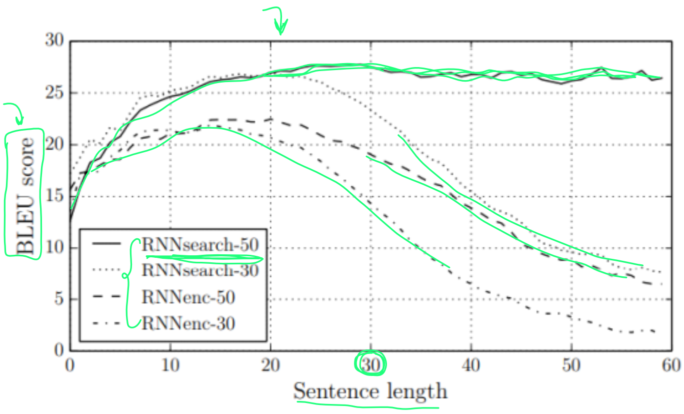
α_{11}

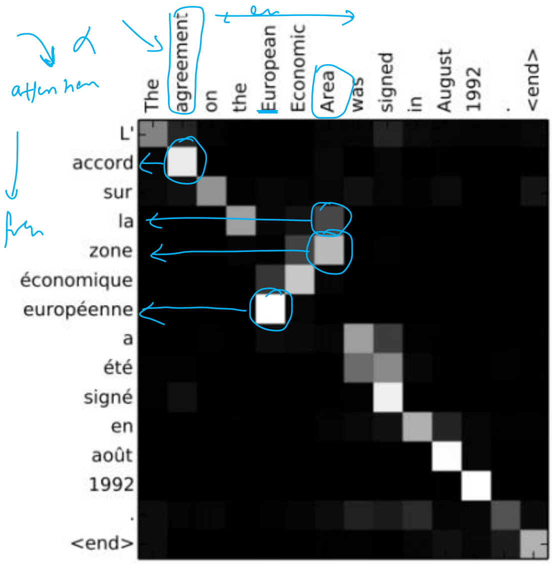
α_{12}

	turn	off	the	lights
लाइट	•	•	α_{13}	α_{14}
बंद	α_{21}	α_{22}	.	.
करो				
end				

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

eng-french

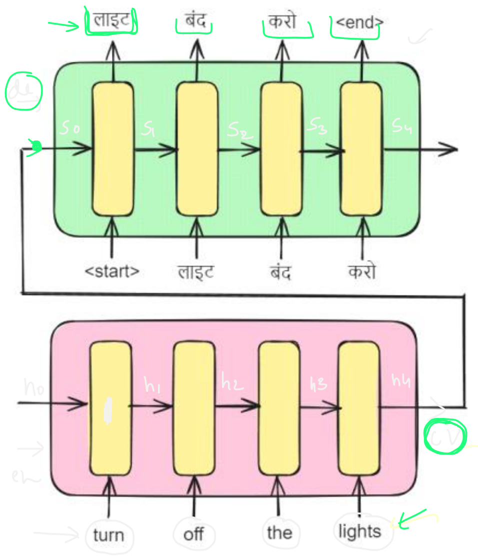




(a)

Recap

16 January 2024 16:10



turn off the lights → लाइट बंद करो

[NMT]

1) Encoder-Decoder

lstm

stacked lstm

set of nos

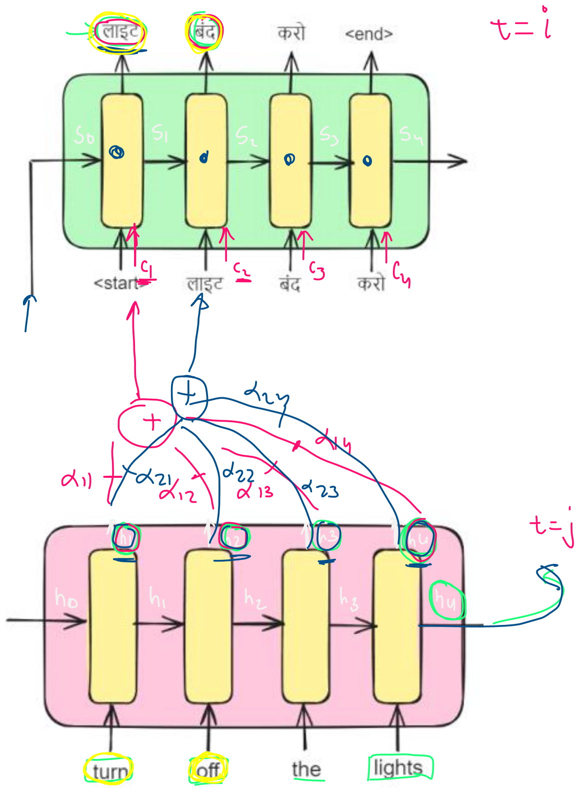
sentence >30 words

paragraph

document

translation

bottleneck → Attention mechanism



$t = i$

$c_1 \; c_2 \; c_3 \; c_4$

$t = j$

weighted sum

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

$4 \times 4 = 16$

α → alignment score

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

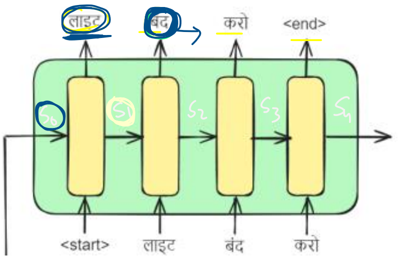
$$c_2 = \alpha_{21} h_1 + \alpha_{22} h_2 + \alpha_{23} h_3 + \alpha_{24} h_4$$

$\boxed{\alpha}$ → find out

- Bahdanau architecture attention
- Luong attention

Bahdanau Attention

16 January 2024 16:11



given

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

alignment score

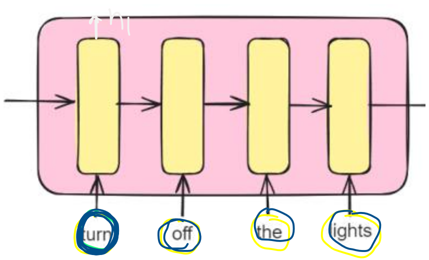
α

alignment

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

$\alpha_{12} \rightarrow$ लाइट $\rightarrow$ off

decoder → prev hidden state



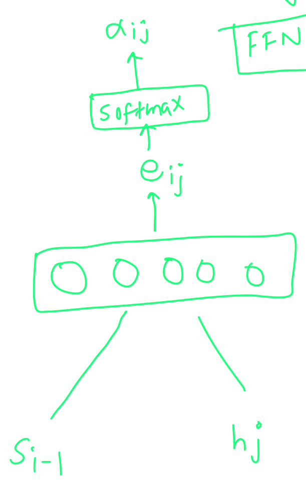
$\alpha_{11} = f(h_1, s_0)$ $\qquad \alpha_{21} = f(h_1, s_1)$

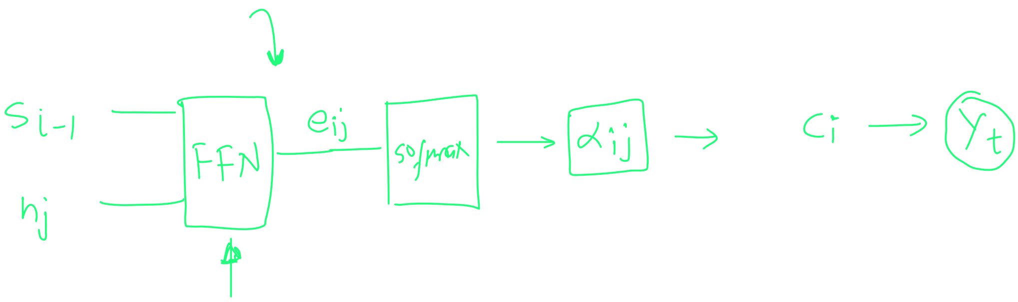
$\alpha_{ij} = f(h_j, s_{i-1})$

math func → the encoder hidden | decoder prev hidden state

approximate

FFN → ANN → UFA





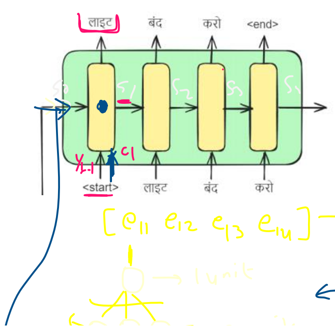
$s_0 \; y_{t-1} \; c_1 \rightarrow$ lstm $\rightarrow y_t$ (लाइट) $\boxed{s_1}$

$i = 1$ $\quad \boxed{t = 2}$

encoder sentence

$\hookrightarrow h_1 \; h_2 \; h_3 \; h_4$

$S_{i-1} \quad h_j$



$c_1 = \sum \alpha_{1j} h_j$

$[e_{11} \; e_{12} \; e_{13} \; e_{14}] \rightarrow$ softmax $\rightarrow [\alpha_{11} \; \alpha_{12} \; \alpha_{13} \; \alpha_{14}]$

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

$\leftarrow$ 3×1

3×1

8×3

batch operation $\underbrace{4\times 8 \quad 8\times 3}$ → $\boxed{4\times 3}$

tan(4×3)

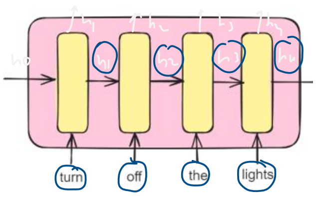
$\boxed{C_1} = \alpha_{11}h_1 + \alpha_{12}h_2 + \alpha_{13}h_3 + \alpha_{14}h_4$

$\alpha($

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

$h_1 \quad h_2 \quad h_3 \quad h_4$

$[s_{i-1} : h_j]$



vector

4 dim

$h_0 = [a\ b\ c\ d]$

4 rows / 8 cols

8cols

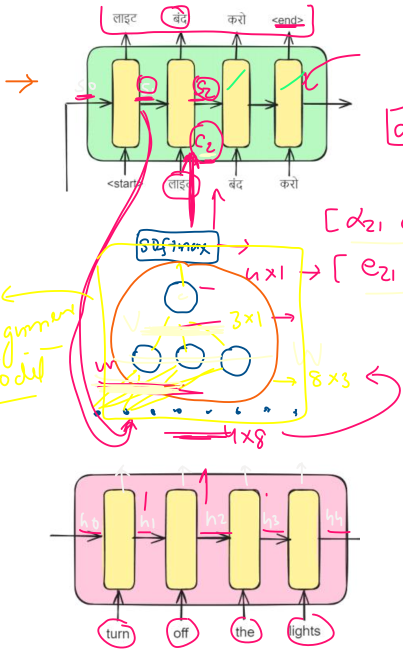
4rows

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

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

$$\left[\frac{e^{e_{11}}}{e^{e_{11}} + e^{e_{12}} + e^{e_{13}} + e^{e_{14}}}\right. \quad (4\times 1) \rightarrow 4 \text{ numers}$$

true label



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

$\boxed{\alpha}$

$[\alpha_{21}\ \alpha_{22}\ \alpha_{23}\ \alpha_{24}]$

4×1 → $[e_{21}\ e_{22}\ e_{23}\ e_{24}]$

4×3

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} & h_{11} & h_{12} & h_{13} & h_{14} \\ s_{11} & s_{12} & s_{13} & s_{14} & h_{21} & h_{22} & h_{23} & h_{24} \\ s_{11} & s_{12} & s_{13} & s_{14} & h_{31} & h_{32} & h_{33} & h_{34} \\ s_{11} & s_{12} & s_{13} & s_{14} & h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix}$$

4×8

time distributed FNN

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

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1} \exp(e_{ik})}$$

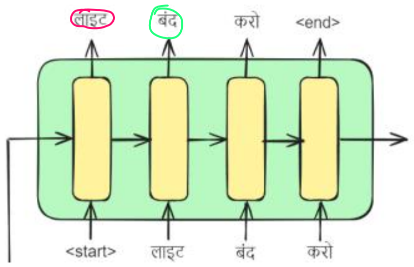
$$e_{ij} = V^T[\tanh(W[s_{i-1} : h_j] + b)]$$

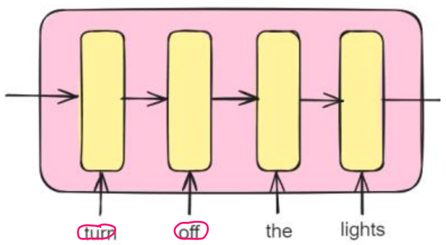
additive attention

Luong Attention

17 January 2024 00:09

Luong





parameters → slow

$c_i = \sum \alpha_{ij} h_j$ → FFN $\alpha\{[V \tanh(W[s_{i-1} : h_j] + b)]\}$

$\alpha_{ij} = f(s_{i-1}, h_j)$ ×

$\alpha_{ij} = f(s_i, h_j)$ → $[s_i^T \cdot h_j]$ → dot product

fast

current

① diff

updated info

dynam

adjust

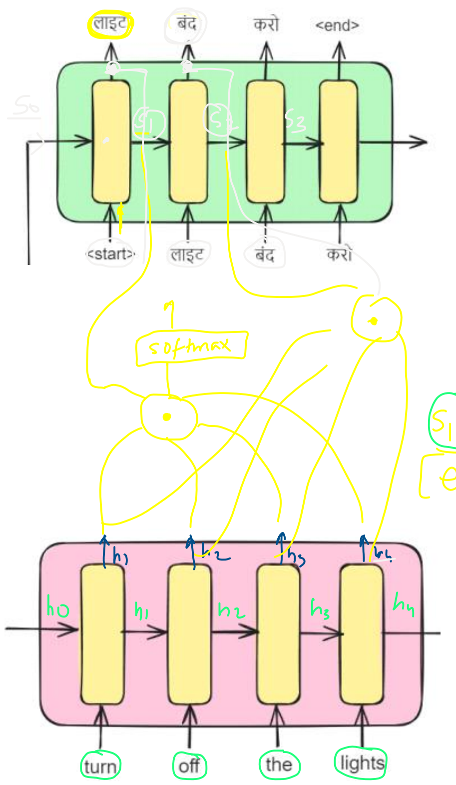
$s_i = [a\ b\ c\ d]$

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

$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} [e\ f\ g\ h]$

$[ae + bf + cg + dh]$ ← scalar → attention

α_{ij} ← softmax ← $\boxed{e_{ij}}$



output

$s_1 : c_1$ → $\tilde{s}_1$ → softm

$s_2 : c_2 \to \tilde{s}_2 \to$

multiplicative atten

$[e_{21}\ e_{22}\ e_{23}\ e_{24}]$ softm → $\alpha_{21}\ \alpha_{22}\ \alpha_{23}\ \alpha_{24}$ → c_2

$s_1 h_1$ $s_1 h_2$ $s_1 h_3$ $s_1 h_4$

e_{11} e_{12} e_{13} e_{14}

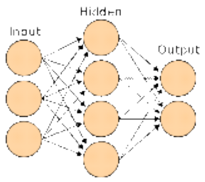
α_{11} α_{12} α_{13} α_{14} → $\boxed{c_1}$

$\sum \alpha_{ij} h_j$

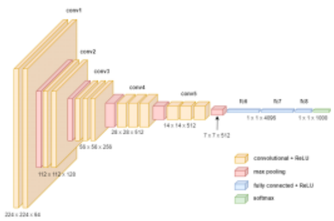
What is Transformer?

27 January 2024 18:41

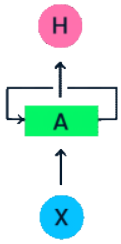
→ self attention scalable



ANN
tabular



CNN
Image



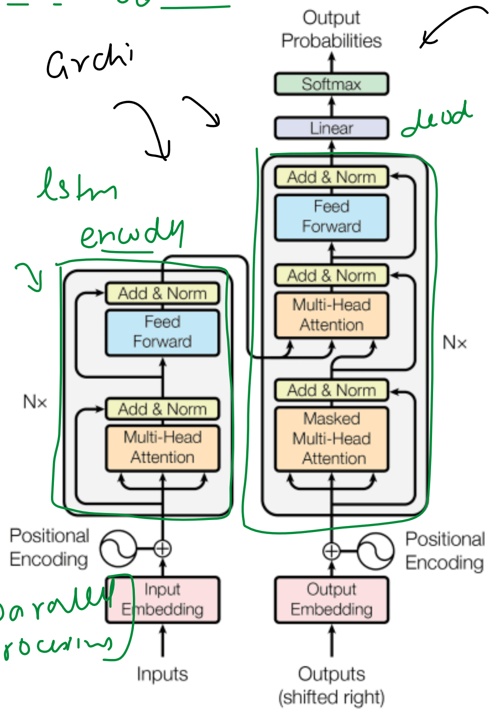
RNN
sequence
↳ text

Archi

lstm

encoder

decod



[parallel processing]

→ [Transformer] → seq2seq task
NN arch
- machine translation
- question ans
- text summary

[Attention Is All You Need]

2017

↘ Google Brain

Deep learning

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
Łukasz 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.

Impact of Transformers

27 January 2024 20:03

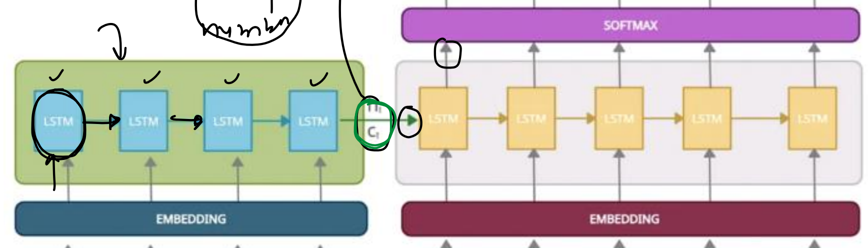
Impact of Transformers

- Revolution in NLP
  - NLP → heuristic
  - 50 years → stat ML
    - → embeddings
    - LSTMs
  - ↓
  - NLP → SOTA
- Democratizing AI
  - Big devs → BERT, GPT
  - → pretrain → task SOTA
  - → specific fine tune
- Multimodal capability
- Acceleration of GenAI
  - GANs →
  - ↓ textgen
- Unification of deep learning paradigm
  - NLP seq2seq
  - GenAI →
  - Computer vision
  - Reinforcement science

The Origin Story!

27 January 2024 22:38

2014-15 → seq2seq machine



context vector (summary)

set of number

करो नाम है <end>

My name is Nitish

enc dec

long small input X

fail

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

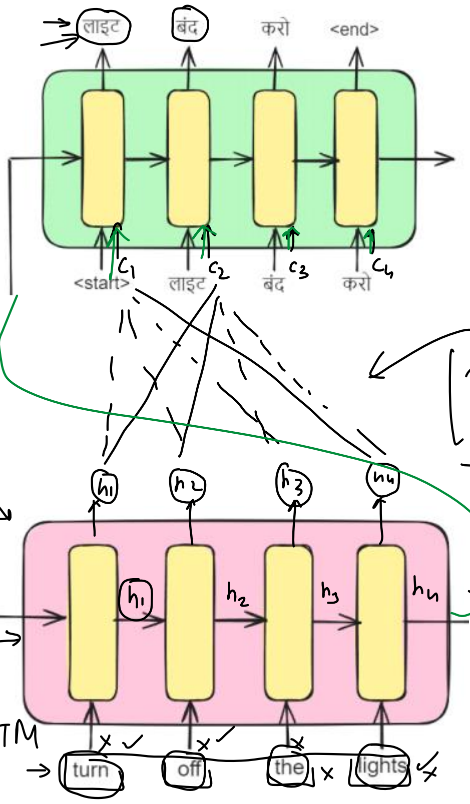
Abstract

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 multilayered 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 on 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 → attention ↳ concept
KyungHyun Cho, Yoshua Bengio*, Université de Montréal

translation sent > 30 words



c_1 c_2 c_3 c_4 h_1 h_2 h_3 h_4

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

attention weights

LSTM

scratch model → time and effort dataset ↓ money

pretrain fine tuning

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 training → slow → huge dataset

Transfer learning → CNN → fine tune, dog vs cat class → imageNET →

research → incremental time
↳ research → travel

→ lstm/self → parallely
→ stable
→ hypertune

NLP → BERT, GPT → fine tune

Attention Is All You Need → 2017

Output Probabilities

↳ research → travel

→ stable

→ hypertune

stable

robust

NLP → GPT / → fine-tune

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

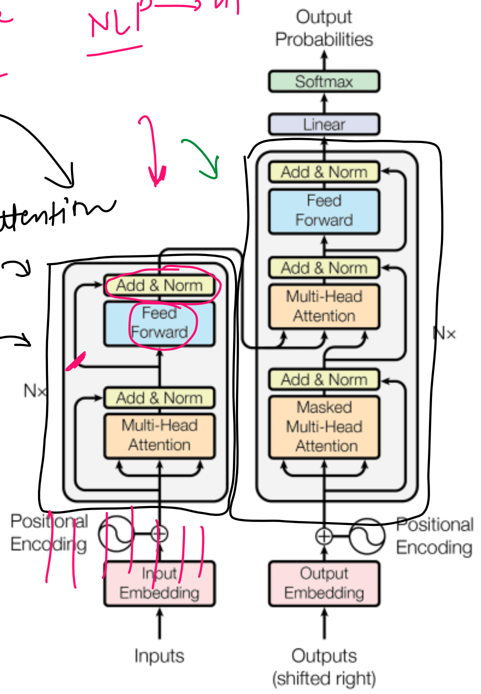
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

self attention

× LSTM/RNN



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 Transfor / Alphafold-2

2021 → Gen AI

2022 → ChatGPT / Stable Diffusion

Advantages

28 January 2024 01:00

- Scalability
- Transfer learning
- multimodal ip/op
- flexible arch
  - encoder only
    - BERT
  - decoder only
    - GPT
- ecosystem
  - hugging face
  - videos/blogs
- Integrated AI techni
  - GANs + Transformer image gener → DALLE
  - RL + Transformer
  - Image + visual
    - CNN + Transfor
      - ↳ vision tran

Famous Applications

28 January 2024 01:17

Disadvantages

28 January 2024 01:51

High Computation (GPU)

Data (LLMs) ↓ text ↓ unsupervi ↳ labels

Overfitting

Energy consump

Interpret

Bias → Ethical

Future

28 January 2024 02:09

Improvements in efficiency
- pruning
- Quantization
- knowledge distillation

Multimodal capabilities
- text
- image/speech
- sensory data
- Biometric feedback
- time series

Responsible
→ Bias
→ Ethical concerns

Domain specific
- DoctorGPT
- legalGPT
- TeacherGPT

Multi lingual
- english
- hindi

Interpretability
- critical

The What

Self attention   computer x

NLP → words → numbers → vectorization

OHE

1) mat cat mat
2) cat rat rat

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

num → [1 0 0] [0 1 0] [1 0 0]

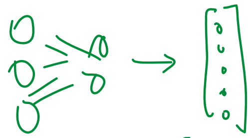
BoW

	mat	rat	cat
S1	[2	0	1]
S2	[0	2	1]

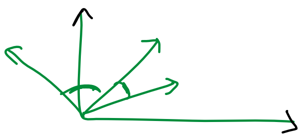
Tf Idf → word embedding

number → semantic meaning

training wiki →



→ 5 dim



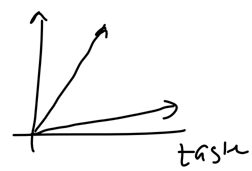
cricket [0.2 0.9 0.9 0.90 1] n-dim vector 256, 512, 64

king → [0.9 0.1 1 0 0.9]
queen → [0.9 0.2 0.4 1 0]
royalty

similar   → similar

Apple → vector

meaning → vector



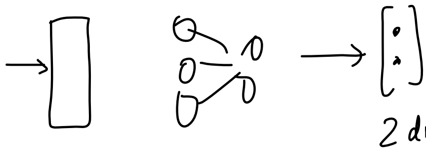
The problem of "Average Meaning"

→ dataset

1) An apple a day keeps the doctor away
2) Apple is healthy
3) Apple is better than orange
4) Apple makes great phones

10000: 9000 fruit, 1000 phone

9000 tech, 1000 fru



2 dim

[x y]
taste   technology

[0.8 0.2] → [0.9 0.3]

word embeddings
create → use → static

smart

static embedding → [contextual embedding]

LLM - trans eng-hindi

NLP → trans eng-hindi

fruit → [Apple launched a new Phone while I was eating an orange]

static → [0.9 0.3]

[0.3 0.8]

word embeddin [static] → self attention ↓ mechanism

[Calender]? How?

