0.1 Ergebnisse der Experimente

0.1.1 Optimierungstechnologien für LLMs

In dem folgenden Kapitel werden die Ergebnisse zu den verschiedenen Verfahren der Modell Optimierung vorgestellt. Dabei handelt es sich um die Verfahren:

- Quantization
- Pruning
- Knowledge Distillation
- Low Rank Optimization (Lora) bzw. Qlora

Für eine Verringerung des Speicherbedarfs eignen sich bevorzugt die Verfahren Quantization und Pruning, während Lora und Knowledge Distillation für das erneute Tuning solcher Modelle eingesetzt werden.

Quantization

Benchmark Ergebnisse der AWQ 4bit Quantization im Vergleich zum jeweiligen Referenz Modell.

Benchmarks:	winogrande	truthfulqa_mc2	hellaswag	arc_challenge	Durchschnitt
Mistral 7 Referenz	0,7395	0,6682	0,8365	0,5606	0,7012
Standard Fehler	*0,0123*	*0,0152*	*0,0037*	*0,0145*	
Mistral 7 AWQ	0,7403	0,6752	0,8310	0,5700	0,7041
Standard Fehler	*0,0123*	*0,0151*	*0,0037*	*0,0145*	
Llama 3 Referenz	0,7206	0,5165	0,7582	0,5674	0,6407
Standard Fehler	*0,0126*	*0,0152*	*0,0043*	*0,0145*	
Llama 3 AWQ	0,7356	0,5099	0,7532	0,5572	0,6390
Standard Fehler	*0,0124*	*0,0152*	*0,0043*	*0,0145*	
Llama 2 Referenz	0,6646	0,4532	0,7547	0,4420	0,5786
Standard Fehler	*0,0133*	*0,0156*	*0,0043*	*0,0145*	
Llama 2 AWQ	0,6456	0,4509	0,7481	0,4462	0,5727
	0,0134	*0,0156*	*0,0043*	*0,0145*	

Tabelle 1: Die Tabelle zeigt die quantisierten Modellen zusammen mit den korrespondierenden Referenzen. Für die Quantisierung wurde das Verfahren AutoAWQ verwendet.

Speicherbedarf der AWQ Modelle im Vergleich zur jeweiligen Referenz

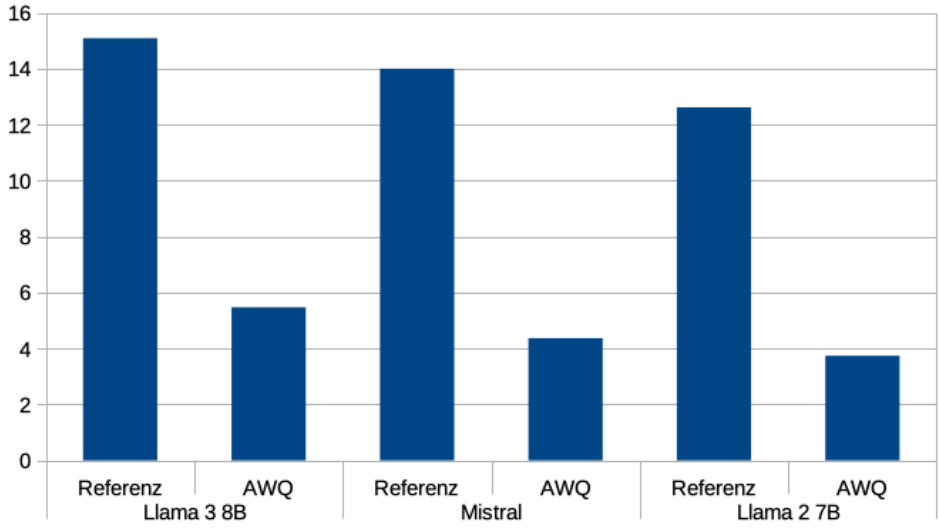

Abbildung 1: Auslastung des GPU Speichers der jeweiligen Modelle auf einer Nvidia A100 80GB GPU.

