**CS795 Course Project Proposals**

Anton Rasmussen, Fall 2025

## TinyLLM: Evaluating Post-Training Quantization on a Small LLM

*Background*: Large Language Models (LLMs) are powerful but computationally expensive, making deployment on laptops or edge devices challenging. This project focuses on post-training quantization (PTQ), a popular approach for reducing model size and improving inference speed without retraining.

*Goals*: Apply, evaluate, and compare two leading PTQ methods (GPTQ, AWQ) on the same open-source model (e.g., Phi-3-mini or Mistral-7B), creating fast, CPU-runnable versions while quantifying accuracy and efficiency trade-offs.

*Data / Benchmarks*: WikiText-103 for perplexity; small subsets of MMLU, ARC-Easy, GSM8K; calibration set of ~1024 samples from C4/WikiText.

*Technical Components*: Model + dataset downloader, quantization pipeline (GPTQ & AWQ), evaluation harness (memory, latency, throughput, perplexity, accuracy), packaging models to GGUF for llama.cpp demo.

*Expected Output*: Comparison table of model size, perplexity, accuracy, latency, memory footprint; notebook + CLI demo for local inference.

*Stretch Goal*: Investigate SparseGPT pruning as a complementary compression step.

## 2) Edge-RAG: A Private, On-Device QA System for Documents

*Background*: Organizations in sensitive fields such as healthcare often need answers from private documents but cannot use cloud LLMs. This project builds a fully self-contained, offline RAG pipeline running entirely on a laptop.

*Goals*: Demonstrate an end-to-end private QA system with local retrieval, quantized reader model, and measurable accuracy, all offline.

*Data / Benchmarks*: Public PDFs (CMS fact sheets, benefit docs); 50–100 labeled QA pairs; metrics: recall@k, EM/F1, latency (<2s).

*Technical Components*: PDF ingestion + chunking to FAISS, retriever (MiniLM), quantized reader model (Phi-3-mini-4k-instruct-q4.gguf), evaluation harness (EM/F1), CLI demo.

*Stretch Goal*: Experiment with re-ranking or chunk-size tuning for improved recall.

## References

* Abdin et al. (2024). Phi-3 Technical Report. arXiv:2404.14219.
* Touvron et al. (2023). LLaMA-2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.
* Frantar et al. (2022). GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. arXiv:2210.17323.
* Lin et al. (2023). AWQ: Activation-Aware Weight Quantization for LLMs. arXiv:2306.00978.
* Merity et al. (2016). WikiText Language Modeling Dataset. Salesforce Research. Available at: [https://www.salesforce.com/blog/the-wikitext-long-term-dependency-language-modeling-dataset/](https://www.salesforce.com/blog/the-wikitext-long-term-dependency-language-modeling-dataset/?utm_source=chatgpt.com)
* Lewis et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401.
* Reimers & Gurevych (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084.
* Johnson, J., Douze, M., & Jégou, H. (2017). *Billion-scale similarity search with GPUs.* arXiv:1702.08734.