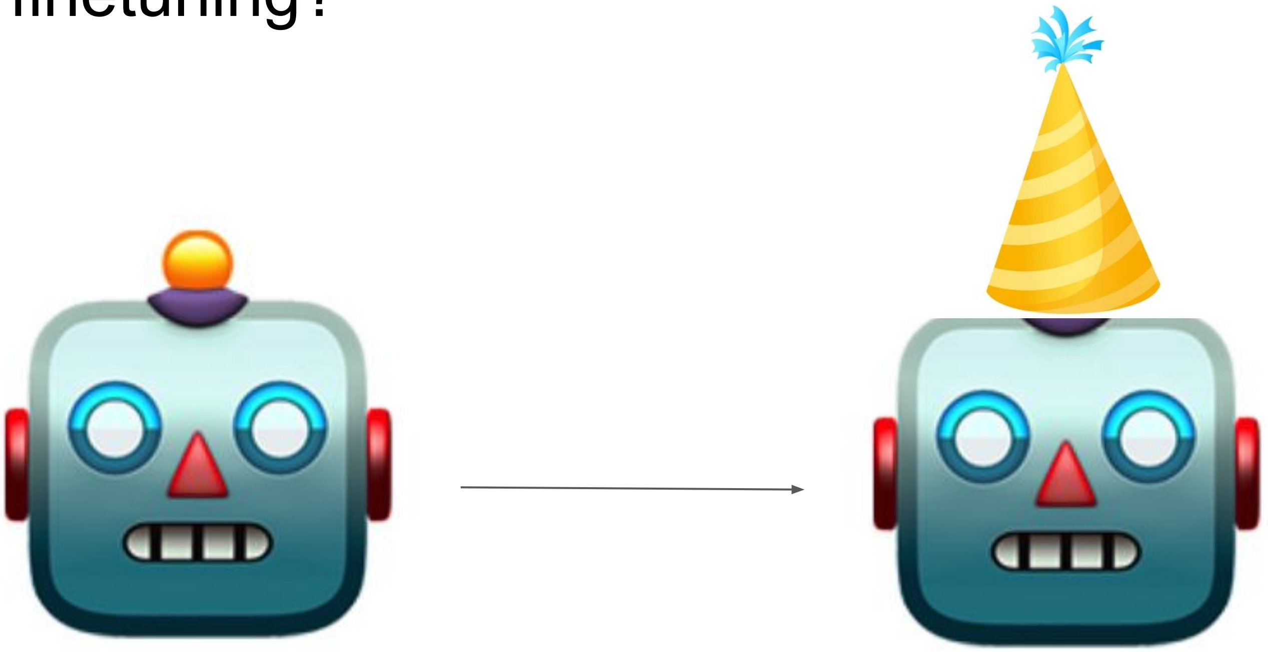
Parameter-Efficient Finetuning

CSE 493G/599G Recitation

Prepared by Scott Geng

What is finetuning?



Take useful model that already knows a lot and update it slightly

Can build applications cheaper, better.

Medical GPT

Can build applications cheaper, better.

(smaller)
domain
dataset

All of the internet

Medical GPT

Can build applications cheaper, better.

(smaller)
domain
dataset

All of the internet

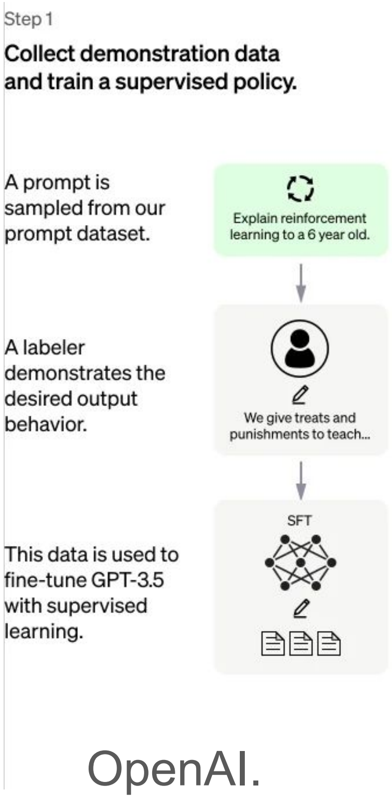
Pretrain
($100M)

GPT-4 **Base**

Finetune
($100k)

Medical GPT

Can build personalized applications.

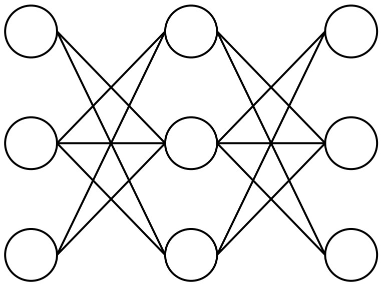


OpenAI.

DreamBooth.

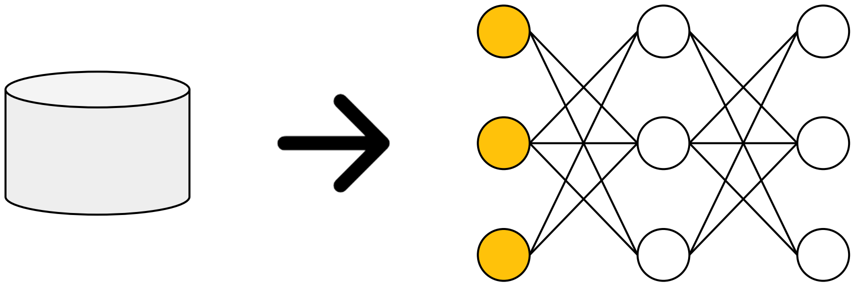
Vision: everyone should be able to easily adapt a very capable (very big) base model to whatever task they want