Pruning

Die Tabelle zeigt die Benchmark Ergebnisse für das Pruning der Referenz Modelle mittels PrunMe in jeweils zwei verschiedenen Ausprägungen

Benchmarks:	Layers Pruned	winogrande	truthfulqa_mc2	hellaswag	arc_challenge	Durchschnitt
Llama 3 8B	**0**	0,7206	0,5165	0,7582	0,5674	0,6407
	Standard Fehler	*0,0126*	*0,0152*	*0,0043*	*0,0145*	
	8	0,6369	0,5211	0,5897	0,4283	0,5440
	Standard Fehler	*0,0135*	*0,0159*	*0,0049*	*0,0145*	
	12	0,5706	0,5201	0,4204	0,3208	0,4580
	Standard Fehler	*0,0139*	*0,0164*	*0,0049*	*0,0136*	
Mistral 7 B	**0**	0,7395	0,6682	0,8365	0,5606	0,7012
	Standard Fehler	*0,0123*	*0,0152*	*0,0037*	*0,0145*	
	8	0,6811	0,6313	0,6646	0,4044	0,5954
	Standard Fehler	*0,0131*	*0,0158*	*0,0047*	*0,0143*	
	12	0,5943	0,5543	0,3239	0,3677	0,4601
	Standard Fehler	*0,0138*	*0,0167*	*0,0047*	*0,0141*	

Tabelle 2: Die Tabelle zeigt Modelle bei denen eine unterschiedliche Anzahl von Schichten mittels Pruning entfernt wurde.

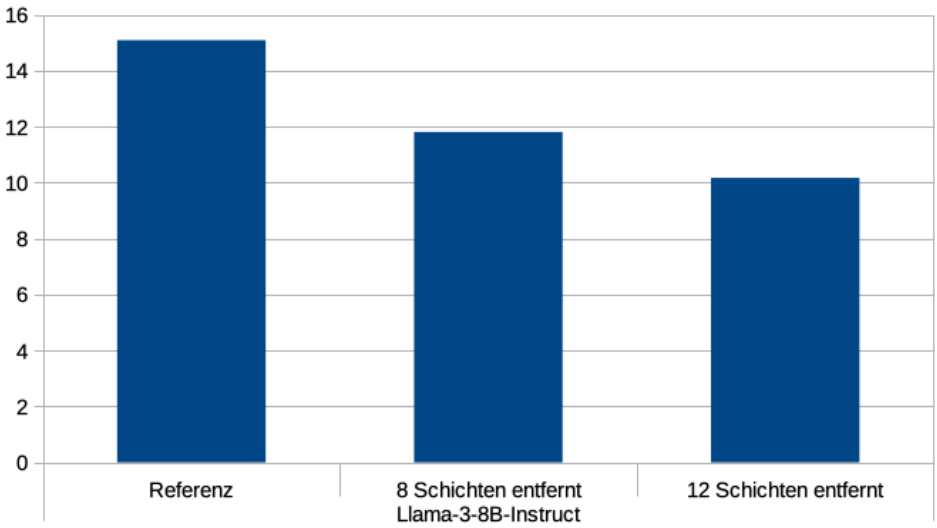

Abbildung 2: Auslastung des GPU Speichers der jeweiligen Modelle auf einer Nvidia A100 80GB GPU.

0.1.2 Kombination von Verfahren zur Kompression von LLM Modellen

Tuning eines geprunten Modells mittels Lora

Die Tabelle zeigt die Benchmark Ergebnisse für die Anwendung des Lora Verfahrens auf geprunte Modelle.

