Pretraining

Self-supervised task

Internet
Stack Overflow
Github

Data Mixture
Corpus for training the data.

Res h ✓
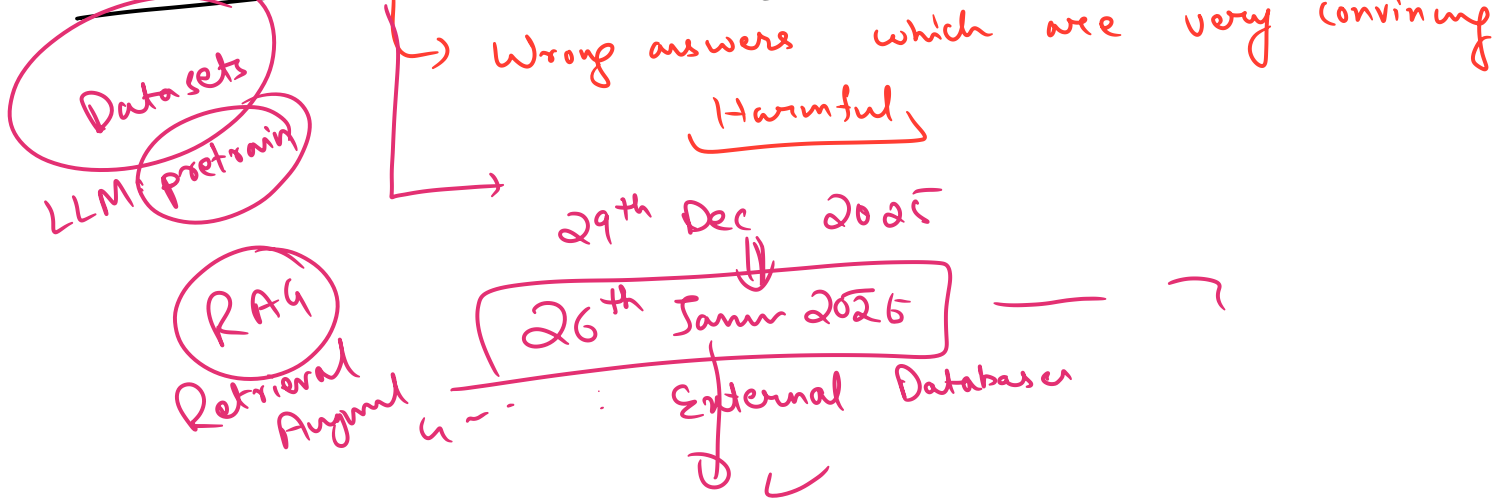Pre ✓

A   B   C   D

$P(W_{t_1} = W | c)$

(100)

8 words

Softmax f^n

Response Generation

Top-4
random.

Greedy
Beam   Approach

A B C

Decoder

h=2

Top-h   K

Memory

A B → D
→ ε

Sampling Tech
Words Variety   Top h
Token Generation

Config.

