DS 542 Deep Learning for Data Science

Fall 2024 Syllabus

Official Course Description

In this course students will gain an understanding of the fundamentals in deep learning and then apply those concepts in exercises and applications in python. We'll start with the origins of artificial neural networks, learn about loss functions, understand gradient descent, back propagation and various training optimization techniques. Students will be familiar with canonical network architecture such as multi-layer perceptions, convolutional neural networks, recursive neural networks, LSTMs and GRU, attention and transformers. Through explanations, examples and exercises students will build intuition on how deep learning algorithms work and how they are implemented in popular deep learning frameworks such as PyTorch. Students will be able to define, train and evaluate deep learning models as well as adapt deep learning frameworks to new functionality. Students will gain exposure to pre-trained large language models and other foundation models and the concepts of few-shot learning and reasoning. Finally, students will be able to apply many of the techniques they learned in a final class project.

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Grasp foundational theories and practices in the deep learning arena.
2. Design and implement various neural network architectures using Python and PyTorch.
3. Employ regularization, optimization, and advanced training techniques to enhance model performance.
4. Analyze real-world datasets, applying suitable deep learning techniques to derive actionable insights.
5. Understand the benefits and drawbacks of various neural architectures in specific contexts.
6. Understand the pros and cons of pre-trained large language and other foundation models and how best to employ them
7. Complete a data-centric project, showcasing end-to-end deep learning implementation.

Prerequisites

- Python Programming – Should be proficient in python and associated data science packages, or studiously working towards proficiency. See for example Scientific Python Lectures for lessons on python language and relevant packages, or The Python Tutorial for a tutorial on the core language.

- Packages such as NumPy NumPy - Learn and SciPy (SciPy User Guide have tutorials and documentation as well.
- The more proficient you are, the more effective you will be at the assignments and projects. We will dedicate some discussion sessions to ensure your environment is set up correctly and review some of the basics as well as answer any questions.
- Math Proficiency – In order to understand the foundational concepts, it is important to have proficiency in a number of areas of mathematics. These include linear algebra, first year calculus and trigonometry as well as some concepts from Real Analysis. We will cover these concepts in the class and some recitation sessions, but refreshing or building your foundation will help.

Weekly Cadence

- Lectures: Mondays and Wednesdays, 12:20pm – 2:05pm, EPC 209
- Discussions: Mondays, 3:35pm – 4:25pm, CAS 313
- Instructor: Jeffrey Considine
    - Office: CDS 1625
    - Office Hours: Tuesdays 11:00am – 12:00pm, Wednesday 3:00pm – 4:00pm, Thursday 1:00pm – 2:00pm
    - email: jconsidi@bu.edu
- Teaching Assistant: Xavier Thomas
    - Office Hours: Fridays 3:00pm – 5:00pm
    - Office Hours Location: CDS, 16th Floor
        - email: xthomas@bu.edu

Textbooks and Reference Materials

- The primary textbook for this course will be Understanding Deep Learning, by Simon Prince . The book is available online..
- We'll also reference Introduction to Linear Algebra, Sixth Edition (2023), by Gilbert Strang.
- Lecture notes, consisting primarily of Jupyter notebooks will be posted online as well.
- Given the fast moving nature of this area, we'll also be citing many articles available online as well as other online reference materials in each lecture. As part of the class, we will guide the students in constructing their own bibliography and give tips on how to efficiently and effectively read research papers.

Course Requirements

- Homework: For the 1st half of the class we'll assign Jupyter Notebook coding assignments and a few homework questions approximately every week to help anchor

the key concepts and python/pytorch coding patterns. These will become less frequent in the 2nd half of the course to allow you to concentrate and make progress on your projects.
- A mid-term deep learning training contest
- A final project where you will apply deep learning methods to a problem or application of interest to you.

Course Assessment

- Final Project: 50%
- Mid-term Project/Competition: 20%
- Homeworks: 30% (includes notebooks and problem sets)

Schedule

This schedule is subject to change, particularly towards the end of the course to allow coverage of new research and final project presentations.

Date	**Lecture Topic**	**Assessments**
Wednesday, September 4	Intro to Deep Learning and Course Logistics	Notebook 1 released
Monday, September 9	Supervised Learning	Problem Set 2 released
Wednesday, September 11	Shallow Neural Networks	Notebook 1 due, Problem Set 3 released
Monday, September 16	Deep Neural Networks	Problem Set 2 due, Notebook 4 released
Wednesday, September 18	Loss Functions	Problem Set 3 due, Problem Set 5 released
Monday, September 23	Fitting Models	Notebook 4 due, Notebook 6 released
Wednesday, September 25	Gradients and Initialization	Problem Set 5 due, Notebook 7 released
Monday, September 30	Measuring Performance	Notebook 6 due, Notebook 8 released
Wednesday, October 2	Regularization	Notebook 7 due, Notebook 9 released
Monday, October 7	Convolutional Networks	Notebook 8 due,

