

Machine Learning and Optimization (MLOPT)
CSCI 6960 / CSCI 4140
4 credit hours
Spring 2026, RPI
JEC 5119, Tue/Fri 12pm–2pm ET

Instructor: Alex Gittens (he/him)
Email: gittea@rpi.edu
Office: Lally 316 Phone: (518) 276-6476
Office Hours: Tuesday 2pm–3pm, Friday 11am–12pm (ET)

TA: Huzaifa Arif
Email: arifh@rpi.edu

Course website: https://cs.rpi.edu/~gittea/old-site/teaching/spring2026/mlandopt.html
Piazza: https://piazza.com/rpi/spring2026/csci41406962/home
Submitty: https://submitty.cs.rpi.edu/courses/s26/csci4140
https://submitty.cs.rpi.edu/courses/s26/csci6960

Prerequisites. CSCI 2300 **AND** (CSCI 2210 **OR** MATH 4100). Students may not receive credit for both the 4000- and 6000-level versions of this course.

Course synopsis. Modern ML formulates goals as optimization problems and applies suitable algorithms. The first part of the course introduces convex optimization for analyzing deterministic and randomized optimization methods; the second part introduces a selection of modern ML architectures. Homeworks give hands-on algorithmic experience and assess understanding of theoretical aspects. A project gives the students experience with proper research methodology, critically reading the research literature, and effective technical communication.

Required Texts. There is no official text for the course, but supplementary reading material will be added to the course website as we go along. I strongly recommend the following free online resources if you find you have difficulty with the linear algebra or probability used in the course, and for learning the PyTorch framework we use for the deep learning assignments:

Linear Algebra:	Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares. Vandenberghe and Boyd.
Probability:	Introduction to Probability, Statistics, and Random Processes. Hossein Pishro-Nik. Jeff Erickson's notes on discrete probability. Jeff Erickson.
PyTorch/Deep Learning:	PyTorch official tutorial.

Shared Learning Outcomes (CSCI 6960 & 4140). Upon successful completion of this course, each student will be able to:

(S1) Formulate ML tasks as optimization problems.
(S2) Apply convex analysis and optimality conditions.
(S3) Implement first-order and stochastic solvers.
(S4) Discuss trade-offs among solvers.
(S5) Implement deep-learning architectures and explain inductive biases.
(S6) Apply ML to real problems and evaluate performance.

Communication-intensive outcomes

(C1) Analyze and adapt their communication to the technical contexts of machine learning and optimization, clearly identifying the goals, assumptions, and methodological approaches of optimization algorithms and ML architectures.

(C2) Communicate effectively in course-relevant genres, including algorithmic write-ups, empirical experiment summaries, technical reports, and technical presentations.
(C3) Explain optimization concepts and machine-learning methods clearly and succinctly, including the formulation of learning problems, interpretation of model behavior, and justification of algorithmic and architectural design choices.
(C4) Identify and apply appropriate conventions for technical communication in ML and optimization, including the use of mathematical notation, pseudocode, convergence plots, learning-curve visualizations, tables of experimental results, and well-structured written and oral exposition.

Additional Outcomes (CSCI 6960). Upon successful completion of this course, each student will be able to:

(G1) Evaluate strong convexity, smoothness, and nonconvexity in algorithm design.

(G2) Critically evaluate ML literature and communicate conclusions.

(G3) Deduce performance guarantees (convergence and computational cost) for stochastic optimization algorithms.

Course Calendar. The material covered in practice may vary to achieve learning objectives.

Lecture	Topic / Milestones
Lec 1	Introduction to optimization in machine learning; gradient-based methods and their limitations.
Lec 2	Empirical risk minimization and regularization; SVMs and stages of the ML workflow.
Lec 3	Noise models for regression and classification; maximum likelihood estimation.
Lec 4	Cross-entropy loss, KL-divergence, and supervised ML workflow.
	Homework 1 assigned.
Lec 5	Gradients, Hessians, convexity, and examples of convex sets and functions.
Lec 6	Properties and operations preserving convexity; global optimality in convex optimization.
Lec 7	Optimality conditions for smooth and constrained problems; projection onto convex sets.
Lec 8	Subgradients, subdifferential calculus, and nonsmooth convex optimization.
	Homework 2 assigned.
Lec 9	Gradient and subgradient descent; convergence analysis and step-size selection.
Lec 10	Augmented Lagrangians and ADMM
Lec 11	Franke-Wolfe methods
Lec 12	Kernel methods and representer theorem; strong convexity, PL inequality, and convergence rates.
	Homework 3 assigned.
	Comprehensive, closed-book midterm.
Lec 13	Newton's method and stochastic gradient descent; convergence and stepsize scheduling.
Lec 14	Adaptive gradient methods: momentum, Nesterov, AdaGrad, RMSProp, and ADAM.
Lec 15	Neural networks as computational graphs; perceptrons, MLPs, and universal approximation.
Lec 16	Autoencoders and representation learning; backpropagation via the chain rule.
	Homework 4 assigned.
Lec 17	Convolutional neural networks; convolutional layers, pooling, and LeNet.
Lec 18	U-Net and Inception architectures; vanishing/exploding gradients and multiscale designs.
Lec 19	Batch and layer normalization; skip connections and residual networks.
Lec 20	Dropout regularization; recurrent neural networks and backpropagation through time.
	Homework 5 assigned.
Lec 21	Tokenization and embeddings; LSTM and bidirectional RNN architectures.
Lec 22	Sequence-to-sequence models; teacher forcing, beam decoding, and attention mechanisms.
Lec 23	Self-attention and the Transformer encoder; BERT and contextual representations.
Lec 24	Encoder–decoder Transformers; cross-attention, masked attention, and GPT models.
	Homework 6 assigned.
Lec 25	Generative Adversarial Networks; generator–discriminator training and variants (DCGAN, CycleGAN).
Lec 26	Latent generative models and variational autoencoders.
Lec 27	Stable diffusion.