Risks R Prompt Env
Prompt Injection
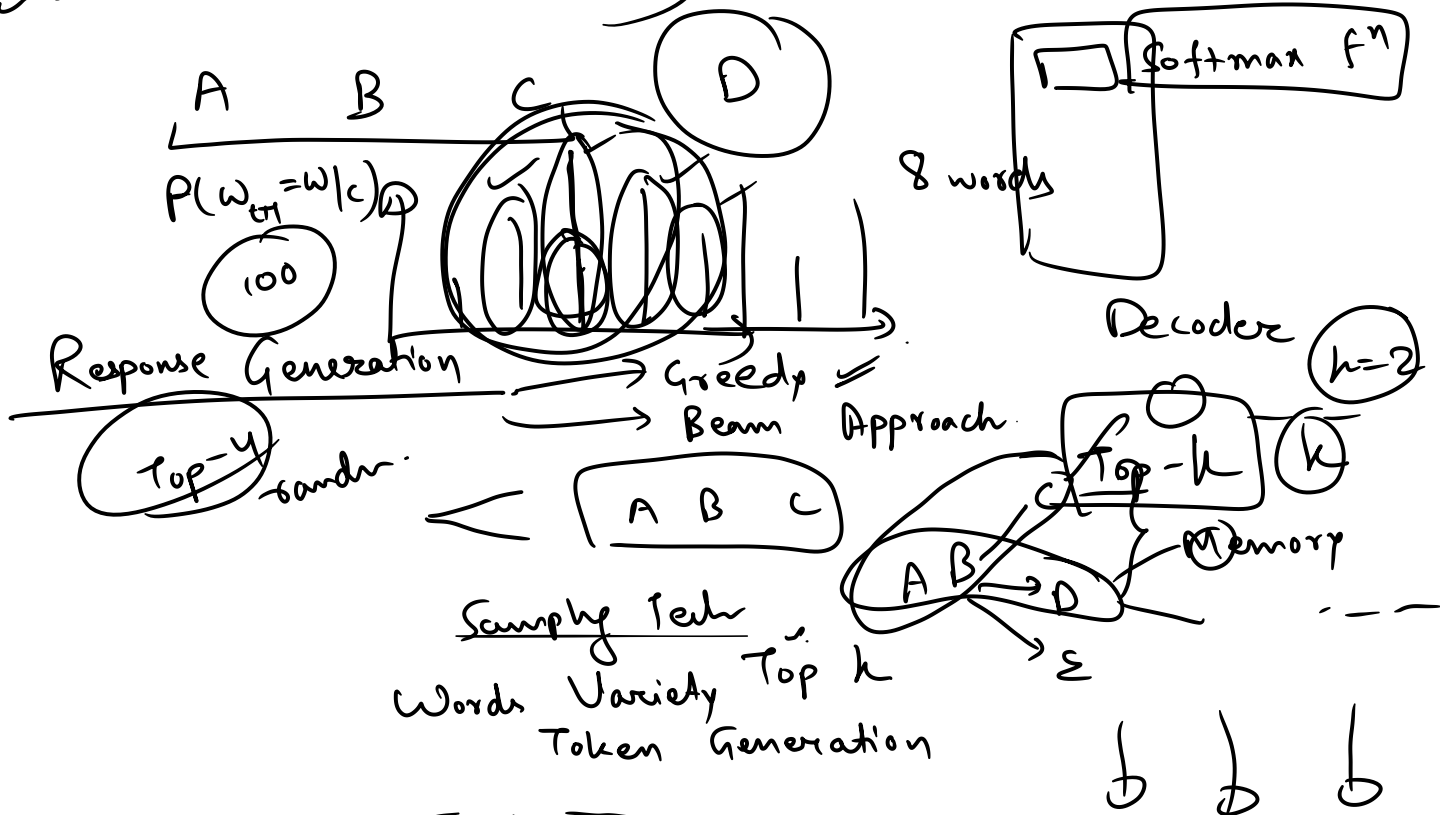
Model Hallucination   **Pretrain**

→ Wrong answers which are very convincing

Harmful

Datasets

LLM pretraining

29th Dec 2025
⇓
26th Janu 2025 — — ⌐

RAG
Retrieval
Augml

External Databases

x → Pretrain → $\hat{y}$ Output token

Input

Objective f^n ✓

LM : assistant

Text Generation

Pretraining → "tune it" fine-tung + Preference tung

" Alignment "

③

fine-tung

Supervised fine-tung

$$f_{SFT} = f_{pretrained}$$

Collect dat

x_i → y_i

prompts   responses

x_2 → y_2

Next Word Predn

x_i → f_{SFT} → $\hat{y}_i → y_i$

Back prop.
Update our weights.

f_{SFT} fine-tung

(SFT) : Data Quality ↑   Pr → Respon
                              Oh -ʃ

update: all the model weights & bias are
         updated.

