Module 0.0 - Introduction

Loading [MathJax]/extensions/Safe.js

Welcome

- CS 5781 - Machine Learning Engineering

Loading [MathJax]/extensions/Safe.js

Contents

- Class Context
- Staff Introduction
- Class Goals
- Class Syllabus

Class Context

- Development of deep learning models
- Deep learning models in industrial context
- Programming large systems

Loading [MathJax]/extensions/Safe.js

Context

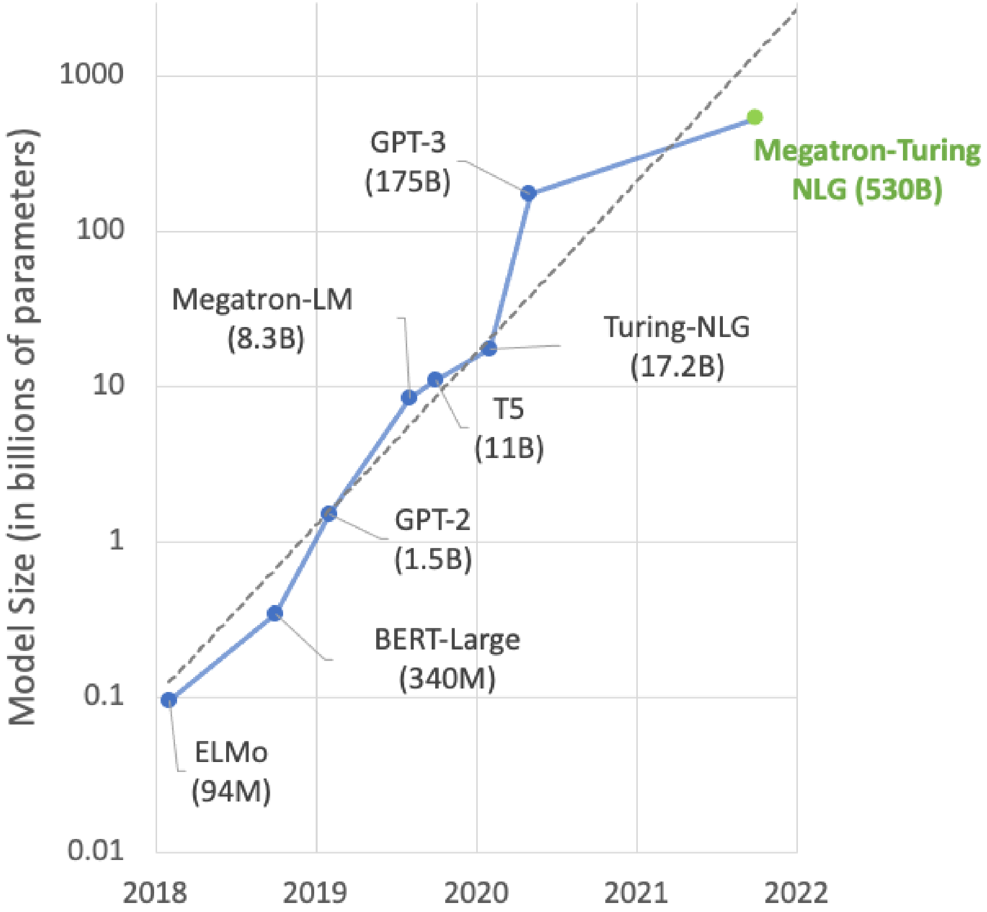
- Stable Diffusion
- ChatGPT
- AlphaFold

Issues: Model Data and Bias

Big Tech builds AI with bad data. So scientists sought better data.

Loading [MathJax]/extensions/Safe.js

Issues: Model Scale



Loading [MathJax]/extensions/Safe.js

Question

- How do I "read" one of these models?
- What technology is powering these tools?
- How can I build models for my own problem?

