AMS 490: Intermediate Topics in Applied Mathematics - Applied Mathematics for ML and AI

Course Code: AMS 490 - Section 0001
Credits: 4 (letter grade)

Class Time: Mondays and Wednesdays, 2:00 PM – 3:50 PM
Location: Chafee Hall, Room 208
Instructor: Pengyu (Peter) Liu
Email: pengyu.liu@uri.edu
Office Hours: TBA

Prerequisites: MTH 141 or MTH 131; MTH 215 or permission of instructor; and CSC 110 or CSC 201 or permission of instructor.

Course Description

This course develops the mathematical foundations underlying modern machine learning and artificial intelligence, with a strong emphasis on computation and implementation. Students build on prior knowledge from single-variable calculus, linear algebra, and programming to explore key mathematical tools used in contemporary machine learning systems.

Topics include selected material from multivariate calculus, vector calculus, optimization theory, differential equations, and advanced linear algebra. Core mathematical concepts are studied in the context of machine learning applications, including activation functions, loss functions, gradient-based optimization methods, and iterative learning algorithms. Students examine how mathematical structures such as gradients, Jacobians, and matrix decompositions arise naturally in machine learning models.

A central feature of the course is the integration of mathematical reasoning with computational practice. Programming assignments require students to implement mathematical concepts using modern tensor-based libraries and numerical tools. These assignments emphasize translating mathematical formulas into efficient computational procedures and interpreting algorithmic outputs.

The class meets twice a week for 110-minute sessions that combine lectures on mathematical theory, examples, and guided problem solving with hands-on computational lab work. During lab time, students collaborate on implementing algorithms, experimenting with models, and solving computational problems in a contemporary programming language.

Throughout the course, emphasis is placed on building mathematical intuition, understanding algorithmic behavior, and developing practical computational solutions to machine learning problems.

Topics Covered

- **Mathematical Foundations Review:** Review of prerequisite topics from linear algebra and calculus, including vector spaces, linear systems, norms, orthogonality, projections, and introductory logic and set concepts.
- **Multivariable Calculus and Vector Calculus:** Partial derivatives, gradients, Jacobians, and the multivariable chain rule; vector-valued functions; interpretation of derivatives as sensitivity; Taylor series and multivariate approximations; conceptual introduction to double and triple integrals.
- **Linear Algebra for Machine Learning:** Orthogonality, projections, least squares methods, symmetric and positive definite matrices, quadratic forms, and matrix decompositions including eigenvalue decomposition, Cholesky decomposition, and singular value decomposition.
- **Tensor Methods and Computational Techniques:** Introduction to tensors and vectorized computation; implementation of tensor operations and mathematical procedures using Python and modern numerical libraries.
- **Optimization Theory and Algorithms:** Formulation of optimization problems; critical points; constrained and unconstrained optimization; Lagrange multipliers; convex optimization; gradient descent and its variants; Newton’s and quasi-Newton methods; iterative methods for solving linear systems.
- **Machine Learning Functions and Models:** Analysis of activation functions (e.g., sigmoid, ReLU, softmax) and loss functions (e.g., mean squared error, likelihood-based losses,

cross-entropy), including differentiation and optimization behavior.
- **Applications and Computational Projects:** Application of mathematical methods to real-world machine learning problems through computational assignments and a final project.

Required Materials

Required textbook: Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong (Cambridge University Press, 2020). A PDF of this text is provided by the authors as a free download. Any supplemental resources will be uploaded on Brightspace or put on library reserve.

Technology requirements: Students will need a laptop or computer with reliable, high-speed internet access that supports the Brightspace learning platform: Linux (Ubuntu 16 or comparable), macOS (10.10 or higher), or Windows (10 Home or higher); 8 GB RAM; at least 2 GB of free disk space; and a web browser (Chrome, Safari, or Firefox; Internet Explorer is not recommended). The course is delivered through Brightspace, Panopto, Zoom, and Google Drive.

