1. RNN / LSTM / GRU
2. Encoder & Decoder
3. Encoder & Decoder with attention
4. Transformer ⇒ 3, 4 ⇒ Bert, GPT ...
5. Pytorch
6. Finetuning with LLM

- RAG, Agent, LLMOps

1. Data ingestion
2. Data Preprocessing
3. Model training
4. Model evaluation
5. Model deployment - (Lamuda)

NLP

- Text preprocessing
- Text encoding
- Text embedding

RNN (1987)

Seq to seq.

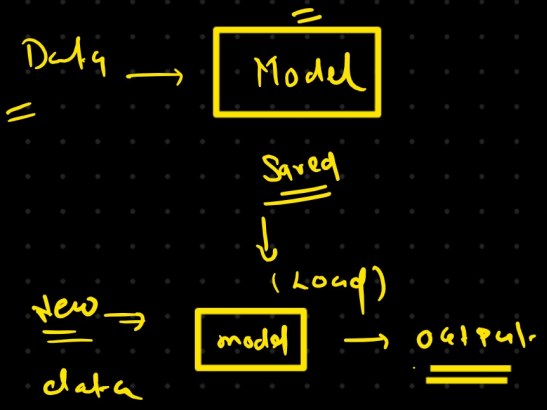
RNN (1987) → LSTM (1997) → GRU (2014) → EPD (2014) → EPD with Attention (2016) → Transformer (2017-2018)

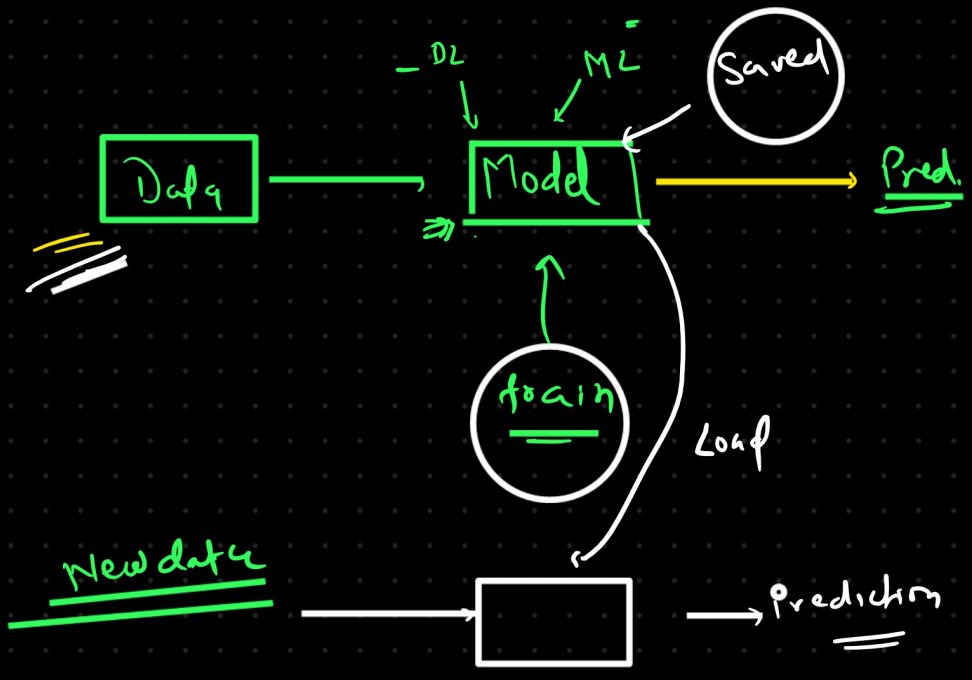
Transformer → BERT (2019) → GPT (2019-2020)

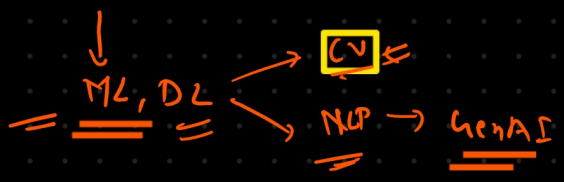
(Low of model.)

