Finetuning

Jordan Boyd-Graber

University of Maryland

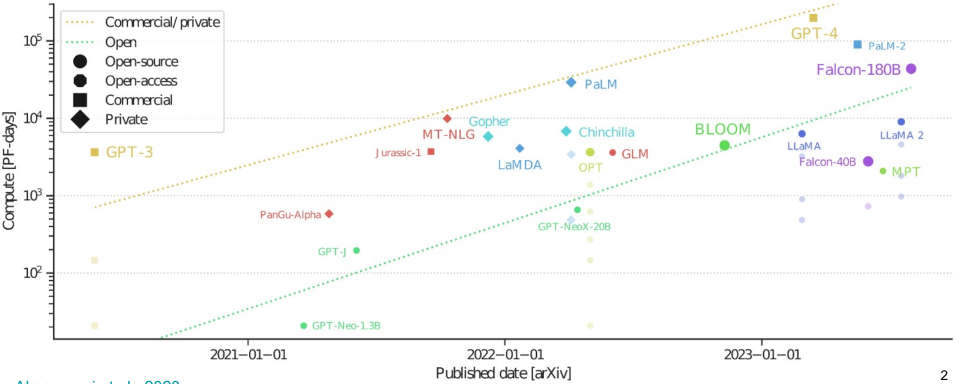
Low Rank Adaptation and Quantization

Slides adapted from Umar Jamil, Vyacheslav Efimov, and Tim Dettmers

Plan for Today

- Depressing fact: You are not OpenAI
- But you still can customize your own models
- General Approaches
  - Distillation
  - Adaptation
  - Quantization
  - Prompting

Language models grew 100x in compute requirements in a few years



Almazrouei et al., 2023

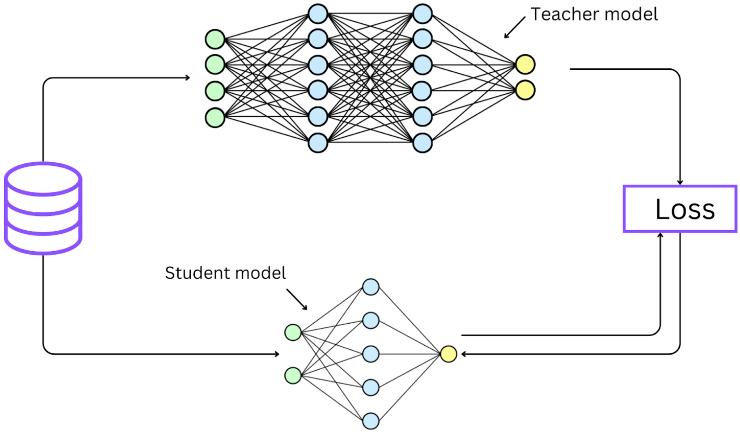
Distillation

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF
Hugging Face
{victor,lysandre,julien,thomas}@huggingface.co

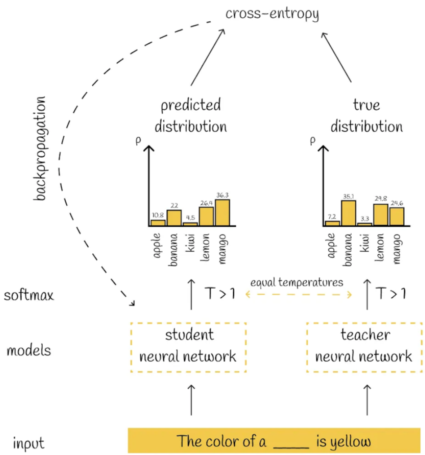
2019: Popular way of reducing big model to smaller model