Obs → pairs : 100-200
         hundreds ✓

fine-tun.   [quality] > [quantity]
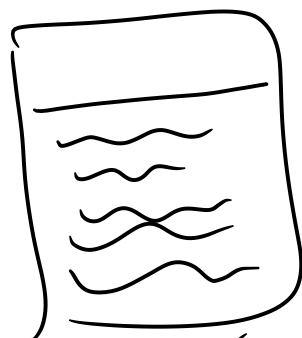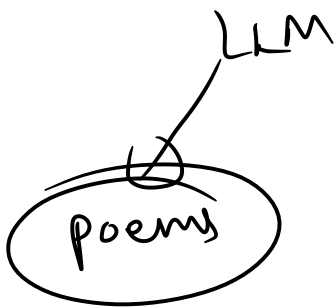
(Mimic)

SFT subset

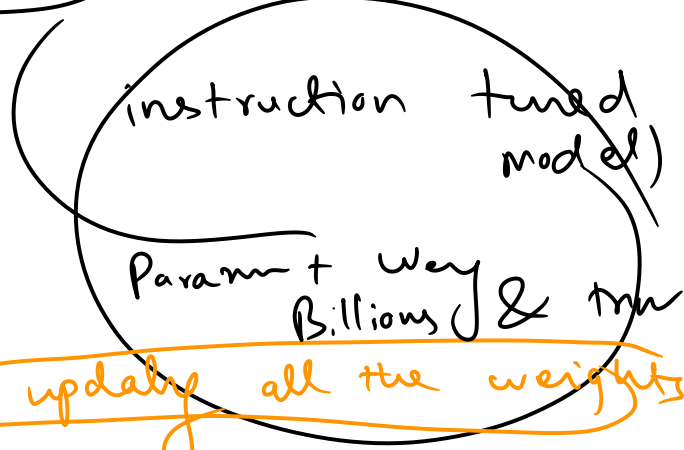## Instruction Tuning :

It is a fine-tune method that improves the
ability of an LLM to perform on unseen
                                        data.

LLM
(poems)
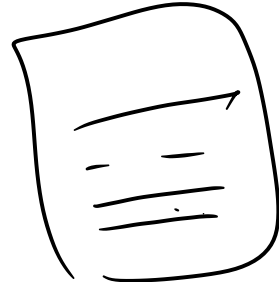
Challenges :

1) Very high quality data.
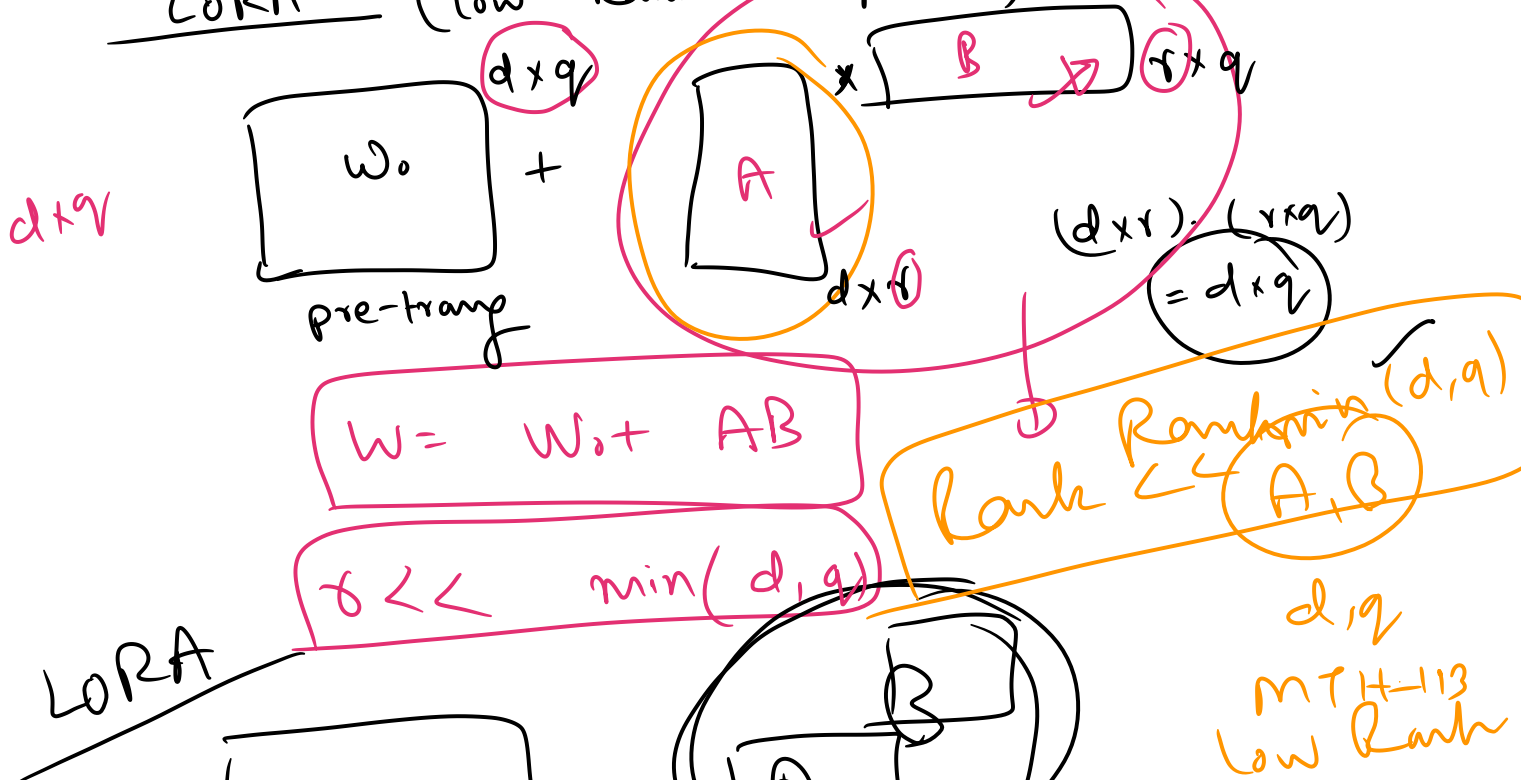2) Very sensitive prompt dist
3) Computatn Expensive

