Course Description

This is a graduate level course with the focus of system perspectives in distributed deep learning. The goal of this course is to develop comprehensive and deep understanding of internals of deep learning systems to inspire and foster students' future professional and research direction.

Deep learning methods, especially large foundation models (LFMs), are enabling exciting new advances in many science and engineering disciplines, including genomics, bioinformatics, meteorology, and natural language processing (e.g., GenSLMs, AlphaFold, MetNet-3, GPT-3/4). Both industry and academia are heavily investing in extreme-scale clusters, including Microsoft, Meta, Google, and xAI. NSF just funded $457 million for the next-generation Leadership-Class Computing Facility and DOE has launched NERSC Perlmutter (>7,000 NVIDIA GPUs), OCLF Frontier (37,888 AMD GPUs), and ALCF Aurora (63,744 NVIDIA GPUs).

Training the next-generation workforce with knowledge of high-performance computing and artificial intelligence for students with diverse domain backgrounds, education levels, and from underrepresented groups is critical. This course covers a wide range of topics of neural network architecture, optimization methods, parallel training paradigms, high-performance computing architecture, and communication algorithms. This course conveys the principles of distributed/parallel system design with the state-of-the-art deep learning progress. The proposed course is project based. Students will work in teams and propose their own project ideas. Students will get a chance to run with 100s of A100 GPUs on NERSC Perlmutter (>7,000 NVIDIA A100s) and NCSA Delta (400 NVIDIA A100s) Supercomputers.

This course aligns with the mission and vision of Rutgers ECE department in that it will broaden the participation and education in electrical and computer engineering; it will prepare graduate students for the cutting-edge deep learning system field, with a solid foundation of distributed and parallel computing principles. Those principles will help them understand and adapt to new emerging areas in the future. This course is at the intersection of two ECE specialization areas: machine learning and computer engineering. It is designed for both MS and PhD students. This course conveys state-of-the-art techniques in distributed neural network training and inference and large language models (LLMs). It also offers assignments and hands-on exercises of LLMs on multiple GPUs. This course also covers the design principles of various machine learning systems with quantitative analysis methods to motivate and train PhD students for their future research;. The project-driven course design will let students find problems, design solutions, and verify the effectiveness, so that they are prepared to continually transform our society.

Prerequisite

- Prior Python programming experience is required
- Basic knowledge of linear algebra is required (at the level of 01:640:250 - Introductory Linear Algebra)
- Basic knowledge of calculus is required (at the level of 01:640:251 - Multivariable Calculus)
- Basic knowledge of deep learning is required (at the level of 16:32:530 - Introduction to Deep Learning)
- Basic knowledge of parallel/distributed programming is recommended (14:332:451 - Introduction to Parallel and Distributed Programming)
- Basic knowledge of high-performance computing architecture is recommended (14:332:451 - Introduction to Parallel and Distributed Programming)
- Prior CUDA knowledge is recommended (students can resort to online tutorials)

Textbooks

There is no text books in this course. All the material come from reading lists. The professor will provide course slides and hands-on exercise instructions.

The key notions of the papers in the reading list are incorporated into course slides. Students need to understand the contents for the assignments. For example, the memory management in optimizers from the ZeRO paper is deeply discussed in class, and the content is reflected in the third assignment, where the students need to train Llama 3.1 8B models on eight A100 40GB GPUs. They need to understand the memory and communication tradeoff before setting the correct ZeRO configuration for fully sharded distributed training.

Grading

- There are three assignments and one course project
    1. (5%) Assignment 1:
    2. (10%) Assignment 2:
    3. (10%) Assignment 3:
    4. (65%) Course project. The course project includes a proposal phase (Week 5), mid-term report phase (Week 10), and final presentation phase (Week 15)
    5. (5%) Class participation in the form of in-classroom discussion and experience sharing on software installation and test
- Late submission policy. Late submissions will be deducted 20% per 24 hours.

Week 1 -- Deep Learning Overview

- Recent Breakthroughs
- History
- From Linear Regression to Neural Networks
- Terminologies
- Neural Network Example

Reading List

1. [ResNet] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.
2. [Attention] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).
3. [BERT] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
4. [Mask R-CNN] He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." In Proceedings of the IEEE international conference on computer vision, pp. 2961-2969. 2017.
5. [3D U-Net] Çiçek, Özgün, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. "3D U-Net: learning dense volumetric segmentation from sparse annotation." In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19, pp. 424-432. Springer International Publishing, 2016.
6. [AlphaFold] Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool et al. "Highly accurate protein structure prediction with AlphaFold." Nature 596, no. 7873 (2021): 583-589.
7. [GPT] Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.

