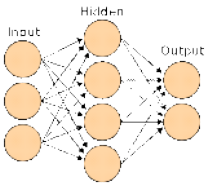
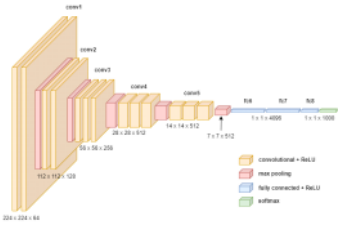
What is Transformer?

27 January 2024 18:41

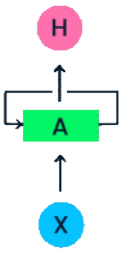
→ self attention scalable



ANN
tabular



CNN
Image

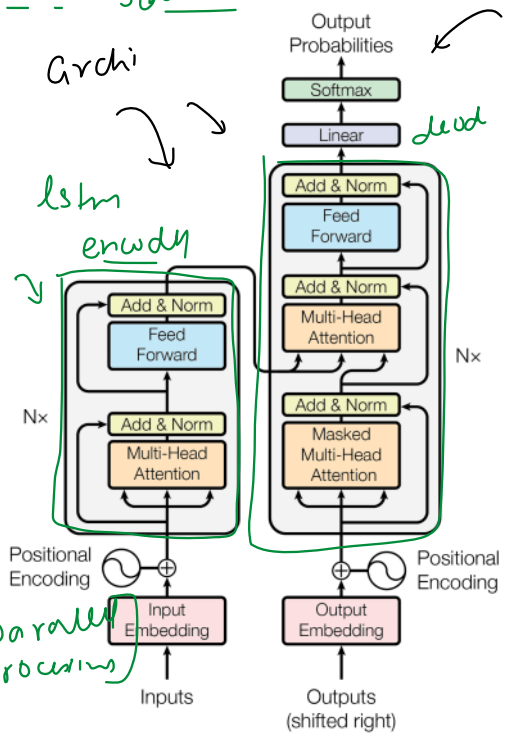


RNN
sequence
↳ text

Archi

lstm

encoder



decoder

→ [transformer] → seq2seq task

NN arch

- machine translation
- question ans
- text summary

[parallel processing]

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

↳ Google Brain

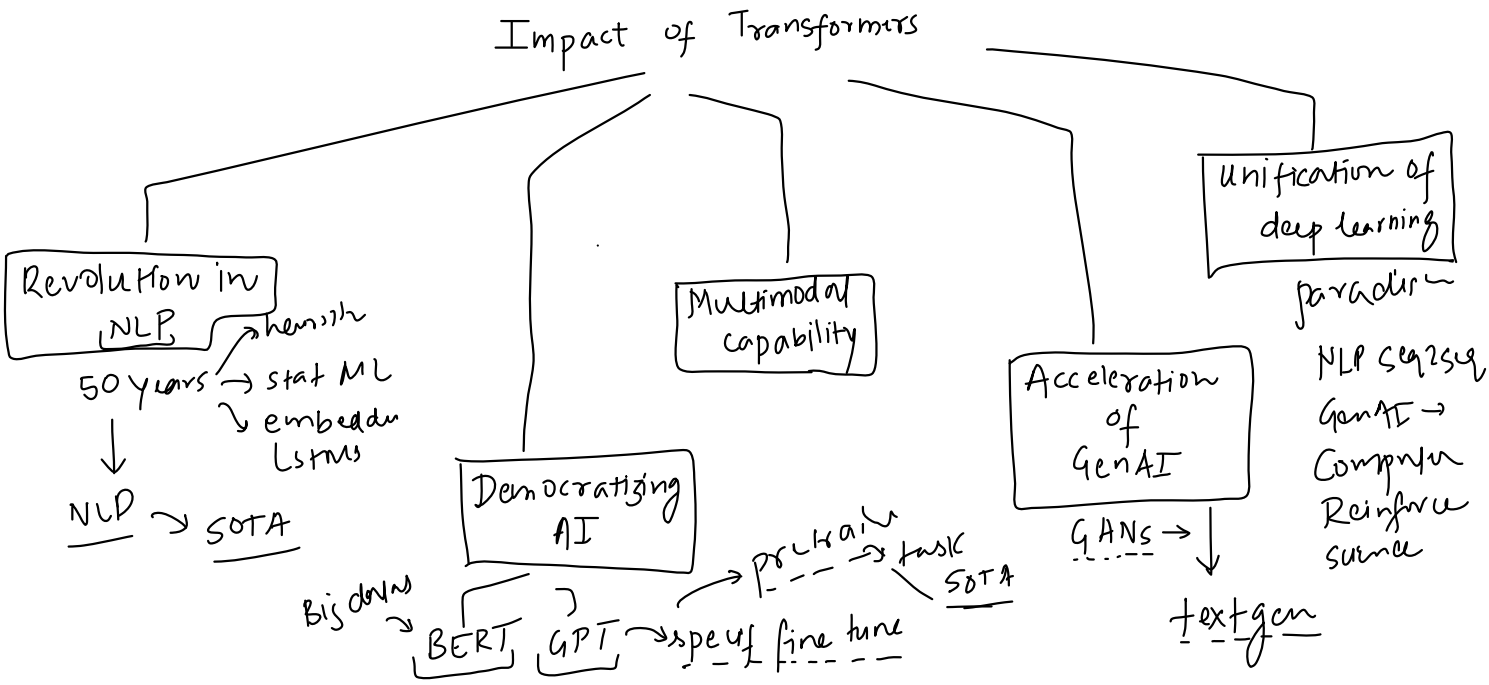
Deep learning

Abstract

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

Impact of Transformers

27 January 2024 20:03



The Origin Story!

27 January 2024 22:38

2014-15 → seq2seq machine

Sequence to Sequence Learning with Neural Networks

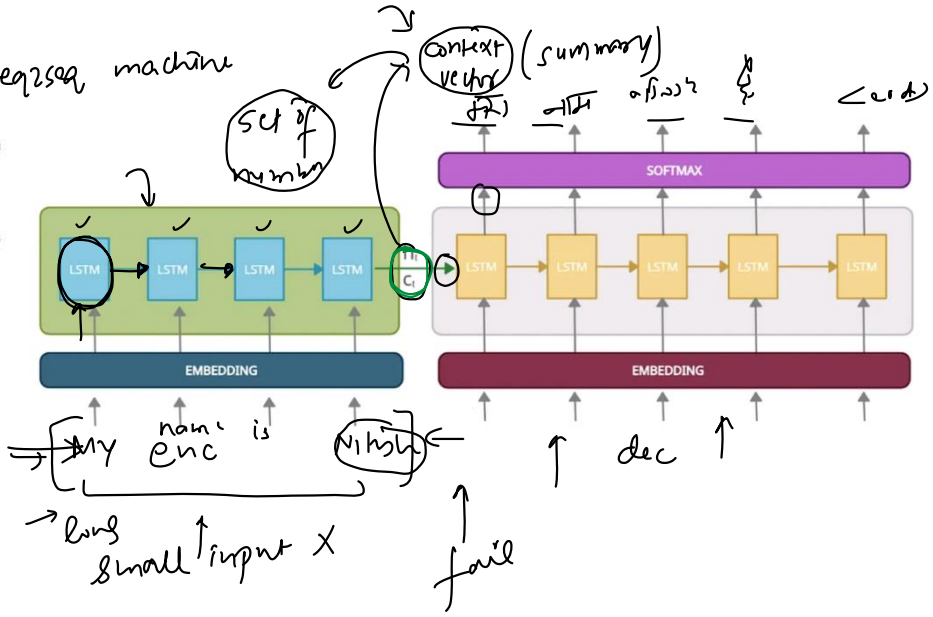
Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a 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

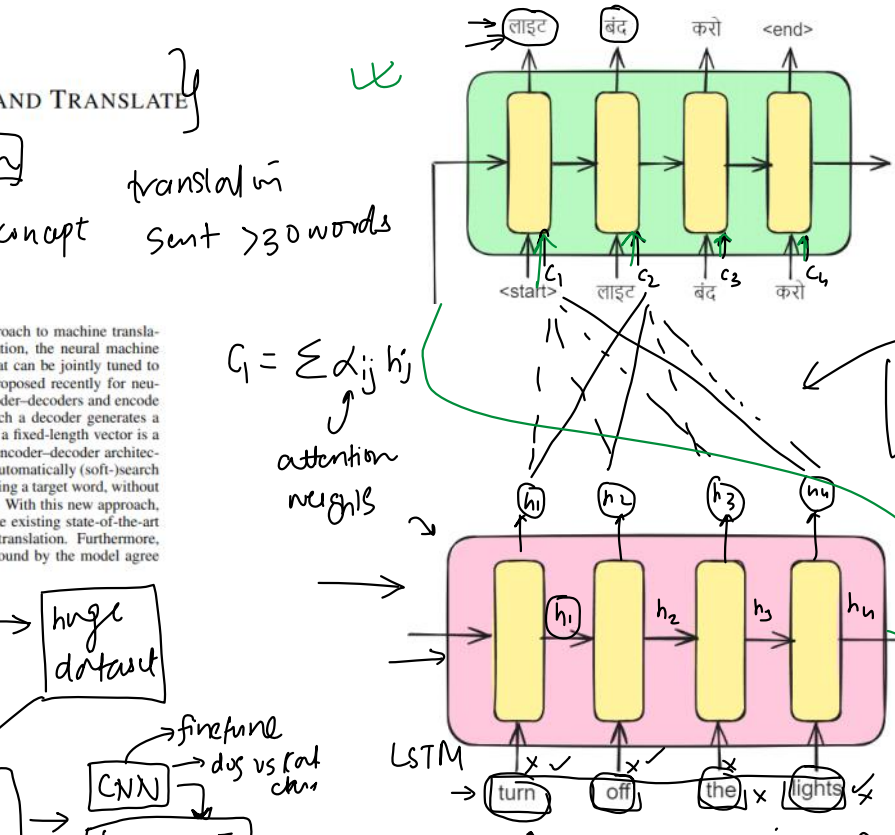
KyungHyun Cho Yoshua Bengio*
Université de Montréal

→ attention → concept

translation sent >30 words

ABSTRACT

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



sequential → slow → huge dataset

training

Transfer learning → CNN → finetune → dog vs cat → ImageNET

research → incremental time travel → vanilla → parallely → stable → hyperfine

NLP → BERT, GPT → fine tune

Attention Is All You Need → 2017

Output Probabilities

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

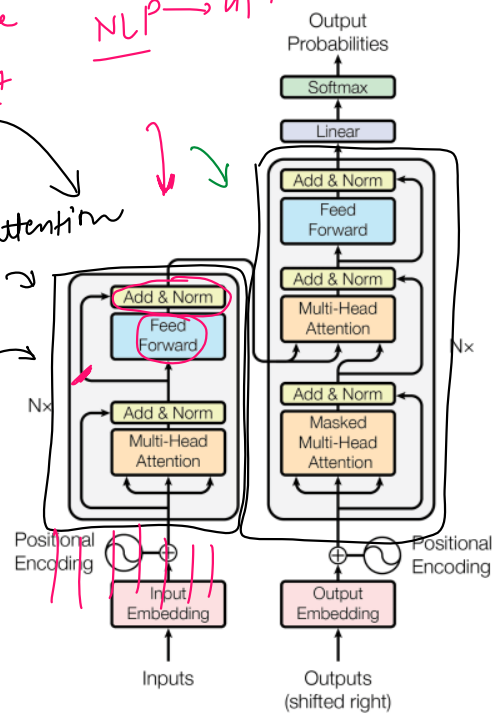
Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

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.



