SYLLABUS (07 Jan 2022)

A Computational Introduction to Deep Learning

EE 541: Spring 2022 (2 units)

Machine learning using large datasets stands as one of the most transformative technologies of the 21st century. It enables reliable face and speech recognition, internet search and monetization, computer vision, and self-driving vehicles. Machine learning proficiency requires software skills as well as an understanding of the underlying mathematics and theoretical concepts. This class introduces important aspects of deep learning using a computation-first approach. It emphasizes using frameworks to solve reasonably well-defined machine learning problems. Two advanced courses provide a deeper study of mathematical concepts: EE 559 Machine Learning I: Supervised Methods and EE 641 Deep Learning Systems.

Instructor:	Brandon Franzke	
Email:	franzke@usc.edu	
Office:	EEB 504B	
Zoom:	meet:	998 5176 5591
	code:	574987
Hours:	Wednesday:	14:00 – 15:30
	Thursday:	10:00 – 12:00 (remote)

Lecture

Monday (section: 31249)
17:30 – 19:20

Discussion

TBA (remote)

Piazza

https://piazza.com/usc/spring2022/ee541

Piazza enables fast and efficient help from classmates and instructors. Use Piazza to post questions about course material, homeworks, and policies instead of emailing questions to the teaching staff.

Canvas

https://canvas.usc-ece.com

Use Canvas to electronically submit your homework and view course grades. You will receive an email to register during the first week of classes. Contact Dr. Franzke with any technical issues.

Autolab

https://autolab.usc-ece.com

Use Autolab to electronically submit programming portions of homework for "auto-grading". You will receive an email to register during the first weeks of the course. Contact Dr. Franzke with technical issues.

TAs and staff

TA:	Tianchen Yuan	TA:	Ganning Zhao
Zoom:	meet: TBA	Zoom:	meet: TBA
	code:		code:
Hours:	Thursday: 09:00 - 11:00	Hours:	Friday: 09:00 - 11:00
Email:	tianchey@usc.edu	Email:	ganningz@usc.edu
CP:	Aditya Anulekh Mantri	Grader:	Sarthak Maharana
Hours:	by appointment	Hours:	by appointment
E-mail:	adityaan@usc.edu	E-mail:	maharana@usc.edu

Recommended preparation

Graduate standing with a typical undergraduate ECE background including the equivalent of EE 105 and EE 155. Some exposure to a high-level scripting language such as Python or Matlab.

Learning objectives

Upon completion of this course a student will be able to:

- Understand the fundamentals and implement linear regression, a linear classifier, and logistic regression.
- Apply common deep architectures such as multilayer perceptron, convolutional, and recurrent neural networks and understand where each is most applicable.
- Be proficient in Python programming, including loops, conditionals, lists, dictionaries, classes, and standard programmatic patterns.
- Be proficient in numerical Python using NumPy, SciPy, and matplotlib for design and analysis.
- Organize, store, and access datasets such as .npz files, .h5 files, pickle, and pandas.
- Understand the role of machine learning frameworks such as scikit-learn.
- Understand the role and use of deep-learning frameworks such as PyTorch.
- Use frameworks to train MLP, convolutional, and recurrent networks to solve machine learning problems.
- Be proficient with relevant computing and cloud computing resources such as:
  - linux command line interface and automation with shell scripts
  - version control software such as git
  - cloud GPUs to train deep networks and understand fundamentals of the Nvidia CUDA SDK.

Course materials

- “*Neural Networks and Deep Learning*”, Michael Nielson. (online: http://neuralnetworksanddeeplearning.com).
- “*Deep Learning with PyTorch*”, Eli Stevens, Luca Antiga, Thomas Viehmann, Manning, 2020. (online: https://pytorch.org/assets/deep-learning/Deep-Learning-with-PyTorch.pdf).
- “*Deep Learning*”, Ian Goodfellow, Yoshua Bengio, Aaron Courville, The MIT Press, 2016. (online: http://www.deeplearningbook.org).
- “*Python Programming And Numerical Methods: A Guide For Engineers And Scientists*”, Qingkai Kong, Timmy Siauw, Alexandre Bayen, Elsevier, 2020. (online: https://pythonnumericalmethods.berkeley.edu/notebooks/Index.html).

