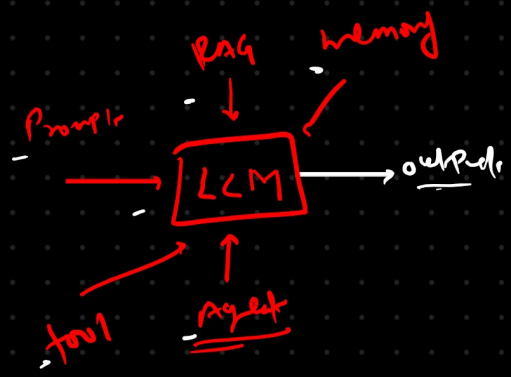
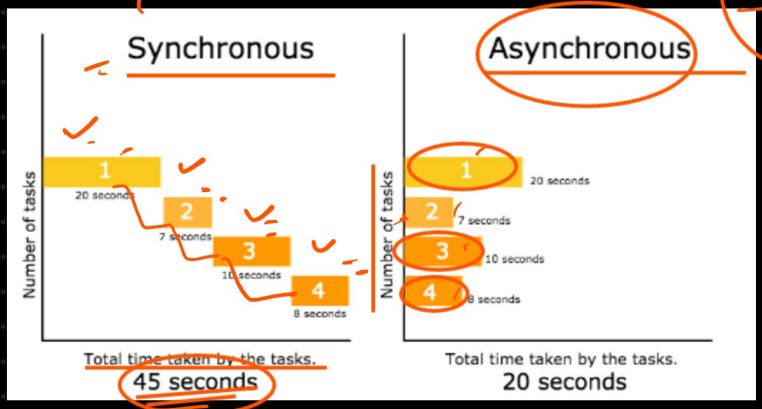


Prompt + LLM = output

Prompt + LLM + StrOutputParser = output

Sequential

Asyncio



Aug

1. Finetuning basic
2. Roadmap
3. LM (Bert, T5 ....)
4. LLM (gpt, Llama, mistral)
5. quantization
6. RLHF (DPO, PPO, KTO)

10-1

SFT → Lora, qlora → PEFT

Scratch → huggingface → NLP

Langchain → so many

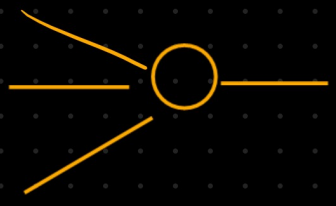
LLM → NN

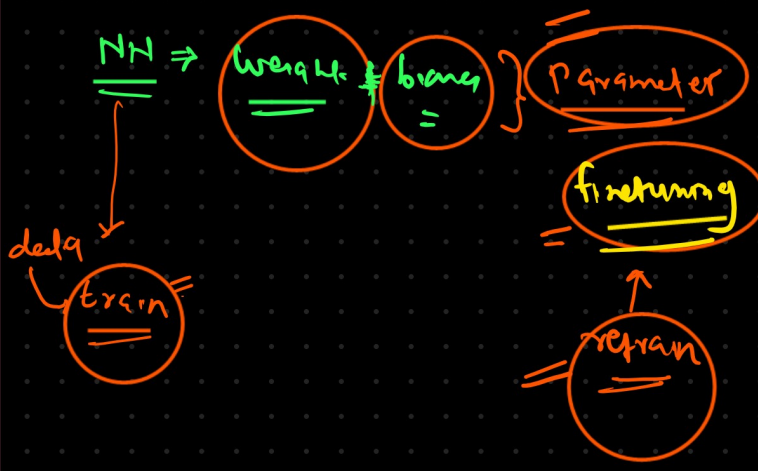
ANN

CNN → Convolution

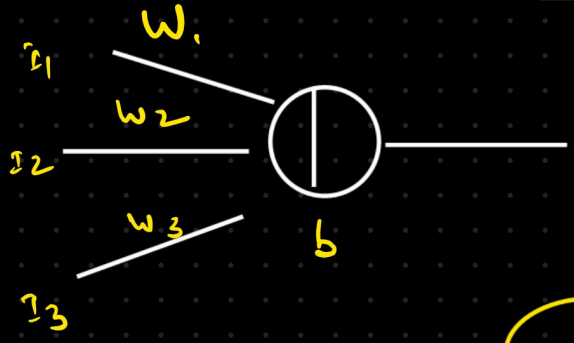
RNN → Recurrent

↳ Perceptron





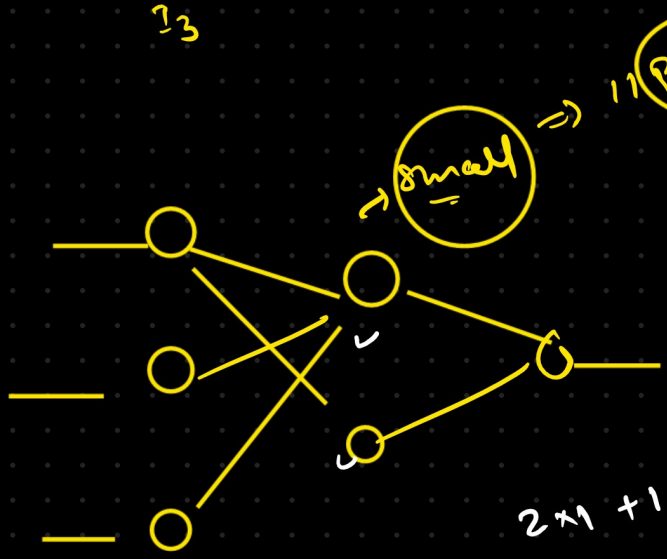
Neural Net -



w_1, w_2, w_3, b

↓

Parameter = 4



$3 \times 2 = 6 + 2$

(8)