The Timeline

28 January 2024 00:55

2000 - 2014 → RNNs / LSTMs

2014 → Attention

2017 → Transformer

2018 → BERT / GPT (Transfer learning)

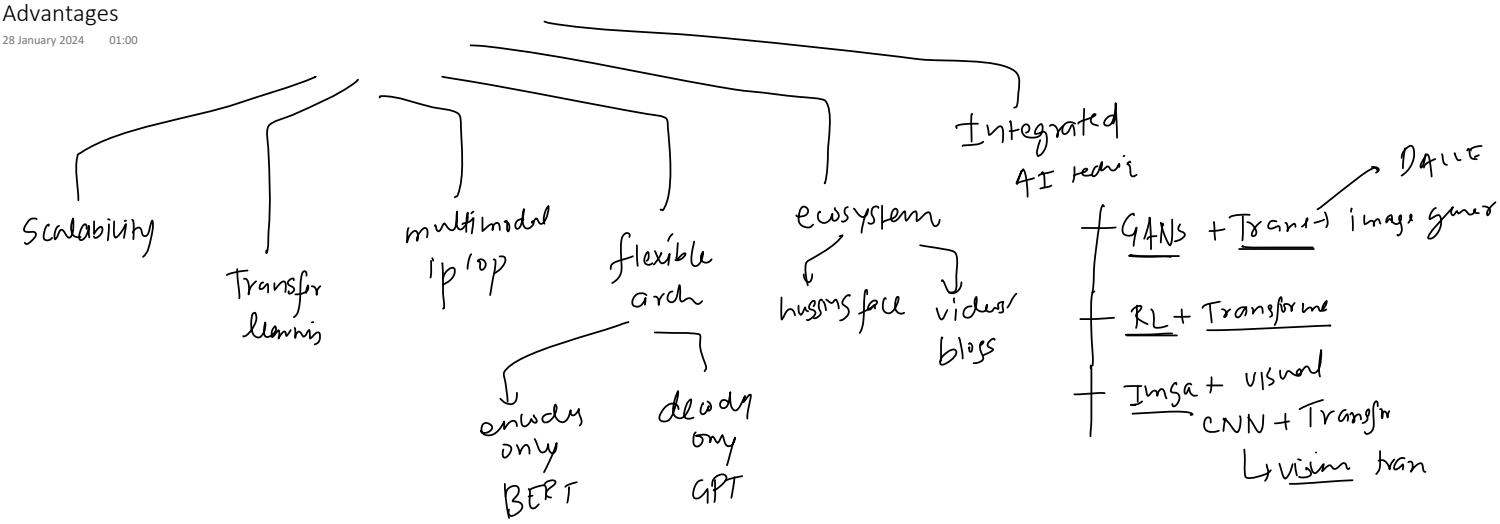
2018 - 2020 → Vision Transformer / Alphafold-2

