Link for Submission: <https://forms.gle/n5KbqaDF2VQEgYTA6>

Note for Submission: This is a challenging homework assignment! Hence we will be doing two things differently: (a) It is ok to search online for implementations to take inspiration from. However, no copying/plagiarism is permitted. (b) giving three weeks to solve this homework instead of the usual two weeks. This also means that no extra days will be provided.

For the code portion of the HW, make 2 subfolders for each part and submit the drive link for the parent folder.

Similarly, add 2 PDFs for each part and submit the drive link for the parent folder.

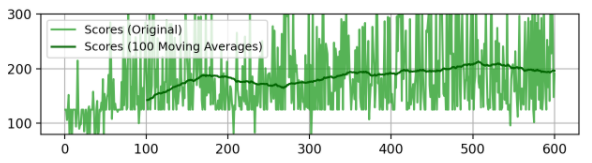
Deadline for submission**: Nov. 5, 2020.**

# Deep Q-network

By now you would be familiar with the car-racing environment, for this part you will implement DQN for car-racing. Please submit the code along with learning curves, visualization of frames, and final performance for this part.

Helpful resources:

* Paper: <https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf>
* DQN for CartPole-v0 task example: <https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html>
* Expected scores (~300 Avg reward per 100 episode)



**Extra credit**: Prioritized replay in DQN for the same environment.

# 

# REINFORCE

Like a car-racing environment, Open-AI gym offers many other environments. One of them is CartPole : <https://gym.openai.com/envs/CartPole-v1/>. For this part implement REINFORCE for CartPole. Please submit the code along with learning curves, visualization of frames, and final performance for this part.

Helpful resources:

* Paper: <https://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.pdf>
* REINFORCE sample code: <https://github.com/pytorch/examples/blob/master/reinforcement_learning/reinforce.py>

**Extra credit**: Implement PPO for the same environment.