NLP

(1) Basic Language Model

(2) Distributed " "

(3) Large Language Model (LLM) + Gen AI

2000-2014 → RNN/LSTM/GRU

2014 - Encoder - Decoder

↓

Seq 2 Seq Learning with NN.

LSTM

Encoder-Decoder

↓

Research Paper

Transformers

Generate Text

understand language

Encoder → BERT

T5 BART

Decoder → GPT

Transformers + BERT + GPT

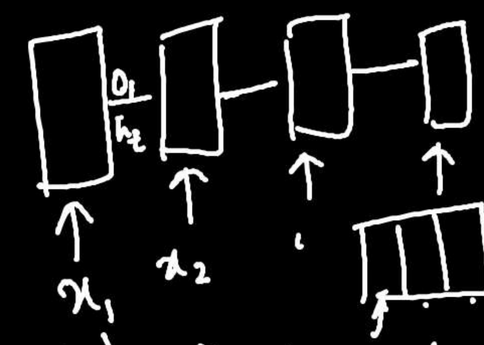
Theory + Case study

Huggingface — Arch, models, datasets

Company + Open Source commun.

NN

1. ANN/MLP/DNN — Tabular data
2. CNN/ computer vision — Image
3. RNN/LSTM/GRU — Text (Sequential data) → LSTM/GRU
4. Transformers — multimodal (1, 2, 3 + Re-inforcement learning)



