EE599: Deep Learning for Engineers (Spring 2020)

USC Viterbi School of Engineering

Units: 4

Instructor: Keith M. Chugg
Office: EEB 500A (5th floor main office)
Phone: 213-740-7294 (Voice)
email: chugg@usc.edu – Include 599 in subject
Office Hours: Monday, Wednesday, 3:00-4:00, 7:00-8:00

TA: Olaoluwa (Oliver) Adigun
Office: EEB 420r
Phone: 213-740-2356
email: adigun@usc.edu – Include 599 in subject
Office Hours: Tuesday, and Thursday, 5:00-6:30

TA: Jiali Duan
Office: EEB B10
Phone: 213-204-0380
email: jialidua@usc.edu – Include 599 in subject
Office Hours: Friday, 4:00-5:00

TA: Kuan-Wen (James) Huang
Office: EEB 522
Phone: 213-390-0196
email: kuanwenh@usc.edu – Include 599 in subject
Office Hours: Thursday 1:00-3:00 PM

TA: Arnab Sanyal
Office: PHE 320
Phone: 213-608-2705
email: arnabsan@usc.edu – Include 599 in subject
Office Hours: Monday, 10:00 AM-12:00 PM

Lecture: Monday, Wednesday, 5:00-6:50 PM in THH 301

Discussions: Thursday, 4:00-4:50 PM in SLH 102, ZHS 252, VKC 151 (students will be assigned a session in Piazza)

Webpages: Piazza Class Page for all handouts and discussion
Campus Class Page for homework submission and grades – this is not yet set-up and URL may change

– *Student has the responsibility to stay current with webpage material*

Prereqs: EE510, EE503

Other Requirements:
Undergraduate ECE background (e.g., Signals and Systems, Fourier Transforms, filtering, etc.)
Python or Matlab proficiency
(if Matlab proficiency, a willingness to learn Python through problems)

Grading: 20% Homework
25% Quizzes
25% Midterm Exam
30% Final Project (including proposal, presentation, report)

Course Description: Deep learning is arguably the most transformation technology of the 21st century thus far – this method enables reliable speech recognition, face recognition, internet search, computer vision, and self-driving cars. The foundations of deep learning have been part of the electrical engineering graduate curriculum for years with neural networks being a popular research topic in the 1980s and 1990s. What changed in the early 2000s was that the amount of data and computational power available increased to a point whereby larger, more complex models could be trained. In the last decade, deep learning has surpassed traditional EE methods for inference in many applications in which accurate models prove evasive, but data is plentiful.

The CS community has embraced deep learning and created many tools that bring many of the most powerful deep learning techniques to relative beginners. In this class, we place deep learning in the context of other EE graduate classes and will develop skills at training neural networks using pyhon-based packages and cloud computing resources.

Expectations for Student Effort and Conduct:

- **Attendance:** Lecture attendance is encouraged but not mandatory. However, students are responsible for all material presented in lecture.
- Conduct yourself professionally. You will soon be entering the workforce and professional conduct is a requirement there. This means treating others (classmates, TAs, instructors) with respect and using professional language in all written and oral communications.
- **Students are expected to show initiative and be active in their own learning experience. This means that not every topic in the class will be described in detail in lecture. This is especially true for the programming/computing components of the class. Use the internet as a resource to learn more and figure out issues for yourself. This is a valuable skill coveted by potential employers.**
- Participate in the learning experience with your classmates. There will likely be a distribution in skills and experience among the students. Please add to the class by helping out your classmates in areas that you are strong – e.g., posting helpful replies and resources on piazza.
- **Note on e-mail vs. Piazza:** If you have a question about the material or logistics of the class and wish to ask it electronically, please post it on the piazza page (not e-mail). You may post it anonymously if you wish. Often times, if one student has a question/comment, other also have a similar question/comment. Use e-mail with the professors and TAs only for issues that are specific to your individually (e.g., a scheduling issue or grade issue).
- The discussion material is determined by the students. Please provide suggestions and questions to the TAs beforehand. If you show up to discussion and ask no questions, the TAs will present summaries of the material and work example problems. The TAs will not work the HW problems for you in discussion, although the TAs will provide hints/help.

Learning Objectives: Upon successful completion of this course a student will

- Understand the basics of classical estimation and detection approaches used in electrical engineering, including the associated limitations.

- Understand the basics traditional regression methods
- Understand the basics of adaptive filtering and stochastic gradient methods
- Understand the different types of machine learning and when deep learning approaches are most suitable
- Be able to generate data-sets and work with on-line public data-sets using standard Python tools
- Be able to perform numerical simulations and experiments in Python (e.g., filters, LMS, regression, MLPs)
- Understand feedforward neural networks and concepts of trainable parameters and hyper-parameters
- Be familiar with the most commonly used activation functions, losses, regularizers and optimizers and understand what choices are reasonable for a given problem
- Be able to derive back-propagation equations for a novel loss and activation.
- Understand convolutional neural networks for one-dimensional and multi-dimensional data-sets
- Understand recurrent neural networks, especially GRUs and LSTMs
- Be able to train feedforward (MLP), convolutional, and recurrent networks for novel applications using the Keras library with Tensorflow backend (Python) or TensorFlow.Keras.
- Use Amazon Web Services for training a neural network using GPU acceleration.
- Have a basic understanding of the speech recognition and image classification and why the most popular architectures are used for those applications
- Propose and demonstrate a novel deep learning application project