$2 \times 1 + 1$

$2 + 1 = (3)$

$8 + 3 =$ (11) → parameter

(11) trainable Param

NN ⇒ ✗

Parameter → trainable param

↳ (w & b)

GPT3 ⇒ 175 B :

Billions ⇒ Size of the model

+

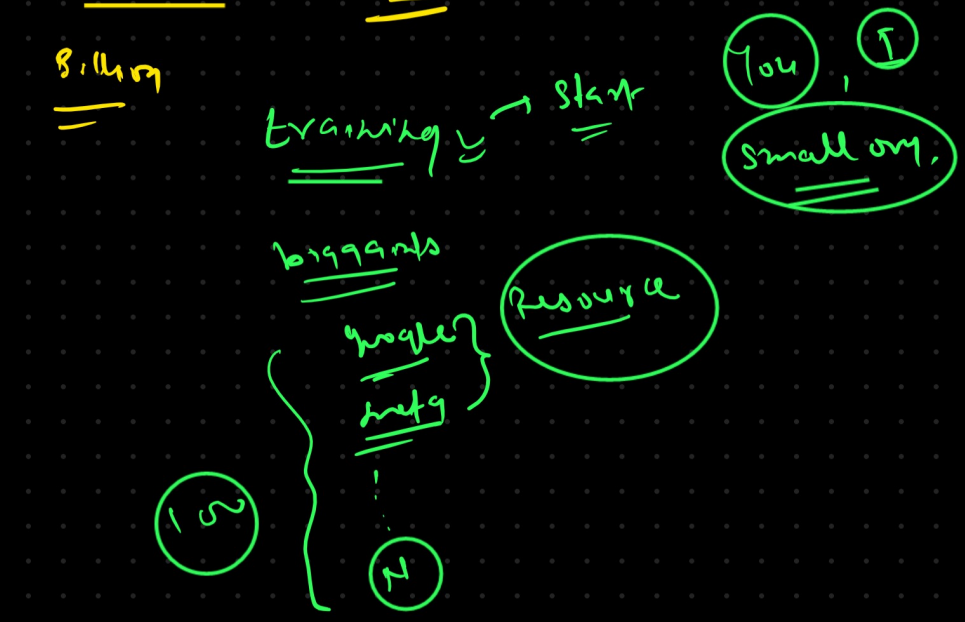
Data

Llama ⇒ 2B, 7B, 40B, 400B

Mistral ⇒ 2, 1.5, 7B, 8B, 80B

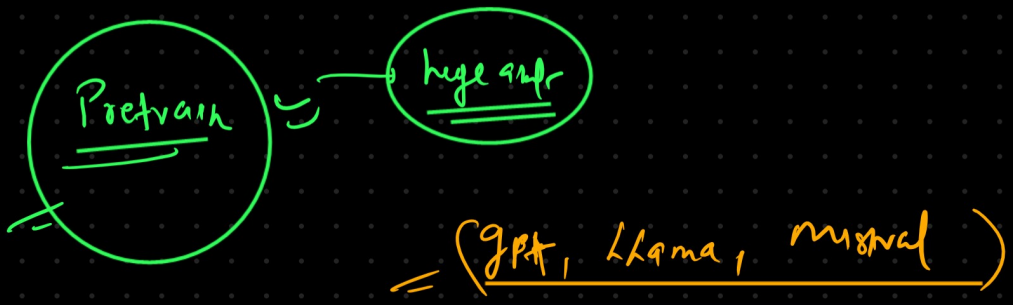
Gemini ⇒ Gemini 1.5 pro ⇒ 175 B.