1. Why TL was not possible in NLP ?
2. 4-Imp research Paper
3. Quick intro Transformer
4. Self Attention

Transfer Learning





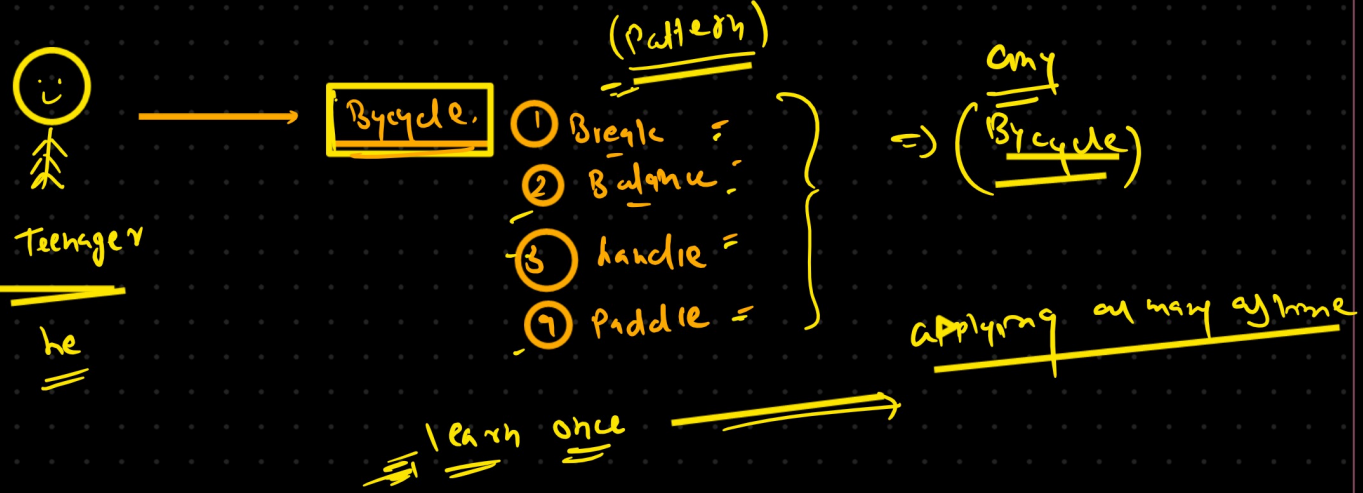


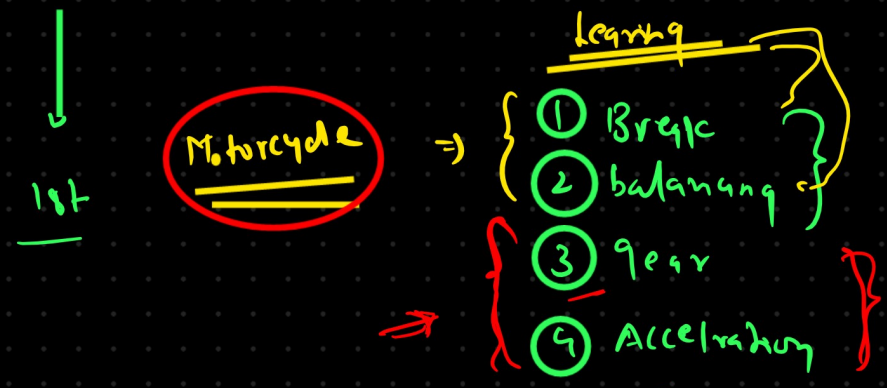
Computer vision

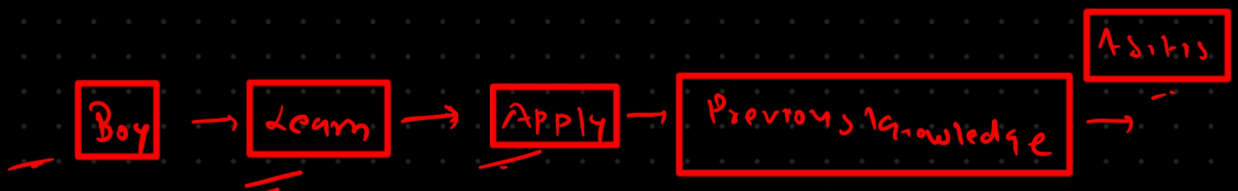
1. Image classification
2. Obj detection
3. obj Segmentation
4. Obj tracking
5. OCR