Week 2 -- Distributed Deep Learning Paradigms

- Data-Parallelism and Model-Parallelism
- Pipeline-parallelism
- Case Study with Megatron-LM and DeepSpeed

Reading List

1. [PyTorch] Li, Shen, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke et al. "PyTorch Distributed: Experiences on Accelerating Data Parallel Training." Proceedings of the VLDB Endowment 13, no. 12.
2. [PipeDream] Narayanan, Deepak, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R. Devanur, Gregory R. Ganger, Phillip B. Gibbons, and Matei Zaharia. "PipeDream: Generalized pipeline parallelism for DNN training." In Proceedings of the 27th ACM Symposium on Operating Systems Principles, pp. 1-15. 2019.
3. [Chimera] Li, Shigang, and Torsten Hoefler. "Chimera: efficiently training large-scale neural networks with bidirectional pipelines." In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1-14. 2021.
4. [Horovod] Sergeev, Alexander, and Mike Del Balso. "Horovod: fast and easy distributed deep learning in TensorFlow." arXiv preprint arXiv:1802.05799 (2018).

Week 3 -- Distributed Deep Learning Paradigms

- 3D-parallelism
- Sequence-parallelism
- Case Study with Ring Attention

Reading List

1. [DeepSpeed] Rasley, Jeff, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters." In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3505-3506. 2020.
2. [4D Parallelism] Li, Shenggui, Fuzhao Xue, Yongbin Li, and Yang You. "Sequence parallelism: Making 4d parallelism possible." arXiv preprint arXiv:2105.13120 (2021).
3. [Megatron-LM] Shoeybi, Mohammad, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. "Megatron-lm: Training multi-billion parameter language models using model parallelism." arXiv preprint arXiv:1909.08053 (2019).
4. [Alpa] Zheng, Lianmin, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang et al. "Alpa: Automating Inter-and {Intra-Operator} Parallelism for Distributed Deep Learning." In 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22), pp. 559-578. 2022.
5. [Ring-attention] Liu, Hao, Matei Zaharia, and Pieter Abbeel. "RingAttention with Blockwise Transformers for Near-Infinite Context." In The Twelfth International Conference on Learning Representations.
6. [Blockwise-attention] Liu, H. and Abbeel, P., 2024. Blockwise parallel transformers for large context models. Advances in Neural Information Processing Systems, 36.

Week 4 -- Large-scale Optimization in Deep Learning

- Optimization Method Overview
- First-order Methods, Computation Efficiency, and Scalability
- Memory-efficient Optimizers

Reading List

1. [Basic] Bottou, Léon, Frank E. Curtis, and Jorge Nocedal. "Optimization methods for large-scale machine learning." SIAM review 60, no. 2 (2018): 223-311.
2. [Adam] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
3. [LARS] You, Yang, Igor Gitman, and Boris Ginsburg. "Large batch training of convolutional networks." arXiv preprint arXiv:1708.03888 (2017).
4. [LAMB] You, Yang, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. "Large batch optimization for deep learning: Training bert in 76 minutes." arXiv preprint arXiv:1904.00962 (2019).
5. [ZeRo] Rajbhandari, Samyam, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. "Zero: Memory optimizations toward training trillion parameter models." In SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1-16. IEEE, 2020.

Week 5 -- Project Proposal

Week 6 -- Large-scale Optimization in Deep Learning

- Second-order Methods
- Scalability in Second-order Methods

Reading List

1. [K-FAC] Ba, Jimmy, Roger Grosse, and James Martens. "Distributed second-order optimization using Kronecker-factored approximations." In International Conference on Learning Representations. 2017.
2. [Shampoo] Gupta, Vineet, Tomer Koren, and Yoram Singer. "Shampoo: Preconditioned stochastic tensor optimization." In International Conference on Machine Learning, pp. 1842-1850. PMLR, 2018.
3. [KAISA] Pauloski, J. Gregory, Qi Huang, Lei Huang, Shivaram Venkataraman, Kyle Chard, Ian Foster, and Zhao Zhang. "Kaisa: an adaptive second-order optimizer framework for deep neural networks." In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1-14. 2021.

