Berkeley DRL HW2

Anton Makiievskyi

Problem 1: State-dependent baseline legitimacy proof

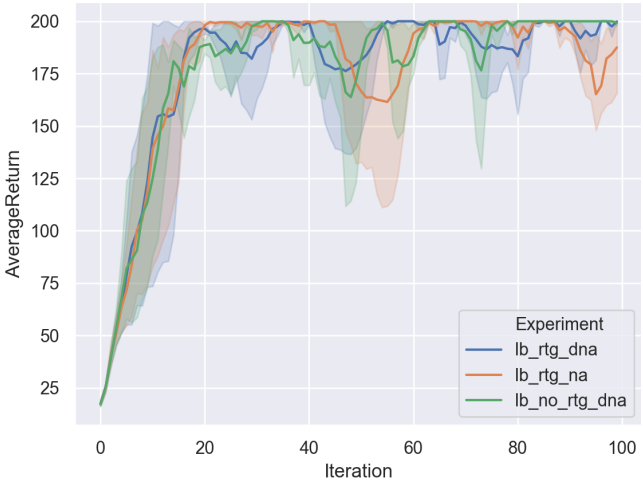
$$\mathbb{E}_{\tau\sim p_\theta(\tau)}\left[\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)b(s_t)\right] = \sum_{t=1}^{T}\mathbb{E}_{\tau\sim p_\theta(\tau)}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\big(b(s_t)\big)\right]$$

$$\mathbb{E}_{\tau\sim p_\theta(\tau)}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\big(b(s_t)\big)\right] =$$
$$\mathbb{E}_{\tau\sim p_\theta(\tau/s_t,a_t|s_t,a_t)}\mathbb{E}_{s_t\sim p_\theta(s_t)}\mathbb{E}_{a_t\sim\pi_\theta(a_t|s_t)}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\big(b(s_t)\big)\right] =$$
$$\mathbb{E}_{\tau\sim p_\theta(\tau/s_t,a_t|s_t,a_t)}\mathbb{E}_{s_t\sim p_\theta(s_t)}\left[b(s_t)\mathbb{E}_{a_t\sim\pi_\theta(a_t|s_t)}\nabla_\theta \log \pi_\theta(a_t|s_t)\right] =$$

$$\mathbb{E}_{\tau\sim p_\theta(\tau/s_t,a_t|s_t,a_t)}\mathbb{E}_{s_t\sim p_\theta(s_t)}\left[b(s_t)\int\nabla_\theta \pi_\theta(a_t|s_t)da_t\right] =$$
$$\mathbb{E}_{\tau\sim p_\theta(\tau/s_t,a_t|s_t,a_t)}\mathbb{E}_{s_t\sim p_\theta(s_t)}\left[b(s_t)\nabla_\theta\int\pi_\theta(a_t|s_t)da_t\right] = \mathbb{E}_{\tau\sim p_\theta(\tau/s_t,a_t|s_t,a_t)}\mathbb{E}_{s_t\sim p_\theta(s_t)}\left[b(s_t)\nabla_\theta 1\right] = 0$$

Problem 4:

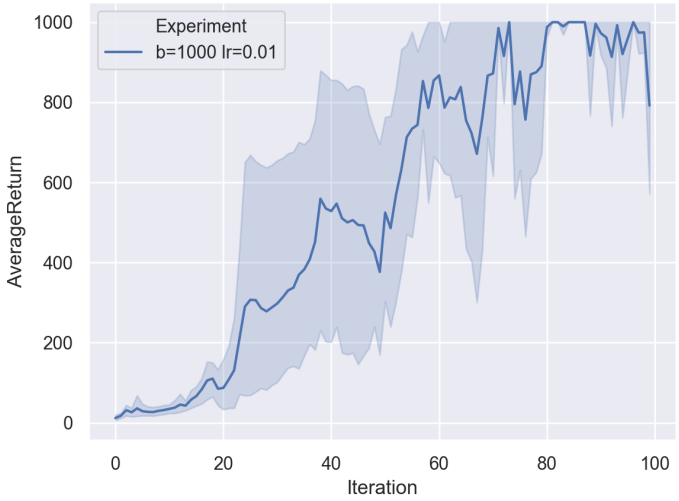
All experiments on the CartPole task were performed exactly as specified in the assignment.
From the results(graphs below) we can see that "reward-to-go" helps to learn faster, while normalization of advantages helps to reduce variance.
Both improvements can be more easily seen in small batch experiments, probably because large batch experiment suffer from these problems less.



