Presentation Script: Hyperparameter Optimization for LLMs

Group: NandaKiran, Anirudh, Phanindra, Revanth, Sri

Spring 2025

Speaker Scripts (Human-like, 2+ minutes each)

NandaKiran Velaga (Slides 1–3)

Slide 1 – Title: "Hi everyone. We're a group of graduate students from the University of Maryland, and today we're really excited to walk you through our project. It's called Hyperparameter Optimization for LLMs. In simple words, we tried to make large language models like T5 faster, smarter, and cheaper to train."

Slide 2 – Problem, Solution, Outcome: "We all know LLMs are powerful, but tuning them—finding the best settings—is often slow, expensive, and honestly, frustrating. We wanted a solution that's intelligent and light on resources. So we used two techniques: LoRA and Optuna. LoRA helps us fine-tune only small parts of the model, and Optuna is a tool that smartly searches for the best settings. With this, we improved the BLEU score by over 24% and saved about one-third of GPU memory."

Slide 3 – Vision: "Our main goal was to make LLM tuning more accessible. Not everyone has the resources of OpenAI or Google. But with LoRA and Optuna, even small teams or students can fine-tune models effectively. We wanted a process that works well, runs fast, and doesn't need massive hardware."

Anirudh Krishna (Slides 4–6)

Slide 4 – Optimization Flow: "Now let's talk about our approach. We used the CNN/DailyMail dataset. It contains long news articles and short summaries. First, we cleaned and tokenized the data. Then we fine-tuned the T5 model using LoRA. Instead of random experiments, we used Optuna to carefully test different settings. We used metrics like ROUGE and BLEU to see how well the model performed."

Slide 5 – PBT and Stability: "To make sure our results were reliable, we didn't just test each setup once—we used three different seeds. We also used something called Population-Based Training. It's like evolution. Bad configurations are removed, and good ones are copied with small changes. This helped us find strong and stable settings."

Slide 6 – LoRA Simplified: "Normally, fine-tuning updates millions of parameters. That takes time and memory. But with LoRA, we just add a few small layers and freeze the rest of the model. It's like editing only one part of a big book instead of rewriting the whole thing. This made training much faster."

Phanindra Tupakula (Slides 7–9)

Slide 7 – Optimization Math: "We also looked at this problem mathematically. We wanted to reduce the model's validation loss, but also limit training cost. So we created a cost model that included training time, memory, and energy. Then we minimized a combination of both performance and cost."

Slide 8 – Constraints + BO: "We used Bayesian Optimization to explore the search space. Instead of guessing random values, Optuna predicts which settings might work better. We also added constraints so that models don't use too much memory or time. It was a smart way to balance quality with resources."

Slide 9 – Lagrangian Relaxation: "Sometimes, constraints are too strict and block good results. So we used Lagrangian Relaxation. It lets us bend the rules a little but applies a penalty when we go too far. This gave us more flexibility to find better models."

Venkata Revanth Vardineni (Slides 10–12)

Slide 10 – Trade-offs: "Here we show a graph of performance versus resource usage. Every point represents a different setting. The best ones form a curve called the Pareto front. These give the best trade-offs—good accuracy without using too much time or memory."

Slide 11 – Search Space PBT: "We tested different values for learning rate, batch size, dropout, and LoRA parameters. We didn't want to rely on luck, so we used early stopping and averaged across three runs. PBT helped evolve better results by continuously improving the good settings."

Slide 12 – Hard Results: "Here are the real improvements. BLEU went from 21.1 to 26.3. ROUGE-1 and ROUGE-L improved too. We used less GPU memory and training time. In short, we made the model better, faster, and cheaper."

Sri Akash Kadali (Slides 13–18)

Slide 13 – Best Config: "This slide shows our best trial. It used a learning rate of 3.21e-4, LoRA rank 8, and dropout of 0.1. This gave us top accuracy with low resource usage."

Slide 14 – Trial Progress: "We ran 50 trials. In the beginning, results were okay. But with each new trial, Optuna learned from the past and got better. Our scores steadily improved. This shows the power of smart tuning."

Slide 15 – Wrap-up: "To wrap up: LoRA helped reduce training effort, and Optuna made the tuning smarter. In the future, we plan to use larger models like T5-base or T5-large, and also deploy the model as an API."

Slide 16 – Demo: "We also built a live demo. It shows how trials run and get better over time. It makes the whole process more visual and easier to understand."

Slide 17 – Summary Output Comparison: "We compared two summaries: one from the default model and one from the optimized model. The optimized version was clearer and more focused. That proves our method works."

Slide 18 – Final Thanks: "A big thank you to Professor Richard La and Amogha Sunil. Their feedback really helped. This project showed us that with the right tools, even small teams can do high-quality work."