EE 628
Deep Learning
Fall 2019

Lecture 1

08/29/2019

Assistant Prof. Sergul Aydore

Department of Electrical and Computer Engineering

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

About Me

- **Assistant Prof** at ECE department at **SIT** since *August 2018*
- **Machine Learning Scientist** at **Amazon** *2016 – 2018*
- **Data Scientist** at **JP Morgan** *2015-2016*
- **Postdoctoral Researcher** at **Columbia University** *2014 – 2015*
- **PhD Student** at Signal and Image Processing Institute of **University of Southern California** *2009 – 2014*
- **BS and MS student** at Electrical Engineering from **Bogazici University** *2002 - 2009*

My Interests

- Machine Learning
- Signal Processing
- Software Development
- Brain Imaging
- Teaching

- http://www.sergulaydore.com/

About the Course

- Will use canvas for communication and grades
- Lecture notes will be shared in public Github repo:

  https://github.com/sergulaydore/EE-628-Fall-2019
- Tools that students will learn how to use are:
  - PyTorch, github, ipython notebook, AWS
- Tools that I will be using to organize assignments are:
  - Github education: https://classroom.github.com/

General Information

- **Meeting Times**: Thursdays at 3.30pm-6.00pm
- **Classroom Location**: Main Campus, ABS 301
- **Contact Info**: Burchard Building 211, saydore@stevens.edu
- **Office Hours:** Thursday at 1.00pm-3.00pm
- **Teaching Assistants:** @ Burchard Building 315
  - Tianhao Zhu: tzhu12@stevens.edu, Office hours: Wednesdays 1.00pm – 3.00pm
- **Formal Name:** Data Acquisition/Proc II

Course Description

- This course presents a solid grasp of the **practical techniques of deep learning**. The course formally introduces intuition and ideas to successfully apply deep learning. Although deep learning requires both mathematics and programming, we will prioritize intuition over mathematical rigor. However, we will assume that the students are familiar with the basics of statistics, linear algebra and Python programming. We will relate all concepts to a practical end.

STUDENT LEARNING OUTCOMES

- After successful completion of this course, students will be able
  - how to cast a problem that can be solved with deep learning,
  - the mathematics of modeling,
  - algorithms for fitting models to data,
  - engineering techniques to implement it all.

FORMAT AND STRUCTURE

- The classes will include dynamic illustrations of the concepts. Students are expected to bring their laptops and run the programs in their own laptops.
- There will be weekly coding and math assignments. The students will return coding assignments via github.
- Midterm exam will be handwritten! Because tech companies will interview you on a white board!

Course Materials

- **Textbook(s):** Dive into Deep Learning by A. Zhang, Z. Lipton, M. Lu, A. Smola https://www.d2l.ai/index.html
  - **Other Readings:**
    - http://cs231n.stanford.edu
    - https://people.csail.mit.edu/madry/6.883/

Canvas will be used for sharing assignment links and grades

Course Requirements

- **Attendance:** Attendance is crucial for an effective learning but will not be graded.
- **Homework:** Assignments will be submitted via github.
- **Team Project:** Students are expected to finish a project that is documented in their github repository.
- **Exams:** There will be a single hand-written midterm exam.

Grading Procedures

- Grades will be based on:
  - Assignments (30 %)
  - Team Project (40 %)
  - Midterm Exam (30 %)

- Late Policy:
  - Assignments submitted after the deadline will not be graded.

IMPORTANT DATES

- Midterm exam **10/18/2019 Thursday at 3.30pm**
- Deadline for Project Proposals **11/01/2019 Friday at 5pm ET**
  - This includes creation of a github repository with READ.md file that contains the summary of the project.
  - Late submissions or repositories with empty READ.md file will lose 30 points from their grade for the project.
- Deadline for Projects **12/02/2019 Monday at 5pm ET**
  - Projects will be graded based on
    - project report with at least 6 pages using the style file (40 %)
    - project presentation (40 %)
    - organization of the github page (20 %)

Graduate Student Code of Academic Integrity

- *All Stevens graduate students promise to be fully truthful and avoid dishonesty, fraud, misrepresentation, and deceit of any type in relation to their academic work. A student's submission of work for academic credit indicates that the work is the student's own. All outside assistance must be acknowledged. Any student who violates this code or who knowingly assists another student in violating this code shall be subject to discipline.*

- All graduate students are bound to the Graduate Student Code of Academic Integrity by enrollment in graduate coursework at Stevens. It is the responsibility of each graduate student to understand and adhere to the Graduate Student Code of Academic Integrity. More information including types of violations, the process for handling perceived violations, and types of sanctions can be found at www.stevens.edu/provost/graduate-academics.