Important Dates:

- **Midterm Exam:** Wednesday, March 11, 5:00 - 6:50 PM
- **Final Project Presentation:** Wednesday, May 6, 4:30-6:30 PM set by the university

Grading Policies:

- **Final grades** will be assigned by a combination of student score distribution (curve) and the discretion of the instructors. Final grades are nonnegotiable. If you cannot accept this condition, you should not enroll in this course.
- **Homework Policy**
  - **Late HW** will not be accepted. A late assignment results in a zero grade.
  - Homework will be assigned and collected approximately every 1.5 weeks.
  - Show your work in your homework solution; the correct answer alone is worth only partial credit.
  - Source code is required for computer assignments.
  - Homework collaboration is encouraged. This is discussing problems and solution strategies with your classmates, the TA, and/or the instructors and is to be distinguished from copying solutions of others which is prohibited.
  - You cannot copy code from a classmate or the internet. If you are using some code from the internet as a reference point, include a note in your code comments which includes the URL.
- **Exam and Quiz Policy**

 - **Make-up Exams/quizzes:** No make-up exams/quizzes will be given. If you cannot make the above dates due to a class schedule conflict, you must notify Prof. Chugg by the last day to add/drop. If we cannot accommodate your schedule, you must drop the class. In the case of a medical emergency, a signed letter from your doctor is required. This letter must include the telephone number of your doctor.
 - All exams/quizzes are cumulative, but with an emphasis on material presented since the last exam.
 - Two quizzes are planned for the semester – one before the midterm and one after the midterm. These quizzes will be announced at least 1 week ahead of time. Quiz duration will be approximately 60 mins.

- **Final Projects (TBR)**
 - Final project teams will be 2-4 students each. Ph.D. will be allowed to do individual projects if they prefer.
 - Topics can be suggested by the students or taken from a list of suggested topics to be provided.
 - Each team will be assigned a mentor. Mentors will be typically be the instructor or a TA, but in some cases may be Ph.D. students or faculty associated with the topic.
 - Each team will produce the following with the percentage of teh overall prject grade shown:
 * Preliminary proposal: 5%
 * Revised proposal (after mentor feedback): 15%
 * Project presentation: 30%
 * Project report: 40%
 * Project video (approximately 5 minute video posted to YouTube): 10%
 - Example project proposals and reports will be provided.
 - Final presentations will take place at an end-of-semester "Deep Learning Symposium" – e.g., 2019 EE599 Deep Learning Symposium
 - Your videos will be posted in a playlist on the DeepLearning USC YouTube channel – e.g., 2019 EE599 YT Playlist

Textbooks: There is no required textbook for this class. The lecture slides will be based on a number of textbooks and other reference materials. The following is a list of books, some of which are legally available for free on the internet. They are ordered roughly by how central they are to the material covered.

- **[GBC]** Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, The MIT Press, 2016.
- **[Nielsen]** Michael Nielsen, Neural Networks and Deep Learning, on-line.
- **[Haykin]** Simon Haykin, Neural Networks And Learning Machines 3rd Edition, Pearson, 2009.
- **[Chollet]** Francois Chollet, Deep Learning with Python, Manning, 2018.
- **[Aggarwal]** Charu C. Aggarwal, Neural Networks and Deep Learning, A Textbook, Springer International Publishing, 2018. PDF is available online from usc.edu domain.
- **[HDBJ]** Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale, Orlando De Jesús, Neural Network Design, 2nd Edition.
- **[Bishop]** Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2016.
- **[Murphy]** Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, 2012.
- **[AML]** Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Yien Lin, Learning from Data, A Short Course, AMLbook.com.
- **[Haykin-AF]** Simon Haykin, Adaptive Filter Theory (4th Edition), Pearson, 2002.
- **[Scholtz]** Robert A. Scholtz, Supplemental Notes on Random Processes, 2013 (provided via piazza).

Course Outline (Non-Computational Material)

1. Course Introduction
   (a) Teaching team introduction, syllabus review, and tools overview.
   (b) Various types of Machine Learning problems and approaches.
   (c) What is Deep Learning and how does it relate to electrical engineering?
2. Estimation and Detection with Statistical Models/Descriptions
   (a) MMSE estimation, MAP estimation/detection, hard vs soft decisions.
   (b) Sufficient statistics
   (c) Karhunen-Loève decomposition
   (d) Ideal features
   (e) Parameter estimation
3. Regression and Classification from Data
   (a) simple linear regression
   (b) linear classifier
   (c) logistical regression
   (d) over/under fitting and regularization
4. Optimization via Steepest descent
   (a) Gradient descent
   (b) LMS algorithm
   (c) Stochastic gradient descent with mini-batches
5. Multi-Layer Perceptrons (Feed-forward Neural Networks)
   (a) Common loss functions and regularizers
   (b) Common activation functions
   (c) Back-propagation
   (d) Trainable parameters vs hyper-parameters
   (e) Universal approximation
   (f) Application to automatic speech recognition (ASR)
6. **MIDTERM approximately here**
7. Variations on SGD Back-Propagation Learning
   (a) Initialization and vanishing gradient
   (b) Momentum and Optimizers (e.g., Adam)
   (c) Drop-out
   (d) Batch Normalization
   (e) Hyper-parameter optimization
8. Working with Data
   (a) Collection, contamination, and cleaning
   (b) Data augmentation