2021 → Gen AI

2022 → ChatGPT / Stable Diffusion

Advantages

28 January 2024 01:00

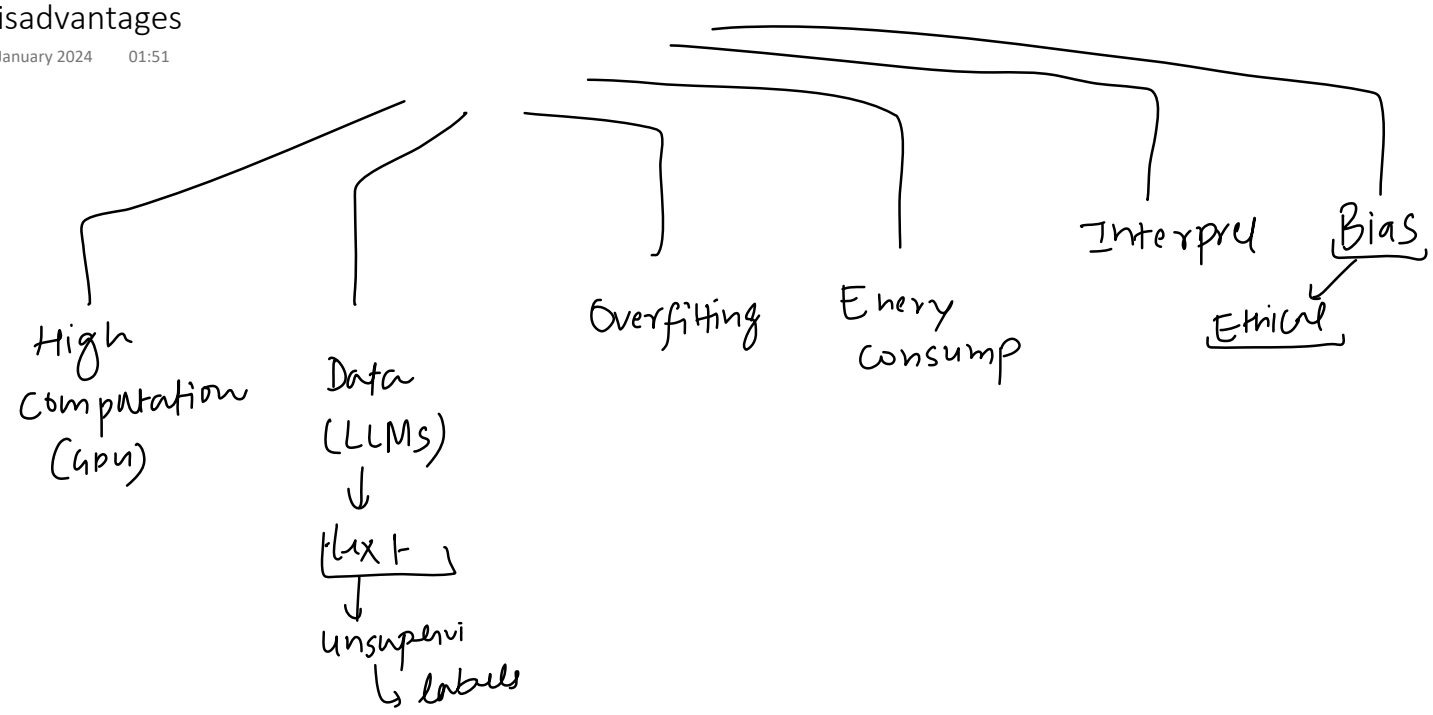


Famous Applications

28 January 2024 01:17

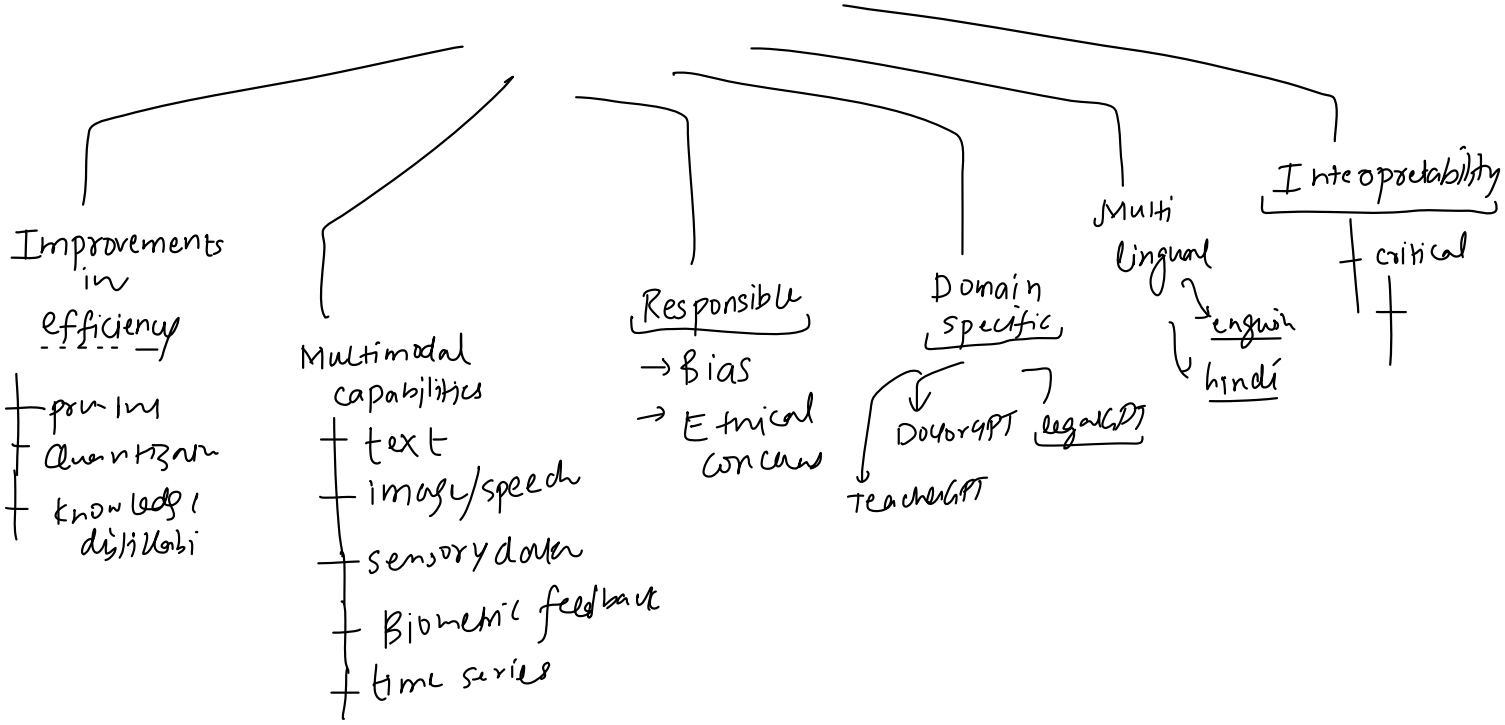
Disadvantages

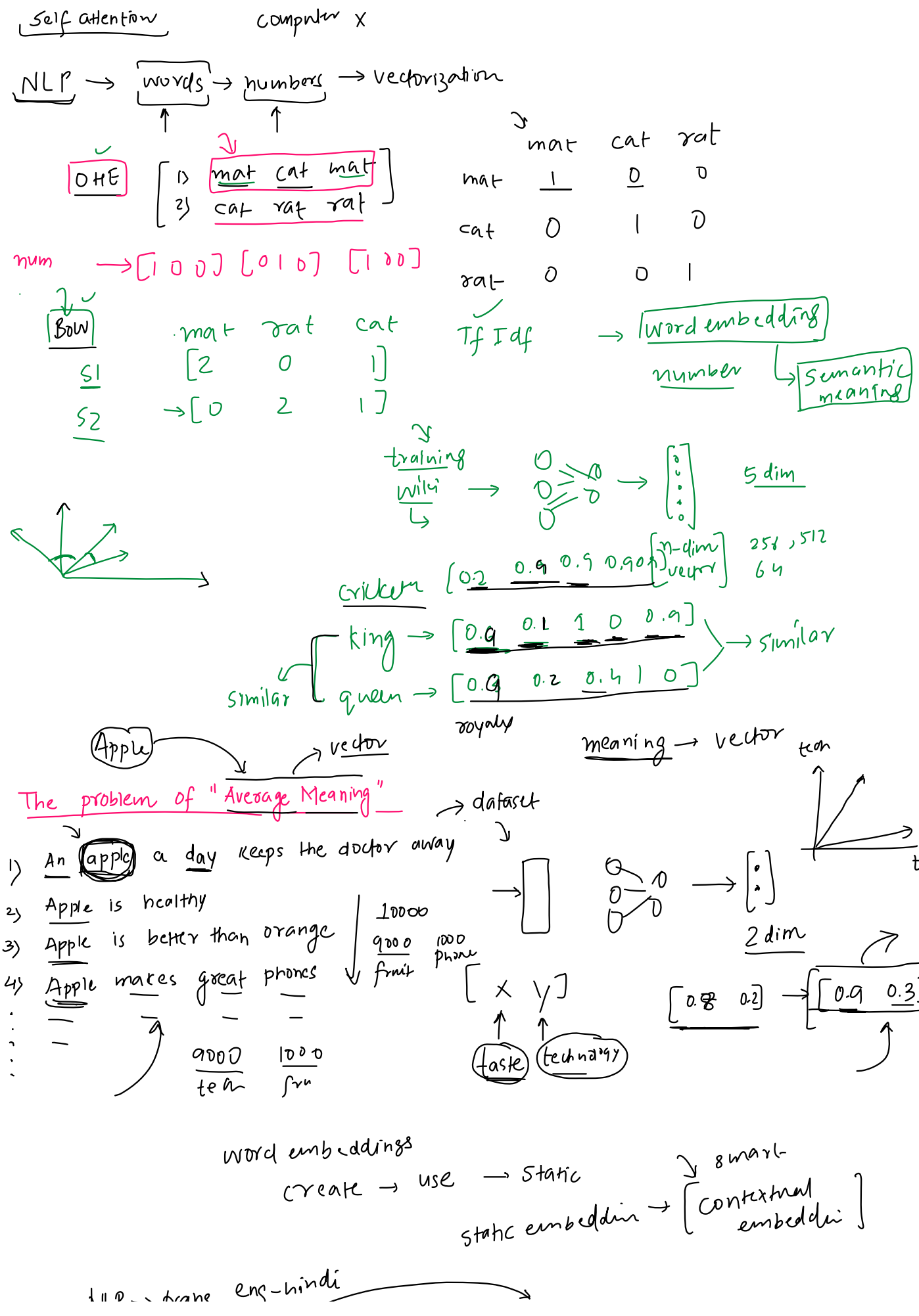
28 January 2024 01:51

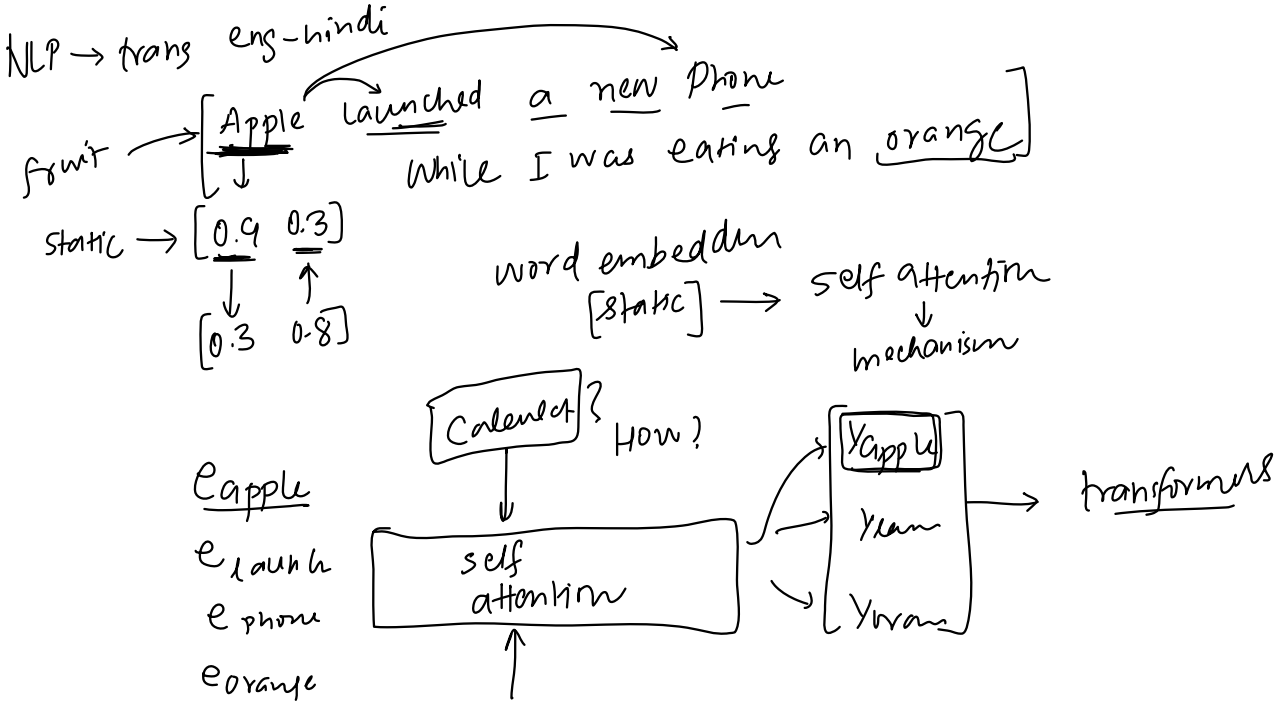


Future

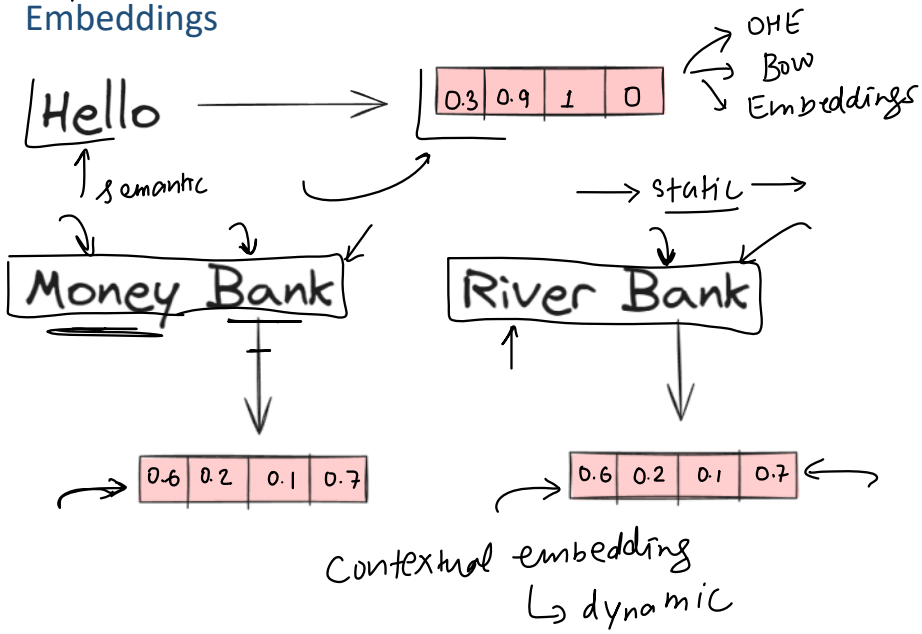
28 January 2024 02:09

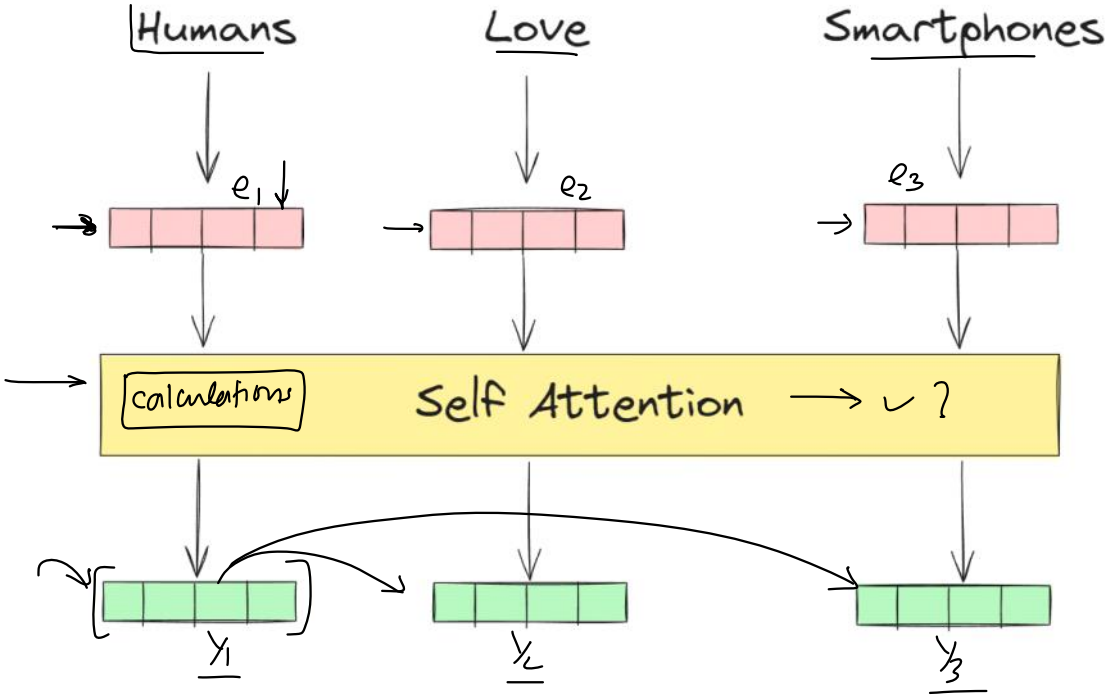



Self attention
computer X
NLP → words → numbers → vectorization
OHE
1) mat cat mat
2) cat rat rat
num → [1 0 0] [0 1 0] [1 0 0]
mat cat rat
mat 1 0 0
cat 0 1 0
rat 0 0 1
BOW
mat rat cat
S1 [2 0 1]
S2 → [0 2 1]
Tf Idf → word embedding
number → semantic meaning
training
wiki
5 dim
n-dim vector
258, 512
64
Cricket [0.2 0.9 0.5 0.90]
king → [0.9 0.1 1 0 0.9]
queen → [0.9 0.2 0.4 1 0]
similar
similar
royalty
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
9000 tea
1000 fruit
10000
9000 fruit
1000 phone
[x y]
taste
technology
2 dim
[0.8 0.2] → [0.9 0.3]
word embeddings
create → use → static
static embeddin → 8 mark- contextual embeddi
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
Calendar? How?
eapple
elaunch
ephone
eorange
self attention
yapple
yfruit
yorange
transformers

Embeddings


Hello
0.3 0.9 1 0
OHE
Bow
Embeddings
semantic
static
Money Bank
River Bank
0.6 0.2 0.1 0.7
0.6 0.2 0.1 0.7
Contextual embedding
↳ dynamic
Self Attention
embedding → dynamic contextual


