Page 11, TargetNet class

The first mode is the standard way to perform a target network sync in discrete action space problems, like Atari and CartPole. We did this in Chapter 6. The latter mode is used in continuous control problems, which will be described in several chapters in Part Four of the book. In such problems, the transition between two networks parameters should be smooth, so alpha blending is used, given by the formula: $w_i = w_i\alpha + s_i(1 - \alpha)$, where w_i is the target networks ith parameter and s_i is the source networks weight. The following is a small example of how TargetNet should be used in code.

Page 15, Experience replay buffers

Provided classes:

1. ExperienceReplayBuffer: A simple replay buffer of predefined size with uniform sampling.

2. PrioReplayBufferNaive: A simple but not very efficient prioritized replay buffer implementation. The complexity of sampling is $O(n)$, which might become an issue with large buffers. This version has a benefit over the optimized class, having much easier code.

3. PrioritizedReplayBuffer: Uses segment trees for sampling, which makes code cryptic, but with $O(\log(n))$ sampling complexity.