Distillation



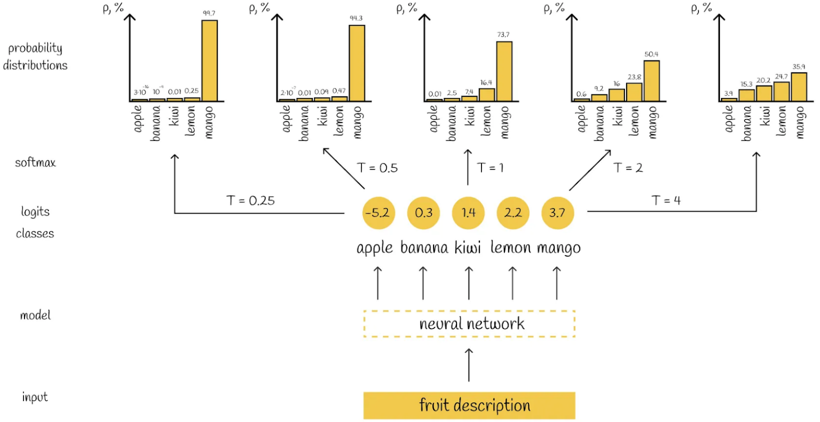
Going to train the student model to match the (bigger) teacher model

Distillation



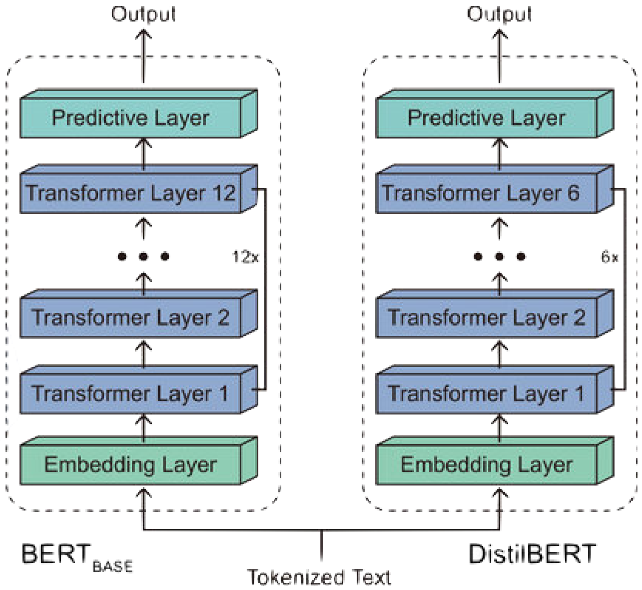
You're not just trying to match label prediction, trying to match teacher predicted distribution

Distillation



Use higher termperature to capture details of distribution

Distillation



Keep every other layer, initialize to their previous values

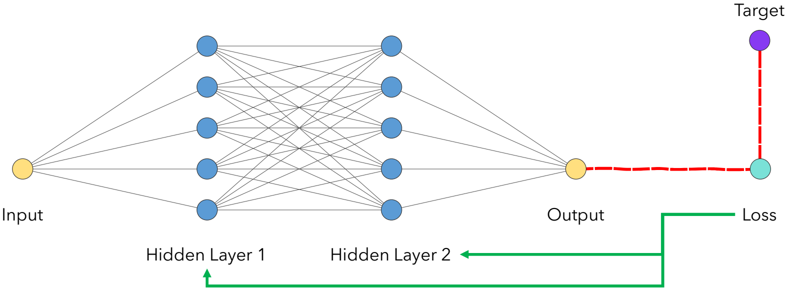
Adaptation: When Distillation isn't enough

LoRA: Low-Rank Adaptation of Large Language Models

Edward Hu* **Yelong Shen*** **Phillip Wallis** **Zeyuan Allen-Zhu**
Yuanzhi Li **Shean Wang** **Lu Wang** **Weizhu Chen**
Microsoft Corporation
{edwardhu, yeshe, phwallis, zeyuana,
yuanzhil, swang, luw, wzchen}@microsoft.com
yuanzhil@andrew.cmu.edu

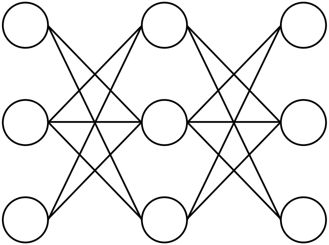
2021, one of the most used techniques today