How to finetune a model



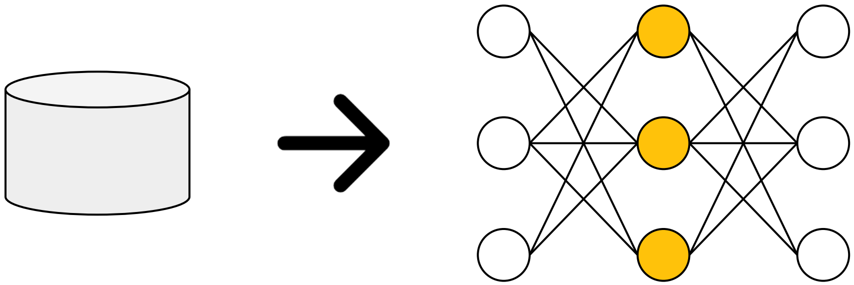
Slide credit to Tim Dettmers

How to finetune a model



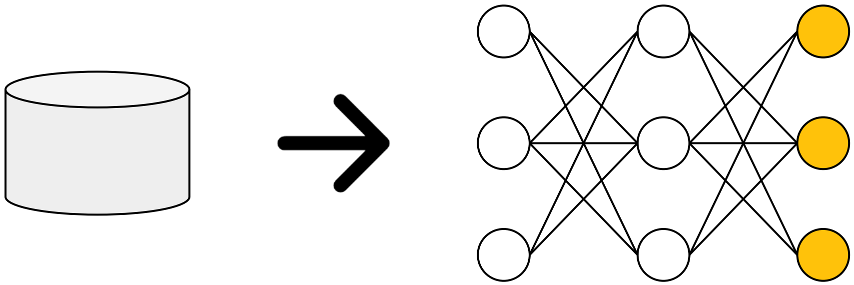
Slide credit to Tim Dettmers

How to finetune a model

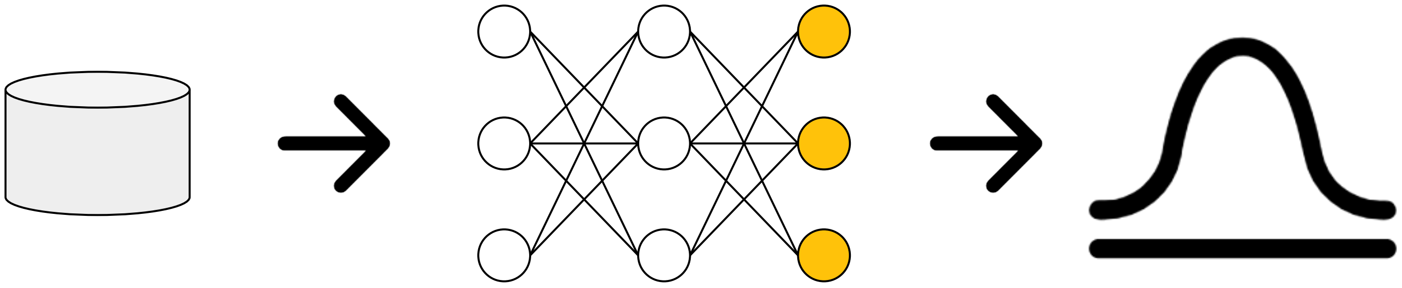


Slide credit to Tim Dettmers

How to finetune a model

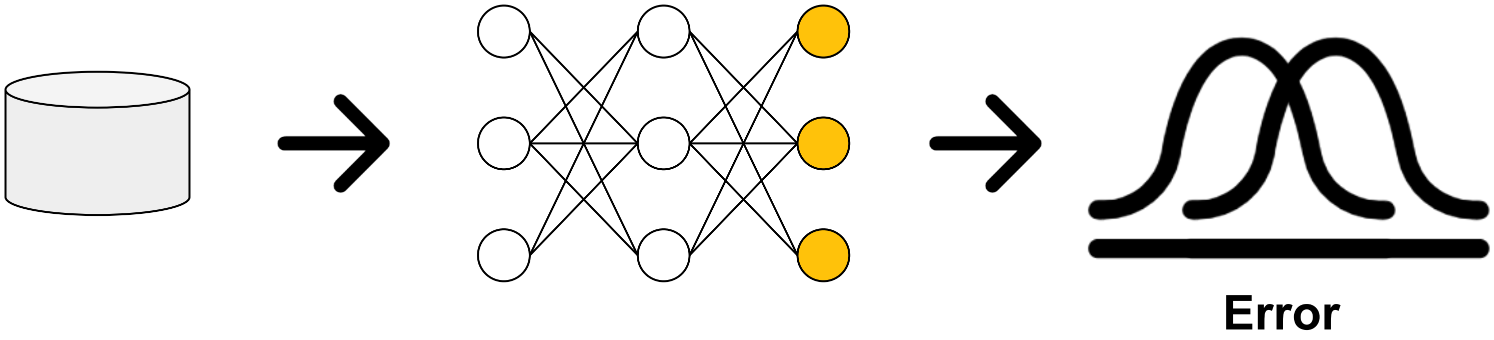


How to finetune a model



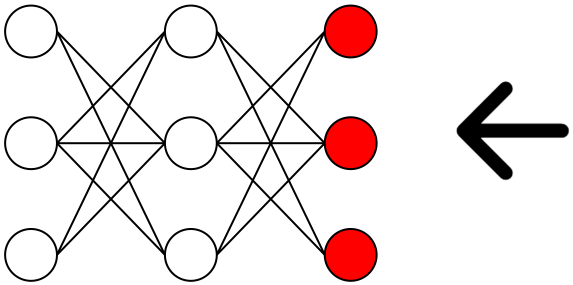
Slide credit to Tim Dettmers

How to finetune a model



Slide credit to Tim Dettmers

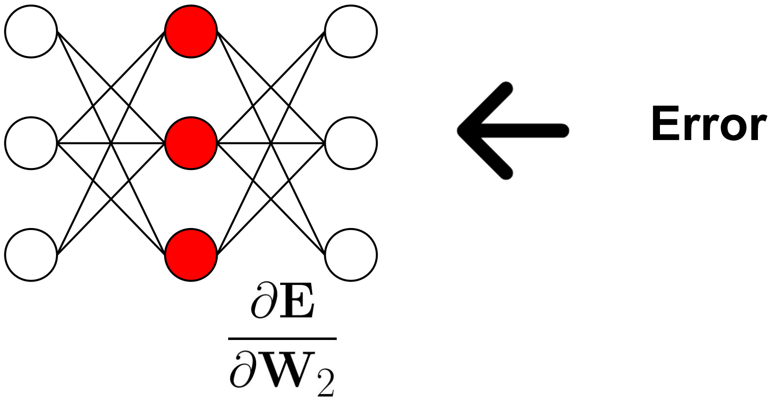
How to finetune a model



Error

Slide credit to Tim Dettmers

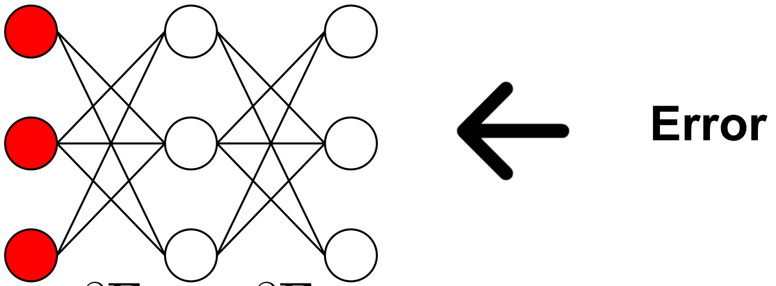
How to finetune a model



Weight gradients

Slide credit to Tim Dettmers

How to finetune a model



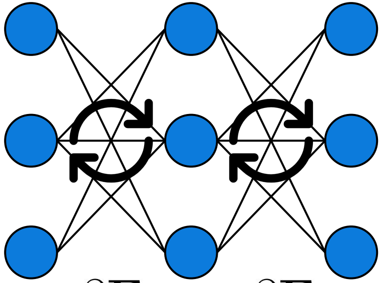
$$\frac{\partial \mathbf{E}}{\partial \mathbf{W}_1} \quad \frac{\partial \mathbf{E}}{\partial \mathbf{W}_2}$$

Weight gradients

Slide credit to Tim Dettmers

Background: How to finetune a model

