https://us06web.zoom.us/j/86750972600?pwd=H8FQrU7a9FCIBG9fhRylcBctYNQvCV.1

Finetuning

GPT → 3.5 turbo ← OPENAI API

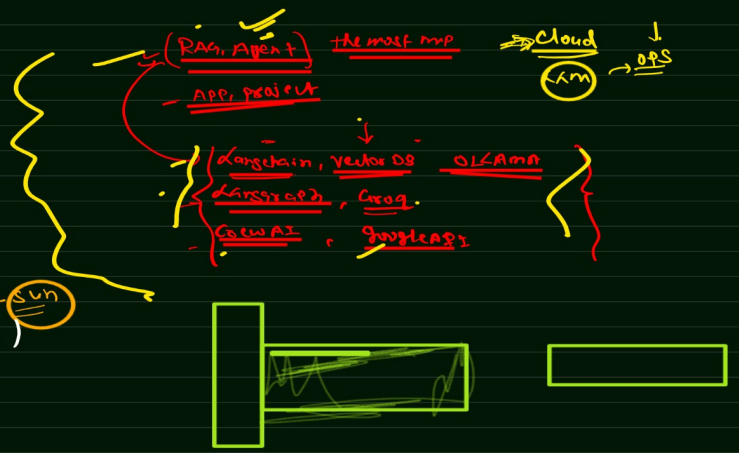
Llama, Phi ← microsoft

⇒ modular coding

Some theory points on LoRA, PEFT, Quanti-zation

Reward learning (RLHF, Preference optimization)

Ready made ⇒ Unsloth, Unsloth



(MARI PPR)

1. Pretraining
2. SFT
3. RLHF

Supervised data

IIP OIP

LLM ⇒ Challenging

Parameters

PEFT ⇒ LoRA, Quantization + LoRA ⇒ QLoRA

W → huge numbers

entire weight

frozen (freeze)

Freeze

$W' = W + \Delta W$

$\Delta W = (A * B)$

A

B

A(r × d)

B (d × r)

↳ A

d ⇒ dimensions

r ⇒ low rank factor

Interview question

LORA ≈

32 GB RAM

QLORA =

12-24 GB RAM

Transformer ⇒ self Attention + NN

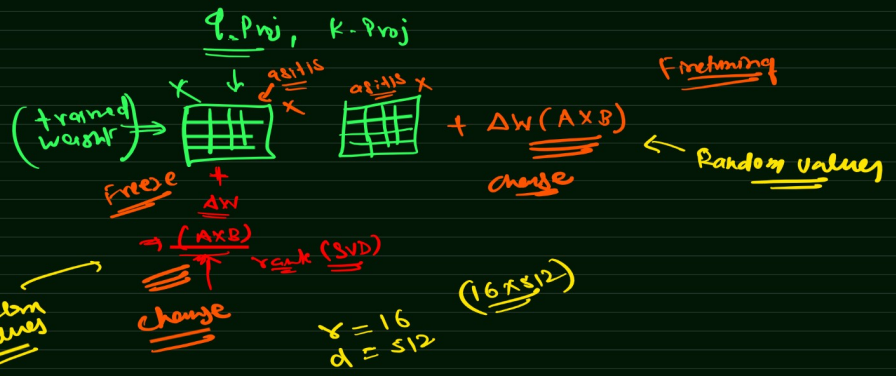
huge number weight

LORA subset

(weight)

(A × B)

Rank ⇒ (SVD)



1 BIT ⇒ 0 or 1

1 BYTE ⇒ 8 BIT

1 KB ⇒ 1024 BYTE

1 MB ⇒ 1024 KB

1 GB ⇒ 1024 MB

System ⇒ 0 & 1

Datatype ⇒ float 32 = 32 bit = 4 byte
float 16 = 16 bit = 2 byte
float 8 = 8 bit = 1 byte
Int 8 = 8 bit = 1 byte

Datatype

weight ⇒ 2.87 ⇒ float32 ⇒ 4 byte

GPT3.5 ⇒ 175B weight = 175B × 4 byte

Quantization → reduce precision FP32 → Int 8

4 Byte → 1 Byte

(PTQ)