Software used in this course: Assignments and projects are written in Python 3 and completed in Jupyter notebooks, with GitHub (an Internet hosting service for software development and version control using Git) used for code. Students should already be familiar with Python; if you feel out of practice, please ask, and see the standard Python 3 Tutorial. Weekly lab assignments and the final project are given as Jupyter notebooks (.ipynb), which can be run in any of the following ways:

- **Google Colab**: run notebooks in the browser and store them in Google Drive, with no local Python installation (intro video; Welcome to Colaboratory).
- **Platform help and accessibility**: Brightspace (login, ITS resources, tutorials, accessibility); Panopto (login, training, accessibility); Google Drive (login, help, accessibility).

Learning Outcomes

By the end of this course, students will be able to:

1. **Compute and apply multivariable calculus concepts.** Compute partial derivatives, gradients, and Jacobians for multivariable functions, and apply the chain rule to analyze activation functions, loss functions, and neural-network computations.
2. **Apply linear algebra methods to machine learning problems.** Apply linear algebra techniques, including orthogonality, least squares, matrix factorizations, and positive definiteness, to regression, quadratic optimization, and related machine learning models.
3. **Formulate and analyze optimization problems.** Formulate real-world problems as mathematical optimization problems, classify common optimization types, and analyze gradient-based optimization methods with respect to convergence, convexity, and critical points.
4. **Implement computational solutions using modern numerical tools.** Implement and modify numerical algorithms in Python to perform tensor operations, vectorized computations, gradient-based methods, and large-scale optimization procedures.
5. **Evaluate computational models and their practical implications.** Evaluate mathematical models and algorithms applied to real-world data, including assessing performance, interpreting results, and recognizing ethical considerations associated with data-driven decision making.

Assignments, Exams, and Projects

Weekly Laboratory Assignments. Students will complete computational assignments during the lab portion of class meetings. These assignments will focus on implementing mathematical concepts using an appropriate contemporary programming language and numerical libraries. Laboratory work will emphasize translating mathematical procedures into computational algorithms, interpreting outputs, and debugging code. Some assignments may extend beyond the scheduled lab time.

Homework Assignments. Homework assignments will be assigned approximately every two weeks and will focus on mathematical

problem-solving related to recently covered topics. These assignments will reinforce theoretical understanding, develop analytical skills, and provide practice with symbolic and numerical methods relevant to machine learning applications.

Midterm Examinations. Two midterm examinations will be administered during the semester. These exams will assess students' understanding of core mathematical concepts, computational techniques, and problem-solving skills. Exam questions may include analytical computations, conceptual explanations, and applied problems related to optimization and machine learning mathematics.

Final Project. In lieu of a final examination, students will complete a final project involving the mathematical modeling and computational solution of one or more real-world problems. Students will be encouraged to work in small groups. The project will typically include problem formulation, implementation of computational methods, analysis of results, and written or oral presentation of findings. Project guidelines will be distributed approximately midway through the semester.

Evaluation

Final grades will be based on the following components:

- Weekly laboratory assignments: 30%
- Homework (biweekly assignments): 20%
- Midterm exams: 30%
    - 2 in-class exams
- Final project (group-based): 20%
    - Proposal, written report, and presentation

Grading scale (based on the overall weighted percentage; cutoffs are rounded to the nearest whole number): A: 90.0% – 100%, A-: 87.0% – 89.9%, B+: 85.0% – 86.9%, B: 83.0% – 84.9%, B-: 80.0% – 82.9%, C+: 77.0% – 79.9%, C: 73.0% – 76.9%, C-: 70.0% – 72.9%, D+: 67.0% – 69.9%, D: 60.0% – 66.9%, F: Below 60%

Workload Expectations

This is a 4-credit course. In accordance with federal credit hour guidelines, students should expect to spend approximately 3.7 hours per week in class (two 110-minute meetings) and a minimum of 8 hours per week on out-of-class work, for a total of approximately 12 hours per week. The approximate weekly breakdown is as follows:

- In-class lectures, worksheets, and lab work: 3.7 hours
- Reading the textbook and reviewing lecture material: 1.5–2 hours
- Homework assignments: 2–2.5 hours
- Completing lab assignments outside of class: 2–2.5 hours
- Exam preparation and final project work: 1.5–2 hours (heavier before midterms and during the final project period)

Policies and Statements

University's AI Safety and Ethics Policy: The University of Rhode Island (URI) embraces artificial intelligence (AI) as a powerful tool for teaching, learning, research, and innovation. However, we recognize the importance of ensuring AI is used securely, ethically, and responsibly to reflect our values of transparency, fairness, and accountability. Please visit https://its.uri.edu/ai-at-uri/ for understanding URI's standards and practices for responsible AI use.

