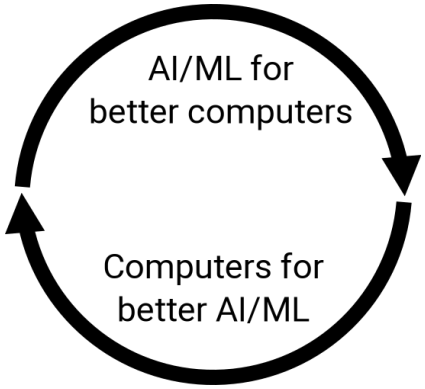
Computer Engineering for and with AI/ML



Robert Dick
https://robertdick.org/

Winter 2027

Summary

This is a new course being offered for the first time, and is currently under design. It is my intention that it ultimately develop into a project-intensive computer engineering MDE in its second iteration. This first offering will cover the appropriate content and will involve several projects, but will have less emphasis on a large-scale culminating projects than future offerings. It is an appropriate course to take if you have some understanding of computer engineering and want to understand how the discipline relates to artificial intelligence / machine learning (AI/ML) and how the two disciplines may influence each other in the future.

The course covers designing computer systems to support AI/ML applications and using AI/ML for computer system design. AI/ML-based applications and solutions are very common: computer engineers will benefit professionally and academically from understanding them.

The course has four stages. In the first, introductory, stage, lectures and a small-scale project are used to establish a basic, shared understanding of AI/ML fundamentals including biological and mathematical understandings of deep neural networks (DNNs), the learning process, and an optimization-based interpretation of learning.

The second stage covers the practical use of AI/ML systems, including big data and small data problems, covariate shift, (misapplication of) the overtraining concept, approaches to understanding what is happening within DNNs, the sudden emergence of capabilities, and beneficial and harmful inductive biases to which deep neural networks are subject.

The third stage covers practical approaches to designing computer systems to efficiently host AI/ML applications. After defining efficiency and explaining why it matters, co-design of AI/ML networks and computer systems in the context of varying applications will be explained. Examples of design optimizations spanning from devices to architectures will be given. The course will also indicate the importance of human-AI/ML interfaces and consider several privacy and security problems brought about by the spread of AI/ML systems, and ways they might be solved or ameliorated.

The fourth stage covers using AI/ML for computer system design, including (semi-)automated software generation; circuit design; verification, validation, and testing; and system-level design automation.

In addition to small-scale, individual projects in the first two stages of the course, there will be a larger-scale team project spanning the last two stages.

Characteristics

- Prerequisites: EECS 370 OR EECS 281.
- Lab schedule: The course will have lab/project assignments, but there will be an open-lab policy so it isn't necessary to schedule a lab time.

- Technical skills: Ability optimize hardware/software architectures for AI/ML applications and to use and modify AI/ML systems for computer system-related applications.
- Evaluation: Homework assignments, projects, and exams.
- Educational outcomes: Understanding of methods to design computer systems for AI/ML tasks and use AI/ML systems to better design computer systems.

Outline of topics

Stage 1: AI/ML fundamentals

1. Biological interpretation.
2. Function approximation and interpretation.
3. Learning and learning rules.
4. Learning is optimization.

Stage 2: AI/ML in the wild

1. Big data and small data problems.
2. Covariate shift.
3. The overtraining concept and its (mis)use.
4. Interpretability and explainability.
5. Knowledge representation.
6. Emergence.
7. Beneficial and harmful inductive bias.

Stage 3: Computer System Design for AI/ML

1. Power, energy, temperature, and reliability in digital systems.
2. Precision.
3. Pruning.
4. Sparse data.
5. Biomimetic systems.
6. Devices.
7. Architectures.
8. Memory hierarchies.
9. User interfaces.
10. Privacy and security.

Stage 4: AI/ML for Computer System Design

1. LLMs for software generation.
2. AI/ML for integrated circuit design automation.
3. AI/ML for testing and error identification.
4. LLMs for system-level design automation.