CS535: Deep Learning

Winter 2018

Fuxin Li

Course Information

- Instructor: Dr. Fuxin Li
  - KEC 2077, lif@eecs.oregonstate.edu
- TA:
  - Xinyao Wang: wangxiny@oregonstate.edu
- My office hour: TBD (vote)
- Class Webpage:

http://classes.engr.oregonstate.edu/eecs/winter2018/cs535/

- Questions/Discussions – on CANVAS

Prerequisites

- Significant knowledge on machine learning, especially the generics (not specific algorithms)
  - CS 534 or equivalent knowledge
  - Refresher will be provided in the next lecture
- Some knowledge of numerical optimization
  - 1.5 weeks will be devoted to optimization and also deep network optimization

Grading

- Initial quiz (5%) based on participation only
- 3 Assignments (30%)
  - Late assignments only on programming assignments (25% penalty for 2 days)
  - Must write your own code!
- Quizzes (2 more quizzes totaling 20%)
  - Based on whether you answer the questions correctly
- Final Project (45%)
  - Final project is to be done with teams not more than 3 participants
  - Grading will be done according to:
  - Initial proposal (10%)
  - Final oral presentation (10%)
  - Final written presentation (25%)

Materials

- Book:
  - I. Goodfellow, A. Courville, Y. Bengio. Deep Learning. MIT Press 2016.
  - Electronic version: http://www.deeplearningbook.org/
- More readings can be found at:
  - http://deeplearning.net/reading-list/
  - http://colah.github.io/
  - http://karpathy.github.io/
  - https://www.coursera.org/course/neuralnets

Toolboxes

- A plethora of deep learning toolboxes around:
  - Caffe
  - Theano
  - Torch, pyTorch
  - TensorFlow
  - CNTK, MXNet, Lasagne, Keras, Neon, etc.
- Toolbox policy:
  - We stick to pyTorch for assignments
  - Final project: select the one you are most comfortable with

Outcome

- Understand the concepts of deep learning
- Gain some intuitions on deep networks
- Understand the training of deep learning
- Be able to use at least one deep learning toolbox to design and train a deep network
- Be able to design new algorithms and new architectures

What will be covered

- Basic neural network structure
- Training tricks (SGD, Momentum etc.)
- CNNs
- LSTMs
- Unsupervised neural networks
- Neural reinforcement learning (Dead week)

Final Project

- Groups of no more than 3 persons
- Jointly work on a significant project
  - Must use deep learning
  - CANNOT be just running an already-trained classifier on some images
  - Try to solve a real problem
  - One can elect projects from paper readings
  - I will try to suggest some standard projects
  - New neural architectures/changes to current architectures are welcome
- Grading – based on the project merit, execution and presentation

Project Presentations

- 2 presentations for the final project
  - Initial design (at least 1 month before finals week)
    - Talk about what is your project about
    - What you plan to do
    - Re-grouping if several people are thinking about similar projects
  - Final presentation (finals week)
    - Need to identify who did what in the team
- 8 minutes per presentation
  - Slides uploaded to a common computer

Computing Resources

- Pelican cluster:
  - 4 nodes with 2 GTX 980 Ti (6GB) each
  - Accessible by SSH at pelican.eecs.oregonstate.edu
  - Policy: 1 GPU per group otherwise risk your jobs be killed
- If you want to buy your own:
  - Website will link you to a good article
  - GTX Titan V ($3,000!), GTX Titan X PASCAL, GTX 1080 Ti (Mar 2017), GTX 1080, GTX 1070 Ti ($450), GTX 1070, GTX 1060 (sorted descendingly by price)

Approximate schedule (will be on website)

- Week 1 (Jan. 8 - 12)
  - 1. Admin + General Introduction + Machine Learning Refresher
  - 2. Optimization Primer #1 (nonconvex optimization, stationary points and saddle points, optima, gradients) + Basic 1 Hidden Layer Neural Network (backpropagation)
- Week 2 (Jan. 15 - 19): Standard neural networks (MLK day break)
  - 3. Neural Network Optimization
- Week 3 (Jan. 22 - 26): Convolutional Networks
  - 5. Theoretical Implications + Convolutional Neural Networks (mostly in computer vision)
  - 6. Continued CNN, Visualization of CNN
- Week 4 (Jan. 29 – Feb. 2): Temporal Neural Models
  - 7. Temporal Neural Models (RNNs and LSTMs)
  - 8. Continued Temporal Neural Models (LSTMs, GRUs, stacked together with CNNs)
- Week 5 (Feb. 5 – Feb. 9): Deciding what project to work on
  - 9. Introduction of deep learning toolboxes (Caffe, Keras, automatic gradients)
  - 10. An overview of other neural models
- Week 6 (Feb. 12 - 16): Project proposals
  - 11. Project Proposals
  - 12. Neural Network Optimization (stochastic mini-batch gradient descent, momentum, dropout, learning rate and weight decay)

Approximate schedule

- Week 7 (Feb. 19 - 23): Neural Network Optimization, Unsupervised Approaches
  - 13. Neural Network Optimization (stochastic mini-batch gradient descent, momentum, dropout, learning rate and weight decay, automatic step-size methods)
  - 14. Unsupervised Deep Learning (Autoencoders and variational autoencoders)
- Week 8 (Feb. 26 – Mar. 2): Unsupervised Approaches
  - 15. Unsupervised Deep Learning II (GANs)
  - 16. ResNet and New Architectures
- Week 9 (Mar. 5 - Mar. 9): Deep Learning Applications
  - 17. More applications
  - 18. Deep Learning in Natural Language Processing (Guest lecture from the Algorithms for Computational Linguistics group)
- Week 10 (Mar. 12 - Mar. 16): Deep Reinforcement Learning
  - 19. Deep reinforcement learning (guest lecture by Alan Fern)
  - 20. Project Presentations
- Week 11 (Mar. 19 - Mar. 23): Finals Week
  - 21. Project Presentations