Quantization of LLMs

Fine-Tuning with QLoRA

Presented by

Greg Loughnane, Founder & CEO
Chris Alexiuk, Co-Founder & CTO

🎯 ALIGNING OUR AIM

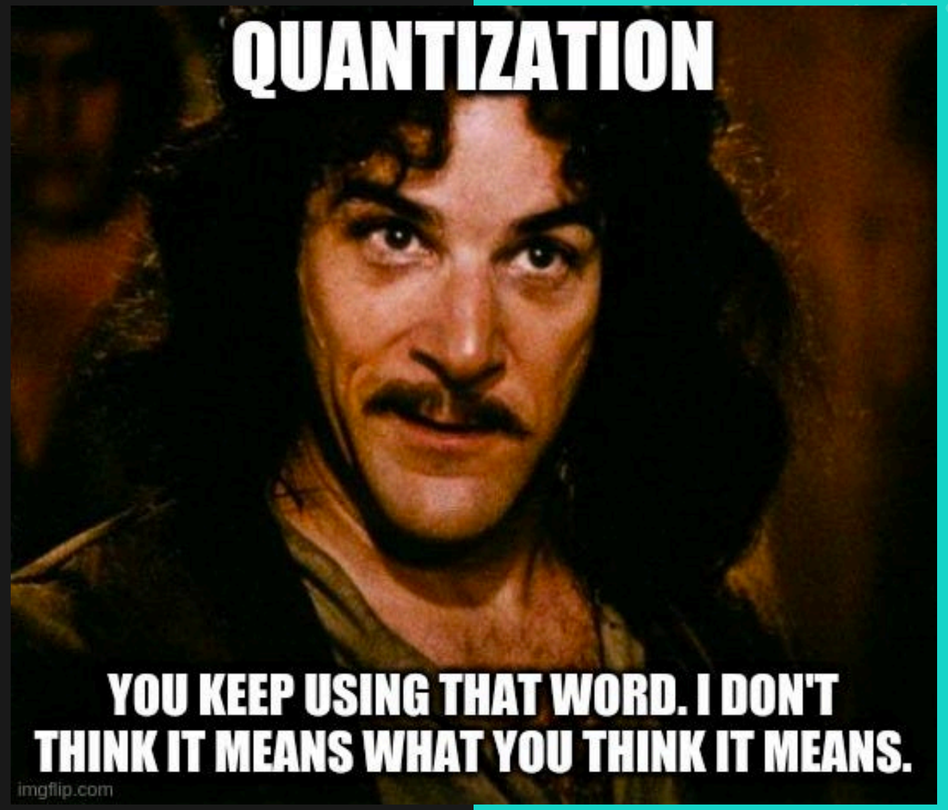
AI MAKERSPACE

BY THE END OF TODAY...

- What is **quantization** and **QLoRA**
- **How to fine-tune LLMs** using a quantized approach **Hugging Face**

OVERVIEW

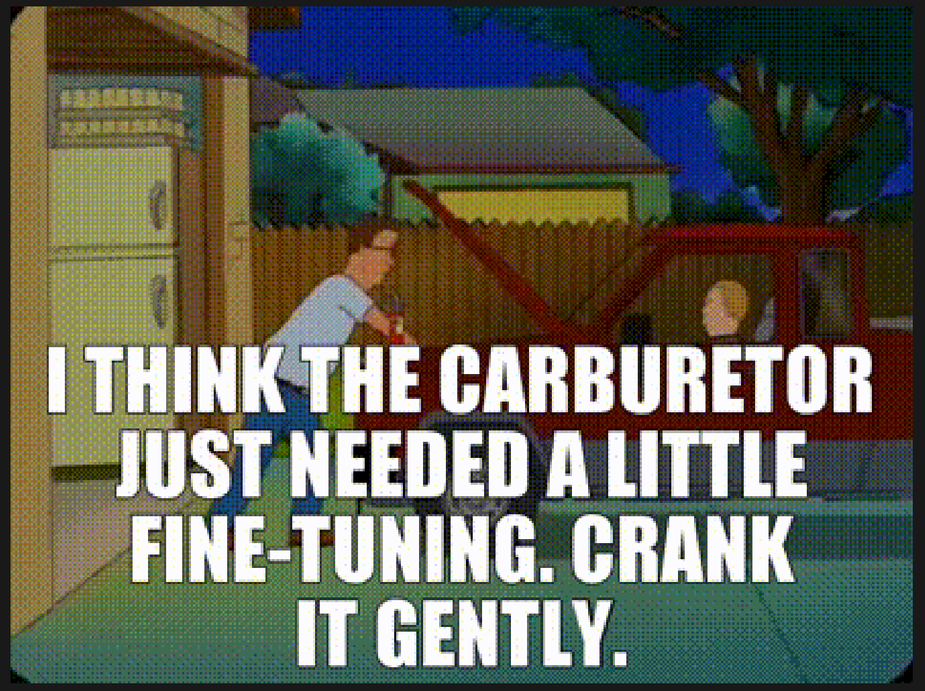
- ⚖️ Fine-Tuning with PEFT-LoRA
- ⚛️ Quantization
- ⚡ QLoRA
- 👷 FT with Hugging Face
- ❓ Conclusions, QA



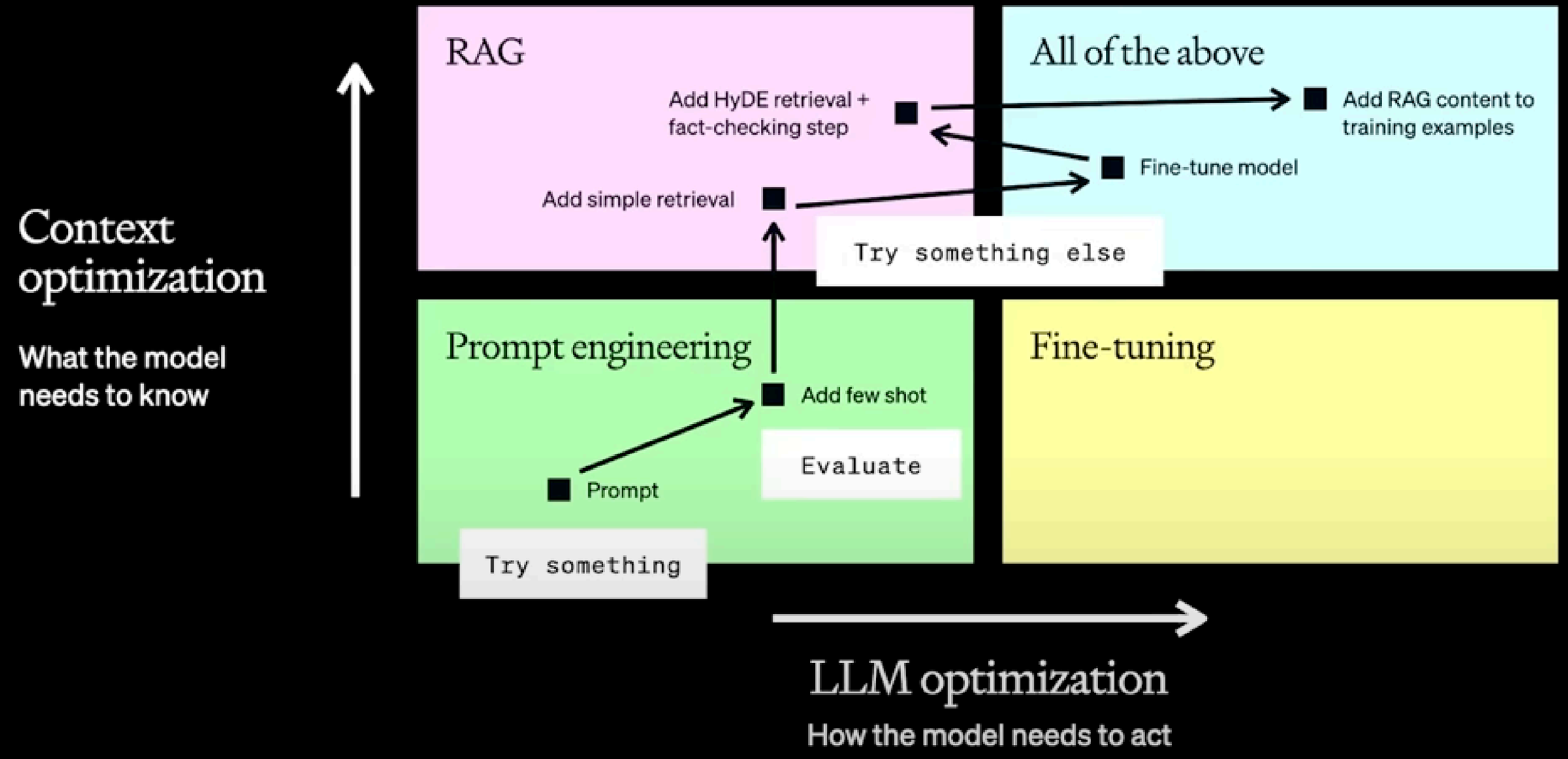
💪 PROTOTYPING

PROTOTYPING LLM APPS

1. Prompt Engineering
2. Question Answering Systems
3. Fine-Tuning Models



The optimization flow



Small Language Models (SLMs)

Efficiency, Transparency , Accuracy, Security

TinyStories: How Small Can Language Models Be and Still Speak...