Lecture	Topic / Milestones (cont.)
	Comprehensive final.

Course Assessment Measures. CSCI 4140 and 6960 students complete homeworks, a project, and a midterm and final exam. Graduate homeworks will cover the additional graduate level learning objectives. Graduate projects must be original research projects, while undergraduates have the option of research or pedagogical projects.

Grading components.

CSCI 4140 (Undergraduate)	CSCI 6960 (Graduate)
Midterm (in-person, closed-book) – 25%	Midterm (in-person, closed-book) – 30%
Final Exam (in-person, closed-book) – 25%	Final Exam (in-person, closed-book) – 35%
Homework (code + theory) – 25%	Homework (code + theory) – 15%
Project – 25%	Project – 20%

Grade Thresholds.

CSCI 6960: A $\geq$ 90, B+ $\geq$ 85, B $\geq$ 80, C+ $\geq$ 75, C $\geq$ 70, F $<$70
CSCI 4140: A $\geq$ 90, B+ $\geq$ 85, B $\geq$ 80, C+ $\geq$ 75, C $\geq$ 70, D+ $\geq$ 65, D $\geq$ 60, F $<$60

Homework Format and Verification. Each homework will include a **Code Portion** (implementation and experiments) and, when appropriate, a **Theory Portion** (derivations/proofs). A statement of authorship must be included in *both* portions: *"I completed this work without assistance from AI tools."*

Oral spot-checks and individual explanations may be required to verify authorship. When significant gaps are observed between the understanding exhibited in person and that in submitted work, penalties may include reduction of the score on the assignment up to a failing grade in the course.

Project Overview. The project provides practical experience applying optimization and machine learning techniques on novel or existing real-world datasets. Projects are completed in small groups and culminate in a written report and presentation. This component of the course emphasizes rigorous methodology, critical reading of research literature, rigorous data-driven evaluation, and effective technical communication.

CSCI 4140 (Undergraduate)	CSCI 6960 (Graduate)
Students may complete either a *pedagogical* project (developing instructional materials or demonstrations) or a *research* project (original application or extension of ML or optimization methods).	Graduate students complete an *original research* project addressing a substantial theoretical or applied question; work must show novelty beyond replication or tuning.
All projects include reproducible code in a public repo, a report in ICLR-style format, and a 20-minute presentation.	Graduate projects are also judged on depth of analysis, literature contextualization, and strength of theoretical or empirical contribution.

Project Milestones and Evaluation:

Milestone	Description / Weight
Project Selection (Lec. 8)	Topic proposal with motivation and plan — 10%.
Progress Report (Lec. 20)	Draft with results and analysis — 35%.
Deliverables (Lec. 25)	Reproducible code, data scripts, final report — 20%.
Presentation (TBD)	20-minute presentation and Q&A — 35%.

Evaluation emphasizes technical soundness, interpretive insight, clarity of communication, and reproducibility.

Academic Integrity. Academic honesty is essential to trust and learning. All graded work must represent your own understanding and effort. Plagiarism or unauthorized collaboration will result in a minimum penalty of a course grade of F and may be reported to the Dean of Students or Graduate Education.

Strict AI Prohibition: Use of AI tools (e.g., ChatGPT, Copilot, Gemini, Claude, etc.) to generate or debug code, proofs, derivations, text, figures, or analysis is **strictly prohibited**. The instructor may orally examine any student to confirm authorship. Violations will receive a course grade of F. Institute policy details: `https://info.rpi.edu/dean-students/student-rights-responsibilities-and-judicial-affairs/`.

Academic Accommodations. RPI is committed to equal access for students with disabilities. Contact the Office of Disability Services for Students (dss@rpi.edu; 518-276-8197) to establish accommodations and provide your Faculty Memorandum early in the term.