Humans
Love
Smartphones
e1
e2
e3
calculations
Self Attention
y1
y2
y3

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

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

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

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

similarity

e_{money} e_{money}

e_{money} e_{bank}

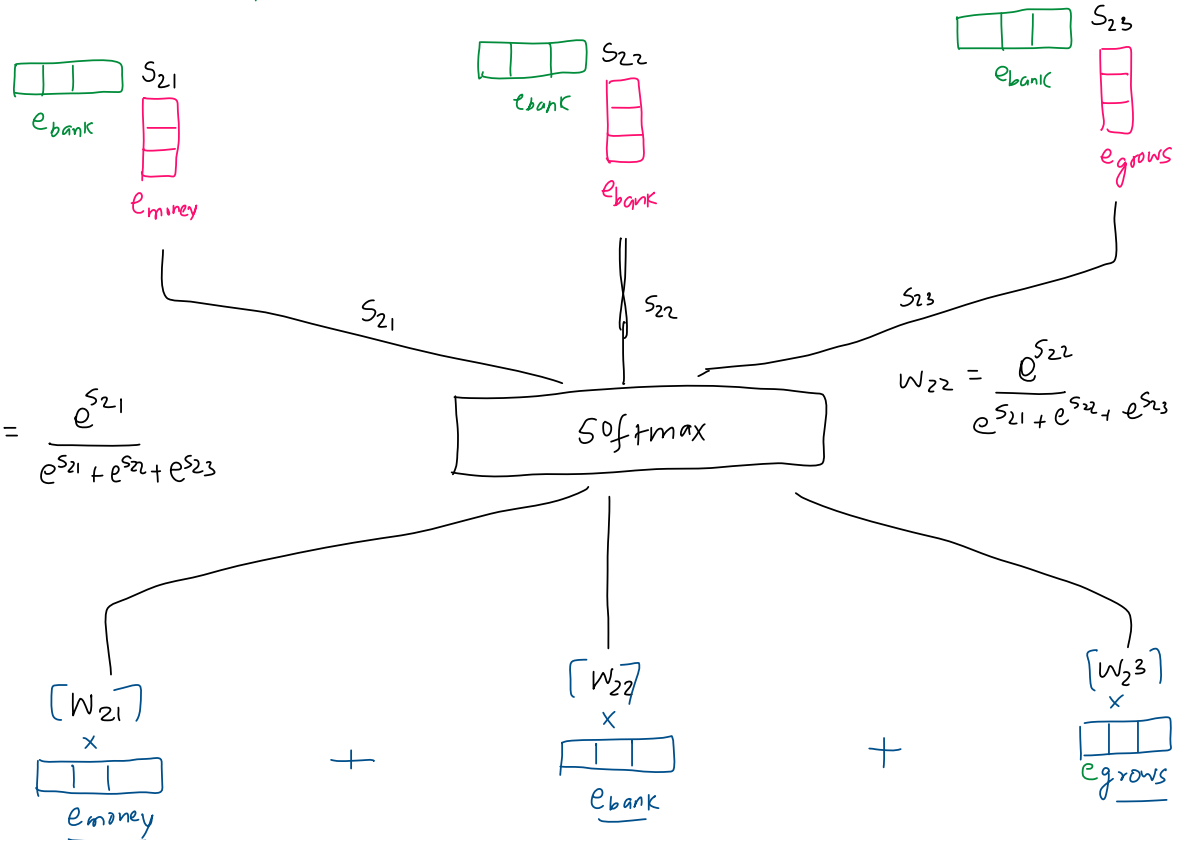
e_{money} e_{grow}

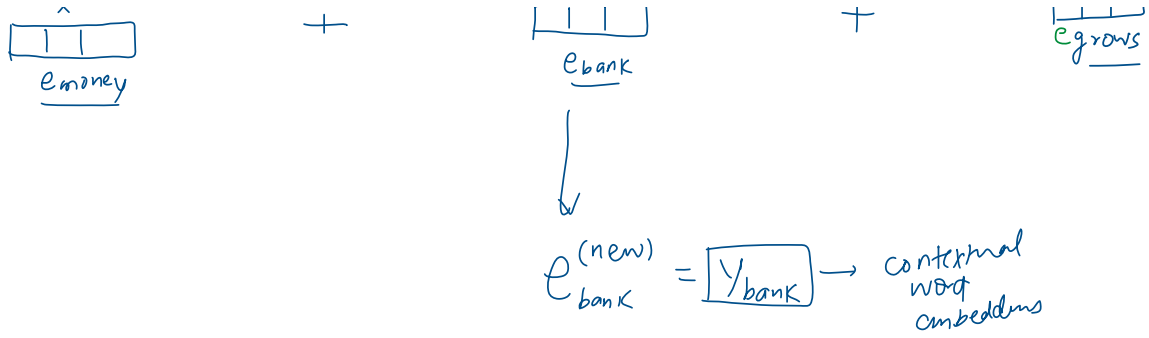
e_{grows} e_{money}

dot products

(1) normalised

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

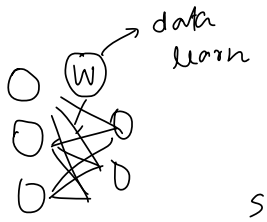




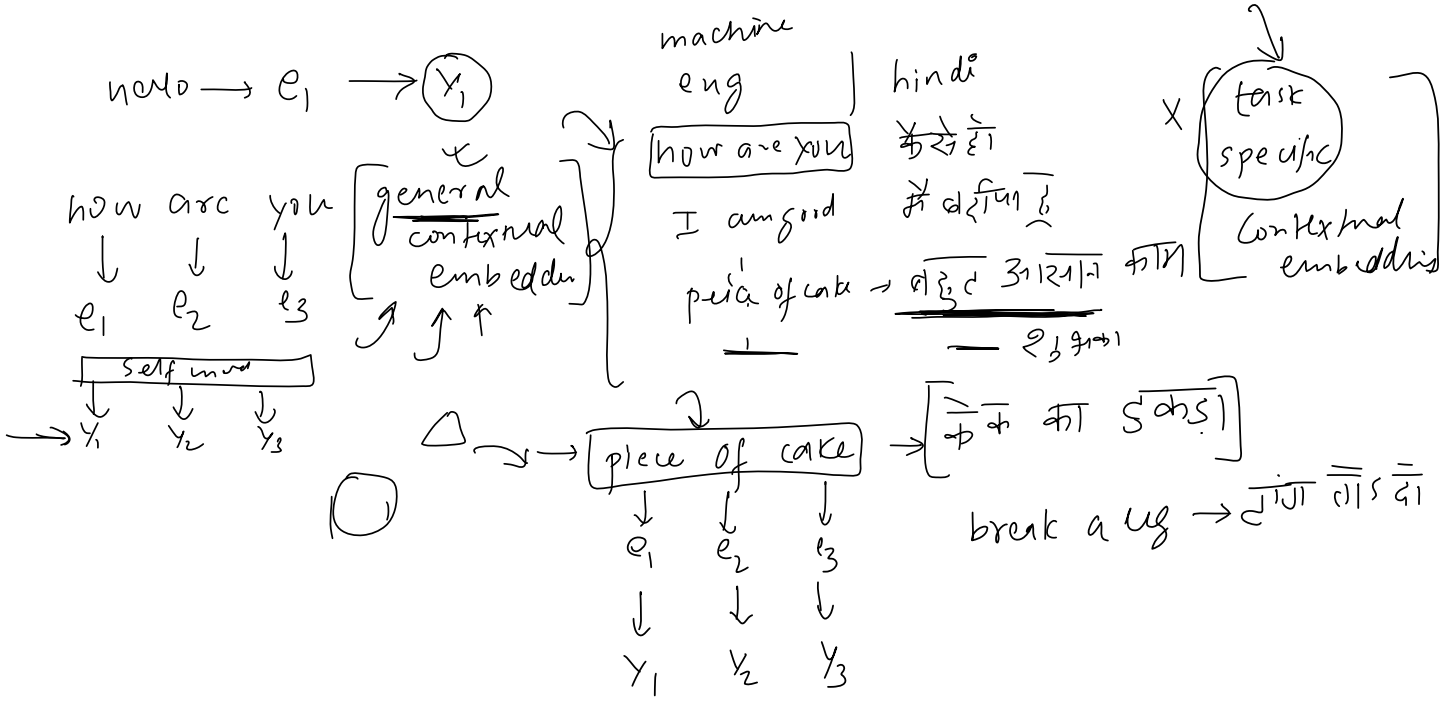