Adaptation: When Distillation isn't enough



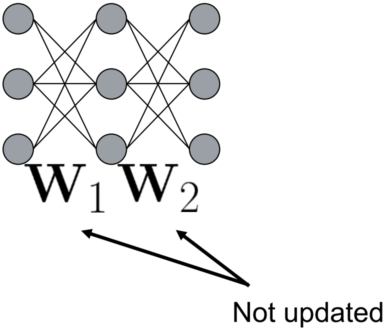
If you fine-tune, you'll need to **store all of the paramters**. . . not always possible

Adaptation: When Distillation isn't enough



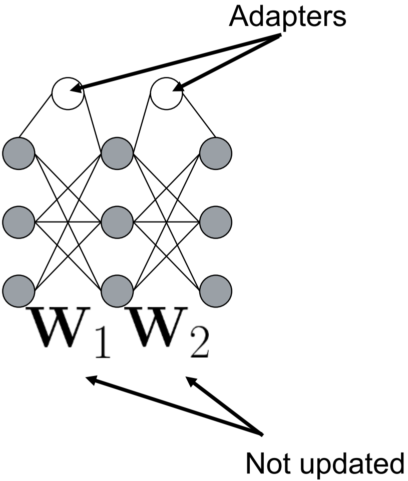
Let's zoom in

Adaptation: When Distillation isn't enough



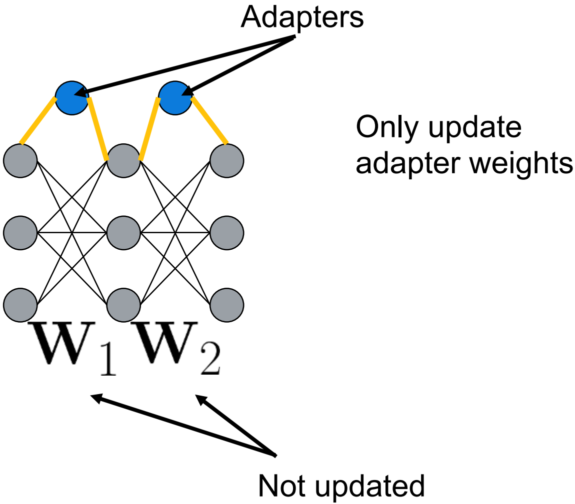
Let's keep the original parameters as-is

Adaptation: When Distillation isn't enough



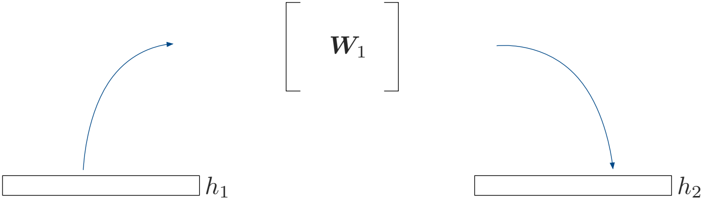
Adapter $\mathbf{D}$: a cheaper addative change: $\mathbf{W}_1' = \mathbf{W}_1 + \mathbf{D}$

Adaptation: When Distillation isn't enough



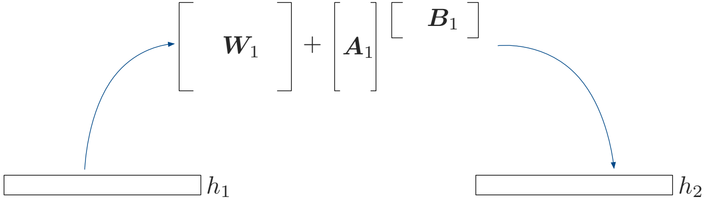
Can be cheaper if $\mathbf{D}$ is low-rank: $\mathbf{W}_1' = \mathbf{W}_1 + \mathbf{A}_1 \cdot \mathbf{B}_1$

Let's think about the Dimensions



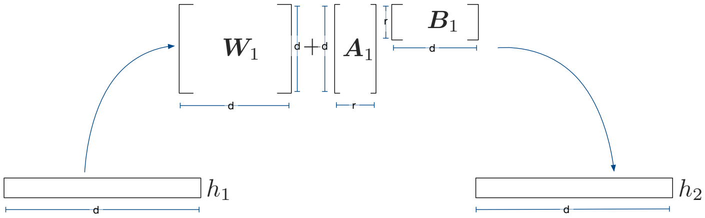
Normally, $\mathbf{W}$ maps hidden state of dimension d to the same thing. This requires d^2 floats.