e_{apple}

e_{launch}

e_{phone}

e_{orange}

self attention

y_{apple}

y_{lam}

y_{oran}

→ transformers

Embeddings

Hello → [0.3 | 0.9 | 1 | 0] → OHE / BOW / Embeddings

↑ semantic

Money Bank

→ static →

River Bank

0.6	0.2	0.1	0.7

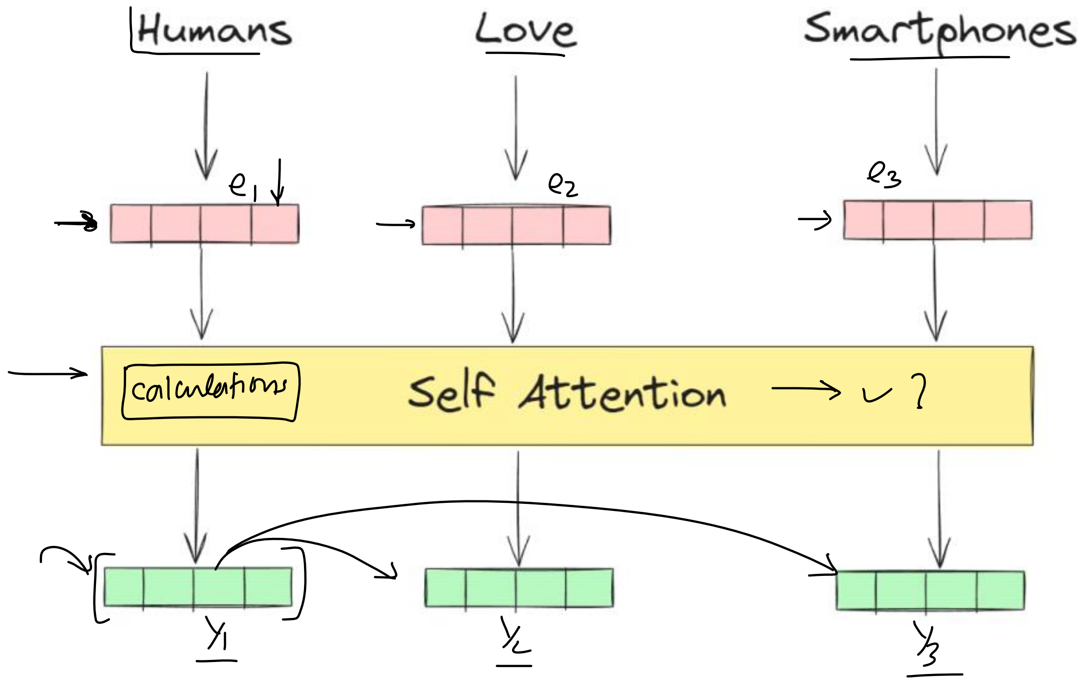
0.6	0.2	0.1	0.7

Contextual embedding

↳ dynamic

Self Attention

embeddings → dynamic contextual


Humans
Love
Smartphones
e_1
e_2
e_3
calculations
Self Attention
?
y_1
y_2
y_3

First Principle Approach

04 February 2024 17:55

money bank grows

bank → bank

bank → 0.3 money + 0.7 bank + 0.1 grows

river bank flows

bank = 0.5 river + 0.4 bank + 0.1 flows

S1 word embe

money = 0.7 money + 0.2 bank + 0.1 grows

bank = 0.25 money + 0.7 bank + 0.05 grows

grows = 0.1 money + 0.2 bank + 0.7 grows

S2

river = 0.8 river + 0.15 bank + 0.05 flows

bank = 0.2 river + 0.78 bank + 0.02 flows

flows = 0.4 river + 0.01 bank + 0.59 flows

n dim

n dim →

$$e^{(new)}_{money} = 0.7\, e_{money} + 0.2\, e_{bank} + 0.1\, e_{grows}$$

$$e^{(new)}_{bank} = 0.25\, e_{money} + 0.7\, e_{bank} + 0.05\, e_{grows}$$

$$e^{(new)}_{grows} = 0.1\, e_{money} + 0.2\, e_{bank} + 0.7\, e_{grows}$$

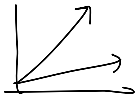
similarity

e_{money} e_{money}

e_{money} e_{bank}

e_{money} e_{grow}

e_{grows} e_{money}



dot products

1

normalized

$$e^{(new)}_{bank} = [e_{bank} \cdot e^T_{money}]\, e_{money} + [e_{bank} \cdot e^T_{bank}]\, e_{bank} + [e_{bank} \cdot e^T_{grows}]\, e_{grows}$$

e_{bank} S_{21} e_{money}

e_{bank} S_{22} e_{bank}

e_{bank} S_{23} e_{grows}

S_{21} S_{22} S_{23}

Softmax

$$W_{21} = \frac{e^{S_{21}}}{e^{S_{21}} + e^{S_{22}} + e^{S_{23}}}$$

$$W_{22} = \frac{e^{S_{22}}}{e^{S_{21}} + e^{S_{22}} + e^{S_{23}}}$$

$$[W_{21}] \times e_{money} + [W_{22}] \times e_{bank} + [W_{23}] \times e_{grows}$$

$\hat{e}_{money}$ + e_{bank} + e_{grows}

$e^{(new)}_{bank} = \boxed{y_{bank}}$ → contextual word embeddings

Points to consider

→ This operation is a parallel operation

→ There are no parameters involved

self model

hello → e_1 → Y_1

how are you

↓ ↓ ↓

e_1 e_2 e_3

$\boxed{\text{Self attention}}$

↓ ↓ ↓

Y_1 Y_2 Y_3

[general contextual embedding]

machine

eng	hindi
$\boxed{\text{how are you}}$	कैसे हो
I am good	मैं बढ़िया हूँ
piece of cake →	बहुत आसान काम
	— हो जाएगा

X task specific contextual embedding

$\boxed{\text{piece of cake}}$ → [केक का टुकड़ा]

↓ ↓ ↓

e_1 e_2 e_3

↓ ↓ ↓

Y_1 Y_2 Y_3

break a leg → टांग तोड़ दो

Progress

06 February 2024 00:42

words → machine

embedding → semantic

→ Context

river bank money bank

→ contextual embeddings

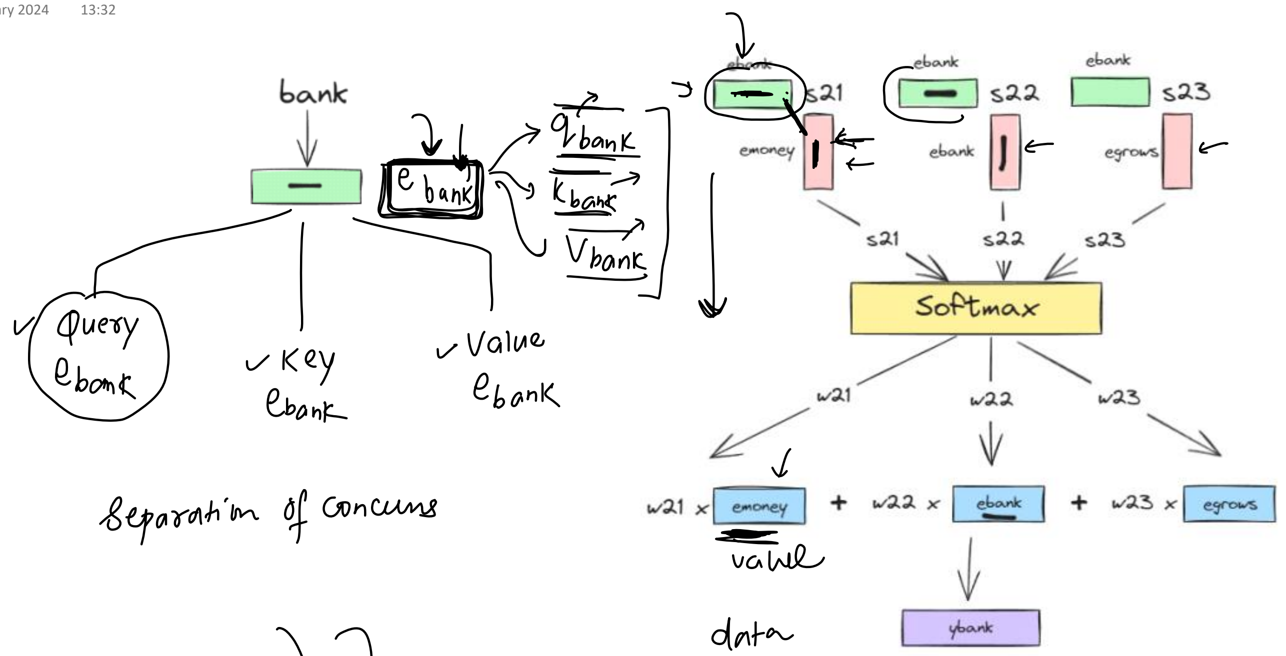
simple self model → parameters

learnable parameter → general

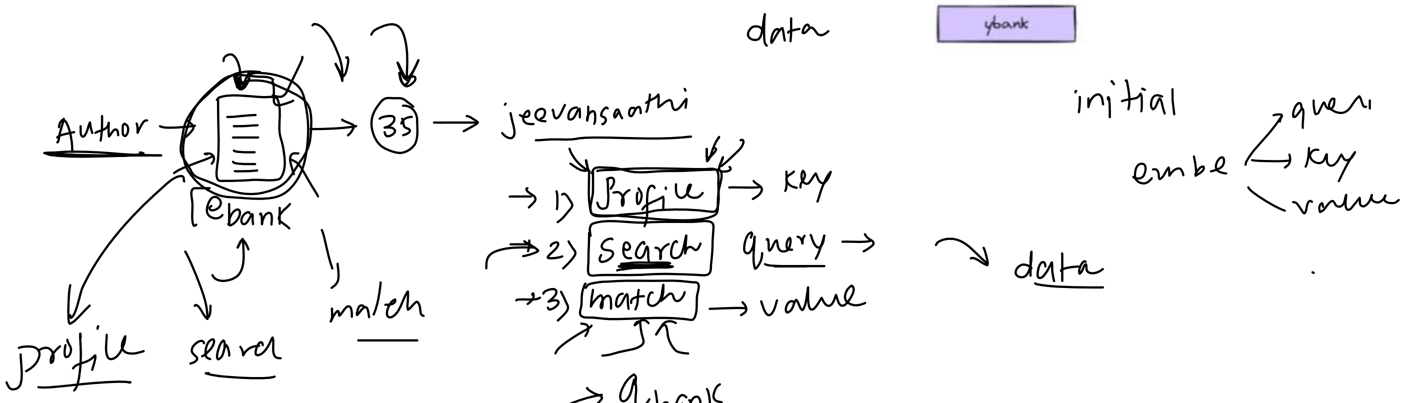
task-specific

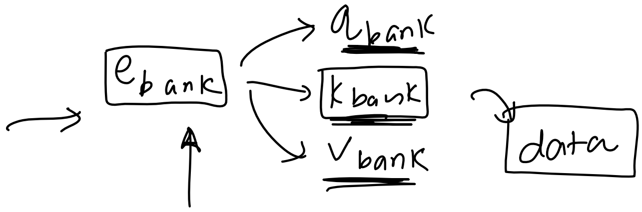
Query, Key & Value Vectors

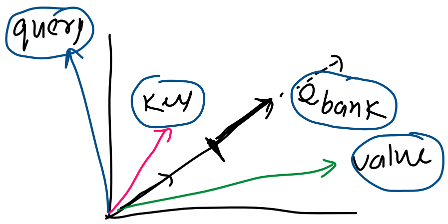
06 February 2024 13:32



Separation of concerns

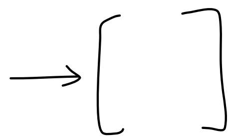


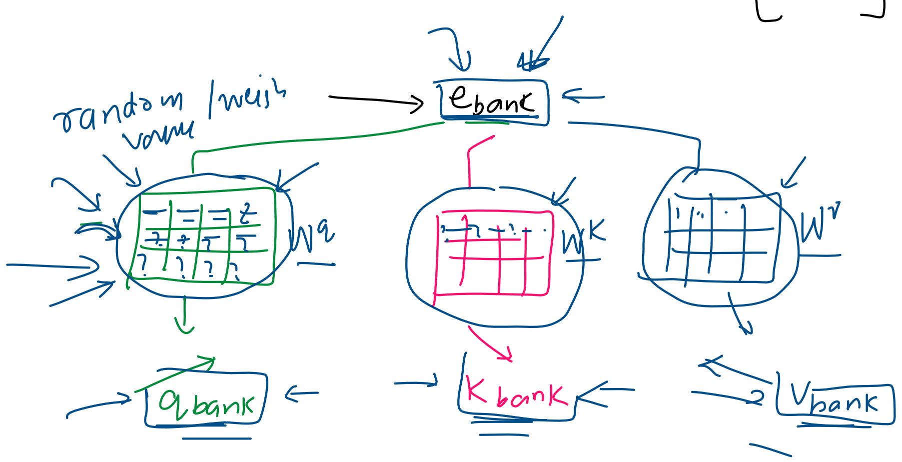




d magnitude (scaling)

→ linear transform





training

data → machine transl

→ translation

$3 \times 3 \rightarrow$ dot $\rightarrow$ 9 values

Revision

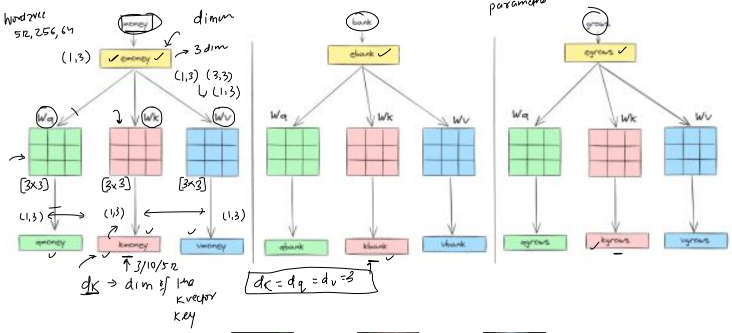
28 February 2024 16:25

money bank grows

embeddings

Wq WK Wv

parameter



3 3 → 9 dot → small → high

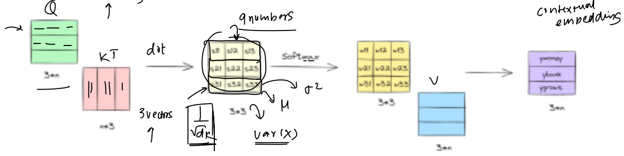
(1,3)

9 vector-vector dot pro

(3 vector)

(1,3)

512 dim



$$\boxed{\text{Attention}(Q,K,V) = \text{Softmax}\left(QK^T\right)V} \rightarrow \text{summary}$$

$\sqrt{3}$ $\sqrt{d_k}$

scaled dot-product

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

$\frac{QK^T}{\sqrt{d_k}}$ → unstable gradient

→ $\sqrt{d_k}$

$[Q K^T]$

matrix matrix

dot product

vectors dot product

Softmax$\left(\frac{QK^T}{\sqrt{3}}\right)$V

$\frac{1}{\sqrt{d_k}}$

why?

→ Dot-product ka Nature

$\left[\frac{1}{\sqrt{d_k}}\right]$ scale?

1) [1, 2, 3] — [3, 2, 1] f
2) [ ] - [ ] g
3) [ ] — [ ] h
4) [ ] — [ ] i
5) [ ] — [ ] j

→ σ_2^2

1) [1, 2] — [3, 2] a
2) [ ] — [ ] b
3) [ ] — [ ] c
4) [ ] — [ ] d
5) [ ] — [ ] e

→ σ_1^2

low dimension → dot product → low variance

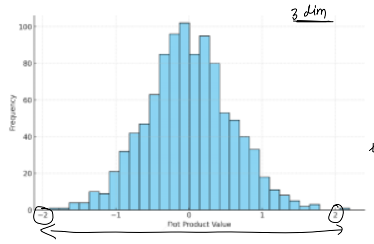
high dim → dot product → high variance

code

3 dim

1000 pair of vectors

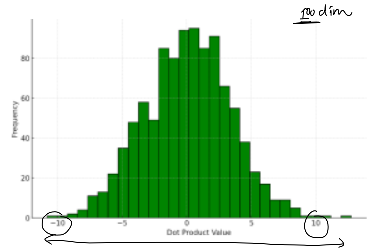
1) — × — →
2) — × — →
⋮
1000 — —



100 dim

1000 pairs of vector

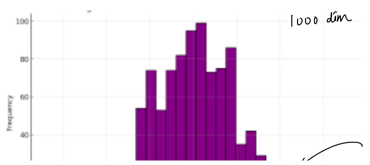
1) — × —
2) — × —
⋮
1000) — × —

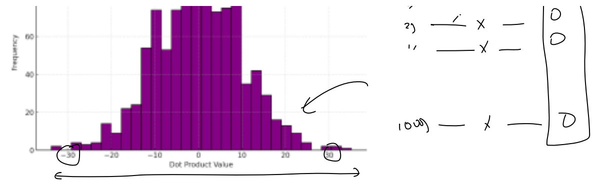


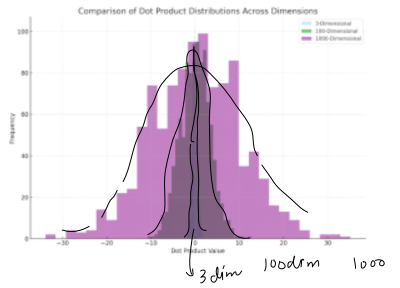
1000 dim

1000 pair of vectors

1) — × —
2) — × —
— × —



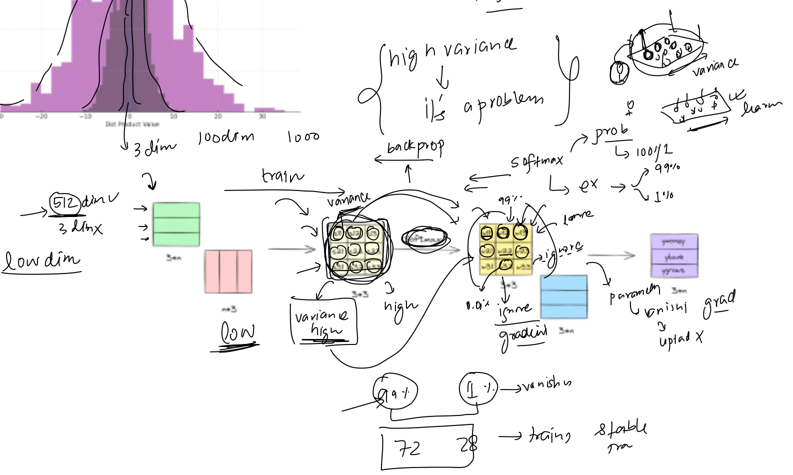




vector

dim → variance
↳ high

high variance
↓
it's a problem



each c

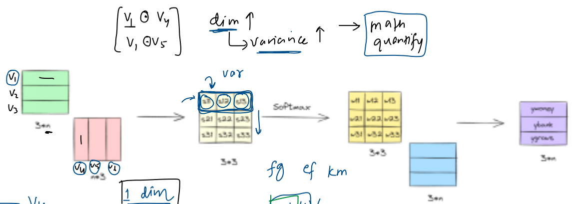
$X \to Var(x)$

$Y \to cX$

$\hookrightarrow c^2 Var(x)$

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

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



1 dim

$[a]$

3

v_1 — v_4, v_5, v_6

2 dim

$[b]$ 2, $[c]$ 1, $[d]$ 4

ab 6, ac 3, ad 12

→ random var → X

→ variance (sample) ↓ expected var

→ $\boxed{Var(x)}$

$X \to var(x)$

$cX \to c^2 Var(x)$

$Y \to Var(Y) \to 2Var(x)$

$\frac{Y}{\sqrt{2}} \to \frac{1}{2}Var(Y) \to \frac{1}{2}\, 2\, Var(x) \to Var(x)$

$[a \quad b]$

$[c \quad d] \to \frac{ac + bd}{\sqrt{2}}$

$[e \quad f] \to \frac{ae + bf}{\sqrt{2}}$ 18

$[g \quad h] \to \frac{ag + bh}{\sqrt{2}}$ 2

→ 2 Var(x)

$Y \to Var(Y)$

$Var(Y) \simeq 2Var(x)$

$Var(Y) > Var(X)$

Yes No

Var(X)

2 dim

$\boxed{\frac{1}{\sqrt{d_K}}}$

$Var(z) > Var(y) > Var(x)$

3 dim

$[a \quad b \quad c]$

$[d \quad e \quad f] \to \frac{ad + be + cf}{\sqrt{3}}$

$[g \quad h \quad i] \to \frac{ag + bh + ci}{\sqrt{3}}$

$[k \quad l \quad m] \to \frac{ak + bl + cm}{\sqrt{3}}$

→ Z

$Var(z) \simeq 3Var(x)$

Var(x)

$z \to var(z) \simeq 3 Var(x)$

$\frac{1}{\sqrt{3}} z \to \left(\frac{1}{\sqrt{3}}\right)^2 Var(z) = \frac{1}{3}\, 3 Var$

d dim → d Var(x)

↳ variance dim = 1

dim → variance linear

1 dim → Var(x) → Var(x)

2 dim → 2 Var(x) → Var(x)

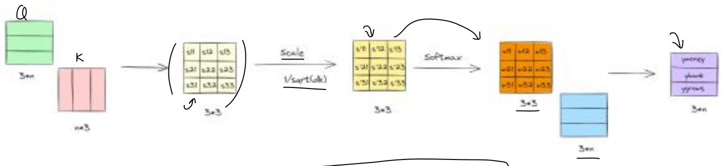
3 dim → 3 Var(x) → Var(x)

:

d dim → d Var(x) → Var(x)

$[\_ \; - \; - \; - \; - \; - \;] \to Var(X) \to$

d dim



$$\boxed{\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V} \longrightarrow \text{detailed}$$