Note: The texts are secondary to in-class lecture material and homework sets.

Course Outline (tentative)

	Topics/Daily Activities	Reading & Homework	Deliverables
Week 1 (10 Jan)	Machine Learning inventory. Configuring your Python environment.	HW 1 assigned.	
(17 Jan)	**No class, Martin Luther King Day, University holiday.**		
Week 2 (24 Jan)	Getting started with Python. Estimation and MMSE.	Supp: [4] Ch. 1. HW 2 assigned.	HW 1 due.
Week 3 (31 Jan)	Numerical Python. Estimation and regression.	lecture slides, supp: [4] Ch. 2-7.	
Week 4 (07 Feb)	Decision theory, logistic regression, multilayer perceptrons (MLPs)	lecture slides, supp: [4] Ch. 14-16, 25. HW 3 assigned.	HW 2 due.
Week 5 (14 Feb)	MLP backpropagation (scalar detail)	supp: [1] Ch. 1, [2] Ch. 6, [3] Ch. 6.	**Quiz 1** (week 1-4).
(21 Feb)	**No class, President's Day, University holiday.**		
Week 6 (28 Feb)	MLP backpropagation (scalar, vector/tensor overview). Introduction to data engineering	lecture slides. HW 4 assigned.	HW 3 due.
Week 7 (07 Mar)	PyTorch: Building and Training MLPs, Optimizing training	supp: [3] Ch. 7-8, [1] Ch. 3.	
(14 Mar)	**No class, Spring break, University holiday.**		
Week 8 (21 Mar)	PyTorch: Optimizing training	HW 5 assigned.	Preliminary proposal due. HW 4 due.
Week 9 (28 Mar)	Advanced data engineering, embedding and auto-encoding	lecture notes.	**Quiz 2** (week 5-8). HW 5 due.
Week 10 (04 Apr)	PyTorch: Convolutional Neural Networks (CNN)	Supp: [2] Ch. 8, [3] Ch. 9. HW 6 assigned.	Revised proposal due
Week 11 (11 Apr)	Introduction to linux CLI and AWS	lecture notes.	
Week 12 (18 Apr)	PyTorch: Recurrent Neural Networks (RNN)	HW 7 assigned.	Supp: [3], Ch. 10. HW 6 due.
Week 13 (25 Apr)	Generative adversarial networks (GAN)	Supp: [3], Ch. 20.	**Quiz 3** (week 9-12). HW 7 due.
(11 May)	**Project reports and videos due**		

Attendance and Participation

Attendance is mandatory to all lectures and discussions. You are responsible for missed announcements or changes to the course schedule or assignments. Taping or recording lectures or discussions is strictly forbidden.

Grading Procedure

Homework

Homework is assigned every 1-2 weeks. Assignments will include computational and numerical programming problems and will encourage experimentation and curiosity. No late submissions for credit.

Your total homework score sums your best homework scores (as a percentage) after removing the one lowest score (of minimum 50%). You may discuss homework problems with classmates but each student must do their own original work. Cheating warrants an F in the course. Turning in identical homework establishes a rebuttable presumption of cheating.

Quizzes

Quizzes are short (45 minute) non-cumulative tests that cover the most recent material (approximately 5-weeks). Quizzes highlight important concepts and methods. They test ability to apply major principles, demonstrate conceptual understanding, and may require writing snippets of Python code. They occur during weeks 5, 9, and 13 (tentative). You may use a single 8.5"x11" reference sheet (front and back OK). You may not use any additional resources. You are expected to bring a scientific (non-graphing) calculator. Any cheating may result in an "F" in the course and will be referred to Student Affairs for other penalties. Alternate arrangements will be considered only for valid medical or family emergency excuses (proof required).

