ways to adapt pretrained models to downstream tasks:

↳ in-context learning

↳ instruction tuning

---

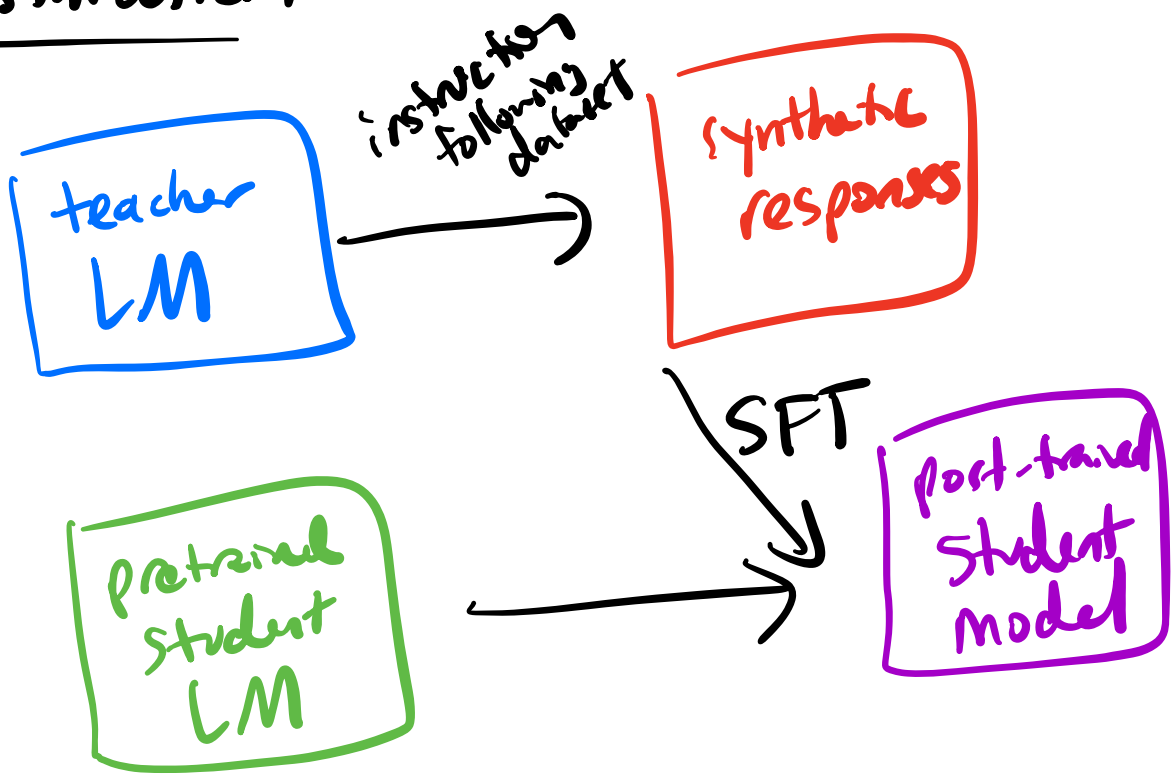
what can SFT actually do?

↳ can it add new knowledge | capabilities to the pretrained model?

↳ "superficial alignment hypothesis" : LIMA paper

↳ you can add new facts via SFT, but risks catastrophic forgetting

distillation



teacher: often a frontier model (GPT, Claude)

student: often a small open-weight model (Qwen, Llama)

↳ distill responses
chains of thought
attention scores
hidden states

how expensive is SFT?

↳ do we really need to compute $\frac{dL}{d\theta}$) or can we keep the majority of params frozen and just do a small update?