Claude ⇒ Billions

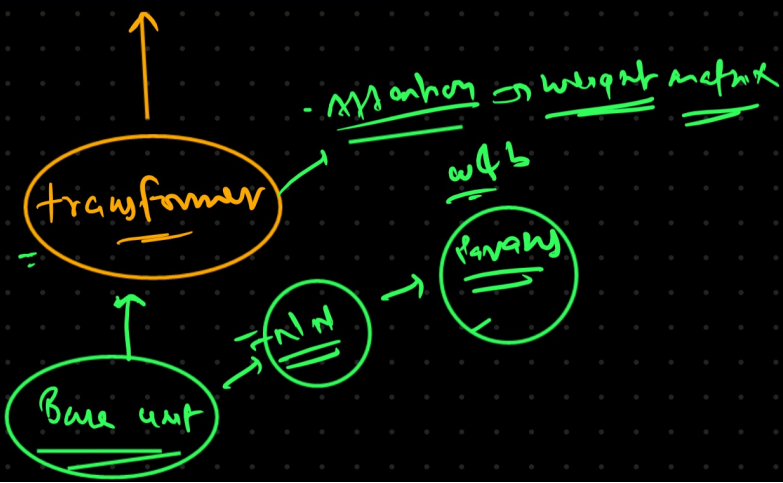


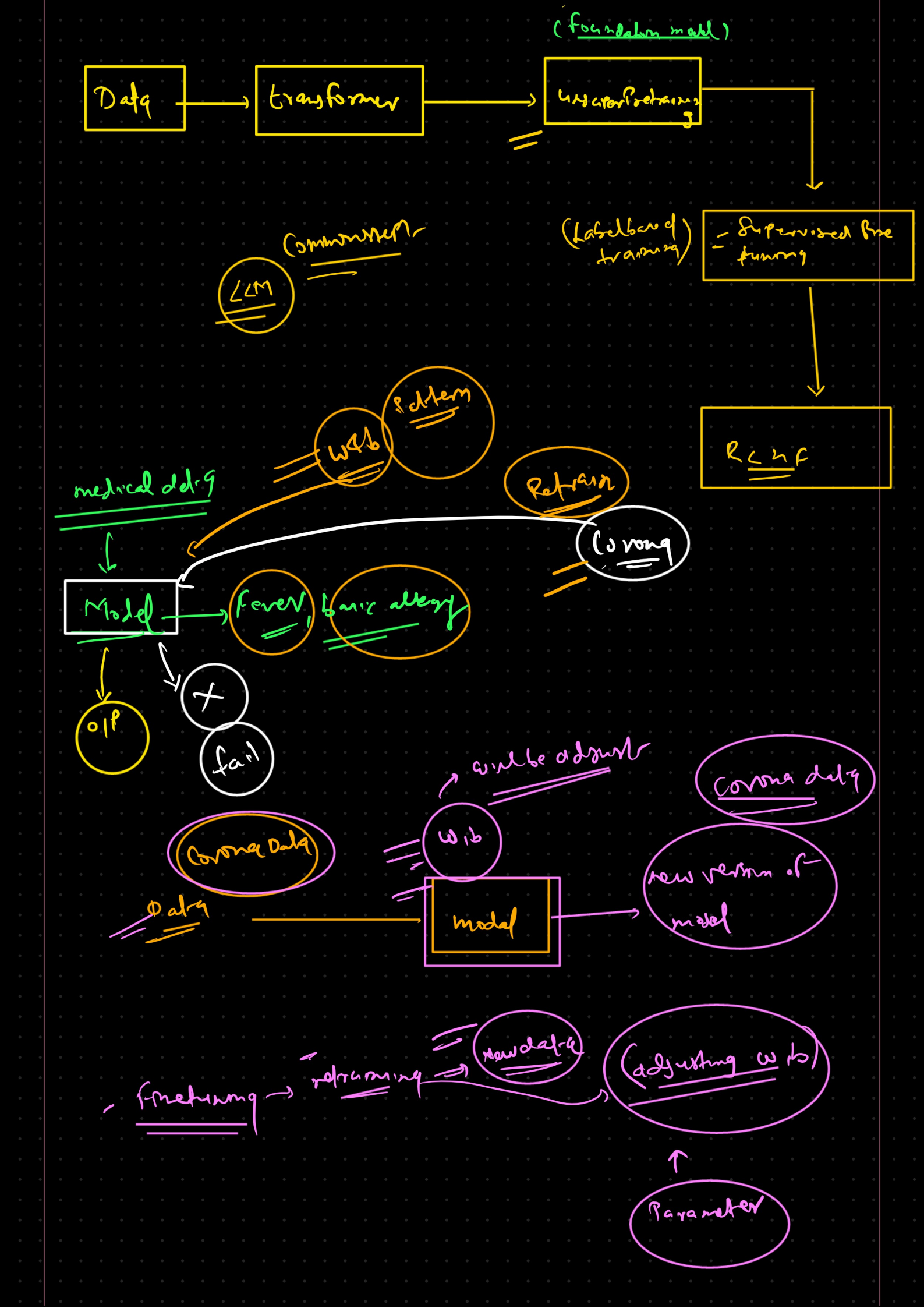