1 Post-training quantization

1. First trained in full precision (FP32)
2. After training the w & b are converted to lower precision
   - FP16
   - Int8

(QAT)

2 Quantization aware training

1. While training use quantization effects
2. Instead of training on full precision the model will train with quantization effects

today's → LoRA + QLoRA ⇒ (PTQ) QAT

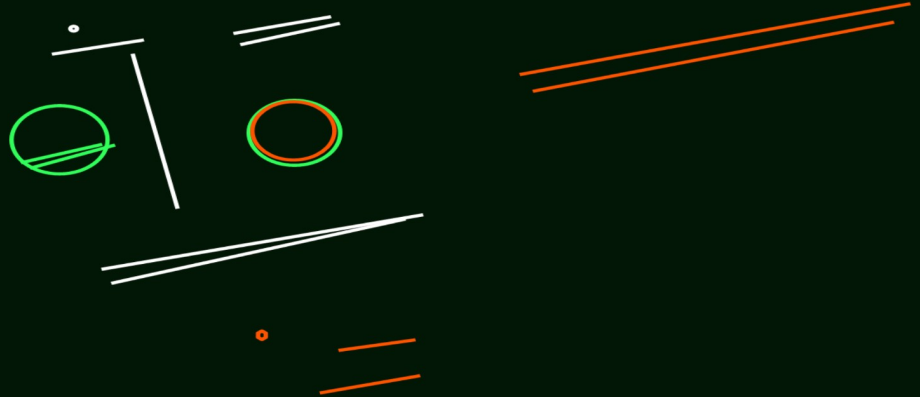
(PQT) → under precision (FP32) ⇒ 1. Pretraining ✗ ⇒ loaded the pretrained model
2. SFT ← Quantization

PQT → Post Quantization technique

1. unsupervised ← FP32 → full precision
2. SFT ← quantization

```
bnb_config=BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
```

← normal float 4 bit (NF4 bit) ← fp32



Finetuning → Realtime

DPO = RLHF → Chatgpt

Today

RAG (Realtime) = Agents

Sunday

Projects

LLM → training = OpenAI

RL ← PPO ✗

Chatgpt

DPO ← Simpler Preference based learning

PEFT ← LoRA, QLoRA

quantization (GPTQ, GGML, GGUF)

Llama, mistral

Unsupervised Pretraining ← Raw training, Entire Internet Data

RLHF

RL ⇒ ✗

DPO ⇒ ✓

Roadmap

Bigger Picture

RL = Agent ← model ← software (mathematical)

- State
- Action
- Reward - +ve, -ve
- Environment

RL

1950
- Bellman eq.
- Dynamic Prog

Agent, Action, Reward
Environment

1990 - monte carlo method

1990-2000 Q-learning

2015 Google deep mind

ANN → DL + RL ⇒ Deepreinforment learning

1995 Polical based method
↳ Polical gradient method

PPO ⇒ OpenAI ← (chatgpt)(RLHF)
↳ Proximal Policy optimization

State of Art
RL

Reintorment learning
1. Q learning
2. DRL
3. Policay based method :- Policay gradient method
4. = PPO ← OpenAI
5. Actor - Critic method
6. KTO KLHF

(Kahneman - tversky optimization)

Data (supervised, unsupervised data)