Update the weights



$$\frac{\partial \mathbf{E}}{\partial \mathbf{W}_1} \quad \frac{\partial \mathbf{E}}{\partial \mathbf{W}_2}$$

Quality 

Slide credit to Tim Dettmers

Challenge: this is expensive compute wise.

100B parameters
base model
(~64-128 GPUs to train)

Challenge: this is expensive storage wise.

100B parameters
base model
(~200GB)

Each finetuned copy is same size!

Research problem: how can we reduce the cost of (1) finetuning a model and (2) storing the updated copy?

Low Rank Adaptation (LoRA)

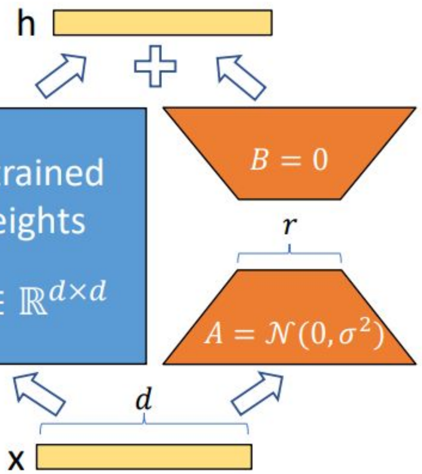


Figure 1: Our reparametrization. We only train A and B.

$$W_{\text{finetuned}} = W_{base} + \Delta W$$

$$h = W_{\text{finetuned}}(x) = W_{base}(x) + \Delta W(x)$$

Low Rank Adaptation (LoRA)

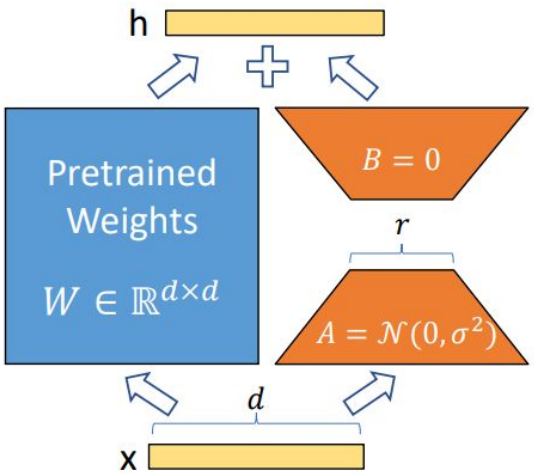
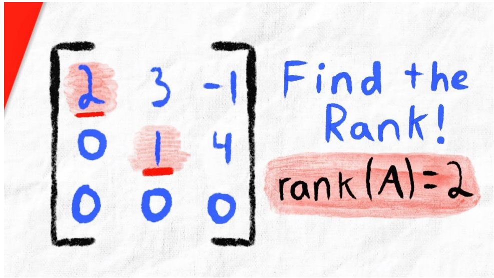


Figure 1: Our reparametrization. We only train A and B.

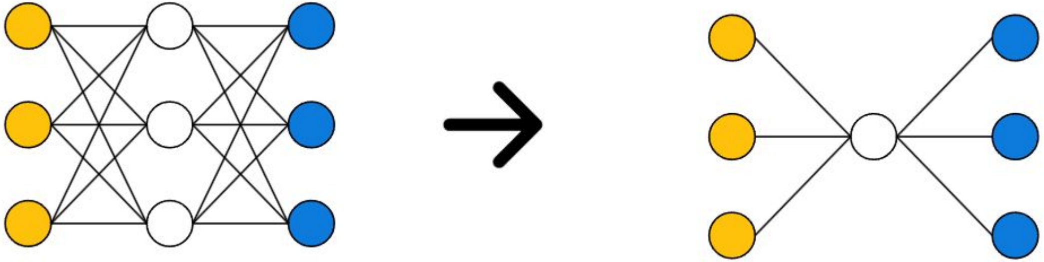
$$W_{\text{finetuned}} = W_{base} + \Delta W$$