Week 7 -- Large Foundation Models

- Large Foundation Model Pretraining
- Fine-tuning
- Retrieval Augmented Generation
- In Context Learning

Readling List

1. [InstructGPT] Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang et al. "Training language models to follow instructions with human feedback." Advances in neural information processing systems 35 (2022): 27730-27744.
2. [RAG] Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in Neural Information Processing Systems 33 (2020): 9459-9474.
3. [In Context Learning] Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama et al. "Emergent Abilities of Large Language Models." Transactions on Machine Learning Research (2022).

Week 8 – Deep Learning Performance Profiling and Modeling

- Deep Learning Performance Profiling
- Discrete Event Simulation Approaches
- Analytical Approaches
- Performance Modeling with Complex Parallelism and Systems

Reading List

1. [PyTorch Profiler] https://pytorch.org/tutorials/recipes/recipes/profiler_recipe.html
2. [PyTorch with Tensorboard] https://pytorch.org/docs/stable/tensorboard.html
3. [DeLTA] Lym, Sangkug, Donghyuk Lee, Mike O'Connor, Niladrish Chatterjee, and Mattan Erez. "DeLTA: GPU performance model for deep learning applications with in-depth memory system traffic analysis." In 2019 IEEE international symposium on performance analysis of systems and software (ISPASS), pp. 293-303. IEEE, 2019.
4. [MDM] Wang, Lu, Magnus Jahre, Almutaz Adileho, and Lieven Eeckhout. "Mdm: The gpu memory divergence model." In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 1009-1021. IEEE, 2020.
5. [GCOM] Lee, Jounghoo, Yeonan Ha, Suhyun Lee, Jinyoung Woo, Jinho Lee, Hanhwi Jang, and Youngsok Kim. "GCoM: a detailed GPU core model for accurate analytical modeling of modern GPUs." In Proceedings of the 49th Annual International Symposium on Computer Architecture, pp. 424-436. 2022.

Week 9 – Reinforcement Learning

- Introduction to Reinforcement Learning
- Gym Implementation in Parallel
- Case Study with Gym and Ray

Reading List

1. [RL] Sutton, Richard S., and Andrew G. Barto. Introduction to reinforcement learning. Vol. 135. Cambridge: MIT press, 1998.
2. [Lunar Lander] https://towardsdatascience.com/breaking-down-richard-suttons-policy-gradient-9768602cb63b
3. [OpenAI Gym] Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "Openai gym." arXiv preprint arXiv:1606.01540 (2016).
4. [Ray] Moritz, Philipp, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol et al. "Ray: A distributed framework for emerging {AI} applications." In 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18), pp. 561-577. 2018.

Week 10 – Project Mid-term Report

Week 11 – High-performance Computing Architecture

- Processor Architecture
- Interconnect Architecture
- I/O System Architecture

Reading List

1. [CUDA] Section 1.1-1.3, 2. https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf
2. [Variability] Sinha, Prasoon, Akhil Guliani, Rutwik Jain, Brandon Tran, Matthew D. Sinclair, and Shivaram Venkataraman. "Not all GPUs are created equal: characterizing variability in large-scale, accelerator-rich systems." In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp. 1-15. 2022.
3. [Interconnet] Torsten Hoefler, "Network topologies for large-scale compute centers: It's the diameter, stupid!" https://spcl.inf.ethz.ch/Publications/.pdf/HotI16-Topologies-SlimFly.pdf
4. [FlashAttention] Dao, Tri, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. "Flashattention: Fast and memory-efficient exact attention with io-awareness." Advances in Neural Information Processing Systems 35 (2022): 16344-16359.
5. [ZOID] Iskra, Kamil, John W. Romein, Kazutomo Yoshii, and Pete Beckman. "ZOID: I/O-forwarding infrastructure for petascale architectures." In Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming, pp. 153-162. 2008.
6. [BurstFS] Wang, Teng, Kathryn Mohror, Adam Moody, Kento Sato, and Weikuan Yu. "An ephemeral burst-buffer file system for scientific applications." In SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 807-818. IEEE, 2016.