(instruction tuned
        model)

Param + Wey
    Billions & tun

Each time update all the weights

W, → (W,)

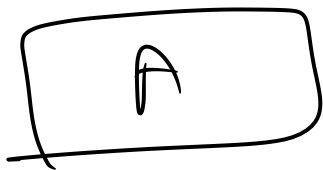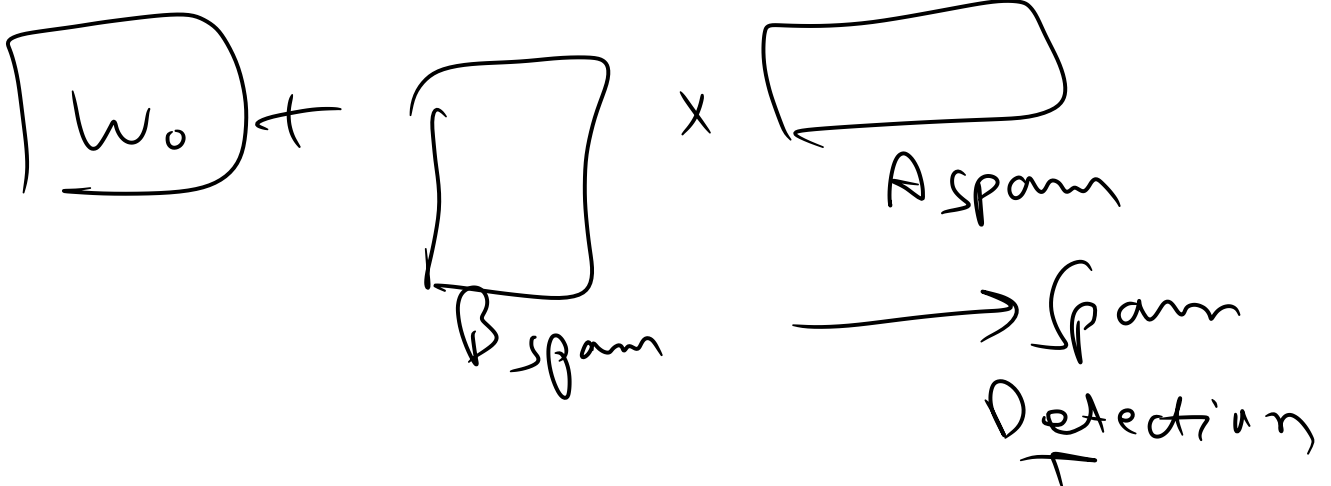# Parameter Efficient fine-tuning