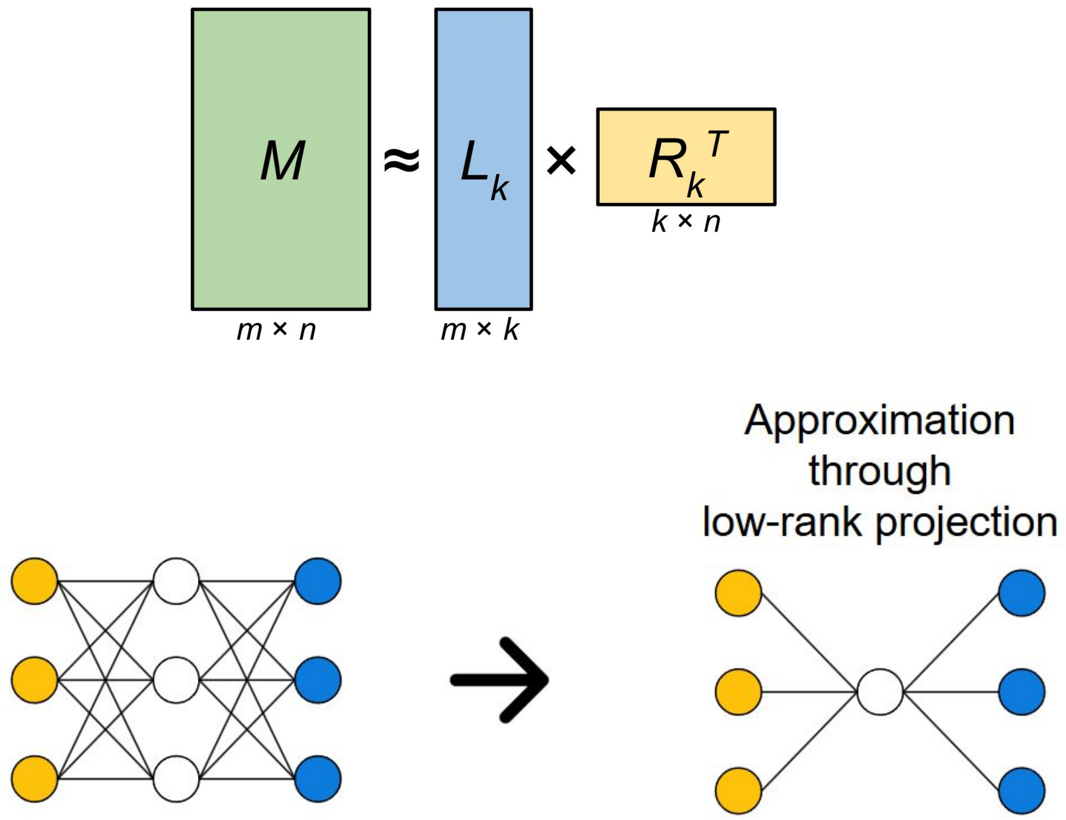
$$h = W_{\text{finetuned}}(x) = W_{base}(x) + \Delta W(x)$$

Key observation: deltaW has low rank, so that we can express it as a product of two simpler matrices



Approximation through low-rank projection




M
≈
L k
×
R k T
m × n
m × k
k × n
Approximation
through
low-rank projection

Low Rank Adaptation (LoRA)

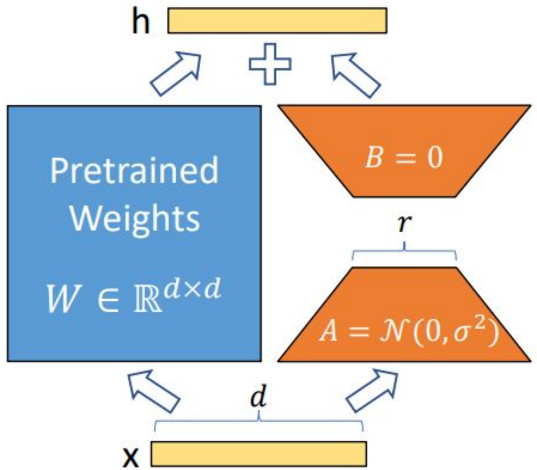


Figure 1: Our reparametrization. We only train A and B.

$$W_{\text{finetuned}} = W_{base} + \Delta W$$

$$h = W_{\text{finetuned}}(x) = W_{base}(x) + \Delta W(x)$$

$$\Delta W = BA$$

$$h = W_{\text{finetuned}}(x) = W_{base}(x) + BAx$$

Low Rank Adaptation (LoRA)

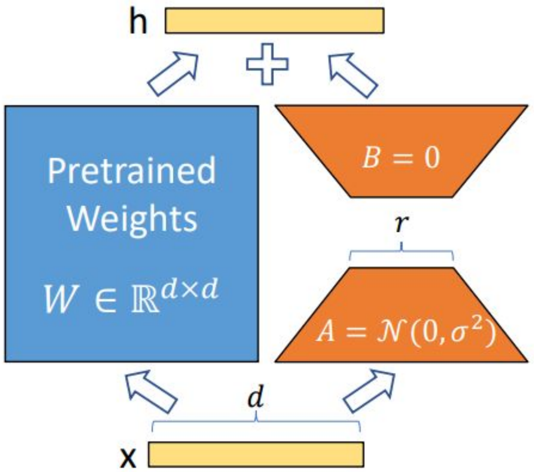


Figure 1: Our reparametrization. We only train A and B.