Language models (LMs) are powerful tools for natural language processing, but they often struggle to produce coherent and fluent text when they are small. Models with around 125M parameters such...

arXiv.org

Textbooks Are All You Need

We introduce phi-1, a new large language model for code, with significantly smaller size than competing models: phi-1 is a Transformer-based model with 1.3B parameters, trained for 4 days on 8...

arXiv.org

Fine-Tuning with PEFT-LoRA

LLMS ARE BIG

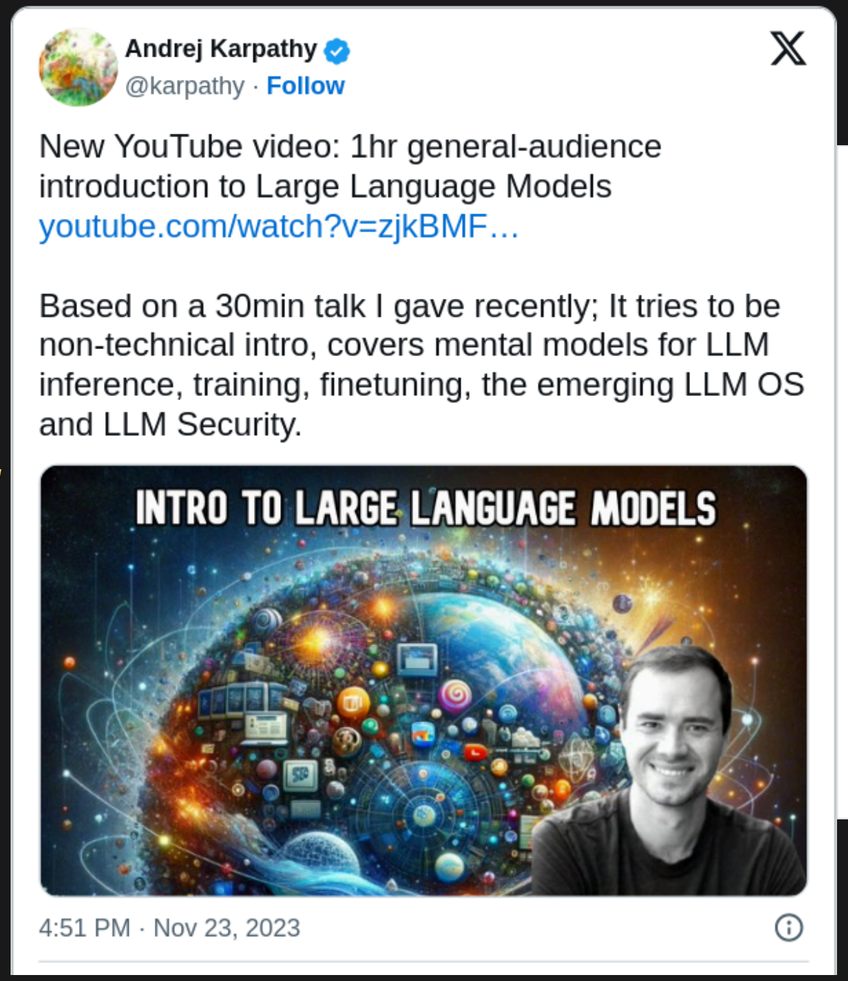
LLMs are so big!

7B, 13B, 34B, 70B **parameters**

SCALING LAWS

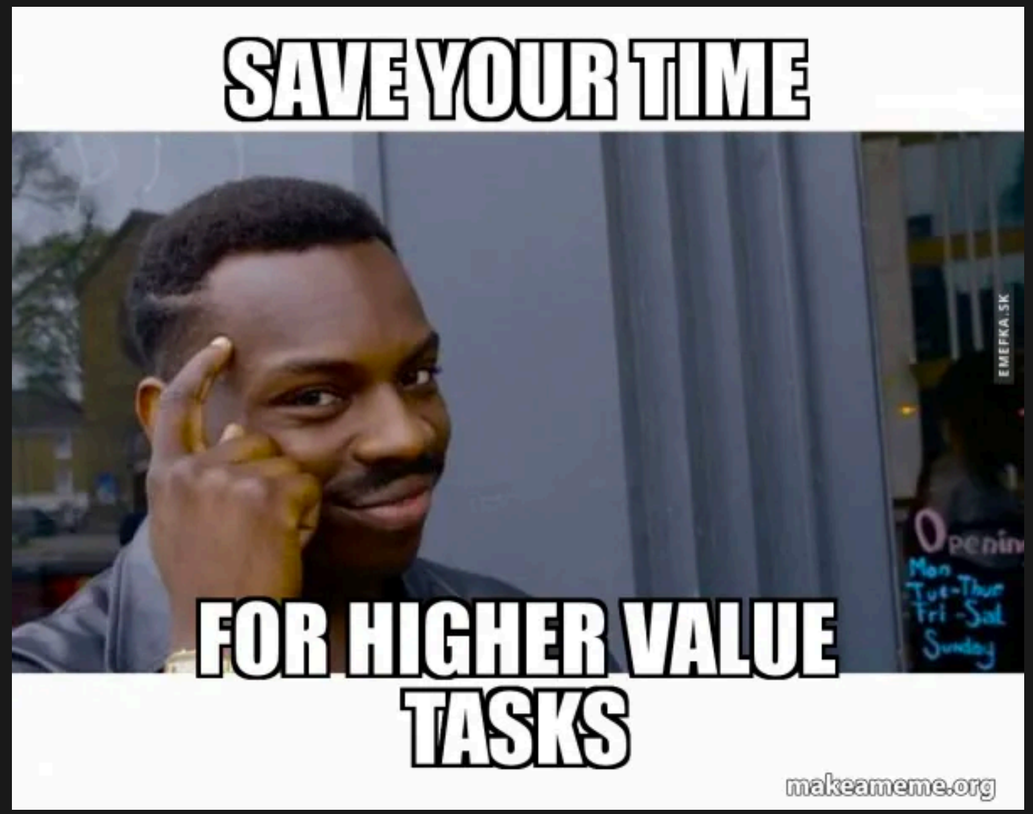
*"If you train a **bigger model** on **more text**, we have a lot of confidence that **next word prediction will improve**."*

~ Andrej Karpathy



TWO CHALLENGES

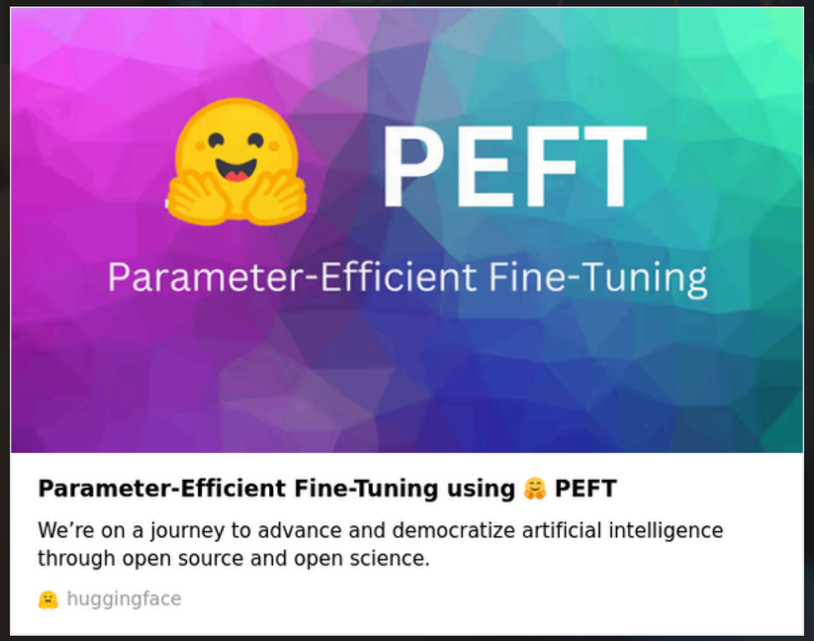
1. **Full fine-tuning** often **infeasible** on consumer hardware.
2. **Storing** and **deploying** fine-tuned models independently for each downstream task becomes **expensive**.



