Homework 4

Ted Xiao

March 28, 2017

1 Continuous Policy Gradient

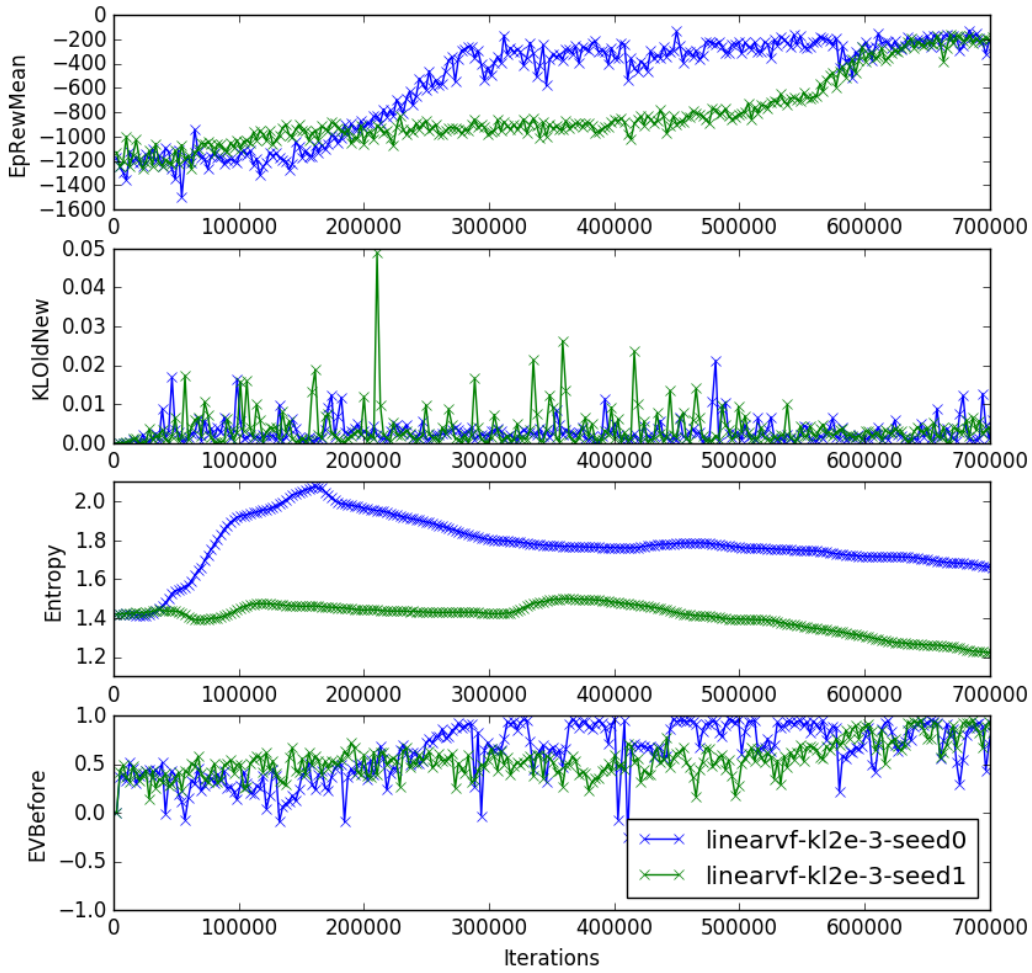

Figure 1: Continuous policy gradients applied to the Pendulum environment. Both seeds converge to rewards of at least -300. I used the default hyperparameters, but cut off the plot as soon as the results converged, which was at around 700000 steps. The default hyperparameters used were: $\gamma = 0.97$, n_iter=300,initial_stepsize=1e-3, desired_kl=2e-3, min_timesteps_per_batch=2500. As seen above, the algorithm learned in around 650000 to 700000 steps.

