[CSCI-GA 3033-091] Fall 2024

Special Topics: Introduction to Deep Learning and LLM based Generative AI Systems

Overview

This course serves as a graduate-level introduction to Deep Learning systems, with an emphasis on practical system performance issues and related research. The course will cover several topics related to Deep Learning (DL) systems and their performance. Both algorithmic and system related building blocks of DL systems will be covered including DL training algorithms, network architectures, and best practices for performance optimization. The latter half of the course will have an in-depth exploration of Large Language Models (LLMs), covering key areas towards advanced topics including attention mechanisms, transformer models, prompt engineering, LLM applications, pre-training strategies, Reinforcement Learning with Human Feedback (RLHF), efficient LLM serving techniques, fine-tuning methods, and benchmarking specifically for LLMs. The students will gain practical experience working on different stages of LLM life cycle, including model pretraining, fine tuning, and deployment. The assignments will be mostly hands-on involving standard DL and LLM frameworks (Pytorch, vLLM) and open-source technologies.

Target Audience

This course is aimed at MS-level students in computer science, data science, and other related disciplines.

General Information

- **Lecture:** Tuesdays 7:10pm-9:10pm (In-person)
    - Bldg: 31 Washington Pl (Silver Ctr)
    - Room: 414
    - Loc: Washington Square

- **Instructor**: Dr. Parijat Dube and Dr. Chen Wang

- **Grading:** Homework (40%) + Final Project (40%) + Quizzes (20%)

- **Homework**
    - Assignments will use Python and PyTorch
    - Assignments will involve running Deep Learning training jobs on GPU enabled public cloud platforms and use of open-source code/technologies.

- **Course project**
    - Team size is 2.
    - Final presentations of all projects towards the end of the course.

Prerequisites

An introductory graduate level machine learning course. Working knowledge of Python, Pytorch and experience using Jupyter Notebook.

Syllabus

Class 1: Fundamentals of Deep Learning (DL)
ML performance concepts/techniques: bias, variance, generalization, regularization; Performance metrics: algorithmic and system level; DL training: backpropagation, gradient descent; DL training hyperparameters, activation functions, exploding and vanishing gradients, weight initializers, learning rate, batch size, momentum, batch normalization; Regularization techniques in DL training: weight decay, dropout, early stopping, dataset augmentation.

Class 2: Distributed Training and Standard DL Architectures
Single node training, model and data parallelism, distributed training, parameter server, all reduce; Hardware support for training: GPUs, Tensor cores, NCCL; Introduction to standard DL architectures: CNNs, RNNs, LSTMs, GANs, Diffusion models.

Class 3: Cloud Technologies and ML Platforms
ML system stack on cloud; Docker, kubernetes; Cloud based ML platforms from AWS, Microsoft, Google, TorchX, Ray, and IBM; System stack, capabilities, and tools support on different platforms; Job scheduling on DL clusters.

Class 4: Operational Machine Learning
Devops principles in machine learning; DL deployment in production environment; Automated Machine Learning, H20 AutoML; MLOps, MLOps opensource platforms: Kubeflow, MLflow, MLOps opensource tools; Open Neural Network Exchange (ONNX).

Class 5: ML Monitoring and Benchmarking
Monitoring tools: TensorBoard, resource usage using nvidia-smi; Training-logs and their analysis; Time-series analysis of monitoring data; Drift detection and re-training; MLperf suite, MLPerf Training, MLPerf Inference, MLPerf Storage, Time-to-Train performance metric.

Class 6: Attention, Transformer, and Popular Large Language Models (LLMs)
Seq2Seq models: encoder and decoder; attention mechanism; Transformer architecture: self-attention, multi-head attention, encoder-decoder attention; LLMs: BERT, OpenAI GPT, LLAMA, Gemini, Claude, IBM Granite.

Class 7: Prompt Engineering and LLM App Development
Explore the diverse applications of Large Language Models (LLMs) and delve into the art of Prompt Engineering and LLM App Development. Students will learn about various use cases of LLMs, including translation, code generation, summarization, and entity extraction. Will cover fundamental prompt engineering concepts, such as prompt elements and tuning strategies, along with advanced techniques like zero-shot, few-shot, and chain-of-thought prompting.

Additionally, students will be introduced to LLM app development frameworks, including LangChain and LlamaIndex, enabling them to create sophisticated, context-aware applications that leverage the power of LLMs in real-world scenarios.

Class 8: LLM Applications

Explore advanced concepts in Large Language Models (LLMs), focusing on their capabilities and limitations. Delve into Retrieval-Augmented Generation (RAG) systems, including keyword search, embeddings, dense retrieval, and answer generation. The class will cover the development of LLM-powered agents capable of multi-step reasoning and API interactions. We'll examine vector databases and their crucial role in enhancing LLM operations. Additionally, students will learn about Graph RAG, which combines knowledge graphs with traditional RAG techniques for improved semantic search and content generation. Finally, the class will introduce Agentic RAG, an advanced approach incorporating autonomous agents for real-time planning and optimization in complex data environments.

