CS230, Deep Learning
Handout #1, Course Information

Andrew Ng, Kian Katanforoosh

Class Time and Location

Monday 11:30AM - 12:50PM, STLC 118 (Science Teaching and Learning Center)

Teaching Staff

Andrew Ng, Office: Gates 112
Kian Katanforoosh, Office: Gates 111
Office hours: Fri 3:00PM - 5:00PM, Gates B30, Sun 5:00PM - 7:00PM Gates B21

Teaching Assistant

Ramtin Keramati
Office hours: Gates B21, Wed 5:30PM - 7:30PM , Thu 1:00PM - 3:00PM

Contact Information

If you are enrolled in CS 230 at Stanford and have a question, for you to get a response quickly we strongly encourage you to post it on our Piazza forum. For private matters, please make a private note visible only to the course instructors. If Piazza does not work, you can also email the course staff at: *cs230-qa@cs.stanford.edu*

Course Description

Deep Learning is one of the most highly sought after skills in AI. We will help you become good at Deep Learning. In this course, you will learn the foundations of Deep Learning, understand how to build neural networks, and learn how to lead successful machine learning projects. You will learn about Convolutional networks, RNNs, LSTM, Adam, Dropout, BatchNorm, Xavier/He initialization, and more. You will work on case studies from healthcare, autonomous driving, sign language reading, music generation, and natural language processing. You will master not only the theory, but also see how it is applied in industry. You will practice all these ideas in Python and in TensorFlow, which we will teach. After this course, you will likely find creative ways to apply it to your work. This class is taught in the flipped-classroom format. You will watch videos and complete in-depth programming assignments and online quizzes at home, then come to class for advanced discussions and work on projects. This class will culminate in an open-ended final project, which the teaching team will help you on.

Prerequisites: Familiarity with programming in Python and Linear Algebra (matrix/vector multiplications). CS 229 may be taken concurrently.

Homeworks and Grading

There will be 22 programming assignments, an open-ended term project and a final presentation. Programming assignments will contain questions that require Python programming. In the term project, you will investigate some interesting aspect of deep learning or apply deep learning to a problem that interests you.

Course grades: Grade will be based 40% on homeworks (≈2% each), 2% on attendance, 18% on quizzes and 40% on the term project (including 2% for project proposal, 2% for project milestone, 6% for final presentation and 30% on the final write-up (jupyter notebook, refer to project section)).

Honor code: We strongly encourage students to form study groups. Students may discuss and work on programming assignments and quizzes in groups. However, each student must write down the solutions independently, and without referring to written notes from the joint session. In other words, each student must understand the solution well enough in order to reconstruct it by him/herself. In addition, each student should submit his/her own code and mention anyone he/she collaborated with.

Late assignments: Each student will have a total of ten free late (calendar) days to use for programming assignments, quizzes, project proposal and project milestone. Each late day is bound to only one assignment. Once these late days are exhausted, any assignments turned in late will be penalized 20% per late day. However, no assignment will be accepted more than three days after its due date, and late days cannot be used for the final project and final presentation. Each 24 hours or part thereof that a homework is late uses up one full late day.

Assignment submission: For this course, you will be invited to a private Coursera Session. In this session, you will be able to watch videos, do quizzes and complete programming assignments. Each quiz and programming assignment can be submitted directly from the session and will be graded by our autograder.

- We will begin to accept regrades for an assignment the day after grades are released for a window of two days. We will not accept regrades for an assignment outside of that window. Regrades are intended to remedy grading errors, so regrade requests must discuss why you believe your answer is correct in light of the deduction you received. We do not accept regrade requests of the form "I deserve more points for this" or "that deduction is too harsh." If you submit a regrade request of this form, you will receive further deductions. When you submit a regrade request, the grader may review your entire assignment, in which case you may lose points on other questions. Your score on an assignment may decrease if you submit for a regrade. To ask a regrade, please send your jupyter notebook and an explanation of your request by email to cs230-fall1718-staff@lists.stanford.edu

Project

40% of your grade is based on your final project, you will investigate some interesting aspect of deep learning or apply deep learning to a problem that interests you. The term project may be done in teams of up to three persons.

Project Proposal

Deadline: Monday 10/16, 09:00AM
The project proposal should include the project team members and a brief overview of the proposed project and project plan that includes the following (1-2 pages):

- What is the problem that you will be investigating? Why is it interesting?
- What are the challenges of this project?
- What dataset are you using? How do you plan to collect it?
- What method or algorithm are you proposing? If there are existing implementations, will you use them and how? How do you plan to improve or modify such implementations?
- What reading will you examine to provide context and background? If relevant, what papers do you refer to?
- How will you evaluate your results? Qualitatively, what kind of results do you expect (e.g. plots or figures)? Quantitatively, what kind of analysis will you use to evaluate and/or compare your results (e.g. what performance metrics or statistical tests)?

Submission: One member on your team should submit your project proposal using the google form. If you are doing this project to count for this and another class, we expect the project to be larger in scope.

Project Milestone

Deadline: Friday 11/18, 11:59PM
Your project milestone report should be in the format of a jupyter notebook. The following is a suggested structure for your report:

- Title, Author(s)
- Introduction: this section introduces your project, why its important or interesting.
- Loading the packages you are using.
- Details on the dataset
- Approach: Describe the current steps you have done. If you are implementing an algorithm, you should have started implementation and ideally have some early stage results. Describe precisely the remaining work you expect to complete. We ideally would like to see a model description and a training strategy (loss function for instance).

Submission: Please send your jupyter notebook by email to *cs230-qa@cs.stanford.edu* with the subject of the email *<your SUNet ID>_ milestone*. Note that, only one group member in a team is required to make submission.

Final Presentation

Time: Mon 12/11, 11:30-12:50pm, Location to be determined.
You will present your project in a short 5 min in-class presentation, open to the public. You can use any

material you would like (slides, video, notebook ...). In this presentation, you will give an overview of your project and your model, explain what were the main technical challenges, discuss your results and future opportunities.

Final Submission

Deadline: Tue 12/12, 11:59PM
Your final project should be a tar folder containing:

- Your jupyter notebook containing all the write-ups in markdown and your code. All the cells should be run (on your side) when you submit. So please run all the cells before downloading/saving your notebook. The notebook should at least contain a thorough explanation of your model, your dataset, your training/validating/testing process, your challenges, an explanation about the hyperparameters, optimization, regularization you choose, the performance of your algorithm, error analysis, some thoughts on future works, .
- Any dependency / utils / source code file needed
- We dont need your dataset, you can send us a link to it but the most important is that you include a full description of your dataset in your jupyter notebooks (with plots and figures if necessary).
- The materials you used in your final presentation.
- Any other folder you think we need. For example containing images youve added in your notebooks markdown cells, or Cool videos, interactive visualizations, demos, etc. (optional)

Examples of things to not put in your supplementary material:

- All of a submodules (numpy, tensorflow, ...) source code.
- Any code that is larger than 1MB.
- Model checkpoints.
- A computer virus.

After the class, we will post all the jupyter notebooks / presentations online so that you can read about each other's work. If you do not want your notebook/presentation to be posted online, then please let us know when you submit your notebook. You should include a brief statement on the contributions of different members of the team. Team members will normally get the same grade, but we reserve the right to differentiate in egregious cases.

Submit your tar file submission by email to *cs230-qa@cs.stanford.edu* with the subject of the email *<your SUNet ID>_ final*. only one group member in a team is required to make submission.