Final Project

This course culminates with a final project in lieu of a final exam. Teams of two students will design and implement a deep learning athat connects two or more independent asynchronous components (often "frontend" and "backend"). Treat the final project as a multi-week in-depth homework assignment that integrates concepts from the entire semester. Problems will include a representative dataset to get you started. All projects must obtain the instructor's written approval. You must experiment and document network architecture search, hyper-parameter optimization, and dataset augmentation.

You will submit two deliverables: (1) a properly referenced written report that includes all source code as well as links to any external code sources and (2) a short video describing your findings, methods, and analyses.

Requirements

Project topics must include sufficient scope and apply course knowledge to a useful end. The project must compose at least two distinct units that operate and act independently but provide greater function when acting together. The project must demonstrate comprehensive understanding of the entire development stack and the product lifecycle from idea to deployment to maintenance. You may use whatever computer language you like but deviations from Python, C++, Node.js, GoLang, or other frameworks used in class require prior instructor approval.

Grading and Milestones

Topic proposal	week 10	10%
Status report	week 13	10%
Final report	final	65%
Video		10%
Final model / Source code		5%

Deliverables and demo

Written project report: the project report should summarize the topic, provide relevant background (theoretical or applied), timeline and contributions, and document challenges and extensions. It should provide discussion sufficient that an uninformed expert could understand the logic, algorithmic decisions, and implementations. Teams should provide quantifiable metrics to justify engineering tradeoffs.

Video: a 3-minute video that describes the problem, your design, and implementation. You may choose to upload this to a video sharing site such as YouTube but that is not required. All team members must participate equally.

Source code: submitted to instructor by providing link to pull from github.

Example projects

Semantic classification of images of faces Classify emotion from facial expressions – e.g., sad, happy, bored, neutral.

Clothing coordination Design a CNN to match shoes, pants, shirts, accessories, etc.

Language classification Design an RNN to perform real-time audio language classification – e.g., Hindi, Mandarin, and English.

Course Grade

Homework	45%	A	if 90 – 100 points
Quizzes	30%	B	if 80 – 89 points
Final Project	25%	C	if 70 – 79 points
		D	if 60 – 69 points
		F	if 0 – 59 points

("+" and "–" within approx. 3% of grade boundary)

Cheating

Cheating is not tolerated on homework or exams. Penalty ranges from F on assignment or exam to F in course to recommended expulsion.

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 Section 11, Behavior Violating University Standards https://scampus.usc.edu/1100-behavior-violating-university-standards-andappropriate-sanctions. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct. Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the Office of Equity and Diversity http://equity.usc.edu or to the Department of Public Safety http://capsnet.usc.edu/department/department-public-safety/online-forms/contactus. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. The Center for Women and Men http://www.usc.edu/studentaffairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage http://sarc.usc.edu describes reporting options and other resources.

Academic Integrity

Academic integrity is critical the assessment and evaluation we perform which leads to your grade. In general, all work should be your own and any sources used should be cited. Gray-areas occur when working in groups. Telling someone how to do the problem or showing your solution is a VIOLATION. Reviewing examples from class or other sources to help a fellow classmate understand a principle is fine and encouraged. All students are expected to understand and abide by these principles. SCampus, the Student Guidebook, contains the University Student Conduct Code in Section 10, while the recommended sanctions are located in Appendix A. Students will be referred to the Office of Student Judicial Affairs and Community Standards for further review, should there be any suspicion of academic dishonesty.

Support Systems

A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the American Language Institute http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. The Office of Disability Services and Programs http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, USC Emergency Information http://emergency.usc.edu will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.

Academic Accommodations

Any student requiring academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me as early in the semester as possible. DSP is located in GFS 120 and is open 08:30 – 17:00, Monday through Friday. The phone number for DSP is (213) 740-0776.