→ CNN

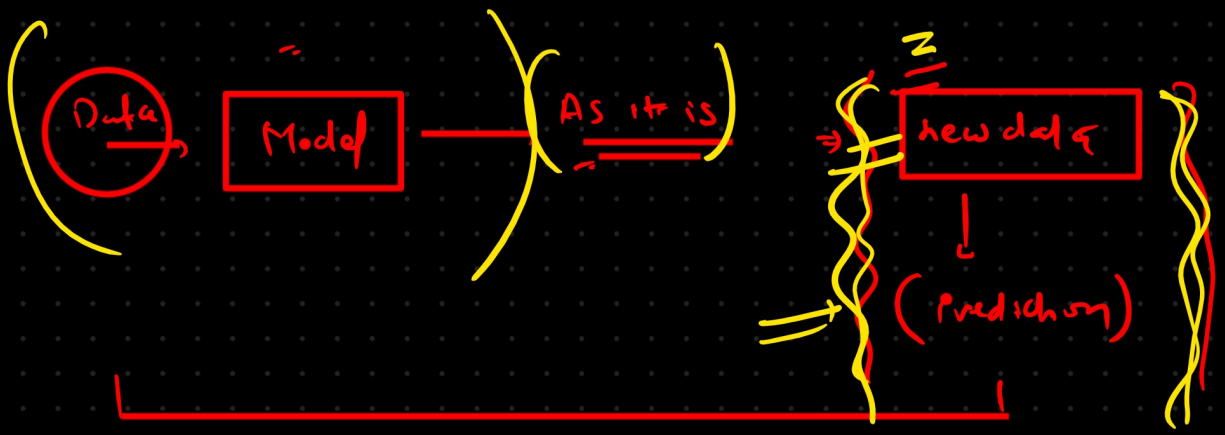
CNN ← {feature extraction}







→ transfer learning



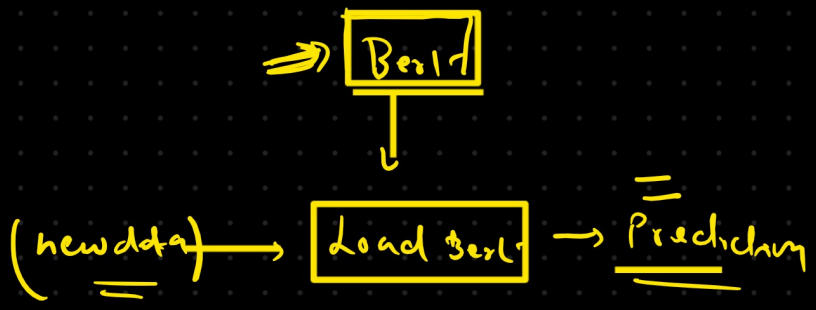
① transfer Learning

(Inferencing)

Define

Model ← training

② finetuning



Inferencing

tagging

load

(Bert + Layer) ⇒ (finetuning)