Llm → training → huge Params

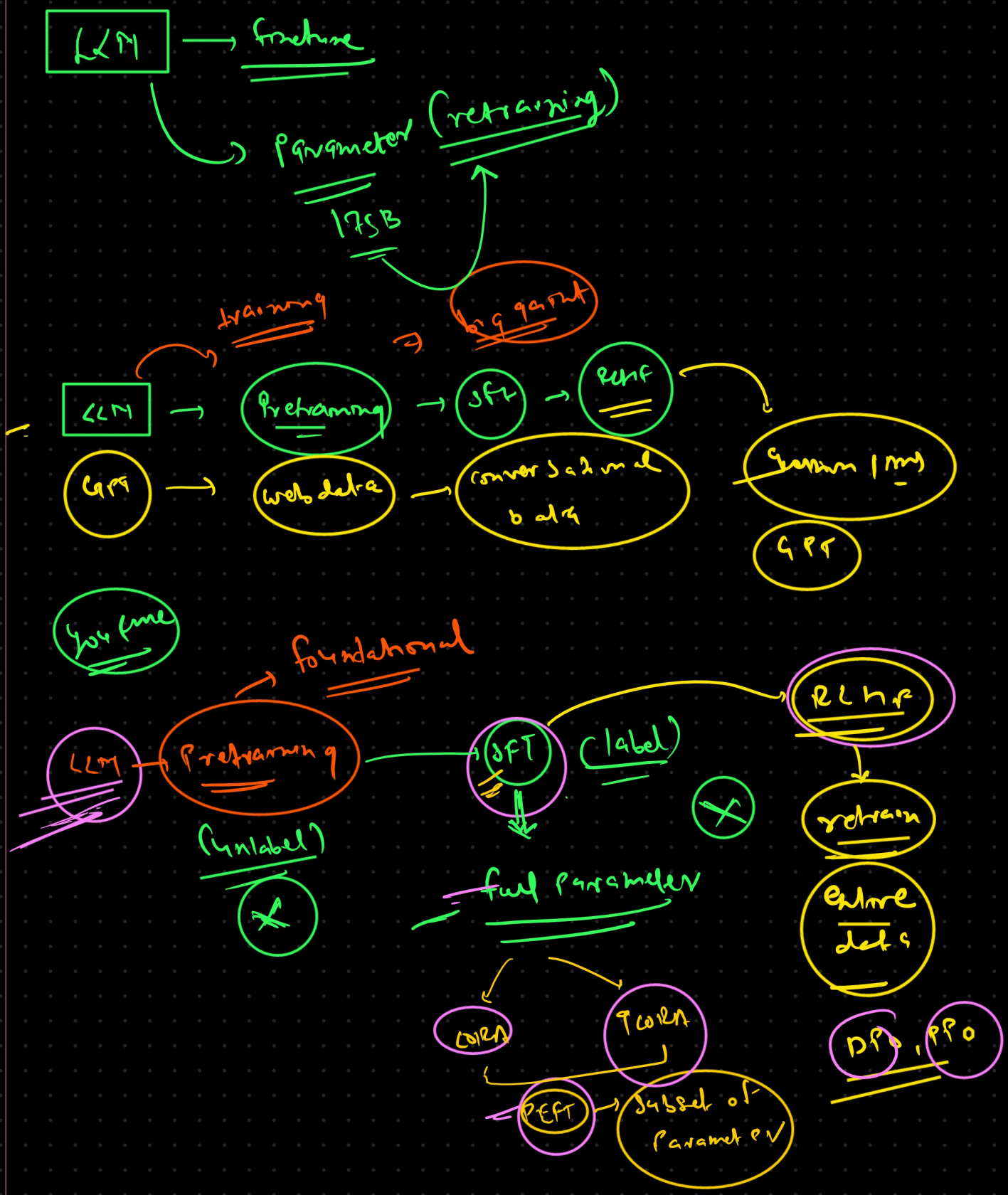
(Billions)