$$W_{\text{finetuned}} = W_{base} + \Delta W$$

$$h = W_{\text{finetuned}}(x) = W_{base}(x) + \Delta W(x)$$

$$\Delta W = BA$$

$$h = W_{\text{finetuned}}(x) = W_{base}(x) + BAx$$

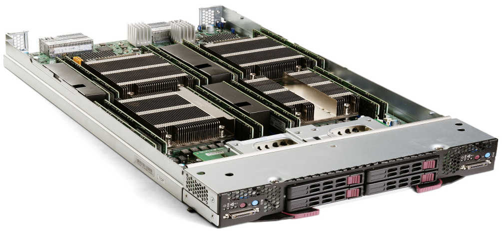
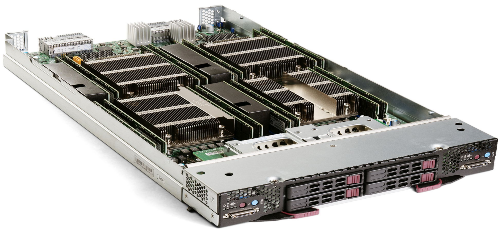
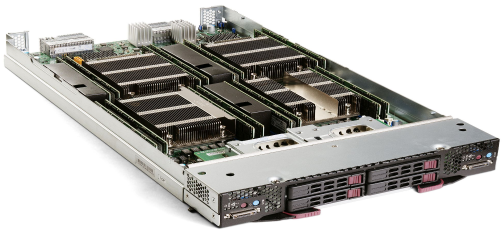
Now, only need to train and store B, A

Rank = 8

Rank = 128

https://medium.com/@dreamsarereal/understanding-lora-training-part-1-learning-rate-schedulers-network-dimension-and-alpha-c88a8658beb7

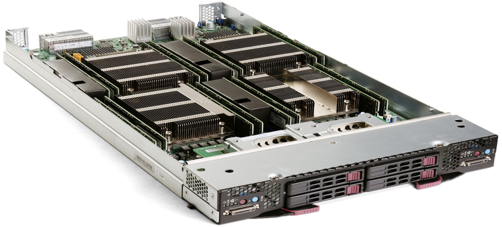
Finetuning a ~11B+ parameter model still requires multiple servers.

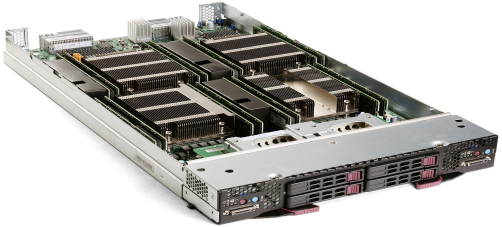




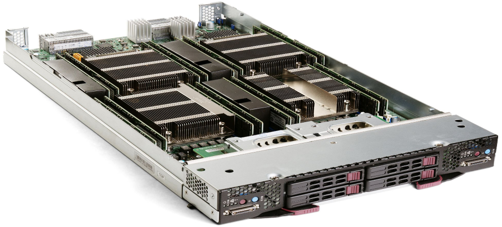
Slide credit to Tim Dettmers

Dettmers et al., 2023

QLoRA: Finetuning large models on a single GPU.







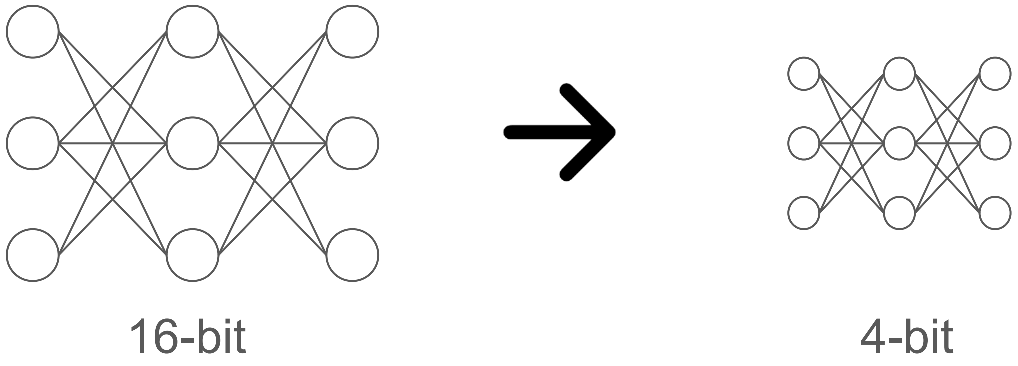
QLoRA

(4-bit finetuning)

Slide credit to Tim Dettmers

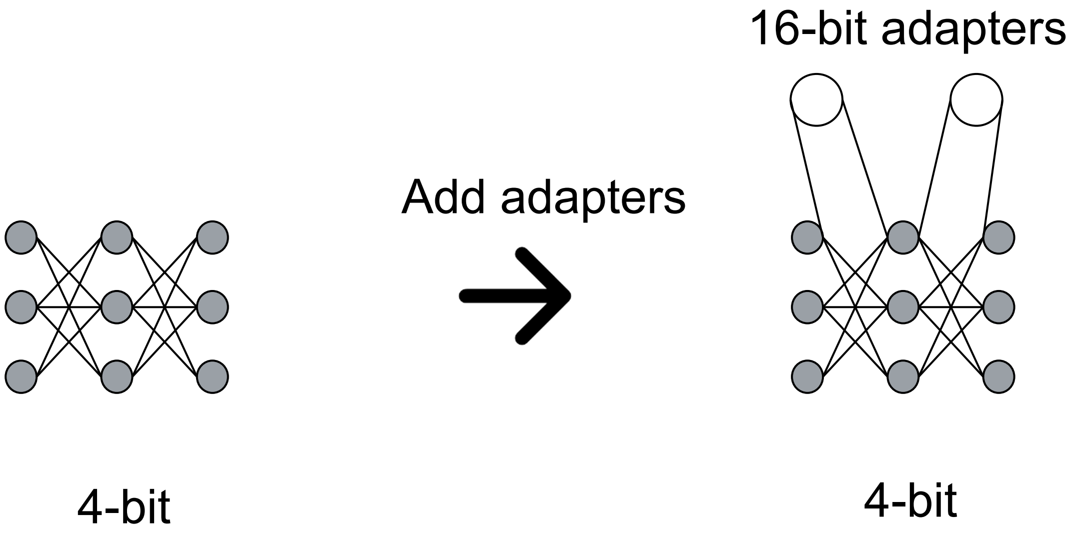
[Dettmers et al., 2023]

Quantized Low-rank Adaptation (QLoRA)