Special Provisions for Undergraduate Students in 500-level courses

- The general provisions of the Stevens Honor System do not apply fully to graduate courses, 500 level or otherwise. Any student who wishes to report an undergraduate for a violation in a 500-level course shall submit the report to the Honor Board following the protocol for undergraduate courses, and an investigation will be conducted following the same process for an appeal on false accusation described in Section 8.04 of the Bylaws of the Honor System. Any student who wishes to report a graduate student may submit the report to the Dean of Graduate Academics or to the Honor Board, who will refer the report to the Dean. The Honor Board Chairman will give the Dean of Graduate Academics weekly updates on the progress of any casework relating to 500-level courses. For more information about the scope, penalties, and procedures pertaining to undergraduate students in 500-level courses, see Section 9 of the Bylaws of the Honor System document, located on the Honor Board website.

Learning Accomodations

- Stevens Institute of Technology is dedicated to providing appropriate accommodations to students with documented disabilities. For more information about Disability Services and the process to receive accommodations, visit https://www.stevens.edu/office-disability-services. If you have any questions please contact: Phillip Gehman, the Director of Disability Services Coordinator at Stevens Institute of Technology at pgehman@stevens.edu or by phone (201) 216-3748.

- See https://github.com/sergulaydore/EE-628-Fall-2019 for the rest

Deep Learning Today

- In just the past five years, deep learning has taken the world by surprise, driving rapid progress in fields as diverse as computer vision, natural language processing, automatic speech recognition, reinforcement learning, and statistical modeling.
- With these advances in hand, we can now
  - build cars that drive themselves
  - smart reply systems that anticipate mundane replies,
  - helping people dig out from mountains of email, and
  - software agents that dominate the world's best humans at board games like Go,
- a feat once deemed to be decades away. Already, these tools are exerting a widening impact, changing the way movies are made, diseases are diagnosed, and playing a growing role in basic sciences – from astrophysics to biology.

- This course will teach you both the concepts, the context, and the code.

Applying Deep Learning Requires Understanding

1. the motivations for casting a problem in a particular way,
2. the mathematics of a given modeling approach,
3. the optimization algorithms for fitting the models to data
4. the engineering required to train models efficiently
   - navigating the pitfalls of numerical computing
   - getting the most out of available hardware

Software and Computation Tools

- Python
- Pytorch
- AWS

Target Audience

- Students who seek a solid grasp of the practical techniques of deep learning
- No previous background in deep learning or machine learning is required
- However, basics in linear algebra, calculus, probability and Python Programming are required.

Imagine Tackling any of the following problems

- Write a program that predicts tomorrow's weather given geographic information, satellite images, and a trailing window of past weather.
- Write a program that takes in a question, expressed in free-form text, and answers it correctly.
- Write a program that given an image can identify all the people it contains, drawing outlines around each.
- Write a program that presents users with products that they are likely to enjoy but unlikely, in the natural course of browsing, to encounter.

Machine Learning (ML)

- Machine learning (ML) is the study of powerful techniques that can learn behavior from experience.
- As ML algorithm accumulates more experience, their performance improves.
- In this course, you will learn fundamentals of ML, focus in particular on deep learning, a powerful set of techniques driving innovations in areas as diverse as computer vision, natural language processing, healthcare, and genomics.

Motivating Examples

- Imagine writing a program to respond to a wake word like Alexa, Okay Google, Siri

- Program a cat detector

Figures are from the book Dive into Deep Learning by Zhang et al.