Let’s think about the Dimensions



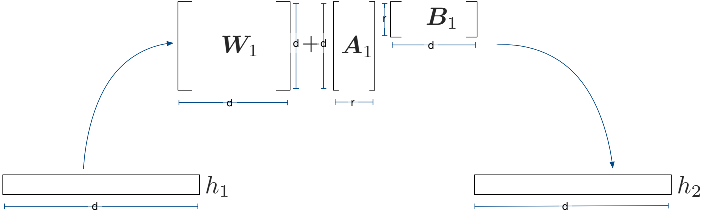
So we keep the updates in two lower rank matrices of dimension $r << d$.

Let's think about the Dimensions



This requires floats but only adds $r2d$ to the total.

Let's think about the Dimensions



This requires floats but only adds $r2d$ to the total.

Does this work?

Method	# of Trainable Parameters	WikiSQL Accuracy (%)	MNLI-m Accuracy (%)	SAMSum R1/R2/RL
GPT-3 175B (Fine-Tune)	175,255.8M	73.0	89.5	52.0/28.0/44.5
GPT-3 175B (Bias Only)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 175B (PrefixEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 175B (PrefixLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 175B (LoRA)	4.7M	73.4	91.3	52.1/28.3/44.3
GPT-3 175B (LoRA)	37.7M	**73.8**	**91.7**	**53.2/29.2/45.0**

Table 1: Logical form validation accuracy on WikiSQL, validation accuracy on MultiNLI-matched and Rouge-1/2/L on SAMSum achieved by different GPT-3 adaptation methods. LoRA performs better than prior approaches, including conventional fine-tuning. The result on WikiSQL has a fluctuation of $\pm 0.3\%$ and MNLI-m $\pm 0.1\%$.

What's the right rank?

	Weight Type	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 64$
WikiSQL($\pm 0.3\%$)	W_q, W_v	73.4	73.3	**73.7**	**73.8**	73.5
	W_q	68.8	69.6	**70.5**	**70.4**	70.0
MultiNLI ($\pm 0.1\%$)	W_q, W_v	91.3	91.4	91.3	**91.7**	91.4

Why does this work?

THE EXPRESSIVE POWER OF LOW-RANK ADAPTATION

Yuchen Zeng
Department of Computer Science
University of Wisconsin-Madison
yzeng58@wisc.edu

Kangwook Lee
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
kangwook.lee@wisc.edu

LoRA can adapt any model f to accurately represent smaller target $\hat{f}$ if LoRA-rank $\geq \frac{\text{width}(f)\text{depth}(\hat{f})}{\text{depth}(f)}$.

We talked about the number of floats, but do we need full precision?

QLoRA: Efficient Finetuning of Quantized LLMs

Tim Dettmers* **Artidoro Pagnoni*** **Ari Holtzman**

Luke Zettlemoyer

University of Washington
{dettmers,artidoro,ahai,lsz}@cs.washington.edu

We talked about the number of floats, but do we need full precision?

QLoRA: Efficient Finetuning of Quantized LLMs

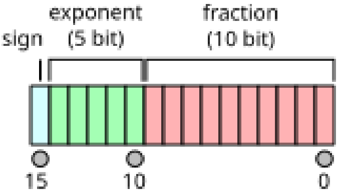
Tim Dettmers* **Artidoro Pagnoni*** **Ari Holtzman**

Luke Zettlemoyer

University of Washington
{dettmers,artidoro,ahai,lsz}@cs.washington.edu

How much space do these take?

