CMSC 838B & 498Z: Differentiable Programming

Tues/Thur 12:30pm – 1:45pm
http://www.cs.umd.edu/class/fall2023/cmsc838b

Ming C. Lin
IRB 5162
lin@cs.umd.edu
http://www.cs.umd.edu/~lin
Office Hours: After Class or By Appointment

Prerequisites

- CMSC 330
- CMSC 351
- CMSC 422
What you should know?

- Fundamentals of statistical learning, MLPs, gradient descent, how to train and evaluate learning machines, supervised-vs-unsupervised)
- Fundamentals of:
  Matrix Algebra (matrix-matrix, matrix-vector and matrix-scalar operations, inverse, determinant, rank, Eigendecomposition, SVD); Probability & Statistics (1st-order summary statistics, simple continuous and discrete probability distributions, expected values, etc); and, Multivariable Calculus (partial differentiation, chain-rule).
- Programming in Python (not required)

What you might already know?

- How to use a deep learning framework (Keras, Tensorflow, PyTorch)?
- How to train an existing model architecture using a GPU?
- How to perform transfer learning?
- How to perform differentiable sampling of a Multivariate Normal Distribution?
Textbook & References

- In-class handouts
- Other research papers
- More references (books, papers, pointers to other interesting resources) available at the course website

Differentiable Programming

- a novel programming paradigm, where programs are treated as compositions of differentiable operations that can then be optimized to fit data with tremendous possibility to transform how we can utilize computers to perform calculations on “big data”
Differentiable Programming

- Coined by Yann Lecun\(^1\) to describe a superset of Deep Learning
- Captures the idea that computer programs can be constructed of parameterized functional blocks in which the parameters are learned using some form of gradient-based optimization
- The implication is that we need to be able to compute gradients with respect to the parameters of these functional blocks. We'll start exploring this in detail next week...
- The idea of Differentiable Programming also opens up interesting possibilities:
  1) The functional blocks don't need to be direct functions in a mathematical sense; more generally they can be algorithms.
  2) What if the functional block we're learning parameters for is itself an algorithm that optimizes the parameters of an internal algorithm using a gradient-based optimizer?!

\(^1\)[https://www.facebook.com/yann.lecun/posts/10155003011462143]

Course Objectives

- Have an overview of the underlying mathematical and algorithmic principles of differentiation
- Understand the key factors that make differentiable programming successful for various applications
- Be able to read papers on differentiable programming
- Gain facility in working with differentiable programming
- Critically appraise the merits and shortcomings of different methods on specific problems
- Apply differentiable programming to real applications
Possible Applications

- Bioinformatics
- Computer Architecture
- Computer Graphics
- Computer Vision
- Programming Languages
- Natural Language Processing
- Quantum Computing
- Robotics and Automation

Topics Covered (Tentative)

- Overview & Introduction (Tues, Aug 29, 2023)
- Background and Reviews (Thur, Aug 31, 2023)
- Differentiation (Tues, Sept 5, 2023)
- MLP & Backpropagation (Thur, Sept 7, 2023)
- Differentiable Programming (Tues, Sept 12, 2023)
- Automatic Differentiation (Thur, Sept 14, 2023)
- Optimization (Tues, Sept 19, 2023)
- Differentiable Physics (Thur, Sept. 21, 2023)
- Differentiable Rendering (Sept/Oct, 2023)
- Differentiable Rendering (Sept/Oct, 2023)
- Differentiable Geometry Processing (Oct, 2023)
- Differentiable Image Processing (Oct, 2023)
- Differentiable Vision (Oct, 2023)
Topics Covered

- Project Proposal (Oct, 2023)
- Special Topics (October-December, 2023)
- Course Project Progress Report (Nov, 2023)
- NO CLASS: THANKSGIVING BREAK
- Final Course Project Presentation (Dec, 2023)

Grading

- 1-Credit Option
  - Participation (100%)

- 3-Credit Option
  - In-Class Exams (30%)
  - Class Presentation & Participation (20%)
  - Final Course Project (50%)
    - In-Class Presentations: Proposal/Progress/Final
    - Large-scale Programming & Software Engineering
See Course Website

www.cs.umd.edu/class/fall2023/cmsc838b

for more details…….