Modell	Verfahren	winogrande	truthfulqa_mc2	hellaswag	arc_challenge	Durchschnitt
Llama 3 8B Instruct	pruned	0,7111	0,5000	0,6255	0,4625	0,5748
		0,0127	*0,0157*	*0,0048*	*0,0146*	
	pruned + lora	0,7316	0,5189	0,7269	0,4974	0,6187
		0,0125	*0,0151*	*0,0044*	*0,0146*	
Mistral 7B Instruct v0.2	pruned	0,7072	0,6370	0,7298	0,4599	0,6335
		0,0128	*0,0158*	*0,0044*	*0,0146*	
	pruned + lora	0,7174	0,5774	0,7457	0,4855	0,6315
		0,0127	*0,0154*	*0,0043*	*0,0146*	
Llama 2 7B chat	pruned	0,6409	0,4955	0,6384	0,3968	0,5429
		0,0135	*0,0159*	*0,0048*	*0,0143*	
	pruned + lora	0,6732	0,4886	0,7082	0,4206	0,5727
		0,0132	*0,0154*	*0,0045*	*0,0144*	

Tabelle 3: Die Tabelle zeigt den Effekt der Anwendung des Lora Verfahrens auf die geprunten Modelle.

Tuning eines geprunten Modells mittels Knowledge Distillation

Einfluss von Knowledge Distillation auf die Ergebnisse.

Benchmarks:	winogrande	truthfulqa_mc2	hellaswag	arc_challenge	Durchschnitt
LLama 3 8B pruned lora	0,7316	0,5189	0,7269	0,4974	0,6187
Fehler	*0,0125*	*0,0151*	*0,0044*	*0,0146*	
LLama 3 8B pruned lora dist	0,6535	0,4425	0,6645	0,4386	0,5498
Fehler	*0,0134*	*0,0166*	*0,0047*	*0,0145*	
Mistral pruned lora	0,7174	0,5774	0,7457	0,4855	0,6315
Fehler	*0,0127*	*0,0154*	*0,0043*	*0,0146*	
Mistral pruned lora dist	0,6330	0,4654	0,6694	0,4437	0,5529
Fehler	*0,0135*	*0,0162*	*0,0047*	*0,0145*	

Tabelle 4: Optimierungsversuch mit Knowledge Distillation auf Modelle die mit pruning verkleinert und mit lora wieder traniert wurden

Vergleichende Darstellung der erzeugten Varianten am Beispiel von Llama 3 8B Instruct

Die nachfolgende Tabelle zeigt die Benchmark Ergebnisse der zuvor beschrieben Llama 3 Modelle.

Benchmarks:	**winogrande**	**truthfulqa_mc2**	**hellaswag**	**arc_challenge**	**Durchschnitt**
Llama 3 8B instruct referenz	0,7210	0,5160	0,7580	0,5670	0,6400
Standard Fehler	*0,0130*	*0,0150*	*0,0040*	*0,0150*	
Llama 3 8B instruct pruned	0,7110	0,5000	0,6250	0,4620	0,5700
Standard Fehler	*0,0130*	*0,0160*	*0,0050*	*0,0150*	
Llama 3 8B instruct awq	0,7370	0,5099	0,7529	0,5580	0,6395
Standard Fehler	*0,0124*	*0,0152*	*0,0043*	*0,0145*	
Llama 3 8B instruct pruned + lora	0,7320	0,5190	0,7270	0,4970	0,6200
Standard Fehler	*0,0130*	*0,0150*	*0,0040*	*0,0150*	
Llama 3 8B instruct pruned + lora + awq	0,7230	0,5270	0,7320	0,4870	0,6200
Standard fehler	*0,0130*	*0,0150*	*0,0040*	*0,0150*	

Tabelle 5: Die Tabelle zeigt die Benchmark Ergebnisse der verschiedenen Llama 3 8B Instruct Modelle im Vergleich.

Perplexity Ergebnisse

Die nachfolgende Tabelle zeigt die Perplexity Ergebnisse für verschiedene vom Referenz Modell abgeleitet Varianten.