= (gpt, Llama, mistral)




(Foundation model)
Data
transformer
Unsupervised pretraining
(Labelled data of training)
- Supervised fine tuning
RLHF
(community)
LLM
Retrain
(wrong)
medical data
Model
Fever, basic allergy
o/p
x
fail
Corona Data
Data
web
(will be adjust)
Corona data
model
new version of model
- finetuning
retraining
new data
(adjusting web)
Parameter



finetuning → retraining ⇒ Params ⇒ Data

↑

Domain specific

LLama, mistral

gpt => model

SFT => (instruction finetuning)

Question Answer

Finetuning

Parameter (w&b)

Retraining

Domain specific data

NN => train => Saved => Load => retrain

mne

(100)

New data

I can retrain => full / partial => last

=> last 2 layer

finetuning

LM v/s LLM

- RNN
- LSTM
- GRU
- Attention

transformation

huge data

LLM (Bert) — Million

T5

BART, GPT — [illegible]

Gemini, Llama, GPT-4

Billion

pretraining => foundational

LLM => X

Data => Not label => training

unsupervised training

- Bert → foundational → Supervised Pretraining
- Llama
- T5
- mistral
- ⋮
- N

API
Scratch

Load

Llama ⇒ foundational
↑ unsupervised

SFT

fine tuning

GPT → 3.5 → chatgpt

1. unsupervised Pretraining
2. supervised fine tuning
3. RLHF

unsupervised → SFT ⇒

SFT

Post tuning (answer) = RLHF

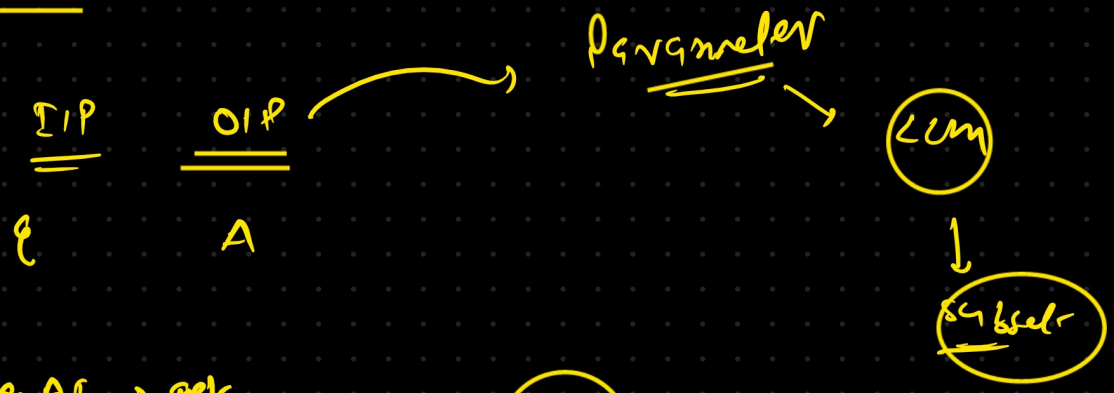
Custom data

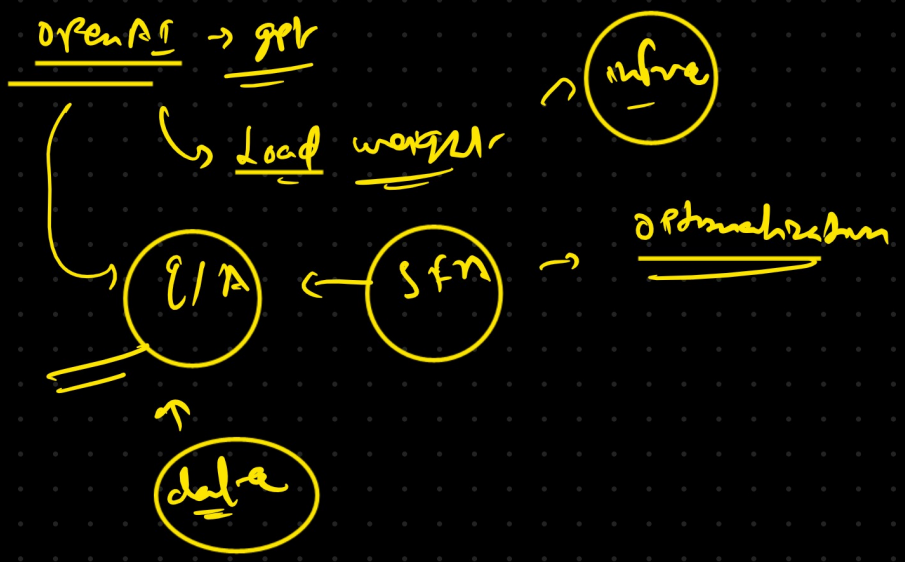
Label data

SFT
- → full finetuning
- → instruction (Prompts) → SFT
- → PEFT (LORA + QLORA) + quantization