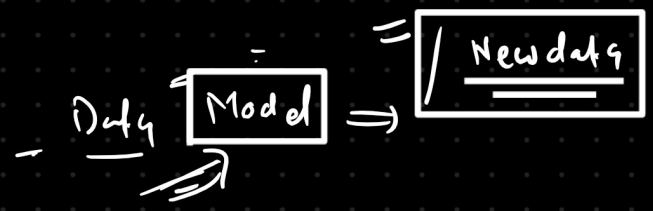
TF

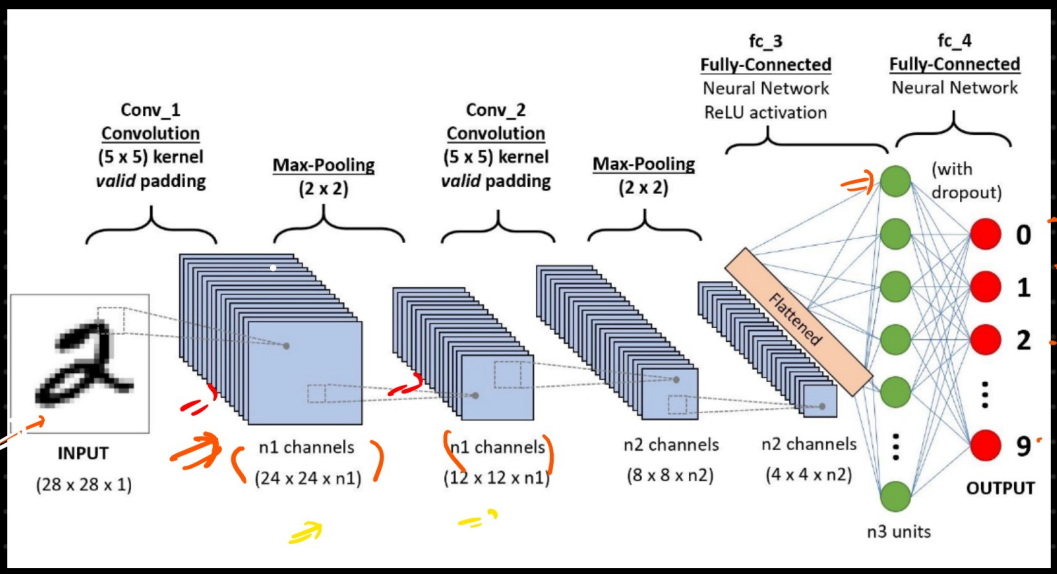
addon

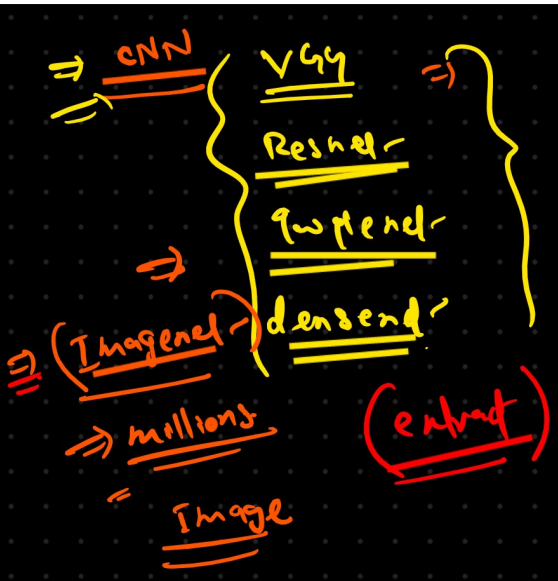
Google word2vec

↓

10







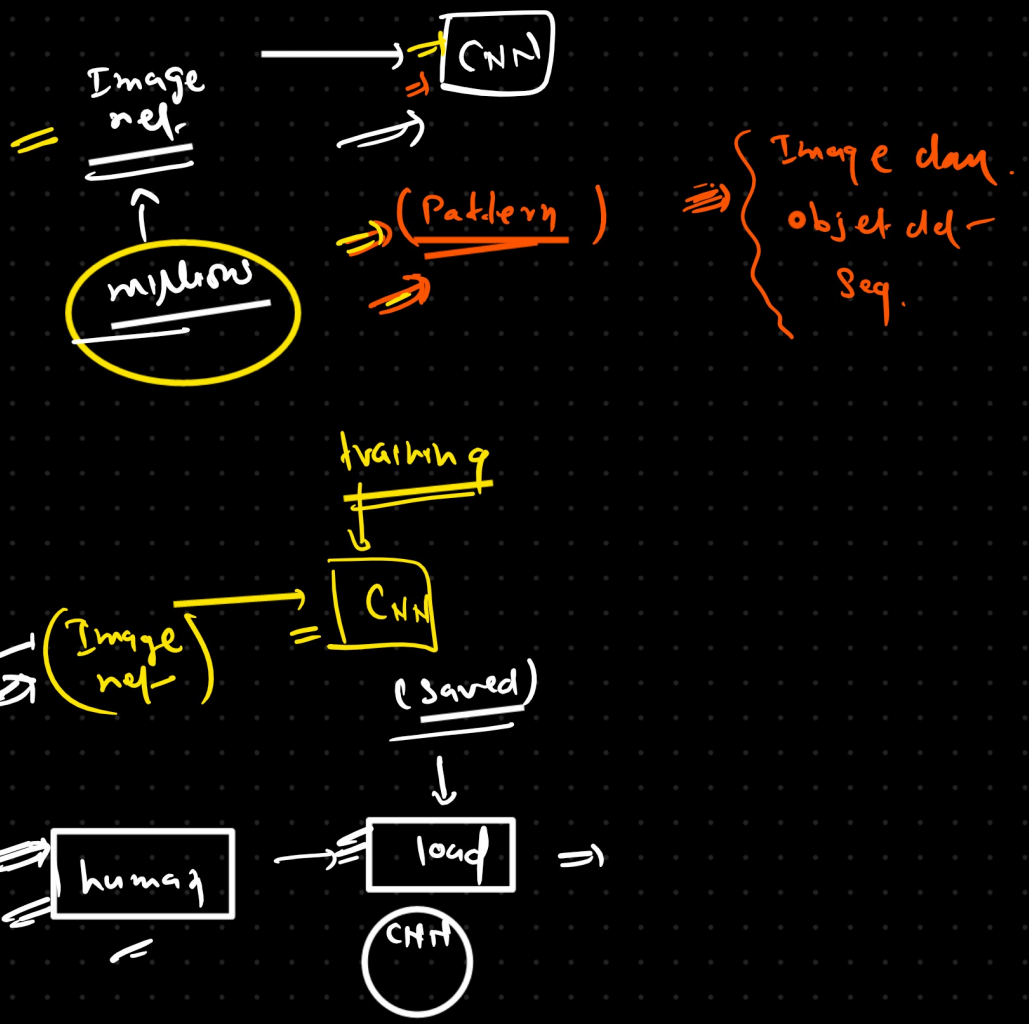


Image net
CNN
millions
Pattern
Image class. object det- Seq.
training
CNN
Image net
(saved)
human
load
CNN
(reloaded)
CNN
Saved
Image net (millions)
load
Data
CNN

