Deep Learning

IE 534/CS 547
Fall 2019

Professor: Justin Sirignano
Teaching Assistants: Yuanyi Zhong, Xiaobo Dong, Lei Fan, Rachneet Kaur, Jyoti Aneja, Peijun Xiao
Website: https://courses.engr.illinois.edu/ie534/fa2019/

Overview:

This course is an introduction to deep learning. Topics include convolution neural networks, recurrent neural networks, and deep reinforcement learning. The course will use PyTorch to train models on GPUs.

Deep learning is computationally intensive. This course is supported by a computational grant for **100,000 GPU node hours.** This provides a unique opportunity for students to develop sophisticated deep learning models.

Grading:

35% Homeworks
35% Midterm (**November 7, 8:00 AM**)
30% Final Project

The 2 lowest homework grades are dropped. Late homeworks are **not** accepted.

Prerequisites:

CS 446 (or equivalent). Python. Basic statistics, probability, and optimization. Basic knowledge of Bash/Linux is recommended.

PyTorch and Linux/Bash:

Lectures and tutorials will cover PyTorch and Linux/Bash. Example code will be provided to students. The OpenAI Gym environment for deep reinforcement learning will also be reviewed.

Topics:

- Fully-connected and feedforward networks
- Convolution networks
- Backpropagation

- Stochastic Gradient Descent
- Hyperparameter selection and parameter initialization
- Optimization algorithms (RMSprop, ADAM, momentum, etc.)
- Second-order optimization (e.g., Hessian-free optimization)
- TensorFlow, PyTorch, automatic differentiation, static versus dynamic graphs, define-by-run
- Regularization (L2 penalty, dropout, ensembles, data augmentation techniques)
- Batch normalization
- Residual neural networks
- Recurrent neural networks (LSTM and GRU networks)
- Video recognition (two-stream convolution network, 3D convolution networks, convolution networks combined with LSTM, optical flow)
- Generative Adversarial Networks
- Deep reinforcement learning (Q-learning, actor-critic, policy gradient, experience replay, double Q-learning, deep bootstrap networks, generalized advantage estimation, dueling network, continuous control, Atari games, AlphaGo)
- Distributed training of deep learning models (e.g., asynchronous stochastic gradient descent)
- Theory of deep learning (universal approximation theorem, convergence rate, and recent mathematical results)
- Convergence analysis of stochastic gradient descent, policy gradient, tabular Q-learning

Homeworks:

See course website for a list of the homeworks.

Reading:

A list of journal and conference papers will be provided to the class.
Textbook: “Deep Learning” by Goodfellow, Bengio, and Courville, MIT Press, 2017.

Office Hours:

Tuesday 9:30 -10:30 AM at CSL 126 (JS)
Friday 2:00-3:00 PM at Transportation Building 304 (RK)
Monday 4:00-5:00 PM at TBA (JA)

Final Projects:

Students must select one the the following projects:

(1) Train Densely Connected Convolutional Networks on CIFAR100 and Tiny ImageNet datasets. Compare results with and without pre-training. Compare results to residual

networks. See the original paper “Densely Connected Convolutional Networks”, CVPR, 2017 by Huang et al.

(2) Implement “Distilling the Knowledge in a Neural Network” by Hinton, Vinyals, and Dean.

(3) Scene/image description as in “Show and Tell: A Neural Image Caption Generator”, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015 by Vinyals et al.

(4) Implement “DeepFace: Closing the Gap to Human-level Performance in Face Verification”, Proceedings of the IEEE conference on computer vision and pattern recognition, 2014 by Taigman et al.

(5) Implement “Very Deep Convolutional Networks for Text Classification” by Conneau, Schwenk, Barrault, and Lecun (2016) and compare against LSTM networks.

(6) Implement one of the following deep reinforcement learning papers for Atari Games in PyTorch: (A) “Deep reinforcement learning with double q-learning”, AAAI, 2016 by Hasselt et al., (B) “Deep exploration via bootstrapped DQN”, NIPS, 2016 by Osband et al., or (C) “Dueling network architectures for deep reinforcement learning”, 2015 by Wang et al. *This is a challenging project.*

(7) Implement “Faster R-CNN: Towards real-time object detection with regional proposal networks”, NIPS, 2015 by Ren et al. in PyTorch. *This is a challenging project.*

(8) Implement Fast Fourier Transforms (FFT) for convolutions for training convolution networks. See “Fast Training of Convolutional Networks through FFTs” by Mathieu, Henaff, and LeCun (2013) and “Fast Convolutional Nets with fbfft: A GPU Performance Evaluation” by LeCun et al. (2015). *This is a challenging project.*

Project teams should be 4-6 students.