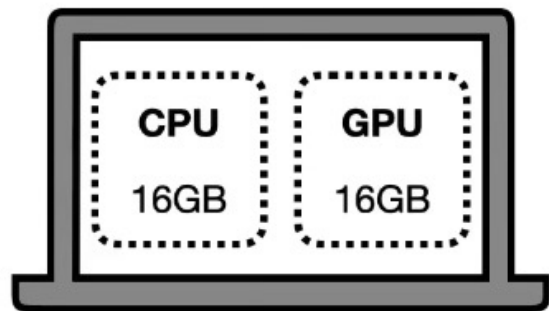


QLoRA

LLM fine-tuning made accessible

The Problem

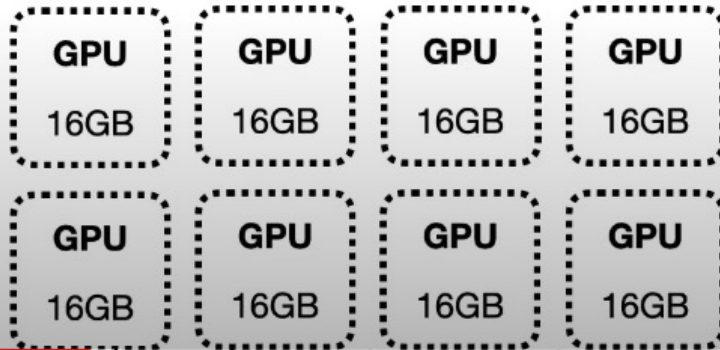
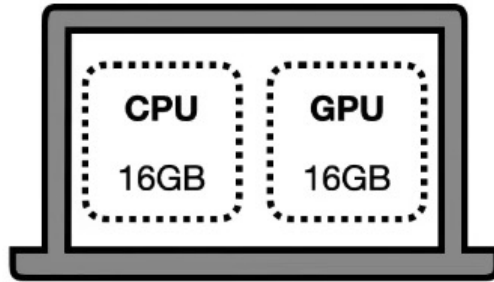
LLMs are (computationally) expensive



10B Parameter Model

The Problem

LLMs are (computationally) expensive



10B Parameter Model = 160GB!

Parameters (FP16) 20GB

Gradients (FP16) 20GB

Optimizer States
(FP32)

Momentum
Variance

120GB

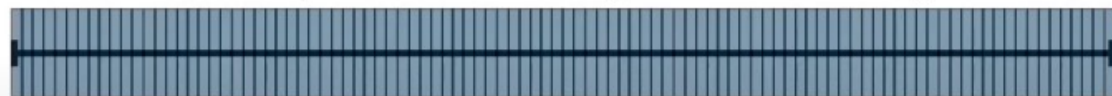
What is Quantization?

Quantization = splitting range into buckets

Any number
between 0 and 100



Quantized by
whole numbers



27

55

83

Quantized by 10s



20

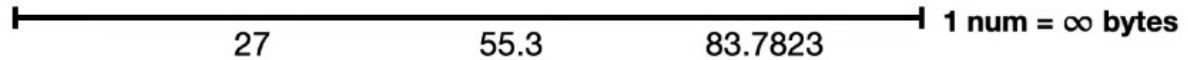
50

80

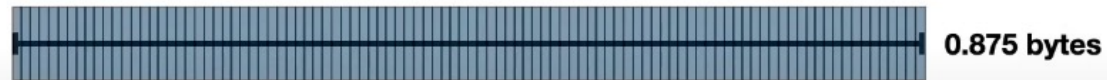
What is Quantization?