NLP => - RNN, LSTM, GRU ← lot's of data

- ANN + Memory

1. text classification
2. text generation
3. Machine translation
4. text summarization

=> Sequential

=> timestamp

1. huge amt. data X
2. task exclusiveness

Transformers

1. RNN, LSTM, GRU ✗
2. Parallel data ✓
3. Huge amt ✓
4. Training extensive ✓

= Conversational AI

NLP => text

- text classification
- question answering
- lang translation
- text summarization

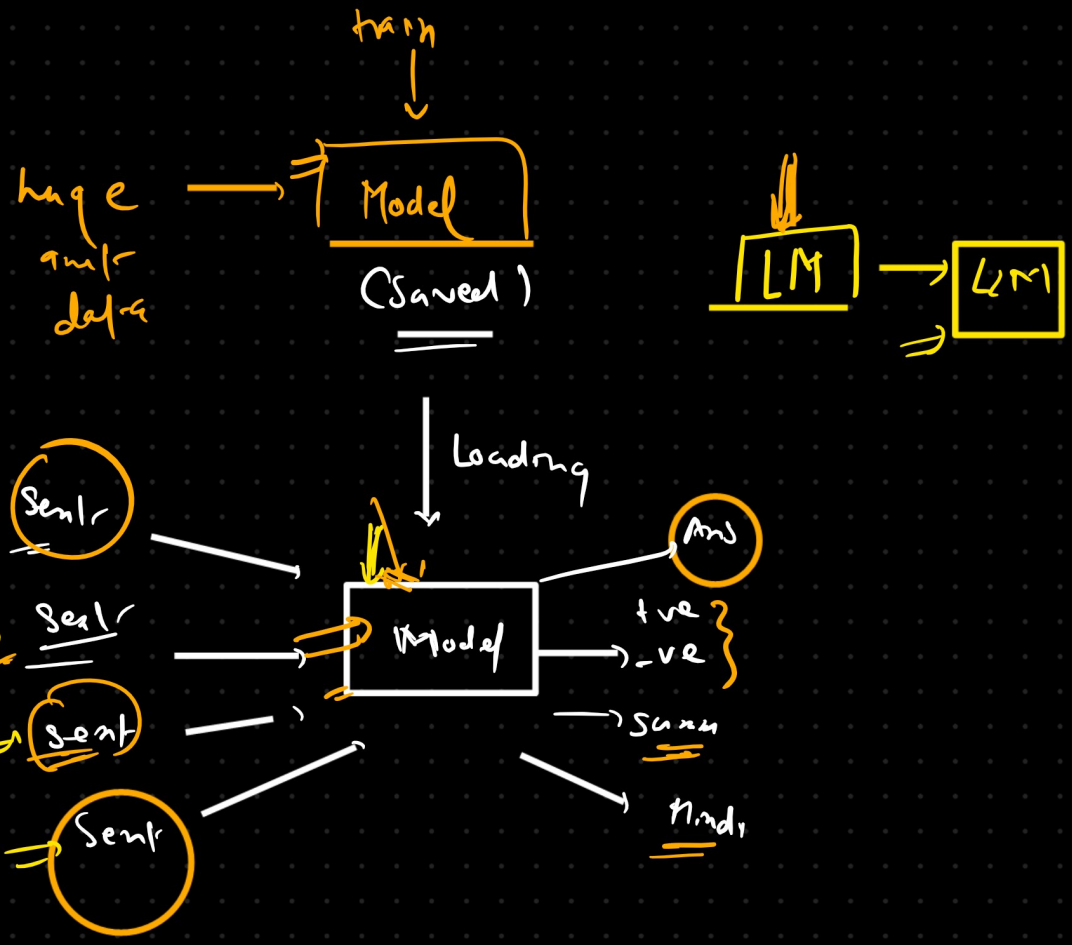
(generation)