Type	Size
int ($-n$ to n)	bits
half-precision float	bits



IEEE 754

We talked about the number of floats, but do we need full precision?

QLoRA: Efficient Finetuning of Quantized LLMs

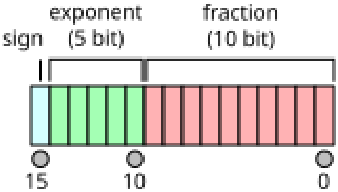
Tim Dettmers* **Artidoro Pagnoni*** **Ari Holtzman**

Luke Zettlemoyer

University of Washington
{dettmers,artidoro,ahai,lsz}@cs.washington.edu

How much space do these take?

Type	Size
int ($-n$ to n)	$\lg(2n)$ bits
half-precision float	bits



IEEE 754

We talked about the number of floats, but do we need full precision?

QLoRA: Efficient Finetuning of Quantized LLMs

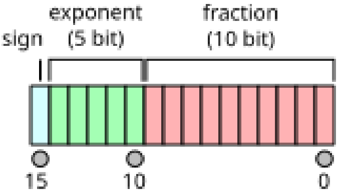
Tim Dettmers* **Artidoro Pagnoni*** **Ari Holtzman**

Luke Zettlemoyer

University of Washington
{dettmers,artidoro,ahai,lsz}@cs.washington.edu

How much space do these take?

Type	Size
int ($-n$ to n)	$\lg(2n)$ bits
half-precision float	16 bits

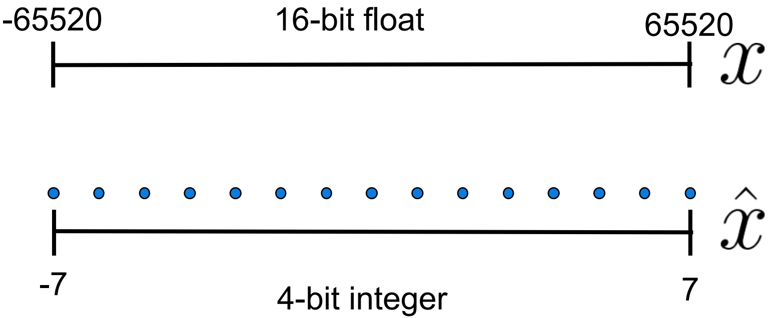


10 IEEE 754

QLoRA

Half-precision floats range: $\pm$ 65504, minimum value above 1 is $1 + \frac{1}{1024}$

QLoRA



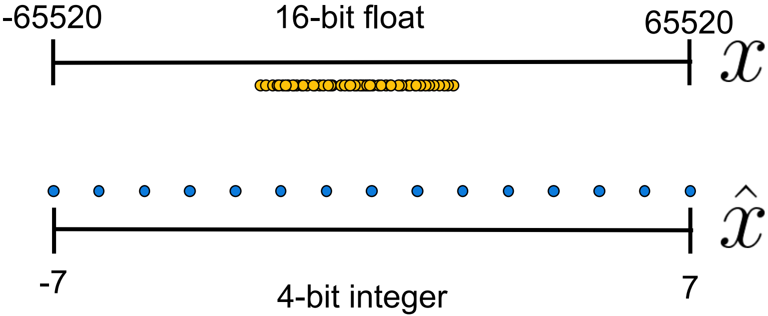
$$\mathbf{Y}^{\text{BF16}} = \mathbf{X}^{\text{BF16}}\text{doubleDequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}, \mathbf{W}^{\text{NF4}}) + \mathbf{X}^{\text{BF16}}\mathbf{L}_1^{\text{BF16}}\mathbf{L}_2^{\text{BF16}}, \tag{5}$$

where doubleDequant$(\cdot)$ is defined as:

$$\text{doubleDequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}, \mathbf{W}^{\text{k-bit}}) = \text{dequant}(\text{dequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}), \mathbf{W}^{\text{4bit}}) = \mathbf{W}^{\text{BF16}}, \tag{6}$$