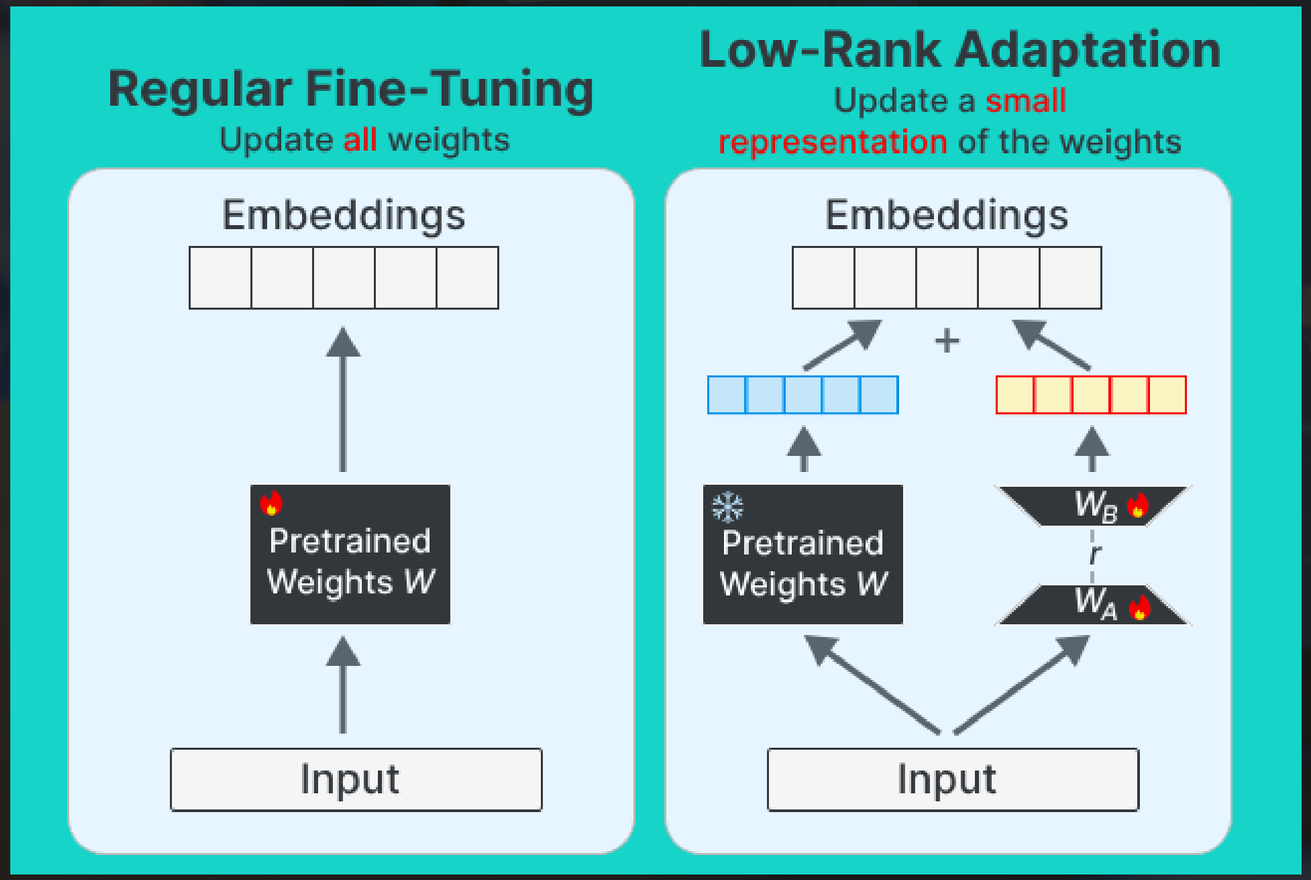
THE SOLUTION

Fine-tune **fewer parameters**

LoRA is the #1 PEFT Method you should know!



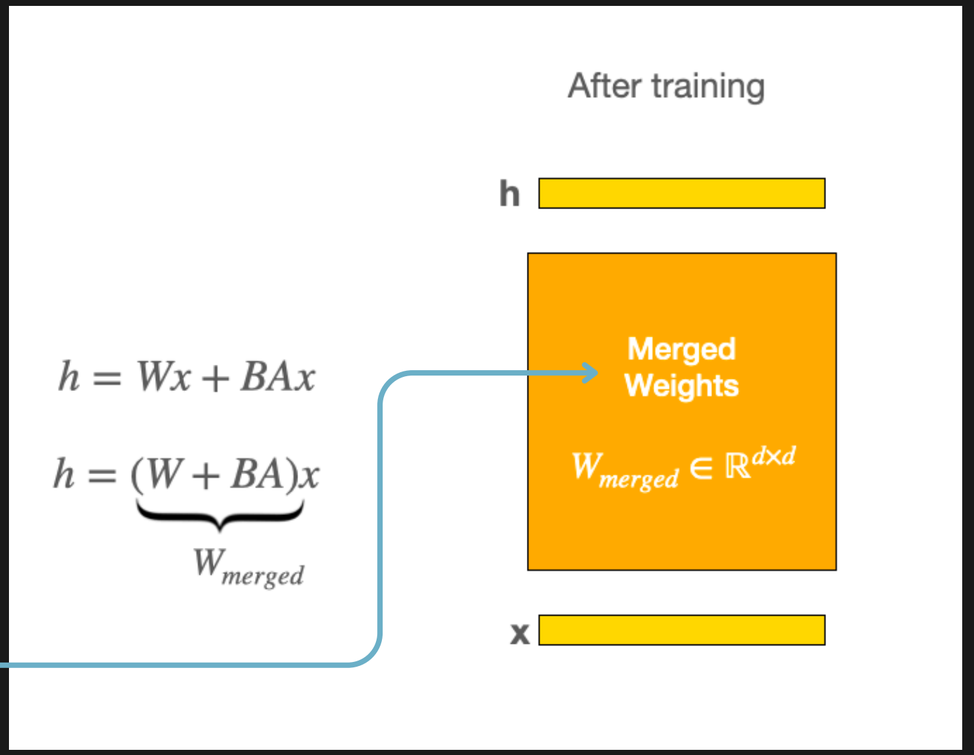
LoRA



Fine-tuning using factorized matrices

LoRA Advantages

- More **efficient**
- **Adapters for many tasks**
- **Combine** with other PEFT methods
- **Comparable** to full fine-tuning
- **No** additional inference **latency**



TO SUMMARIZE

Fine-Tuning

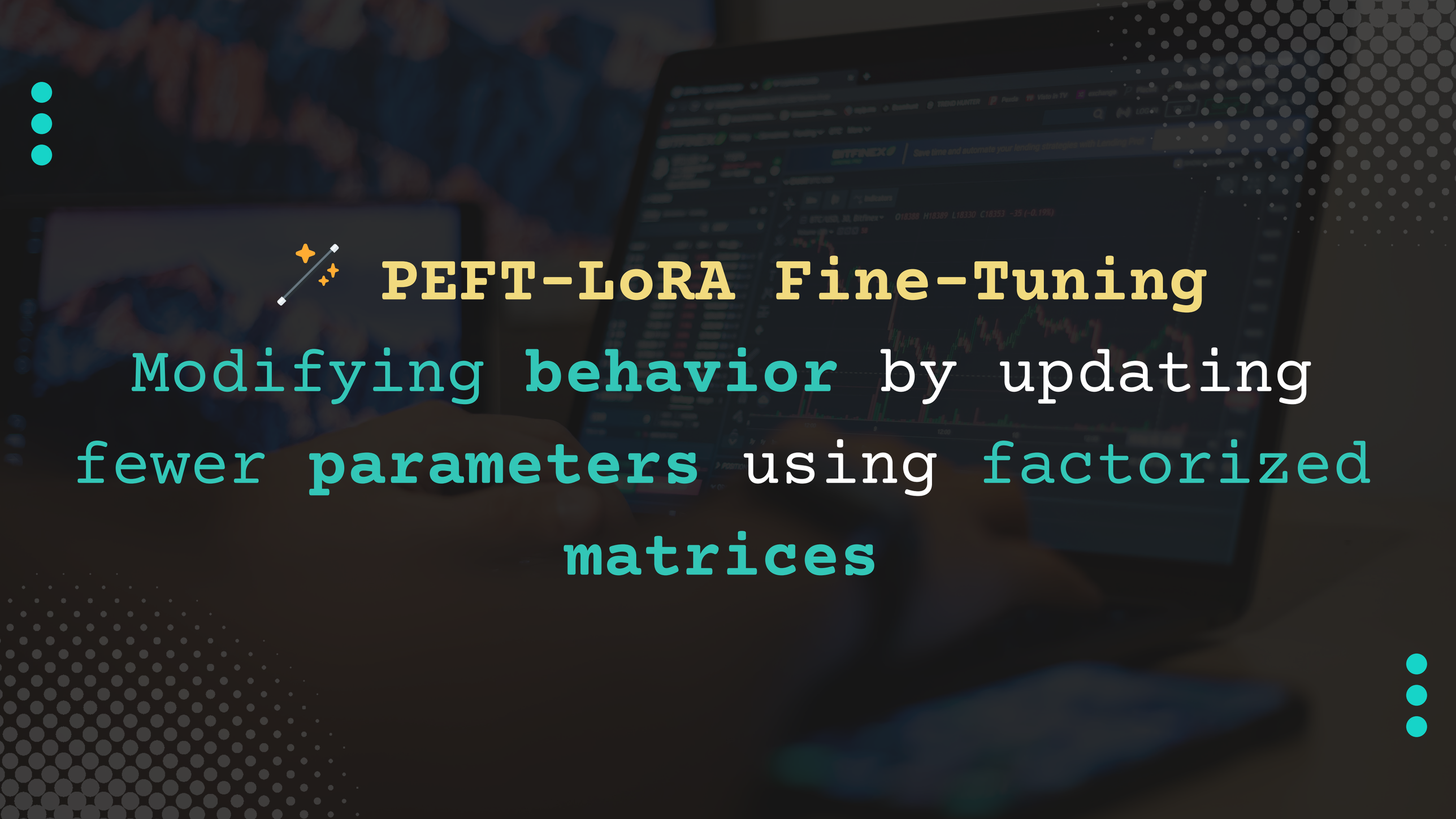
Modifying behavior by updating parameters

