Master of Information and Data Science (MIDS)
DATASCI W251 Deep Learning in the Cloud and at the Edge

Course Description

This hands-on course introduces students to technologies related to building and operating live, end-to-end deep learning systems that include components running in the cloud (model training, data annotation, and ETL) as well on lower-power devices at the edge of the network (data collection and inference).

To participate, each student will need to purchase an edge GPU-equipped device along with several additional peripherals. The course currently uses the NVIDIA Jetson NX developer kit, a low-power system on a chip (SOC) edge GPU-equipped device as the required edge device. Prior to the start of the class, students will receive an NVIDIA discount on the NX developer kit. In addition, each student will receive a $1,000 Amazon Web Services (AWS) credit to cover their hands-on model training and data preparation cloud activities.

As students progress through the class, they learn to apply the concepts covered by completing hands-on homework and labs on the Jetson NX as well as in the cloud, culminating in the final project.

The class material consists of a set of practical approaches, code recipes, and lessons learned. It is based on the latest developments in the industry and industry use cases as opposed to pure theory. It is taught by professionals with decades of industry experience.

While this class is an advanced elective and we generally assume familiarity with the basics of machine learning, operating systems, big data, and infrastructure, we fill in the blanks when it comes to some practical tooling as well as the latest developments in deep learning.

Every asynchronous segment is followed by a hands-on homework assignment as well as a synchronous hands-on lab, where students get to explore the concepts and technologies covered in the lecture. By the time you complete the course, you should be able to recognize the application problem you are facing, select one or more proper neural architectures for it, along with the appropriate tooling, and put together an end-to-end pipeline around it.

We begin the class with some real-life examples of how deep learning is used, an overview of the state-of-the-art advances across industries, and a review of prominent datasets. We will review the NVIDIA Jetson NX developer kit, unbox it, and get it to work. There's no deep learning without cloud today, so we will talk about the cloud concepts and dive into the cloud capabilities next. We will also learn the basics of the environments that will host our labs, home assignments, and the final project. Virtualization and containerization is the fabric that binds the applications together today, so we will dive into these concepts in detail next. The fourth lecture covers fundamentals of deep learning, such as artificial neurons, activation functions, loss functions, backpropagation, regularization, and dropout. Next on the agenda is an in-depth overview of leading deep learning frameworks (such as PyTorch, TensorFlow, and Hugging Face Transformers) that provide runtimes for DL applications. We then go over data processing acceleration and model optimization methods for the edge devices. A second lecture on Deep Learning fundamentals follows, where we look at fundamental concepts such as learning rate scheduling, hyperparameter optimization, and some of the common neural architectures, such as CNNs, RNNs, and transformers. We move on to the datasets as well as tooling required to prepare datasets for deep learning, such as speech recognition, image classification, and object detection, as well as covering techniques for data augmentation. We next study distributed model training, starting with the basics of high-performance computing, Message Passing Interface, and InfiniBand, and learn how to train models on clusters of hardware. Generative adversarial networks are next, where we learn to apply deep learning to generate photo-realistic images. A lecture on deep reinforcement learning follows. We dive into applying deep learning to natural language processing (NLP), Automatic Speech Recognition (ASR), text to speech (TTS), and conversational agents next. We conclude the class by reviewing several industry-specific DL use cases such as deep recommenders and deepfakes along with associated end-to-end tooling for large volume data processing, such as Dask, NVIDIA RAPIDS, NVTabular, and Triton Inference Server (3 units).

Course Goals and Objectives

After completing this course, students will:

- Gain a basic understanding of key deep learning fundamental theoretical concepts
- Achieve hands-on experience with key deep learning tooling and frameworks
- Learn about the modern/major deep learning datasets, such as ImageNet, MS COCO, and LibriSpeech
- Be able to code in at least one DL framework, such as PyTorch
- Develop an understanding of designing and implementing end-to-end deep learning applications on the edge and in the cloud
- Develop fluency with the NVIDIA Jetson edge platform
- Understand some of the deep learning industry use cases
- Understand the infrastructure required to power deep learning applications, such as on-premise HPC, big data, and cloud environments
- Apply deep learning to large-scale big data problems

Key Dates

Academic Calendar: https://www.ischool.berkeley.edu/intranet/students/mics/calendar

You may prefer to leave the syllabus "evergreen" without dates and link to a separate calendar each semester with weekly schedule and due dates

Course Evaluation

- Graded homework assignments (3, 5, 9, 11): 100-point scale, 40% of total grade (evenly weighted).
- Credit/No-Credit homework assignments (1, 2, 4, 6, 7, 8, 10, 12) 10% of total grade (evenly weighted).
- Participation: 100-point scale, 10% of total grade
- Final Project: 100-point scale, 40% of total grade. Students will organize into groups of four to five to perform an analysis on a large dataset and prepare a final presentation (slides) and a live demo or video (10 min).