Points to consider

→ This operation is a parallel operation

→ There are no parameters involved

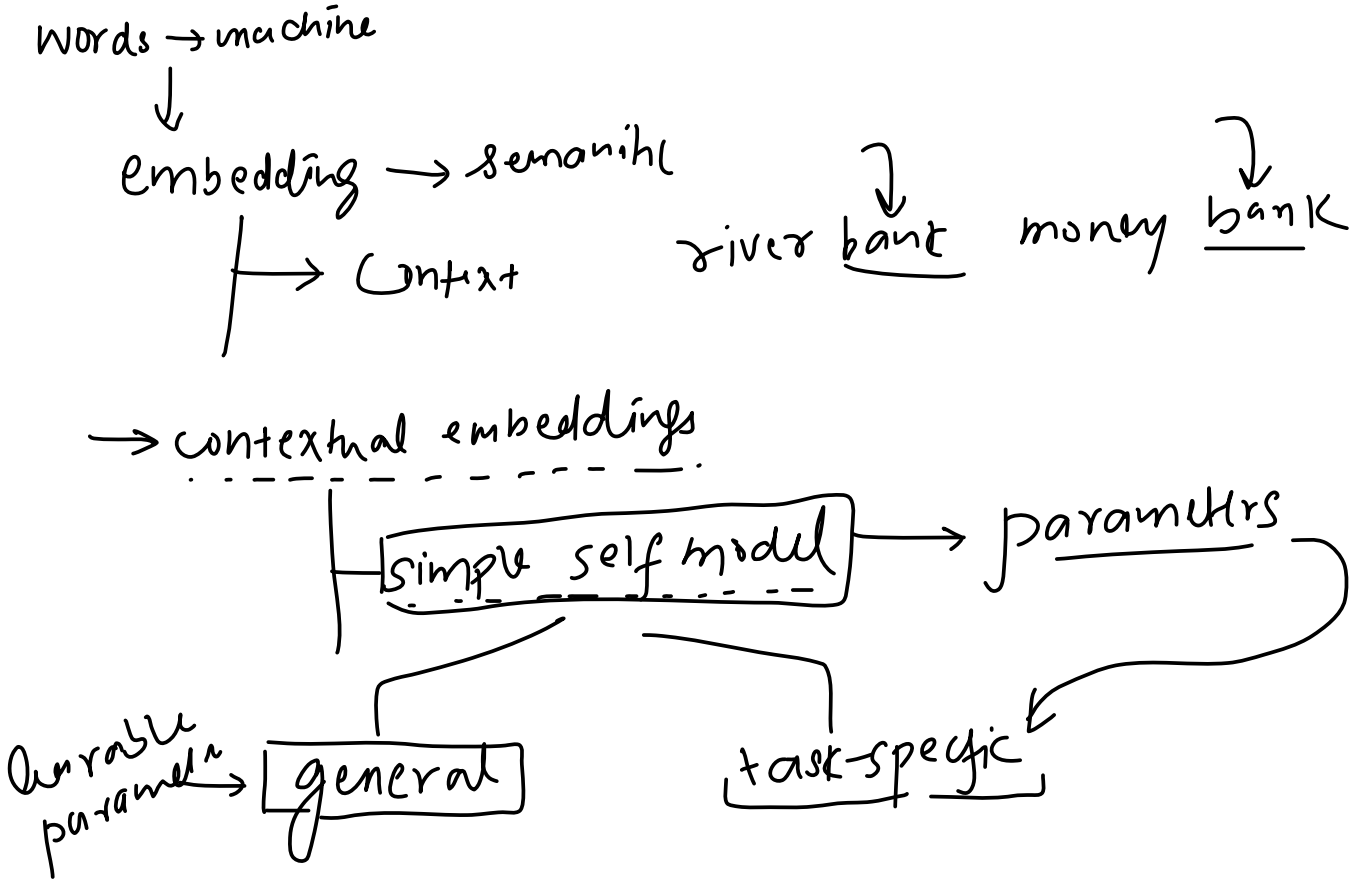


stuf model



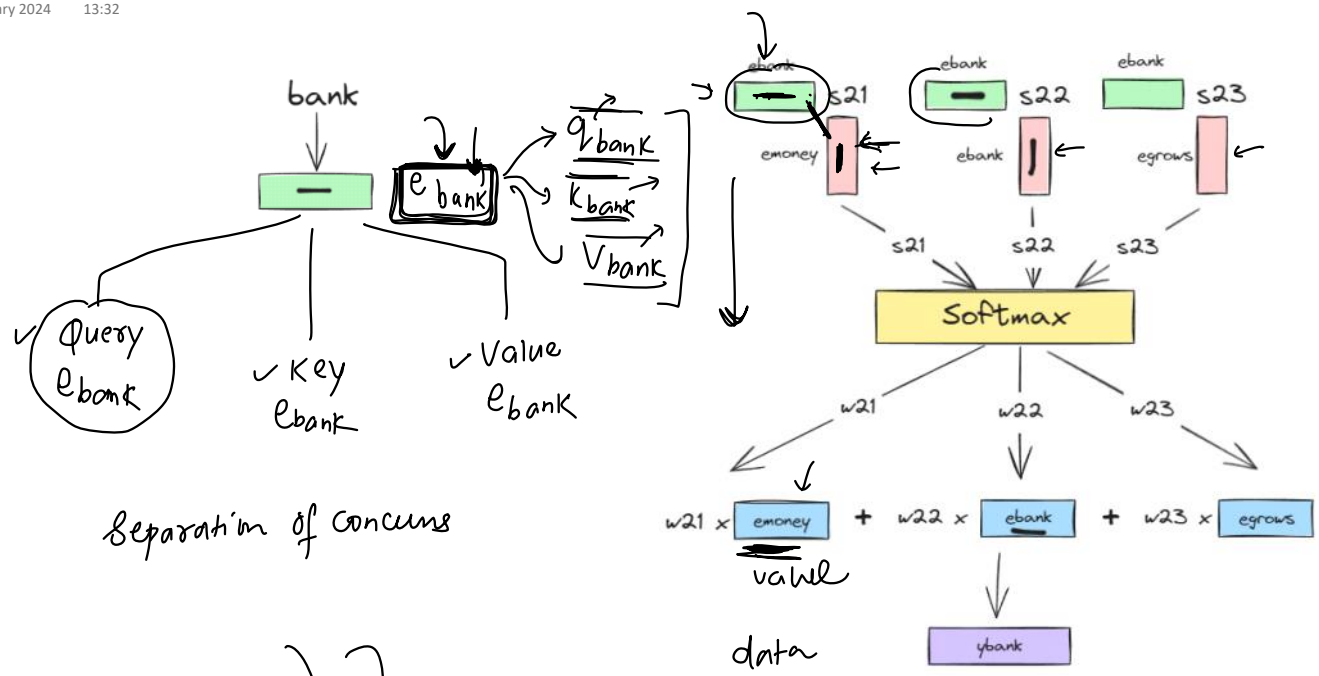
Progress

06 February 2024 00:42

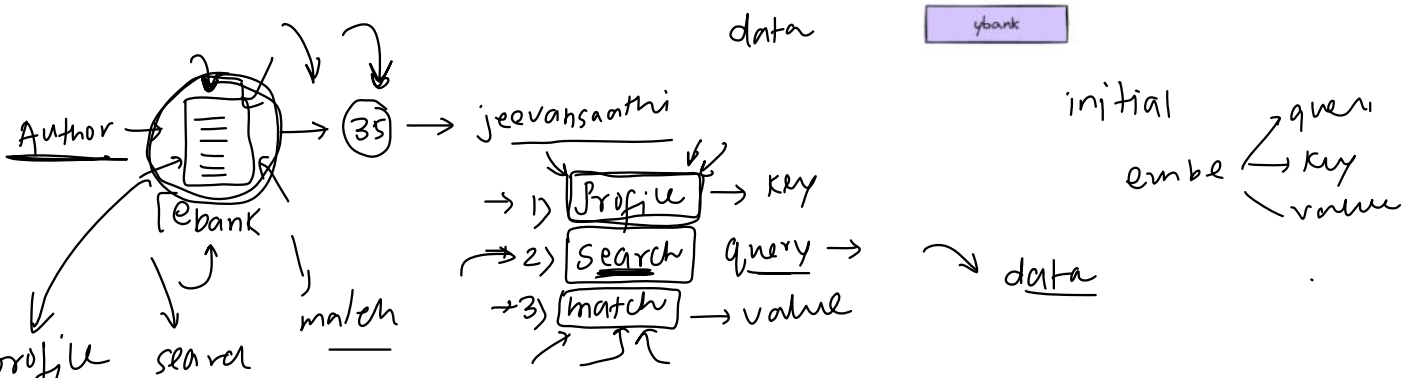


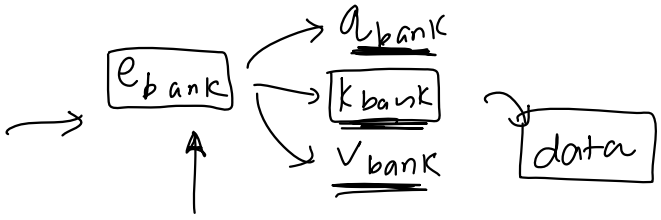
Query, Key & Value Vectors

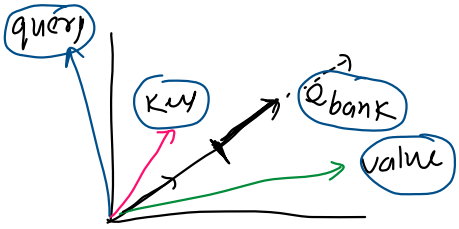
06 February 2024 13:32



Separation of concerns



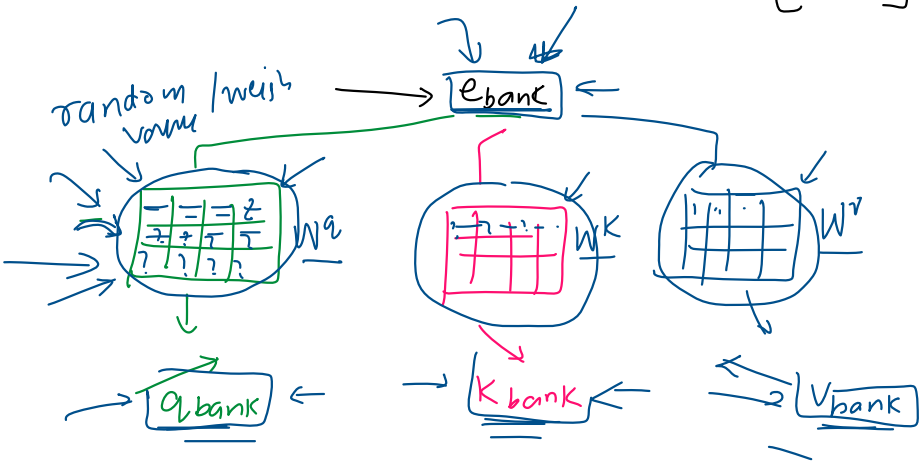




d magnitude (scaling)

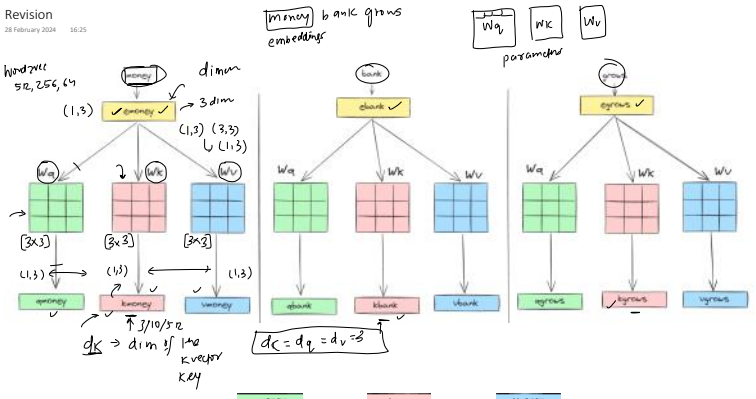
→ linear transform

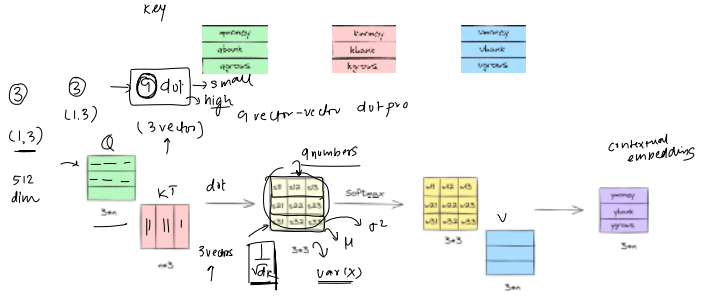
→ [ ]