DL = FP + loss + BP

Data ✗ Supervised / unsupervised

RL → Agent, Action, Reward → Env ← learning

RL
DRL
Deep learning

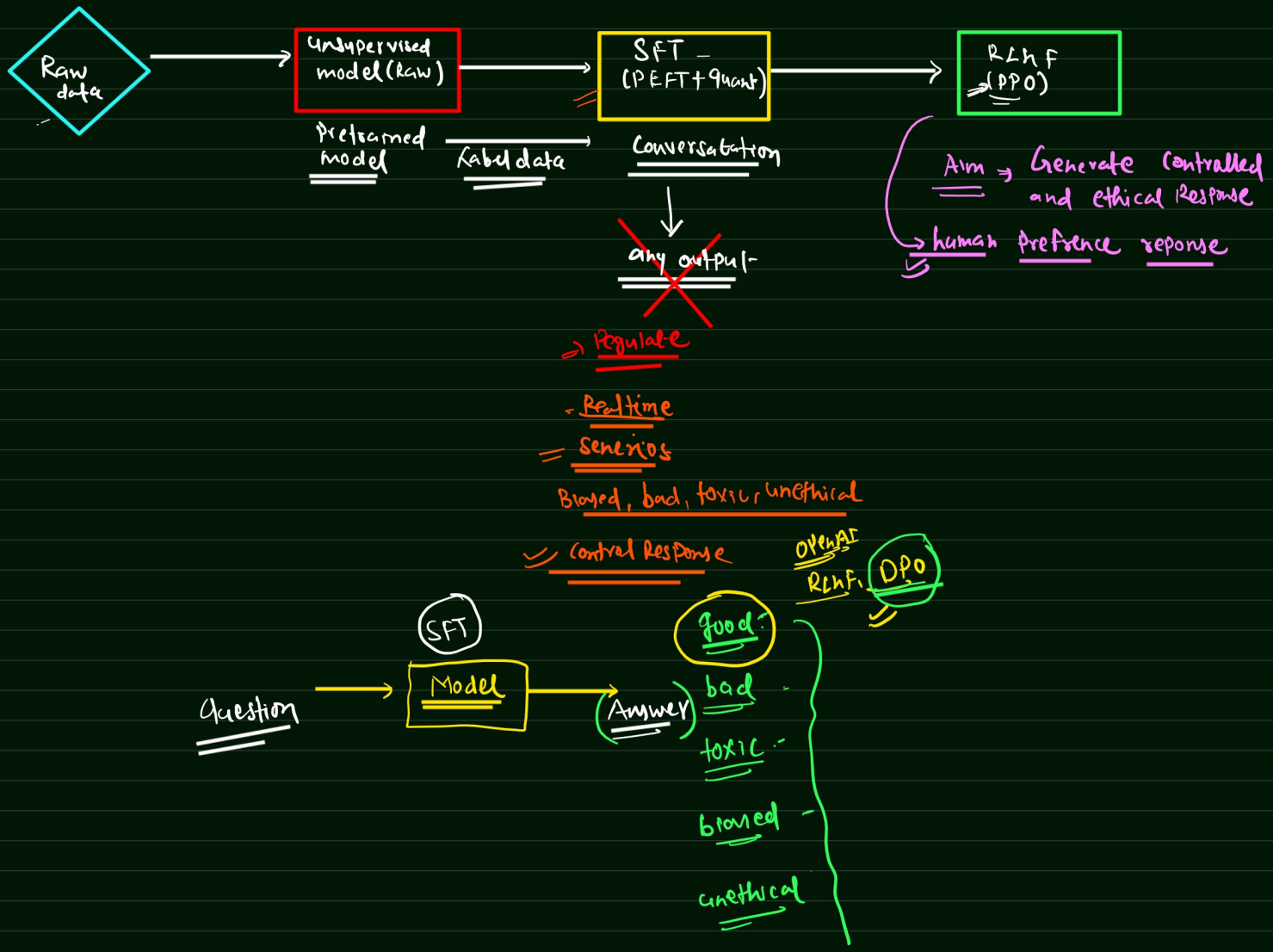
RL
(ppo, critic based method)