Course-Specific AI Use Policy: Because this course focuses on the mathematics of machine learning and artificial intelligence, students are encouraged to use AI tools (such as ChatGPT, Claude, GitHub Copilot, etc.) as learning aids for understanding concepts, debugging code, and exploring ideas. However, students must demonstrate their own understanding of the material. For all submitted work, students are required to (1) disclose any AI tools used and (2) include the key prompts or a brief description of how AI was used in an appendix to their submission. Copying AI-generated content without understanding or attribution is considered a violation of the Academic Honesty Policy.

Academic Honesty Policy: Cheating is defined in the University Manual section 8.27.10 as the claiming of credit for work not done independently without giving credit for aid received, or any unauthorized communication during examinations. Students are expected

to be honest in all academic work. The resolution of any charge of cheating or plagiarism will follow the guideline set forth in the University Manual sections 8.27.10-8.27.21. Furthermore, course content and outlines, exams, and assignments created by instructors shall be considered the instructors' intellectual property. Course materials shall not be distributed, shared in any public domain or third party website, or sold without prior written consent of the instructor. See the University Manual section 8.27.2

Academic Accommodations: Your access in this course is important. Please send your instructor your Disabilities, Access, and Inclusion (DAI) accommodation letter early in the semester so that we have adequate time to discuss and arrange your approved academic accommodations. If you have not yet established services through DAI, please contact them to engage in a confidential conversation about the process for requesting reasonable accommodations in the classroom. DAI can be reached by calling: 401-874-2098, visiting: https://www.uri.edu/disability, or emailing: dai@etal.uri.edu

Anti-bias Statement: We respect the rights and dignity of each individual and group. We reject prejudice and intolerance, and we work to understand differences. We believe that equity and inclusion are critical components for campus community members to thrive. If you are a target or a witness of a bias incident, you are encouraged to submit a report to the URI Bias Response Team at www.uri.edu/brt. There you will also find people and resources to help.

Health and Safety: The University is committed to delivering its educational mission while protecting the health and safety of our community. While the university has worked to create a healthy learning environment for all, it is up to all of us to ensure our campus stays that way. As members of the URI community, students are required to comply with standards of conduct and take precautions to keep themselves and others safe. If you are unable to attend class, please notify your instructor prior to the start of class through the medium they have established for the class.

Religious Holidays: It is the policy of the University of Rhode Island to accord students, on an individual basis, the opportunity to observe their traditional religious holidays. Students who plan to be absent from classes or examinations for religious holy days that

traditionally preclude secular activity shall discuss this with the appropriate instructor(s) in advance of the holy day. See the University Manual section 8.51.11 for details.

Standards of Behavior: Students are expected to treat faculty and fellow classmates with dignity and respect. Students are responsible for being familiar with and adhering to the published "Community Standards of Behavior: University Policies and Regulations" which can be accessed in the University Student Handbook https://web.uri.edu/studentconduct/university-student-handbook.

Getting Help in This Course: Post general questions in the "Course Questions, Concerns, and Feedback" discussion forum on Brightspace, where classmates can also respond. For questions specific to you regarding this course or its assignments, use the message feature in Brightspace to contact the instructor for personal feedback. Questions regarding technology should be directed to the URI IT Help Desk at 401-874-4357 or www.uri.edu/helpdesk, which also contains Brightspace help and instructions for students.

University Statements: Please visit the following webpage or scan the QR code to read more detailed University Statements and Policies. https://web.uri.edu/atl/syllabus-statements/

Course Schedule (Tentative)

Spring 2027; topics and dates are subject to change. Class meets Mondays and Wednesdays, 2:00 PM – 3:50 PM, in Chafee Hall, Room 208. Lab activities take place during these class meetings.

Due dates: Unless noted otherwise, homework is due Mondays at 11:59 PM and lab assignments are due Thursdays at 11:59 PM, as listed below, submitted through Brightspace. Daily worksheets are completed and

turned in during class. Lab 9 is due Monday 5/3 because the following Thursday falls in the final exam period.