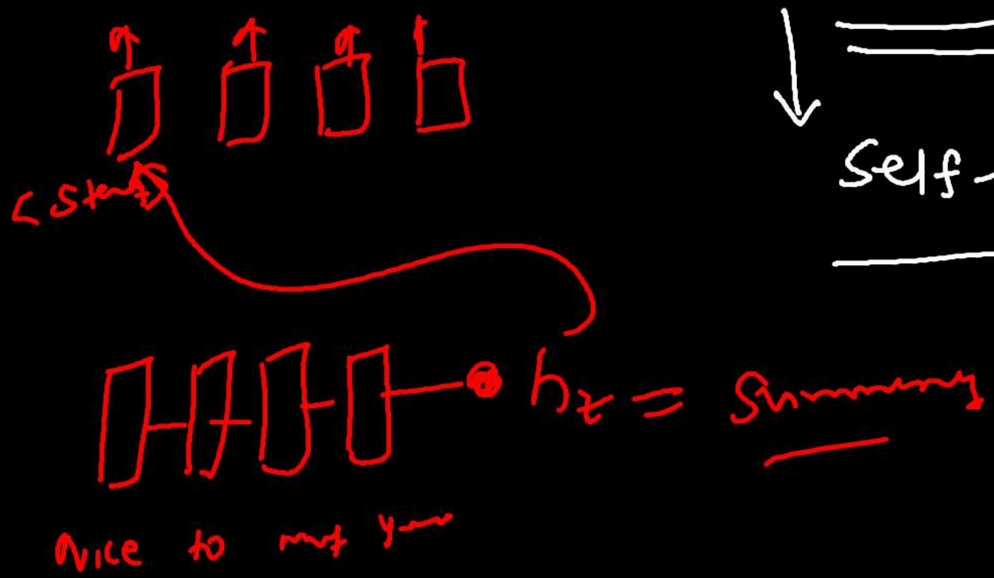
↳ seq 2 seq data

↓

Self-attention model, parallel processing

↓

very fast



2015 - Bahdanau - Neural Machine Translation - sequenty model

↳ Attention model → very famous

Architecture

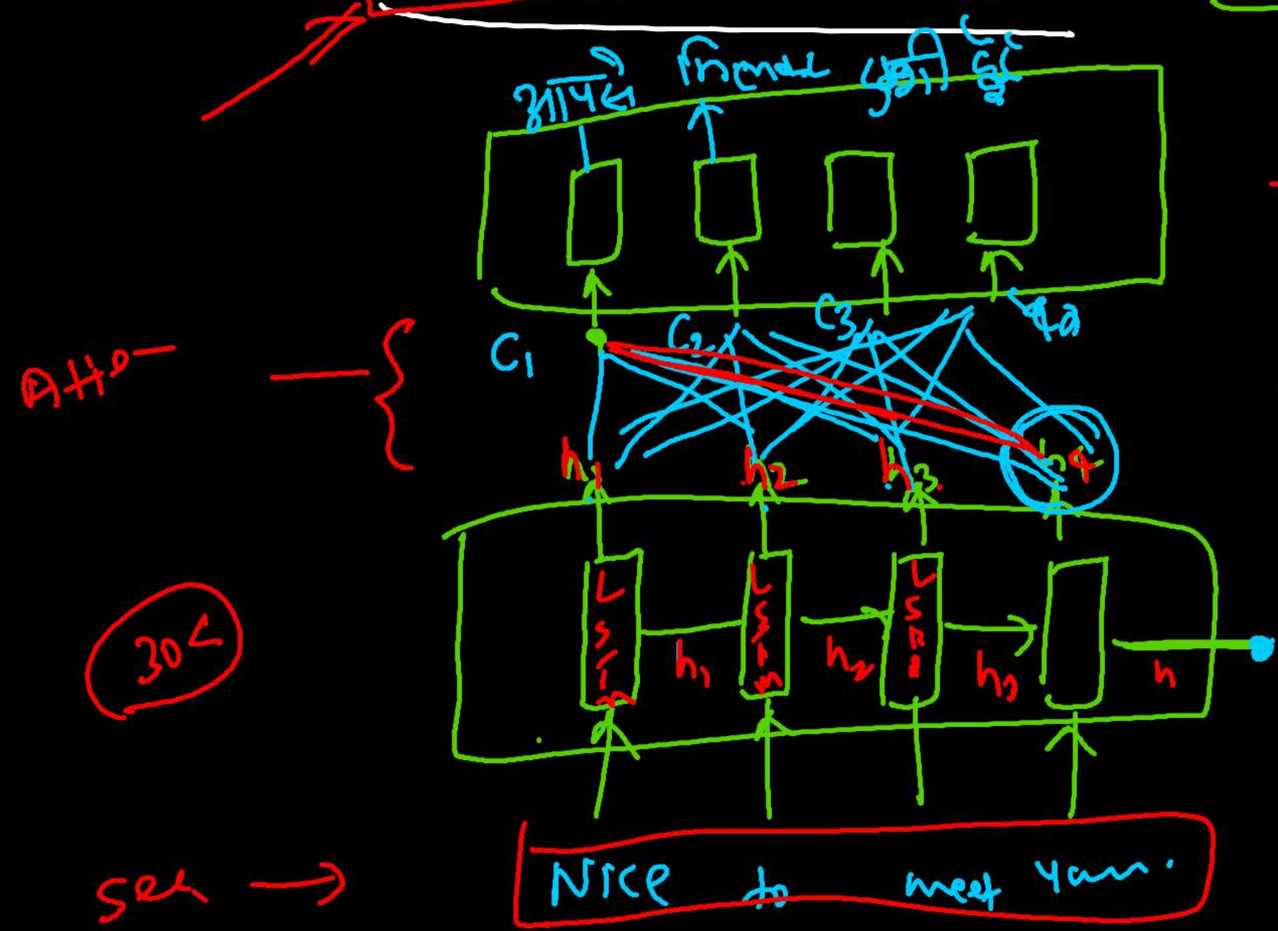
LSTM

traditional

slow

huge data

Problem



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

attention weight.

$1 \times a = 10$

$2 \times b = 10$

$100 \times c = 10$ = d

$500 \times d = 1,2,4$ → weight

V.V. Significant

2017 – Transformers – Attention is All you need

① What is model?

② What is language model?

LLM — Model

Language

Relationship / Pattern b/w words & sentences

x input → output
y

Language data

④ Language modeling Techniques?

