[COMSE6998-015] Fall 2024

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 LLM based Generative AI systems. 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, electrical engineering and other related disciplines.

General Information

- **Lecture details: TBA**
- **Instructor**: Dr. Parijat Dube and Dr. Chen Wang
- **Grading:** Homework (30%) + Final Project (30%) + Final Exam (30%) + Quizzes (10%)
- **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, activation functions, data preprocessing, batch normalization, exploding and vanishing gradients, weight initialization, learning rate policies; Stochastic and mini-batch gradient descent; DL training hyperparameters: batch size, learning rate, momentum, weight decay; Regularization techniques in DL Training: dropout, early stopping, data augmentation;

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

Class 3: Cloud Technologies and ML Platforms
ML system stack on cloud; Micro-services architecture: docker, kubernetes, kubeflow; Cloud based ML platforms from AWS, Microsoft, Google, TorchX, Ray, and IBM; System stack, capabilities, and tools support on different platforms.

Class 4: Operational Machine Learning
Training-logs and their analysis; Checkpointing; Scalability: learners, batch size, single node, distributed; Devops principles in machine learning, Model lifecycle management, MLOps; DL deployment and production cycle; Drift detection and re-training; Robustness and adversarial training; Automated Machine Learning; MLOps tool-chain.

Class 5: ML/DL Benchmarking and Performance Modeling
MLperf and Performance metrics for DL jobs; Runtime, cost, response time, accuracy, time to accuracy (TTA); Open Neural Network Exchange (ONNX); Monitoring (TensorBoard), performance, availability, and observability; Monitoring GPU resources (nvprof, nvidia-smi); Time series analysis of resource usage data; Predictive performance modeling techniques: black-box vs white-box modeling; Predictive performance models for DL convergence and runtime.

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 2, Gemini

Class 7: Prompt Engineering and LLM App Development
Prompt Engineering, Prompt Tuning, LLM app development framework, LangChain, Tool assisted LLMs

Class 8: LLM Applications
Use cases and applications of LLMs, including 1) the application of specialized chatbots in various customer service settings; 2) the application of LLMs in writing tasks; 3) the utilization of LLMs for various reading tasks such as proofreading, summarizing articles, analyzing customer service interactions, and routing emails, showcasing how LLMs can efficiently process and synthesize information to support decision-making in business applications.
Capabilities and limitations of LLMs. Specific limitations such as knowledge cutoffs, hallucinations, and struggles with structured data, alongside potential biases in outputs. Introdution to the Retrieval-Augmented Generation (RAG) systems, including keyword search, embeddings, dense retrieval, rerank, and answer generation. Hands on building LLM-powered agents capable of multi-step reasoning and interacting with the environment via functions and APIs.

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: Reinforcement Learning with Human Feedback (RLHF)
Introduction to RLHF. Introduce fine-tuning with instructions and path methods. Challenges like toxicity and misinformation. Aligning LLMs with human preferences for helpfulness, honesty and harmlessness. Process of RLHF; the process of training the reward model in RLHF; Replace human labelers by automatically selecting preferred completions based on human feedback. Use the reward model to update LLMs based on human feedback. Deep dive in using Proximal Policy Optimization (PPO) to fine-tune LLM through RLHF. Introdue the concept of Constitiutional AI.

Class 11: Efficient Serving of LLMs
Introduce available techniques to improve resource optimization and efficiency for serving LLMs, including the batching technique (static, dynamic, continuous batching) and various memory optimization techniques (Flash attention, Paged attention kernel, unified paging, etc.). Introduce popular LLM serving frameworks including vLLM, Deepspeed-MII, TensorRT, and HuggingFace Text Generation Inference (TGI) server.

Class 12: Fine Tuning Techniques / Systems
Introduce popular fine-tuning techniques, including full parameter fine tuning, parameter-efficient fine tuning (PEFT): adapters, LoRA (Low Rank Adaptation), QLoRA (Quantized Low Rank Adaptation).
Introduce existing serving system to serve fine-tuned models, including vLLM, S-LoRA, LoRAX, etc. Introduce the performance and tradeoff between performance, throughput and fairness in scheduling and routing requests. GPU sharing, optimization and multiplexing for efficient serving of LLMs.

Class 13: Benchmarking for LLM
Objectives, motivations, and types of LLM benchmarking. Difference between model benchmarking and system benchmarking. Key evaluation metrics for model benchmarking. Benchmarking in LLM training. Key evaluation metrics for LLM serving system benchmarking. Existing LLM benchmarking tools including LLMPerf and HuggingFace LLM-Perf Leaderboard, etc.

Class 14: Multimodal Generative AI systems (Optional)
Introduction to multimodal machine learning & AI. Multi-model generative AI goes beyond LLMs. How to incorporate additional modalities to LLMs and create LMMs (Large Multimodal Models). Introduction to the multimodality revolution and exploring LMMs' use-cases.