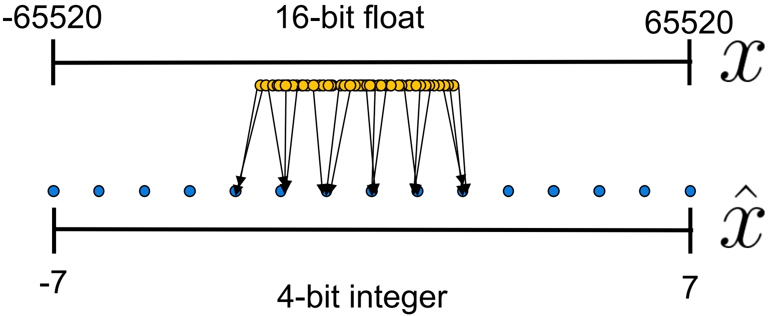
At first, this doesn't seem that great, as a 4-bit integer only covers a small portion of the range

QLoRA



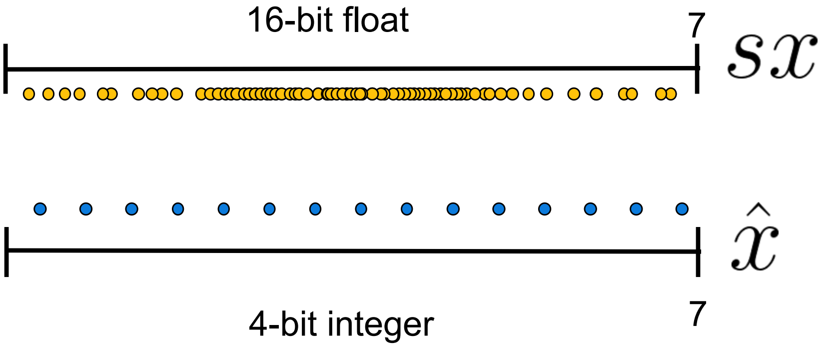
But thanks to initialization / regularization, most weights are small.

QLoRA



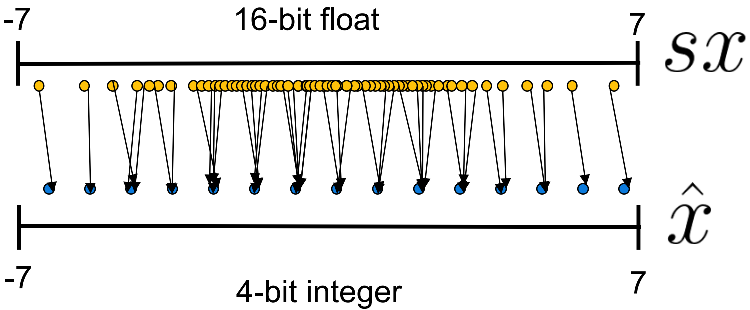
So it doesn't make sense to only use part of our mapping.