		Notebook 10 released
Wednesday, October 9	Residual Networks	Notebook 9 due
Tuesday, October 15	Midterm review	Notebook 10 due
Wednesday, October 16	Midterm starts (bring laptop)	Midterm due Friday, October 18
Monday, October 21	Recurrent Neural Networks	Final Project proposal released
Wednesday, October 23	Transformers	
Monday, October 28	Final Project Introductions (students present for class feedback)	Final Project proposal draft due
Wednesday, October 30	Graph Neural Networks	Final Project proposal draft feedback released
Monday, November 4	Unsupervised Learning	Final Project proposal final due
Wednesday, November 6	Generative Adversarial Networks	Final Project proposal final feedback released
Monday, November 11	Final Project previews	Final Project preview presentation in class
Wednesday, November 13	Normalizing Flows	
Monday, November 18	Variational Autoencoders	Final Project check in
Wednesday, November 20	Diffusion Models	
Monday, November 25	Reinforcement Learning	Final Project check in
Monday, December 2	Why Does Deep Learning Work?	
Wednesday, December 4	Class Choice / Recent Developments	Final Project due Friday, December 6
Monday, December 9, 2024	Final Project presentations	Final Project presentation in class

Student Code of Conduct

All students are expected to abide by University conduct policies as detailed in the following links:

- Boston University Student Codes of Conduct
- College of Arts & Sciences Codes of Conduct

- Boston University Student Responsibilities

Academic Honesty

You may discuss homework assignments with classmates, but you are solely responsible for what you turn in. Collaboration in the form of discussion is allowed, but all forms of cheating (copying parts of a classmate's assignment, plagiarism from books or old posted solutions) are NOT allowed. We – both teaching staff and students – are expected to abide by the guidelines and rules of the Academic Code of Conduct.

Graduate students must also be aware of and abide by the GRS Academic Conduct code.

You can probably, if you try hard enough, find solutions for homework problems online. Given the nature of the Internet, this is inevitable. Let me make a couple of comments about that:

1. If you are looking online for an answer because you don't know how to start thinking about a problem, talk to the TA or instructor, who may be able to give you pointers to get you started. Piazza is great for this – you can usually get an answer in an hour if not a few minutes.
2. If you are looking online for an answer because you want to see if your solution is correct, ask yourself if there is some way to verify the solution yourself. Usually, there is. You will understand what you have done much better if you do that. So … it would be better to simply submit what you have at the deadline (without going online to cheat) and plan to allocate more time for homeworks in the future.

Generative AI Assistance (GAIA) Policy

In general, we follow the policy outlined in the CDS GAIA Policy.

Extracting and paraphrasing from the student responsibilities of that policy. Where there is conflicting information between the CDS policy and below, the policy below should take precedence.

Students shall:

1. Give credit to AI tools whenever used, even if only to generate ideas rather than usable text, illustrations or code.
2. When using AI tools on written assignments, unless prohibited, add an appendix showing
    1. the entire exchange, highlighting the most relevant sections;
    2. a description of precisely which AI tools were used (e.g. ChatGPT private subscription version or DALL-E free version),
    3. an explanation of how the AI tools were used (e.g. to generate ideas, turns of phrase, elements of text, long stretches of text, lines of argument, pieces of evidence, maps of conceptual territory, illustrations of key concepts, etc.);
    4. an account of why AI tools were used (e.g. to save time, to surmount writer's block, to stimulate thinking, to handle mounting stress, to clarify prose, to translate text, to experiment for fun, etc.).
    5. Optional but recommended: Employ AI detection tools and originality checks prior to submission, ensuring that their submitted work is not mistakenly flagged.

3. When using AI tools on coding assignments, unless prohibited
    1. Add the prompt text and tool used as comments before the generated code. Clarify whether the code was used as is, or modified somewhat, moderately or significantly.
4. Not use AI tools during in-class examinations, or assignments, unless explicitly permitted and instructed.
5. Use AI tools wisely and intelligently, aiming to deepen understanding of subject matter and to support learning.

As these generative assistive tools become widely deployed and pervasive, we believe they will become integral to most people’s workflow. However, for foundational concepts, as are taught in this course, it is in your best interest and worth it to struggle some in creating your answers and solutions. It is just as important to learn what doesn’t work, and which paths are dead ends, as it is to learn what does work. When you are posed with new and unique problems, that intuition you develop will be vital in choosing directions. More pragmatically, some of the most coveted jobs at the most selective companies require technical interviews where they expect you to know these foundational concepts without assistance.

And finally, to reiterate, it is vitally important, and a core part of academic integrity, to cite when you are using Generative AI Assistive technologies. Arguably, not citing and risking plagiarism is worse than taking short cuts and using and then citing GAIA.

Accommodations for Students with Disabilities

If you have a disability and have an accommodations letter from the Disability & Access Services office, I encourage you to discuss your accommodations and needs with me as early in the semester as possible. I will work with you to ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with BU Disability & Access Services, I encourage you to find more information at https://www.bu.edu/disability/.
This syllabus provides a general plan for the course; deviations may be necessary depending on the progress of the class.

Acknowledgements

This syllabus and the general course design is based on the first offering of this course by Tom Gardos.