MARYLAND APPLIED GRADUATE ENGINEERING

Maryland Applied Graduate Engineering, University of Maryland at College Park

Fundamentals of AI and Deep Learning

ENPM 703

Professor: George Zaki

Email: gzaki@umd.edu

Office hours: Monday and Tuesday at 7:30pm, or email to schedule

Term: Summer 2026

Credits: 3

Course Dates: From June 1st – August 2st

Course Times and Locations:

Sections 0101 and CY01 : Wednesday 6:00 - 9:15 PM, JMP 2121 and Online

Google credit:

The credits will help students complete the course project. This class introduces fundamentals of deep learning. Students will have extensive hands-on experience with state-of-the-art deep learning frameworks to build and evaluate these networks.

Course Description

The application of data science in visual and text understanding, consumer products, health care, and banking, has been boosted by major advances in three primary areas: (1) Data: diversity, amount, and availability data; (2) Advances in Artificial Intelligence (AI) and Machine Learning (ML) algorithms that enable learning from complex, large-scale data; and (3) Advances in computer architectures allowing unprecedented acceleration of machine learning algorithms. Artificial neural networks aka Deep Learning is a specific type of machine learning algorithms that have gained much attention in the last decade due to their ability to learn and improve with increasing amounts of data. Advances in neural network techniques help build in silico ML models that can match and sometimes exceed human performance.

This class will introduce fundamentals of machine learning techniques and deep dive in cutting edge concepts that enabled neural networks to achieve state of the art performance in many visual, textual, and biomedical problems. Fundamental concepts like feed forward networks, convolution networks, recurrent neural networks, back propagation, loss functions, batch gradient descent, and stochastic optimization will be studied. Students will have extensive hands-on experience with state-of-the-art deep learning frameworks like Keras/TensorFlow/PyTorch to build, evaluate, use, and debug these networks for real life applications.

Course Objectives

By successfully completing this course, the students should be able to:

- Describe the fundamental concepts in machine learning, and the reasons behind the rise of neural networks to scale with today's big datasets.
- Formulate machine learning problems and identify suitable neural networks models to solve them.
- Use modern neural networks frameworks (e.g., Keras, TensorFlow, PyTorch) to train, validate, test, and debug state of the art models.
- Address challenges, identify solutions, and explore opportunities in using neural networks in various application domains.

Course Outline

Here is a tentative list of topics covered in the list. Detailed information about course modules, assignments and exams will be posted on ELMS.

- Introduction to machine learning
  Introduction to artificial intelligence and machine learning, supervised/unsupervised learning, train/validation/test datasets, evaluation metric, exploratory data analysis, overview of classical machine learning techniques
- Artificial neural networks
  Feed forward networks, multiple layers perceptron, back propagation, gradient descent, learning rate, batch size, neural network for classification and regression
- Neural network architectures/layers
  convolutional neural networks, batch normalization, recurrent neural networks, attention, transformers, autoencoders, dimensionality reduction, generative adversarial networks
- Application in Security and biomedical application (time permitting)
  Image classification, semantics/instance segmentation for medical images and pathology whole slide images, classification of unstructured texts, neural network models for next generation sequencing data, drug data, synthesis of high-resolution images, sharing valuable datasets using pretraining.
- Algorithm Fairness, Accountability, Transparency, and Ethics
- Student term project

Detailed information about course modules, assignment and exams will be posted on ELMS.

Prerequisites:

- Linear algebra
- Fundamentals of programming
- Fundamentals of statistics

Textbooks and other Required Reading Materials

- Hands-on Machine Learning with Scikit-Learn & Keras: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurelien Geron
- Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Grade Breakdown

- 5% programming assignments
- 25% quizzes
- 25% midterm
- 10% paper presentation
- 35% term project

Details about every assignment, exam, and the project will be shared on ELMS. The final grade of the course will use the grade breakdown shown above. There are 11 assignments contributing equally to 5%, two quizzes (12.5% each), one midterm (25%), and one paper presentation. The breakdown of the term project will be shared on Canvas.

Please note the total grade that will be automatically calculated on ELMS does not represent your weighted grade. At the end of the course, I will share with you the weighted grade based on the formula above. The weighted grade will be mapped to the letter grade.

The final mapping of the weighted grade to the letter grade will be made available based on the distribution of the weighted grade. Please note that mapping will be final to be fair to all students. If there is a cutoff at 80%, a grade of 79.99% would not be approximated to 80%.

Grading:

All submitted assignments will be marked as seen if the final report meets the requirements, quiz midterm and final will be graded by the co-instructor and instructor. The projects will be graded by the co-instructor and instructor. Given the class size, you should expect the grades to be available within 10 days of submitting your assignment. If you have any questions regarding your grade, please contact the co-instructor. If your issue is not resolved, please contact me. For every assignment, the grade rubric will be described on ELMS. You have a week to review and dispute any grade.

Course Project: Apply Deep Learning to Real-World Problems

The purpose of the course project is to develop skills in defining, implementing, and executing original machine learning tasks by applying deep learning technologies to a real-world problem in a field of interest. This project focuses on practical applications in areas such as biology, physics, healthcare, robotics, security, and signal processing. Students will be asked to define a specific problem within their chosen field, design and implement a deep learning model to address it, and conduct a comprehensive analysis of the model's performance. The project involves submitting a proposal, an interim milestone report, a final written report, and a presentation. More information, including detailed guidelines, resources and deadlines, will be posted on Canvas.