QLoRA



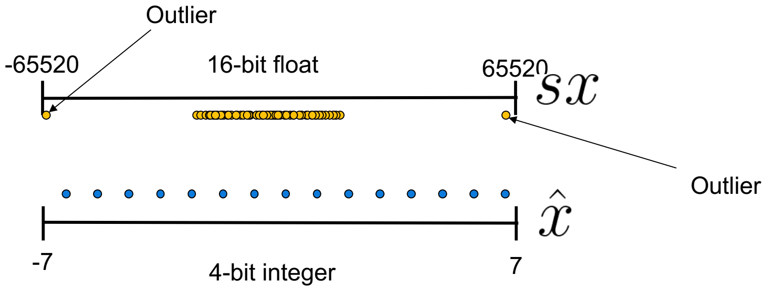
So let's restrict our range

QLoRA



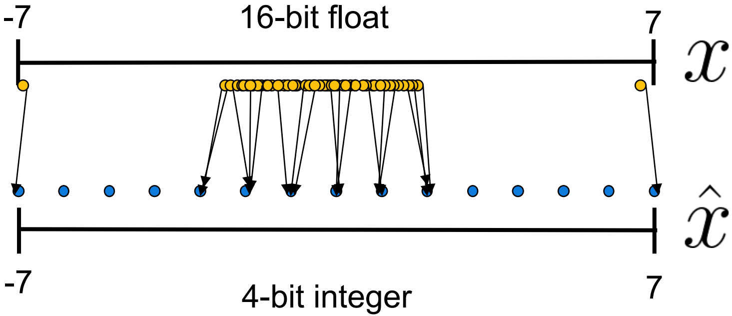
Fewer collisions

QLoRA



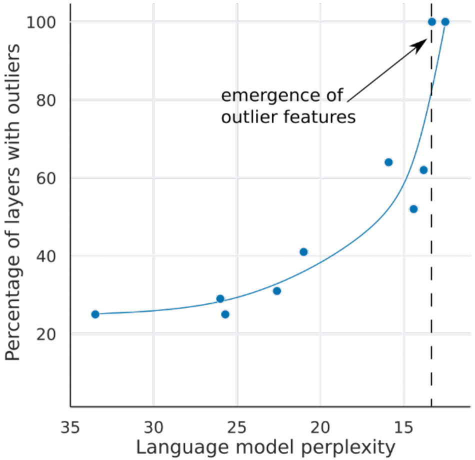
But this doesn't work if we have "outliers"

QLoRA



A uniform mapping would be mostly wasted, with lots of collisions

QLoRA



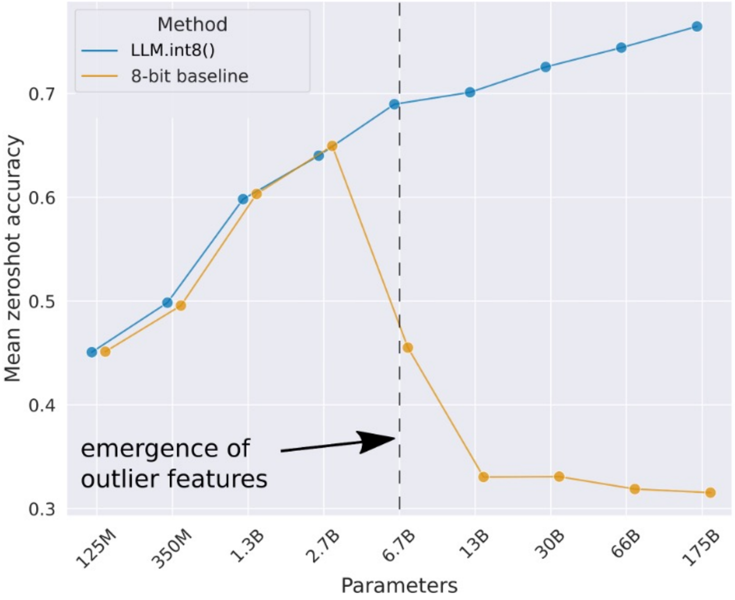
But the heads with outliers are actually really important for low perplexity

QLoRA

$$\mathbf{C}_{f16} \approx \sum_{h \in O} \mathbf{X}_{f16}^{h} \mathbf{W}_{f16}^{h} + \mathbf{S}_{f16} \cdot \sum_{h \notin O} \mathbf{X}_{i8}^{h} \mathbf{W}_{i8}^{h}$$

So we look for the heads with outliers and handle them separately (more bits)

QLoRA


Method
LLM.int8()
8-bit baseline
Mean zeroshot accuracy
0.3
0.4
0.5
0.6
0.7
emergence of
outlier features
125M
350M
1.3B
2.7B
6.7B
13B
30B
66B
175B
Parameters

Wrapup

- You **cannot** fine tune the largest models
- LoRA lets you keep track of backprop changes with fewer parameters
- QLoRA lets you keep track of those changes with even less memory
  - Finetuning possible on laptops
  - And machines without GPUs