PEFT

Fine-tuning w/ fewer parameters

Low Rank Adaption

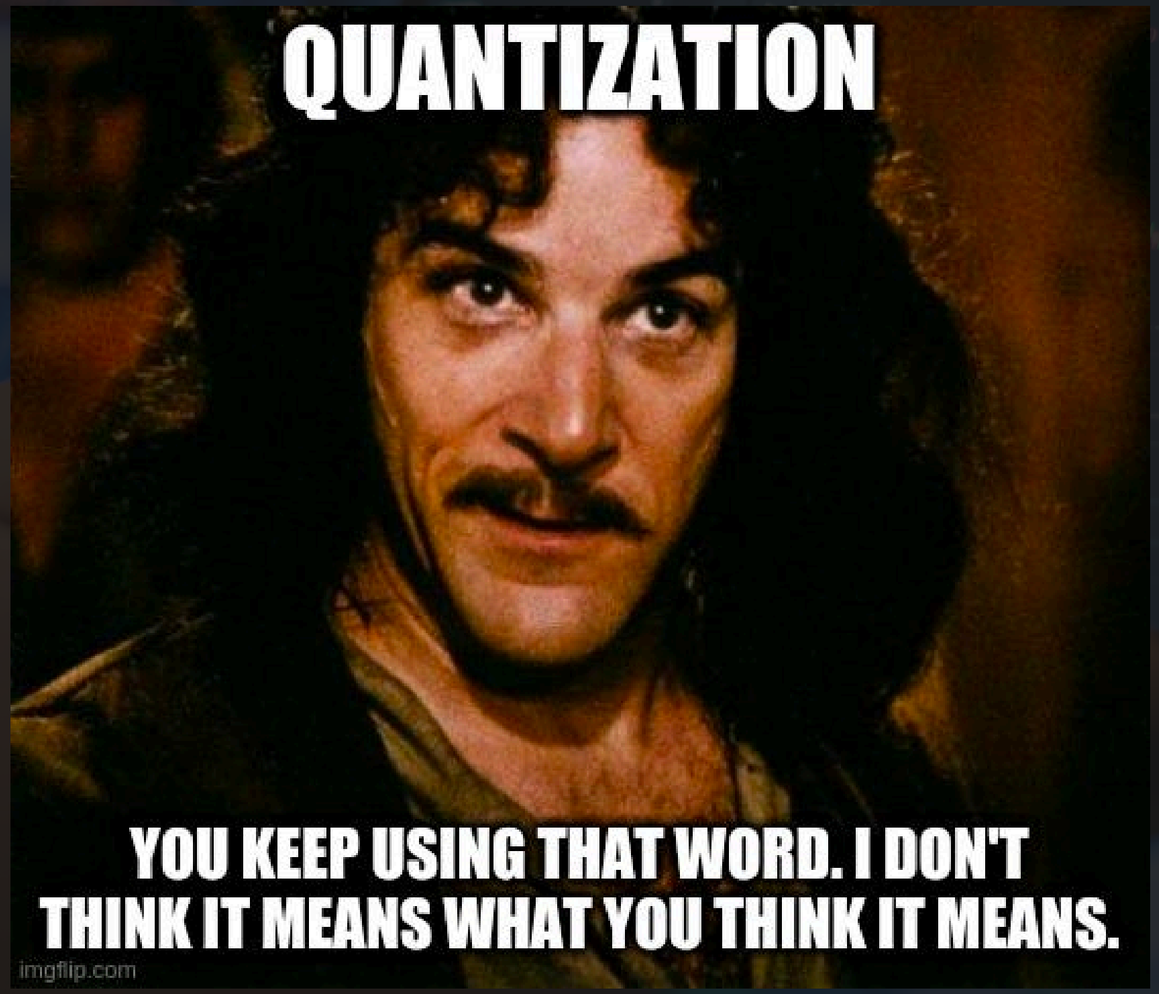
Fine-tuning using factorized matrices



PEFT-LoRA Fine-Tuning

Modifying **behavior** by updating fewer **parameters** using factorized **matrices**

Quantization

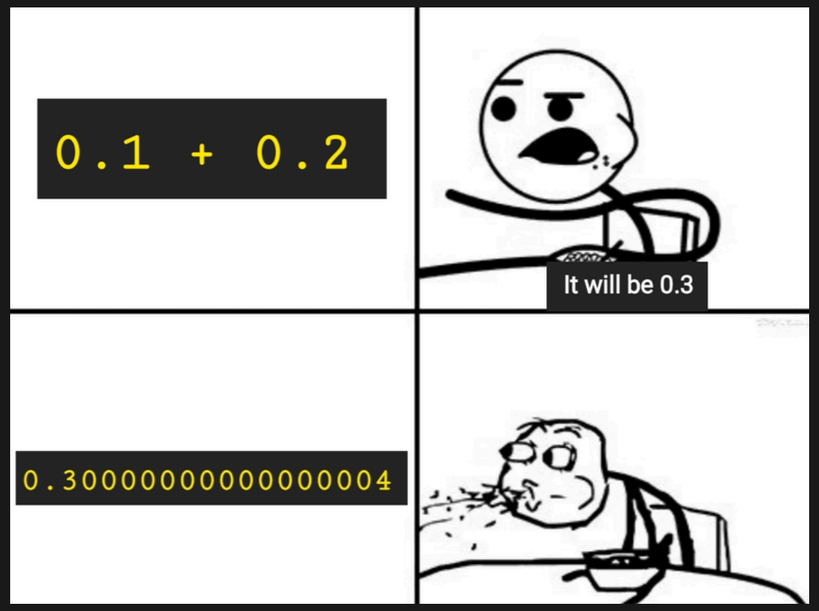


PARAMETERS I

weights = parameters

PARAMETERS II

parameters = floating point numbers



FLOATS

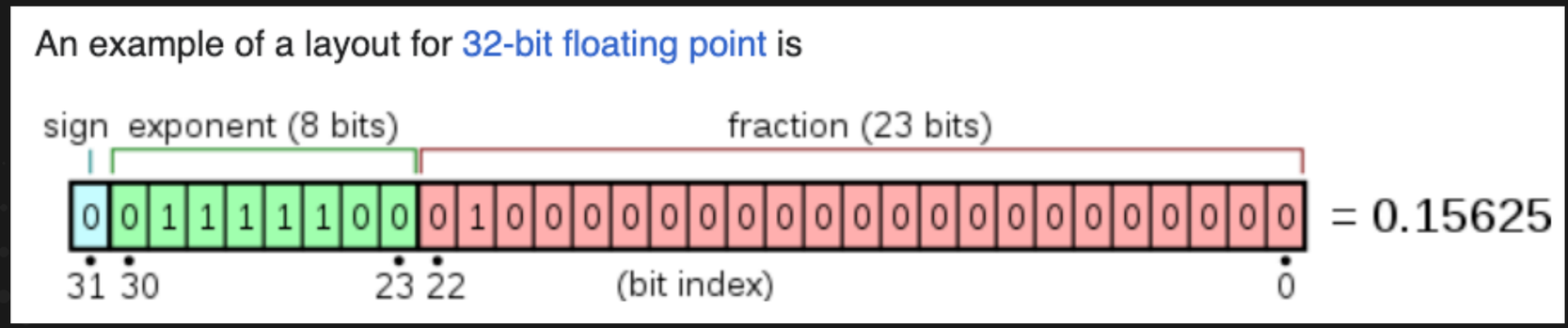
- Decimal point can "float" to any position necessary.

- **Integer** with **fixed precision**

$$12.345 = \underbrace{12345}_{\text{significand}} \times \underbrace{10^{\overbrace{-3}^{\text{exponent}}}}_{\text{base}}$$