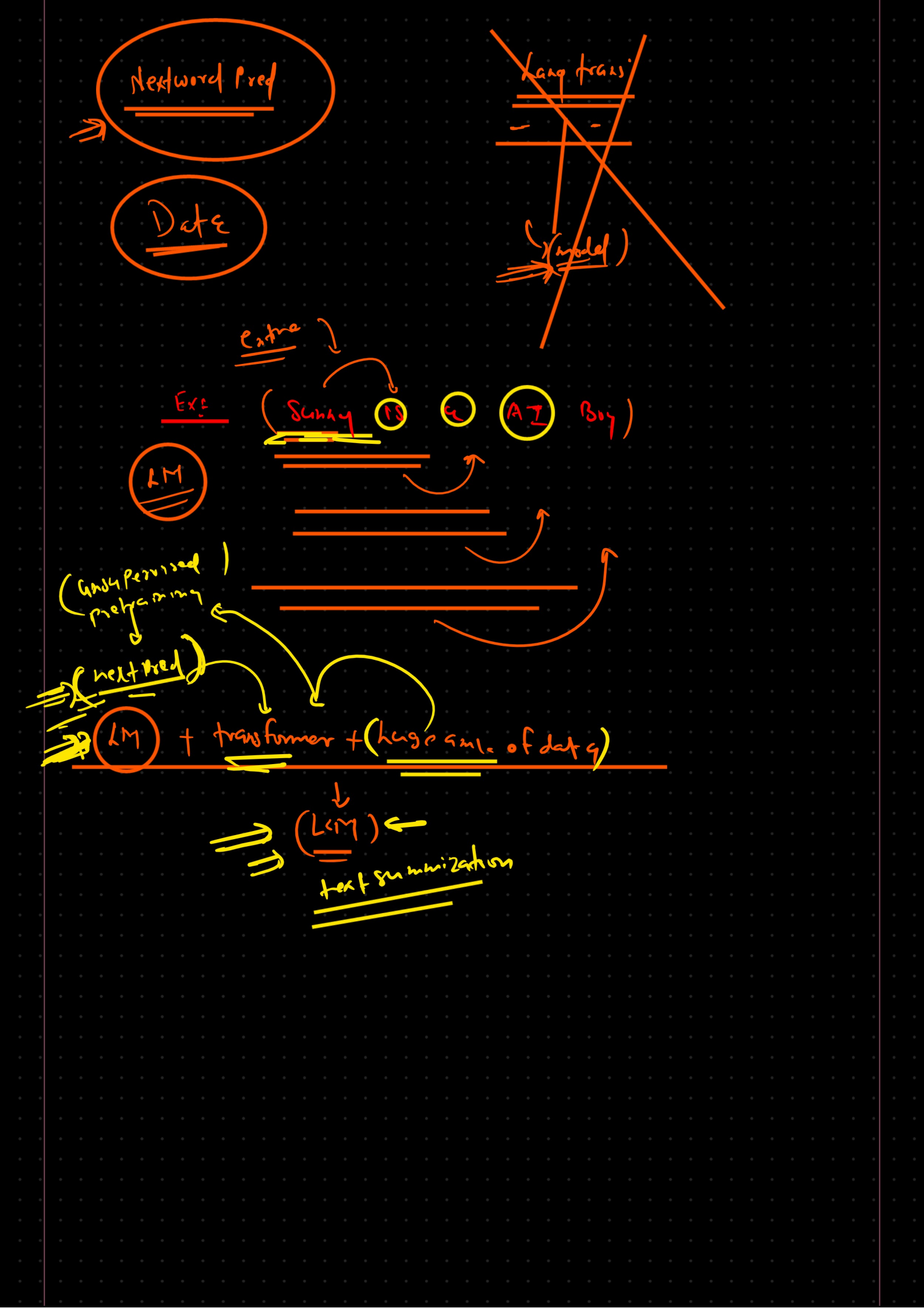
- text generation

Q → A

E → H

H → S




Nextword Pred
Date
Lang trans
(model)
Extra
Ex:
(Sunny is a AI Boy)
LM
(Unsupervised pretraining)
(next word)
LM + transformer + (huge amt of data)
(LLM)
text summization