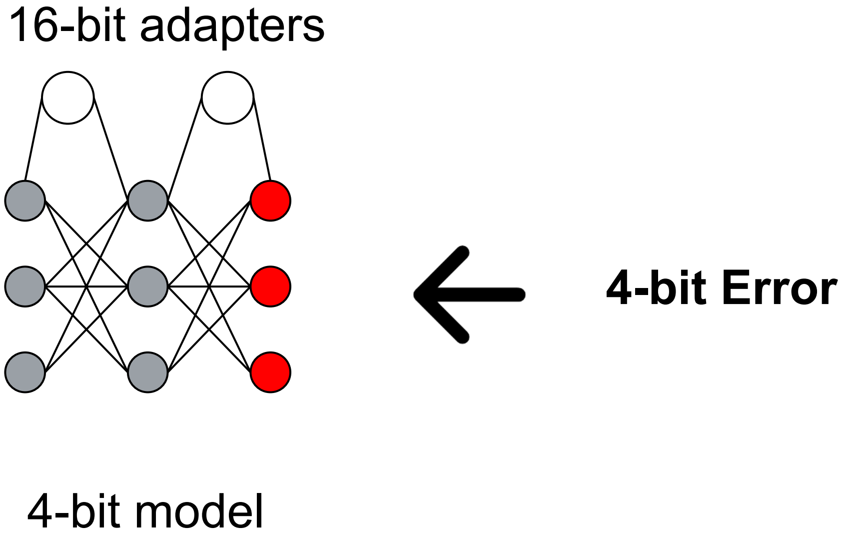
Slide credit to Tim Dettmers

Quantized Low-rank Adaptation (QLoRA)



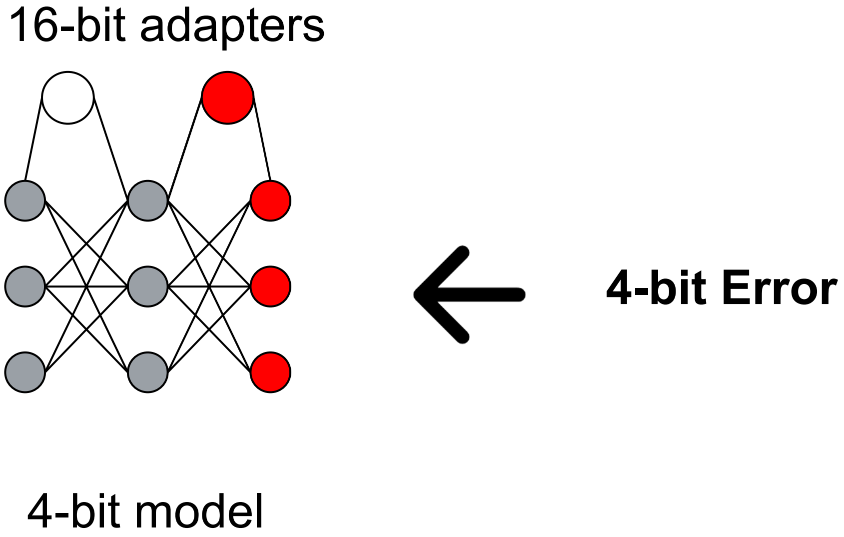
Slide credit to Tim Dettmers

Quantized Low-rank Adaptation (QLoRA)



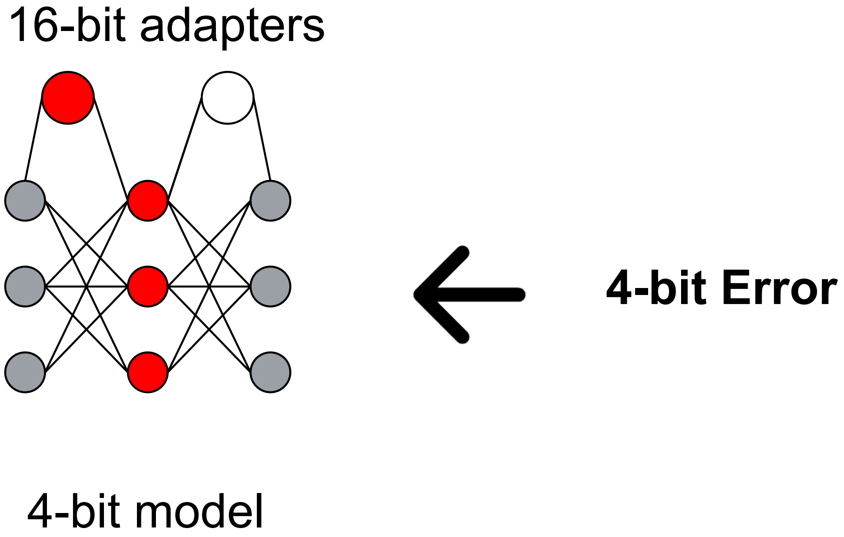
Slide credit to Tim Dettmers

Quantized Low-rank Adaptation (QLoRA)



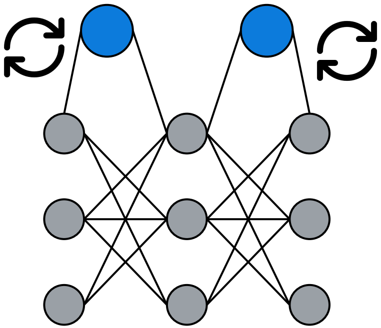
Slide credit to Tim Dettmers

Quantized Low-rank Adaptation (QLoRA)