PRECISION

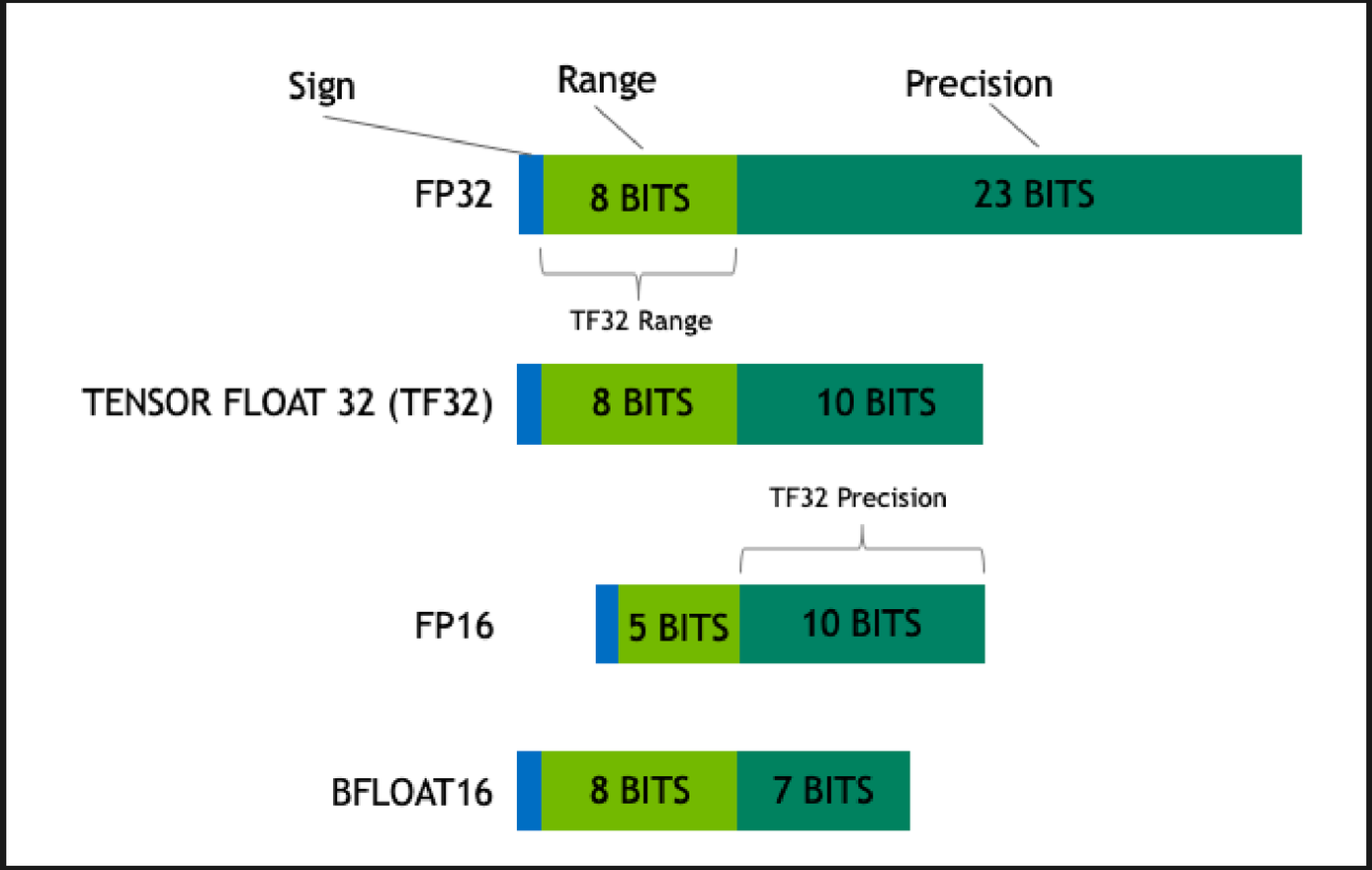
Full precision = 32 bits

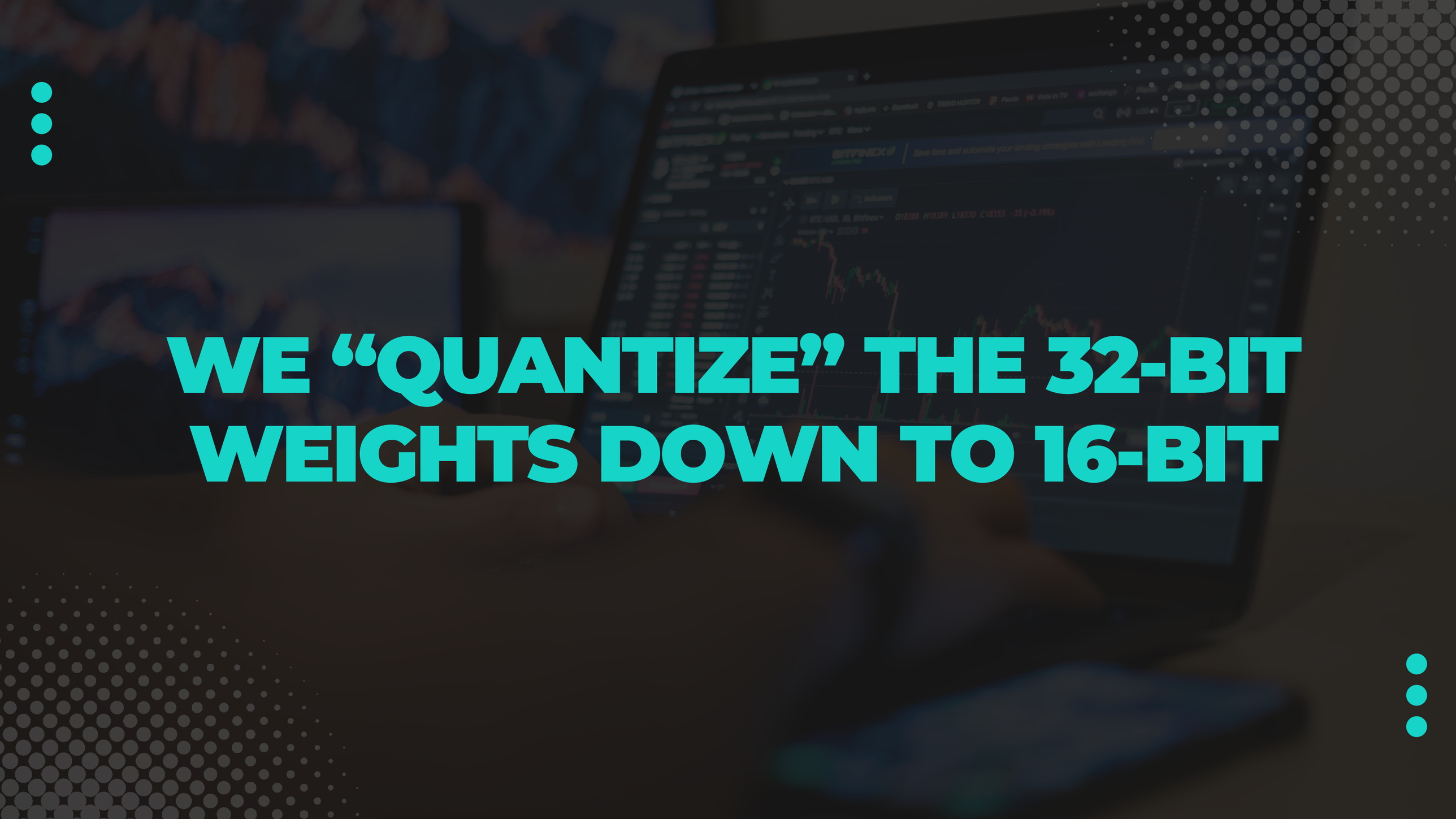


COMMON TYPES IN ML

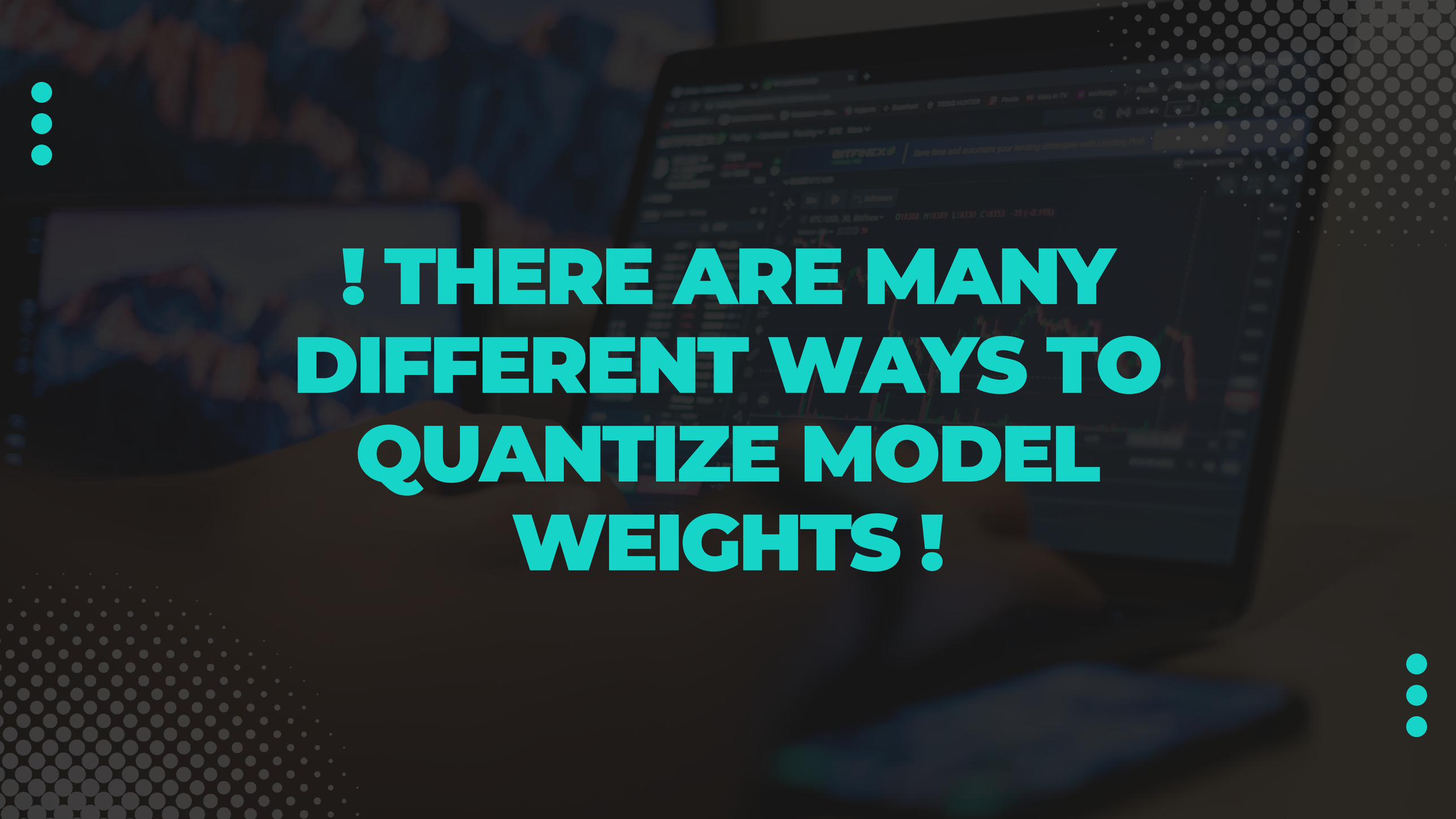
Instead of using the **32-bit** precision, we **can get an almost identical inference outcome** with **16-bit** half-precision.

This halves the model size.