Week	Events	Topics
Week 1 Jan 25–29	Classes begin Tue 1/26	**Wed 1/27:** Review, notation, language and foundations Lab 0: Python, JupyterHub and GitHub setup (not graded)
Week 2 Feb 1–5		**Mon 2/1:** Vectors, norms and distances **Wed 2/3:** Partial derivatives and the gradient vector Lab 1: K-nearest neighbors
Week 3 Feb 8–12	HW 1 due Mon 2/8 Lab 1 due Thu 2/11	**Mon 2/8:** Geometric meaning of the gradient; directional derivatives **Wed 2/10:** The multivariable chain rule Lab 2: Gradient descent
Week 4 Feb 15–19	HW 2 due Mon 2/15 Lab 2 due Thu 2/18	**Mon 2/15:** The chain rule in machine learning; computational graphs **Wed 2/17:** Matrix primer; second-order derivatives and the Hessian Lab 3: Single-neuron neural network
Week 5 Feb 22–26	HW 3 due Mon 2/22 Lab 3 due Thu 2/25	**Mon 2/22:** Taylor approximation and critical points **Wed 2/24:** Vector-valued functions and the Jacobian matrix Midterm 1 review and problem session
Week 6 Mar 1–5	HW 4 due Mon 3/1 **Midterm Exam 1** Wed 3/3, in class	**Mon 3/1:** The chain rule for Jacobians; ML building blocks **Wed 3/3: Midterm Exam 1 (Part I: multivariable calculus)** Lab 4: Three-neuron neural network

Week	Events	Topics
Week 7 Mar 8–12	HW 5 due Mon 3/8 Lab 4 due Thu 3/11	**Mon 3/8:** Vectors, operations and matrix algebra **Wed 3/10:** Trace, block matrices and systems of linear equations Lab 5: Introduction to PyTorch and tensors
Week 8 Mar 15–19	HW 6 due Mon 3/15 Lab 5 due Thu 3/18 **Final Project Guidelines Released** Wed 3/17	**Mon 3/15:** Vector spaces, subspaces and rank **Wed 3/17:** Linear transformations, inverse and pseudo-inverse Lab 6: Data augmentation and image transformations
Break Mar 22–28		**Spring Break (no classes)**
Week 9 Mar 29–Apr 2	HW 7 due Mon 3/29 Lab 6 due Thu 4/1	**Mon 3/29:** Determinants and eigenvalues **Wed 3/31:** Diagonalization, spectral theorem and quadratic forms Lab 7, Part 1: PCA foundations and SVD computation
Week 10 Apr 5–9	HW 8 due Mon 4/5 Lab 7 Pt 1 due Thu 4/8	**Mon 4/5:** The singular value decomposition **Wed 4/7:** Tensors, broadcasting and practical manipulation Midterm 2 review and problem session
Week 11 Apr 12–16	HW 9 due Mon 4/12 **Midterm Exam 2** Mon 4/12, in class **Project Proposal Due** Thu 4/15	**Mon 4/12: Midterm Exam 2 (Part II: linear algebra)** **Wed 4/14:** Finite differences and forward-mode automatic differentiation Lab 7, Part 2: Face reconstruction and eigenfaces

Week	Events	Topics
Week 12 Apr 19–23	Lab 7 Pt 2 due Thu 4/22	**Mon 4/19:** Reverse-mode automatic differentiation and backpropagation **Wed 4/21:** Convexity and gradient descent Lab 8: PyTorch autograd and computational graphs
Week 13 Apr 26–30	HW 10 due Mon 4/26 Lab 8 due Thu 4/29	**Mon 4/26:** Convergence analysis, stochastic gradient descent and variants **Wed 4/28:** Activation functions and their derivatives Lab 9: PyTorch optimizers and empirical performance
Week 14 May 3	HW 11 due Mon 5/3 Lab 9 due Mon 5/3	**Mon 5/3:** Softmax, loss functions and course synthesis (last day of classes)
Finals May 6–12	**Final Project Due**: report and presentation at the final exam slot	Final project presentations during the scheduled final exam period (date set by the URI final exam schedule)