RNN, LSTM, GRU ✗

EFD, EFD with attention

↳ huge amt of

transfer learning
fine tuning

↓ ↓

ULMFiT + transformer + huge amt of data

LM

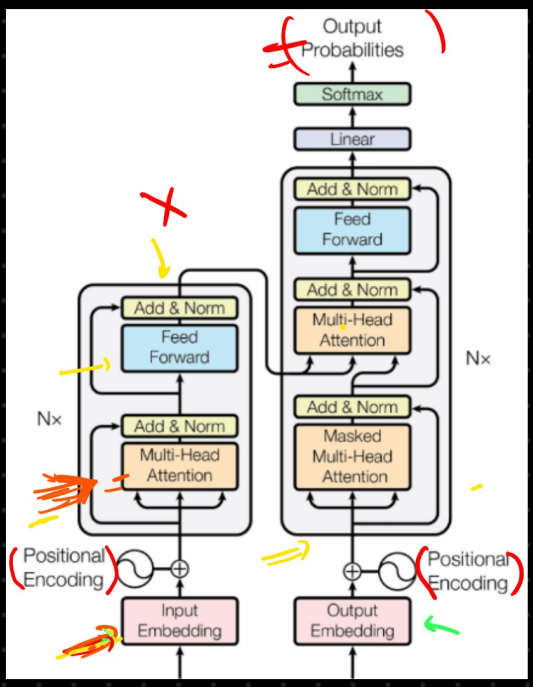
EFD

LLM

Bert, GPT

text

PDF → LM



1. Positional encoding
2. Self Attention / Multihead Self attention
3. Encoder working
4. Decoder working
5. Data flow in transformer

Sunny is a teacher
t=1 t=2 t=3 t=4

9,4000

t=1

self

River bank flow (Parallel)

(E)

Positional encoding layer

8

Self attion layer (Multihead)

ANN

BERT → Encoder

GPT → Decoder

Seq. → NLP model.

(text) HEE...

(vector)

vectorization

(embedding) (ANN)

(word2 vec) { Skip gram, CBOW }

encoding (freq)

1. OHE
2. BoW → N-Grams
3. TF IDF

(Self attention) - q, k, v → value

Contextual embedding

query key

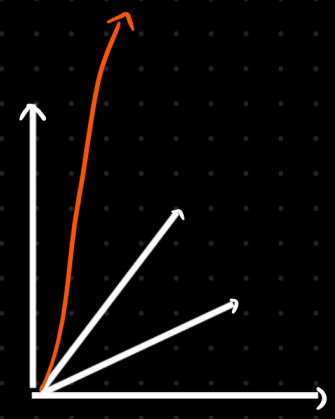
dynamic embedding

Word embedding

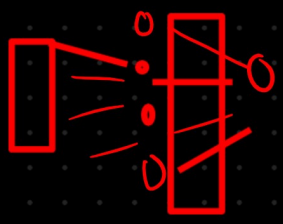
5-dim

F1 F2 F3 F4 F5

Similar { King → [0.8, 0.2, 0.1, 0.06, 0.3] ← vector (word2vec)
Queen → [ ]
Sunny → [ ]



1. An apple a day keep the doctor away ←
2. Apple is healty ←
3. Apple is better than banana ←
4. Apple make great mobiles ←



1000 - 10000 => 3000 F
2000 T

Apple → [fruit, tech]

good low

apple → [0.85, 0.15] ← (Model word2vec)

[0.85, 0.15] ← but fine

→ New Sent: Apple launch a new phone while I was eating Apple ← ⇒ ?

[0.85, 0.15]

fine

Contextual embedding & dynamic embedding (self attention)