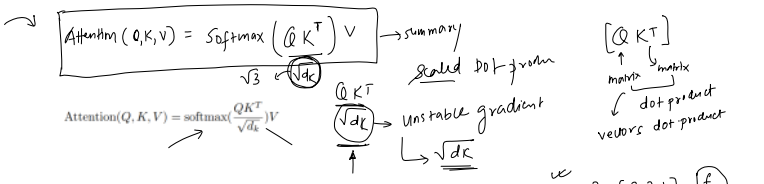


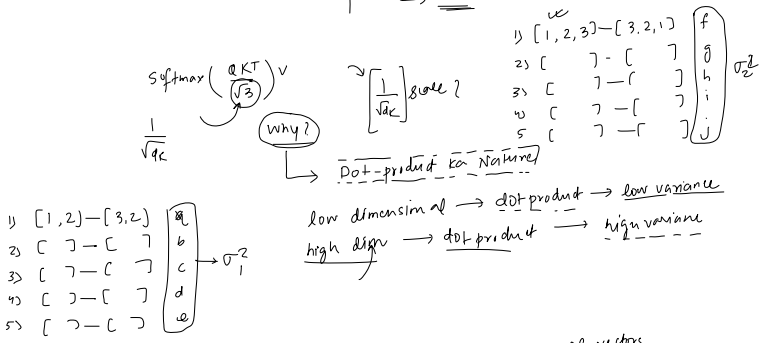
3 x 3 → dot → (9) values

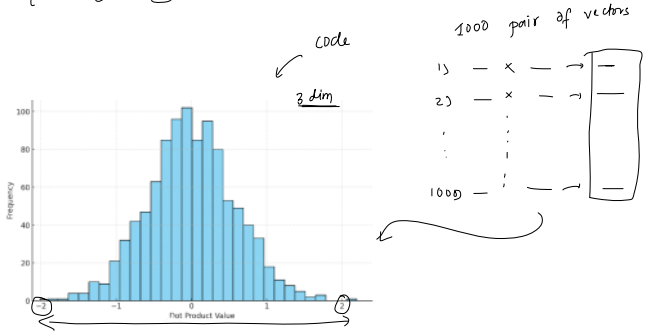
Revision
28 February 2024 16:25

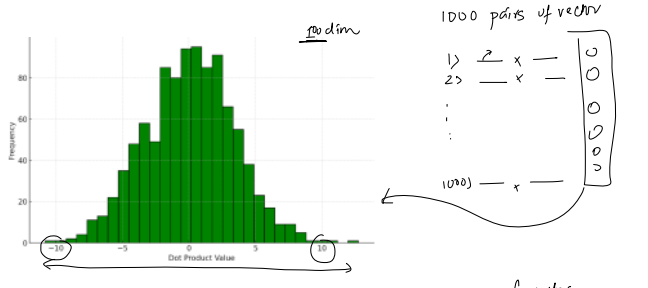


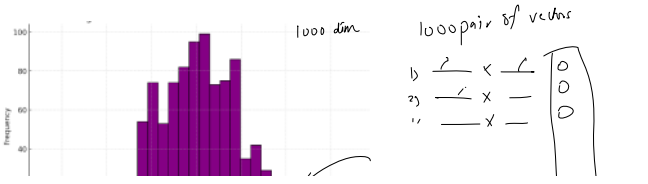


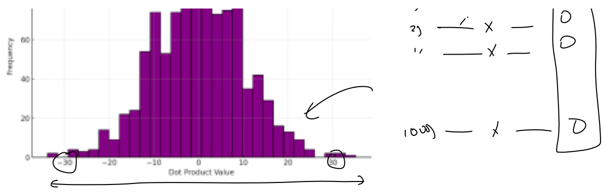


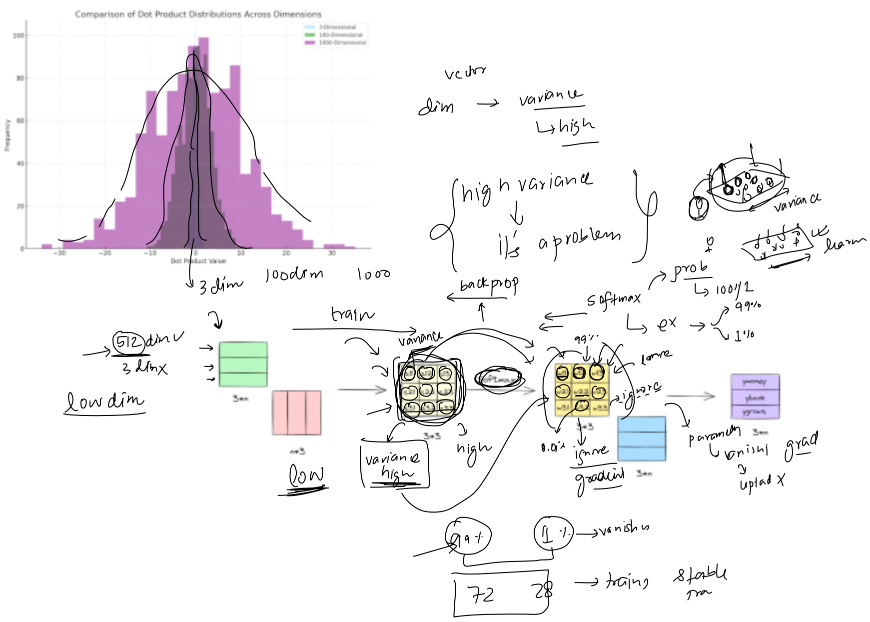


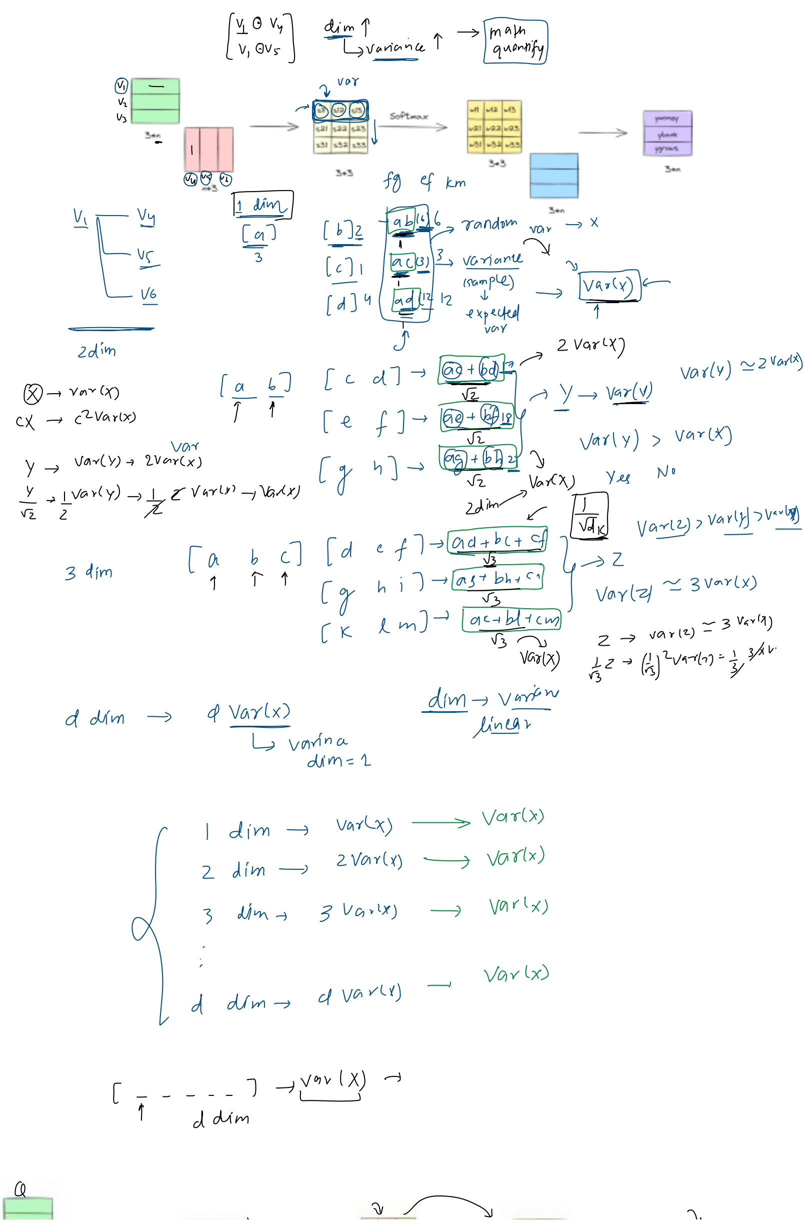


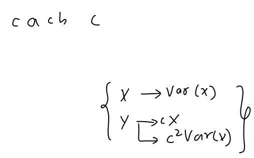






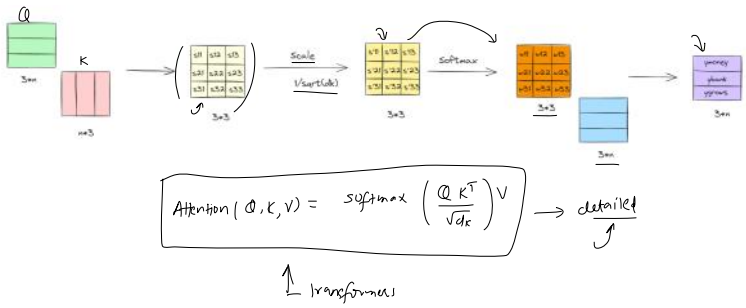






If you have a random variable X with a variance of $\mathrm{Var}(X)$, and you create a new variable Y by scaling X with a constant c, so that $Y = cX$, the variance of Y ($\mathrm{Var}(Y)$) is related to the variance of X by the square of the scaling factor c. Mathematically, this relationship is expressed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

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



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

→ detailed

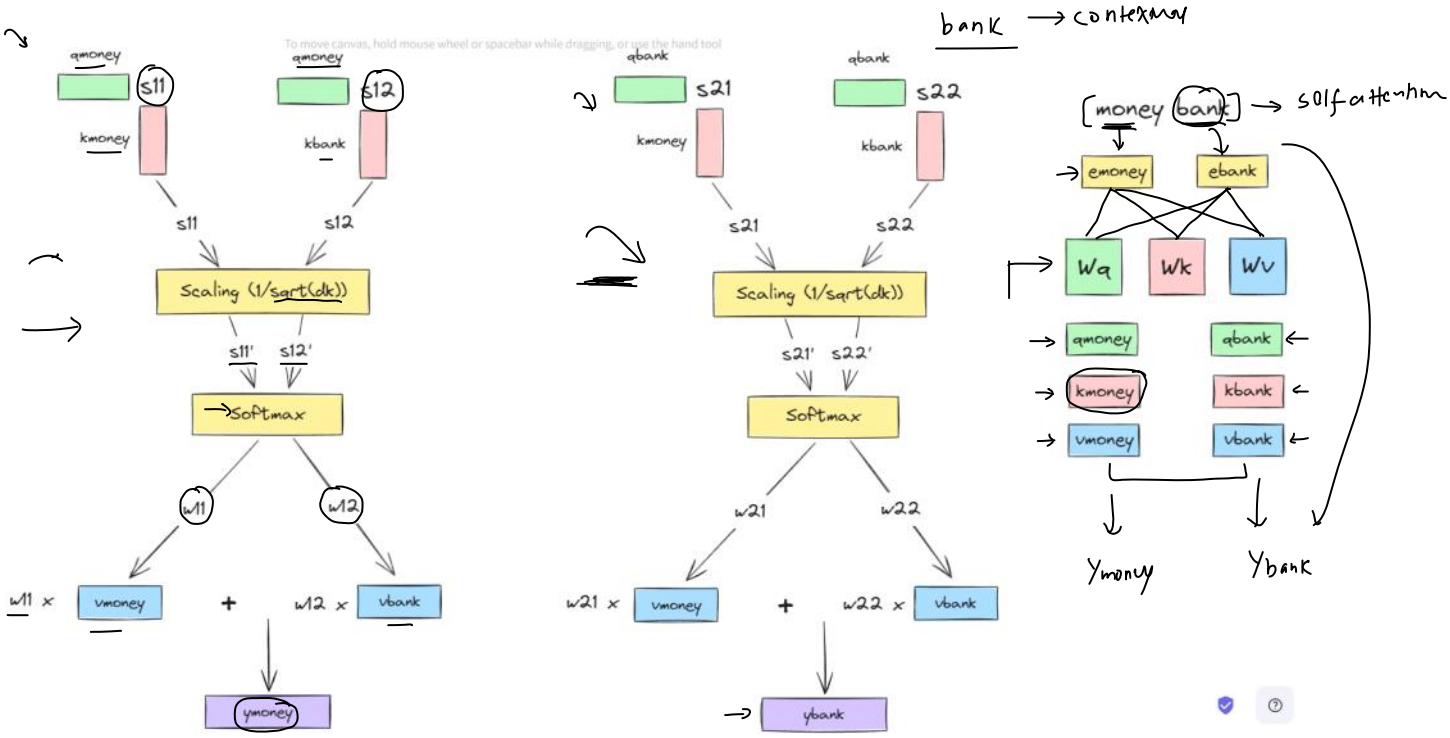
↑ transformers

What is d_k

28 February 2024 16:59

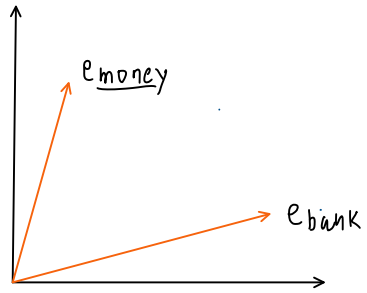
Recep

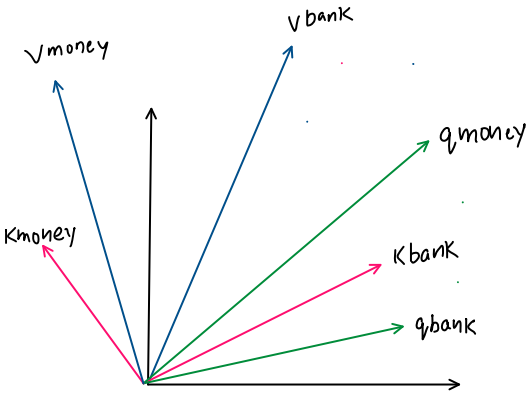
08 March 2024 15:14



Geometric Intuition

08 March 2024 15:16



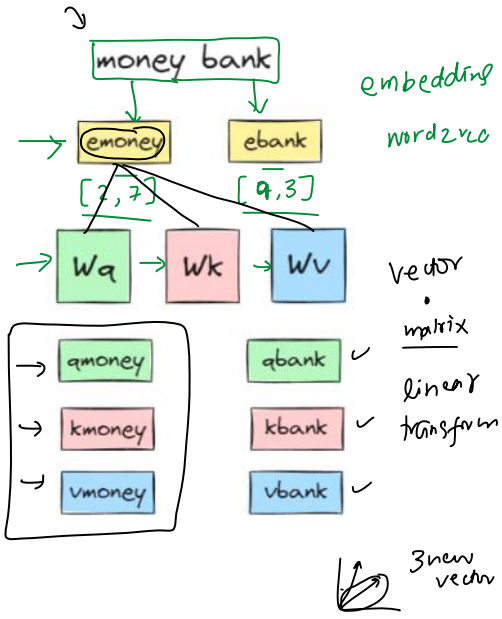


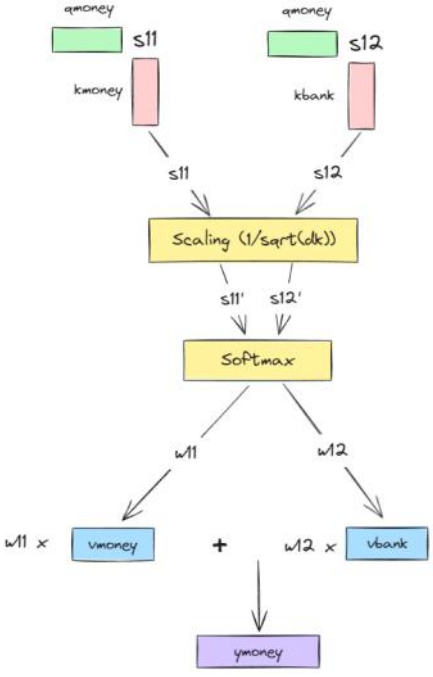
$W_q = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ (2x2)

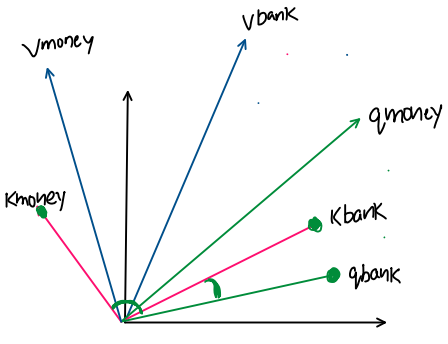
$W_k = \begin{bmatrix} 3 & 4 \\ 5 & 1 \end{bmatrix}$ (2x2)

$W_v = \begin{bmatrix} 4 & 1 \\ 2 & 1 \end{bmatrix}$ (2x2)

* All values are hypothetical







[Dot Product]

$S_{21} = 10$ $\quad S_{22} = \frac{32}{\sqrt{2}}$

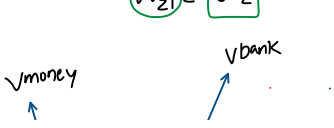
[Scaling]

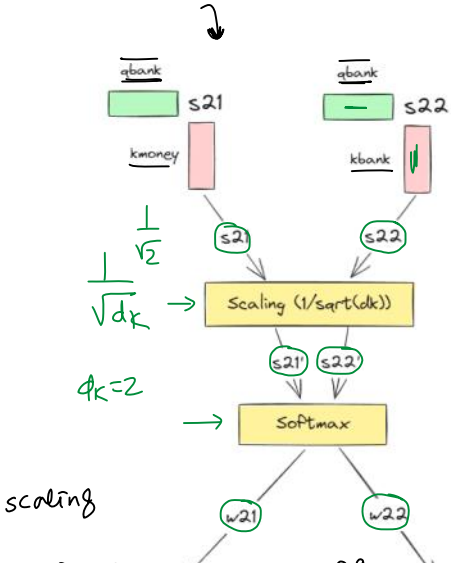
$S'_{21} = \frac{10}{\sqrt{2}} = 7.09$ $\quad S'_{22} = 22.69$

[Softmax]