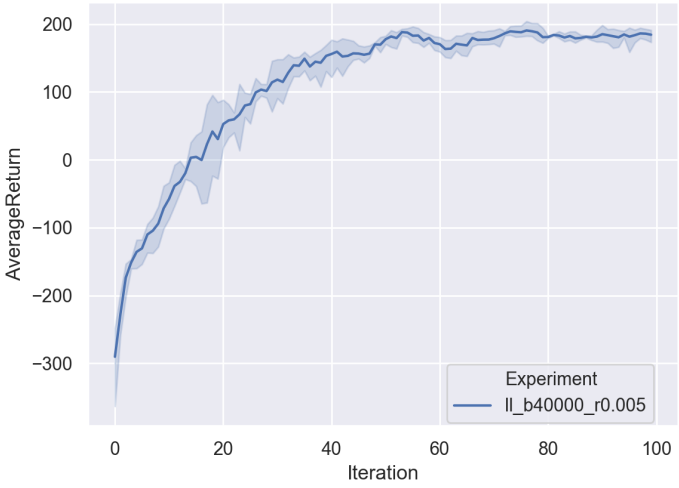
Problem 5:

To find the best parameters for InvertedPendulum task, I've ran a grid search over batch sizes [500, 1000, 2000, 4000], and learning rates [0.005, 0.01, 0.03, 0.05, 0.1]

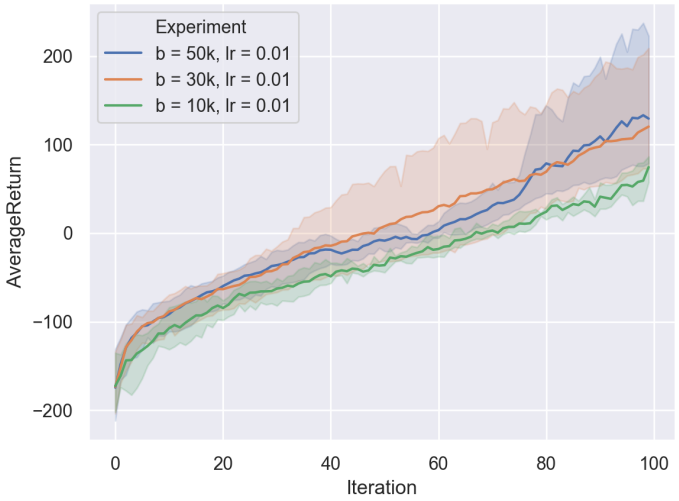
The best result was achieved with batch size 1000 and learning rate of 0.01, I ran 5 experiments with those setting

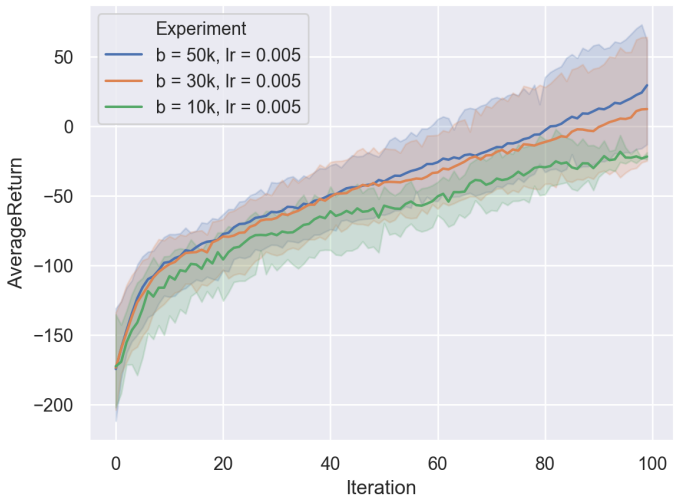


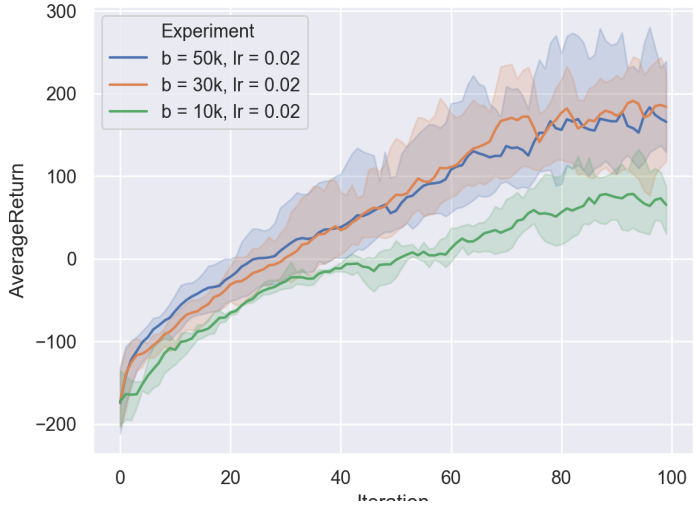
Problem 7:
LunarLanderContinuous task testing a baseline was ran as specified in the assignment



Problem 8:
After running experiments for 9 different hyperparameter settings, I've plotted the learning curves, first comparing different batch sizes using fixed learning rate.