Assignment details and grading rubrics, including opportunities for extra credit, as applicable, are available in the course GitHub.

Grading Information

The course will be graded on an absolute scale, and the grades will not be fitted to a specific curve. This is a graduate-level course, and we trust that different students will have varying levels of interests in the different subjects in the course. As such, the grading scheme is designed to acknowledge this intellectual diversity.

Late Submission Policy

Solutions of homeworks and labs will be discussed during the live sessions of the course. Therefore, any assignment that is submitted after the deadline will be returned without grading and will receive a grade of zero.

Weekly Schedule Outline/Assignments/Readings

Unit 1: Introduction and Overview

Big Data. Cloud Computing. Storage. Artificial Intelligence. Machine Learning. Deep Learning Frameworks and Hardware. Datasets. Edge Computing.

Reading

- See GitHub.

Assignments:

- Homework 1 assigned.

Unit 2: Clouds, Infrastructure, and Machine Learning Cloud Services
Introduction to Cloud Computing and Cloud AI. Defining the Cloud. How Clouds Are Used. Hypervisors in a Nutshell. Types of Clouds. Cloud Services. Cloud Storage. AI as a Service. Deep Learning as a Service.

Reading
- See GitHub.

Assignments:
- Homework 2 assigned.
- Homework 1 due prior to the beginning of the live session.

Unit 3: Introduction to Containers
Review Virtual Machines, What Enables Them, and Their Benefits. Introduction to Containers. What Challenges Do Containers Address? Software Containers and Shipping Containers. How Containers Work. What Enables Containers. The Container Ecosystem. Introduction to Docker. Containers and GPUs on the NVIDIA Jetson Family. Introduction to Orchestration and Kubernetes.

Reading:
- See GitHub

Assignments:
- Homework 3 assigned.
- Homework 2 due prior to the beginning of the live session.

Unit 4: Deep Learning 101
The Definition of Deep Learning. How Is It Different From AI and ML? Artificial Neurons. Neural Layers. Feed Forward Networks. Multilayer Perceptron. Backpropagation. Normalization. Regularization. Convolutional Layers. Attention. Introduction to RNNs.

Reading:
- See GitHub

Assignments:
- Homework 4 assigned.
- Homework 3 due prior to the beginning of the live session.

Unit 5: Deep Learning Frameworks
Key Components of a Deep Learning Framework. Tensors. Data Loaders. Optimizers. The Training Loop. The Validation Loop. Eager Mode vs. Graph Mode. Inference Runtimes. TensorFlow 1.0, TensorFlow 2.0, PyTorch, PyTorch Lightning, and the Hugging Face NLP Frameworks.

Reading:
- See GitHub.

Assignments:

- Homework 5 assigned.
- Homework 4 due prior to the beginning of the live session.

Unit 6: Optimizing Models for the Edge and GStreamer
Models at the Edge. Model Optimization. Pruning. Clustering. Numerical Precision. Quantization. FP16. Int8. TensorFlow Lite. PyTorch. TensorRT. Introduction to GStreamer. Pipelines. Properties. Capabilities. NVIDIA and GStreamer on the Jetson Family.

GPUs, created to accelerate computer graphics, have revolutionized deep learning. This section looks at why deep learning at scale requires special hardware, like GPUs. We will cover different chip architectures commonly used to increase performance of different frameworks.

Reading:

- See GitHub

Assignments:

- Homework 6 assigned.
- Homework 5 due prior to the beginning of the live session.

Unit 7: Deep Learning 201
Weight Initialization. Optimizers. Adjusting the Learning Rate. Loss Functions. Transfer Learning. Autoencoders. Embeddings. Inference vs. Training vs. Model Design. Recurrent Neural Networks.

Reading:

- See GitHub.

Assignments:

- Homework 7 assigned.
- Homework 6 due prior to the beginning of the live session.

Unit 8: Datasets and Dataset Processing
Types of Datasets. Key Public Datasets. Cloud Platforms for Hosting and Processing of Large Datasets. ETL/ELT Overview. Datasets and Data Loaders of Major Deep Learning Frameworks. Data Annotation Systems. Active Learning. Targeted Learning. Amazon SageMaker Ground Truth. Data Augmentation Techniques. Dataset Collection From Model Runtimes.