A typical Training Process

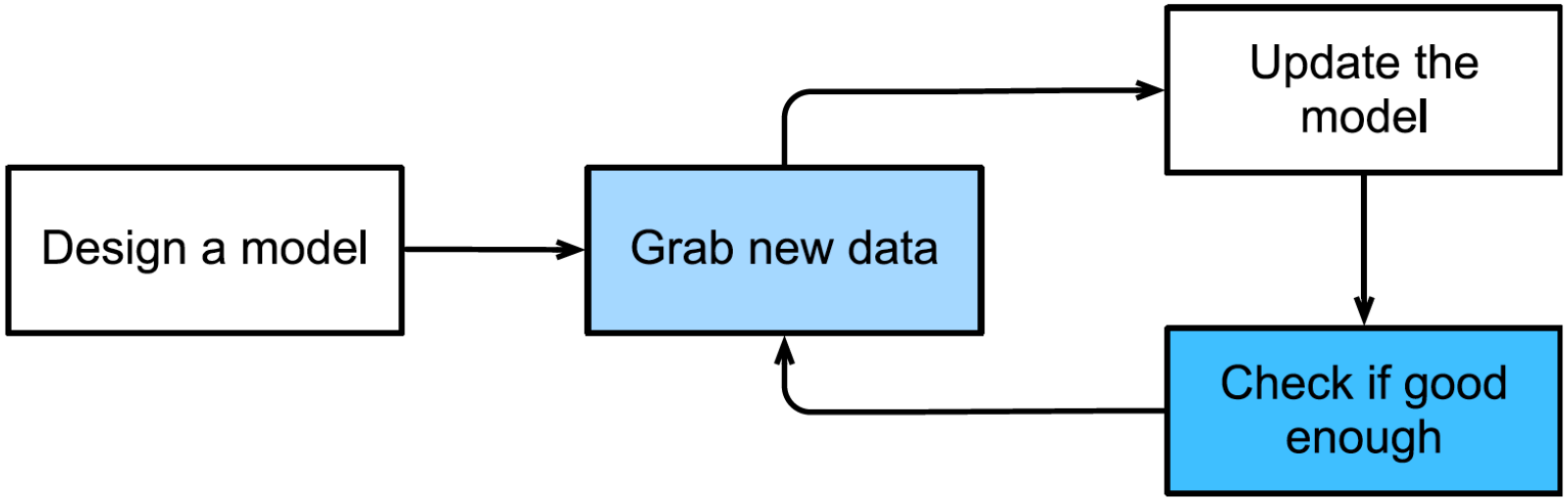


Figure is from the book Dive into Deep Learning by Zhang et al.

1. Start off with a randomly initialized model that can't do anything useful.
2. Grab some of your labeled data (e.g. audio snippets and corresponding {yes,no} labels)
3. Tweak the knobs so the model sucks less with respect to those examples
4. Repeat until the model is awesome.

Deep Learning

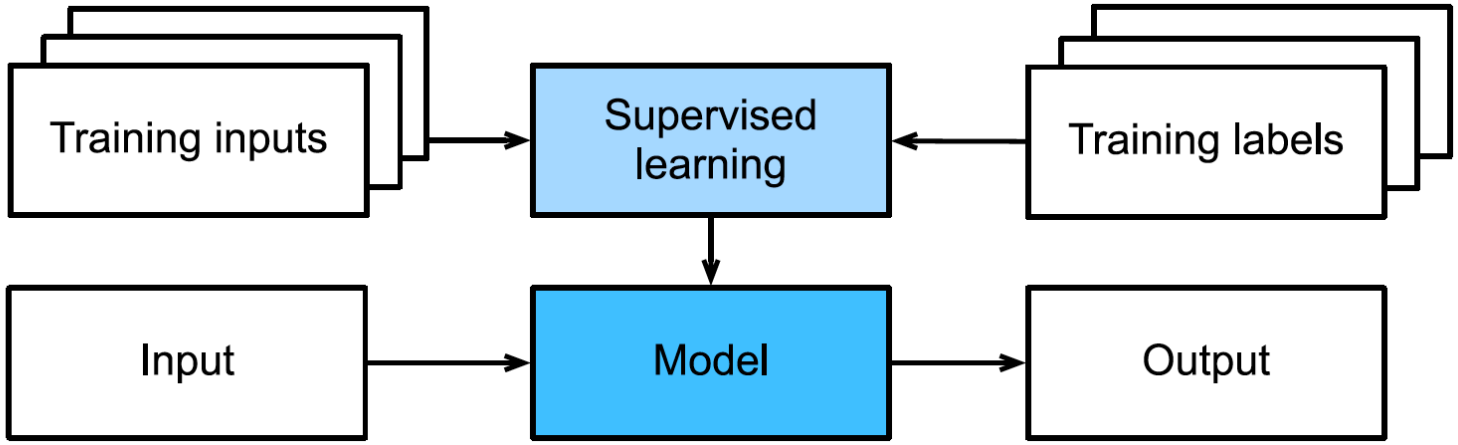
- Deep learning is just one among many popular frameworks for solving machine learning problems.
- In the problems we discussed deep learning excels and traditional ML tools faltered
- Deep models are deep: they learn many layers of computation
- Advantage: replaces labor-intensive feature engineering

The Key Components

- No matter what kind of an ML problem that we take on, core components are:
  1. The **data** that we can learn from
  2. A **model** of how to transform the data
  3. A loss function that quantifies the **badness** of our model
  4. An algorithm to adjust the model's parameters to minimize the loss

Kinds of Machine Learning

1. Supervised Learning: addresses the task of predicting targets given input data.
2. Unsupervised Learning
3. Interacting with an Environment
4. Reinforcement Learning



Supervised Learning

- Regression:
  - How many?
- Classification:
  - Is this a _ ?
- Tagging
- Search and ranking
- Recommender systems
- Sequence Learning
- …

Unsupervised Learning

- Clustering
- Subspace estimation (e.g. PCA)
- Representation Learning
- Directed Graphical Models and causality
- Generative Adversarial Networks

Install tools

- We will be using the following tools extensively so please make sure that you installed them
  - Python 3 https://www.python.org/downloads/
  - pip https://pip.pypa.io/en/stable/
  - PyTorch https://pytorch.org/get-started/locally/
  - Mxnet http://mxnet.incubator.apache.org/versions/master/install/index.html?platform=Linux&language=Python&processor=CPU
  - Dl2 https://www.d2l.ai/chapter_install/install.html
  - Ipython notebook https://ipython.org/install.html