Part 2 (15 points)

You have implemented the policy gradient algorithm to learn how to play the two games. However, the training suffers from high variance of the rewards in the sampled episodes. Using a baseline helps to reduce the variance. In Part 2, you will implement **policy gradient with (non-constant) baselines** to solve the two problems.

a) **CartPole-v0.** Again, 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.** Again, train a neural net that uses images of the game as state and models the policy. Use discount factor = 0.99. Use frames of the game as the 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).
For Pong-v0,

- 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.
- When you finish training, roll out 500 episodes using your trained model. Plot the histogram of the 500 episode rewards and report the mean and the standard deviation.

In your submission, include:

- your code used to train the model and create the plots;
- the report of required plots for both questions;
- the instruction of how to run your code.

Your code should work correctly when a non-constant baseline is used, i.e. the algorithm is learning for both questions. Your score for part 2 will be based on the performance of your algorithm and how fast it converges.