Assignments:

The course includes three main assignments designed to provide practical experience in various aspects of machine learning. Assignment 1 focuses on building an image classification model using k-Nearest Neighbor (kNN) and Support Vector Machine (SVM) classifiers. Assignment 2 involves training neural networks and convolutional neural networks (CNNs), where students will implement backpropagation and various optimization techniques. Assignment 3 centers on language networks and image captioning, where students will implement RNN and Transformer networks, and optionally explore GANs and self-supervised learning techniques. Each assignment requires coding and commenting on implementations, with detailed instructions and starter code available on Canvas. Every assignment is divided into multiple parts. The details and due dates for every part will be available in Canvas.

Usage of Large Language Models:

Assignments:

This is a fundamentals course, and the purpose of the assignments is for you to gain hands-on experience in designing and implementing the core building blocks of deep learning from scratch. To support this goal:

- You may not copy or paste code or text generated by large language models (LLMs, free or paid) into your assignment solutions or reports.
- Writing the code yourself is essential for building a deep understanding of how neural networks work and for developing the ability to debug, tune, and extend models independently.
- Limited use of LLMs for practical conveniences (e.g., autocompletion in your IDE, or looking up help on error messages) is acceptable, but relying on them to generate solutions defeats the purpose of the course.
- Your work should reflect your own implementation and understanding.

Examples of acceptable vs. unacceptable use for assignments

- Acceptable usage of LLM to:
    - Clarify what an error message means
    - Look up syntax reminders (e.g., how to reshape a NumPy array)
    - Generate test data to check your own implementation
    - Help with grammar or clarity in your written report

- Unacceptable usage of LLMs to:
  - Generate functions, classes, or entire blocks of code for your assignment
  - Write explanations, derivations, or analysis sections of your report
  - Debug by pasting in your entire assignment and asking for fixes

Final Project

For the final project, you may use external resources more flexibly. This includes code or text generated by LLMs or taken from repositories such as GitHub. However:

- You must clearly attribute any code, text, or ideas you use that are not your own.
- You should be able to explain in your own words how the borrowed components work and why you chose to include them.

Why this matters

The ultimate goal of this policy is to ensure you develop both the conceptual understanding and the practical skill to implement and adapt algorithms yourself. These are foundational skills that will give you the confidence and flexibility to use advanced tools responsibly later on.

Late policy:

All students are given 48 hours of grace period that they can use as they wish during the semester. After this period is exhausted, the late policy below will take effect. For example, if you were late 24 hours for each of the first two assignments, and 13 hours for the third assignment, there is no penalty for the first two assignments, and you will be deducted 10% of the grade of the third assignment.

- Less than 1 hour late: No penalty (one hour grace period)
- 1-24 hours late: 20% penalty
- 24-48 hours late: 40% penalty
- …etc. (20% for every additional 24 hours late)

Academic Integrity

From The Code of Academic Integrity:

Academic dishonesty is a serious offense which may result in suspension or expulsion from the University. In addition to any other action taken, such as suspension or expulsion, the grade XF denoting "failure due to academic dishonesty" will normally be recorded on the transcripts of students found responsible for acts of academic dishonesty.

Unless otherwise stated, all quizzes, exams, programming assignments and any other assignments are individual assignments: collaboration is not permitted unless explicitly stated on the assignment handout. Students may discuss among themselves concepts pertaining to the programming assignments. However, at no point should any code, pseudocode, or anything that resembles code be exchanged or copied from a website, a peer, an AI generated website.

Students should write the pledge of honor on all submitted assignments, projects, and exams**: "I pledge on my honor that I have not given or received any unauthorized assistance on this exam/assignment."**

Course Structure

This course includes both on-campus and online sections.

For online students, all lectures will be recorded and made available on ELMS-Canvas under "Panopto Recordings/Video Lectures" within 24 hours of the class time. Be sure to review the recorded lecture in a timely manner.

If online students wish to attend synchronously online, you can do so by logging into ELMS-Canvas at the time of the Section 0101 class (Thursdays 7:00 - 9:40 PM) and selecting "Video Conference" from the left side menu. This will open a Zoom link to the live classroom.

On-campus students are expected to attend in-person class sessions and be prepared to engage with the lecture and materials. If you have a conflict on a particular day, please reach out to me in advance to discuss. Online students, be sure to log into Canvas regularly and participate in discussions and activities. Regardless of the section you are enrolled in, participation is expected.

Please note that F1 students enrolled in the on-campus section are required to attend in person.

Communicating with the Instructor

If you have any questions about the course, please reach out to me at gzaki@umd.edu. Make sure you check the syllabus and EMLS if you have questions about deadlines or policies. I will respond to you no later than 48 hours. If you do not receive a response after 48 hours, please email me again.

When constructing an email to me please put "ENPM 809K: Your Topic" in the subject line. This will draw my attention to your email and enable me to respond to you more quickly.

Announcements

I will send important messages, announcements, and updates through ELMS-Canvas. To ensure you receive this information in a timely fashion, make sure your email and announcement notifications (including changes in assignments and/or due dates) are enabled in ELMS-Canvas (How to change notification settings in CANVAS).

Log into our ELMs-Canvas course site at least once every 24-hour period to check your inbox and the announcement page.

Students with Disabilities

If you have a documented disability and wish to discuss academic accommodation with me, please contact me as soon as possible and no later than the end of the second week.

University Resources, General Guidelines, Policies and Procedures:

Please check on course website the document called university-resources.pdf

Looking forward to a mutually enjoyable semester!!