Modell: Llama 3 8B instruct	bits_per_byte	byte_perplexity	word_perplexity
Referenz	0,62	1,54	9,94
AWQ	0,63	1,55	10,30
pruned	1,08	2,11	53,86
prune + lora	0,71	1,64	13,96
prune + lora + awq	0,70	1,62	13,24

Tabelle 6: Perplexity Messungen für die verschiedenen Varianten des LLama 3 8B Instruct Modells

Inference Ergebnisse

Die Tabelle zeigt die Inference Ergebnisse ermittelt auf einer Nvidia A100 80GB GPU (CUDA 11.7)

Modell	Referenz	awq*	pruned	lora	qlora	Pruned + lora	Pruned + lora +awq*
Llama-3-8B-Instruct	27,35	2,40	34,42	15,99	29,39	36,08	3,03
Mistral-7B-Instruct-v0.2	35,03	2,47	41,35	33,86	31,60	38,50	3,01
Llama-2-7b-chat	31.42	1.40	38.22	17.01	28.41	40.03	1.84

Tabelle 7: Inference Performance von nativen zu bearbeiteten Modellen. Die mit einem * gekennzeichneten Modell wurde in 4 Bit Genauigkeit geladen, da die verwendete T4 GPU nicht genug Speicherkapazität für die native Größe besaß.

Die Tabelle zeigt die Inference Ergebnisse ermittelt auf einer Nvidia T4 15GB GPU (CUDA 12.x)

Modell Name	Verfahren	Token pro Sekunde
Llama 3 8B	Referenz	10,22
	prune + lora + awq	33,34
Mistral-7B-Instruct-v02	Referenz	11,10
	prune + lora + awq	38,99

Tabelle 8: Inference Performance von nativen zu AWQ quantisierten Modellen. Die Messungen wurden in einer Google Colab Umgebung mit einer Nvidia T4 15 GB durchgeführt. Hier konnten Cuda 12.x Treiber genutzt werden, die eine Verwendung des AutoAWQ python Moduls ermöglichten.