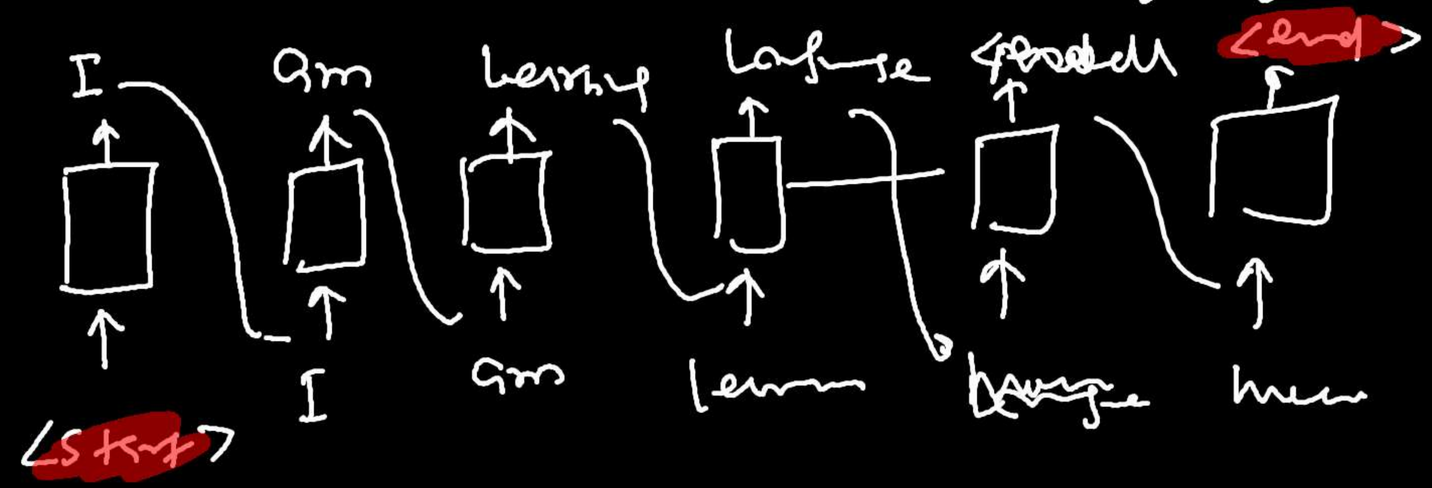
① AutoRegressive Task ⇒ uni-directional

A mask is applied to fill the sent—

② Auto Encoding Task

↳ Bi-directional

<S> I am learning language modeling <s>

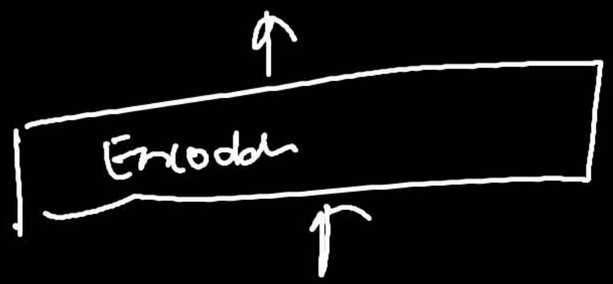


Auto Regressive → Predict one token at a time

Auto encoder → Generate the entire target seq ~~token together~~

Aims to max^m the likelihood of the next token given the previous token

Min^m Reconstruction error.



Summary

2014 → Encoder - Decoder architecture

2015 → Bahdanau Attention Mechanism

2017 :— Transformer Architecture

↳ Attention is all you need

↳ universal model

2018 → LLM → fine-tuned mod.—

Paper :- Attention is all you need

Goal → Proposed Transformer Architecture which used ——

→ Encoder - Decoder Architecture

→ Self - attention Mechanism

→ Positional Encoding

→ Multihead self - attention Model

→ Cross Attention bin Multihead.