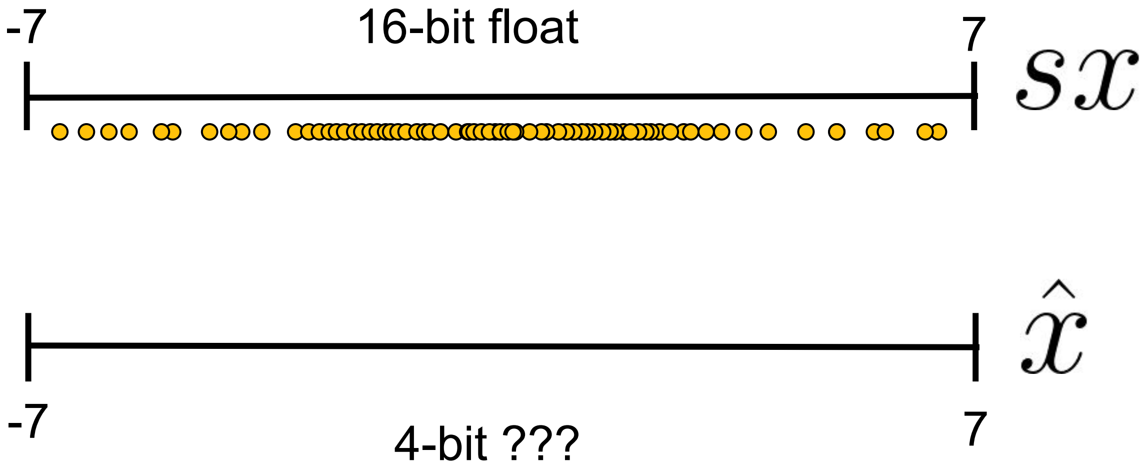
Slide credit to Tim Dettmers

Quantized Low-rank Adaptation (QLoRA)



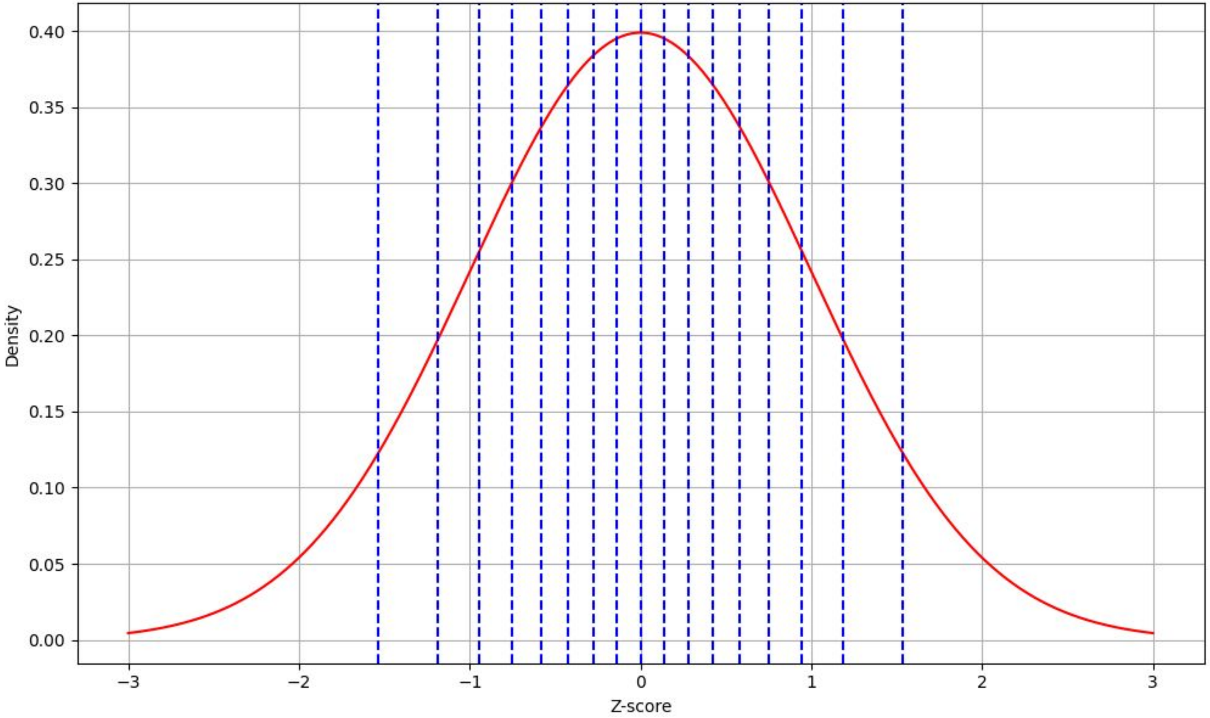
Slide credit to Tim Dettmers

What 4-bit data type is information theoretically optimal?



Slide credit to Tim Dettmers

4-bit NormalFloat (NF4) an information-theoretically optimal data type for normal distributions