↳ "parameter-efficient fine-tuning"

↳ LoRA (low-rank adaptation) is the most popular approach

↳ freeze pretrained weights

↳ actually train a small number of new params

linear layer: $h = Wx$

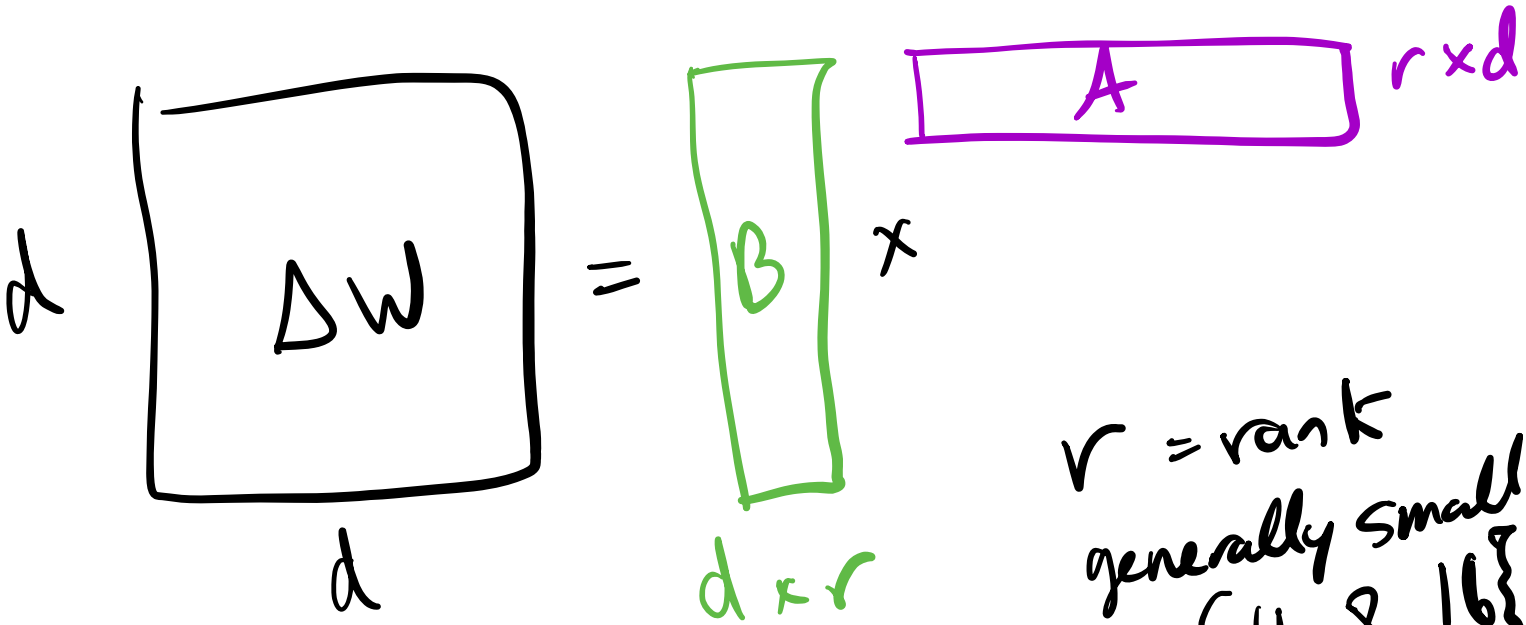
$\hookrightarrow d \times d$

$$\frac{dL}{dW} \rightarrow d \times d$$

gradient descent:

$$W_{new} = W_{OLD} - \nu \frac{dL}{dW}$$

$$W_{SFT} = \underbrace{W_{pretrained}}_{\text{frozen}} + \underbrace{\Delta W}_{\text{trained}}$$



r = rank

generally small

r: {4, 8, 16}

$$h = Wx + BAx$$

update

how many params are trained?

$d \times d$ matrix : d^2 for full finetuning

$2dr$ for LoRA

↳ $d = 4096$, $r = 8$ } LoRA trains 0.4% of the params

↳ which matrices to update?

↳ initially, the LoRA paper proposed to do this on FF layers, or just the attn

↳ now, all linear layers

↳ at test-time,

$$W_{SFT} = W_{pretrained} + BA$$

T

can be shared w/ anyone who downloaded the pretrained model