Updation of only a (subset)
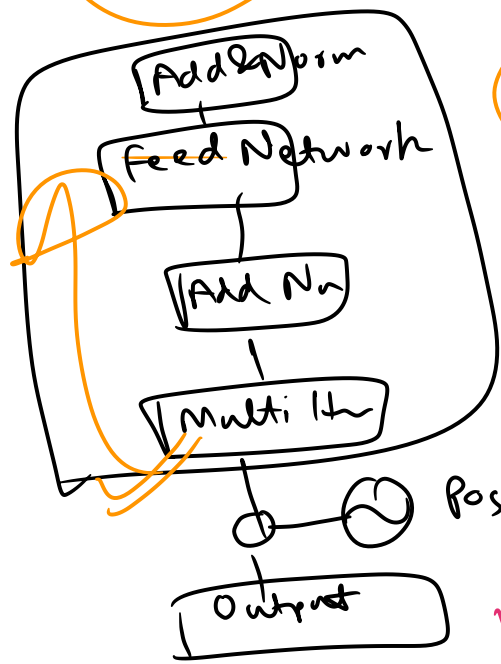
LORA (low Rank Adaptation)

$d \times q$ ⟶ W_0 $+$ A $\times$ B ⟶ $r \times q$

$d \times q$ pre-trang $d \times r$

$(d \times r) \cdot (r \times q) = d \times q$

$$W = W_0 + AB$$

$$r \ll \min(d, q)$$

Rank $\ll$ (A, B)

d, q

MTH-113 Low Rank

LORA

W_0 $+$ $\boxed{A \; B}$

Trang (fine-tung)

"freeze" $\boxed{A}$

$\boxed{A}$

fine-tn Spam Detection
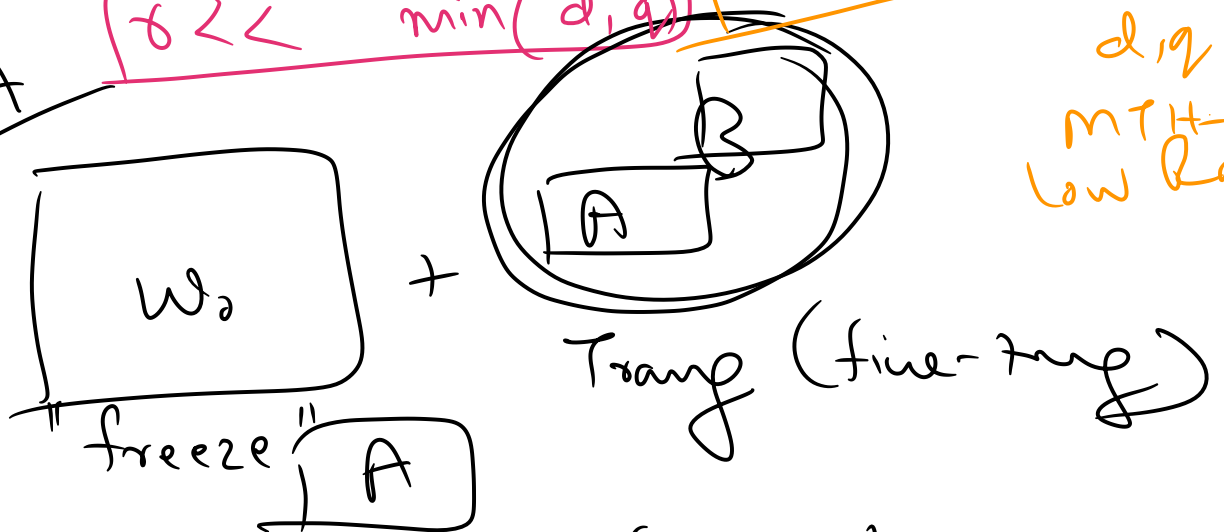
$\boxed{W_0}$ $+$ $\boxed{B \; spam}$ $\times$ $\boxed{A \; spam}$ $\longrightarrow$ Spam Detection

LORA Microsoft Experiment LORA features

Add & Norm
Feed Network
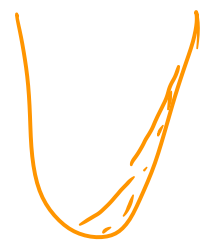**Add Nn**
Multi Ln

LORA P

→ LORA needs a higher learning rate than full fine-tuning

→ LORA does poorly on large batch size compared to full fine-tuning.