Quantization = splitting range into buckets

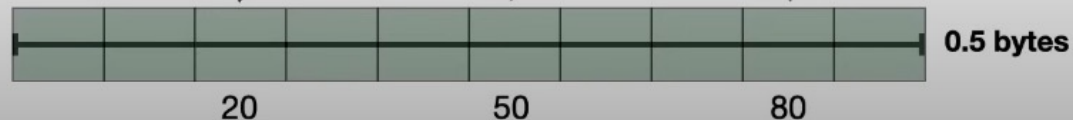
Any number
between 0 and 100



Quantized by
whole numbers



Quantized by 10s



Ingredient 1: 4-bit NormalFloat

A better way to bucket numbers

4-bit e.g. 0101

$\Rightarrow 2^4 = 16$ unique combinations

$\Rightarrow 16$ buckets for quantizations

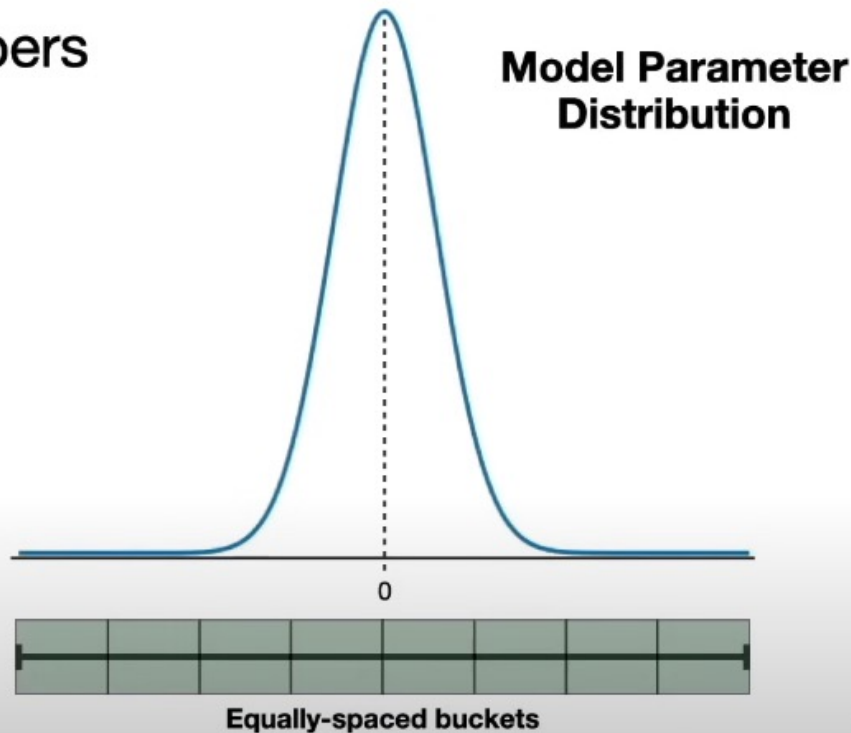
Ingredient 1: 4-bit NormalFloat

A better way to bucket numbers

4-bit e.g. 0101

$\Rightarrow 2^4 = 16$ unique combinations

$\Rightarrow 16$ buckets for quantizations



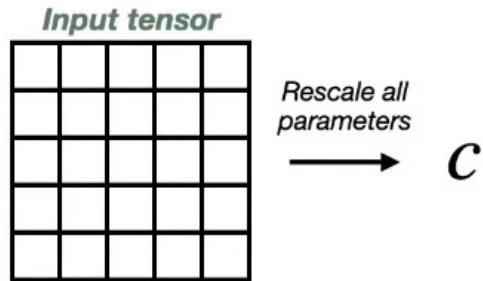
Ingredient 2: Double Quantization

Quantizing the Quantization Constants

$$x^{\text{Int8}} = \text{round} \left(\frac{127}{\text{absmax}(x^{\text{FP32}})} x^{\text{FP32}} \right)$$

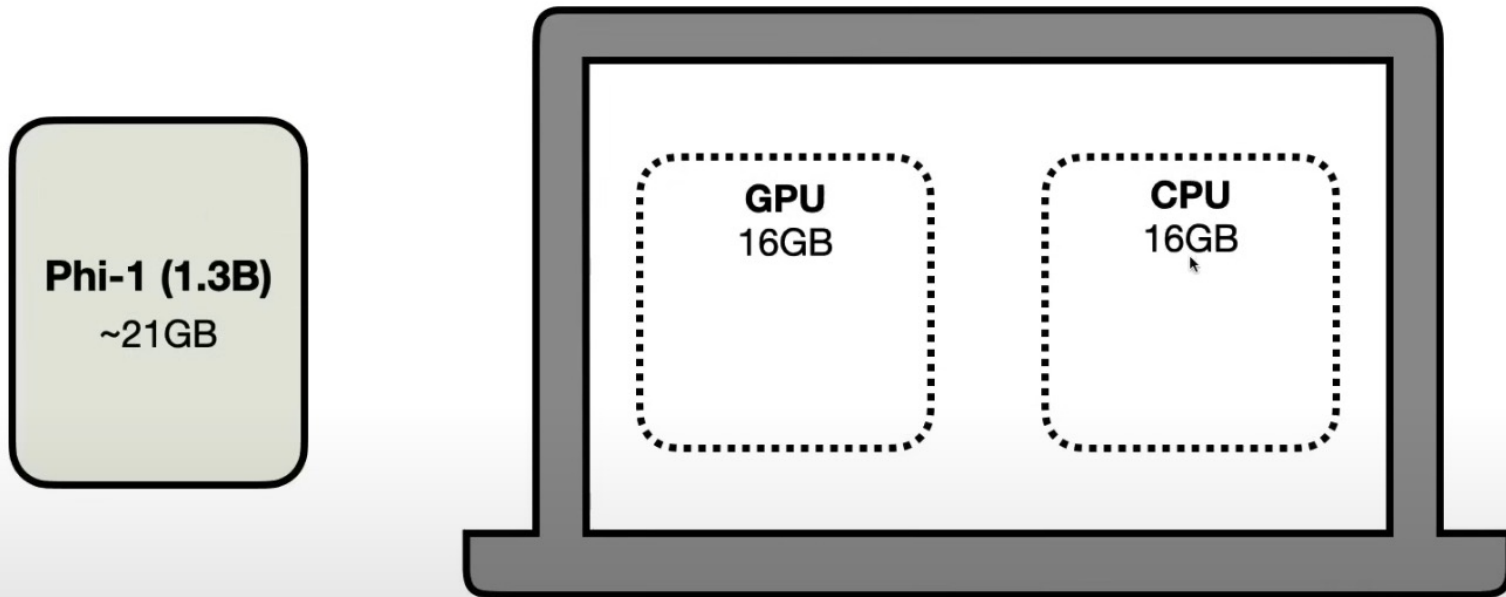
$$= \text{round} \left(c^{\text{FP32}} \cdot x^{\text{FP32}} \right)$$