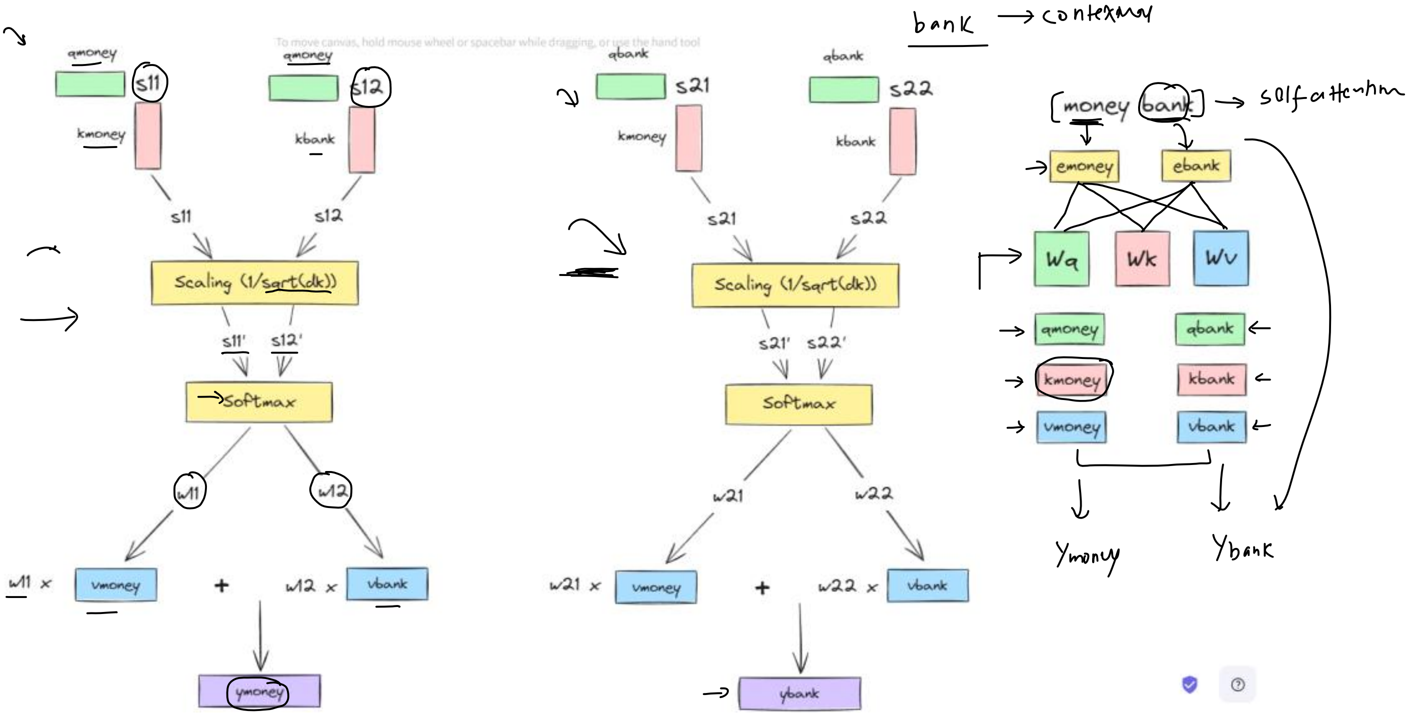
Transformers

What is d_k

28 February 2024 16:59

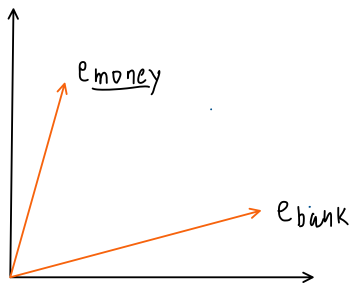
Recep

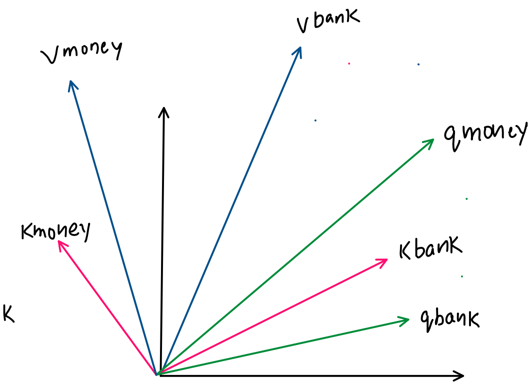
08 March 2024 15:14

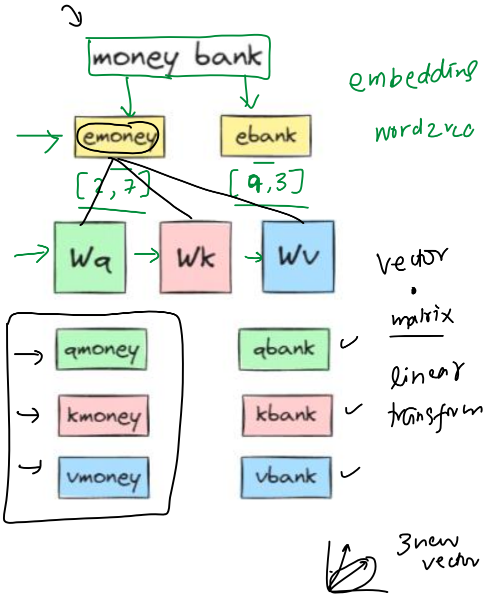


Geometric Intuition

08 March 2024 15:16

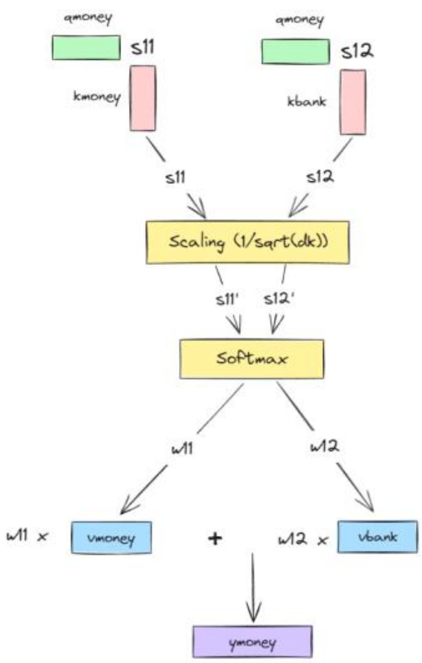


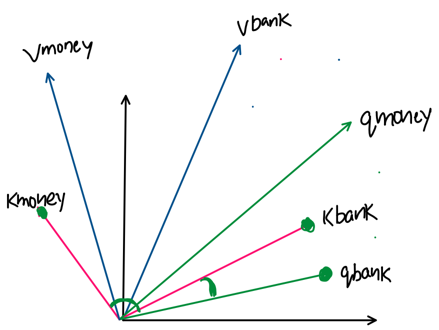




$$W_q = \underbrace{\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}}_{2\times 2} \qquad W_k = \underbrace{\begin{bmatrix} 3 & 4 \\ 5 & 1 \end{bmatrix}}_{2\times 2} \qquad W_v = \underbrace{\begin{bmatrix} 4 & 1 \\ 2 & 1 \end{bmatrix}}_{2\times 2}$$

* All values are hypothetical





Dot Product

$S_{21} = \boxed{10}$ $\qquad S_{22} = \boxed{32}$

Scaling

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

Softmax

$\boxed{1}$

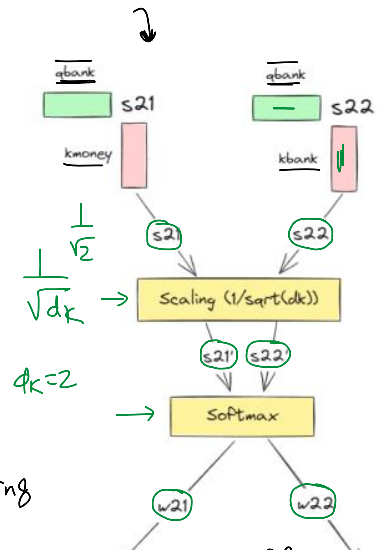
$W_{21} = \boxed{0.2} \qquad W_{22} = \boxed{0.8}$

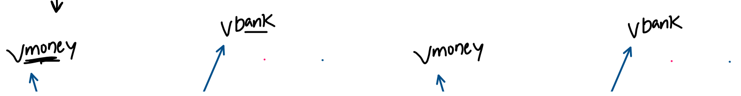
$\frac{1}{\sqrt{2}}$

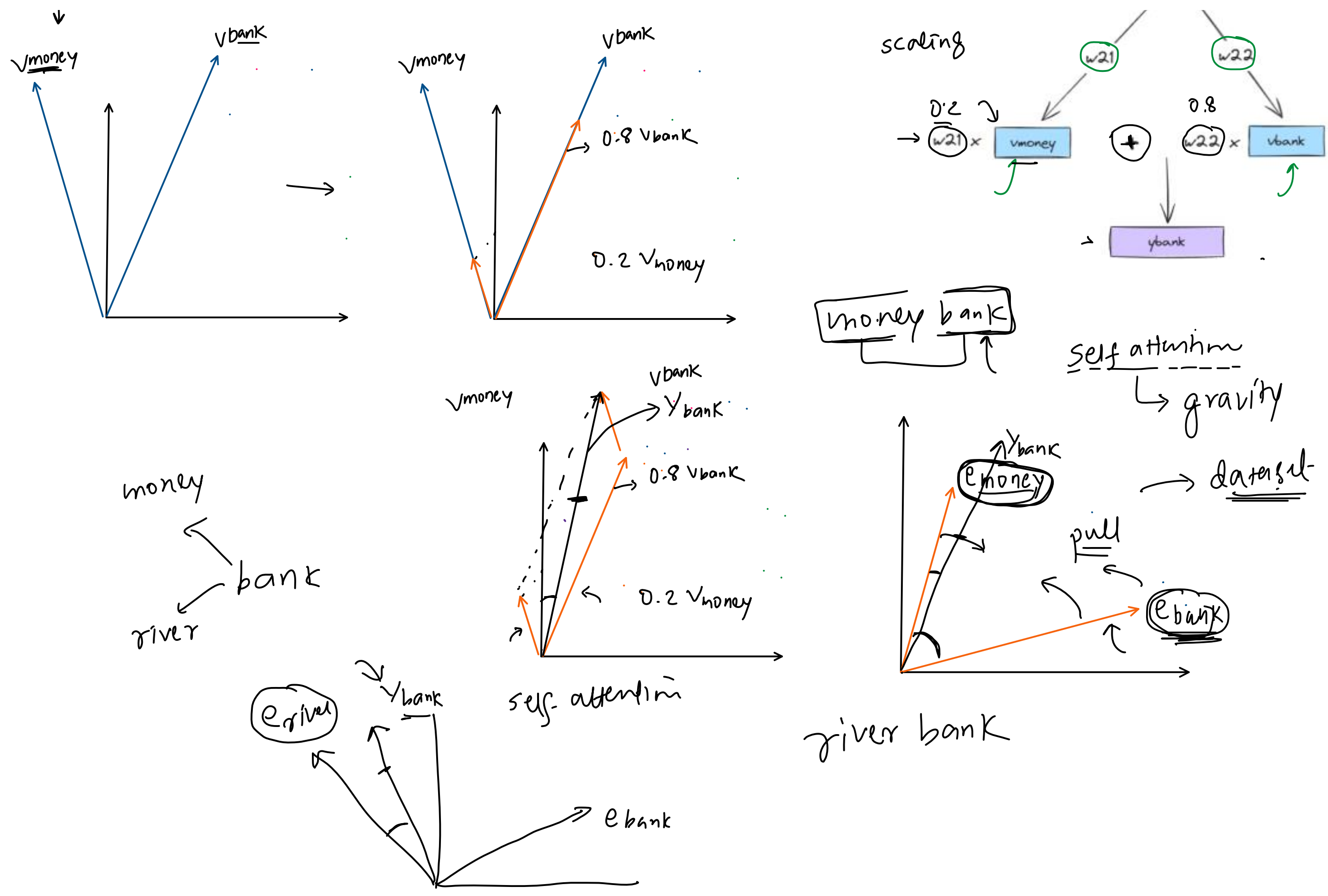
$\frac{1}{\sqrt{d_k}}$

$d_k = 2$

scaling






Vmoney
Vbank
Vmoney
Vbank
0.8 Vbank
0.2 Vmoney
scaling
w21
w22
0.2
0.8
w21 ×
vmoney
+
w22 ×
vbank
ybank
money bank
self attention
gravity
money
bank
river
Vmoney
Vbank
Ybank
0.8 Vbank
0.2 Vmoney
Ybank
emoney
dataset
pull
ebank
self attention
river bank
eriver
Ybank
ebank

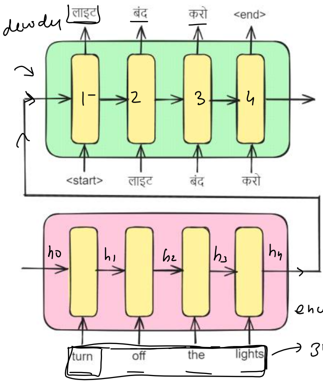
Recap of Attention

11 March 2024 17:09

long trans → NN

↓

encoder-decoder



30 words

[seq of numbers]

context vector

encoder

→ 30 words

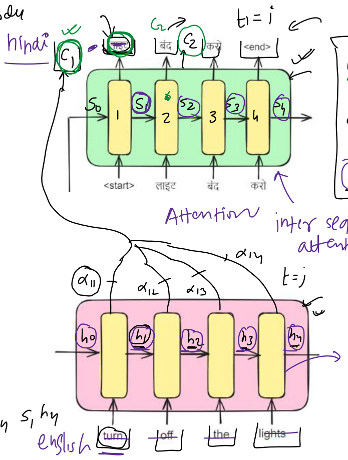
e_{11} s_1h_1

e_{12} s_1h_2

e_{13} s_1h_3

e_{14} s_1h_4

Attention



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

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

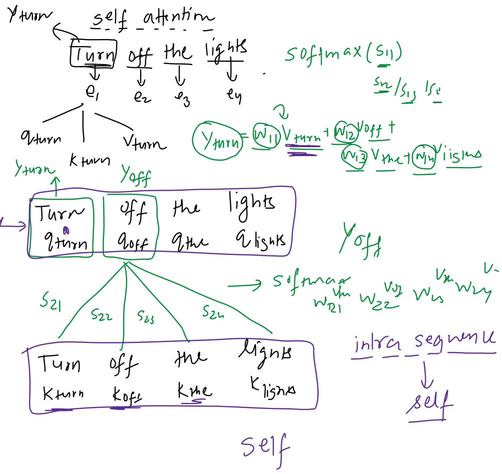
$e_{ij} = s_i^T h_j$

s_i → query

h_j → key

h_j → value

self attention



intra sequence

↓

self

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

→ weights

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

→ alignment score

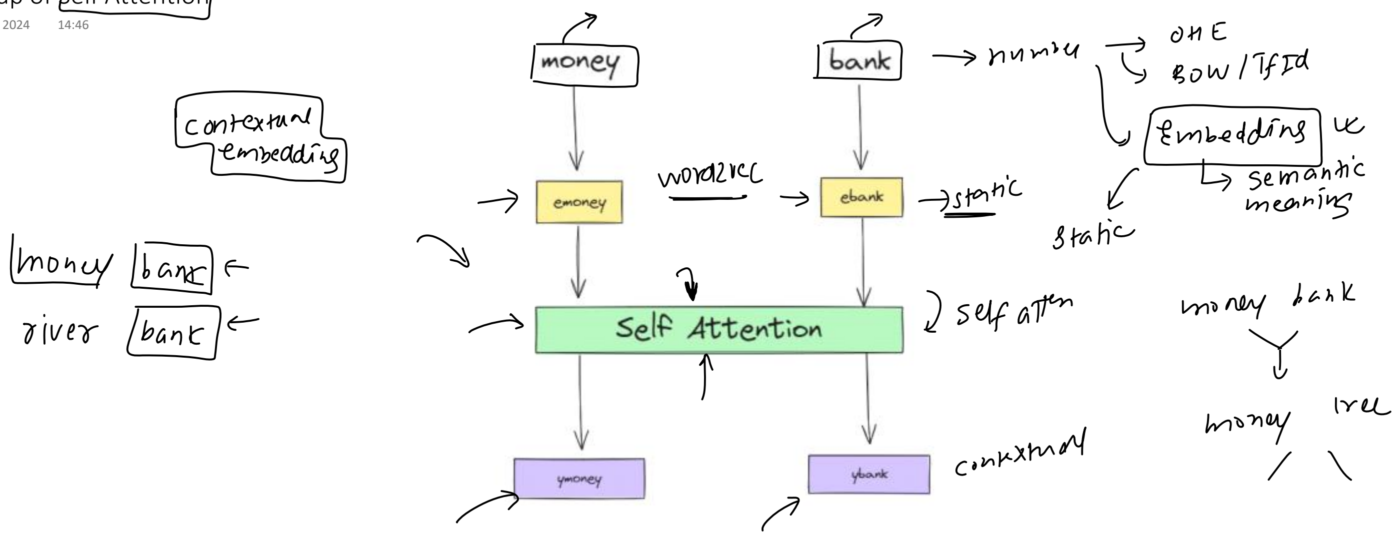
$$\boxed{e_{ij} = s_i^T h_j}$$

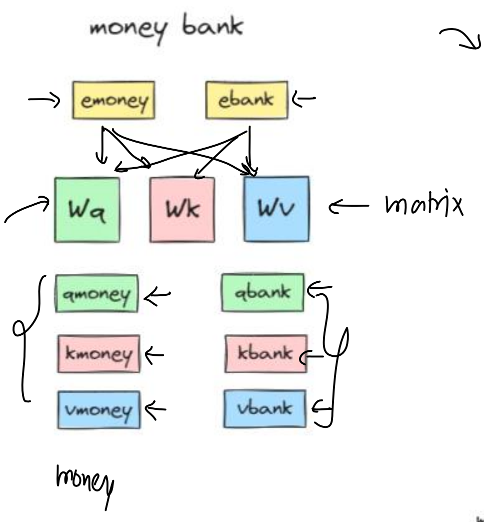
→ Luong attention

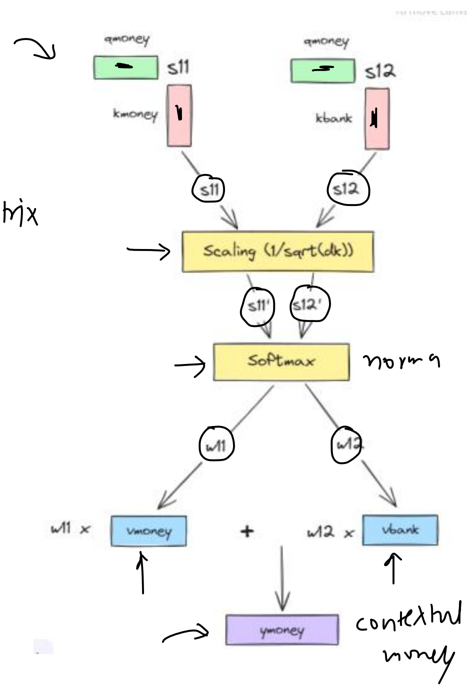
self attention

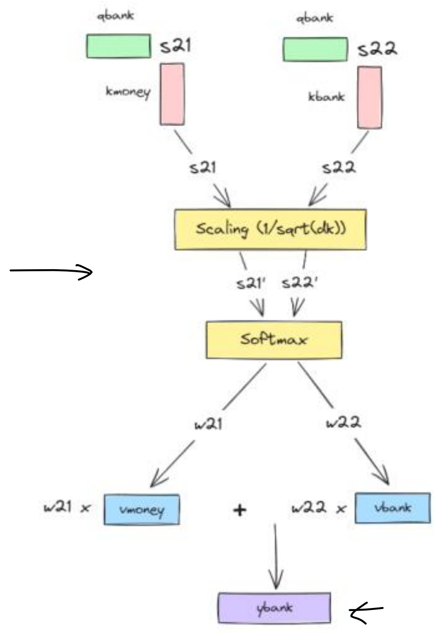
Recap of $\boxed{\text{Self Attention}}$

08 April 2024 14:46









Problem with Self Attention

08 April 2024 14:47

Multihead attention

The man saw the astronomer with a telescope

table

→ self
↓
single perspective

	The	man	saw	the	astronomer	with	a	telescope
The	0							
man								
saw								
the								
astronomer								
with								
a								
telescope								

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.

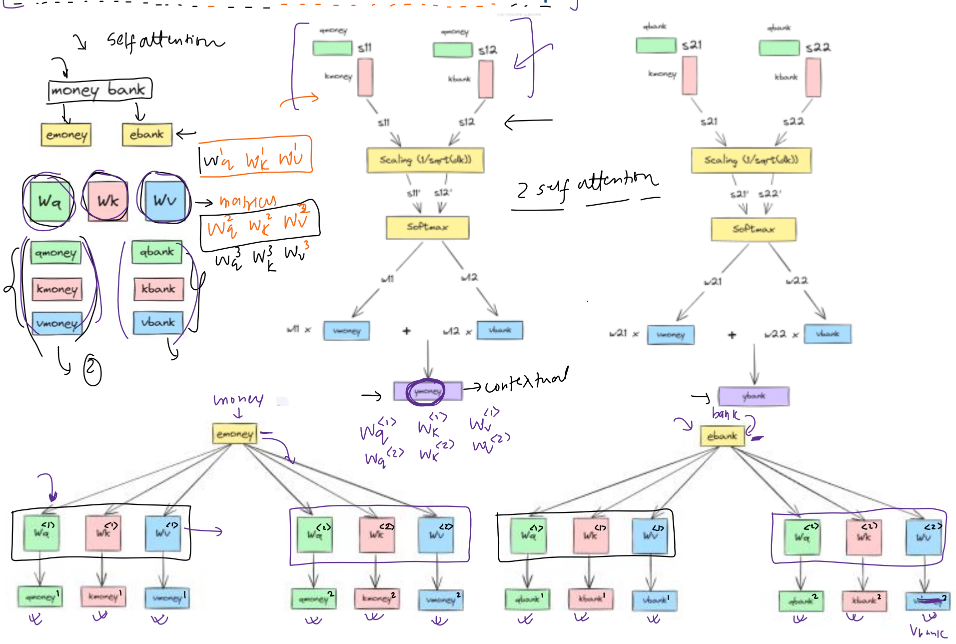
India's AI future holds promise as it harnesses a burgeoning talent pool and government initiatives to pioneer AI-driven innovation. With a focus on sectors like healthcare and education, India aims to leverage AI for societal development. The emphasis on ethical AI and data security underscores India's 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.

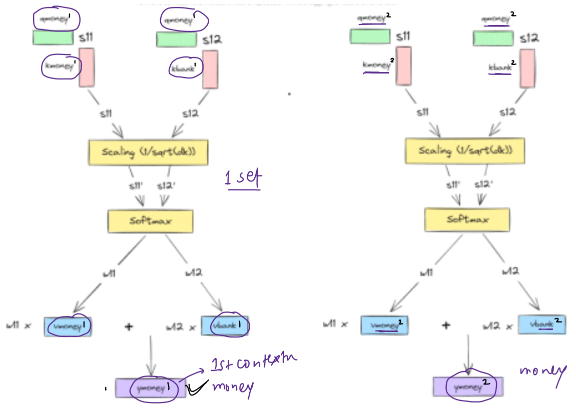
Multi-head Attention

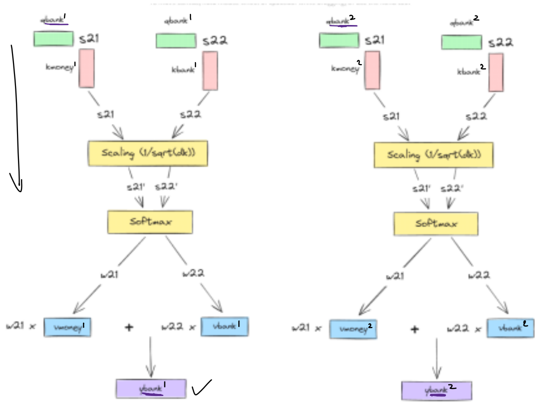
10 April 2024 09:50

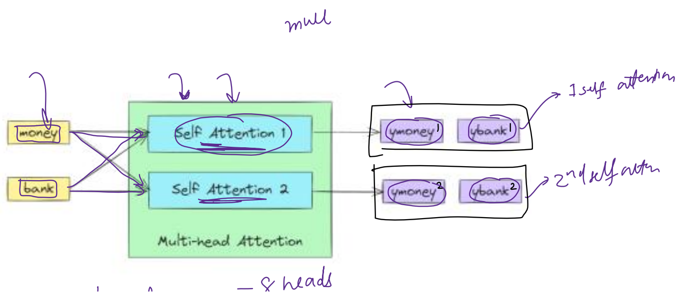
The man saw the astronomer with a telescope