SFT => Supervised Learning

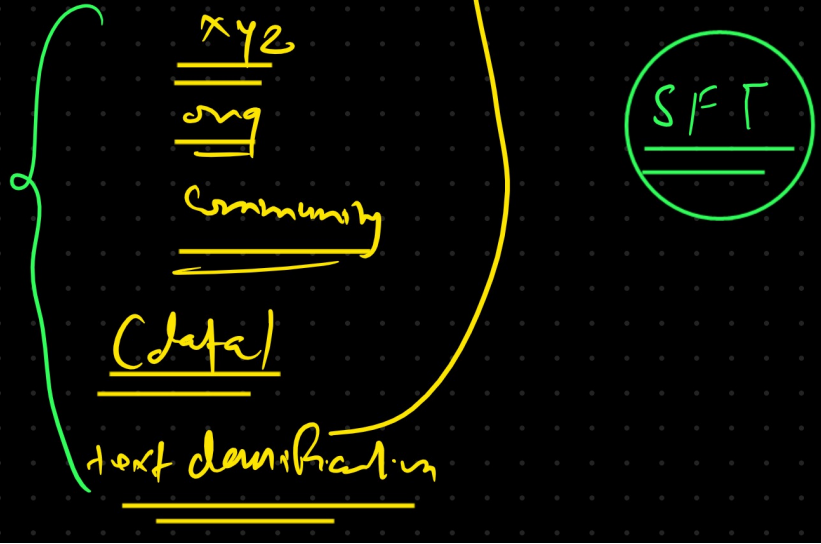
Label data





Foundational => Unsupervised model

Pretrain model

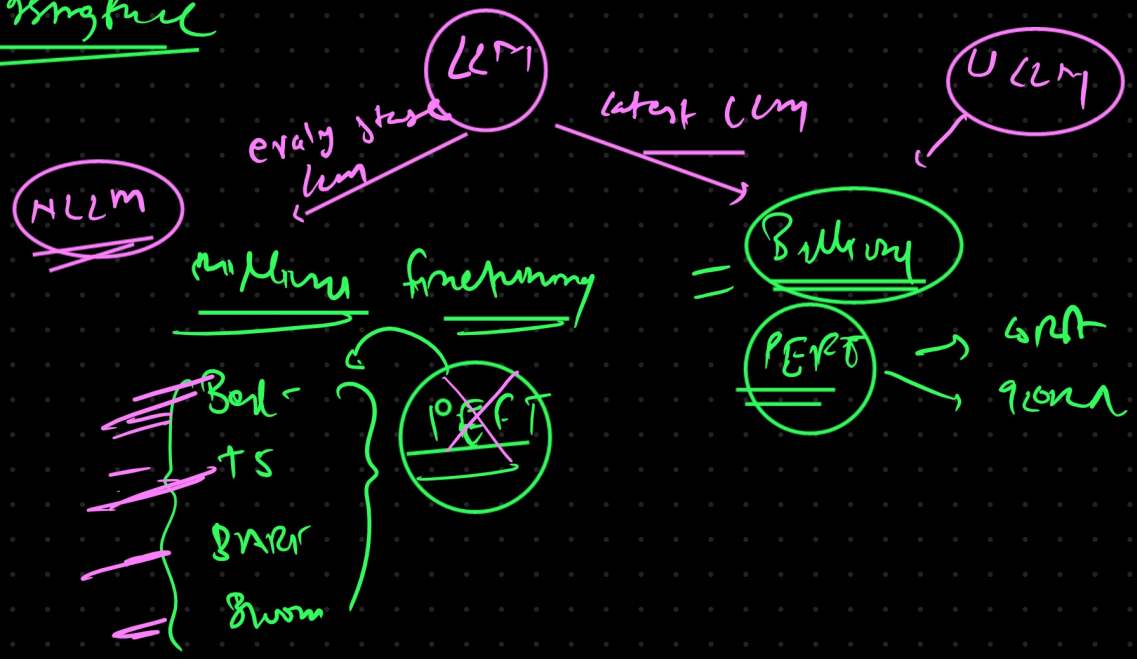


Question	Answer

RLHF

Llama mistral, Zephyr or any model

Fine tuning



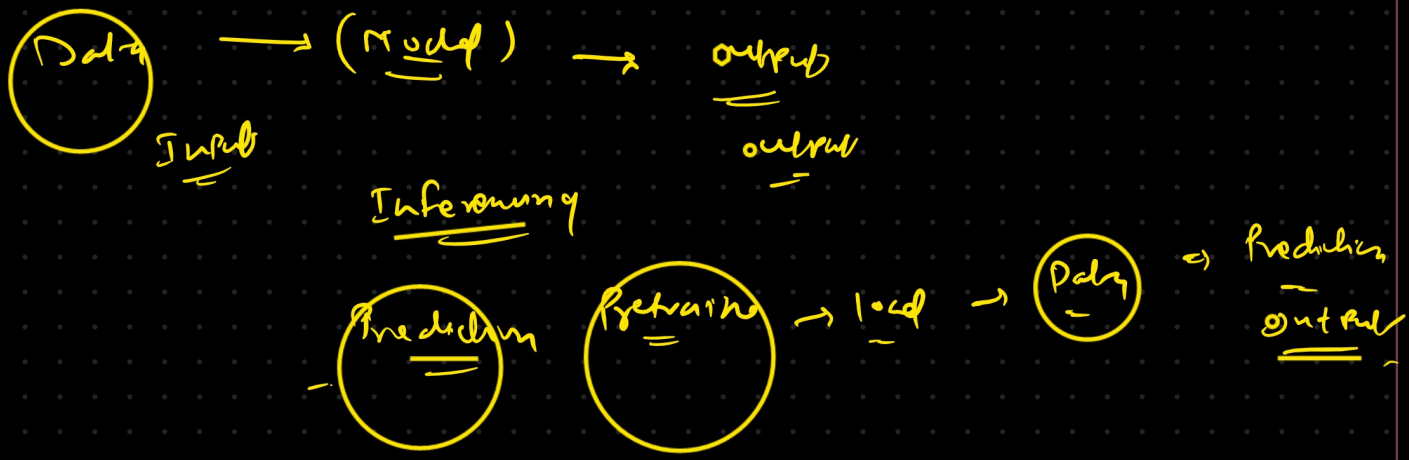
NN → train → save → load → fine tuning

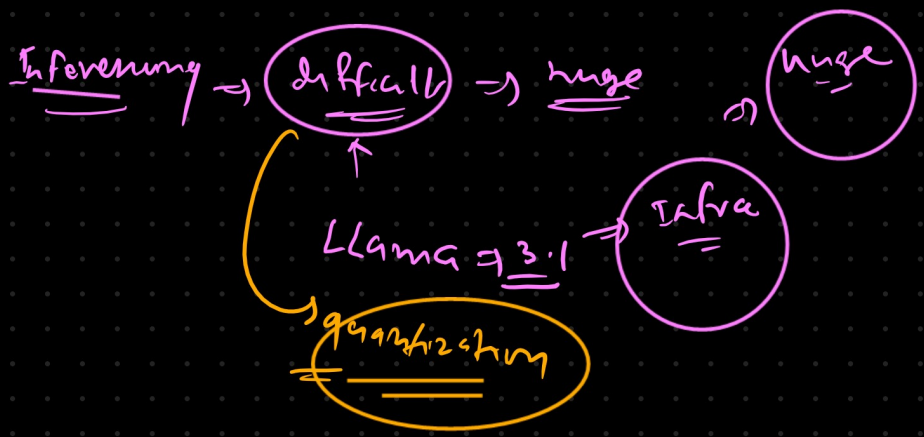
- Model ⇒ Pretrain (unsupervised Pretraining)

↓

load model (w4b) Param

↓





quantization → fast inferencing

(?)

Inferencing

↓

Prediction

Model → Head

↳ (w4B) → Parameter

Access → API ← Infra

Name

- Bing (hDD, sDD)

BERT

① Inferencing

LM → LLM

↓

B

BERT

↓

empty

Llama 3.1 → 405B →

(Quantize)

② Finetuning

Next

SFT → PEFT (LoRA, QLoRA)

BERT. (GPT-3.5, LLama, Mistral)

↓ ↓

Mistral

↓

~~PEFT~~

Pretrain

↓

SFT

LoRA, QLoRA

SFT → PEFT

(LLM)

LM → ① (web) Unsupervised Pretraining (Self-Supervised)

(② SFT)

③ RLHF → Preference → DPO
→ PPO

Reduce the size.

ML ⇒ (DT) → Pruning (Reduce the size)

DL ⇒ CNN ⇒ Quantization (Awd. QAT) LLM

⇒ Distillation -

↓

GGUF, GGML, GPTQ

Quantization ⟶ Parameter (w, b)

w = float32

storage size

(w = 23.72) → (float32) → = (bits) ⇒ (1 byte)