Staff Introduction

Loading [MathJax]/extensions/Safe.js

Professor

- Alexander "Sasha" Rush

Academic Work

- Website: http://rush-nlp.com/
- Area of Study: Natural Language Processing (NLP)
- Area of Study: Deep Learning

Loading [MathJax]/extensions/Safe.js

Academic Work: Projects

- Automatic text summarization
- Accurate math OCR
- Machine learning on cell phones

My Path

- Coder -> Student -> Industry -> Professor
- Professor at Harvard for 5 years
- Moved to Cornell Tech 3 years ago

Intro: Open-Source

- Open-source development projects for NLP OpenNMT
- Contributor to PyTorch
- Part-time at Hugging Face

TAs

- TAs

Loading [MathJax]/extensions/Safe.js

Loading [MathJax]/extensions/Safe.js

Class Introduction

Loading [MathJax]/extensions/Safe.js

Loading [MathJax]/extensions/Safe.js

Class Focus

- Machine Learning **Engineering**
- Focus: software engineering behind machine learning

Applied Machine Learning

- Coverage of different models and learning setups
- Focus on algorithms and mathematical underpinnings
- Broad coverage of the field and its future

Loading [MathJax]/extensions/Safe.js

Machine Learning Engineering

- Focus on implementation details and design
- Deep dive into implementation
- (For those who care about the weeds)

Machine Learning

- Rich and interesting field
- Building models is a core skill
- Probabilistic reasoning for decision making

Hidden Factor

Many recent successes based on:

- Hardware
- Tooling
- Brute-force search

Skill Set of a ML Engineer

- Math
- Experimentation
- *Systems*

Loading [MathJax]/extensions/Safe.js

Machine Learning Systems

Machine Learning Engineering

- ML practitioners build large-scale mathematical systems.
- Tooling has been key to speed up ML development.
- Most work done in *Deep Learning frameworks*.

Loading [MathJax]/extensions/Safe.js

Deep Learning Frameworks

- Implement mathematical functions as efficient code
- Provide organization and structure to ML projects
- Allow for easy training and deployment
- Think: "Programming language for machine learning"

Loading [MathJax]/extensions/Safe.js

Popular Frameworks

- TensorFlow

- PyTorch

Deep Learning Frameworks

Example of code in PyTorch.

```
class Network(torch.nn.Module):
    def __init__(self, hidden):
        super().__init__()
        self.layer1 = torch.nn.Linear(2, hidden)
        self.layer2 = torch.nn.Linear(hidden, hidden)
        self.layer3 = torch.nn.Linear(hidden, 1)

    def forward(self, x):
        h = self.layer1.forward(x).relu()
        h = self.layer2.forward(h).relu()
        return self.layer3.forward(h).sigmoid()
```

Deep Learning Frameworks

- Used for all the major projects shown.
- Provide easy user programming interface
- Connect to fast hardware under the hood

ML Day-to-Day

- Data scientist or ML practitioners and use these systems
- However, an ML Engineer should really know what is going on...

CS 5781

We're going to build PyTorch.

Course Outline

Loading [MathJax]/extensions/Safe.js

My Learning Philophy

- Engineering is learned through implementing
- You don't understand it until the tests pass
- Build your own demos

Learning Objectives

- Reason about the requirements for large system systems
- Be comfortable designing and testing mathematical code
- Gain confidence reading large open-source codebases

Learning Non-Objectives

- Rigorous understanding of mathematical foundations
- Development of new or creative models
- Details of state-of-the-art ML systems

Loading [MathJax]/extensions/Safe.js

Course Style

- Highly applied, focus on building
- Project directed, questions from students
- Interactive and grounded in the project

PyTorch

- Big codebase on CPU and GPU
- Large team of professional developers
- Used in thousands of academic papers
- Deployed by Facebook, Uber, Tesla, Microsoft, OpenAI ...

Challenge

How are you going to build PyTorch?

Loading [MathJax]/extensions/Safe.js

Course Project

- 5 modules walking you through the process
- Each covers a different topic in MLE
- Final module yields a full image recognition system.

Loading [MathJax]/extensions/Safe.js

Course Work

```python
class ReLU(ScalarFunction):

    @staticmethod
    def forward(ctx, a):
        # TODO: Implement for Task 1.2.
        raise NotImplementedError('Need to implement for Task 1.2')

    @staticmethod
    def backward(ctx, d_output):
        # TODO: Implement for Task 1.4.
        raise NotImplementedError('Need to implement for Task 1.4')
```

Modules

- ML Programming Foundations
- Autodifferentiation
- Tensors
- GPUs and Parallel Programming
- Foundational Deep Learning

Grading

- Assignments - Completion and Correctness
- Midterm
- In-Class Quizzes
- Assignments are done individually

Tools

- Github Classroom
- Ed Discussions
- Slido

Loading [MathJax]/extensions/Safe.js

Caveats

Course Prerequisites

- Programming experience
- Mathematical notation / calculus experience
- Willingness to debug

Next Lecture

- Getting dev setup
- Getting started for Module-0
- Come ready to program.

Q & A

Loading [MathJax]/extensions/Safe.js