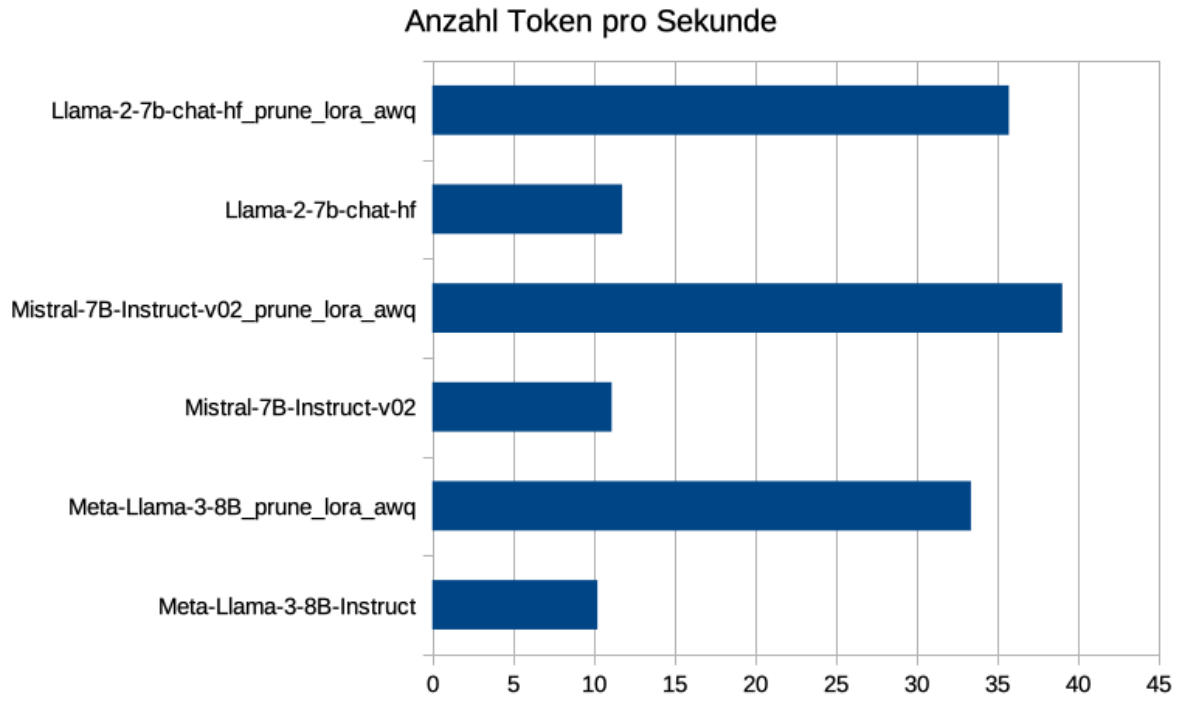

Abbildung 3: Token pro Sekunde ermittelt in einer Colab Umgebung mit CUDA 12.x und eine Nvidia T4 15GB GPU. Die Referenz Modelle wurden im 4bit mode geladen, da ansonsten der Speicher der GPU nicht ausgereicht hätte.

Systemresourcen

Speicherbedarf

Die Grafik zeigt die Speicherauslastung aller Varianten der drei Referenz Modelle

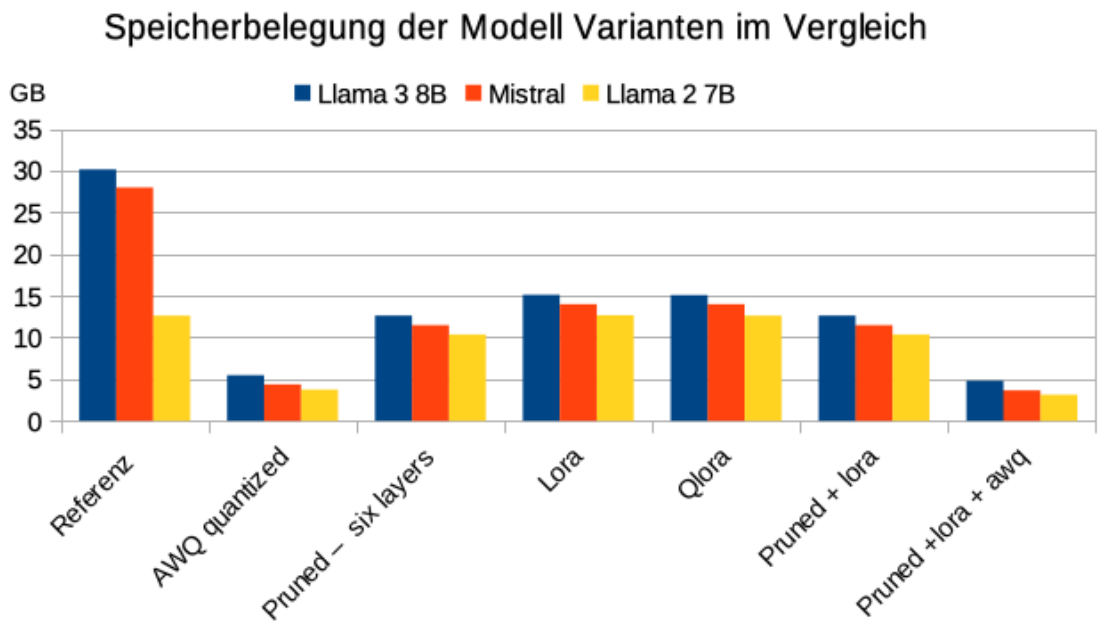

Abbildung 4: Speicherbedarf der Modelle auf einer Nvidia A100 80GB GPU. Aus dem jeweiligen Referenz Modell wurden alle weiteren Modelle abgeleitet. Die X-Achse zeigt die verwendeten Verfahren.

Abbildungsverzeichnis

Tabellenverzeichnis

Listingverzeichnis

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und dabei keine anderen als die angegebenen Hilfsmittel benutzt habe. Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Werken anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht. Die Arbeit wurde bisher weder gesamt noch in Teilen einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

31. Juli 2024

Dr. Thomas Schmitt