↓

(4 Byte) = 32 bit

↓

= 8 bit

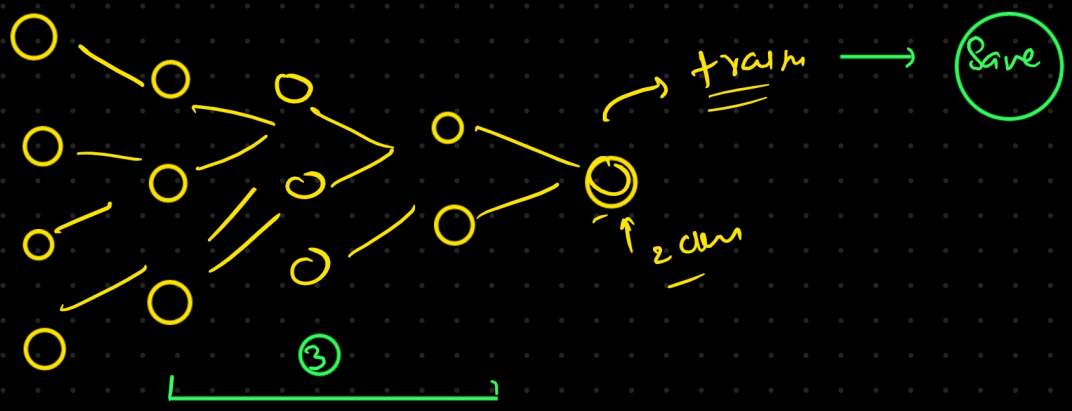
float 8, 16, 32

int 8, 16, 32

↓

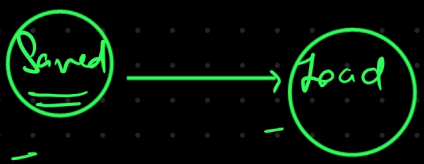
memory → RAM

= Model.



$(4 \times 3 + 3) + (3 \times 3 + 3) + (3 \times 2 + 2) + (2 \times 1 + 1)$

$12 + 3 + 9 + 3 + 6 + 2 + 2 + 1 = 38$



Next task ⇒ 3 classes

Saved → Load → 3 class

2 class

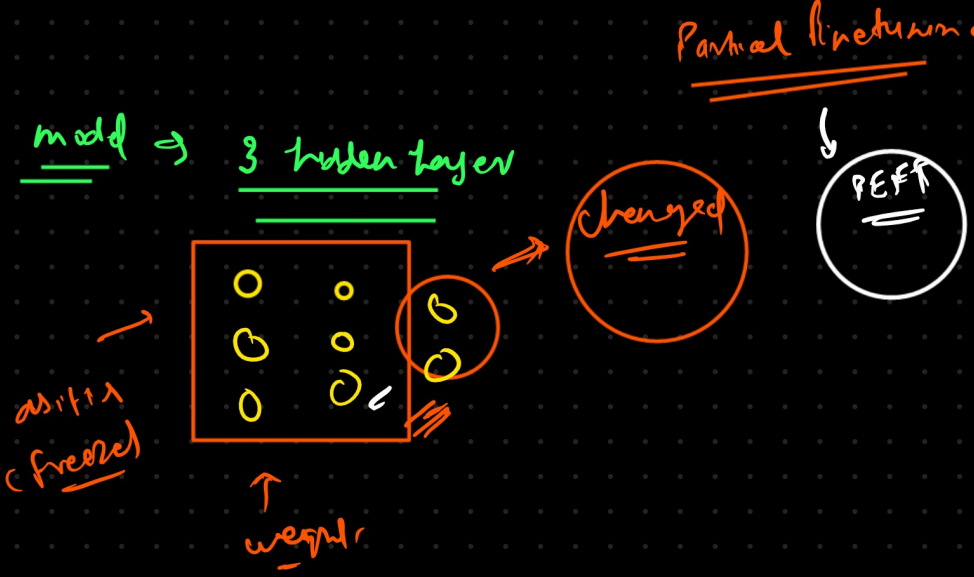
output layer (last year)

transfer learning

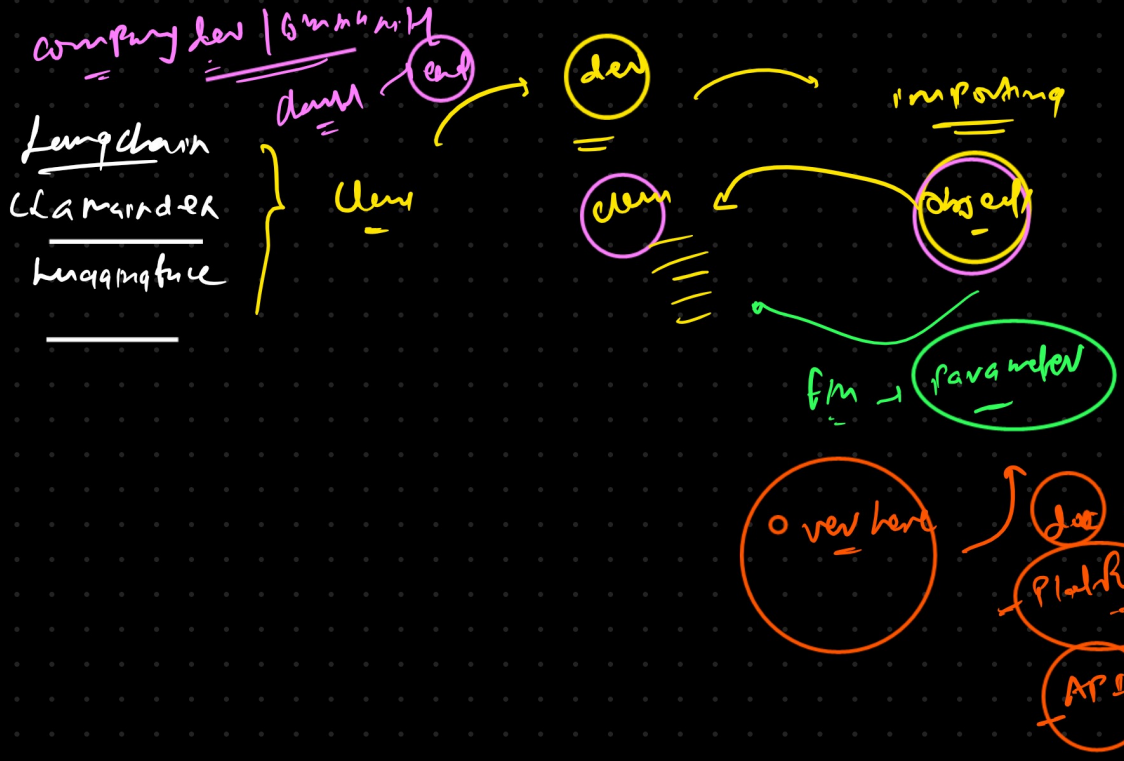
Saved (trained) → As it is → (0.8) → Output layer

