ECSE 4850/6850

Introduction to Deep Learning

Spring, 2020

Instructor: Dr. Qiang Ji,
Email: jiq@rpi.edu
Phone: 276-6440
Office: JEC 7004
Meeting Hours & Place : 2:00-3:20 pm, Mondays and Thursdays, CARNEG 113.
Office Hours: 3:00-4:00 pm Wednesdays or by Appointment
TAs: Gourav Saha (sahag@rpi.edu) and Ziyu Su (suz4@rpi.edu)

Lecture notes: Available on RPI Learning Management System

This course introduces fundamentals in deep learning and demonstrates its applications in computer vision. It covers both deterministic and probabilistic deep models. For deterministic deep models, the course covers discriminative models, including Deep Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks; generative models, including Generative Adversarial Networks and the auto-encoders; and deep reinforcement learning. Probabilistic deep models include Bayesian Neural Networks, Deep Boltzmann Machine, Deep Belief Networks, and Deep Bayesian Networks. The course is self-contained. It starts with an introduction of the background needed for learning deep models, including probability, linear algebra, standard classification and optimization techniques. To demonstrate various deep models, we will apply them to different computer vision tasks.

Prerequisites

This is a senior and graduate level course. Students are required to have basic knowledge linear algebra and optimization. Strong programming skills in one of the high level languages such as C++, Matlab, or Python are required. Prior courses in machine learning/ pattern recognition and computer vision/image processing are preferred but not required.

Optional textbooks

Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016, http://www.deeplearningbook.org (download the book quickly for free!)

Linear Algebra and Learning from Data, Gilbert Strang, Wellesley-Cambridge Press , 2018. http://math.mit.edu/~gs/learningfromdata/

Software

Many of the assignments and projects will be implemented in Google's TensorFlow (https://www.tensorflow.org/tutorials/), which is implemented in Python. Mastery of Python or being able to quickly learn it is required.

Course Evaluation

The course will involve homework assignments (15%), class projects (40%), a mid-term exam (25%), and a final project (20%). All exams are open book and comprehensive.