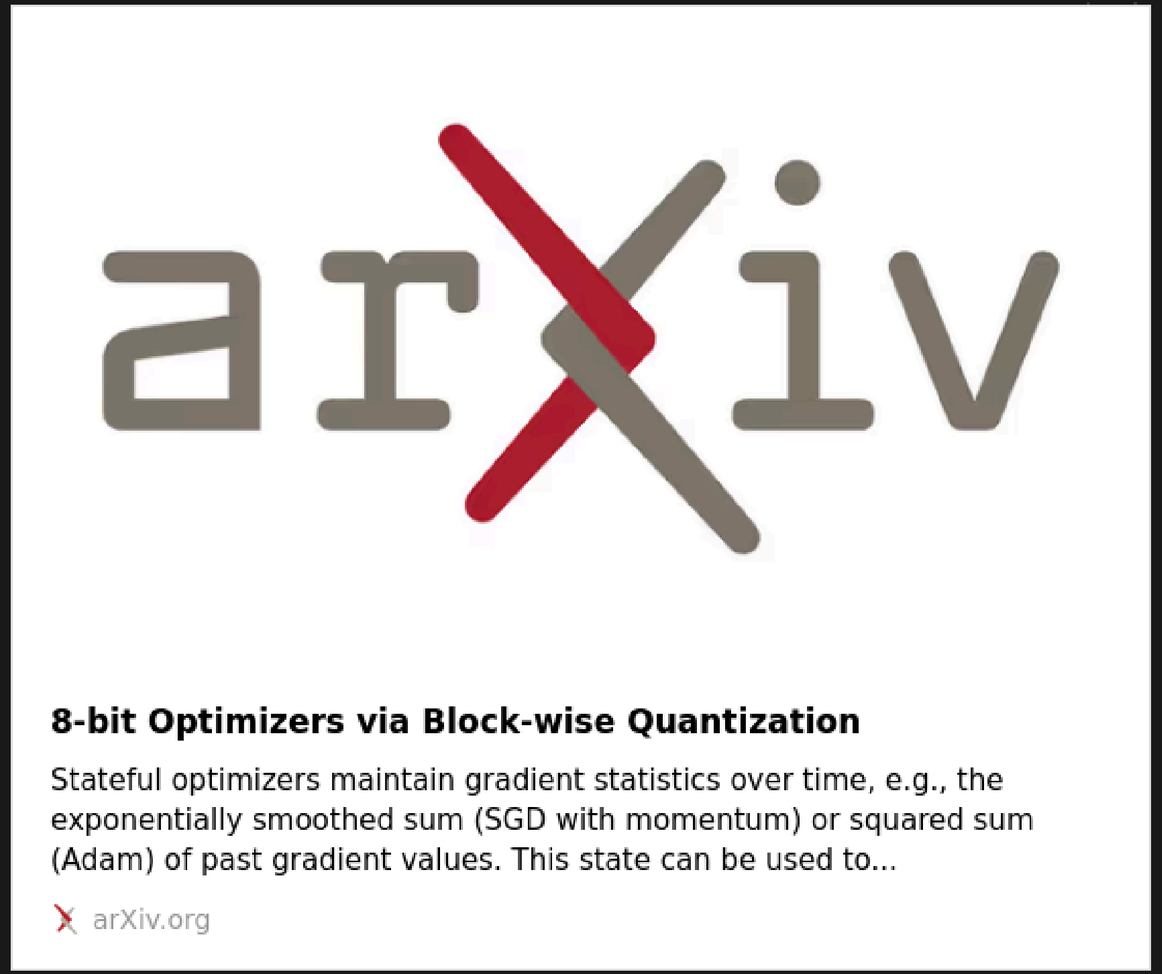


WE “QUANTIZE” THE 32-BIT WEIGHTS DOWN TO 16-BIT


! THERE ARE MANY
DIFFERENT WAYS TO
QUANTIZE MODEL
WEIGHTS !

BITSANDBYTES

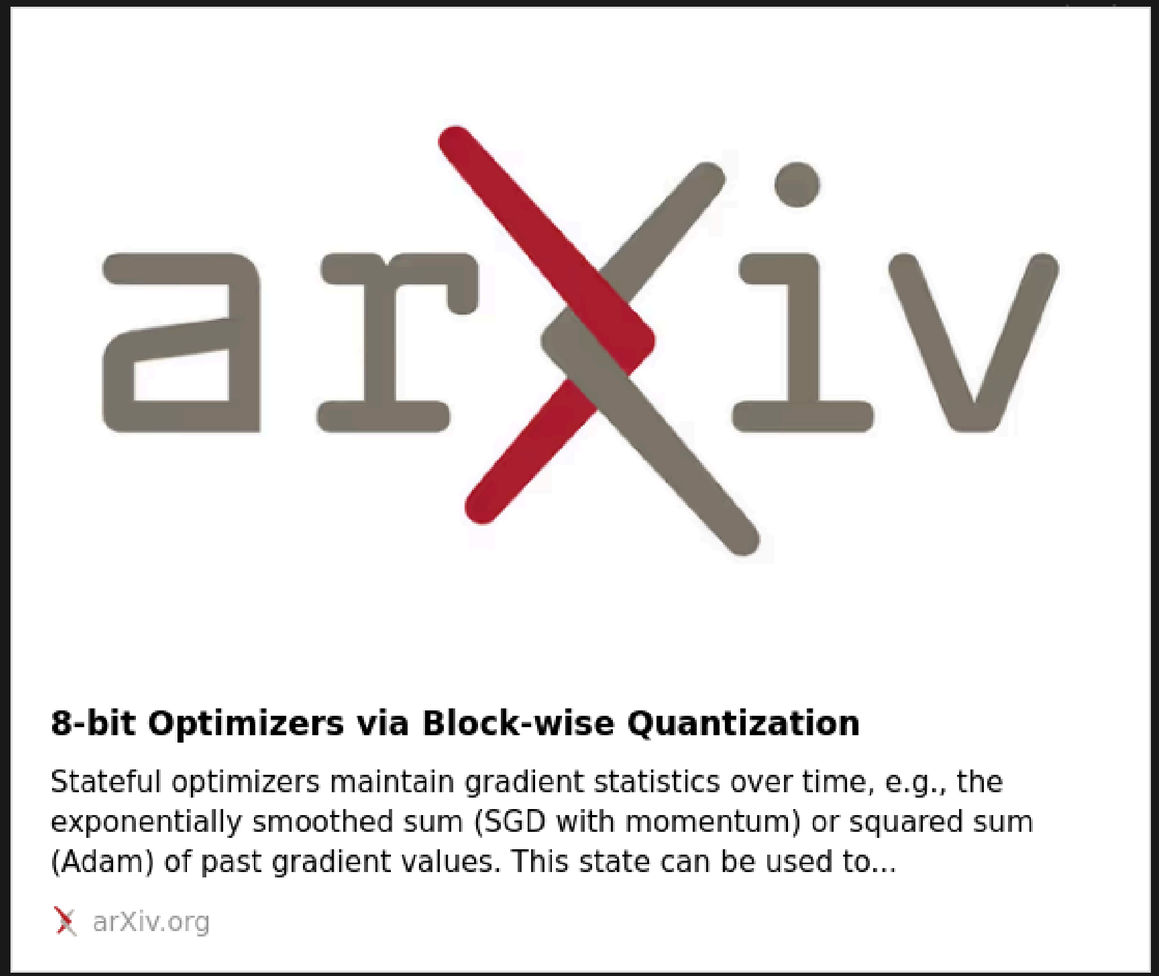
The first optimizers that ***use 8-bit statistics*** *while maintaining the performance levels of using* ***32-bit*** *optimizer* ***states***



BITSANDBYTES TRADEOFFS

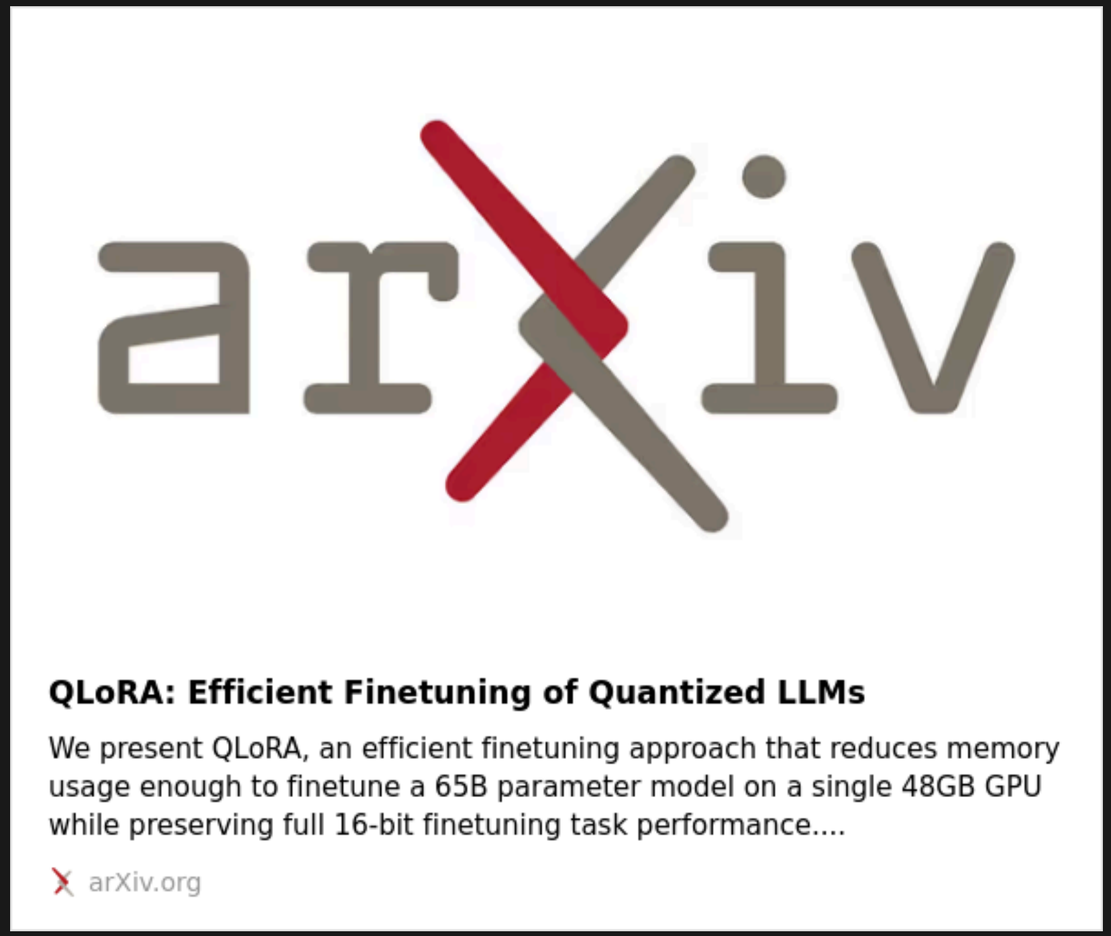
*Does **not provide** any inference **latency benefits***

*Allows for **flexible use with LoRA adapters***



QLoRA

Train model using quantization (4-bit) and dequantization (16-bit)!

