Machine Learning Systems Implementation
https://cs.rit.edu/~wjz/csci637/

Weijie Zhao

01/13/2026

Machine Learning Systems Implementation

https://cs.rit.edu/~wjz/csci637/

Weijie Zhao

01/13/2026

Machine Learning Systems Implementation

https://cs.rit.edu/~wjz/csci637/

Weijie Zhao

01/13/2026

Machine Learning Systems Implementation

https://cs.rit.edu/~wjz/csci637/

Weijie Zhao

01/13/2026

Machine Learning Systems Implementation

https://cs.rit.edu/~wjz/csci637/

Weijie Zhao

01/13/2026

Course Goals

- Be capable to write your own high-performance machine learning tools on CPUs, GPUs, and multi-node computing clusters
  - Common parallel computing algorithms
  - Communication and synchronization overhead
  - Storage I/O
  - Underlying mechanisms of the framework you use

Course Schedule (Tentative)

- Support vector machines, logistic regression, and gradient descent. Parallel implementations of gradient descent (and its variants) to optimize support vector machines and logistic regressions. The parallel implementation covers how to efficiently utilize multi-threading and GPUs to accelerate the computation. A brief introduction to multi-threading and GPU programming basics is included. (4 weeks)
- Gradient boosting decision trees. Parallel split, quantization implementations, and optimizations on GPUs for tree models are discussed. (3 weeks)
- Neural network preliminary. Parallel tensor operator implementations. Lazy evaluation and code generation. (2 weeks)
- Neural network basic layers. Parallel implementations for backpropagation. (2 weeks)
- Approximate nearest neighbor searching framework. Parallel implementation for locality sensitive hashing, quantization, and proximity graph on CPUs and GPUs. (3 weeks)

Grading

- Homework 40%
- Reading 10%
- Project 50%

- 90%<=A<=100%
- 80%<=B<90%
- 70%<=C<80%
- 60%<=D<70%
- 0%<=F<60%

Grading

- Homework 40%
- Reading 10%
- Project 50%

- 90%<=A<=100%
- 80%<=B<90%
- 70%<=C<80%
- 60%<=D<70%
- 0%<=F<60%

- 5 pieces of homework.
- No late submissions.
- No 3rd party code
- Automatically tested: Please strictly follow the output format. An incorrect format is considered as a wrong answer.
- The best 4 scores among the 5 are counted in your final grade.
- The fastest correct solution in each homework gets 10% bonus score in the final grade.
- Other correct solutions that are no slower than 2X of the fastest one gets 5% bonus score in the final grade.

Grading

- Homework 40%
- Reading 10%
- Project 50%

- 90%<=A<=100%
- 80%<=B<90%
- 70%<=C<80%
- 60%<=D<70%
- 0%<=F<60%

- You may discuss general issues with other students, or use other reference materials. Here, the general discussion precludes the detailed techniques or algorithms for solutions. In this case, any help you receive from someone or the references must be acknowledged.
- Note that this does not mean that someone else can do your work. Unless otherwise explicitly specified, all programming projects and written assignments you submit must be done independently. You are not allowed to look at another student's code/solutions or use the code that others have submitted in the past.
- Failure to acknowledge the source of a significant idea or approach will be considered cheating or plagiarism. It results a direct F.

Grading

- Homework 40%
- Reading 10%
- Project 50%

- 90%<=A<=100%
- 80%<=B<90%
- 70%<=C<80%
- 60%<=D<70%
- 0%<=F<60%

- Present a related technique (from a pool or a pre-approved one) in 20-30 minutes.

https://dblp.org/db/conf/mlsys/mlsys2024.html

Grading

- Homework 40%
- Reading 10%
- Project 50%

- 90%<=A<=100%
- 80%<=B<90%
- 70%<=C<80%
- 60%<=D<70%
- 0%<=F<60%

- Up to 2 students form a group.
- Complete a machine learning framework (from a project pool) and present the project in 20-30 minutes.
- Everyone in the group should be familiar to the entire project. The one who presents the project and answers questions is randomly chosen at the present time.

A Very Basic Example: Linear Regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$

$$X = \begin{pmatrix} \mathbf{x}_1^{\mathsf{T}} \\ \mathbf{x}_2^{\mathsf{T}} \\ \vdots \\ \mathbf{x}_n^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Logistic Regression

- Standard logistic function: $\sigma : \mathbb{R} \to (0, 1)$

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \times log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \times log(h_\theta(x^{(i)})) \right]$$

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

Logistic Regression

What if we have multiple classes?

- Standard logistic function: $\sigma : \mathbb{R} \to (0, 1)$

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \times log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \times log(h_\theta(x^{(i)})) \right]$$

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

Gradient Descent

- SGD
- Momentum
- AdaGrad
- RMSProp
- Adam

Further reading: https://en.wikipedia.org/wiki/Stochastic_gradient_descent