Reading:

- See GitHub

Assignments:

- Homework 8 assigned.
- Homework 7 due prior to the beginning of the live session.

Unit 9: HPC, MPI, and Multi-node/MultiGPU (MNMG) Training
History of HPC. HPC vs. HTC/Big Data. Typical HPC Problems. Architecture of Supercomputers. Interconnect Topologies: Fat Tree, Torus. FLOPs. Top500. Amdahl's Law. Programming for HPC Systems. MPI. HPC Schedulers. InfiniBand vs. TCP/IP. Google TPUs and TPU Pods. NVIDIA DGX Systems and SuperPODs. Magnum IO. Distributed Deep Learning Model Training. Uber Horovod. Distributed Training in TensorFlow and PyTorch. Distributed Training in AWS and Azure. Experiment tracking.

Reading:

- See GitHub.

Assignments:

- Homework 9 assigned.
- Homework 8 due prior to the beginning of the live session.

Unit 10: Generative Adversarial Networks (GANs)
Introduction to Generative Adversarial Networks (GANs). Learn about Generators and Discriminators. Uses of GANs Such as Deepfakes, Image Generation and Editing, Lip Reading, Speech Synthesizing, Generating Data, and Security. GAN Creation (Datasets, Creating a Generator, and Creating a Discriminator).

Reading:

- See GitHub.

Assignments:

- Homework 10 assigned.
- Homework 9 due prior to the beginning of the live session.

Unit 11: Deep Reinforcement Learning
Introduction to Deep Reinforcement Learning (DRL). Overview of Q-Learning. DRL Platforms. Implementing DRL. Review Common Uses of DRL Such as Robotics, Gaming, Traffic Management, and Autonomous Automobiles.

Reading:

- See GitHub.

Assignments:

- Homework 11 assigned.
- Homework 10 due prior to the beginning of the live session.

Unit 12: Speech, Natural Language Processing, and Conversational Design
Introduction to Natural Language Processing (NLP). Types of NLP Problems: Q&A, NER. Types of Tokenizers: BPE, WordPiece. Modern Neural Architectures for NLP: Transformers, BERT, RoBERTa, XLNet. Automatic Speech Recognition (ASR). Key Neural Architectures for Speech Recognition. Text to Speech (TTS): Problem Definition and Key Neural Architectures. Data

Annotation Systems for NLP and ASR. Conversational Systems. NVIDIA NeMo and Jarvis Frameworks. Amazon Alexa and Google Assistant. Skills Development.

Reading:
- See GitHub.

Assignments:
- Homework 12 assigned.
- Homework 11 due prior to the beginning of the live session.

Unit 13: AI and DL: Applying AI to Real World Applications
Deep Recommenders. NVIDIA Merlin. NVTabular. Extract Load Transform (ETL) for Tabular Data. GPU-Accelerated RAPIDS Data Transformation Libraries: Cupy, Cudf, Cuml, Cugraph, DMLC XGBoost. Distributed Processing With Dask. HugeCTR: Optimized Model Training Framework for Recommenders. Triton Inference Server: Unified Runtime for Machine Learning Models. Deepfakes: Recognizing Fake vs. Real Videos.

Reading:
- See GitHub.

Assignments:
- Homework 12 due prior to the beginning of the live session.

Unit 14: Final Projects
There is no asynchronous coursework in this unit. Students should spend the time to complete their final projects to present in the live session.

Assignments:
- Final Projects due during the live session.

Collaboration Policy/Academic Integrity

All students must be familiar with and abide by the provisions of the “Student Code of Conduct,” including those provisions relating to Academic Misconduct. All forms of academic misconduct, including cheating, fabrication, plagiarism, or facilitating academic dishonesty, will not be tolerated. The full text of the UC Berkeley Honor Code is available at https://teaching.berkeley.edu/berkeley-honor-code, and the Student Code of Conduct is available at https://sa.berkeley.edu/student-code-of-conduct#102.01_Academic_Misconduct.