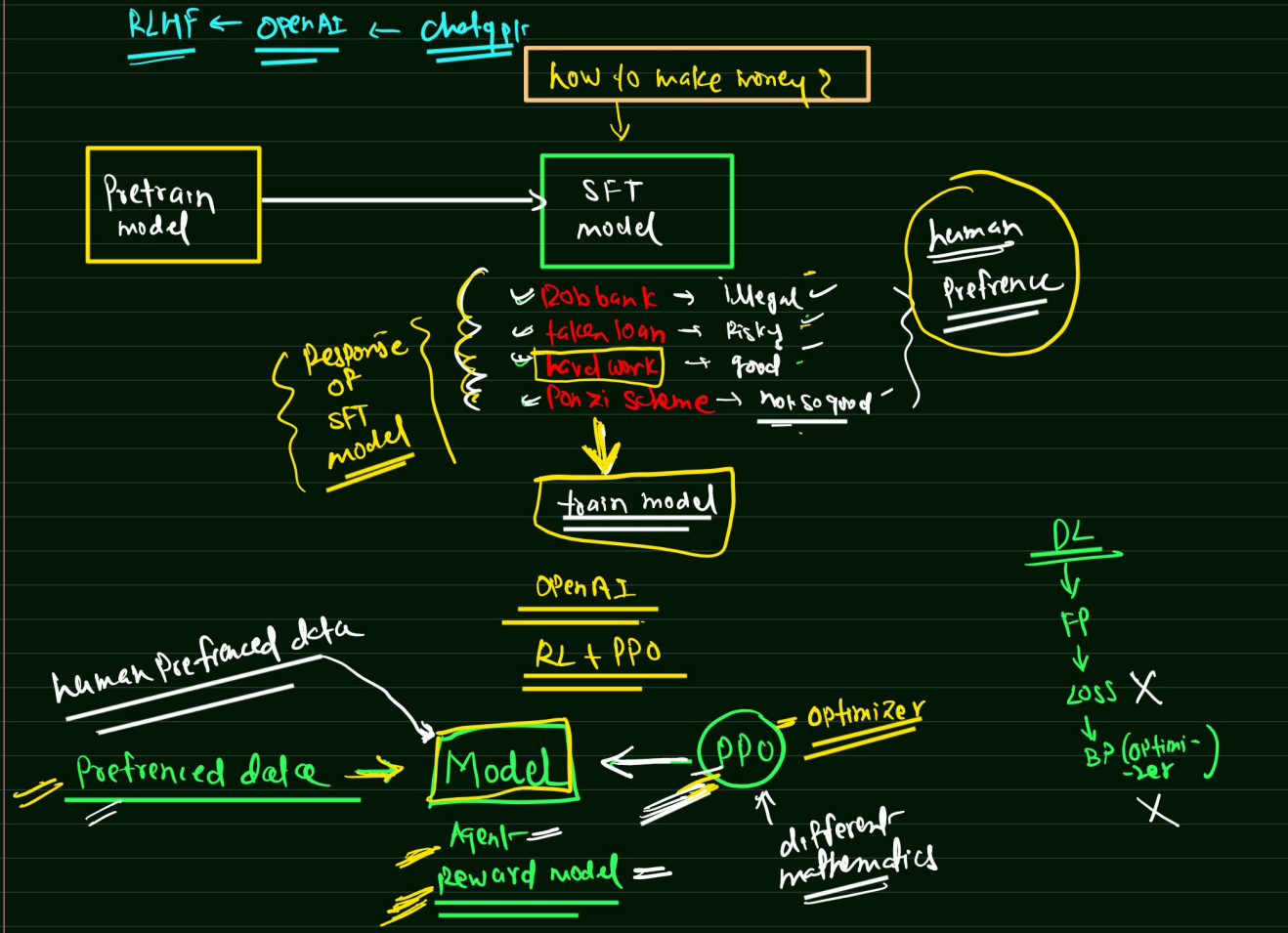
RIL
Self driving CAR
env Data

RLHF → training framework (Reinforcement Learning (PPO))

↳ openAI

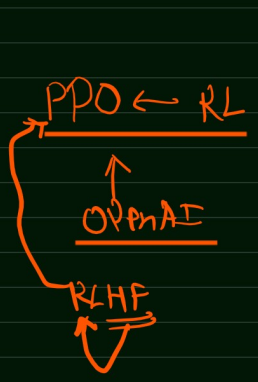
Reinforcement Learning from human feedback (Data)





