

The purpose of this homework is to learn how to play games using policy gradient. In each of the games mentioned below, the objective for the agent is to win the game or maximize the total score achieved during the game.

1 Part 1 (85 points)

a) **CartPole-v0**. Take a look [here](#) to learn how the game is played. Train a simple neural net that models the policy. Use discount factor = 0.95. Plot the episode reward against the number of training episodes.

b) **Pong-v0**. Take a look [here](#) to learn how the game is played. Train a neural net that uses images of the game as state and models the policy. The discount factor is 0.99. Use frames of the game as input to a neural net which models the policy. The game has six actions [NOOP, FIRE, RIGHT, LEFT, RIGHTFIRE, LEFTFIRE], but you should train a network that chooses the best action only between [RIGHT, LEFT] (actions 2 and 3).

Plot the episode reward against the number of training episodes, and overlay it with the simple moving average of the episode rewards for the last 100 episodes.

You may pre-process all the images from the game with following function before feeding them into the neural network.

```
def preprocess(image):  
    """ prepro 210x160x3 uint8 frame into 6400 (80x80) 2D float array """  
    image = image[35:195] # crop  
    image = image[:, :, ::2, 0] # downsample by factor of 2  
    image[image == 144] = 0 # erase background (background type 1)  
    image[image == 109] = 0 # erase background (background type 2)  
    image[image != 0] = 1 # everything else (paddles, ball) just set to 1  
    return np.reshape(image.astype(np.float).ravel(), [80,80])
```

Notes:

- You are required to use the specified version of the game environments, i.e. CartPole-v0 and Pong-v0, for both questions.
 - In your submission, include your code used to train the model and create the plots, the report, and the instruction of how to run your code.
 - All of you must use deepdish for question 2 with a single GPU card.
 - You are responsible to run your code and create the plots.
-