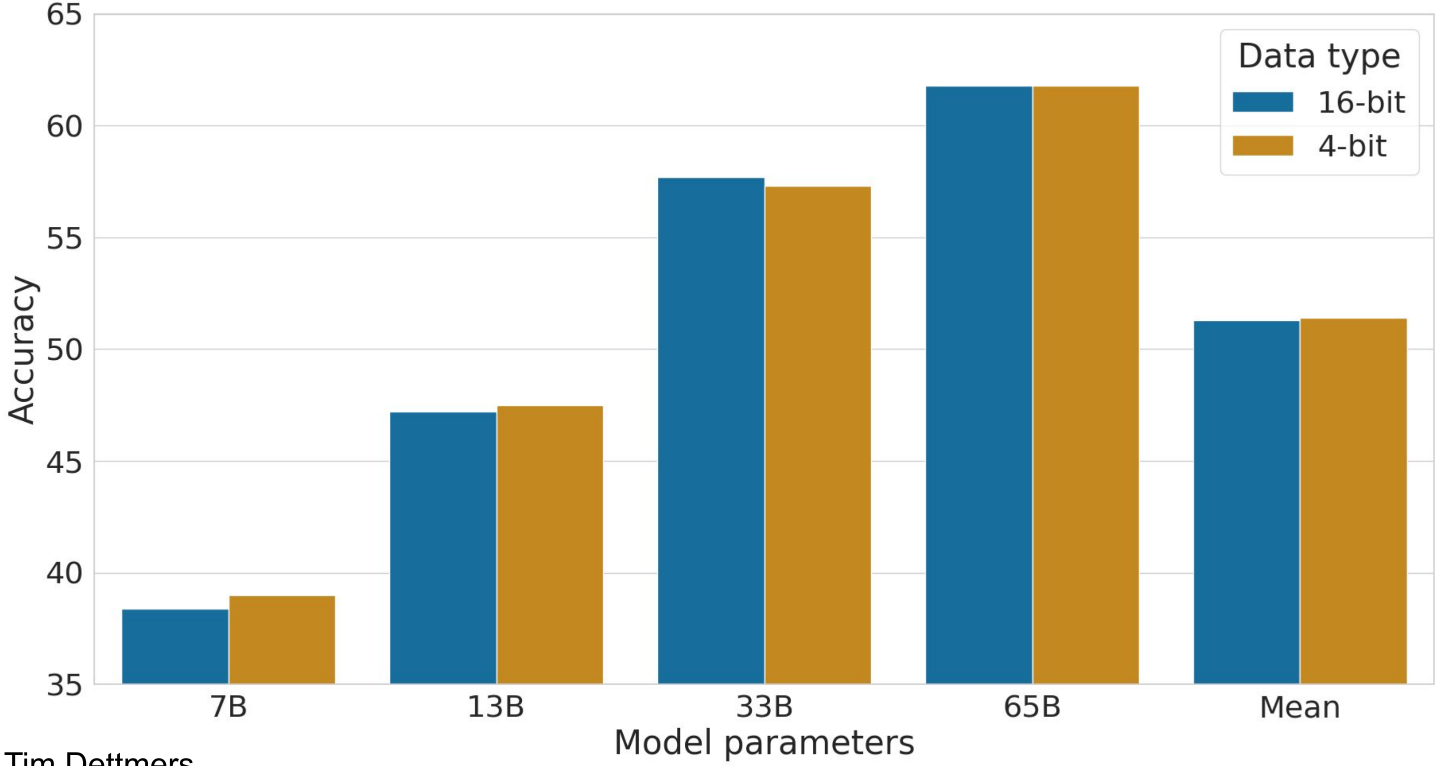
Slide credit to Tim Dettmers

QLoRA systems contributions

- Double quantization
- GPU memory paging for optimizer

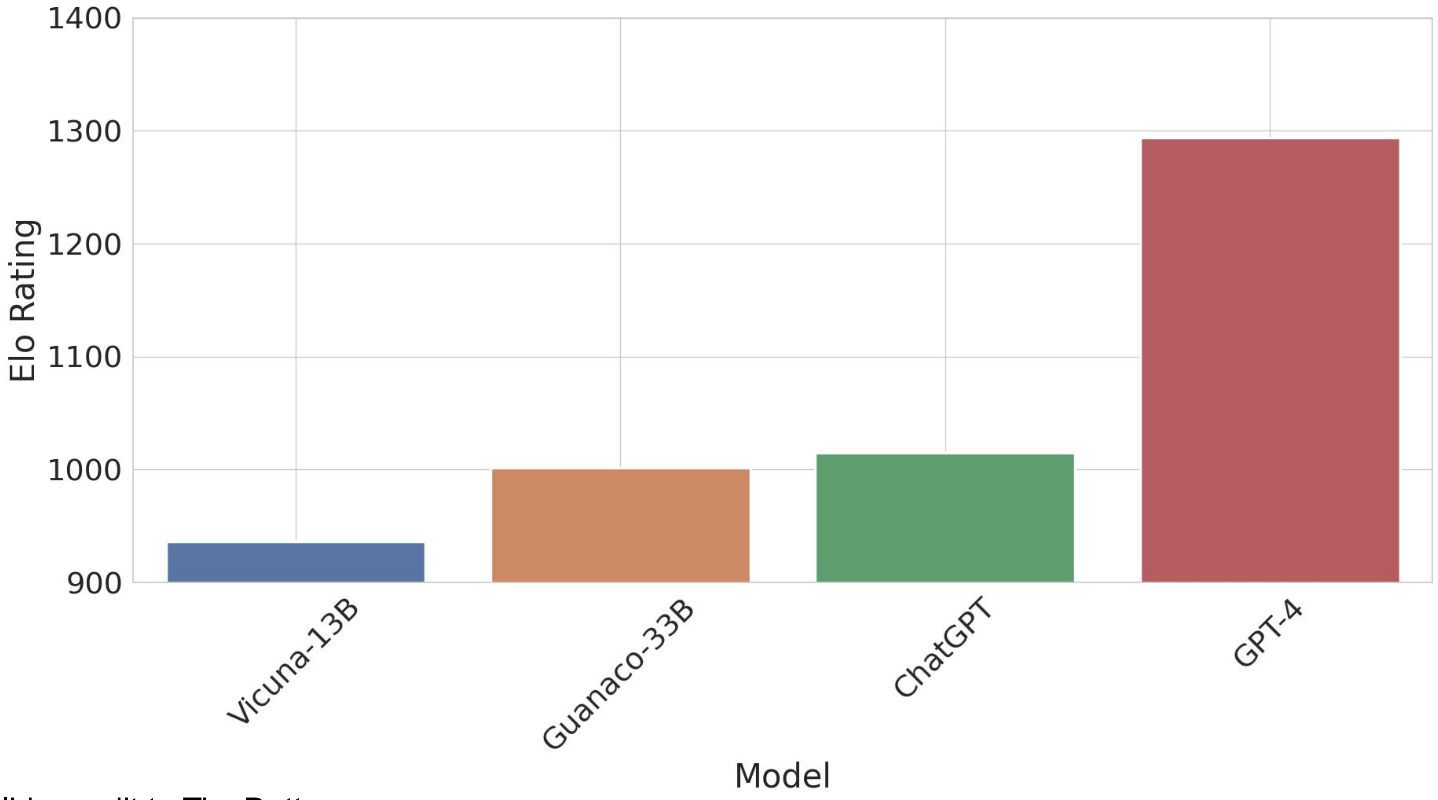
Results

QLoRA recovers lost performance through fine-tuning



Slide credit to Tim Dettmers

4-bit Guanaco: A ChatGPT-quality 4-bit chatbot finetuned in 24h on a single GPU



Slide credit to Tim Dettmers

Take-away

4-bit finetuning is possible by passing gradients through a 4-bit neural network to 16-bit adapters.

Slide credit to Tim Dettmers