Posn

$$W' = W - \eta \frac{\partial J}{\partial W}$$

Rank ↓

Output

QLORA

weights, bias

7. 4107 — — ——

Add n & mantissa

x , +

Computational Cost ↓

Memory → Efficiency

Quantizahu

w: 7.4107
b: 2.310

32 bits

2^{32}
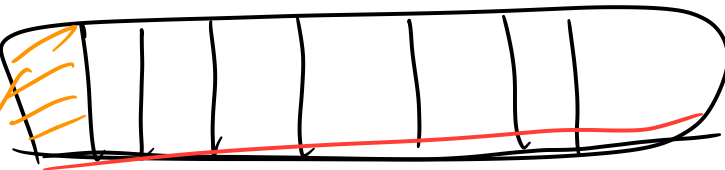
4)   8 blocks.   23 block : ma

32 blocks.

0-255

8 bits,   4 bits

$$0 - 2^8 - 1$$
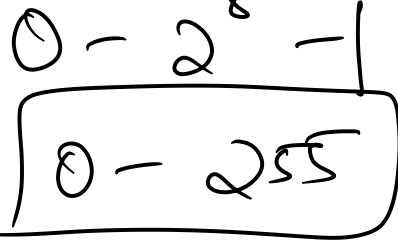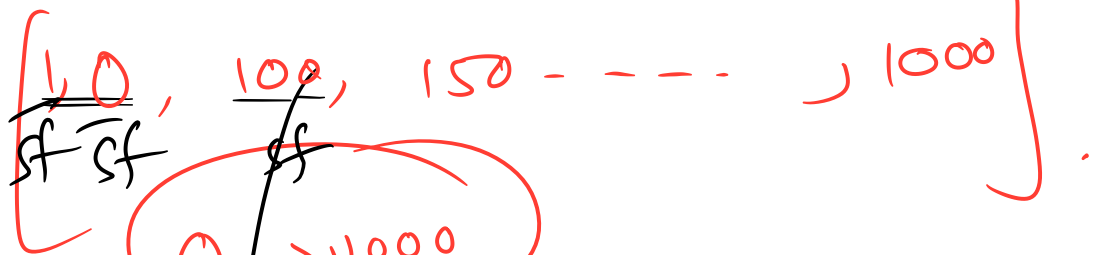
$$0 - 255$$

Weights: $[10, \ 100, \ 150 - - - - \ , 1000]$ 32 bits

sf sf   sf

binary

$0 \rightarrow 1000$

$0-255$

4 bits

min-max

Norm

$Sf = \dfrac{1000 - 0}{255 - 0} = \dfrac{1000}{255} =$ $3 - - - -$

$\dfrac{1}{1000/255} \Rightarrow \dfrac{0}{3}, \dfrac{100}{3} - - \ \ -----$

$-30, -10, 0, -50, 100$

$(-30) \to 100$
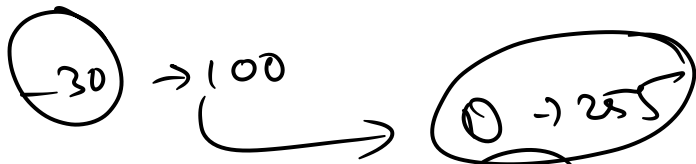
$\qquad \to (0 \to 255)$

$S.f = \dfrac{100 + 30}{255} = \dfrac{130}{255} =$

Compute to

$-30 \overline{\qquad} = - - \cdot - - + (\alpha) = \to 0$

$(0 R.f)$

$\dfrac{-10}{S.f} + \alpha \;,\; \dfrac{0}{S.f} + \alpha \;,$

Mem Sav
(an upaln
loss of
Infer.

QLORA

$(W) = (W) + (AB)$ full precision

W_2

fine tu $\{$ $32/16$ bits

4 bits

wan $\longrightarrow$ full

Pretrang

Post Trang
Quah

$W...$
$\to$ Qun

Quantization
Aware Trang

$-$

forward
(Quantization Error/os)