AMATH 840: Advanced Numerical Methods for Computational and Data Sciences

Giang Tran
Department of Applied Mathematics, University of Waterloo
Winter 2024

Introduction to AMATH 840

Outline

Course Outline

Basic Steps of a Learning Process

Course Introduction

Goal: Study some computational and mathematical perspectives of machine learning and data science.

1. Sparse Optimization and Compressed Sensing: Sparse optimization methods; reconstruction guarantees; applications to image and signal processing, model selection, and function approximation

2. Neural Networks: Mathematical formulations of popular NN architectures, universal approximation, adjoint methods & automatic differentiation, implicit and explicit regularizations, stochastic gradient method and its accelerations, overparametrization, the importance of effective initialization.

3. Randomized Linear Algebra: Johnson-Lindenstrauss lemma, matrix approximation by sampling, randomized QR and SVD, random projections, with applications to model reduction and large-scale problems.

Course Logistics

- Student Assessments: 50% Assignments + 50% Final project.
- Assignments: Theoretical + Computational questions.
- Final Project:
  - Each student will give a presentation of 25 minutes.
  - Each student will submit the slides and a short report (10-25 pages).
  - Suggested content in the short report: problem statement/formulation, how to solve the problem, and/or numerical experiments.

Course Logistics

Course Websites:

- LEARN: To check course outline, course notes, recorded videos, assignments, supplementary materials, and important announcements.
- Crowdmark: To submit and see marked assignments. For each assignment, you will receive an invitation from Crowdmark to submit your assignment.
- Discussion Forum: To post questions about lectures, assignments, textbooks, etc, please sign up for the course discussion board at Piazza, via the following link: `piazza.com/uwaterloo.ca/winter2024/amath840`.

More details can be found in the course outline on Learn or on `online.uwaterloo.ca`.

Outline

Basic Steps of a Learning Process

1. Collect and preprocess data: data cleaning, data augmentation, normalized/standardized data
    - Data resources: public datasets (UCI dataset, Kaggle,...), data from experiments, simulated data
2. From raw or preprocessed data, generate training data, validation data, and/or test data
    - Training data = a collection of samples that will be used to learn the model. For example, in a regression problem, a training dataset is

      $$\left(\mathbf{x}^{(i)}, y^{(i)}\right)_{i=1}^{m},$$

      where

      $$\begin{aligned} m &:= \#\text{training samples} \\ \mathbf{x}^{(i)} &:= \text{sample's features/input data} \in \mathbb{R}^d \\ y^{(i)} &:= \text{sample's label/output data} \in \mathbb{R}^n. \end{aligned}$$

    - Validation (test) data = a collection of samples that will be used to validate(test) the learned model.

Basic Steps of a Learning Process (cont'd)

3. Choose a suitable learning model.

   Example 1: Given a training dataset $\left(\mathbf{x}^{(i)}, y^{(i)}\right)_{i=1}^{m} \in \mathbb{R}^d \times \mathbb{R}$, we would like to learn a function $f : \mathbb{R}^d \to \mathbb{R}$ such that $f(\mathbf{x}^{(i)}) \approx y^{(i)}, \ \forall i \in [m]$.

   - Some potential candidates for f are:

     3.1 Linear model:

     $$f(\mathbf{x}) = f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_d x_d = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}^T \mathbf{w},$$

     where $\mathbf{w} = \begin{bmatrix} w_0 & w_1 & \cdots & w_d \end{bmatrix}^T$.

     3.2 Generalized linear model:

     $$f(\mathbf{x}) = \sum_{k=1}^{H} w_k \, \phi_k(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) & \phi_2(\mathbf{x}) & \cdots & \phi_H(\mathbf{x}) \end{bmatrix} \mathbf{w},$$

     where $\phi_k(\mathbf{x})$ are prescribed nonlinear functions, such as monomials, orthogonal polynomials, and trigonometric functions. For example, curve-fitting data by a polynomial of degree p:

     $$f(\mathbf{x}) = w_0 + w_1 x_1 + \cdots + w_d x_d + w_{1,1} x_1^2 + w_{1,2} x_1 x_2 + \cdots + w_{d,\ldots,d} x_d^p.$$

Basic Steps of a Learning Process (cont'd)

3.3 Neural network model:

$$f(\mathbf{x}) = W_L\sigma(W_{L-1}\sigma(\ldots(W_2\sigma(W_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2)\ldots) + \mathbf{b}_{L-1}) + \mathbf{b}_L,$$

where unknowns $W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}, \mathbf{b}_\ell \in \mathbb{R}^{n_\ell}$, for $\ell = 1, \ldots L$. Here L is number of layers (prespecified).