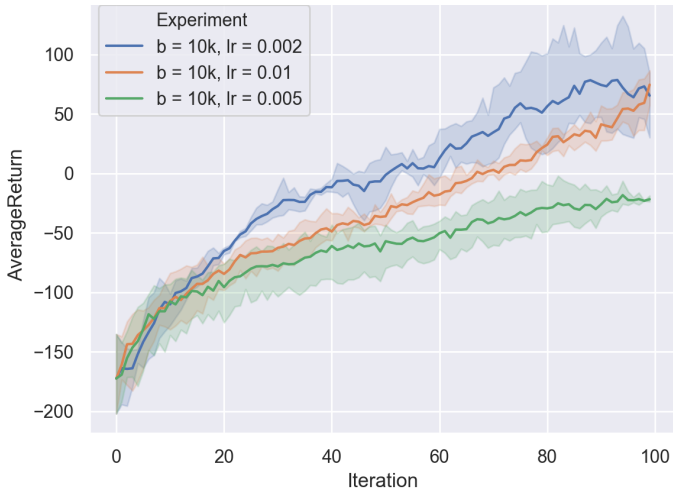


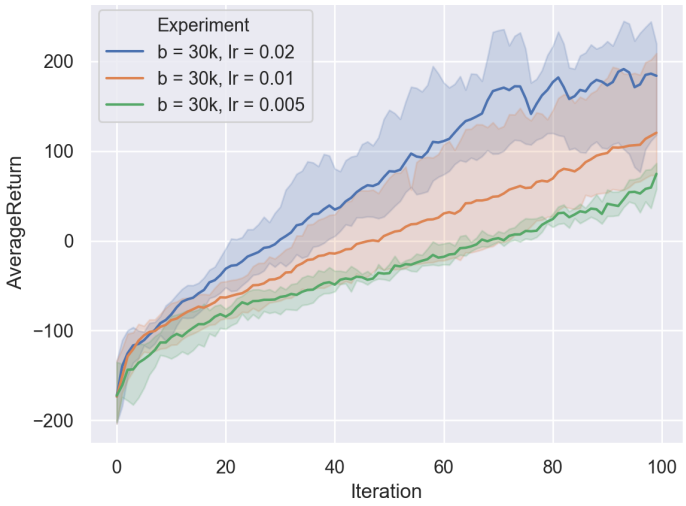


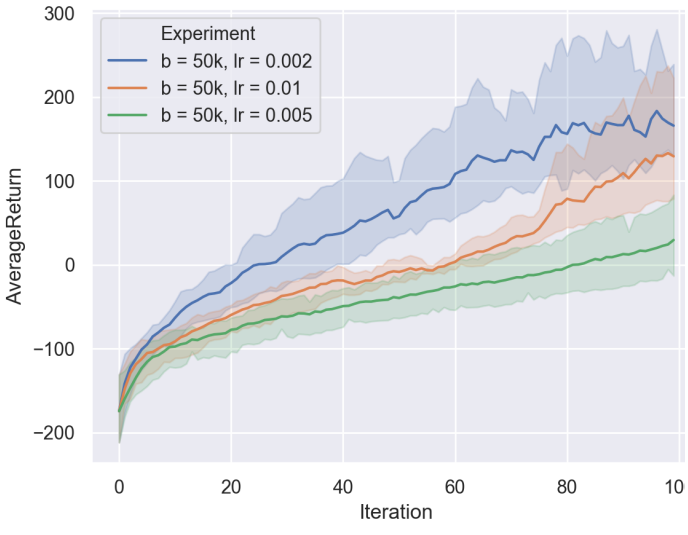


I concluded that higher batch size results in better performance which is reasonable since with larger batch size we use more training data with less noise in the gradient.

Similarly comparing learning rates I concluded that higher learning rate leads to better performance. In this case the dependence was more pronounced:







Thus, for the final 4 experiments, I've used the batch size of 50k and learning rate 0.02

In this experiment "reward to go" gave a better boost of performance compared to baseline. The run with both of them achieved the best result