(c) Synthetic data generation
(d) Designing features and dimensionality reduction (e.g., PCA, LDA)
(e) Best practices for deep learning design flow

9. Convolutional Neural Networks (CNNs)
   (a) 1D CNNs
   (b) 2D CNNs
   (c) Pooling and kernel selection
   (d) Some commonly used image classification architectures
   (e) Complexity reduction methods – e.g., separable filters
   (f) Applications in computer vision.
   (g) Generative Adversarial Networks (GANs)
10. Recurrent Neural Networks
   (a) RNNs
   (b) General "gating" and "filtering" concepts in RNNs
   (c) Gated Recurrent Units (GRUs) networks
   (d) Long-Term/Short-Term Memory (LSTM) networks
   (e) Backprop Through Time (BPTT)
   (f) Recurrent networks for nonlinear filtering
   (g) Recent advances (Transformers, BERT)
11. Additional Topics (time permitting)
   (a) Auto-encoders and generative models
   (b) Generative Adversarial Networks (GANs)
   (c) Deep Reinforcement Learning

Course Outline (Computing and Numerical Material)

1. Setting up Python
   (a) Anaconda
   (b) pyenv
   (c) editors, IDEs, Jupyter Notebooks and qtConsole.
2. Getting Started with Python
   (a) numpy and scipy
   (b) plotting
   (c) generating and processing signals
   (d) working with vectors, matrices, and tensors.
   (e) working with data
3. Training a simple MLP using numpy.
   (a) implementing back-propagation
   (b) mini-batch SGD
   (c) learning curves
4. Getting started with Keras
   (a) simple MLP training and inference
   (b) CNNs
   (c) RNNs and conversion to streaming inference.
5. Unix basics
   (a) moving around and copying files using scp, rsynch, sftp
   (b) installing packages
   (c) setting up and running Python through the CLI.
6. Getting started with AWS
   (a) setting up and shutting down an EC2 instance
   (b) managing and monitoring costs
   (c) setting up a GPU instance for training.
7. Other deep learning frameworks (time permitting and projects)
   (a) PyTorch
   (b) Keras with Tensorflow backend vs. tensorflow.keras
   (c) TensorFlow

Statement on Academic Conduct and Support Systems

Academic Conduct:

Plagiarism: presenting someone else's ideas as your own, either verbatim or recast in your own words is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Part B, Section 11, "Behavior Violating University Standards" policy.usc.edu/scampus-part-b. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct.

Support Systems:

Student Health Counseling Services – (213) 740-7711, 24/7 on call. Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention.

National Suicide Prevention Lifeline – 1 (800) 273-8255, 24/7 on call suicidepreventionlifeline.org Free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week.

Relationship and Sexual Violence Prevention Services (RSVP) – (213) 740-4900, 24/7 on call Free and confidential therapy services, workshops, and training for situations related to gender-based harm.

Office of Equity and Diversity (OED) and Title IX – (213) 740-5086 Information about how to get help or help a survivor of harassment or discrimination, rights of protected classes, reporting options, and additional resources for students, faculty, staff, visitors, and applicants. The university prohibits discrimination or harassment based on the following protected characteristics: race, color, national origin, ancestry, religion, sex, gender, gender identity, gender expression, sexual orientation, age, physical disability, medical condition, mental disability, marital status, pregnancy, veteran status, genetic information, and any other characteristic which may be specified in applicable laws and governmental regulations.

Bias Assessment Response and Support – (213) 740-2421 Avenue to report incidents of bias, hate crimes, and microaggressions for appropriate investigation and response.

The Office of Disability Services and Programs – (213) 740-0776 Support and accommodations for students with disabilities. Services include assistance in providing readers/notetakers/interpreters, special accommodations for test taking needs, assistance with architectural barriers, assistive technology, and support for individual needs.

USC Support and Advocacy – (213) 821-4710 Assists students and families in resolving complex personal, financial, and academic issues adversely affecting their success as a student.

Diversity at USC – (213) 740-2101 Information on events, programs and training, the Provost's Diversity and Inclusion Council, Diversity Liaisons for each academic school, chronology, participation, and various resources for students.

USC Emergency - UPC: (213) 740-4321, HSC: (323) 442-1000, 24/7 on call Emergency assistance and avenue to report a crime. Latest updates regarding safety, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible.

USC Department of Public Safety - UPC: (213) 740-6000, HSC: (323) 442-120, 24/7 on call Non-emergency assistance or information.