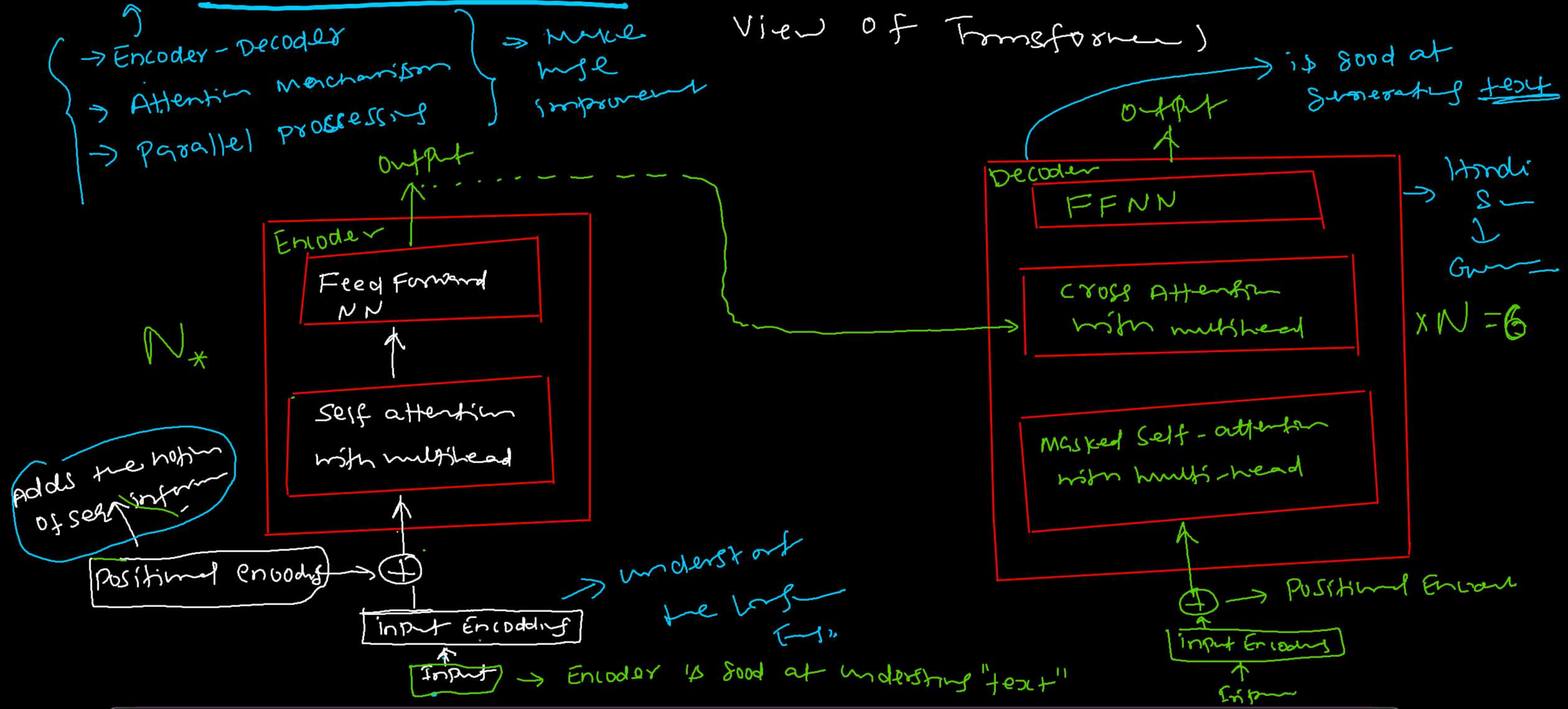
W2V - 50 to 300

Building Blocks

- Please note, this is a Simplified View of Transformer)

- → Encoder - Decoder
- → Attention Mechanism
- → Parallel processing

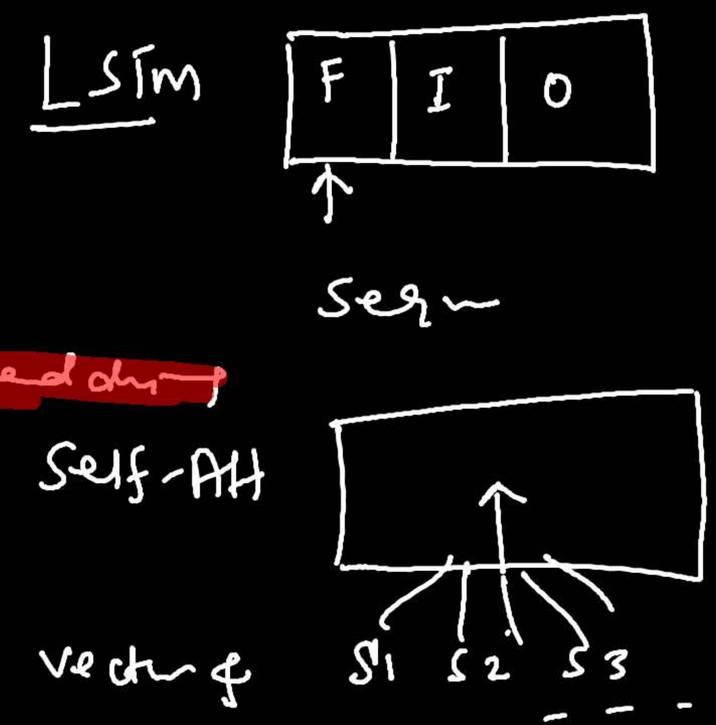
→ Make huge improvement



Self-attention vs multihead

① Attention replaced need to recurrence & make the process of embedding calculation parallel.