self attention → 2

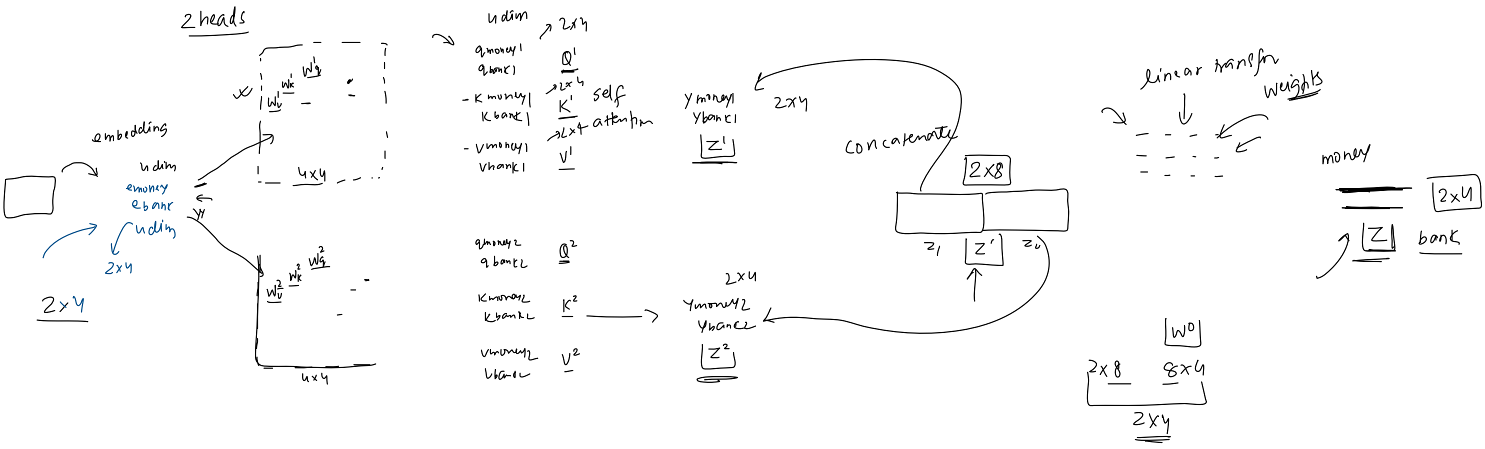


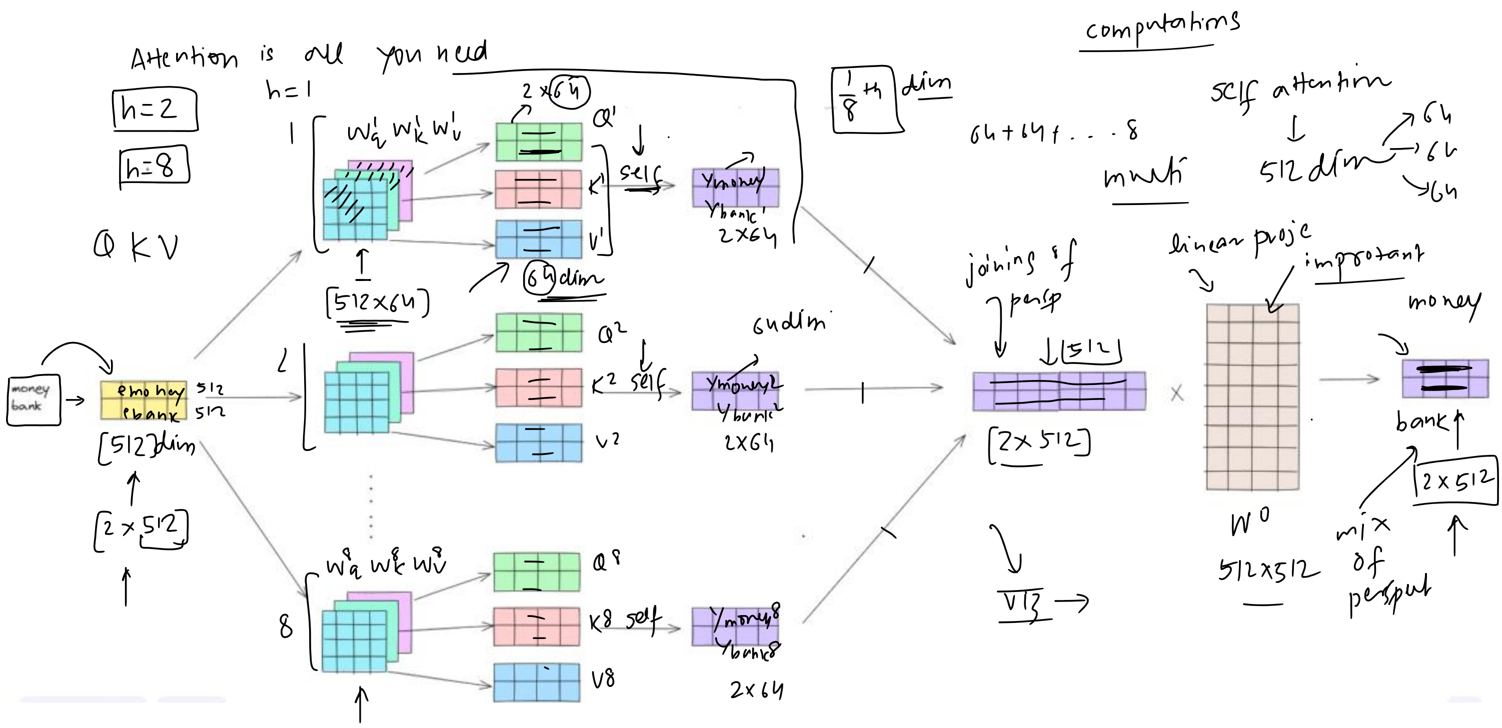






transformer – 8 heads

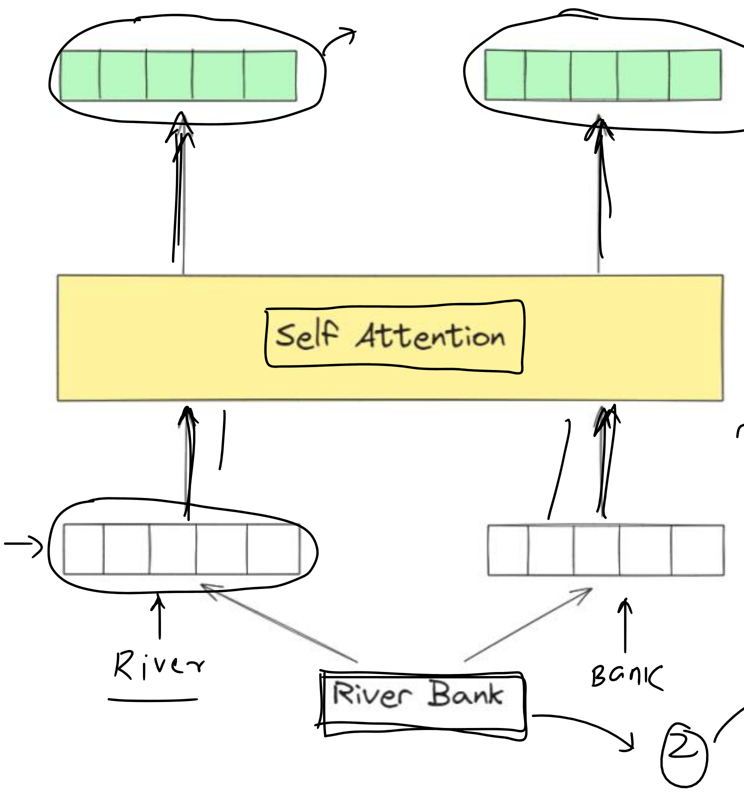



Attention is all you need
h=1
h=2
h=8
QKV
computations
self attention
512 dim
64
64+64+...8
multi
1/8th dim
joinings of persp
linear proj
improtant
mix of persput
W0
512x512
money
bank
2x512
[512x64]
64dim
2x64
self
Q1
K1
V1
Q2
K2
V2
Q8
K8
V8
[512]dim
[2x512]

The Why

23 May 2024 14:33

Self-Attention



River Bank →

money Bank →

Benefit

Parallel → drawback

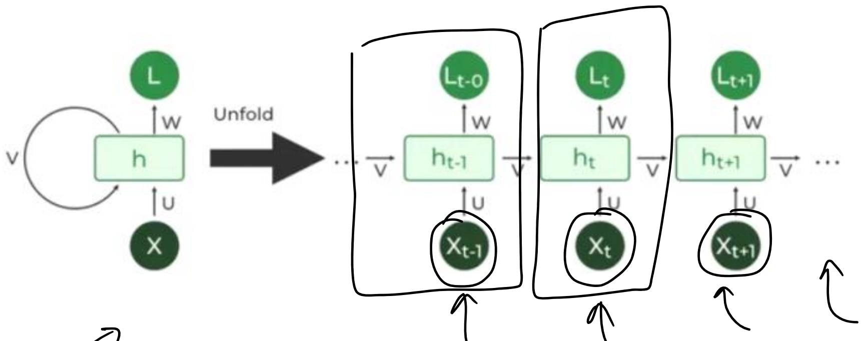
words order

Nitish killed lion

Lion killed Nitish

RNN → sequential

Nitish killed him



River bank river bank

Proposing a simple solution

23 May 2024 15:37

Smooth transition continuous

Problem 2

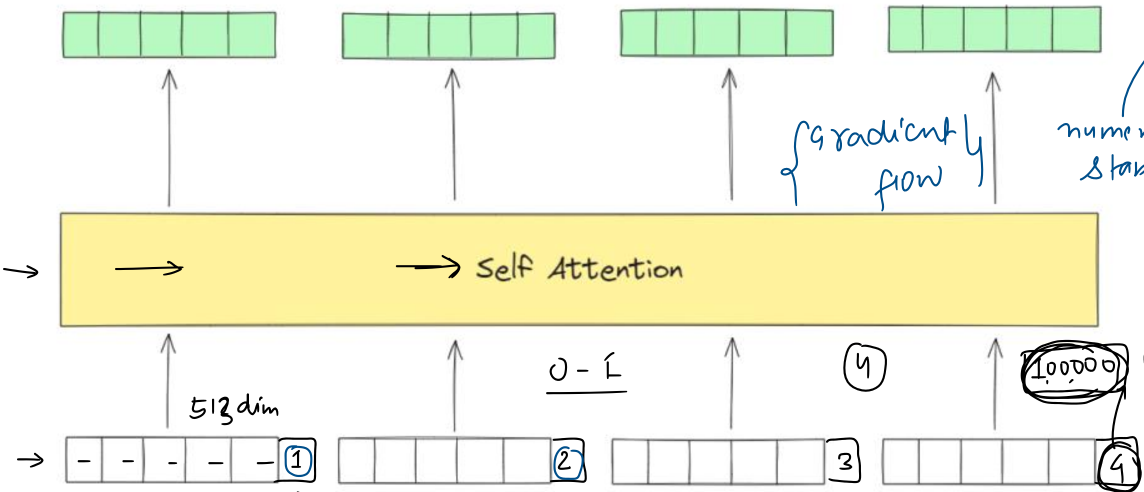
Discrete

good neural networks

numerical stabi

gradient flow

counting



0-1

4

100000 book → pdf

process

500

×200

11 word

Nitish	Killed	the	lion	You
1	2	3	4	
$\frac{}{4}$	$\frac{}{4}$	$\frac{}{4}$	$\frac{}{4}$	

0

Problem

1) Unbounded → upper limit (huge)

normalize

-1 to 1

$1 \; \frac{100000}{11}$

NN architecture

big number ← backprop

unstability

gradient unstable

0-1 ② No

→ Thank You → ①

$\frac{1}{2}$ $\frac{2}{2}$ ②

Nitish	Killed	the	lion
$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{4}{4}$

0-1

0.5 ①

Problem 3 → can't relative positioning

3-1 = 2

Nitish	killed	the	lion
1	2	3	4

discrete

1 2 3 4

absolute position

reative position

discrete

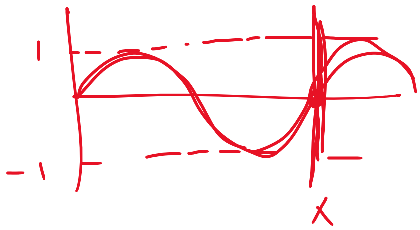
↓

periodic → relative positioning

trigonometric

↓

sine



Problem

→ unbounded (bounded) ✓✓

→ discrete (continuous) → function →

→ relative position (periodic) ✓

positional encoding → $\boxed{\text{sine}}$ → better solution

The sine function as a solution

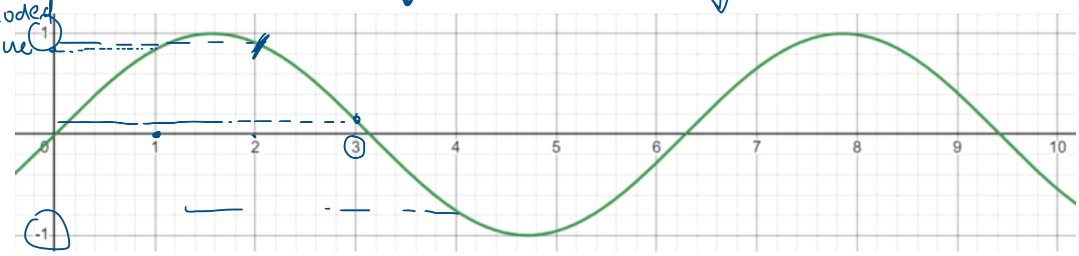
23 May 2024 17:17

periodic

$\boxed{\text{Bis problem}}$. $y = \sin(pos)$

45

position → 35



1) unbounded
2) discrete
3) relational positioning

encoded value

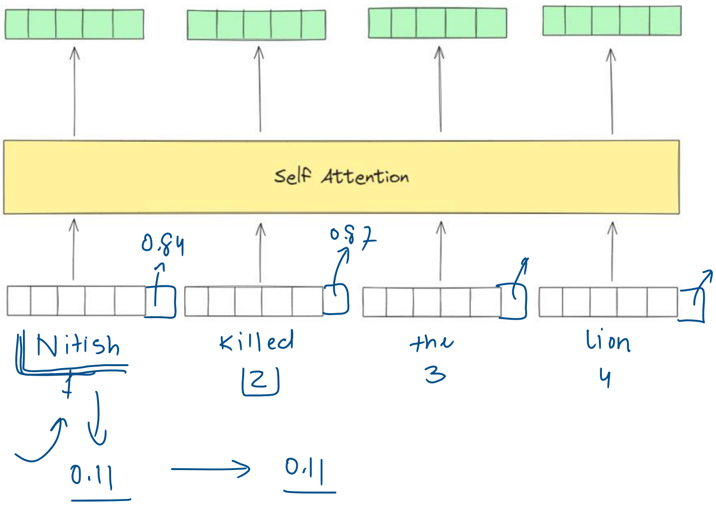
$y = \sin(\text{position})$

$y = \sin(1) = 0.84$

$y = \sin(2) = 0.87$

$y = \sin(3) = 0.14$

$y = \sin(4) = -0.90$

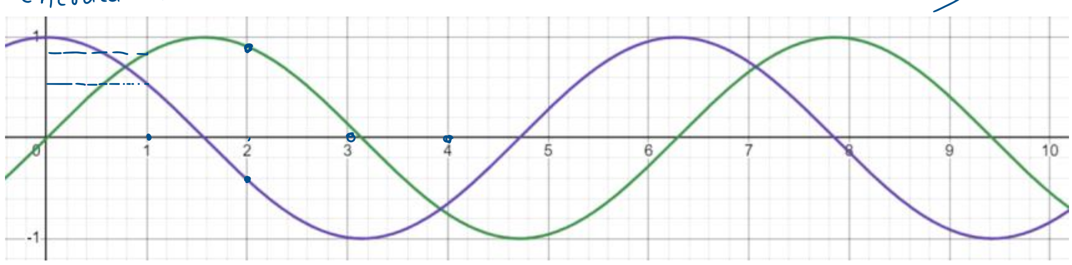


$y = \sin(1) = 0.84$

$y = \cos(1) = 0.5$

$y = \boxed{\sin(pos)}$ $y = \boxed{\cos(pos)}$

encoded value

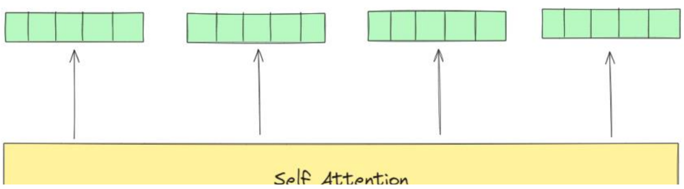


position

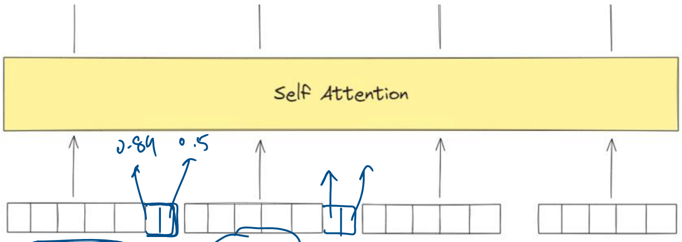
Nitish → [0.84, 0.5]

$y = \sin(2)$ $y = \cos(2)$

[0.87, -0.4]



scalar → vector



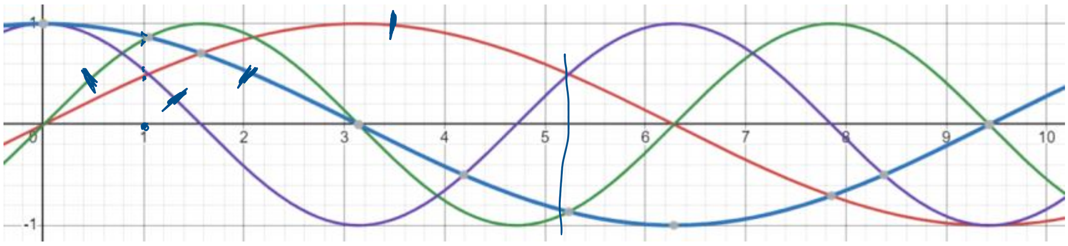
[0.87, -0.4]

scalar → vector

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

1 Nitish 2 Killed 3 the 4 lion

↳ vector → [_ , _ , _ , _] [4 number] $y = \sin(pos/2)$ $y = \cos(pos/2)$

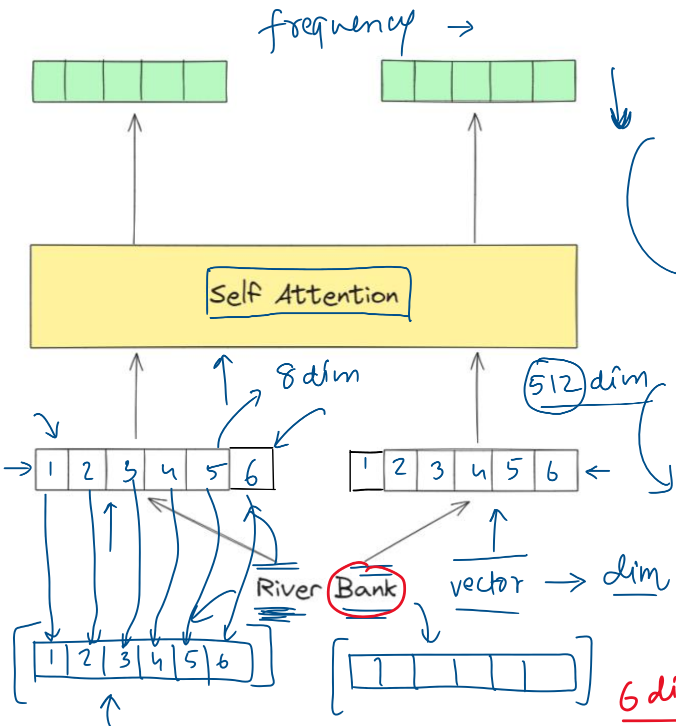


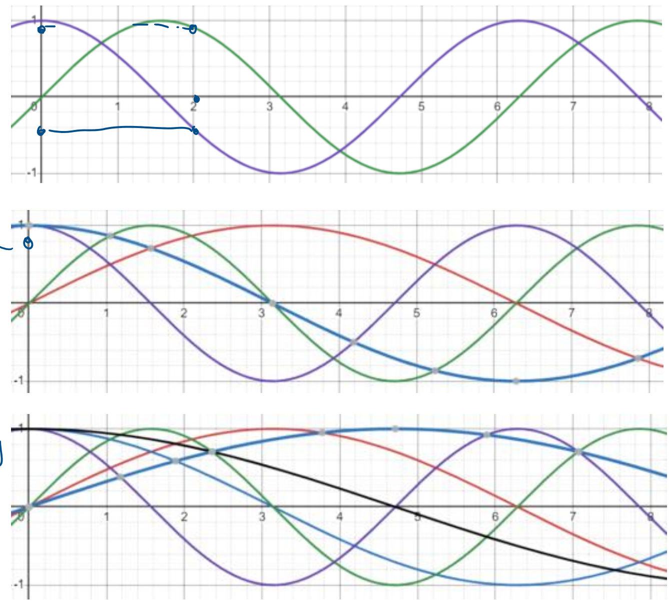
posit 6 number vector

$y = \sin(pos/3)$ $y = \cos(pos/3)$

Positional Encoding

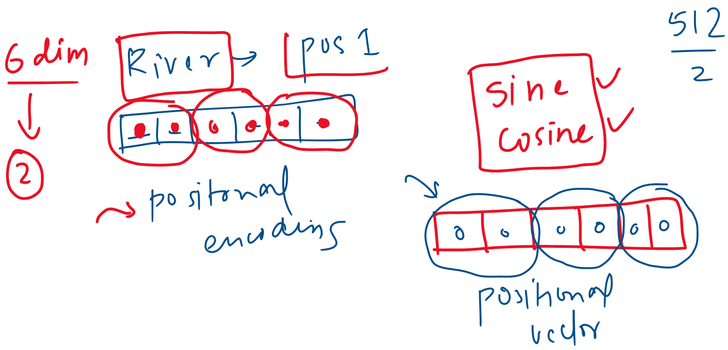
23 May 2024 20:35





embd (6 dim) + pos encd (6 dim)

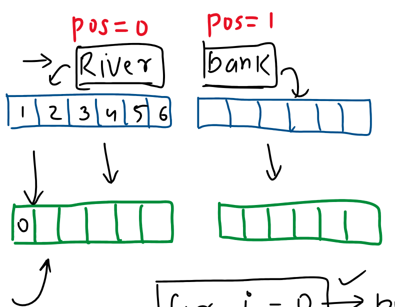
= 6 dim vector



concatenation → addition

↳ 6 dim | 6 dim

↳ 12 dim → parameters increase → training double up



pos → position

pos = 0 pos = 1

d_{model} → dim of embedding

$d_{model} = 6$

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

i = 0, 1, 2

i = 0 – $d_{model}/2$ – 1   0 - 6/2 – 3   0 – 3

0 - 2

for i = 0 → pos = 0

PE(0, 0) = sin(0 / 10000⁰) = 0

2i + 1

PE(0, 1) = cos(0 / 10000⁰) = 1

for i = 0

PE(1, 0) = sin(1 / 10000⁰) = 0.84

PE(1, 1) = cos(1 / 10000⁰) = 0.54

for i = 1

for i = 1

... = 0.04

for i=1 ✓

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

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

for i=1

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

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

for i=2 ✓

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

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

for i=2

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

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

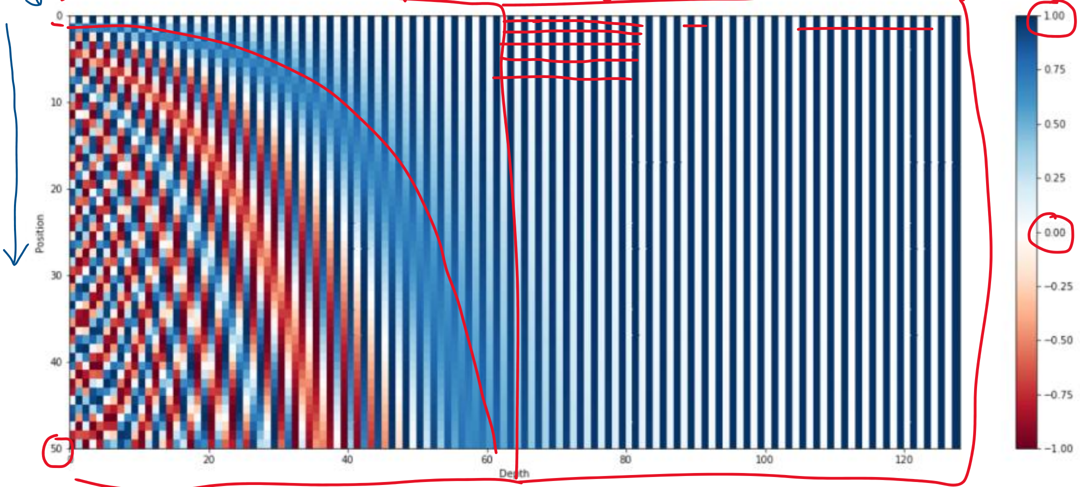
River →	0	1	0	1	0	1
bank →	0.84	0.54	0.04	0.99	0.00	0.99

Interesting Observations

24 May 2024 02:06

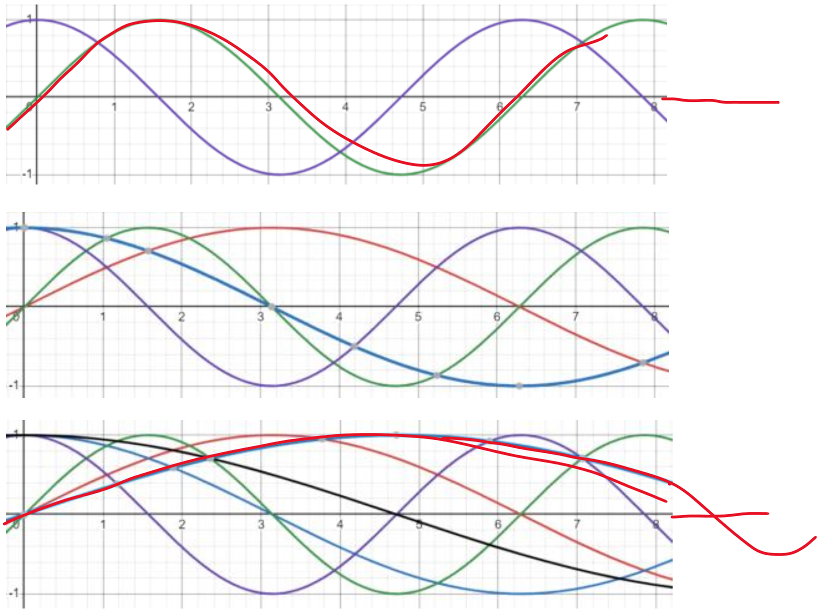
sentence → 50 words ↓ 50 → d model → 128

