MATH xxx – Physics-Inspired AI Methods

Laurent Pagnier

Course Description As AI becomes increasingly prevalent across various fields, from finance to engineering, it is essential for our students – who often find themselves in roles at the intersection of IT and management – to have at least a basic understanding of AI methods, along with their strengths and limitations.

In this course, students will explore AI techniques that rely on (or mimic) principles from physics. This unique approach makes these methods more intuitive and appealing for math students. The course is project-oriented, enabling students to tackle real-world problems and develop skills applicable to research, industry, and technology development in an AI-driven world.

Course Prerequisites A foundation in Linear Algebra (e.g., Math 215, 310, or 313), Calculus and Analysis (e.g., Math 125 or 223), Probability and Statistics, and Differential Equations (e.g., Math 254 or 355) is recommended. Additionally, students should be proficient in programming in a science-oriented language (e.g., Python, Matlab, or Julia).

Expected Learning Outcomes

– Understand and apply physics-inspired AI techniques.
– Acquire Practical experience and perfect teamwork skill.
– Conduct and document a short research project.

Scheduled Topics by Week

Week 1 Introduction to the Course, Project, and Objectives
Week 2 Math Preliminary (mainly Linear Algebra, ODE and Statistics)
Week 3 Introduction to Machine Learning: Neural Networks, Autodiff, etc.
(Project Proposal)
Week 4 Dynamical Systems and Governing Equations (if time permits, DMD)
Week 5 Sparse Identification of Nonlinear Dynamics (SINDy)
Week 6 Neural Ordinary Differential Equations (Neural ODEs)
Week 7 Introduction to PDEs, the Heat Equation, Burgers' equation, etc.
(Project Interim Report)
Week 8 Physics-Informed Neural Networks (PINNs)
Week 9 Inverse Problems and Sensitivity Methods
Week 12 Diffusion Models and Stochastic Processes
(Project Draft Report)
Week 13 Invited Lectures (U of A Faculty or Industry Partner)
(or Advanced Simulation and Modeling Techniques (Bayesian Modeling))
Week 14 Group Presentations of Project to Class
(Final Project Report and Deliverables)

References This course will rely mostly on research articles (and maybe on chapters of "Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control").

Project Description and Grading

Description: Students will conduct a semester-long research project utilizing AI methods, preferably one covered in the lectures. Students are encouraged to seek out a relevant problem, potentially by consulting professors for project ideas (e.g., replicating recent research or assisting on ongoing projects).

Group Work: For team projects, students should work in groups of 2 or 3, depending on project scope. Teams will be balanced based on members' backgrounds and experience. All team members are expected to contribute equally, and the entire team will share the same score for each submission.

- **Project Proposal – 10%**
  - Problem Definition (4%): Detailing objectives, feasibility, and relevance.
  - Methodology (4%): Justification of selected methods.
  - Expected Outcomes (2%): Realistic and structured project goals.
- **Interim Report – 20%**
  - Background Research (5%): Related work and theoretical foundation.
  - Preliminary Results (10%): Progress in implementation and initial findings.
  - Troubleshooting (5%): Challenges encountered and solutions.
- **Final Project Report – 30%**
  - Motivation (5%): Comprehensive introduction, context, and objectives.
  - Methodology (10%): Detailed explanation of methods.
  - Results (10%): Quality and analysis of results.
  - Conclusion (5%): Summary and future work suggestions.
- **Project Presentation – 20%**
  - Clarity (5%): Structure, coherence, and pacing of the presentation.
  - Technical Depth (10%): Explanation of methods and challenges.
  - Highlighting of Results (5%): Effective use of visuals to present findings.
- **Project Deliverables (Code & Models) – 20%**
  - Reproducibility (5%): Code runs smoothly and reproduces key results.
  - Quality (10%): Organized code, modularity, and clear documentation.
  - Innovation (5%): Original application of methods to the problem.