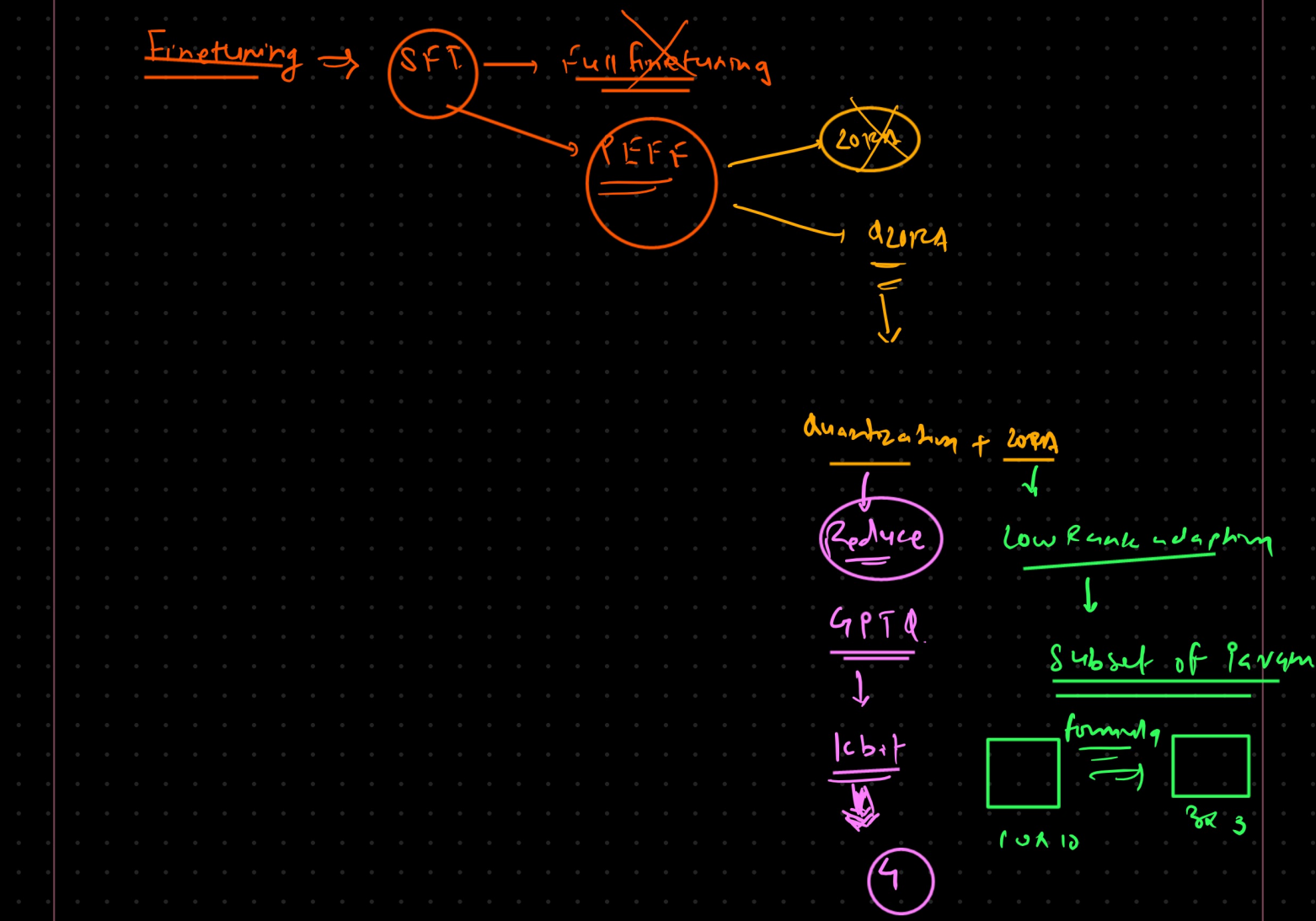
Rest of layer keep as it is (freezing)

Transfer learning



3 layer → full finetuning




Finetuning
SFT
Full finetuning
PEFF
LoRA
QLoRA
Quantization + LoRA
Reduce
Low Rank adaptation
GPTQ
Subset of params
4bit
formula
for 1
for 3
4
mistral
Pretrain model
SFT
label data
PEFF
QLORA
in our case
quantization + LoRA
config
config
Activation
(W x b)
loss
equation
NN
evaluating
equation - params
equating
Update
Optimizer (Loss)
(Adagrade, Adadelta, RMSProp, Adam)

epochs, batch size

1. implement LoRA in BERT finetuning
2. perform PEFT → QLoRA over the zephyr finetuning → LLM

optional (Partial)
{create your own repo over the huggingface}

GPT 3.5 turbo
PaLM

Save & load ⇒ inferencing
↓
Data specific
(working fine or not)

Sat & Sun
{4 5 oct}
→ generative

Data

Sir, Can we prevent pretrained model knowledge from finetuned model (for specific chatbot model)

(Prompting)