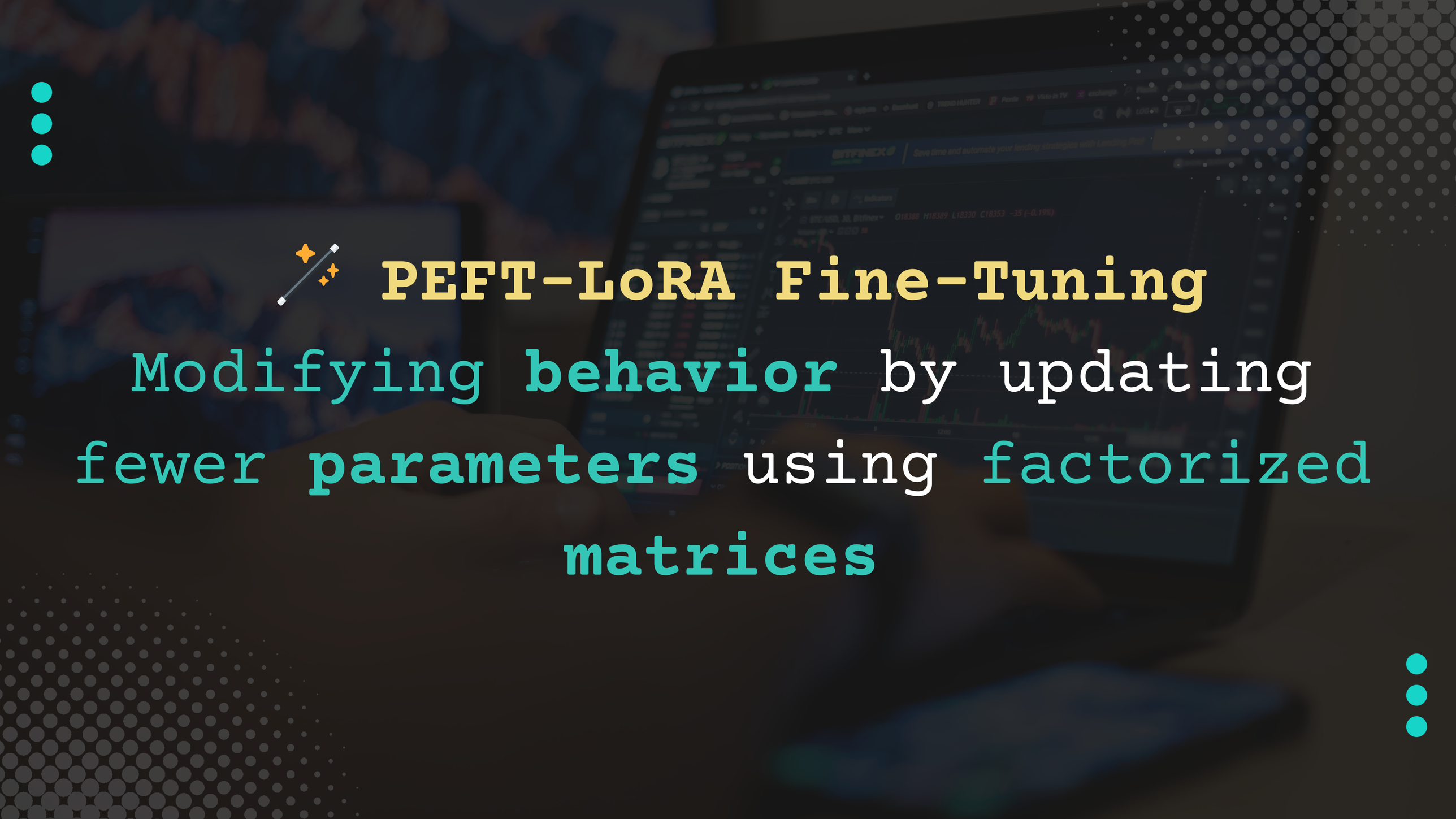


Great, 8x less precise

*"**16-bit finetuning** of a **LLaMA 65B** parameter model [57] requires more than **780 GB of GPU memory**"*

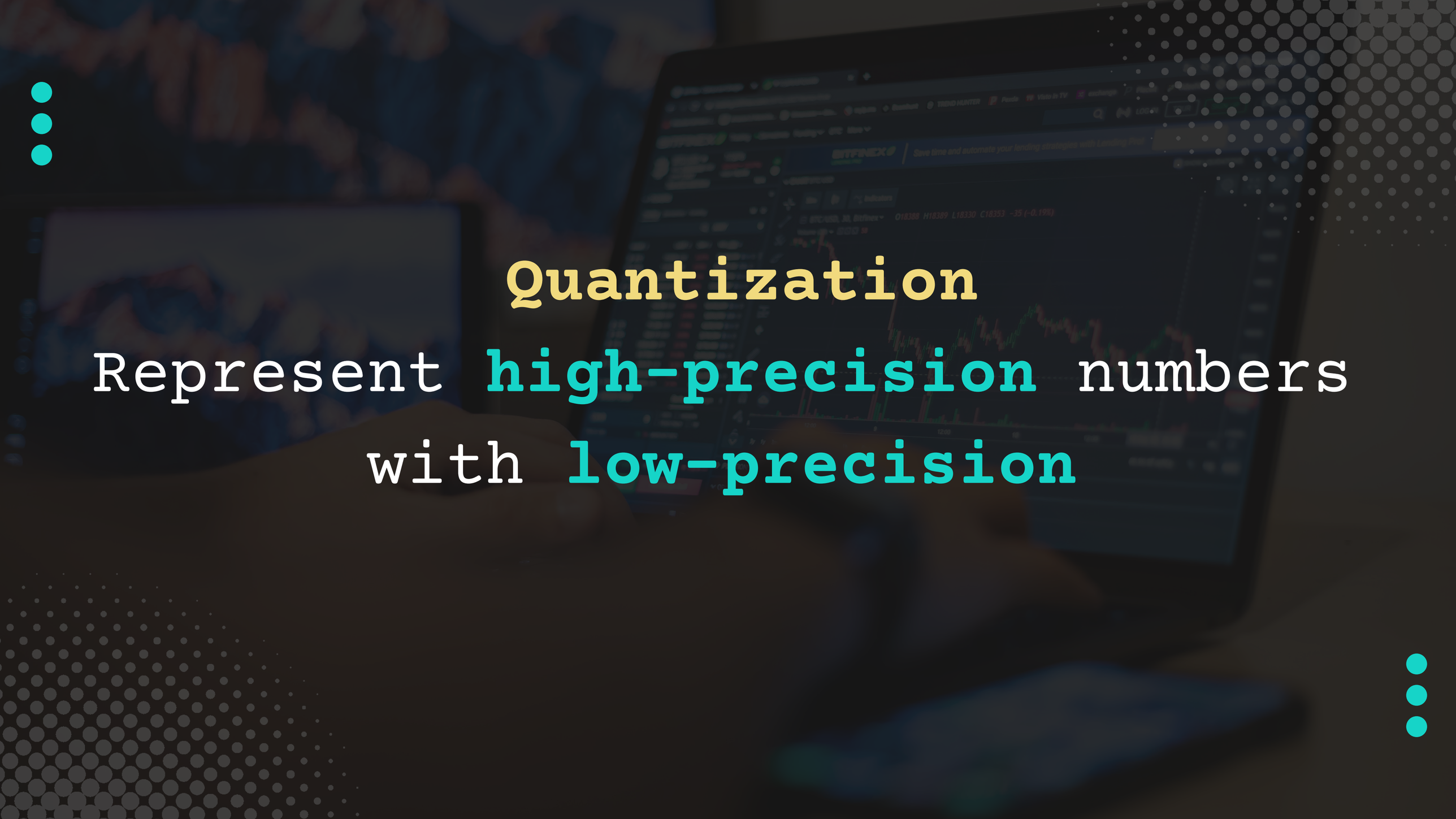
Abstract

We present QLoRA, an efficient finetuning approach that reduces memory usage enough to finetune a 65B parameter model on a single 48GB GPU while preserving full 16-bit finetuning task performance. QLoRA backpropagates gradients through a frozen, 4-bit quantized pretrained language model into Low Rank Adapters (LoRA). Our best model family, which we name **Guanaco**, outperforms all previous openly released models on the Vicuna benchmark, reaching 99.3% of the performance level of ChatGPT while only requiring 24 hours of finetuning on a single GPU. QLoRA introduces a number of innovations to save memory without sacrificing performance: (a) 4-bit NormalFloat (NF4), a new data type that is information theoretically optimal for normally distributed weights (b) Double Quantization to reduce the average memory footprint by quantizing the quantization constants, and (c) Paged Optimizers to manage memory spikes. We use QLoRA to finetune more than 1,000 models, providing a detailed analysis of instruction following and chatbot performance across 8 instruction datasets, multiple model types (LLaMA, T5), and model scales that would be infeasible to run with regular finetuning (e.g. 33B and 65B parameter models). Our results show that QLoRA finetuning on a small high-quality dataset leads to state-of-the-art results, even when using smaller models than the previous SoTA. We provide a detailed analysis of chatbot performance based on both human and GPT-4 evaluations showing that GPT-4 evaluations are a cheap and reasonable alternative to human evaluation. Furthermore, we find that current chatbot benchmarks are not trustworthy to accurately evaluate the performance levels of chatbots. A lemon-picked analysis demonstrates where **Guanaco** fails compared to ChatGPT. We release all of our models and code, including CUDA kernels for 4-bit training.[2]



