SYLLABUS

A Computational Introduction to Deep Learning

EE 541: Spring 2024 (2 units)

Machine learning using large datasets is the most transformative technology of the 21st century. Advances in generative ML promise solution to almost any problem imaginable. Machine learning proficiency requires software skills as well as an understanding of the underlying mathematics and theoretical foundations. 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.

Instructor:	Brandon Franzke	
Email:	franzke@usc.edu	
Office:	EEB 504B	
Hours:	Tuesday:	12:00 – 13:30
	Thursday:	11:00 – 13:00 (remote)

Lecture

Monday (section: 31249)
15:00 – 16:50

Enrollment is in-person ONLY. Attendance is mandatory to all lectures. Taping or recording lectures or discussions is strictly forbidden without the instructor's explicit written permission.

Teaching assistants

TA:	Vasileios Magoulianitis	TA:	Haodi Hu
Email:	magoulia@usc.edu	Email:	haodihu@usc.edu
Office:	(see canvas)	Office:	(see canvas)

Grader/CP:	Revathi Prasad
E-mail:	revathip@usc.edu
Hours:	(by appointment)

Course materials

[1] "*Deep Learning*", Ian Goodfellow, Yoshua Bengio, Aaron Courville, The MIT Press, 2016. (online, http://www.deeplearningbook.org).

[2] "*Deep Learning with PyTorch*", Eli Stevens, Luca Antiga, Thomas Viehmann, Manning, 2020. (online, https://pytorch.org/assets/deep-learning/Deep-Learning-with-PyTorch.pdf).

[3] "*Mathematics for Machine Learning*", Marc Deisenroth, Aldo Faisal, and Cheng Ong, Cambridge University Press, 2020 (online, https://mml-book.github.io/book/mml-book.pdf).

[4] "*Neural Networks and Deep Learning*", Michael Nielson. (online, http://neuralnetworksanddeeplearning.com).

[5] "*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: Texts are secondary to in-class lecture material and homework sets.

Piazza https://piazza.com/usc/spring2024/ee541

Canvas https://canvas.usc-ece.com

Electronically submit homework and view grades. You will receive a registration email during the first week of classes. Contact Dr. Franzke with technical issues.

Autolab https://autolab.usc-ece.com

Electronically submit programming homework for *auto-grading*. You will receive a registration email during the first week of classes. Contact Dr. Franzke with technical issues.

Course Outline (tentative)

	Topics	Recommended Reading	Homework
Week 1 09 Jan	Machine Learning inventory. Configuring your Python environment.	[1] Ch. 1, 5.1-5.1.3, 5.3-5.4.	
(15 Jan)	**No class, Martin Luther King Day.**		
Week 2 22 Jan	Getting started with Python. Numerical Python.	[5] Ch. 1-7.	**HW 1 assigned**.
Week 3 29 Jan	Estimation and MMSE.	[1] Ch. 2, 3.	HW 1 due. **HW 2 assigned**.
Week 4 05 Feb	Regression and maximum likelihood.	[1] Ch. 4, 5.5.	
Week 5 12 Feb	Logistic regression. Multilayer perceptron networks (MLPs).	[1] Ch. 5.6, [5] Ch. 14-16, 25.	HW 2 due. **HW 3 assigned**.
(19 Feb)	**No class, Presidents' Day.**		
Week 6 26 Feb	MLP backpropagation (scalar, vector/tensor).	[4] Ch. 1, [2] Ch. 6, [1] Ch. 6.	HW 3 due. **HW 4 assigned**.
Week 7 04 Mar	**Quiz #1 (weeks 1-5).**		HW 4 due. **HW 5 assigned**.
(11 Mar)	**No class, Spring Break.**		
Week 8 18 Mar	PyTorch: Introduction.	[1] Ch. 7-8, [4] Ch. 3.	HW 5 due (21 Mar).
Week 9 25 Mar	PyTorch: Building MLPs.		**HW 6 assigned**.
Week 10 01 Apr	Convolutional Neural Networks (CNN). Project overview.	[2] Ch. 8, [1] Ch. 9.	
Week 11 08 Apr	Introduction to linux CLI and AWS. PyTorch: Optimizing training.		HW 6 due. **HW 7 assigned**. **Project proposal due (11 Apr)**.
Week 12 15 Apr	Advanced data engineering. Embedding and auto-encoding.		HW 7 due (18 Apr).
Week 13 22 Apr	**Quiz #2 (weeks 6-12).**		
03 May	**Project report, due 12:00.**		