Basic Steps of a Learning Process (cont'd)

4. Choose a loss function + (optional) a regularization: How well a model fits the data. In Example 1, the loss can be

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^{m} |y^{(i)} - f(\mathbf{x}^{(i)})|^2 = \frac{1}{2m} \|\mathbf{y}_{given} - \mathbf{y}_{model}\|_2^2,$$

where

$$\mathbf{y}_{given} = [y^{(1)}\, y^{(2)}\, \cdots y^{(m)}]^T, \quad \mathbf{y}_{model} = [f(\mathbf{x}^{(1)})\, f(\mathbf{x}^{(2)})\, \cdots f(\mathbf{x}^{(m)})]^T.$$

- Beside the Euclidean distance $\|\cdot\|_2$, Wasserstein distance is another popular choice.
- Popular regularizations: $\frac{\lambda}{2}\|\mathbf{w}\|_2^2$ and $\lambda\|\mathbf{w}\|_1$.

Basic Steps of a Learning Process (cont'd)

Note that $\mathbf{y}_{model}$ can be written in matrix-vector multiplication form. Specifically, in Example 1,

4.1 Linear model: Assume

$$f(\mathbf{x}) = f(\mathbf{x}) = f(\mathbf{x}; \mathbf{w}) = w_0 + w_1\, x_1 + \ldots + w_d\, x_d,$$

then

$$\mathbf{y}_{model} = \begin{bmatrix} 1 & \mathbf{x}^{(1),T} \\ 1 & \mathbf{x}^{(2),T} \\ \vdots & \\ 1 & \mathbf{x}^{(m),T} \end{bmatrix} \mathbf{w} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_d^{(2)} \\ \vdots & \vdots & & \vdots & \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots & x_d^{(m)} \end{bmatrix} \mathbf{w} = [\mathbf{1}_m\ X]\mathbf{w},$$

where the data matrix $X \in \mathbb{R}^{m \times d}$ and the unknown $\mathbf{w} \in \mathbb{R}^{d+1}$.

Basic Steps of a Learning Process (cont'd)

4.2 Generalized linear model: Assume

$$f(\mathbf{x}) = w_0 + w_1 x_1 + \cdots + w_d x_d + w_{1,1} x_1^2 + w_{1,2} x_1 x_2 + \cdots + w_{d,d} x_d^2,$$

then

$$\mathbf{y}_{model} = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_d^{(1)} & (x_1^{(1)})^2 & x_1^{(1)} x_2^{(1)} & \cdots & (x_d^{(1)})^2 \\ 1 & x_1^{(2)} & \cdots & x_d^{(2)} & (x_1^{(2)})^2 & x_1^{(2)} x_2^{(2)} & \cdots & (x_d^{(2)})^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_1^{(m)} & \cdots & x_d^{(m)} & (x_1^{(m)})^2 & x_1^{(m)} x_2^{(m)} & \cdots & (x_d^{(m)})^2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \\ w_{1,1} \\ \vdots \\ w_{d,d} \end{bmatrix}$$

$$= \Phi(X)\mathbf{w},$$

where the dictionary matrix (given) $\Phi(X) \in \mathbb{R}^{m \times \frac{(d+1)(d+2)}{2}}$ and the unknown vector $\mathbf{w} \in \mathbb{R}^{(d+1)(d+2)/2}$.

Basic Steps of a Learning Process (cont'd)

4.3 Neural network: Approximate f by a shallow network:

$$f(\mathbf{x}) = W_2\sigma(W_1\mathbf{x} + b_1) + b_2 = \sigma(\mathbf{x}^T W_1^T + \mathbf{b}_1^T)W_2^T + b_2,$$

then

$$\mathbf{y}_{model} = \sigma(XW_1^T + \mathbf{1}_m\mathbf{b}_1^T)W_2^T + b_2\mathbf{1}_m.$$

Basic Steps of a Learning Process (cont'd)

5. Learn the model (model parameters) to minimize the loss on training data.
6. Compute the generalization error, i.e., error of the trained model on new data.

   Computation resources:

   - Compute Canada: `https://www.computecanada.ca/`
   - Google Colab: `https://colab.research.google.com/`
   - Jupyter Notebook Server from MFCF: `https://uwaterloo.ca/math-faculty-computing-facility/services/jupyter-notebook-server`
   - GPU Computing from MFCF with four A100 GPU: `https://uwaterloo.ca/math-faculty-computing-facility/services/specialty-research-linux-servers`