Week 12 – Communication in Deep Learning

- Collective Operations in Parallel Computing
- Allreduce Algorithms
- System Study of PyTorch and Horovod

Reading List

1. [MPI Collectives 1] Pješivac-Grbović, Jelena, Thara Angskun, George Bosilca, Graham E. Fagg, Edgar Gabriel, and Jack J. Dongarra. "Performance analysis of MPI collective operations." Cluster Computing 10 (2007): 127-143.
2. [Reduce Scatter] Patarasuk, Pitch, and Xin Yuan. "Bandwidth optimal all-reduce algorithms for clusters of workstations." Journal of Parallel and Distributed Computing 69, no. 2 (2009): 117-124.
3. [MPI Collectives 2]C han, Ernie, Marcel Heimlich, Avi Purkayastha, and Robert Van De Geijn. "Collective communication: theory, practice, and experience." Concurrency and Computation: Practice and Experience 19, no. 13 (2007): 1749-1783.
4. [Recursive Doubling] Thakur, Rajeev, Rolf Rabenseifner, and William Gropp. "Optimization of collective communication operations in MPICH." The International

Journal of High Performance Computing Applications 19, no. 1 (2005): 49-66.
5. [PyTorch] Li, Shen, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke et al. "PyTorch Distributed: Experiences on Accelerating Data Parallel Training." Proceedings of the VLDB Endowment 13, no. 12.
6. [Horovod] Sergeev, Alexander, and Mike Del Balso. "Horovod: fast and easy distributed deep learning in TensorFlow." arXiv preprint arXiv:1802.05799 (2018).

Week 13 – Sparsification in Deep Learning

- Gradient Sparsification
- Activation Sparsification

Reading List

1. [DGC] Lin, Yujun, Song Han, Huizi Mao, Yu Wang, and William J. Dally. "Deep gradient compression: Reducing the communication bandwidth for distributed training." arXiv preprint arXiv:1712.01887 (2017).
2. [Ok-TopK] Li, Shigang, and Torsten Hoefler. "Near-optimal sparse allreduce for distributed deep learning." In Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pp. 135-149. 2022.
3. [Activation Sparsity] Bian, Song, Dacheng Li, Hongyi Wang, Eric P. Xing, and Shivaram Venkataraman. "Does compressing activations help model parallel training?." arXiv preprint arXiv:2301.02654 (2023).
4. [Sparsification Survey] Hoefler, Torsten, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks." The Journal of Machine Learning Research 22, no. 1 (2021): 10882-11005.

Week 14 – I/O in Deep Learning

- File System Review
- I/O Trace Collection for ResNet, BERT, and GPT-2
- I/O Pattern Analysis

Reading List

1. [strace] https://man7.org/linux/man-pages/man1/strace.1.html
2. [Darshan] Snyder, Shane, Philip Carns, Kevin Harms, Robert Ross, Glenn K. Lockwood, and Nicholas J. Wright. "Modular hpc i/o characterization with darshan." In 2016 5th workshop on extreme-scale programming tools (ESPT), pp. 9-17. IEEE, 2016.

Week 15 – Final Project Presentation

Statement on Academic Integrity

As an academic community dedicated to the creation, dissemination, and application of knowledge, Rutgers University is committed to fostering an intellectual and ethical environment based on the principles of academic integrity. Academic integrity is essential to the success of the University's educational, research, and clinical missions, and violations of academic integrity constitute serious offenses against the entire academic community.

The principles of academic integrity require that a student:

- make sure that all work submitted in a course, academic research, or other activity is the student's own and created without the aid of impermissible technologies, materials, or collaborations.
- properly acknowledge and cite all use of the ideas, results, images, or words of others. properly acknowledge all contributors to a given piece of work.
- obtain all data or results by ethical means and report them accurately without suppressing any results inconsistent with the student's interpretation or conclusions.
- treat all other students ethically, respecting their integrity and right to pursue their educational goals without interference. This principle requires that a student neither facilitate academic dishonesty by others nor obstruct their academic progress.
- uphold the ethical standards and professional code of conduct in the field for which the student is preparing.

Adherence to these principles is necessary to ensure that:

- proper credit for ideas, words, images, results, and other scholarly work, no matter the form or media, is attributed to the appropriate individual(s).
- all student research and work are fairly evaluated, and no student has an inappropriate advantage over others.
- the academic and ethical development of all students is fostered.
- the reputation of the University for integrity, ethics, scholarship, and professionalism is maintained and enhanced.

Failure to uphold these principles of academic integrity threatens both the reputation of the University and the value of the degrees awarded to its students. Every member of the University community, therefore, bears a responsibility for ensuring that the highest standards of academic integrity are upheld.

To uphold these principles, the University administration is responsible for:

- working with faculty, staff, and students to foster a strong institutional culture of academic integrity,
- providing effective educational programs that create an understanding of and commitment to academic integrity, and
- establishing equitable and effective procedures to deal with allegations of violations of academic integrity.