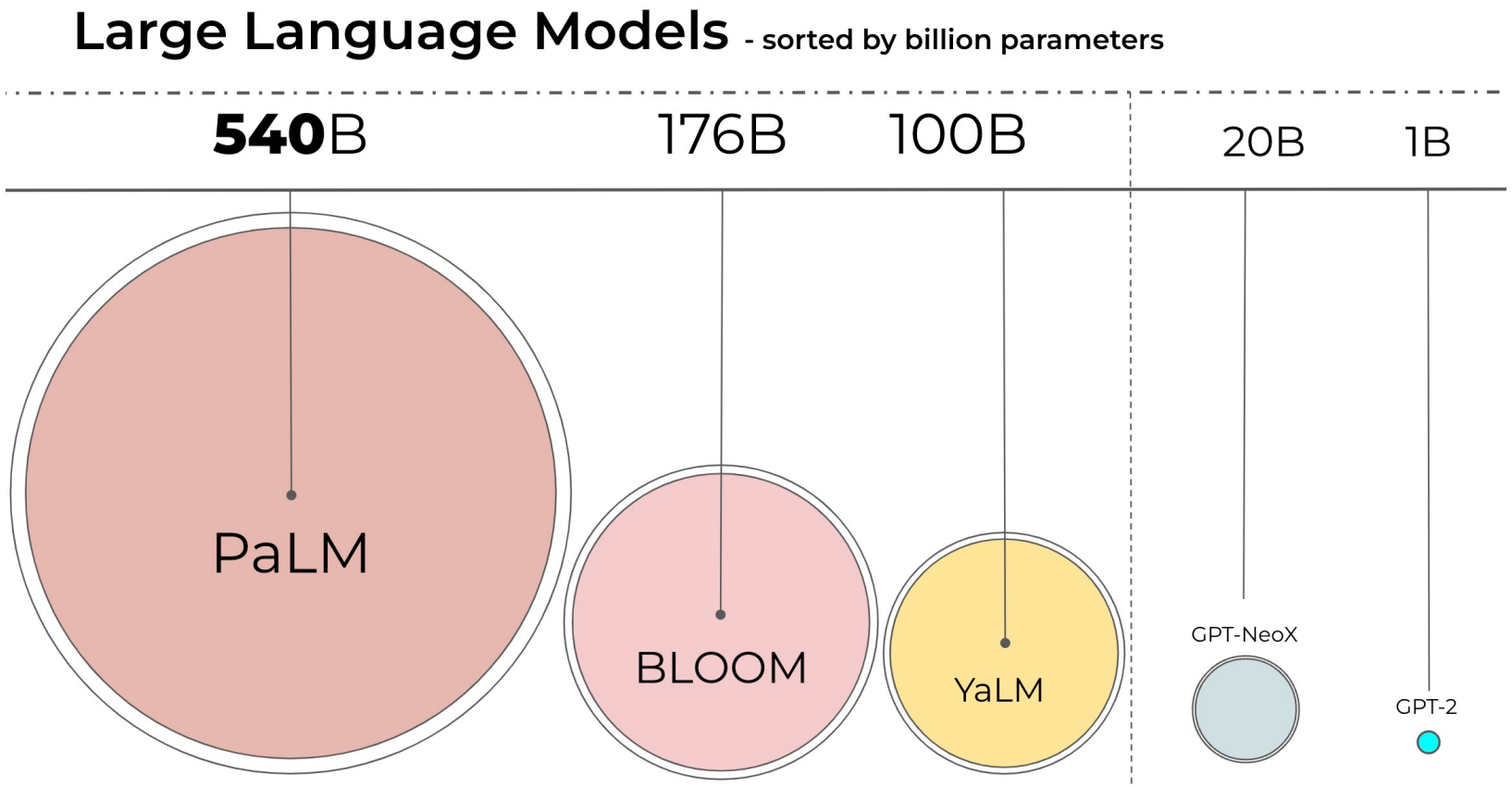
Paper Review: *LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale*