PEFT-LoRA Fine-Tuning

Modifying **behavior** by updating fewer **parameters** using factorized **matrices**



Quantization

Represent high-precision numbers with low-precision

PEFT-QLoRA Fine-Tuning

Modifying **behavior** by updating fewer **quantized** **parameters** using factorized **matrices**

Process

1. Download weights (FP = 32-bit)
2. Load PEFT model
3. Initialize LoRA config
4. Train model using quantization (4-bit) and dequantization (16-bit)!

```
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16
)
```

TODAY'S BUILD

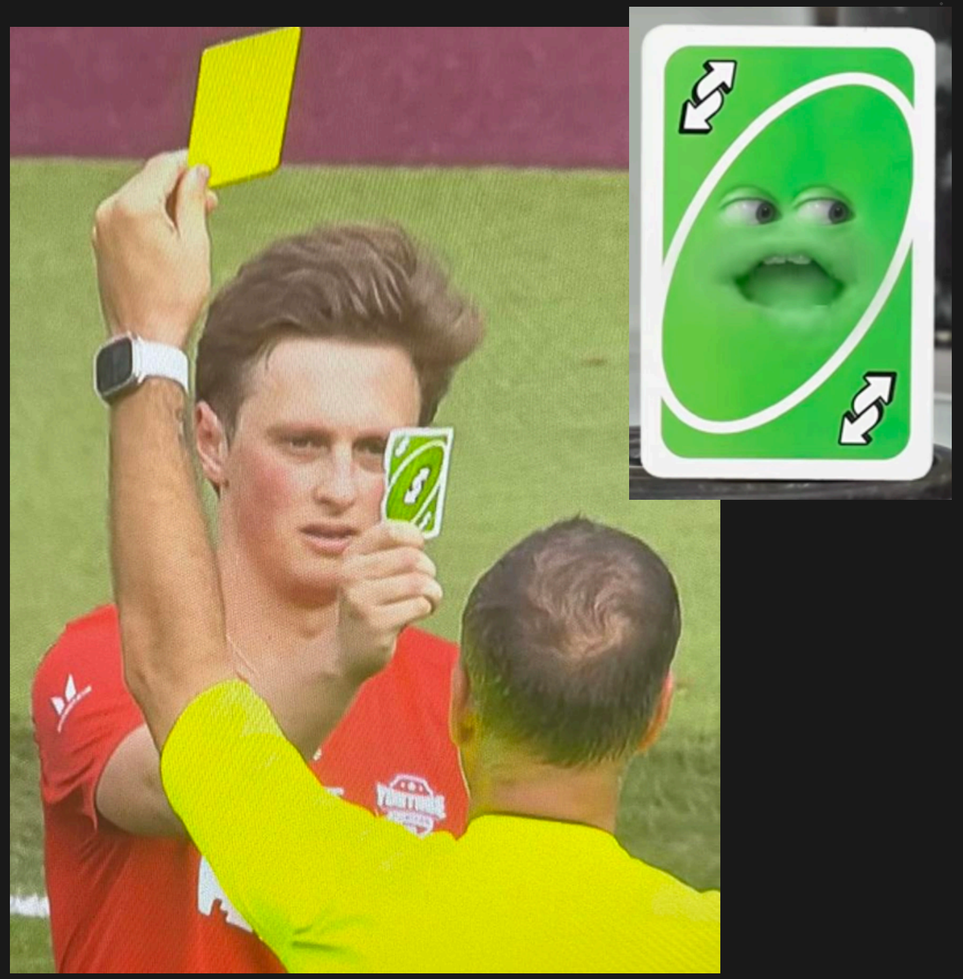
AN UNO REVERSE CARD APPLICATION

Given:

- **Response** (LLM Output)

Predict:

- **Instruction** (Prompt/LLM Input)

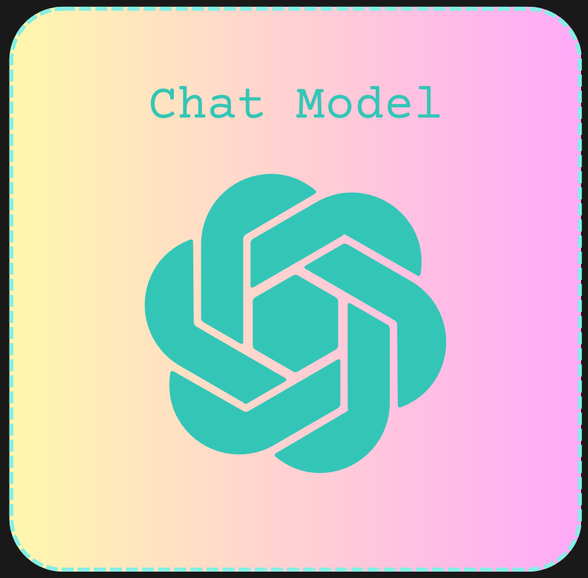


OUR MODEL

CHAT (LLM) MODEL

Chat Model (e.g., LLM)

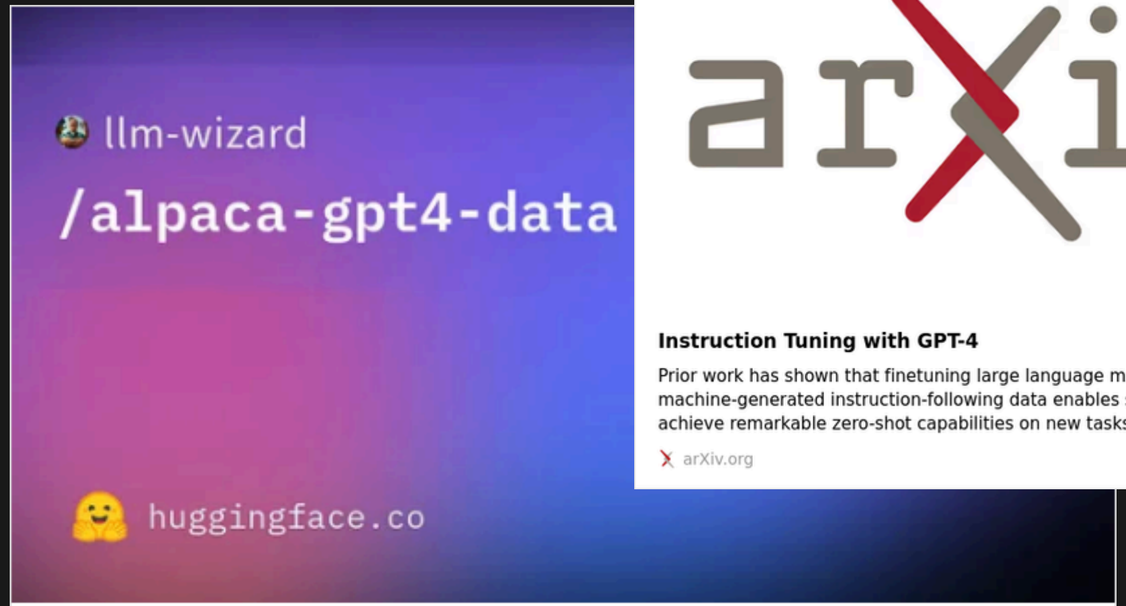
- Mistral 7B



OUR FINE-TUNING DATA

Fine-Tuning Dataset

output string · *lengths* 1 4.57k	instruction string · *lengths* 9 489
1. Eat a balanced and nutritious diet: Make sure…	Give three tips for staying healthy.
The three primary colors are red, blue, and yellow.…	What are the three primary colors?
An atom is the basic building block of all…	Describe the structure of an atom.



llm-wizard/alpaca-gpt4-data · Datasets at Hugging Face

We're on a journey to advance and democratize artificial intelligence through open source and open science.

huggingface

Instruction Tuning with GPT-4

Prior work has shown that finetuning large language models (LLMs) using machine-generated instruction-following data enables such models to achieve remarkable zero-shot capabilities on new tasks,...

arXiv.org

Fine-Tuning with PEFT-QLoRA!

Presented by

Chris Alexiuk, LLM Wizard 🪄

CONCLUSIONS

- 🪄 **PEFT-QLoRA Fine-Tuning:** *Modifying **behavior** by updating **fewer** **quantized** **parameters** using **factorized** matrices*
  - *PEFT-LoRA (3.8B params --> 2.7M params)*

- **Quantization** (Block-wise k-bit quantization)
  - Express **more** with **less** information
  - **Train** model using **quantization** (4-bit) and **dequantization** (16-bit)!

- Next stop … inference, serving, and vLLM!

QUESTIONS?

Thank you!