HARVARD
Summer School

CSCI S-89C Deep Reinforcement Learning
Syllabus Summer 2020

<u>**Lectures:**</u> Online web conference, Tuesdays and Thursdays, 3:15 - 6:15 pm
Lectures will be live-streamed with the video being available via the course website within 24 hours.

<u>**Instructor:**</u> **Dr. Dmitry Kurochkin**, Senior Research Analyst, Harvard University
<u>**E-mail:**</u> dkurochkin@fas.harvard.edu
<u>**Website:**</u> `https://canvas.harvard.edu/courses/72462`
<u>**Office Hours:**</u> By request
<u>**Teaching Fellows:**</u> TBA e-mail: TBA

<u>**Prerequisites:**</u>

Introductory probability and statistics, multivariate calculus equivalent to MATH E-21a, and proficiency in Python programming equivalent to CSCI E-7.

Note on the prerequisites:
We will be formulating value (cost) functions and performing optimization. Students are expected to be comfortable taking derivatives. Basic knowledge of probability theory (in particular, conditional probability distributions and conditional expectations) is necessary. Understanding matrix vector operations and notation is helpful but not required. All coding exercises are performed in Python. Students are required to take a short pretest at the beginning of the course. The pretest score will <u>not</u> count toward the final grade but will help you understand whether your background in calculus, probability theory, as well as command of coding positions you for success in this course.

<u>**Course Description:**</u>

This course introduces Deep Reinforcement Learning (RL), one of the most modern techniques of machine learning. Deep RL has attracted the attention of many researches and developers in recent years due to its wide range of applications in a variety of fields such as robotics, robotic surgery, pattern recognition, diagnosis based on medical image, treatment strategies in clinical decision making, personalized medical treatment, drug discovery, speech recognition, computer vision, and natural language processing. Deep RL is often seen as the third area of machine learning, in addition to supervised and unsupervised algorithms, in which learning of an agent occurs as a result of its own actions and interaction with the environment. Generally, such learning processes do not need to be guided externally, but it has been difficult until recently to use RL ideas practically. This course primarily focuses on problems that emerge in healthcare and life science applications.

Tentative List of Topics:

I. **Deep Learning**

- Neural Networks (NN): Classification & Regression
- Training NNs: Backpropagation
- Tuning NNs: Regularization
- Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN)

II. **Reinforcement Learning (RL)**

- Markov Decision Processes (MDP): Value Functions and Policies
- Dynamic Programming (DP): Bellman Equation
- Monte Carlo (MC) Methods
- Temporal-difference (TD) Prediction and Control: SARSA and Q-learning

- n-step TD
- Approximation Methods: Stochastic-gradient, Semi-gradient TD Update, Least-squares TD

III. **Deep RL**

- Value-based Deep RL: Q-network
- Policy-based Deep RL: REINFORCE
- Asynchronous Methods for Deep RL: Advantage Actor-Critic (A2C)
- Model-based Deep RL

Text:

Richard Sutton and Andrew Barto, *Reinforcement Learning: An Introduction*, 2nd ed.
ISBN: 978-0-262-03924-6
Electronic copy of the book is available at the author's webpage (under "Full Pdf pdf without margins")
`http://incompleteideas.net/book/the-book-2nd.html`

Optional reading:
Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016
ISBN: 978-0-262-03561-3
HTML version of the book is available at `http://www.deeplearningbook.org`

Homework:

Except when especially noted, homework assignments will be due before each lecture. The assignments will be posted on Canvas website and will consist of series of programming exercises (solutions should be implemented in Python) as well as analytical problems (knowledge of calculus and probability theory should suffice) that help students enhance their understanding of the underlying theory. Solutions to the programming exercises should be submitted via Canvas in a form of a single .ipynb (Jupyter Notebook) file. The solutions to the theoretical problems should be submitted in a form of a single PDF file.

Note on the deadline and penalty:
Solutions to the assignments submitted later than 1, 2, and 3 days after the due date will be penalized by 10%, 50%, and 100%, respectively. In case you need an extension, please coordinate with the instructor prior to the due day.

Quizzes:

An online quiz will be due before each class, unless announced otherwise. The quiz will consist of approximately 5 basic questions on understanding of studied principals. No late quizzes will be allowed.

Midterm Exam:

The midterm exam will be due TBA. The test will be similar to Homework exercises but cover topics studied up to this date. Late midterm will not be accepted.

Final:

The final examination will be due at 11:59 pm (Eastern Time) on August 9. The exam will be cumulative covering all topics studied. Late final will not be accepted.

Attendance:

Regular attendance (whether on campus or online) is expected but will not be taken. Recorded lectures will be available via the course website within 24 hours after the lecture.

Participation:

Although no credit is allocated for participation, everyone is encouraged to constructively participate in class by asking relevant questions. It is important that you check the e-mail registered with Canvas regularly and monitor course announcements and also participate in discussions on Piazza, the forum available at `https://piazza.com/class/k70vdldzqmj79`. All technical and data science related questions will be discussed on Piazza.

Grading:

A letter grade will be given in accordance with the School's grading policy (see `https://www.extension.harvard.edu/grades`). The semester average is calculated using the formula:

$$\text{Grade} = 0.25 \cdot \text{Homework} + 0.20 \cdot \text{Quizzes} + 0.25 \cdot \text{Midterm} + 0.30 \cdot \text{Final}$$

Student Learning Objectives:

- proficiency in building optimal NNs using Python
- understanding of RL including MDP, Bellman equation, and optimal policy
- firm understanding of Deep RL and getting comfortable with approximation methods used in conjunction with RL
- hands-on experience on estimating the optimal policy and value functions

Academic Integrity:

You are responsible for understanding Harvard Summer School policies on academic integrity (`www.summer.harvard.edu/resources-policies/student-responsibilities`) and how to use sources responsibly. Not knowing the rules, misunderstanding the rules, running out of time, submitting the wrong draft, or being overwhelmed with multiple demands are not acceptable excuses. There are no excuses for failure to uphold academic integrity. To support your learning about academic citation rules, please read "Plagiarism" section, where you'll also find links to the Harvard Guide to Using Sources and two free online 15-minute tutorials to test your knowledge of academic citation policy. The tutorials are anonymous open-learning tools.

Disability Accommodations:

The Summer School is committed to providing an accessible academic community. The Accessibility Office offers a variety of accommodations and services to students with documented disabilities. More information can be found at `https://www.summer.harvard.edu/resources-policies/accessibility-services`

Dates of Interest:

- Harvard Summer School classes begin, June 22, 2020
- Pretest is due, June 23
- Last day to change the credit status, June 24
- Course drop deadline for full-tuition refund, June 24
- Quiz 1 is due, June 25
- Assignment 1 is due, June 25
- Course drop deadline for half-tuition refund, July 1
- **Midterm Exam** is due TBA
- Withdrawal deadline, July 24
- **Final Exam** is due, August 9, 11:59 pm (Eastern Time)