Songhe Wang

CSE 587 Spring 2023

LLM Nowadays

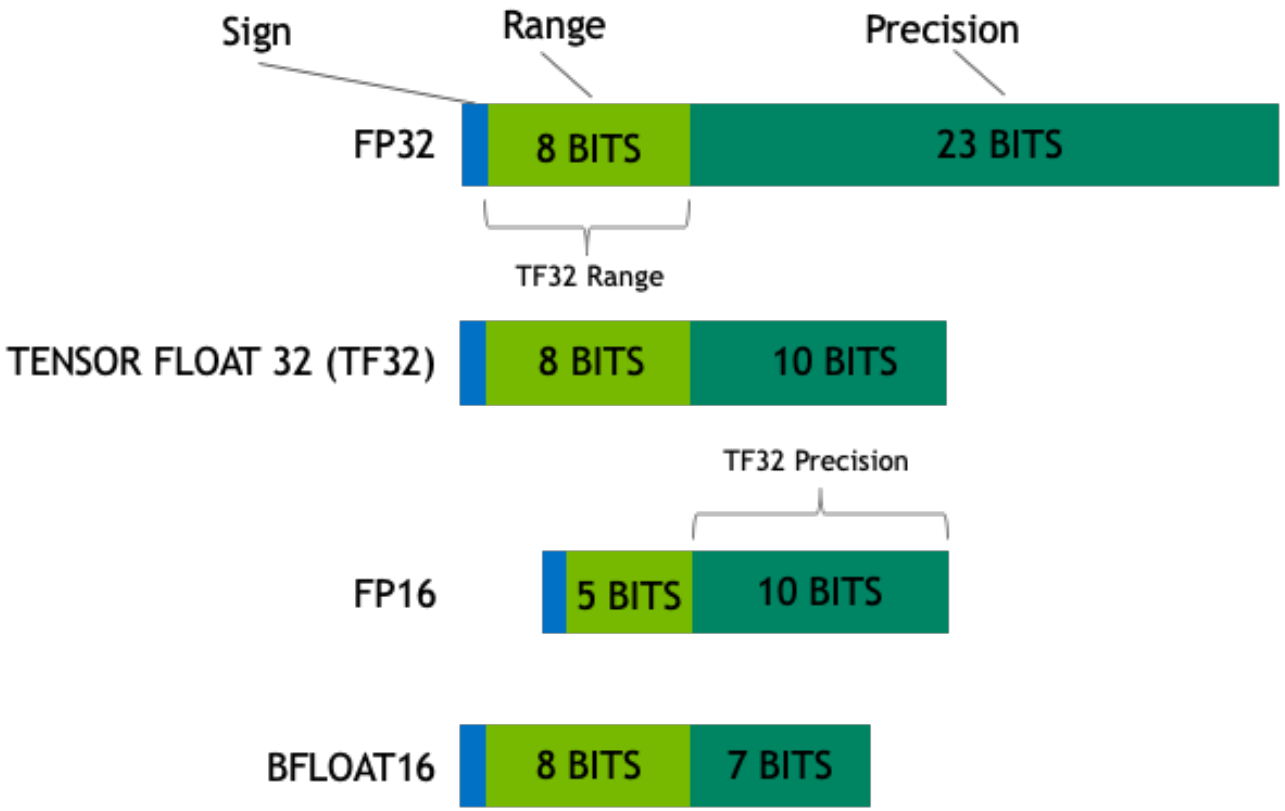


Computational Resources

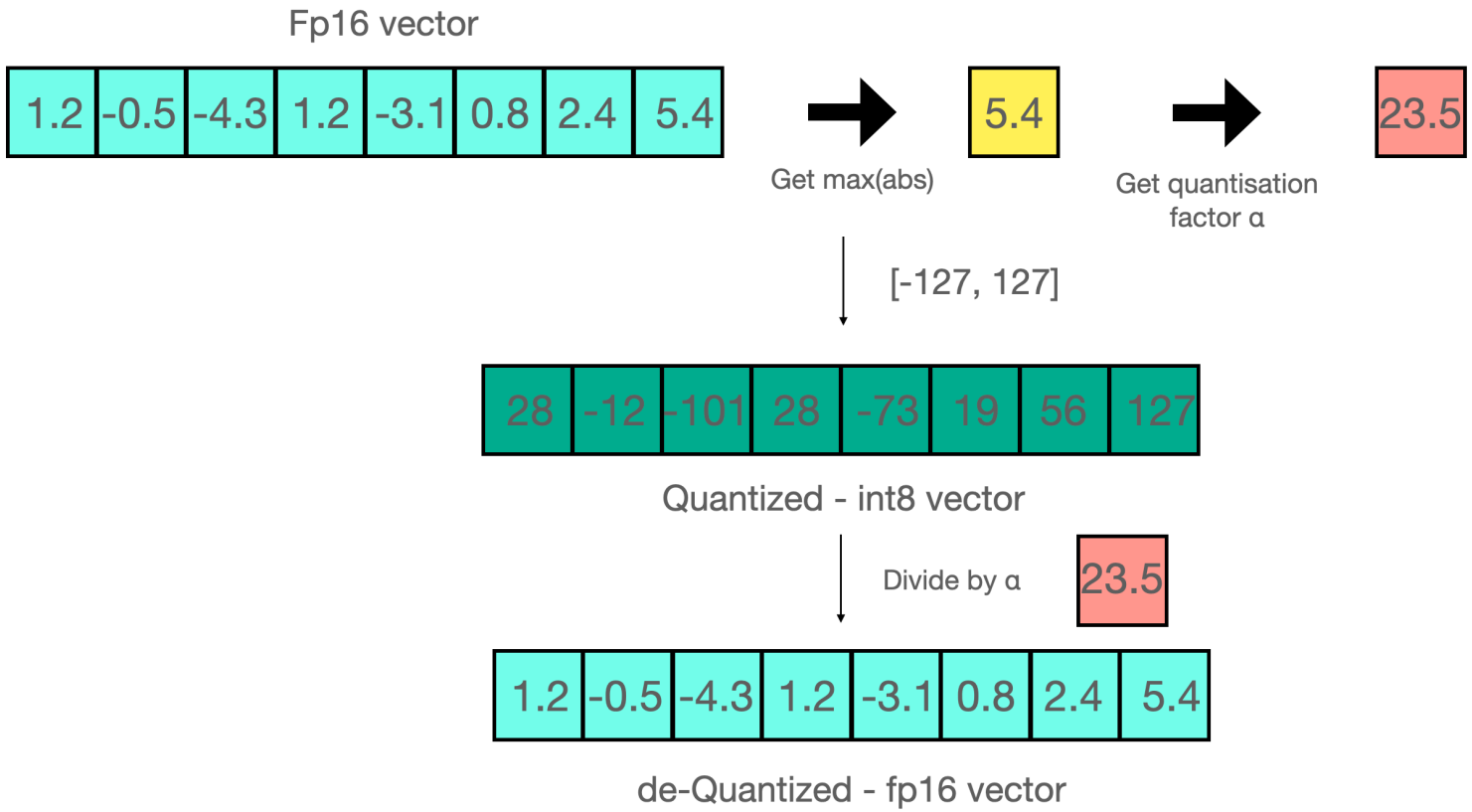
- Inference on BLOOM-176B: 8x 80GB A100 GPUs (~$15k each)
- Fine-tune BLOOM-176B: 72x 80GB A100 GPUs models
- PaLM 540B: much more

How can we reduce the size of the these huge models?