↑
Takes up precious
memory



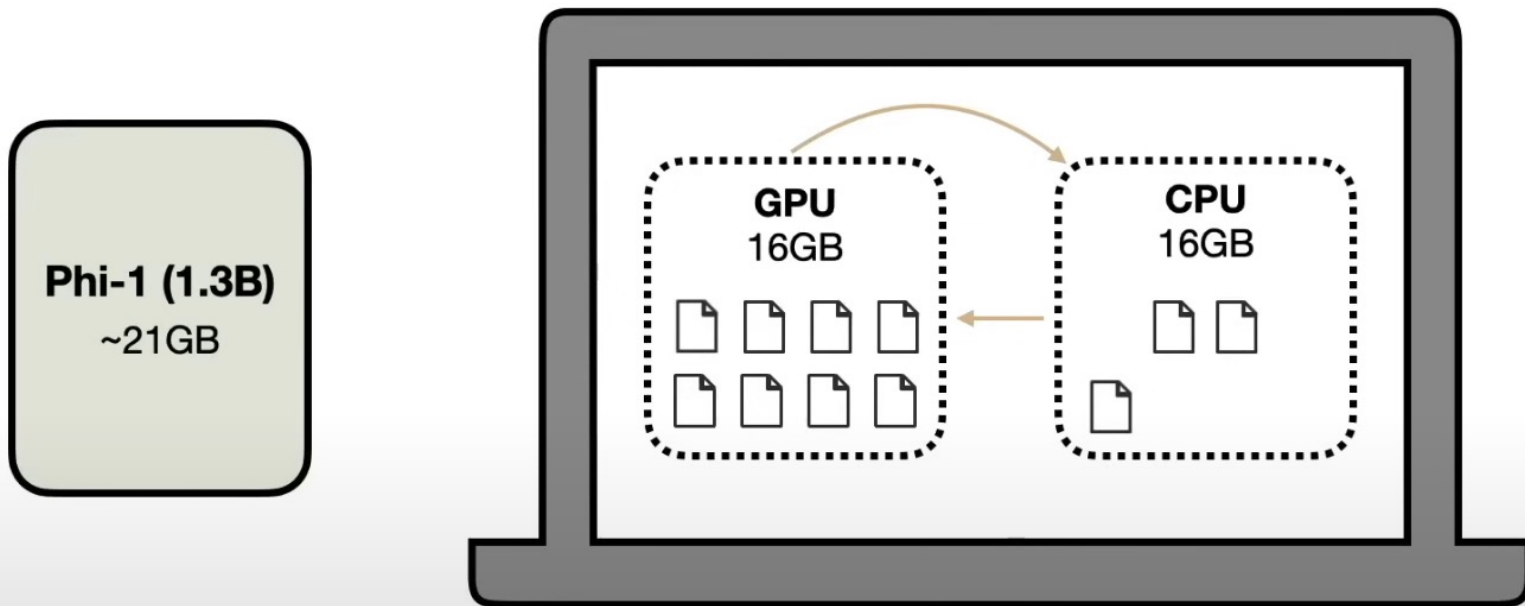
Ingredient 3: Paged Optimizer

Looping in your CPU



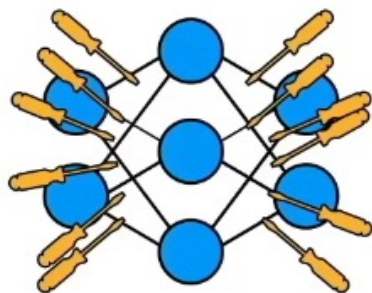
Ingredient 3: Paged Optimizer

Looping in your CPU

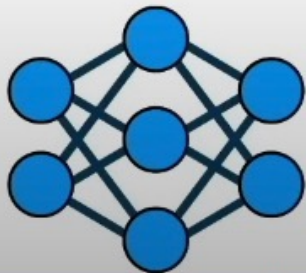


Ingredient 4: LoRA

Fine-tunes model by adding **small set** of trainable parameters



$x \quad h(x) \quad y$

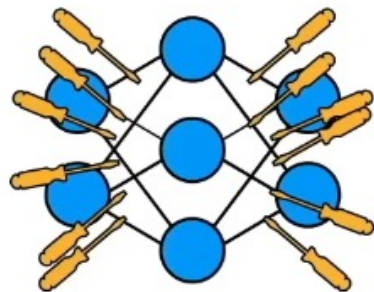


Full Fine-tuning: $h(x) = W_0 x$

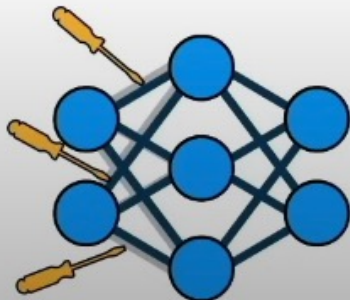
$$\begin{array}{c} \boxed{W_0} x = h(x) \\ \text{Trainable} \end{array}$$

Ingredient 4: LoRA

Fine-tunes model by adding **small set** of trainable parameters



$x \quad h(x) \quad y$



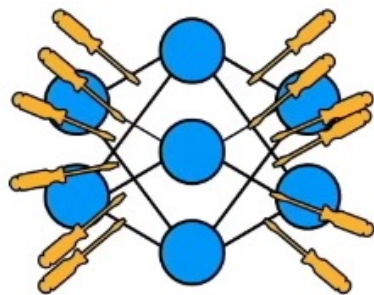
Full Fine-tuning: $h(x) = W_0x$

$$\underbrace{W_0}_{\text{Trainable}} x = h(x)$$

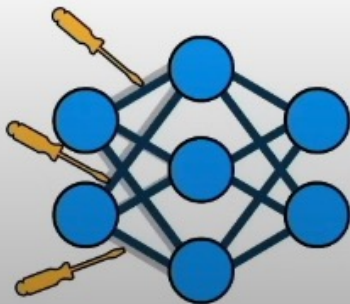
LoRA: $h(x) = W_0x + \Delta Wx = W_0x + BAx$

Ingredient 4: LoRA

Fine-tunes model by adding **small set** of trainable parameters



$x \quad h(x) \quad y$



Full Fine-tuning: $h(x) = W_0x$