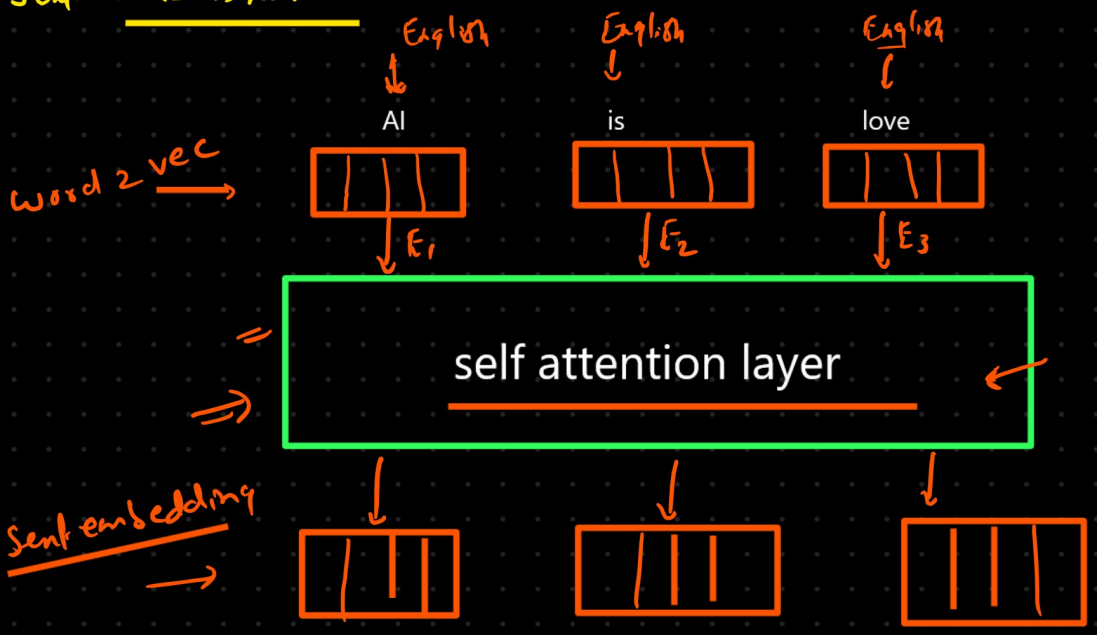
BANK →

Bank → [0.4, 0.6]

⇒ Money Bank ⇒ [0.4, 0.6]

River Bank [0.4, 0.6]

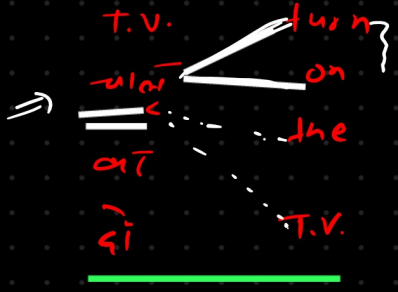
Sent. → AI is Love

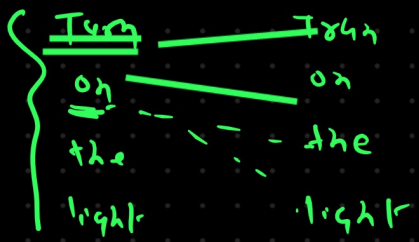


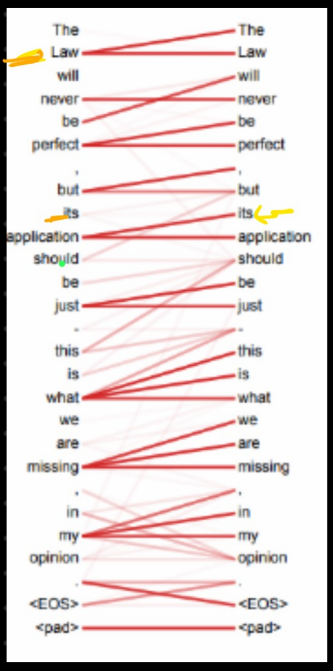
Translation

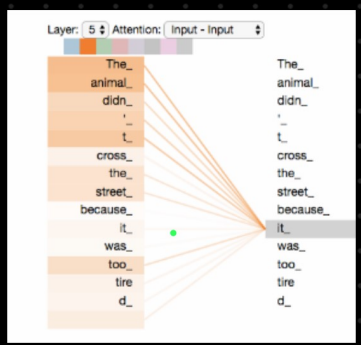
T.V. चालू कर दो

Attention X → (Self attention)









Example

S1 → money [bank] grow   S2 → River [bank] flow

S1: [bank] → w_1 * money + w_2 * bank + w_3 * grow

↓ {Q, K, V}

S2: [bank] → w_1 * River + w_2 * bank + w_3 * flow

(Random weight)

selfattention(Q, K, V) ⇒ (Query), (Key), (Value) = (Learning)

{ S1 → bank → 0.25 * money + 0.7 * bank + 0.05 * grow
{ S2 → bank → 0.5 * River + 0.4 * bank + 0.1 * flow

S1   (Q, K, V) = 1 (Random weight)

Money = 0.7 * money + 0.2 bank + 0.1 grow
bank = 0.25 * money + 0.7 bank + 0.05 grow
grow = 0.1 * money + 0.2 * bank + 0.7 * grow

S2

River = 0.8 * river + 0.15 bank + 0.05 flow
bank = 0.25 river + 0.78 bank + 0.02 flow
flow = 0.4 river + 0.01 bank + 0.59 flow

= Q, K, V ⇒ Contextual vector
- general
- specific contextual vector

Q, K, V → weight ⇒ Learnable

→ Positional
→ Encoder   (decoder)

GenAI → (finetuning) → {OpenAI, ClaudeAI, huggingface} (paid) → genAI
+ GeminiAPI (paid)
hugging face → API, langchain

(ollama, LMstudio)

Llamaindex