We hope to reduce LLM size without sacrificing much accuracy or speed of inference.

**Objective**

Large Language Models (LLMs) simplify life by retrieving information, serving as personal assistants, improving learning and accessibility, and increasing productivity. However, LLMs' enormous size and RAM utility makes storage and inference possible using only the most powerful computers. This fosters privacy, security, and judicial concerns about accessing LLMs via the internet.

We aim to reduce the size of LLMs with 7 billions parameters from 3GB to <2GB using combinations of Quantization, Pruning and Knowledge Distillation. We benchmark how these methods interact with our model on various processing units of our Orangefi-5 to demonstrate the viability of size-reduction of LLMs, with minimal accuracy drops.

**Approach**

- Orange pi 5 with Rockchip RK3588 new generation 8-core 64-bit processor, integrated ARM Mali-G610 GPU, built-in NPU with 6Tops computing power.
- Inference speed is determined by throughput in the form of seconds per token.
- Generally, the NPU is faster than the GPU which is faster than the CPU. This holds true because both NPU and GPU are designed for higher throughput.

**Hardware**

<table>
<thead>
<tr>
<th>Model in gpl format</th>
<th>Size (GB)</th>
<th>NPU avg s/token</th>
<th>GPU avg s/token</th>
<th>ARM avg s/token</th>
</tr>
</thead>
<tbody>
<tr>
<td>Llama-7B</td>
<td>7.36</td>
<td>0.948</td>
<td>1.068</td>
<td>2.336</td>
</tr>
<tr>
<td>llama-7b-4k</td>
<td>6.67</td>
<td>0.877</td>
<td>1.197</td>
<td>2.977</td>
</tr>
<tr>
<td>llama-7b-8k</td>
<td>7.36</td>
<td>0.948</td>
<td>1.068</td>
<td>2.336</td>
</tr>
<tr>
<td>ShortGPT-3.93b-4k</td>
<td>1.86</td>
<td>0.1098</td>
<td>0.197</td>
<td>0.582</td>
</tr>
<tr>
<td>ShortGPT-12.5b-4k</td>
<td>3.8</td>
<td>0.2213</td>
<td>0.677</td>
<td>2.854</td>
</tr>
<tr>
<td>llama-2-6B</td>
<td>2.4</td>
<td>0.3851</td>
<td>0.273</td>
<td>1.25</td>
</tr>
<tr>
<td>llama-2-3B</td>
<td>3.1</td>
<td>0.2833</td>
<td>0.399</td>
<td>1.527</td>
</tr>
<tr>
<td>llama-2-1B</td>
<td>3.1</td>
<td>0.2833</td>
<td>0.399</td>
<td>1.527</td>
</tr>
<tr>
<td>tinyLlama-3B</td>
<td>3.8</td>
<td>0.2213</td>
<td>0.677</td>
<td>2.854</td>
</tr>
<tr>
<td>TinyLlama-gpt-2.0M</td>
<td>0.69</td>
<td>0.04365</td>
<td>0.4</td>
<td>1.812</td>
</tr>
</tbody>
</table>

Table 1: The results of hardware inference per model on each processing unit.

Knowledge Distillation

Two distinct approaches for knowledge distillation: task-specific and task-agnostic. The former relies on labeled data to transfer task-specific knowledge, while the latter employs unsupervised learning for distillation, followed by model fine-tuning.

1. Task-specific

   Dataset: training or pre-training
   Open-assistant and dolly-15k
   Teacher: llama-7b
   Student: TinyLlama-1.1B-Chat-v0.4

2. Task-agnostic

   Dataset: babyLlama-10M, babyLlama-10M-dev
   Teacher: llama-2-360M, gpt-2-1.1B
   Student: babyLlama-58M

**Methodologies**

**Quantization & Pruning**

1. GPTQ & AWQ: GPTQ accelerates OBS by randomly selecting weights to quantize, avoiding the time-consuming greedy approach, while AWQ focuses on activation distribution to identify crucial weights for model performance.

2. ShortGPT (Dataset: Tulu-v2) & Stanford-Alpaca_Data, Open-Assistant: Directly removes layers based on dataset importance, reducing model size and restoring its capabilities with further LORA processing.

**Quantized Low Rank Adapters (QLoRA)**

1. QLoRA is a Parameter Efficient Fine-Tuning technique to fine-tune LLMs without utilizing much computational resources.

   - Aims to apply two-level quantization by applying 4-bit NormalFloat (NF4) quantization and then using LoRA to fine-tune the model.
   - After fine-tuning using LoRA, the adapter layers are then merged with the base model by adding the learned weights.

   - Beside is a overview of working of QLoRA module during quantization & pruning.

**Conclusion**

**General Trends:**

- Hardware Throughput: Models performed significantly better using NPU while CPU was the worst choice
- Accuracy: Most models have a higher BERT score followed by BLEURT and ROUGE

**Best Performing Models:**

- Hardware Throughput: Tiny Llama model (using Knowledge Distillation)
- Accuracy: GPTQ models (using Quantization)
- Perplexity: QLoRA models
- Size: Baby Llama (using Knowledge Distillation)

Each technique has its own merits and demerits and edge devices need to prioritize LLM’s use case before employing these methods. Smaller models generally had faster throughputs, usually with less accuracy.

**Future Work**

- Different use-cases math & coding, LLM inference on general hardware
- Further reduce inference latency
- General rescaling process for any other pre-trained model, such as Mistral, OpenELM, Llama3.

**References:**


**Students:** Chaitanya K Mullapudi, Delilah Yan, James Yao, Sai Jayanth Kalisi, Lucas Xie, Vincent Wang, Yao Zhang

**ADVISORS:** Mani Manohara, Mahdi Heydari, Sankalp Dayal, LarryArnstein, Amisha Somaiah

**SPONSOR:** Amazon