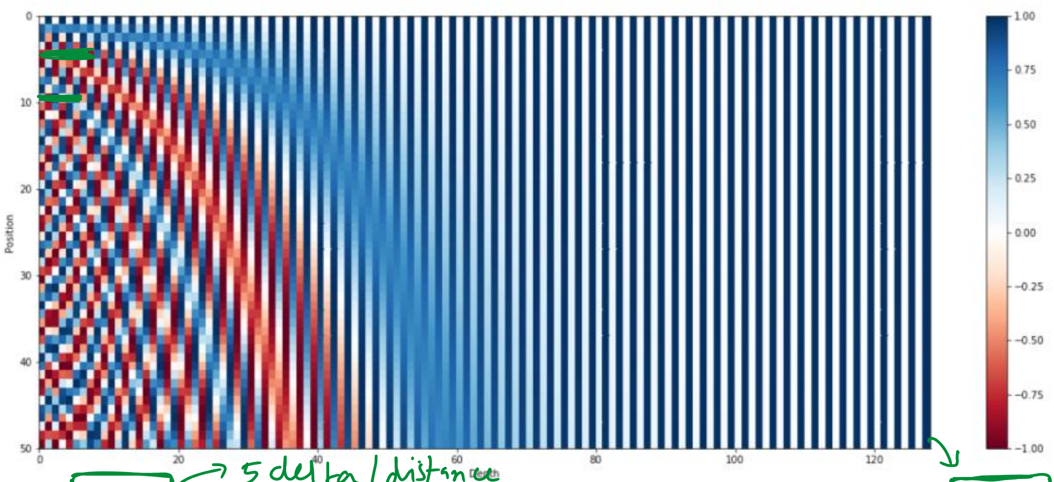
└→ embedding → 128 d model = 128



100 binary encoding 4 bit 8 bit

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

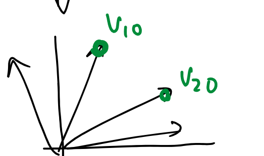




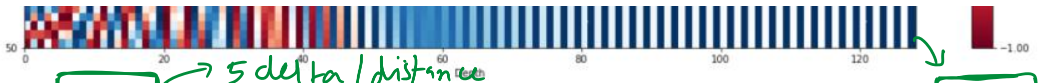
5 delta / distance

matrix matrix →

50 words ↓ 50 vectors ↓ 128 dim



10 delta



$V_5 \rightarrow \boxed{\text{matrix}} \rightarrow$ 5 delta / distance

properly

$\hookrightarrow V_{10}$

$V_{12} \rightarrow V_{17}$ $V_{21} \rightarrow V_{26}$

$\boxed{V_{10}} \rightarrow$ linear transformation $\rightarrow \boxed{\text{matrix}} \rightarrow$ 10 delta distance

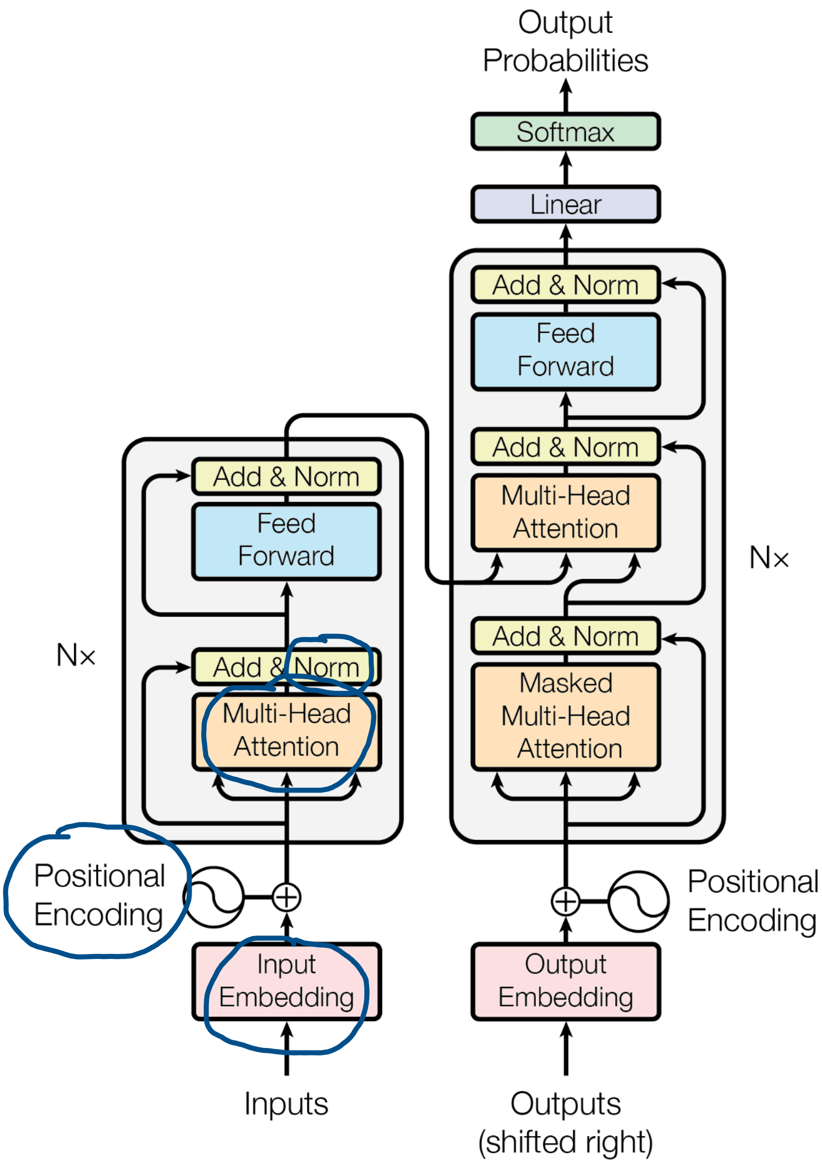
$\rightarrow V_{20}$ for each delta $V_{30} \rightarrow V_{40}$

$V_{40} \rightarrow V_{50}$

linear / matrix

Agenda

07 June 2024 02:03

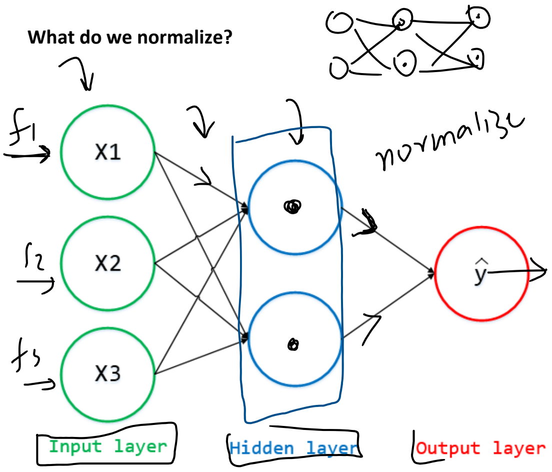


What is Normalization

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.

What do we normalize?



f1
27
23
41

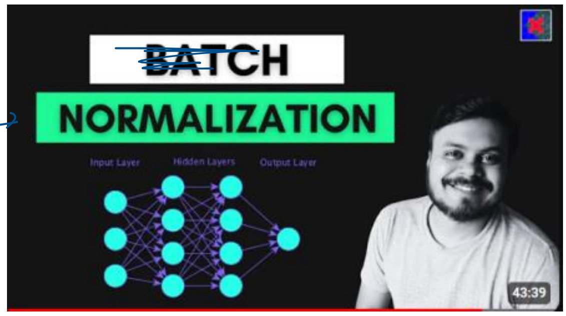
min-max

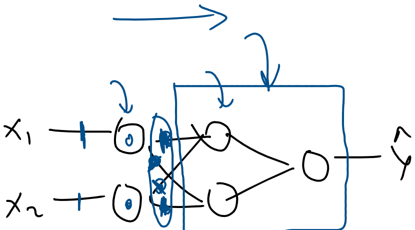
$$\frac{x_i - \mu}{\sigma} \rightarrow \begin{matrix}\mu = 0\\ \sigma = 1\end{matrix}$$

f1 | f2 | f3

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.





input data → dist

covariate shift

internal

red roses

pred → yellow white

Activation

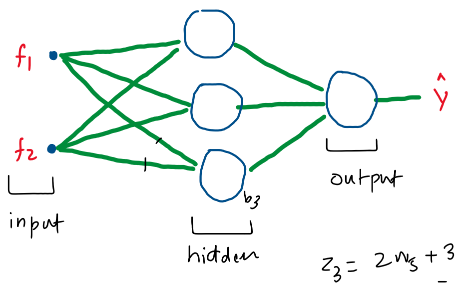
input

Batch Norm → Layer

+ Batch Norm → Layer Norm

Batch Norm(Revision)

05 June 2024 10:39



$z_3 = 2w_5 + 3w_6 + b_3$
$= 4$

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

$z_1 = \underbrace{2w_1 + 3w_2 + b_1}_{} = 7$

$z_2 = 2w_3 + 3w_4 + b_2 = 5$

$\frac{7-\mu_1}{\sigma_1} = \underline{0.36}\, \underset{(1)}{\gamma_1} + \underset{(0)}{\beta_1} = 0.36$

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

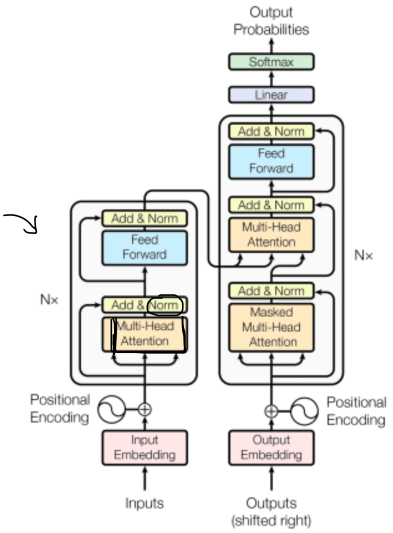
$\frac{5-\mu_2}{\sigma_2} = -0.21\gamma_2 + \beta_2 = -0.21$

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

Why don't we use Batch Norm in Transformers?

05 June 2024 10:40

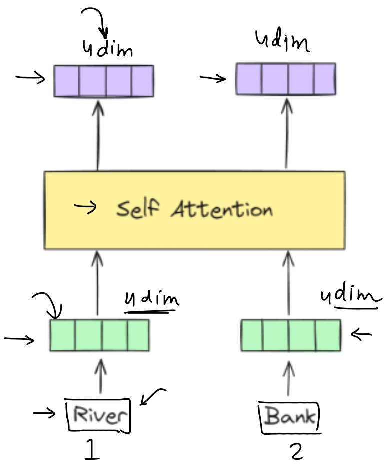
↳ self attention



Batch Norm

↑

Self-Attention



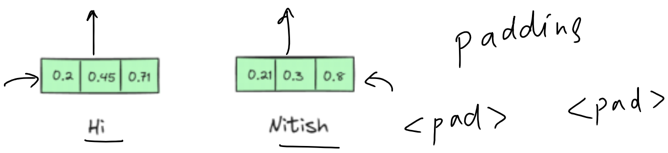
Batching

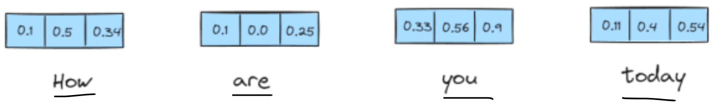
dim=3

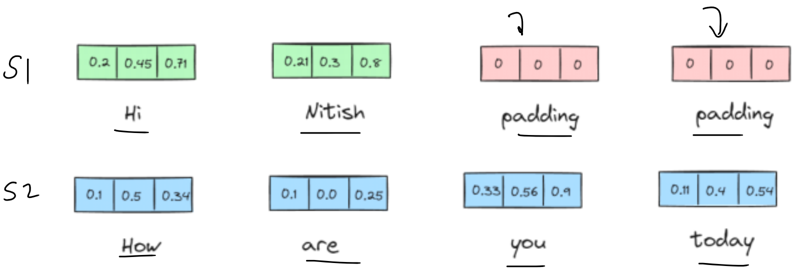
	Review	Sentiment
r1	Hi Nitish	1
r2	How are you today	0
r3	I am good	0
r4	You?	1

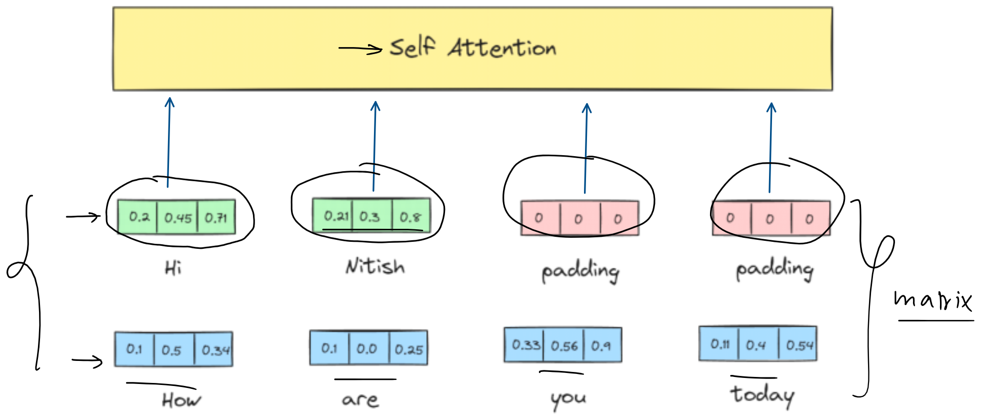
Embedding dimension - 3

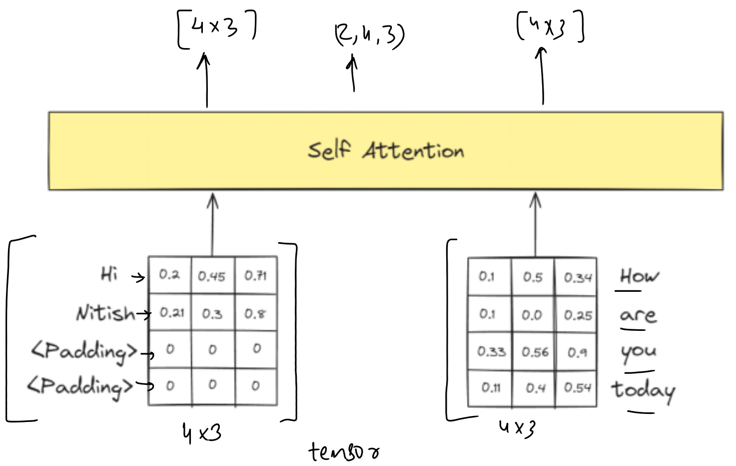
Batch Size - 2

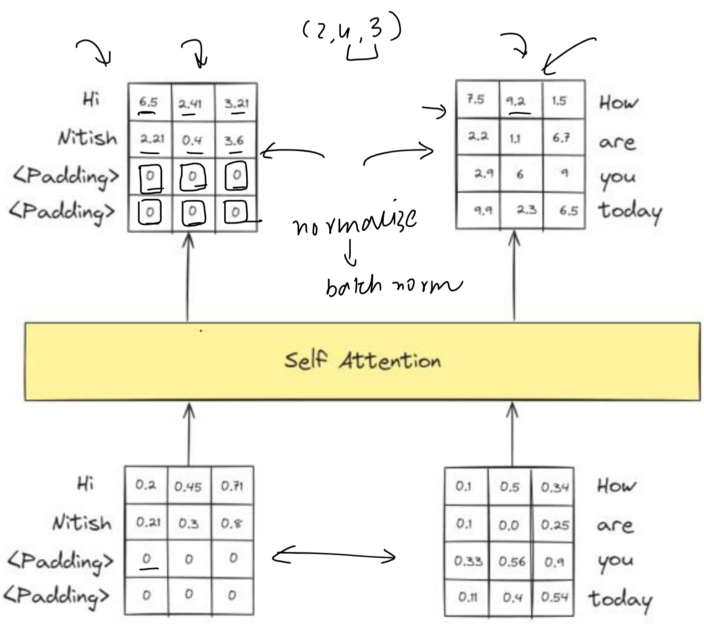


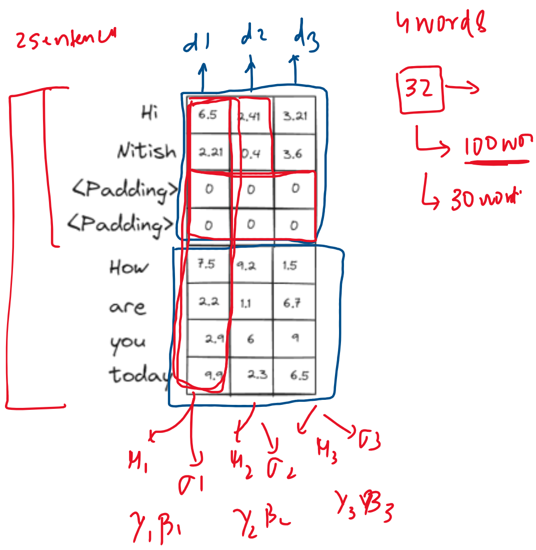





Self Attention
0.2 0.45 0.71
Hi
0.21 0.3 0.8
Nitish
0 0 0
padding
0 0 0
padding
0.1 0.5 0.34
How
0.1 0.0 0.25
are
0.33 0.56 0.9
you
0.11 0.4 0.54
today
matrix


[4x3]
(2,4,3)
[4x3]
Self Attention
Hi
Nitish
<Padding>
<Padding>
0.2 0.45 0.71
0.21 0.3 0.8
0 0 0
0 0 0
4x3
0.1 0.5 0.34
0.1 0.0 0.25
0.33 0.56 0.9
0.11 0.4 0.54
How
are
you
today
4x3

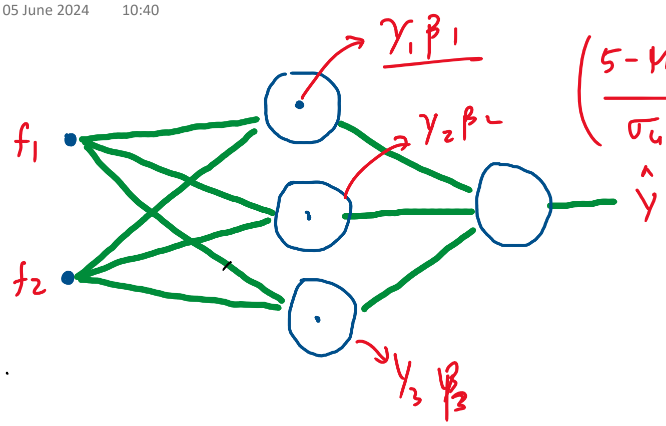

tensor
(2,4,3)
Hi
Nitish
<Padding>
<Padding>
6.5 2.41 3.21
2.21 0.4 3.6
0 0 0
0 0 0
7.5 9.2 1.5
2.2 1.1 6.7
2.9 6 9
9.9 2.3 6.5
How
are
you
today
normalize
batch norm
Self Attention
Hi
Nitish
<Padding>
<Padding>
0.2 0.45 0.71
0.21 0.3 0.8
0 0 0
0 0 0
0.1 0.5 0.34
0.1 0.0 0.25
0.33 0.56 0.9
0.11 0.4 0.54
How
are
you
today


2 sentence
d1 d2 d3
4 words
32
100 words
30 words
Hi
Nitish
<Padding>
<Padding>
How
are
you
today
6.5 2.41 3.21
2.21 0.4 3.6
0 0 0
0 0 0
7.5 9.2 1.5
2.2 1.1 6.7
2.9 6 9
9.9 2.3 6.5
μ1 σ1
μ2 σ2
μ3 σ3
γ1,β1
γ2,β2
γ3,β3

Layer Norm

05 June 2024 10:40

across batch across feature



$$\left(\frac{5-\mu_4}{\sigma_4}\right)\gamma_2+\beta_2$$

f_1	f_2	z_1	z_2	z_3	
2	3	7	5	4	μ_1, σ_1
1	1	2	3	4	μ_2, σ^2
5	4	1	2	3	
6	1	7	5	6	
7	1	3	3	4	μ_5, σ_5

$$\left(\frac{2-\mu_2}{\sigma_2}\right)\gamma_1+\beta_1$$

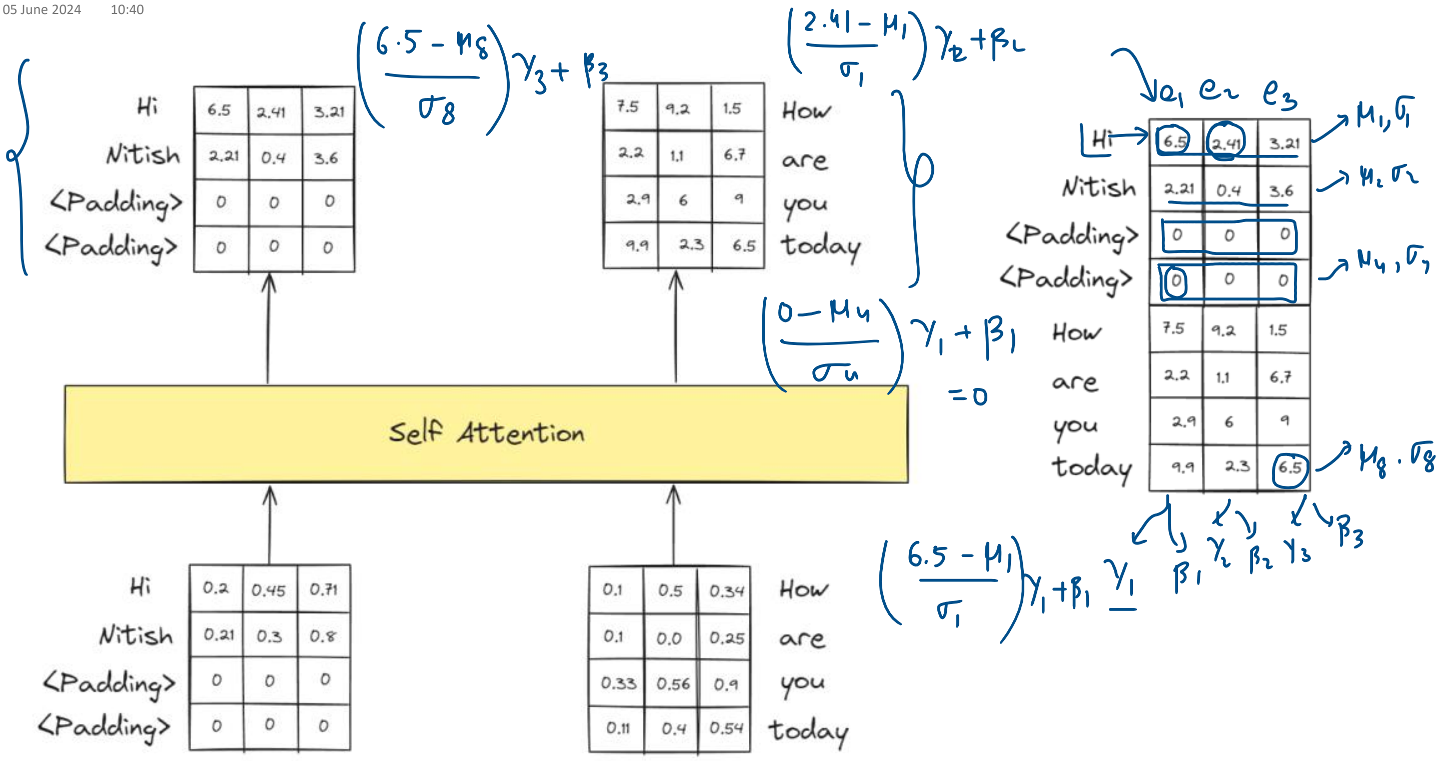
$$\left(\frac{4-\mu_1}{\sigma_1}\right)\gamma_3+\beta_3$$

$$\left(\frac{5-\mu_1}{\sigma_1}\right)\gamma_2+\beta_2$$

$$\frac{7-\mu_1}{\sigma_1} = \boxed{0.3\,\gamma_1+\beta_1}$$

Layer Norm in Transformers

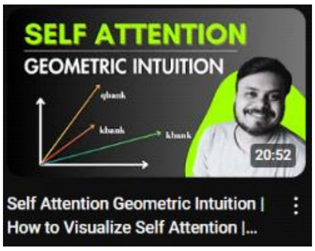
05 June 2024 10:40



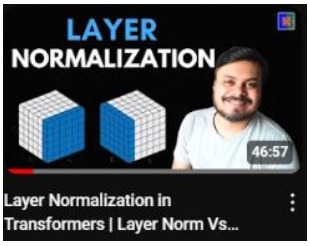
Recap

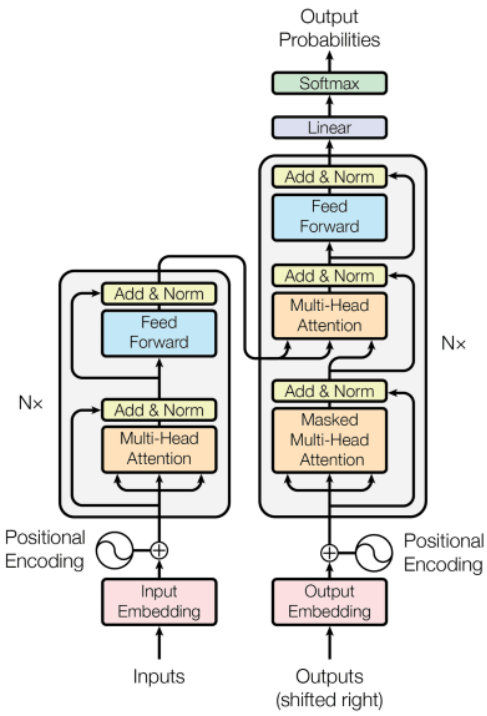
09 July 2024 08:47

✓



$\boxed{7hrs}$



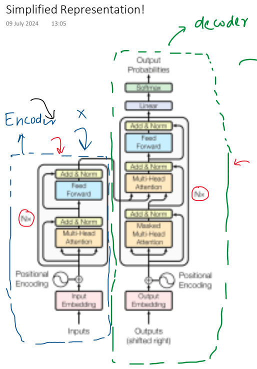


complex

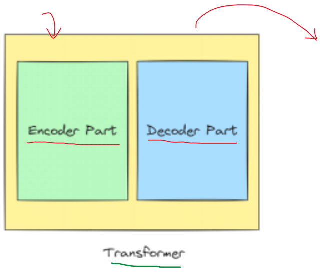
[encoder
Decoder ✓]

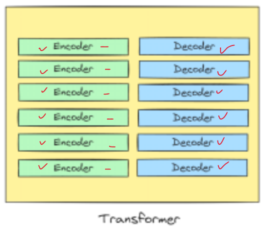
Simplified Representation!

