Deep Learning

CS 537 / IE 534
Instructor: Justin Sirignano

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 TensorFlow to train models on GPUs.

Deep learning is computationally intensive. This course is supported by a computational grant for **50,000 GPU node hours.** This provides a unique opportunity for students to develop sophisticated deep learning models.

Grading:

30% Homeworks
30% Midterm
40% Final Project

Homeworks:

HW #1: Implement a single layer neural network for MNIST from scratch in Python. Implement a single layer neural network with TensorFlow for MNIST.

HW #2: Implement a convolution neural network for MNIST and CIFAR10 from scratch in Python. Implement a deep convolution neural network with TensorFlow for CIFAR10.

HW #3: Implement a deep residual network for CIFAR100 using TensorFlow.

HW #4: Generative adversarial networks. See https://github.com/logan-courtney/HW4_GAN for 2017 version.

HW #5: Implement a deep convolutional neural network to label videos. The model will be pre-trained on ImageNet. See https://github.com/logan-courtney/HW5_Action_Recognition for 2017 version.

HW #6: Deep reinforcement learning for Atari video game. See https://github.com/tgangwani/IE598_RL/tree/master/hw6 for 2017 version.

HW #7: Deep reinforcement learning for continuous control. See https://github.com/tgangwani/IE598_RL/tree/master/hw7 for 2017 version.

TensorFlow, PyTorch, Linux/Bash, and MPI:

Lectures and tutorials will cover TensorFlow, PyTorch, Linux/Bash, and MPI. Example code will be provided to students. Students will learn how to use Distributed TensorFlow and Distributed PyTorch for parallelizing training of deep learning models across multiple GPU nodes. The OpenAI Gym environment for deep reinforcement learning will also be reviewed.

Topics:

- 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
- Deep residual neural networks
- Dense 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 (asynchronous stochastic gradient descent, parameter servers, synchronous stochastic gradient descent, asynchronous advantage actor-critic, MPI All-Reduce, Ring All-Reduce)
- Theory of deep learning (universal approximation theorem, convergence rate, and recent mathematical results)
- Convergence analysis of stochastic gradient descent, policy gradient, tabular Q-learning

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.

Examples of Final Projects in 2016, 2017:

Driverless car, video recognition, text-to-image generation, automatic labeling and description of images, earthquake simulation analysis, laser guidance (LIDAR), deep reinforcement learning.