Class 9: Pre-Training for LLM and Resource Requirements

Pre-training concepts of LLMs include training from existing foundation models and training from scratch. Model selection from HuggingFace and PyTorch hubs. Training process for different model architectures, including encoder-only, decoder-only, and seq-to-seq. Strategies for managing the high memory requirements of training LLMs, focusing on quantization and highlighting challenges of training on consumer-grade hardware. Scaling model training across multiple GPUs using techniques such as DPP, FSDP, Zero Redundancy Optimizer (ZeRO). Finding optimal balance between model size, training data volume, and compute budget for training LLMs; Chinchilla study's findings. Use cases of pretrain your own LLMs from scratch, focusing on domain adaptation in fields like law/medicine and introduction to BloombergGPT.

Class 10: Fine Tuning Techniques

This class will explore advanced fine-tuning techniques for Large Language Models (LLMs). Students will learn about Instruction Fine-Tuning, which enhances LLMs' ability to respond to specific prompts, and Parameter Efficient Fine-Tuning (PEFT), which allows for model adaptation with reduced computational resources. The course will cover the instruction fine-tuning process, including data preparation and evaluation. We'll delve into Multitask Fine-Tuning and its benefits for improved generalization. Special focus will be given to PEFT methods, particularly Low-Rank Adaptation (LoRA) and Prompt Tuning. Students will understand the principles, benefits, and practical applications of these techniques, including their impact on model performance, efficiency, and accessibility. The class will also introduce QLoRA and compare different PEFT approaches for various scenarios.

Class 11: Benchmarking for LLM

This class will explore the critical aspects of benchmarking Large Language Models (LLMs). Students will learn about the objectives and motivations behind LLM benchmarking, including both model and system evaluation. The course will cover essential evaluation metrics such as ROUGE, BLEU, and F1 scores, as well as advanced techniques using comprehensive benchmarks like GLUE. We'll discuss the importance of evaluating models on unseen data to ensure generalization and assess potential risks. Students will be introduced to existing benchmarking tools including LLMPerf, HuggingFace LLM-Perf Leaderboard, and Fmperf. The class will emphasize the significance of these benchmarking techniques in guiding LLM improvements and assessing their real-world applicability.

Class 12: Efficient Serving of LLMs
This class will focus on efficient serving techniques for Large Language Models (LLMs). Students will explore resource optimization strategies, including various batching techniques (static, dynamic, and continuous) and memory optimization methods like Flash Attention and Paged Attention Kernel. The course will cover popular LLM serving frameworks such as vLLM, Deepspeed-MII, TensorRT, and HuggingFace TGI Server. Special attention will be given to serving systems for fine-tuned models, including S-LoRA and LoRAX. Students will learn about performance trade-offs in serving, balancing individual request speed with overall throughput. The class will also delve into GPU sharing, optimization, and multiplexing techniques to maximize resource utilization and efficiency in LLM deployment.

Class 13: Reinforcement Learning with Human Feedback (RLHF)
This class will explore Reinforcement Learning with Human Feedback (RLHF), a method for aligning Large Language Models (LLMs) with human values and preferences. Students will learn about instruction fine-tuning and path methods to enhance LLMs' responsiveness to specific prompts. The course will address challenges in RLHF, including mitigating toxicity and misinformation. We'll cover the RLHF process, focusing on training reward models and updating LLMs based on human feedback. The class will introduce Proximal Policy Optimization (PPO) as a key algorithm in RLHF implementation. Additionally, students will explore the concept of Constitutional AI, understanding its role in developing ethically aligned AI systems. Throughout the course, emphasis will be placed on making LLMs more helpful, honest, and harmless through the integration of human judgments in the training process.

Class 14: Multimodal Generative AI systems (Optional)
This class will explore Multimodal Generative AI Systems, focusing on AI models that can process and generate multiple types of data, including text, images, audio, and video. Students will learn about the expansion from text-only Large Language Models (LLMs) to Large Multimodal Models (LMMs), understanding their architectures and training techniques such as cross-modal and contrastive learning. The course will cover key examples like DALL-E and GPT-4 with vision capabilities, demonstrating the integration of various modalities. We'll discuss the creation of LMMs, including encoder-decoder models and transformer-based architectures, with a focus on few-shot learning capabilities as demonstrated by models like Flamingo. Students will explore diverse use cases of LMMs, including visual question answering, image captioning, and cross-modal retrieval. The class will also address the challenges in aligning different modalities and the ethical considerations in developing and deploying multimodal AI systems.

Homework

There will be **five homework assignments**. All the assignments must be **submitted via Gradescope**. All programming assignments should be done using Jupyter notebook. Both the notebook and its pdf should be submitted.

Remarks

- A student in this course is expected to act professionally. Please also follow the GSAS regulations on academic integrity found here: http://gsas.nyu.edu/page/academic.integrity
- Academic accommodations are available for students with disabilities. Please contact the Moses Center for Students with Disabilities (212-998-4980 or mosescsd@nyu.edu) for further information. Students who are requesting academic accommodations are advised to reach out to the Moses Center as early as possible in the semester for assistance.