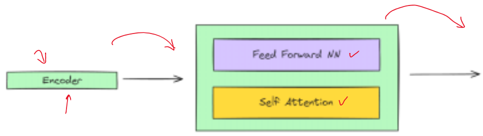
09 July 2024 13:05

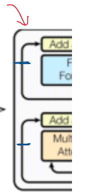


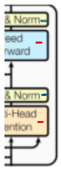
Encoder → 6
Decoder → 6

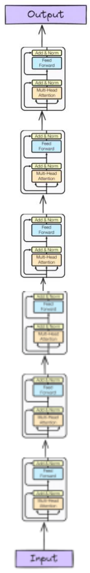






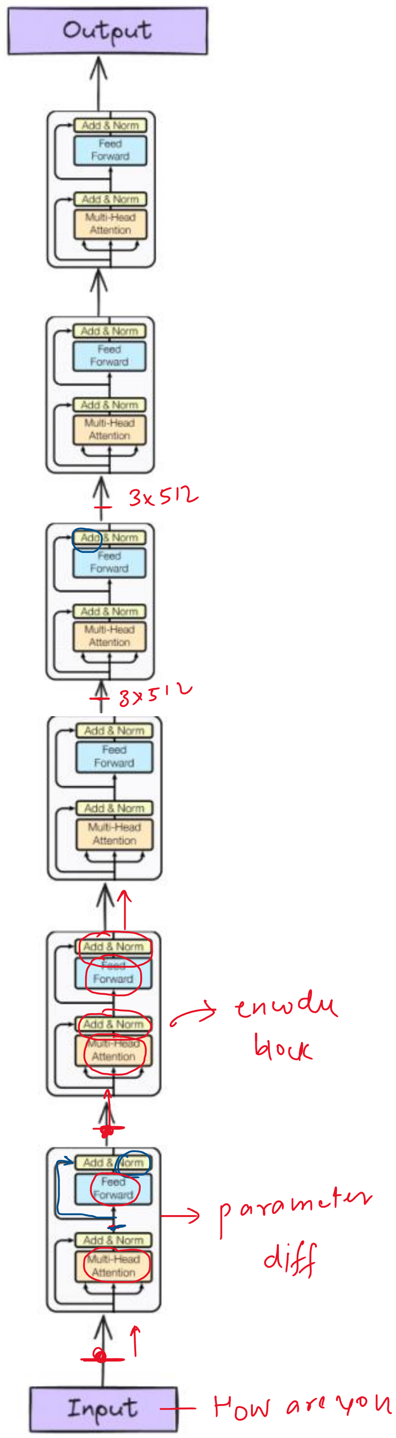


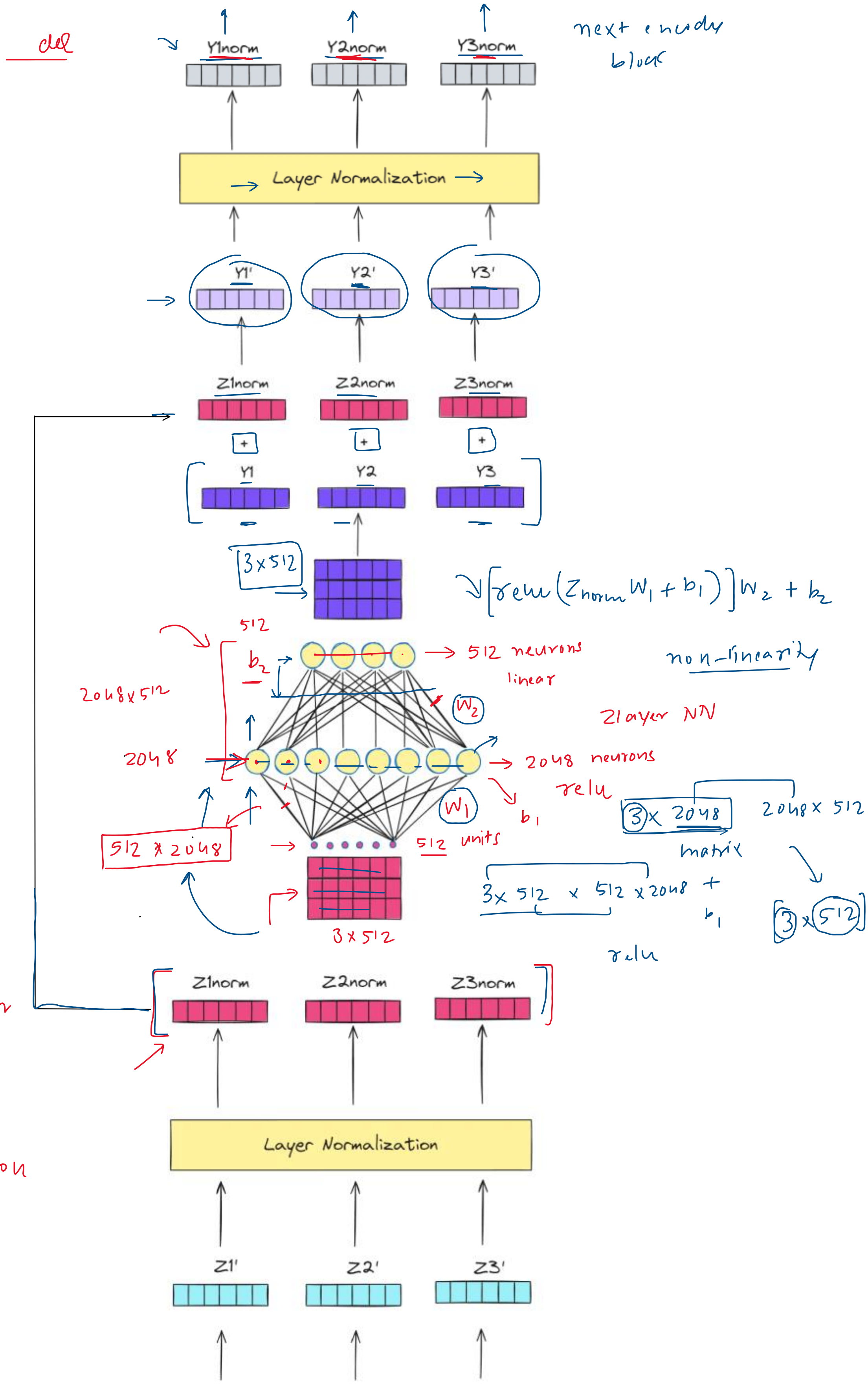

& Norm
& Norm

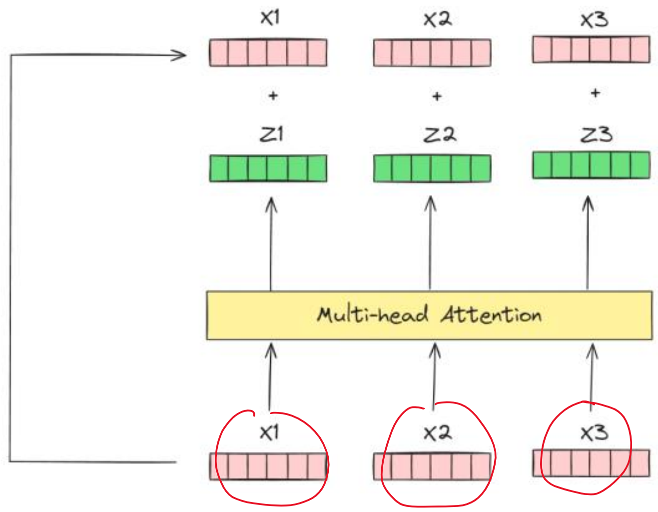

Output
Input

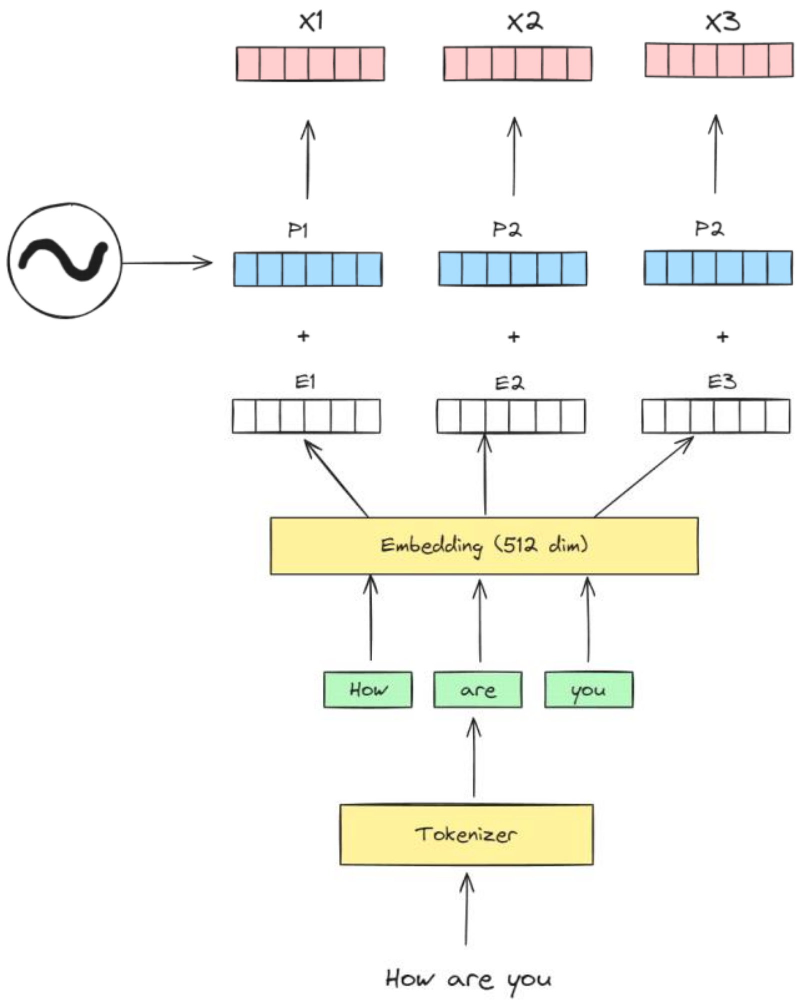
Encoder Architecture

09 July 2024 16:31






x1
x2
x3
+
+
+
Z1
Z2
Z3
Multi-head Attention
x1
x2
x3


x1
x2
x3
P1
P2
P2
+
+
+
E1
E2
E3
Embedding (512 dim)
How
are
you
Tokenizer
How are you

Some questions

09 July 2024 20:48

alternate

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

gradient flow

scratch pytorch

language ↓ complex

1) stable training — deep NN

ResNet

vanishing grad

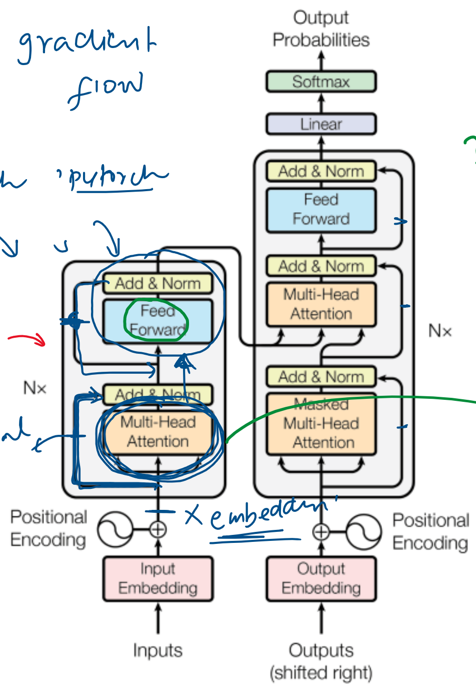
residual

relu

non-linearity

linear

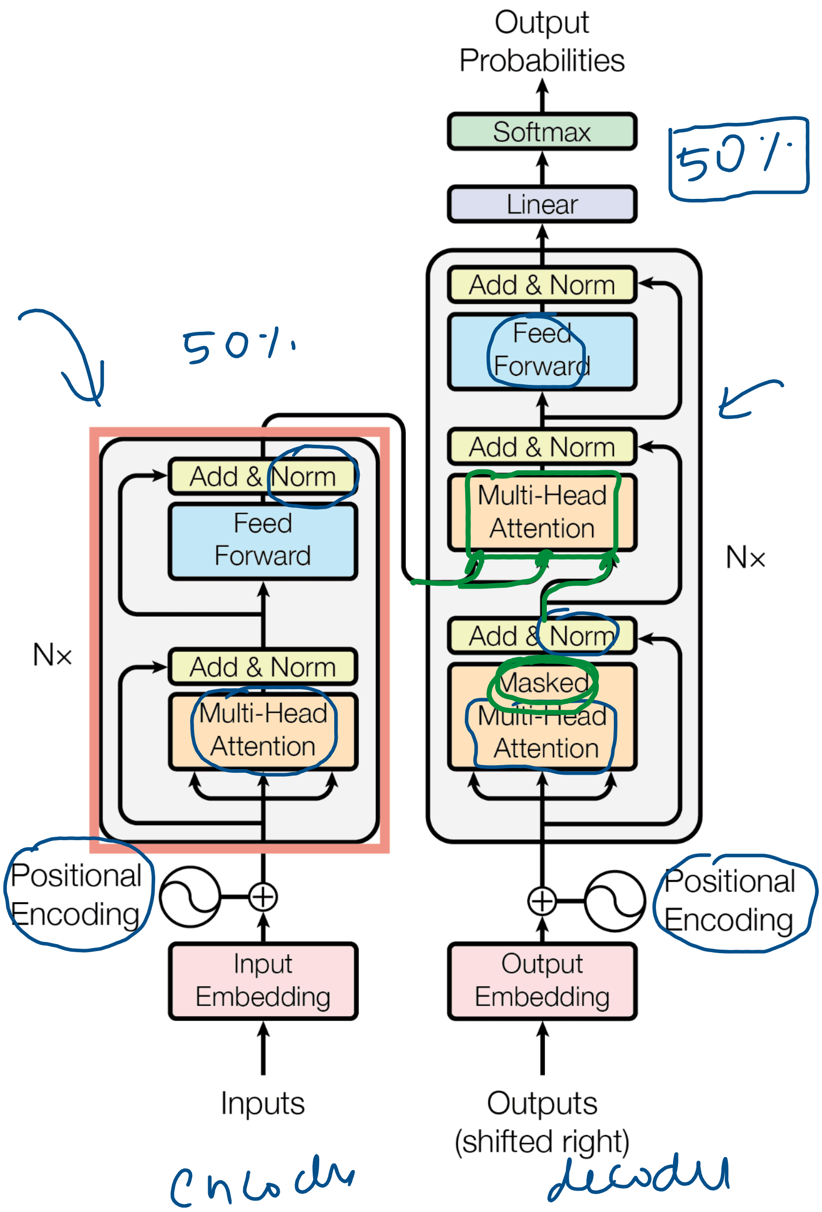
x embedding



2)

Recap

23 July 2024 17:09



Autoregressive models
23 July 2024 17:39

economics → Time series

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

masked self attention

TF decoder

prediction → Inference → Autoreg

Training → Non-auto

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. →

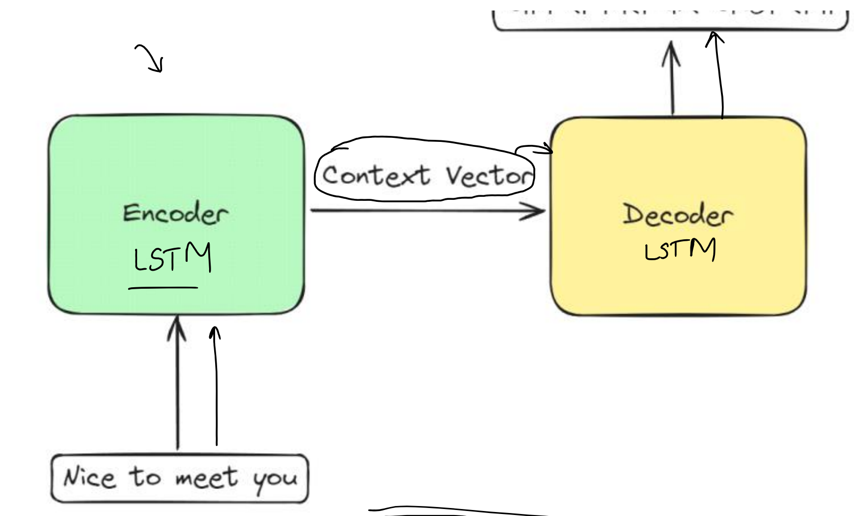
F WT

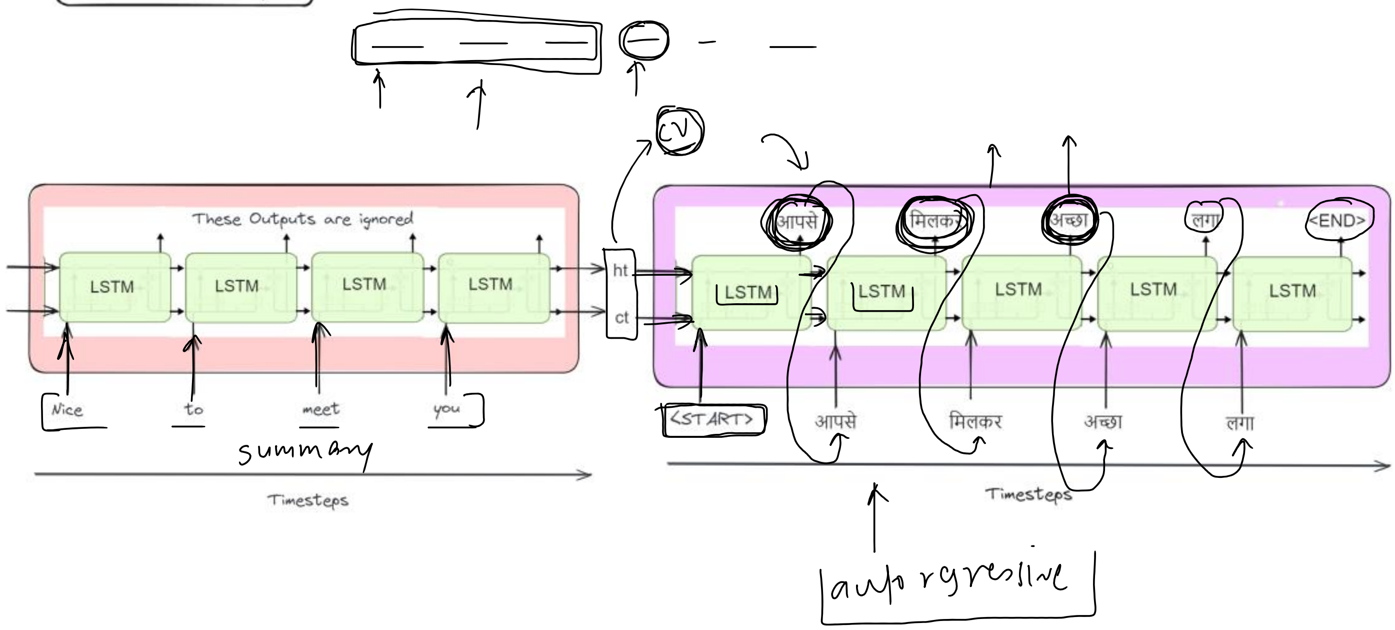
ML model → stock pred → 29 30 O

W T F

↦ Encoder - Decoder → Ilya

आप से मिल कर अच्छा लगा


Encoder
LSTM
Context Vector
Decoder
LSTM
Nice to meet you


These Outputs are ignored
LSTM
LSTM
LSTM
LSTM
Nice
to
meet
you
summary
Timesteps
ht
ct
CV
आपसे
मिलकर
अच्छा
लगा
<END>
LSTM
LSTM
LSTM
LSTM
LSTM
<START>
आपसे
मिलकर
अच्छा
लगा
Timesteps
auto regressive