$$\underbrace{\begin{matrix} \boxed{W_0} \end{matrix}}_{\text{Trainable}} \begin{matrix} \boxed{x} \end{matrix} = \begin{matrix} \boxed{h(x)} \end{matrix}$$

LoRA: $h(x) = W_0x + \Delta Wx = W_0x + BAx$

$$\left(\underbrace{\boxed{W_0}}_{\text{Frozen}} + \begin{matrix} \boxed{B} \end{matrix} \begin{matrix} \boxed{A} \end{matrix} \right) \begin{matrix} \boxed{x} \end{matrix} = \begin{matrix} \boxed{h(x)} \end{matrix}$$

Original
Model Weights

+

LoRA
Weight Changes

=

Fine-tuned
Model Weights

LoRA Low-rank Matrices

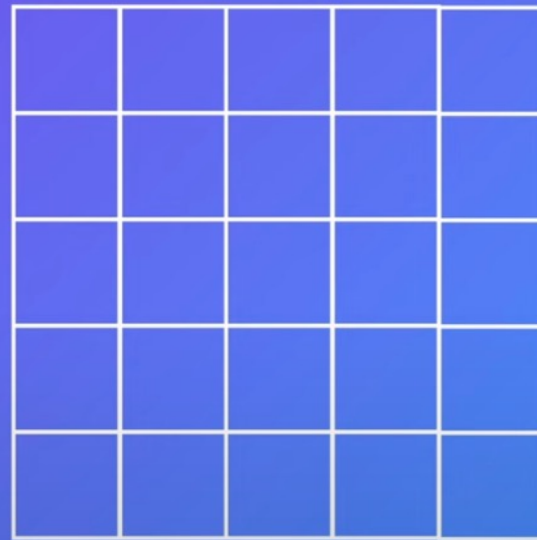



x



=

LoRA Weight Changes




Rank = 1

Increasing Precision by Increasing Rank

LoRA Matrices, Rank 2

x

=

Higher Precision
Weight Changes

Rank = 2

Number of Trainable Parameters

Rank	7B	13B	70B	180B
1	167k	228k	529k	849k
2	334k	456k	1M	2M
8	1M	2M	4M	7M
16	3M	4M	8M	14M
512	86M	117M	270M	434M
1,024	171M	233M	542M	869M
8,192	1.4B	1.8B	4.3B	7.0B

In reality, LLMs are made up of multiple layers of differing sizes. This is a generalization as if the model were a single layer.

Bringing it all together