② Attention mechanism will find how each word related to all other words in sequence & generate contextual embedding

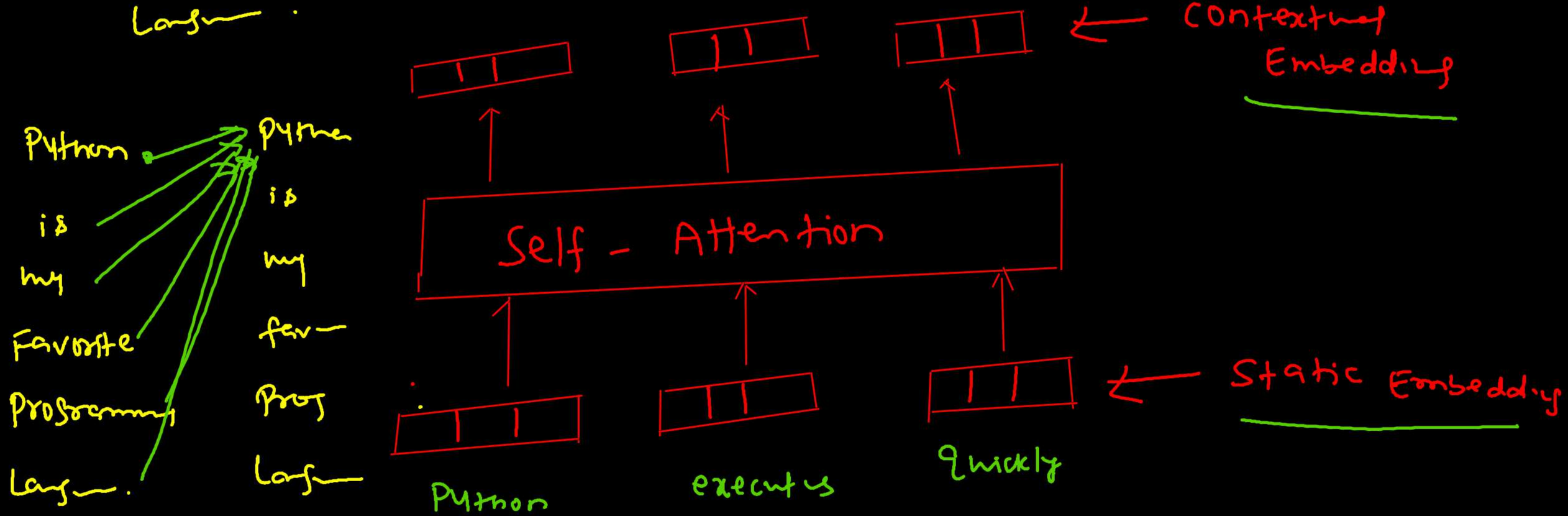


* How to find word similarity?

→ Attention will run dot product b/w word vector & determine the strongest relationship of a word with all other words.

Self - Attention

eg :- Python is my favorite Programming Lang.



dense = 5

W2V

		Avg
python	= [0.1 0.2 0.9 0.7 0.6]	= [0.6]
executes	= [0.2 0.3 0.4 0.6 0.3]	= [0.5]
quickly	= [0.3 0.1 0.8 0.7 0.6]	= [0.3]

e_{python} $e_{executes}$ $e_{quickly}$

python = 0.6 * python + 0.5 * executes + 0.3 * quickly

executes = 0.2 * python + 0.7 * execute + 0.1 * quickly

quickly = 0.1 * python + 0.1 * executes + 0.8 * quickly

$e_{python} * e_{python}$ $\quad$ $e_{python} * e_{exec}$ $\quad$ $e_{python} * e_{quick}$

$$z_{python} = \boxed{0.5} * e_{python} + \boxed{0.3} * e_{executing} + \boxed{0.2} * e_{quickly}$$