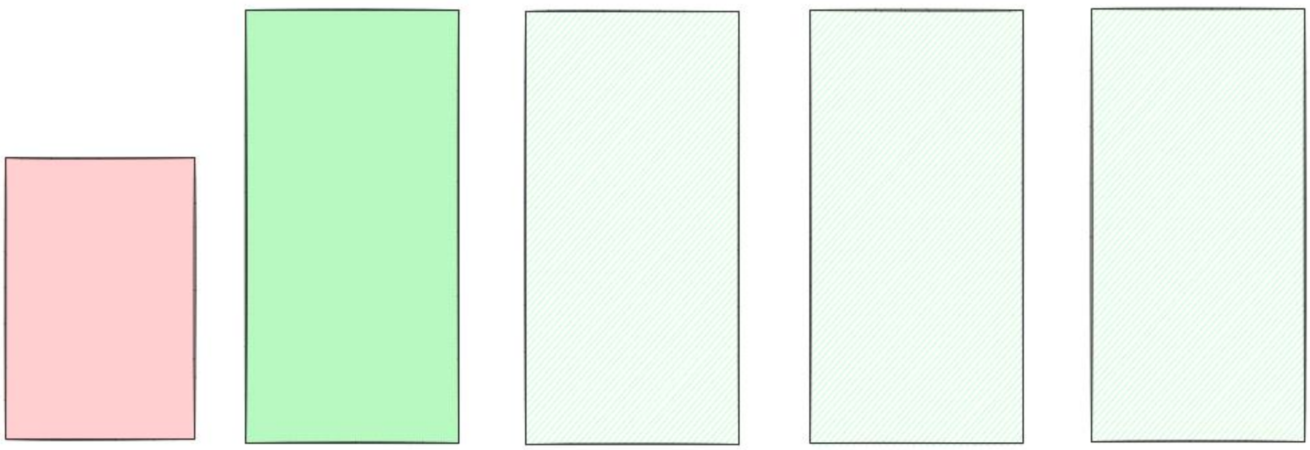
Transformer as an Autoregressive Model

23 July 2024 23:16

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

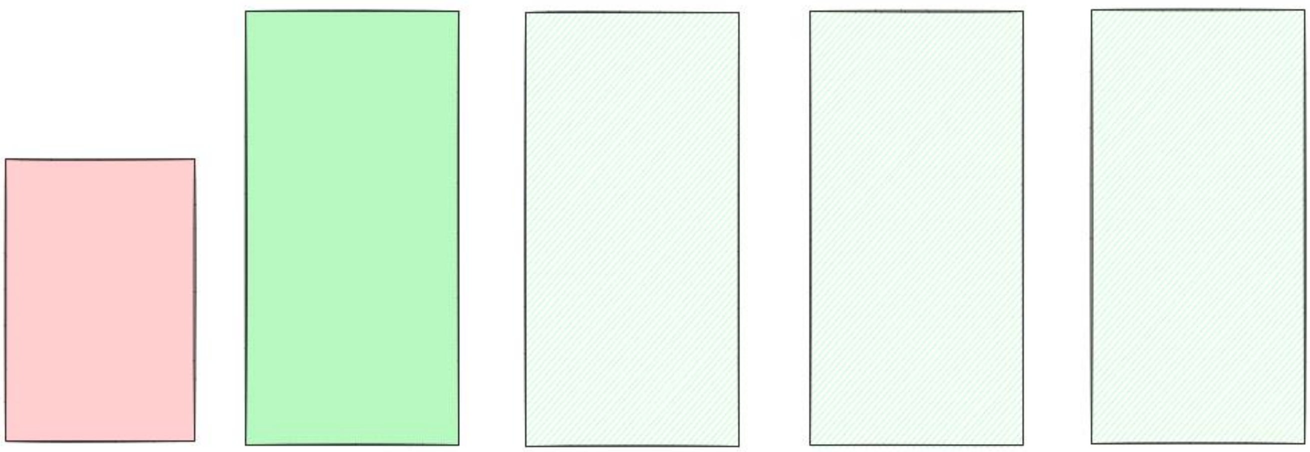
Inference

Query Sentence -> I am fine



Training

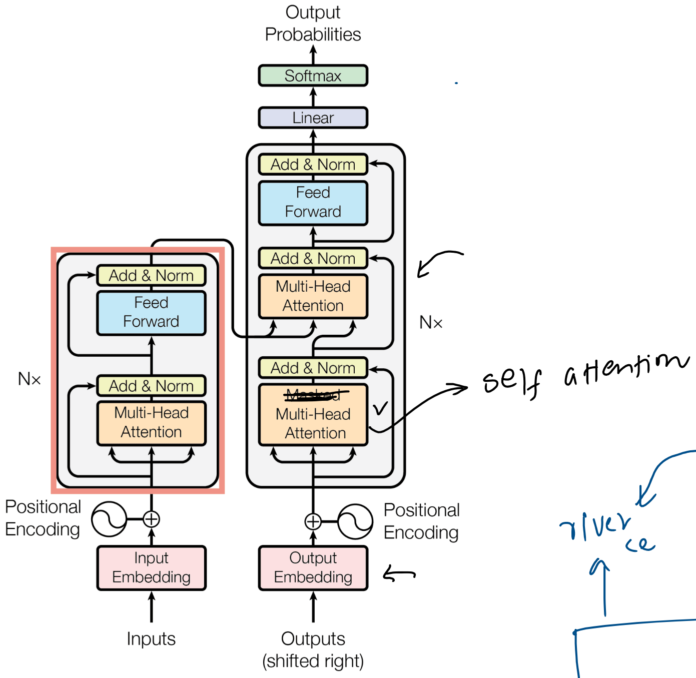
S.No	English Sentence	Hindi Sentence
1	How are you?	आप कैसे हैं
2	Congratulations	बधाई हो
3	Thank you	धन्यवाद

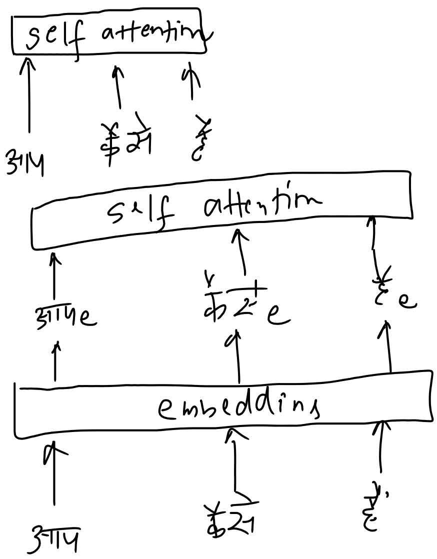


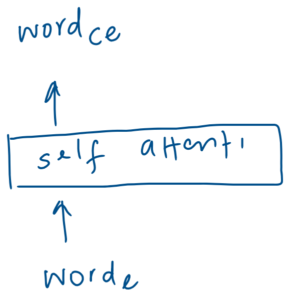
The problem in parallelizing

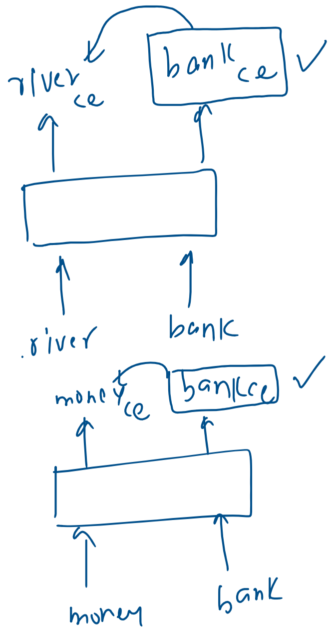
25 July 2024 22:55

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







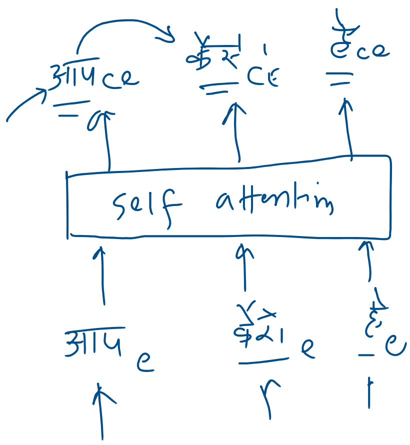


river bank

money bank

Data leakage

आप कैसे हैं



आप_ce = 0.8 आप_e + 0.1 कैसे_e + 0.1 हैं_e

कैसे_ce = 0.15 आप_e + 0.75 कैसे_e + 0.1 हैं_e

हैं_ce = 0.1 आप_e + 0.2 कैसे_e + 0.7 हैं_e

current token value

↓

future token value

Cheating

AR | NAR

→ No Data Leakage | → fast

→ slow | → Data leakage

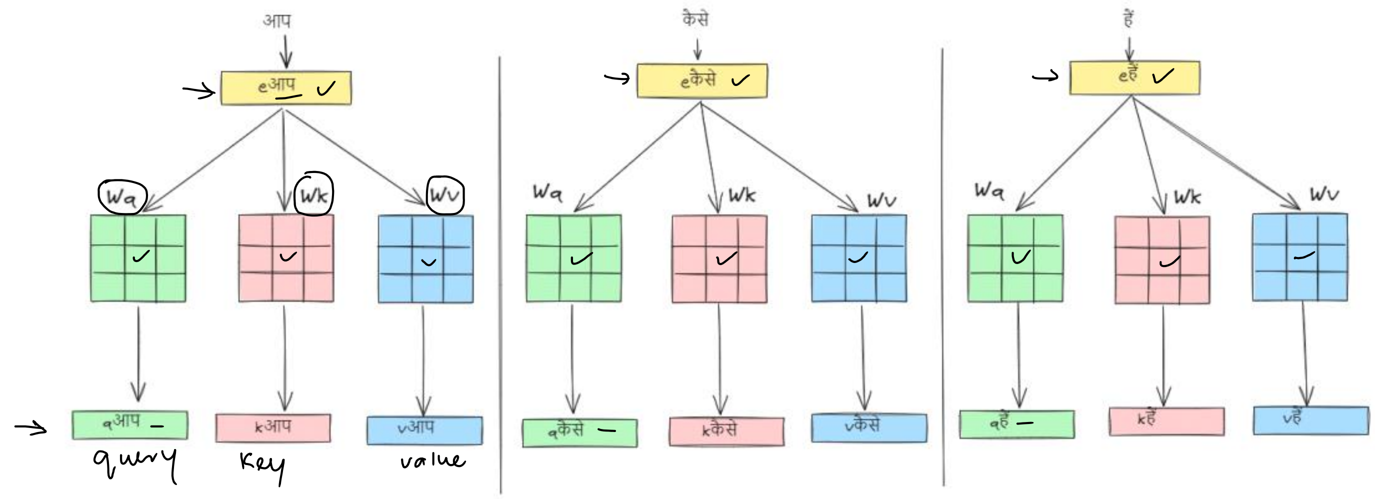
→ yes → self attention

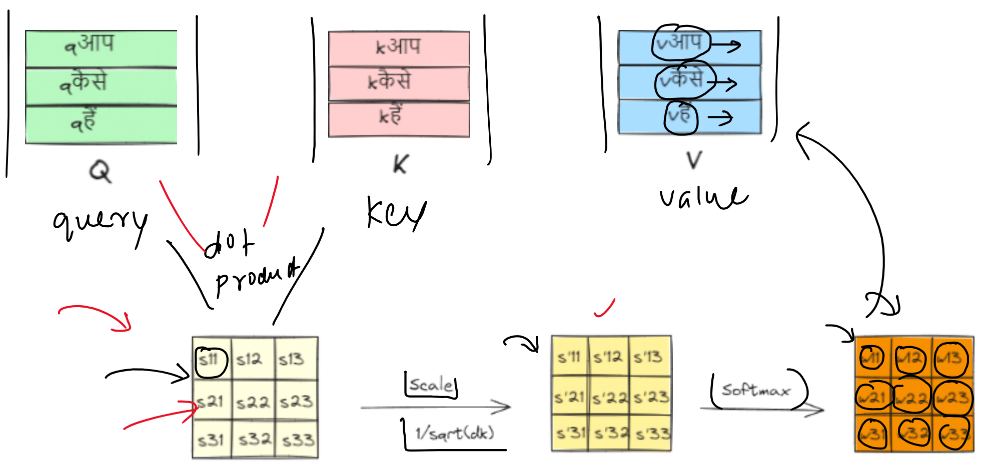
$\boxed{\text{how}}$

Duvidha

Finding the answer

26 July 2024 00:21





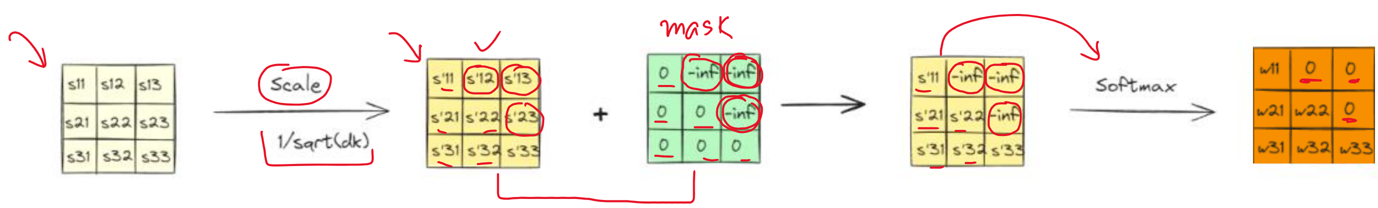
आपce = w11 * vआप + w12 * vकैसे + w13 * vहैं

कैसे ce = w21 * vआप + w22* vकैसे + w23 * vहैं

हैंce = w31 * vआप + w32 * vकैसे + w33 * vहैं

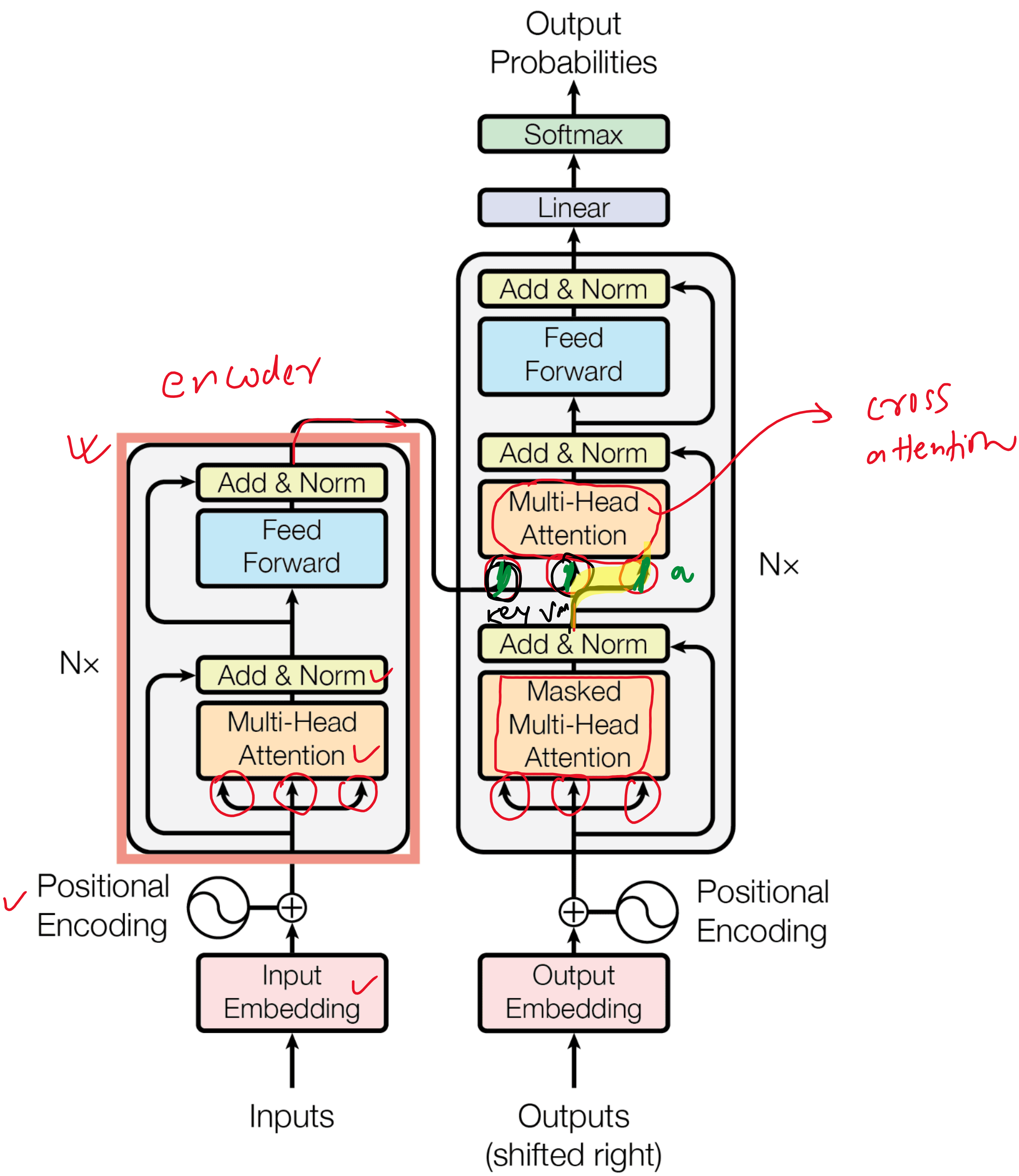
$W_{12} = W_{13} = W_{23} = 0$

$softmax(-\infty) = 0$



Plan of Action

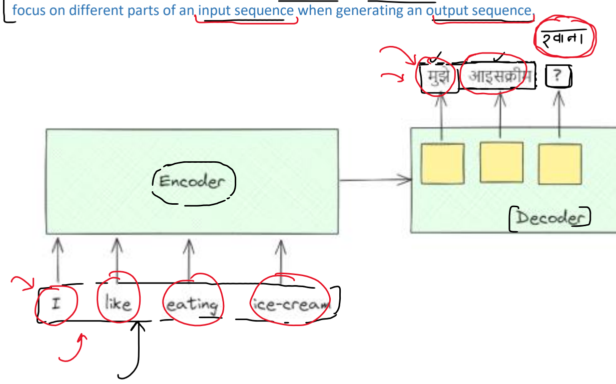
12 August 2024 17:53



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.



translation

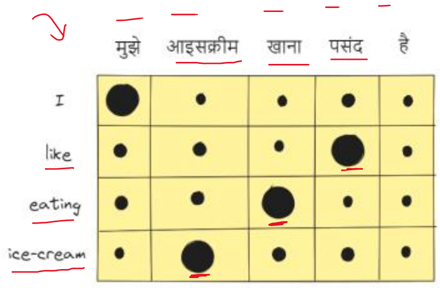
t = 3

→ 1) what generated till now

2) input sentence

self attention

cross-attention



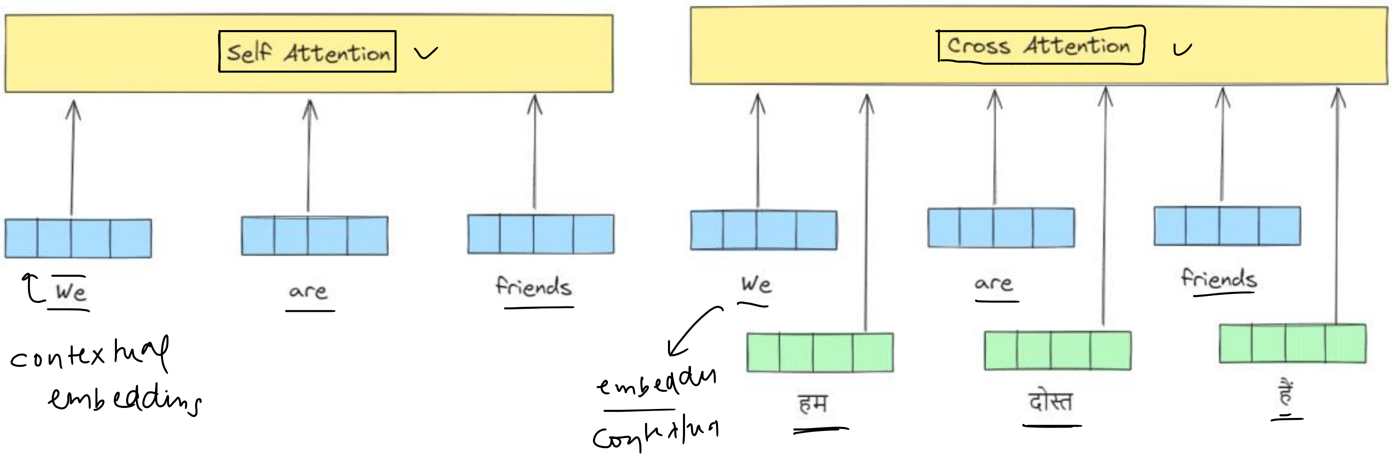
Cross Attention is conceptually very similar to Self-Attention

Self-Attention Vs Cross Attention

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

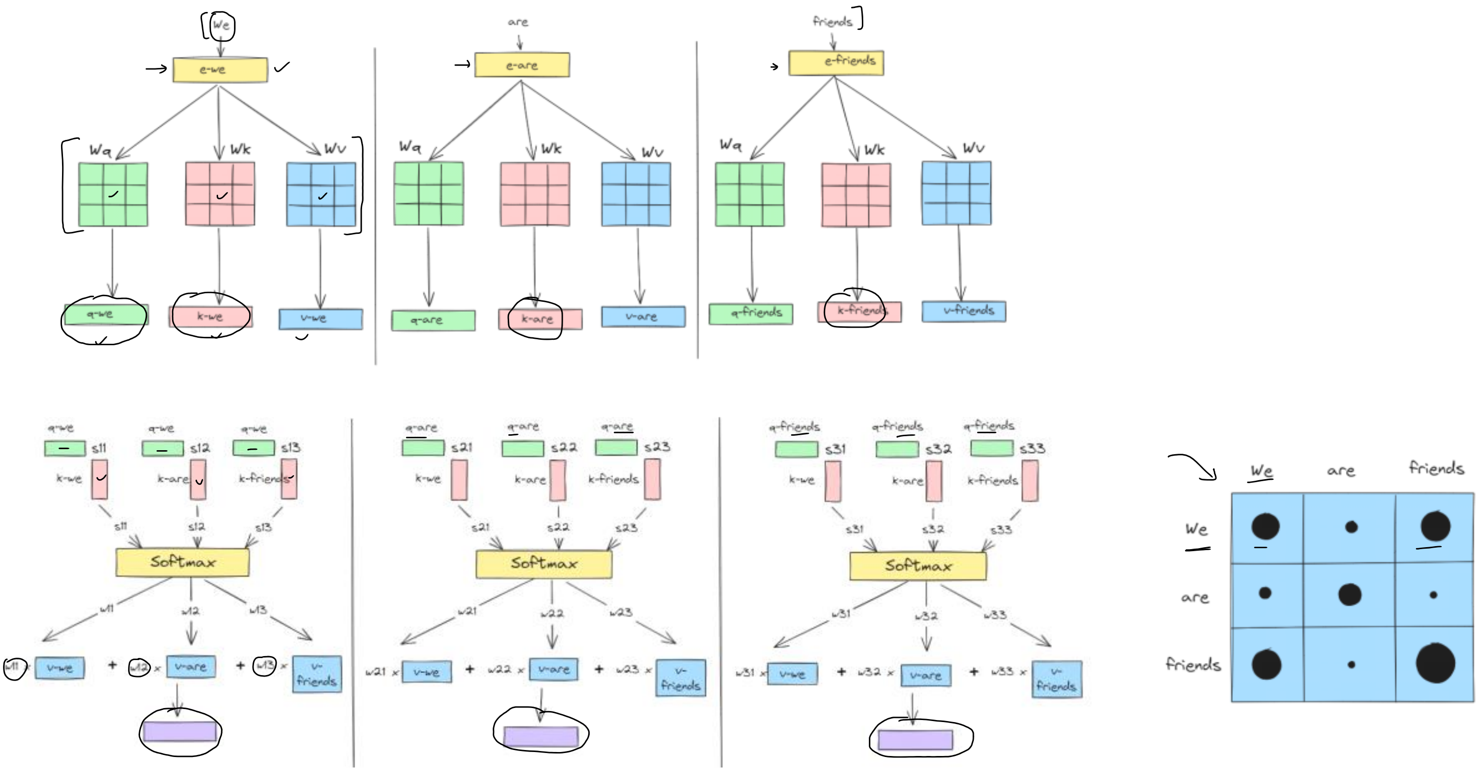
Self-Attention Vs Cross Attention (Input)

13 August 2024 08:22



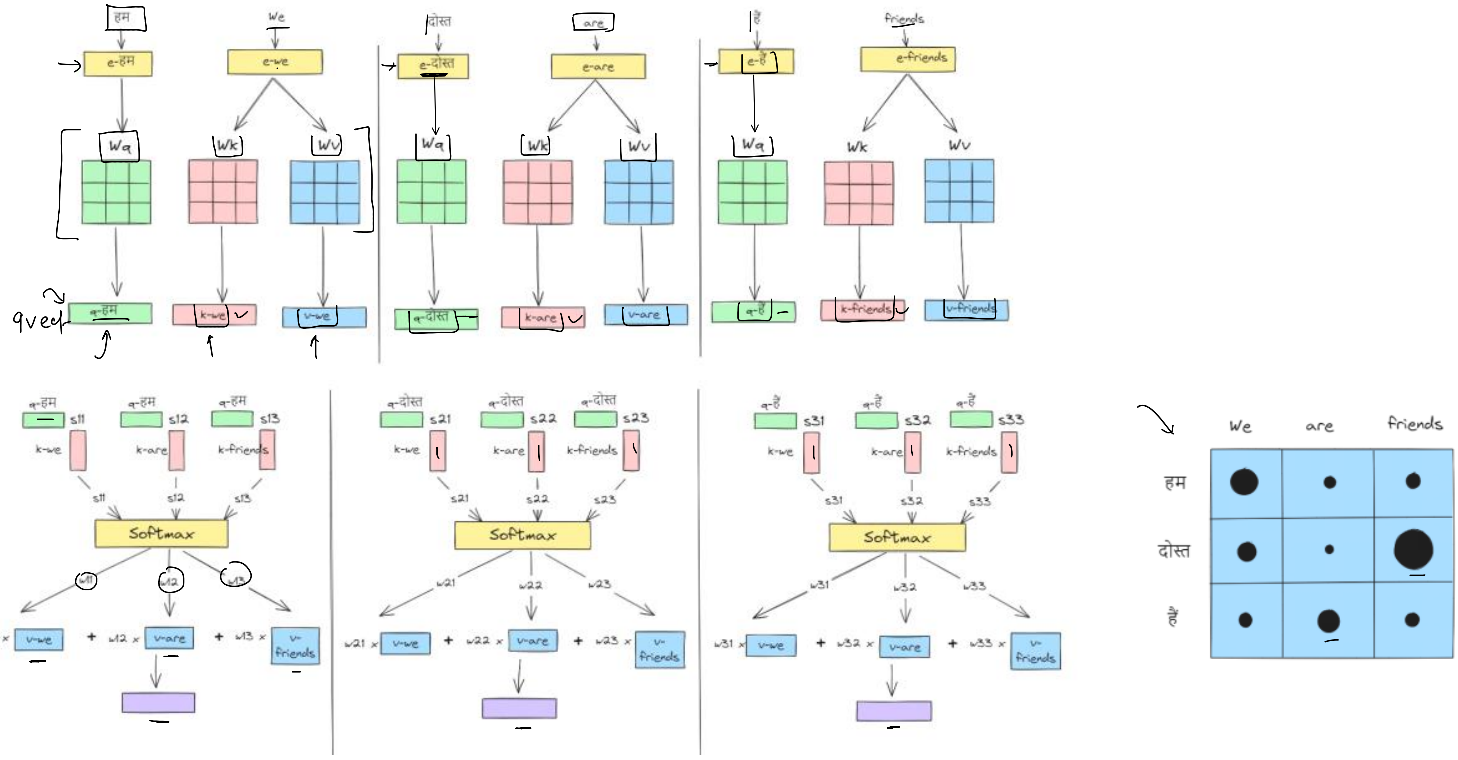
Self-Attention Vs Cross Attention (Processing)

