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:

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 (average) episode reward versus training epochs.

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).

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,::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])
```

Note that for this game an episode finishes when one of the players wins 21 games. Plot (average) episode reward versus training epochs.

Notes:

- All of you must use deepdish and your code must use a single GPU card.
- You are responsible to run your code and create the plots.
- You are allowed to use any trick that can help with faster convergence.
- Using a baseline helps to reduce variance of rewards in your sample paths. Although you can train the networks without baseline, you are encouraged to use a baseline. Students who do not use a baseline can get at most 85 points out of 100.
- Your assignment score will be based on the performance of your algorithm and how fast it converges. You will get the majority of the points as long as your code is correct and your plots show that the

network is learning how to play. If your code works correctly, i.e. the algorithm is learning, you will get 85 points in the case you are not using a baseline. If you are using a non-constant baseline, then we will use the average reward over 48 hours of wall time. The following criterion will be used for those students using a baseline:

- Top 20% performance get 100 points.
- Those ranked from 50% to 80% get 95 points.
- Those ranked from 20% to 50% get 92.5 points.
- Bottmon 20% get 90 points.
- The winner gets an extra 10 points (thus 110 points).