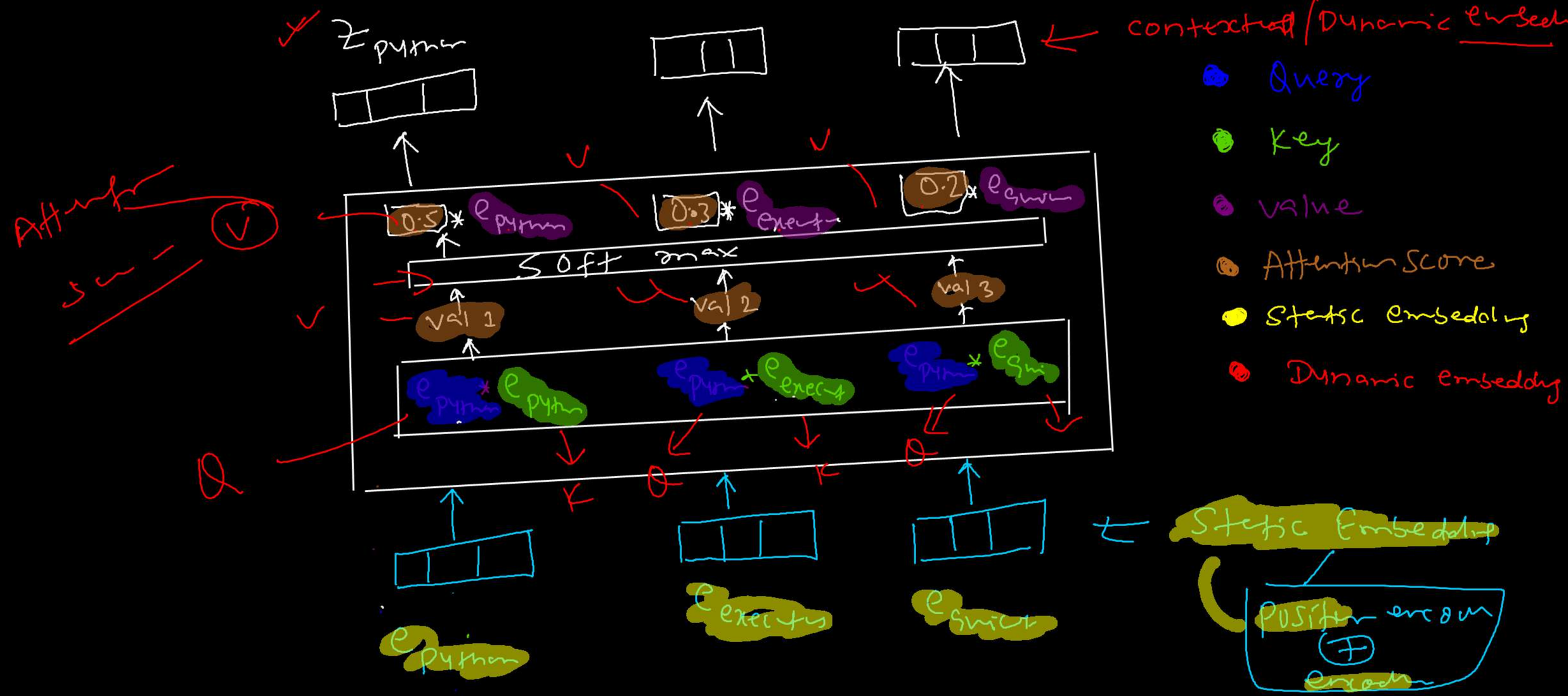
$e_{exec} * e_{python}$ $\quad$ $e_{exec} * e_{exec}$ $\quad$ $e_{exec} * e_{quick}$

$$z_{execut} = \boxed{0.2} * e_{python} + \boxed{0.7} * e_{execut} + \boxed{0.1} * e_{quick}$$

$$z_{quickly} = \boxed{0.1} * e_{python} + \boxed{0.1} * e_{exec} + \boxed{0.8} * e_{quick}$$

$e_{quickly} * e_{python}$ $\quad$ $e_{quick} * e_{py}$ $\quad$ $e_{quick} * e_{quick}$

To complete Process → calculation for the "python" query



V.V.V. Imp

Intern

Q① What is Q, K & V?

Q② :- Why Q, K & V?

Q :- Query - What information the model is trying to extract.

K - Key :- Content or features that the model can pay attention to

→ token embedding directly input transformation in Q, K & V, the model would be limited to just one fixed way of computing relationship between tokens.

① init the 3 weight : W_q W_k W_v

② cal : $Q = XW_q$ $K = XW_k$ $V = XW_v$

$[Q]_{T \times d}$ $[K]_{T * d}$ $[V]_{T \times d}$

③ cal $\boxed{\text{Soft}\left(\frac{Q * K^T}{\sqrt{d_x}}\right) V}$