RLHF Vs Normal PPO

Feature	Normal PPO (Reinforcement Learning)	RLHF (Human Feedback-Based RL)
Goal	Maximize numeric rewards (Game score, robotics control)	Maximize human preference-based rewards
Reward Source	Direct environment rewards	Human-labeled reward model
Application	Games, robotics, simulations	AI assistants, ChatGPT, LLaMA models
Output Control	Less control	More ethical & human-friendly control



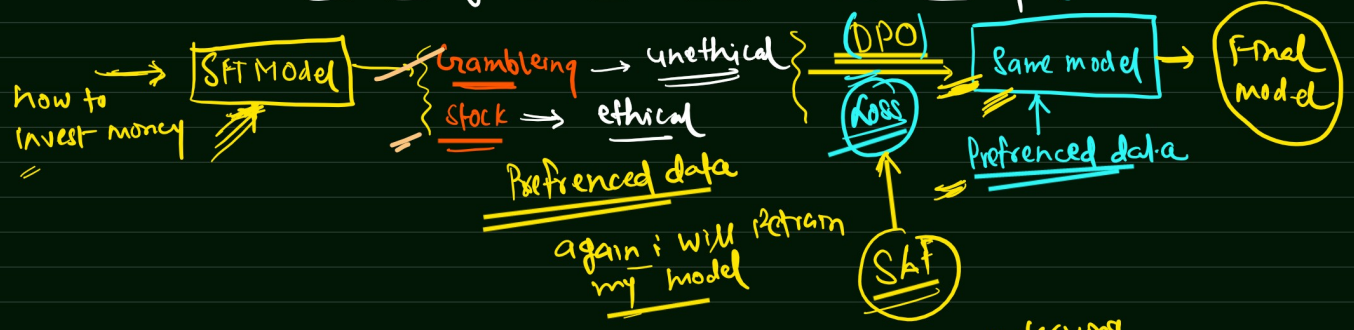
DPO ⇒ Direct Preference optimation

Supervised Loss function

Chat GPT

Retraining

DPo directly work for Preference based learning

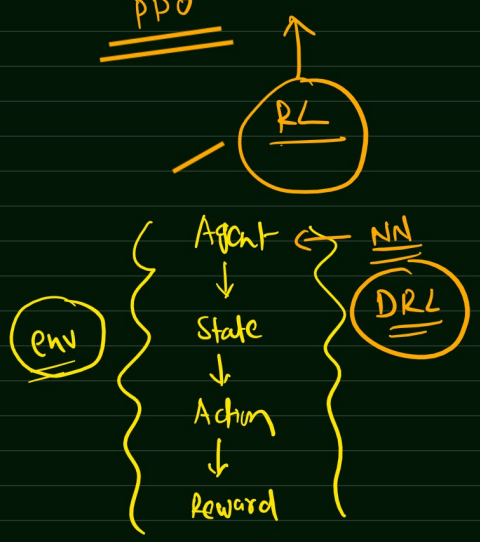


Old

OpenAI 2022-23

Proximal Policy optimation

PPO

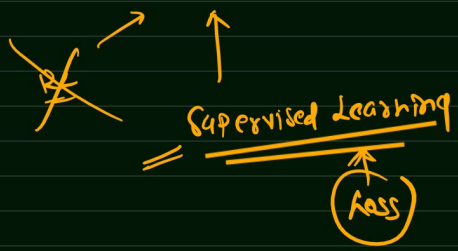


Qlearning, KTO, PPO, critic method

Latest

DPO

direct Prefrence optimization



{global minima} {minimum loss}

FP ⇒ calculation

Loss, MSE, MAE, CE

BP ⇒ optimize ⇒ update weight

GD, SGD, MiniGD, ADAGRAD, ADAM

Feature	Reinforcement Learning (RL)	DPO (Direct Preference Optimization)
Learning Type	Reward-based learning	Direct supervised learning
Algorithm Used	PPO, Q-Learning, Actor-Critic	Simple loss function
Training Complexity	Zyada tuning aur instability	Simple aur stable
Compute Requirement	High (Zyada tuning chahiye)	Low (Efficient & fast)