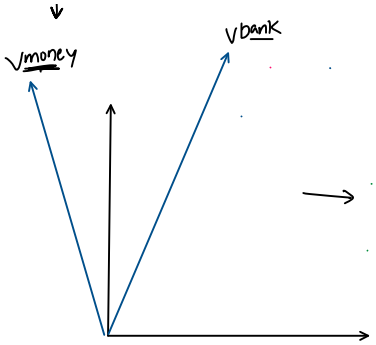
$W_{21} = 0.2$ $\quad W_{22} = 0.8$

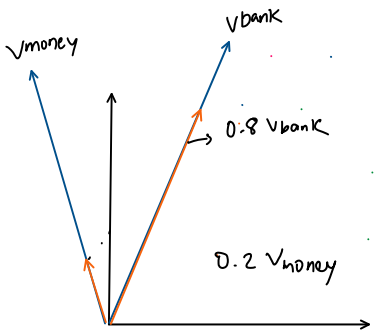
1

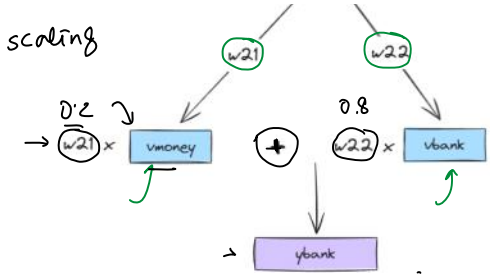


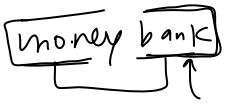


scaling

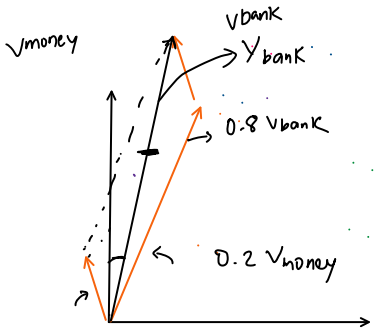


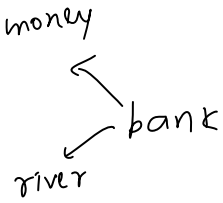


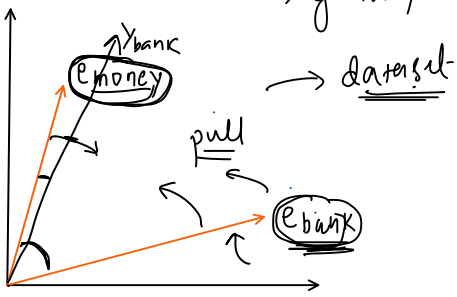




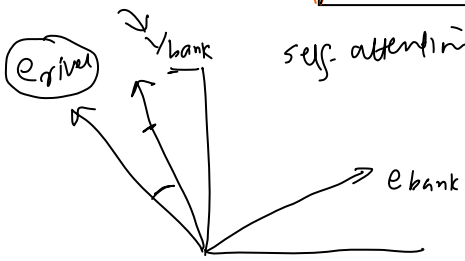
self attention
↳ gravity







dataset



self attention

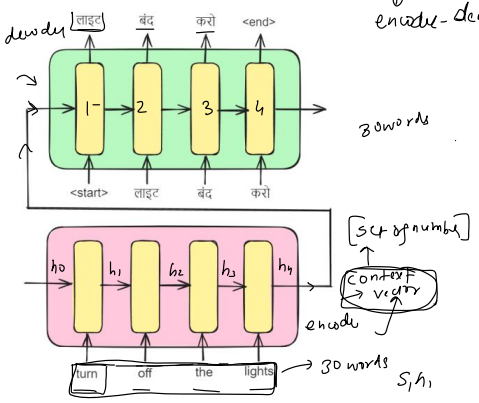
river bank

Recap of Attention

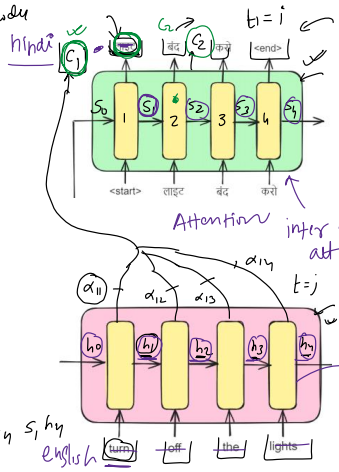
11 March 2024 17:09

long trans → NN

encoder-decoder



Attention



$c_i = \sum \alpha_{ij} h_j$ ✓

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

$e_{ij} = s_i^T h_j$ ✓

e_{11} $s_1 h_1$

e_{12} $s_1 h_2$

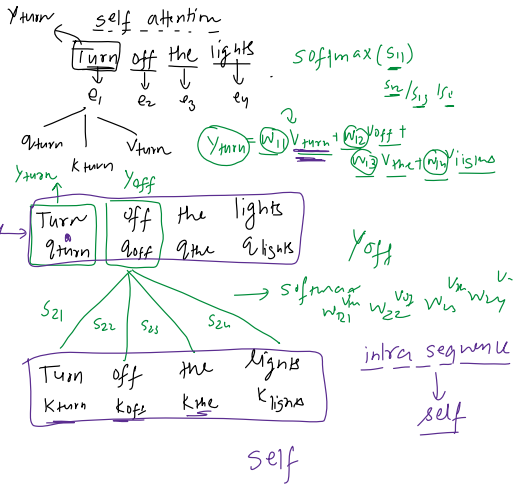
e_{13} $s_1 h_3$

e_{14} $s_1 h_4$

$s_i \rightarrow$ query

$h_j \rightarrow$ key

$h_j \rightarrow$ value



$$c_i = \sum \underbrace{\alpha_{ij}}_{\text{weights}} h_j$$

$$\alpha_{ij} = \text{softmax}(\underbrace{e_{ij}}_{\text{alignment score}})$$

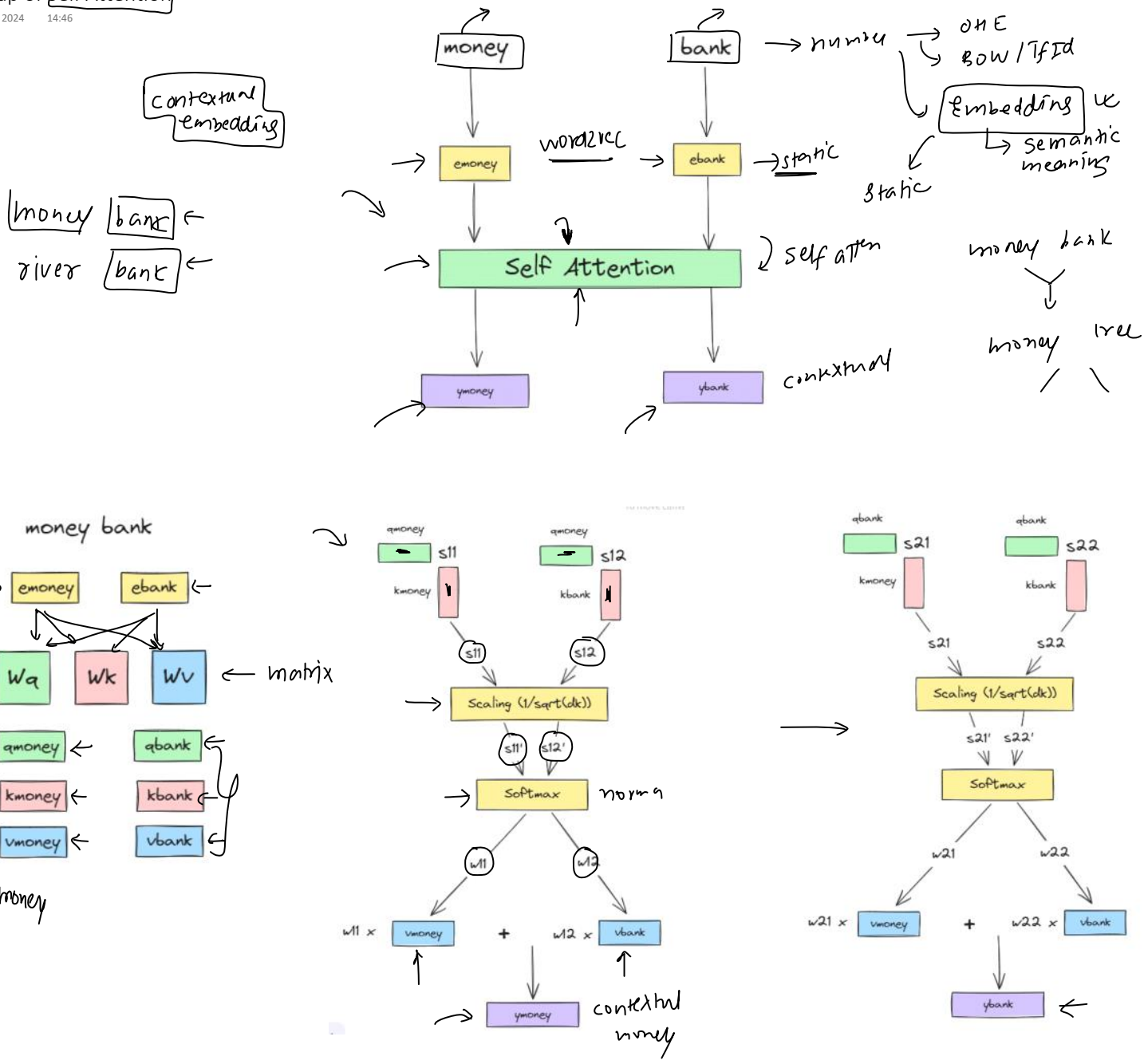
$$e_{ij} = s_i^T h_j$$

→ Luong attention

self attention

Recap of Self Attention

08 April 2024 14:46

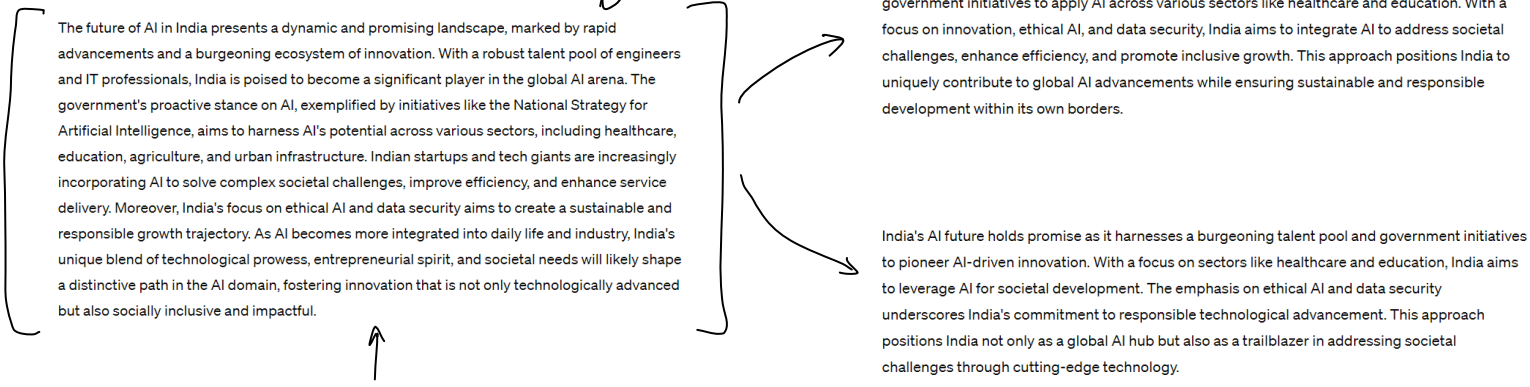


Problem with Self Attention

08 April 2024 14:47

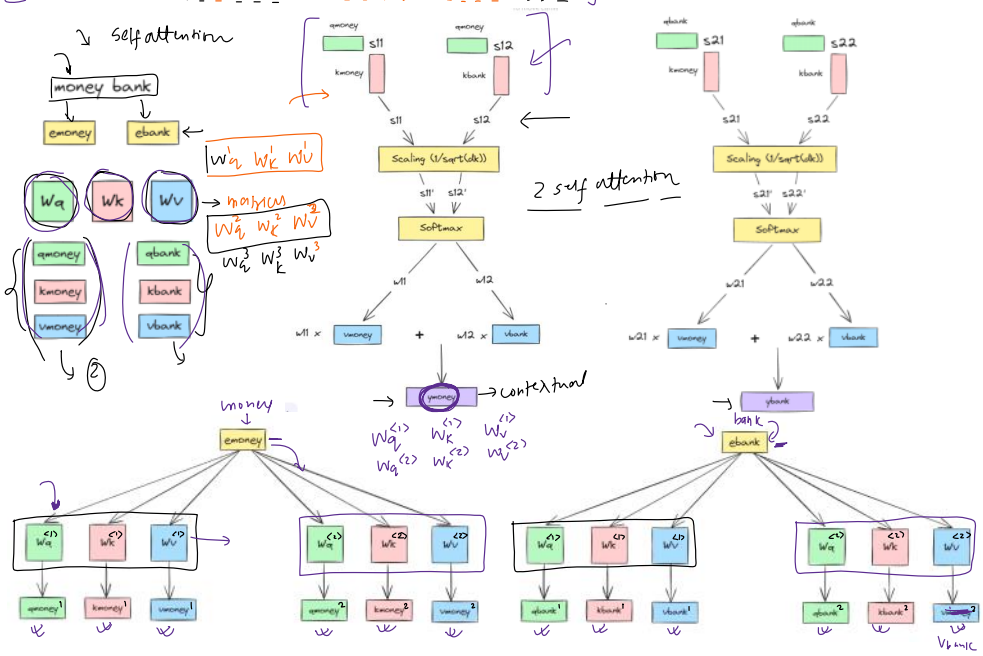
The man saw the astronomer with a telescope