Grading Procedure

Homework (50%) Homework is assigned every 1-2 weeks. Assignments include analytic and programming problems and encourage experimentation and curiosity. 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 submit their own original work. Cheating warrants an "F" on the assignment. Turning in substantively identical homework solutions counts as cheating.

Late homework is accepted with a 0.5% deduction per hour, up to 48-hours – **no exceptions**. Technical issues while submitting are not grounds for extension. No submissions will be accepted 48-hours after the due date. Graders score what is submitted and will not follow up if the file is incorrect, incomplete, or corrupt. It is your responsibility to ensure you submit the correct files and that they are accessible.

Quizzes (30%) Quizzes are non-cumulative and cover the most recent material (approximately 6-weeks). They test your ability to apply major principles, demonstrate conceptual understanding, and may require writing Python code. They occur during weeks 7 and 13 (tentative). You are expected to bring a scientific (non-graphing) calculator. You may use a single 8.5"x11" reference sheet (front and back OK). You may not use any additional resources.

Quizzes include multiple-choice and short answer questions. They may also include free-response or open-ended questions to demonstrate conceptual understanding. You are expected to write reasonably correct code as well as determine expected behavior of novel computer code. Grading primarily follows correct reasoning but may include deductions for major syntax errors, algorithmic inefficiency, or poor implementation.

Final project (20%) This course culminates with a final project in lieu of a final exam. Teams of two students apply deep-learning to problem selected from a set of instructor-defined options. Each option will include a *complete* starter dataset. Teams must experiment and document network architecture search, hyper-parameter optimization, and dataset augmentation. Students should treat the final project as a multi-week in-depth homework assignment and work to integrate concepts from the entire semester.

Course Grade

A if 90 - 100 points, **B** if 80 - 89 points, **C** if 70 - 79 points, **D** if 60 - 69 points, **F** if 0 - 59 points.
("+" and "–" at $\approx$ 2.5% 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.

Final Project

Requirements

All projects must use PyTorch as the primary deep learning framework unless approved explicitly in writing by the instructor. Projects may use additional languages and frameworks for tooling and support. The instructor may provide additional requirements when introducing the final project assignment.

Scoring and Milestones

Topic proposal	week 11	10%
Project report	final	35%
Model and source code		50%
Model card		5%

Project Deliverables

Topic proposal: restate the selected problem and include a preliminary discussion about methods, techniques, or proposed variations/modificaitons/extensions to the problem handout. Proposals should be written in anticipation of fulfilling final report requirements. But the proposal is merely a guidepost and reasonable deviations in method, approach, and scope are expected.

Written report: a properly referenced written report that incorporates all source code as well as links to any external code sources. The project report should summarize the topic, provide relevant background (theoretical or applied), timeline and contributions, and document challenges and future extensions. It should provide discussion sufficient for an uninformed expert to understand the problem, model and training decisions, and implementation. Teams should provide quantifiable metrics to justify engineering tradeoffs.

The report must include the following additional elements

(1) a mini "literature survey" of work already done in your problem domain. Understand network architectures used in similar models,

(2) experimental results with various models (both ML and DL) and performance comparison. Use different architectures and hyperparameters,

(3) evidence of good data handling and processing, *i.e.*, maintain separate test, train, and validation sets.

Source code: submitted as a GitHub repository archive file (zip). It must include README file(s) that describe the repository structure, execution instructions, and special technical requirements. It should not include any training data or model files.

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.