Syllabus

Spring 2019

Instructors

Professor Konrad Kording, Jeffrey Cheng, David Rolnick, Nidhi Seethapathi

Course Description

Deep learning techniques now touch on data systems of all varieties. Sometimes, deep learning is a product; sometimes, deep learning optimizes a pipeline; sometimes, deep learning provides critical insights; sometimes, deep learning sheds light on neuroscience.

The purpose of this course is to deconstruct the hype by teaching deep learning theories, models, skills, and applications that are useful for applications. The class will focus on the following 5 questions.

1. How do we decide which problems to tackle with deep learning?

   Not every problem should be tackled with deep learning. It's data-hungry, catastrophically forgets, extrapolates poorly, cannot easily perform estimates of uncertainty, cannot easily transfer knowledge, and cannot be audited.

   Deep learning has been wildly successful in certain domains (CV, NLP, and RL come to mind), and students will understand how to determine what kinds of models are best for which problem settings.

2. Given a problem setting, how do we determine what model is best?

   Suppose a problem has variable-length input. There are a range of popular ways to deal with such problems: truncation, padding, aggregated convolution, and recurrence. For any given problem, some approaches are preferable. We will cover popular models, discuss how to find interesting niche research, and gain intuition for the best approaches.

3. What's the best way to implement said model?

   The course begins with the mechanics of modern deep learning frameworks and autograd systems. Students will build on this knowledge by implementing widely-used models, cutting-edge research, and an open project of their choosing on a unique dataset. We will also cover interesting research on how to design deep learning systems (e.g. choosing the depth and the number of hidden neurons).

4. How can we best visualize, explain, and justify our findings?

   The success of a deep learning project often depends on the interpretation and presentation of results. The course will cover how to choose evaluation criteria, how to determine next steps from the empirical performance of a model, which visualizations are the most useful and intuitive, and how to write up applied findings for both technical and non-technical audiences. Perhaps most importantly, the course will discuss when to use deep learning rather than simpler models (e.g. SVM, shallow networks).

5. How can neuroscience inspire deep learning?

   For many innovations in deep learning research, neuroscience has directly or indirectly acted as an inspiration. We will thus, along with the material on implementations always teach ideas from neuroscience, presented compactly, about how DL ideas may relate to biology.

Prerequisites

Exposure to linear algebra, probability, statistics, Python. One of CIS 519/520 is highly encouraged.

Reference Materials

Deep Learning (Adaptive Computation and Machine Learning) (Goodfellow, Bengio, Courville)

Grading

The course will have theoretical and applied homeworks as well as a final project. The intent of the project is to target the 2 most important skills in ML practice: the ability to implement academic findings, and the ability to address datasets with high-performing models. The project and the PyTorch homeworks will be evaluated on the correctness of the code, performance of the models, and the clarity of the reports.

Assignment	Proportion
Participation	10%
Homework 0	10%
Homework 1	10%
Homework 2	15%
Homework 3	15%
Final Project	40%

Assignment	Due Date	Description
Homework 0: Finger Exercises	1/30	Review of classification and regression setup, loss functions, cross-validation, backpropagation.
Homework 1: Setup	2/18	Test local setup and Google Colab; get familiar with the PyTorch documentation and the basic ML benchmarks (Iris, MNIST variations, CIFAR-10). Implement variations of feedforward nets. Populations of neurons. The hierarchy of the visual system
Homework 2: Computer Vision	3/11	Compare regression, feedforward nets, CNNs, and capsule nets on facial recognition data.
Homework 3: NLP	4/3	Create a word embedding and detect sarcasm on Reddit while varying encoding, featurization, and model.
Final Project	4/29	Find or construct a dataset, implement a range of models on it, evaluate and visualize performance of neural and non-neural techniques, and write 2 reports: one for a technical audience and one for a non-technical audience.

Other logistics

- The course size will be roughly 30-40 students.
- We plan to publish course materials over Canvas and maintain an active Piazza for student discussion.
- For student GPU access, we plan to use Google Colab. Colab is a free service that gives users a GPU (or TPU) instance within Google's cloud computing framework for computations lasting up to 24 hours. This is sufficient for the computational tasks we will require of students. Colab is easy to use (similar to a Jupyter notebook) and interfaces easily with PyTorch.

Schedule

Week	Topics	References
1	Lecture W: Intro to Deep Learning	Goodfellow Ch. 1-5
2	Lecture W: PyTorch and Computational Graphs	Goodfellow Ch. 6
3	Lecture M: Neural Nets Lecture W: Training Deep Nets	Goodfellow Ch. 7-8
4	Lecture M: Deep Learning vs. Shallow Learning Lecture W: CNNs	Goodfellow Ch. 9
5	Lecture M: Capsule Net and ResNets Lecture W: Autoencoders	Goodfellow Ch. 20
6	Lecture M: GANs Lecture W: Intro to NLP	Jurafsky Ch. 6-10
7	Lecture M: RNNs / LSTMs Lecture W: Generative RNNs	Goodfellow Ch. 10 Unreasonable Effectiveness of RNNs
8	Spring Break	
9	Lecture M: Reinforcement Learning Lecture W: Attention	Seminal attention paper: translation Image captioning Generating parse trees Natural language understanding
10	Lecture M: Advanced Optimization Lecture W: Unsupervised Learning	
11	Neuroscience	
12	Causality	
13-15	Special Topics	
16	Project Presentations	