We encourage studying in groups of two to four people. This applies to working on homework, discussing labs and projects, and studying for the exam. However, students must always adhere to the UC Berkeley Code of Conduct (http://sa.berkeley.edu/code-of-conduct) and the UC Berkeley Honor Code (https://teaching.berkeley.edu/berkeley-honor-code). In particular, **all materials that are turned in for credit or evaluation must be written solely by the submitting student or group**. Similarly, you may consult books, publications, or online resources to help you study. In the end, **you must always credit and acknowledge all consulted sources in your submission (including other persons, books, resources, etc.)**

Attendance and Participation

A note on participation in live sessions: We believe in the importance of the social aspects of learning—between students, and between students and instructors—and we recognize that knowledge building is not solely occurring on an individual level, but that it is built by social activity involving people and by members engaged in the activity. Participation, teamwork, and communication are key aspects of this course that are vital to the learning experiences of you and your classmates.

To maximize opportunities for live session engagement, we like to remind all students of the following requirements:

- Students are required to join live class sessions from a study environment with video turned on and with a headset for clear audio, without background movement or background noise, and with an internet connection suitable for video streaming.
- Students are expected to engage in class discussions and breakout room discussions and exercises and to be present and attentive for their and other teams' in-class presentations.
- Students should keep the microphone on mute when not talking to avoid background noise. Students should do their best to minimize distractions in the background video and ensure that their camera is on while they are engaged in discussions.

That said, in exceptional circumstances, if students are unable to meet in a space with no background movement and with a good internet connection, they should arrange with the instructors (beforehand if possible) to explain the situation. Sometimes connections and circumstances make turning off video the best option. If this is a recurring issue in students' study environment, they are responsible for finding a different environment that will allow full participation in classes, without distraction to classmates.

Failure to adhere to these requirements will result in an initial warning from the instructor(s), followed by a possible reduction in grades or a failing grade in the course.

Diversity and Inclusion

Integrating a diverse set of experiences is important for a more comprehensive understanding of cybersecurity. We will make an effort to read papers and hear from a diverse group of practitioners. Still, limits exist on this diversity in the field of cybersecurity. I acknowledge that it is possible that there may be both overt and covert biases in the material due to the lens with which it was created. I would like to nurture a learning environment that supports a diversity of thoughts, perspectives and experiences, and honors your identities (including race, gender, class, sexuality, religion, ability, veteran status, etc.) in the spirit of the UC Berkeley Principles of Community (https://diversity.berkeley.edu/principles-community).

To help accomplish this, please contact your instructors or submit anonymous feedback through I School channels if you have any suggestions to improve the quality of the course. If something was said in class (by anyone) or you experience anything that makes you feel uncomfortable, please talk to your instructors about it. If you feel like your performance in the class is being impacted by experiences outside of class, please don't hesitate to talk with your instructors or with Student Affairs. We want to be a resource for you. Also, anonymous feedback is always an option, and may lead to instructors making a general announcement to the class, if necessary, to address your concerns. As a participant in teamwork and course discussions, you should also strive to honor the diversity of your classmates.

If you prefer to speak with someone outside of the course, the MIDS Equity and Inclusion Advisor Michael Rivera, the I School Assistant Dean of Academic Programs Catherine Cronquist Browning, and the UC Berkeley Office for Graduate Diversity are excellent resources. Also see the following: https://www.ischool.berkeley.edu/about/community.

Disability Services and Accommodations

The I School recognizes disability in the context of diversity, and the Disabled Students' Program (DSP) equips students with appropriate accommodations and services to remove barriers to educational access. Students seeking accommodations in this class are responsible for completing the DSP application process to obtain an accommodation letter. (510) 642-0518, https://dsp.berkeley.edu.

Publishing Your Work

You are highly encouraged to use your program coursework to build an academic/professional portfolio.

- Blog about your coursework (and other ideas) and share on the I School Medium.
    - Instructions are here on the intranet for students: https://www.ischool.berkeley.edu/intranet/connect
    - and here public for alumni: https://www.ischool.berkeley.edu/alumni/stay-connected.
- Publish projects to your I School project portfolio gallery (more than just for capstone).
- Publish your work on LinkedIn and tag the @UC Berkeley School of Information.
- Publish your coursework in a GitHub repo (MIDS sample skeleton).
- Publish in academic journals—contact your professors for assistance. (Note that multiple review iterations are usually required; this can be a time-intensive endeavor.)

    - For help writing professional academic papers, students are encouraged to contact Sabrina Soracco, the director of the Graduate Writing Center, in the Graduate Division; see https://grad.berkeley.edu/staff/sabrina-soracco/ and
    - https://grad.berkeley.edu/professional-development/graduate-writing-center/—for links to resource guides, appointments with consultants, workshops, etc.
- Publish your news (e.g., conference talks, award, scholarships) to the I School internal newsletter.