12 August 2024 17:54



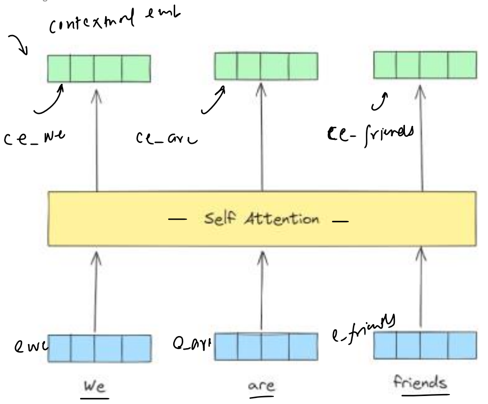
query → output seq
key and value - input seq

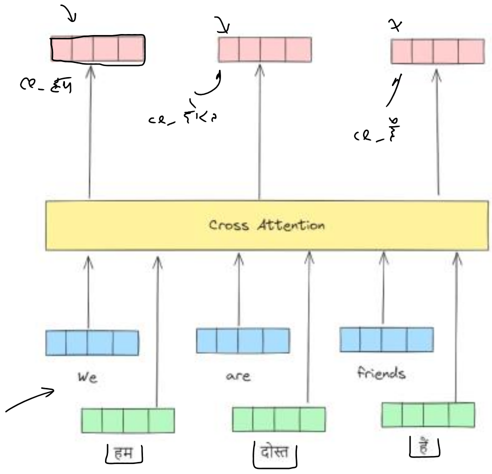
eng - hindi



Self-Attention Vs Cross Attention [Output]

12 August 2024 18:05





$$\text{ce\_हम} = 0.5 \times \text{e\_we} + 0.3 \times \text{e\_are} + 0.2 \times \text{e\_friends}$$

$$\text{ce\_दोस्त} = 0.2 \times \text{e\_we} + 0.2 \times \text{e\_are} + 0.6 \times \text{e\_friends}$$

$$\text{ce\_हैं} = 0.3 \times \text{e\_we} + 0.4 \times \text{e\_are} + 0.3 \times \text{e\_friends}$$

$$\text{ce\_we} = 0.8 \times \text{e\_we} + 0.1 \times \text{e\_are} + 0.1 \times \text{e\_friends}$$

$$\text{ce\_are} = 0.15 \times \text{e\_we} + 0.75 \times \text{e\_are} + 0.1 \times \text{e\_friends}$$

$$\text{ce\_friends} = 0.2 \times \text{e\_we} + 0.1 \times \text{e\_are} + 0.7 \times \text{e\_friends}$$

Cross Attention Vs Bahdanau/Luong Attention

12 August 2024 18:06

Use-cases

12 August 2024 18:07

image caption

$\underbrace{\square \rightarrow \text{text}}_{\text{cross atten.}}$

$\underbrace{\boxed{\text{text}}\ \text{to}\ \boxed{\text{image}}}_{\downarrow\ \text{Cross}}$

$\underbrace{\underbrace{\text{text}}\ \text{to}\ \underbrace{\text{speech}}}_{\text{cross}}$

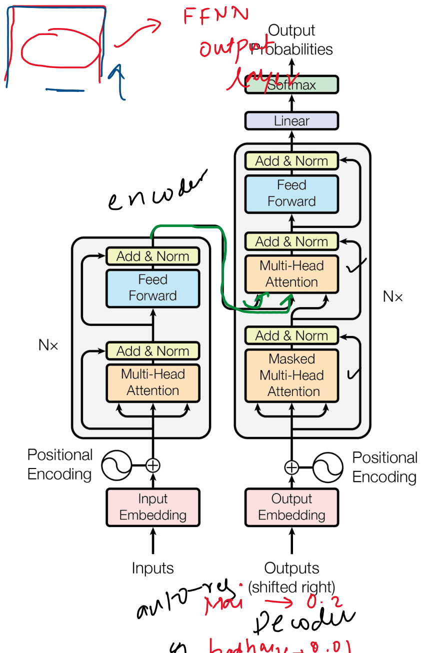
Plan of Attack

22 August 2024 15:36

FFNN

output layer

encoder



auto-reg

Mai → 0.2

Decoder

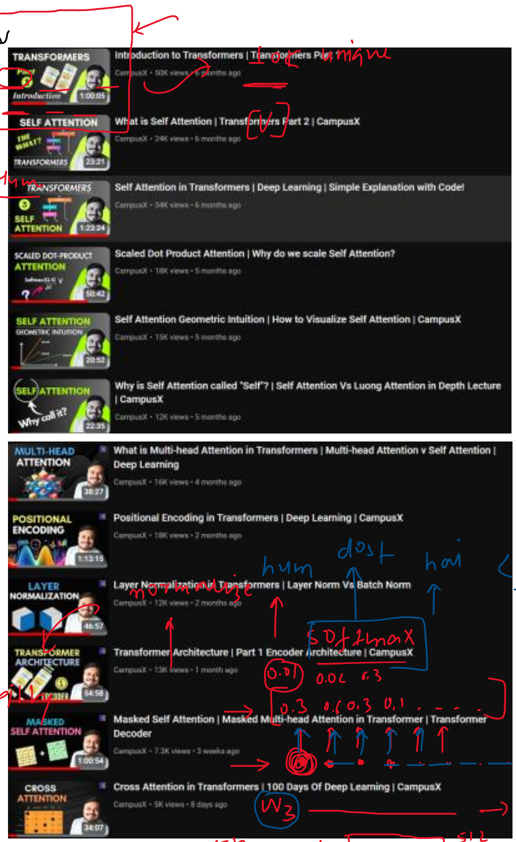
badhaye → 0.01

[Training not Inference]

hum → 0.25

Dost → 0.3

5000



hum dost hai <end>

normalize

softmax

0.01 0.02 0.3

0.3 0.1 0.3 0.1 . . .

probability

→ neurons ? → [V] = 1000

W_3

→ weights

512 x 10000

[10000]

W_3 Y final norm

1×512 $\quad 512 \times 10000$

↓

1×10000

512 dim

(1×512)

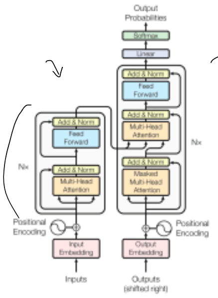
512

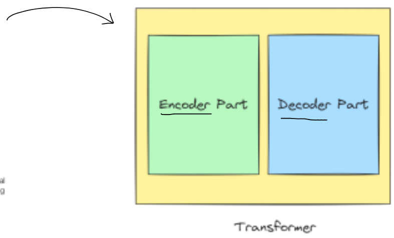
4 x 512

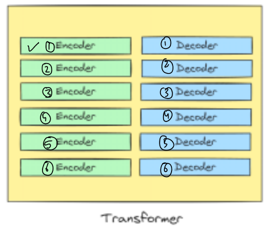
<start> hum dost hai

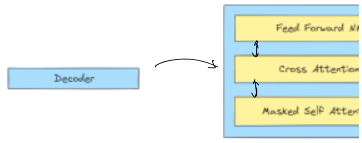
Simplified View

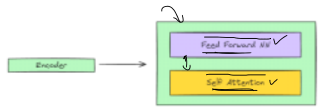
22 August 2024 00:33

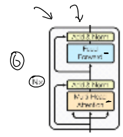


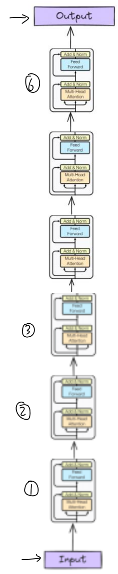




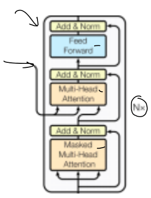


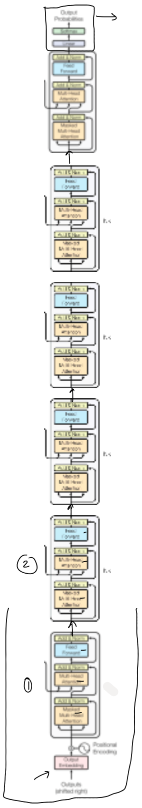







tion

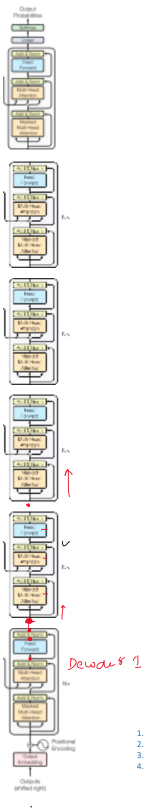

Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Add & Norm
Masked Multi-Head Attention


Output Probabilities
2
1
Positional Encoding
Output Embedding
Outputs (shifted right)

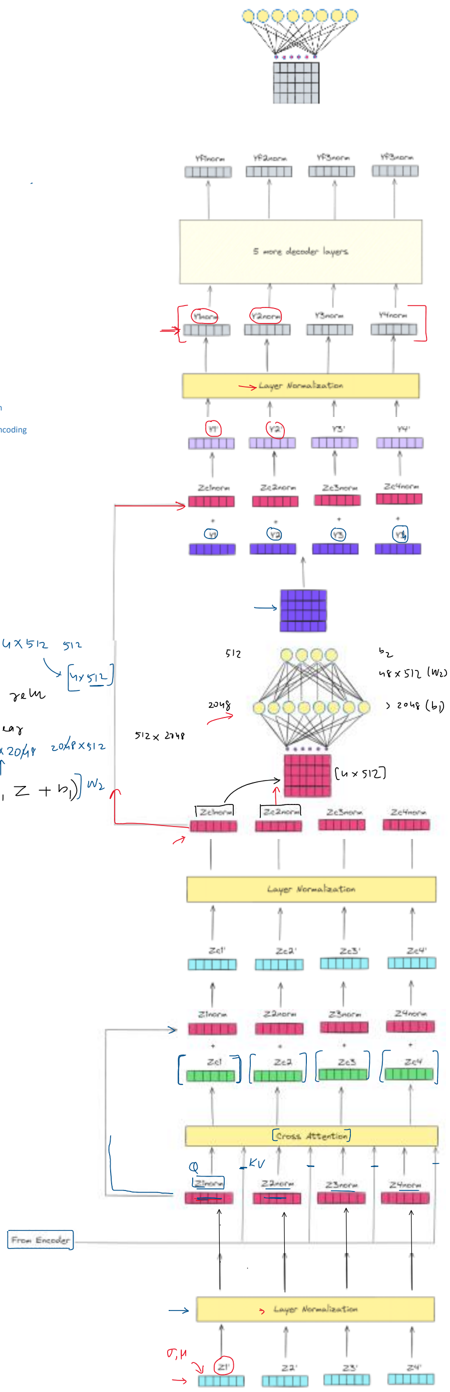
Decoder Architecture

22 August 2024 00:56

Eng	Hindi
I am good	Mai badhiya hu
We are friends	Hum dost hai



1. Shifting
2. Tokenization
3. Embedding
4. Positional Encoding



2 layers

2048 neurons → relu

512 neurons → linear

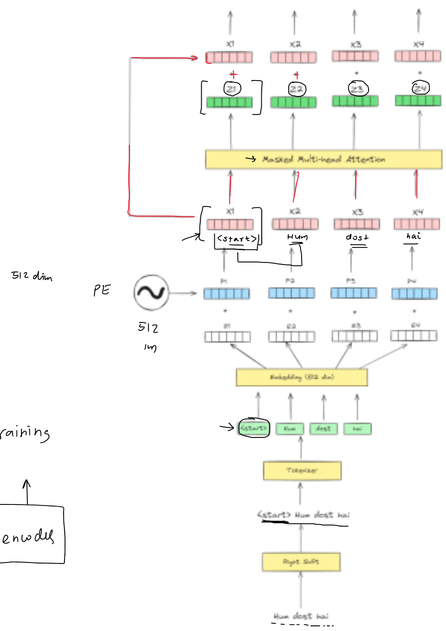
4 x 512 512

$\boxed{4 \times \underline{512}}$

4 x 2048 2048 x 512

$\underbrace{4 \times 512 \quad 512 \times 2048}$ $[relu(W_1 Z + b_1)] W_2$

$\boxed{4 \times 2048}$ + 2048



Machine Translation Task

English Sentence - We are friends

Hindi Sentence - Hum dost hai

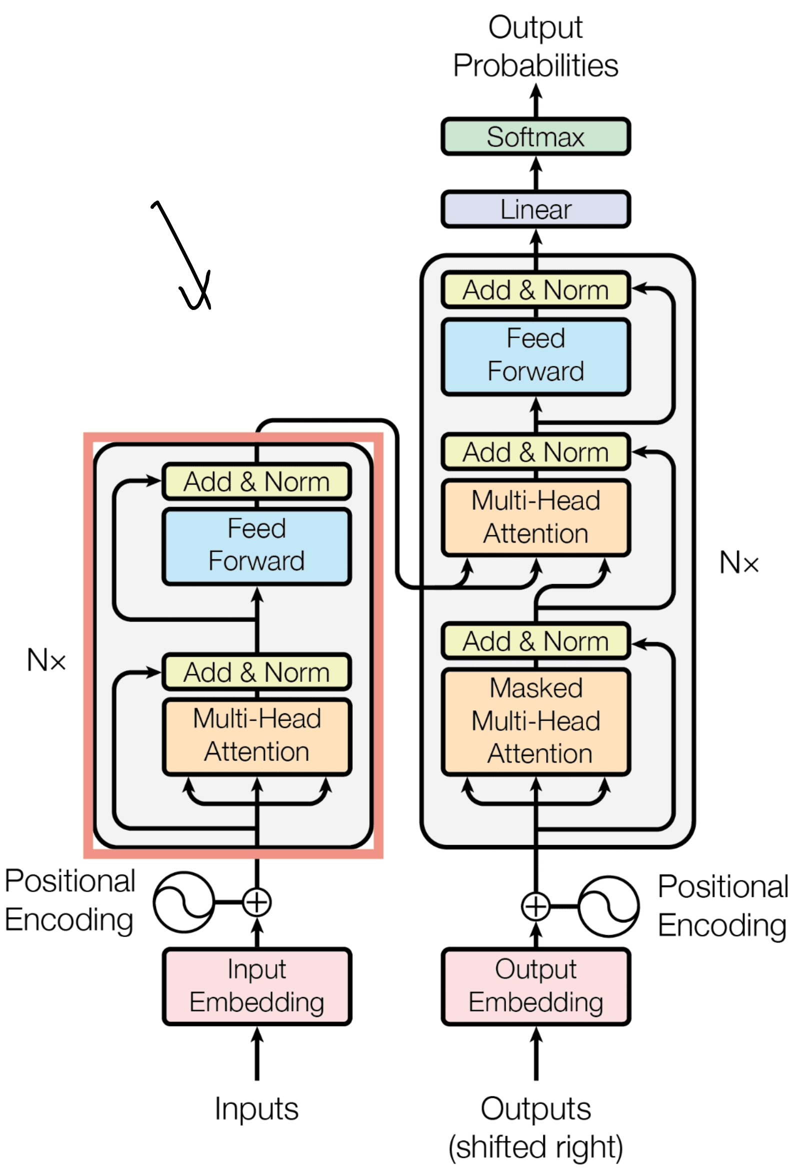
Training

encoder

Plan of Attack

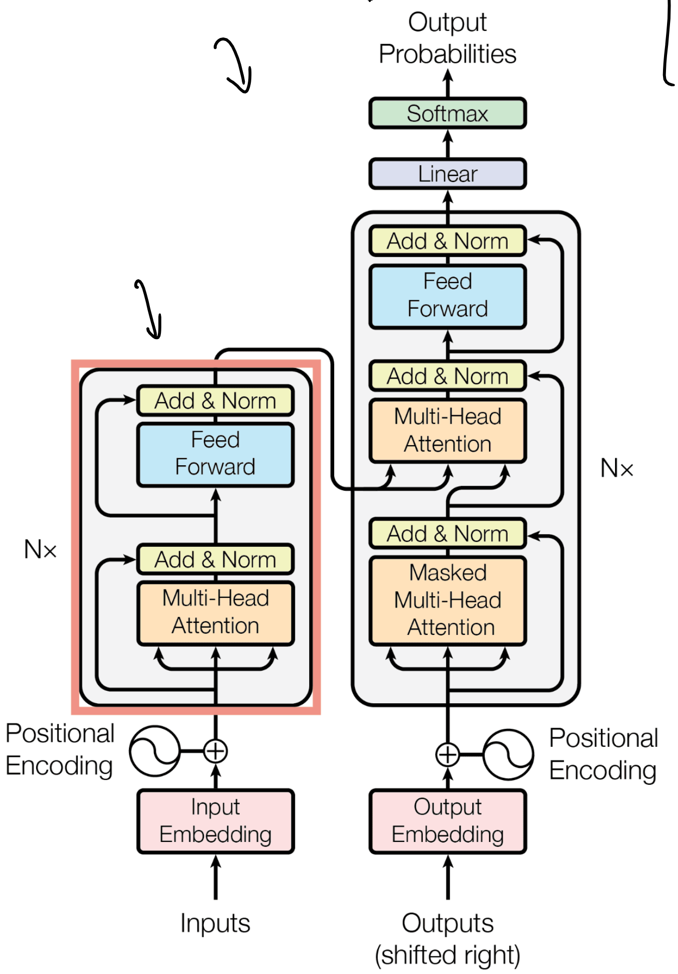
03 September 2024 00:37

training



The Setup

03 September 2024 00:39



Data

S no	English Sentence	Hindi Sentence
1	How are you	Aap kaise ho
2	Thank you	Dhanyawad
3	You are welcome	Apka swagat hai

Query Sentence

We are friends → ?

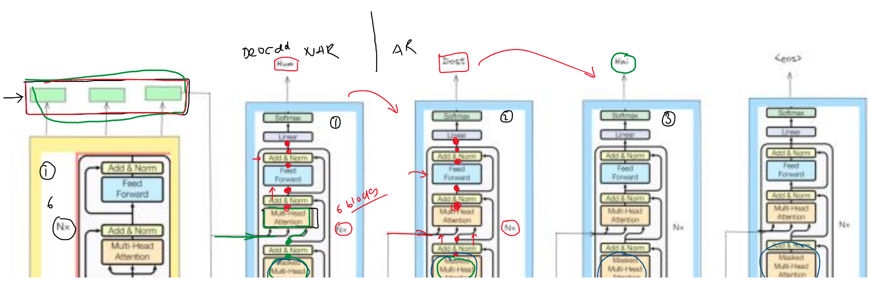
Transformer during Inference

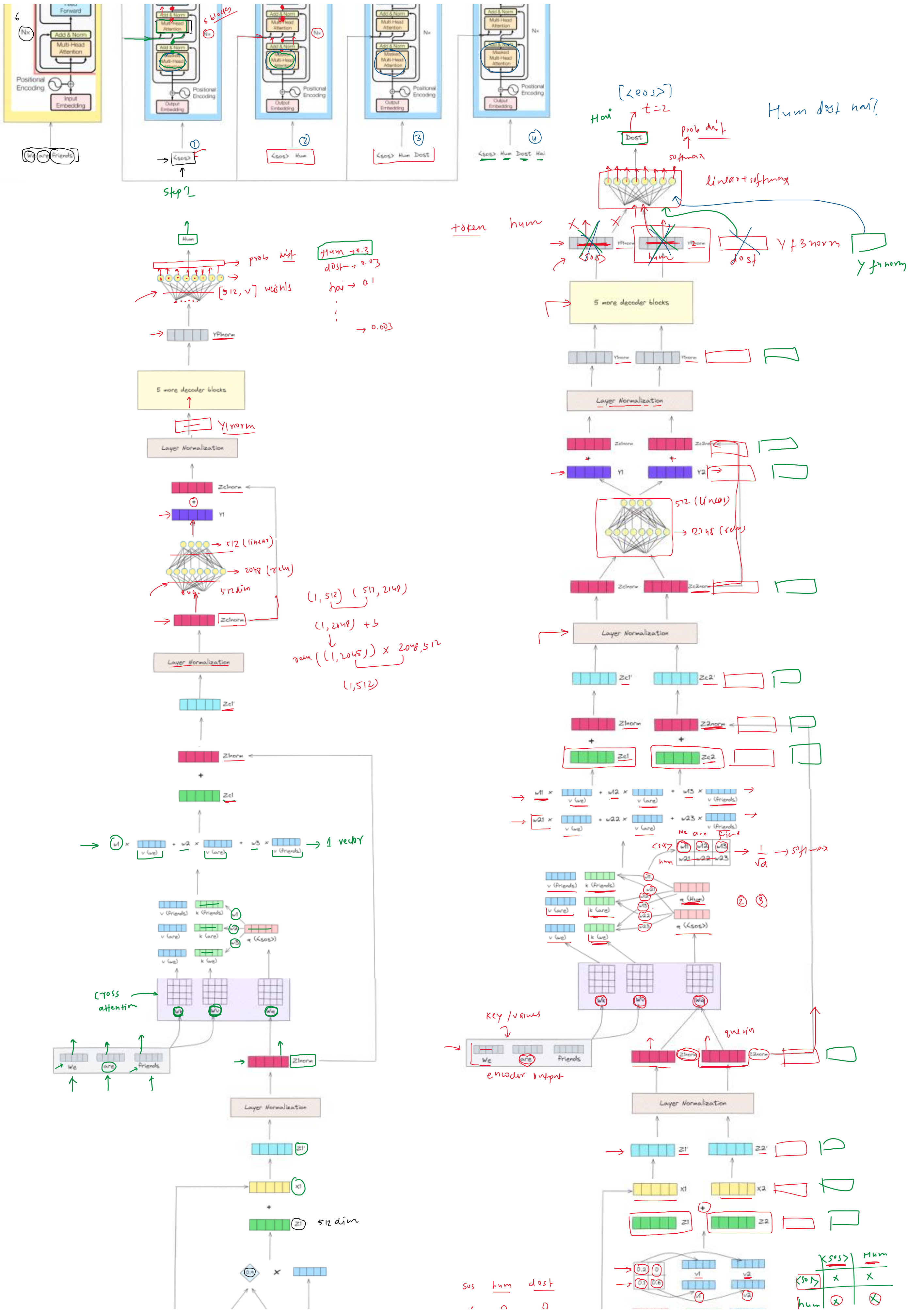
03 September 2024 00:50

Training | Inference

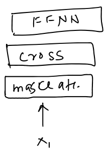
Encoder same same

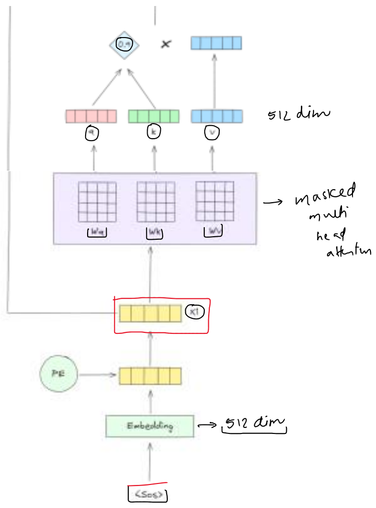
R





6 blocks
Step 1
prob dist
[512, v] weights
Hum → 0.3
dost → 0.03
hai → 0.1
5 more decoder blocks
Layer Normalization
512 (linear)
2048 (relu)
512 dim
cross attention
1 vector
Hum dost hai?
linear + softmax
softmax
token
hum
key / values
encoder output
query in





	sos	hum	dost
sos	X	O	O
hum	X	X	O
dost	X	X	X