Background: Common Data Type

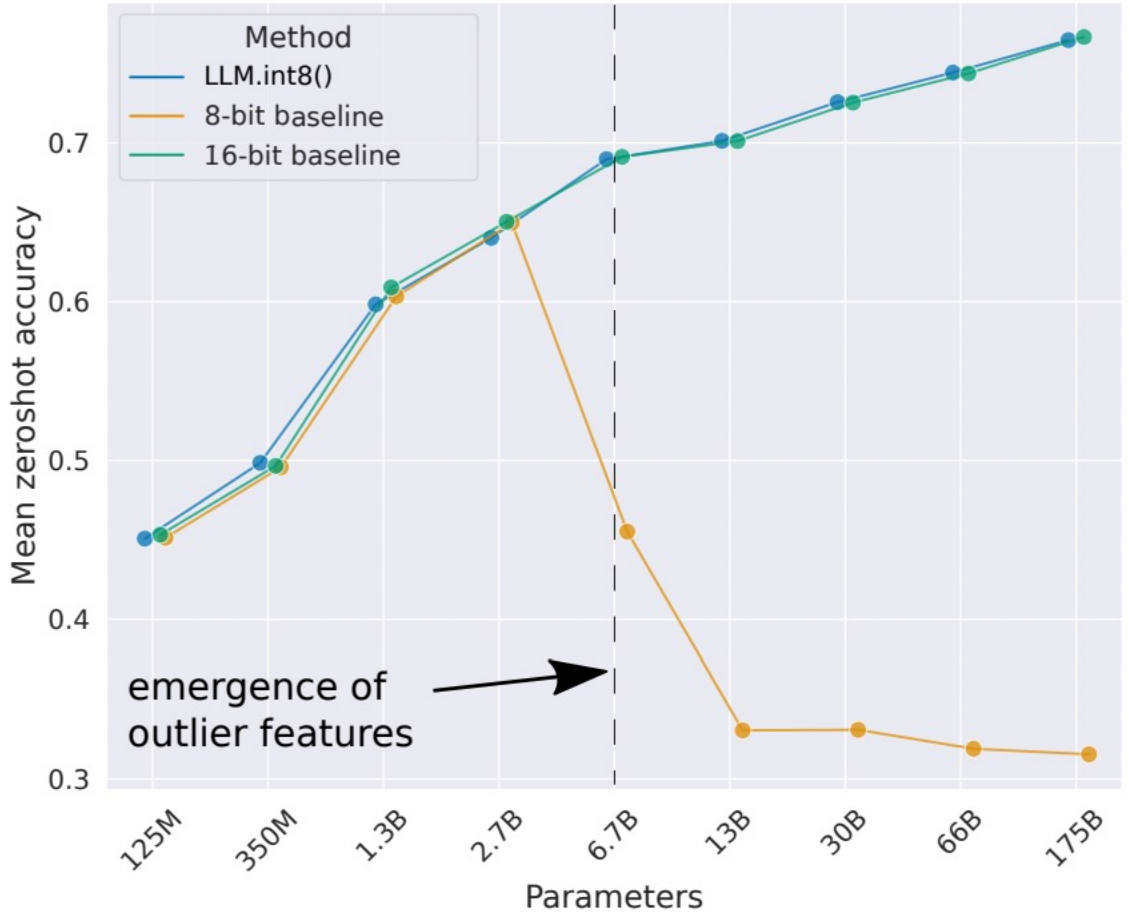


Absmax Quantization



Limitations

- The performance is retained at scales up to 2.7B parameters
- The accuracy drops drastically when the model is larger than 6.7B
- The problem is the outlier features



Mixed-precision Decomposition

Performance: 0 degradation

benchmarks	**-**	**-**	**-**	**-**	**difference - value**
name	metric	value - int8	value - bf16	std err - bf16	-
hellaswag	acc_norm	0.7274	0.7303	0.0044	0.0029
hellaswag	acc	0.5563	0.5584	0.005	0.0021
piqa	acc	0.7835	0.7884	0.0095	0.0049
piqa	acc_norm	0.7922	0.7911	0.0095	0.0011
lambada	ppl	3.9191	3.931	0.0846	0.0119
lambada	acc	0.6808	0.6718	0.0065	0.009
winogrande	acc	0.7048	0.7048	0.0128	0

Is it faster? No.

Precision	Number of parameters	Hardware	Time per token in milliseconds for Batch Size 1	Time per token in milliseconds for Batch Size 8	Time per token in milliseconds for Batch Size 32
bf16	176B	8xA100 80GB	239	32	9.9
int8	176B	4xA100 80GB	282	37.5	10.2
bf16	176B	14xA100 40GB	285	36.5	10.4
int8	176B	5xA100 40GB	367	46.4	oom
fp16	11B	2xT4 15GB	11.7	1.7	0.5
int8	11B	1xT4 15GB	43.5	5.3	1.3
fp32	3B	2xT4 15GB	45	7.2	3.1
int8	3B	1xT4 15GB	312	39.1	10.2

The 3 models are BLOOM-176B, T5-11B and T5-3B

Questions

Thank you!