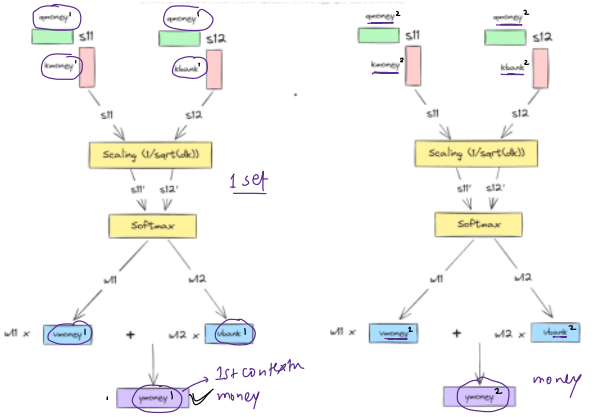
	The	man	saw	the	astronomer	with	a	telescope
The								
man								
saw								
the								
astronomer								
with								
a								
telescope								

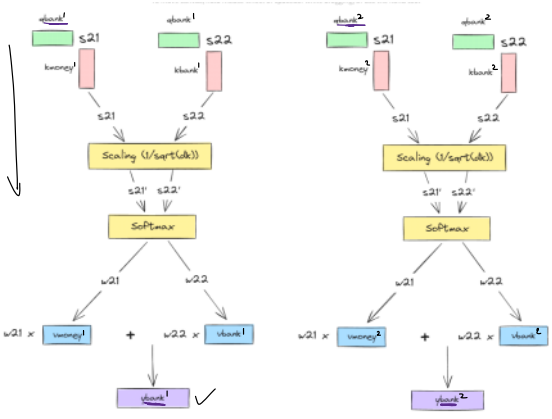


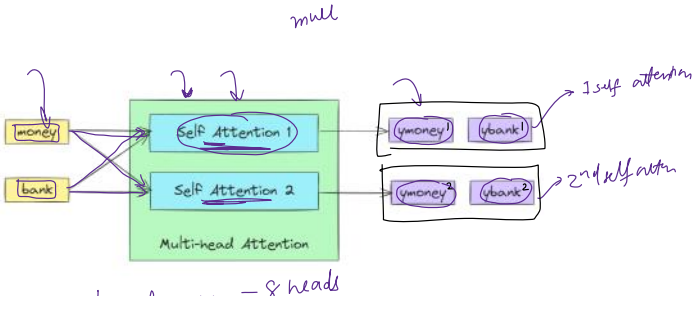
The man saw the astronomer with a telescope

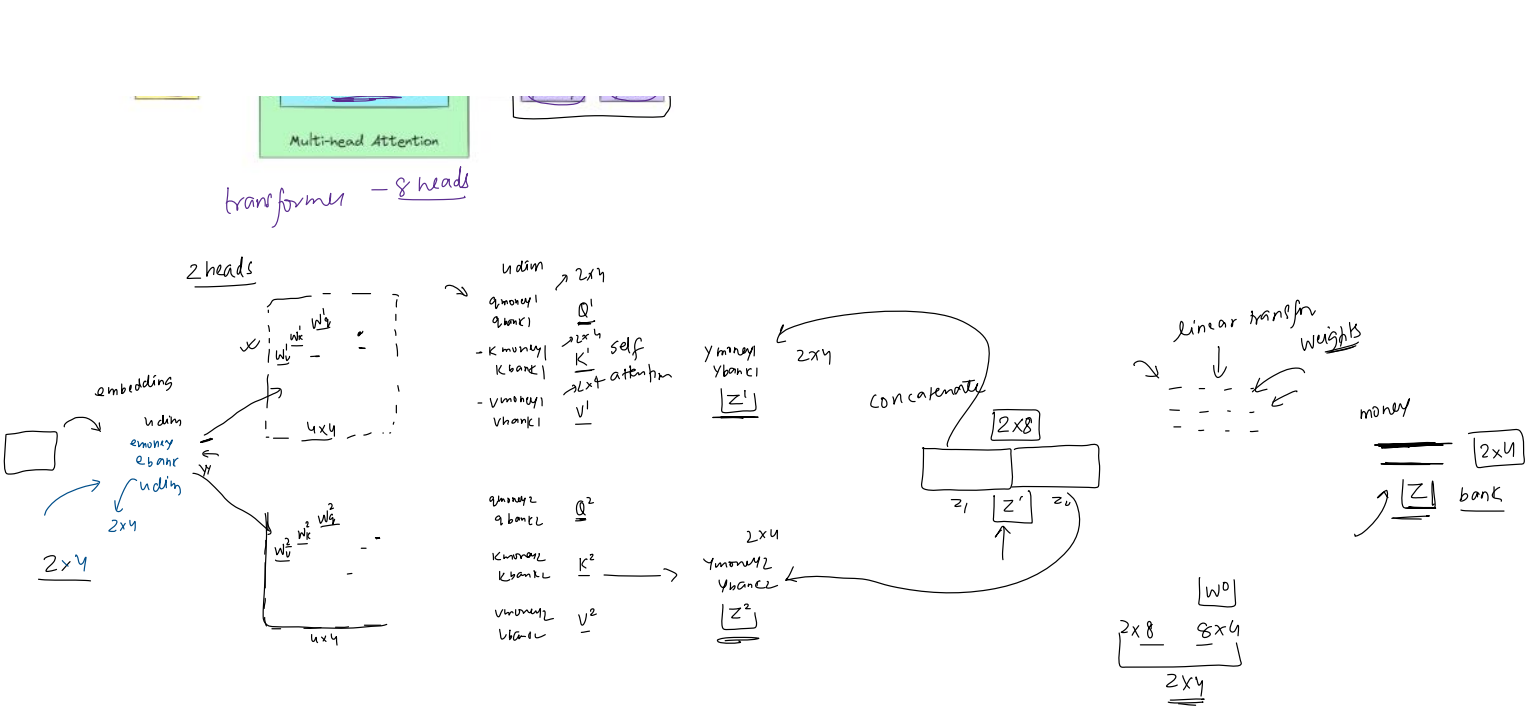
self attention → (2)

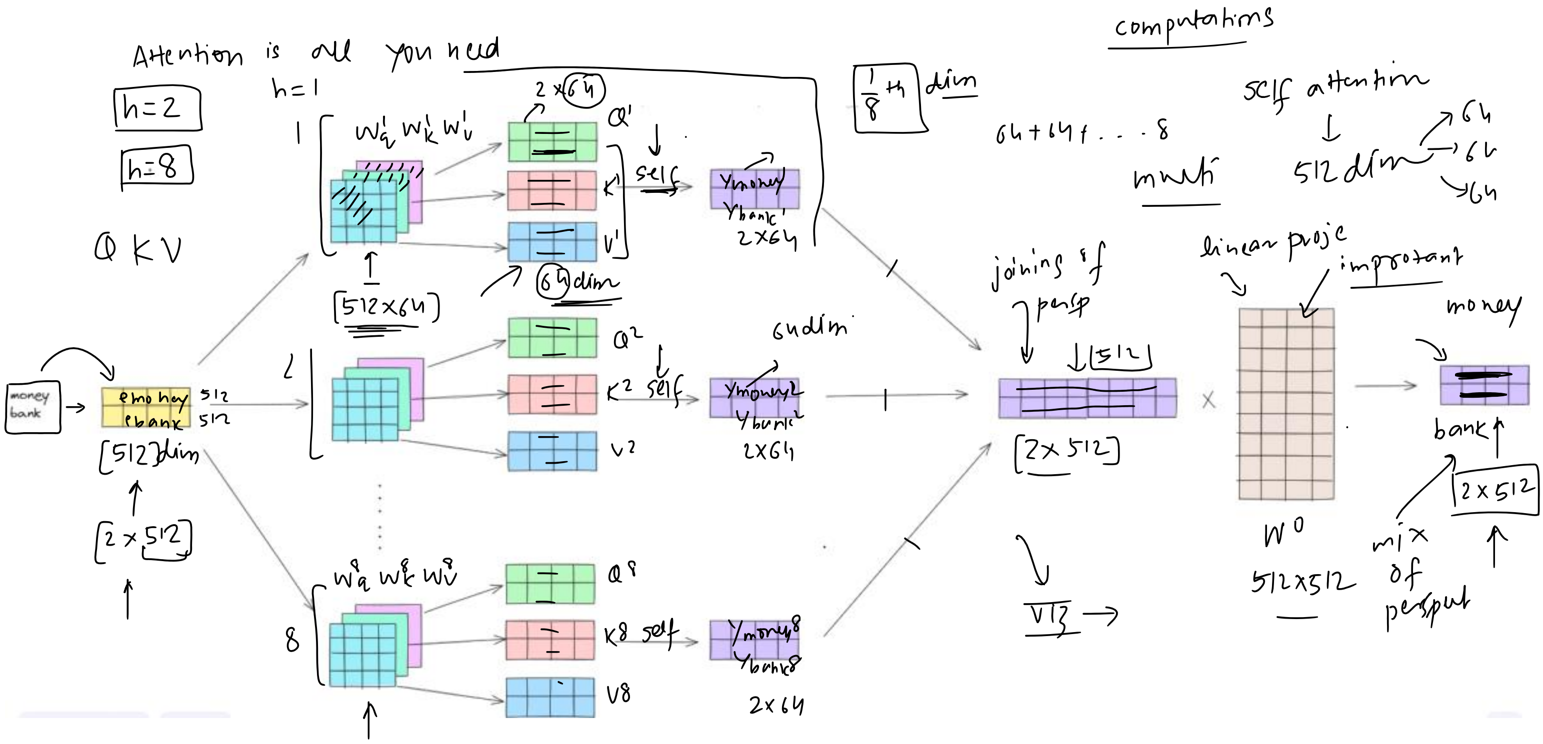









Multi-head Attention
transformer – 8 heads
2 heads
embedding
4dim
money
bank
adding
2x4
4x4
4dim
2x4
q money1
q bank1
Q1
2x4
K money1
K bank1
K1
self attention
2x4
V money1
V bank1
V1
q money2
q bank2
Q2
K money2
K bank2
K2
V money2
V bank2
V2
4x4
y money1
y bank1
Z1
2x4
y money2
y bank2
Z2
2x4
concatenate
2x8
z1
z'
z2
linear transform
weights
money
bank
2x4
Z
W0
2x8
8x4
2x4


Attention is all you need
h=1
h=2
h=8
Q K V
money
bank
emoney
ebank
512
512
[512] dim
[2 x 512]
W_q^1 W_k^1 W_v^1
[512x64]
2 x 64
64 dim
Q1
K1
V1
self
Ymoney
Ybank1
2x64
Q2
K2
V2
64dim
Ymoney2
Ybank2
2X64
W_q^8 W_k^8 W_v^8
Q8
K8
V8
Ymoney8
Ybank8
2x64
1/8th dim
computations
64+64+ . . . 8
multi
self attention
512 dim
64
64
64
joining of persp
[512]
[2x 512]
linear proje
improtant
W0
512x512
money
bank
[2 x 512]
mix of persput
VIZ