2 Neural Network Value Function

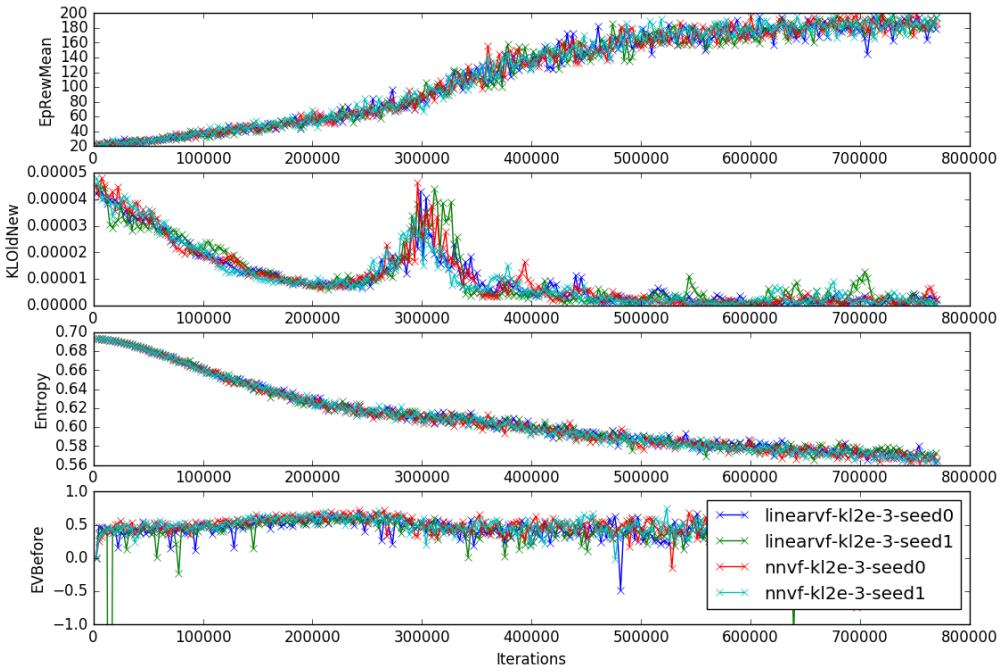

Figure 2: The Neural Network Value function performs marginally better than the Linear Value Function on the cartpole task. The Neural Network Value function converges faster and has less variance overall. For the hyperparameters, the default training parameters were used: n_epochs=10, stepsize=1e-3. The architecture was a simple two hidden layer feedforward neural network, with one hidden layer of 32 neurons and the second layer of 16 neurons.

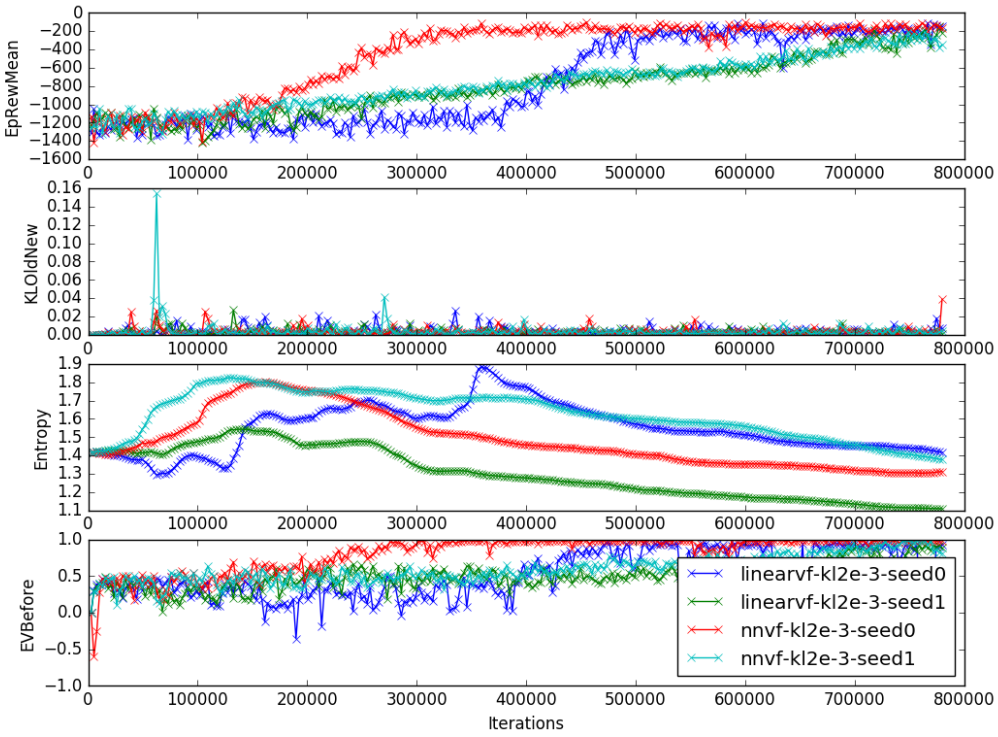

Figure 3: The Neural Network Value function performs much better than the Linear Value Function on the pendulum task. The overall mean episode reward increases much faster than the linear value function. This may be because the task is harder to learn for a linear matrix model, so a neural network is better able to learn this more complex task. For the hyperparameters, the default training parameters were used: n_epochs=10, stepsize=1e-3. The architecture was a simple two hidden layer feedforward neural network, with one hidden layer of 32 neurons and the second layer of 16 neurons.