Deep Recurrent Q-Learning:

Depicted in Figure 1, the architecture of DRQN replaces DQN’s first fully connected layer with a Long Short Term Memory (Hochreiter and Schmidhuber 1997). For input, the RNN-CNN network takes 4 80 x 80 preprocessed game screen images. (We will experiment with a reduced state space of 3 or 2 game screens in the future to show that reducing the state space does not degrade performance). The first hidden layer convolves 16 19 x 19 filters with stride 8 across the input image and applies a rectifier nonlinearity. The second hidden layer convolves 32 8 x 8 filters filters with stride 4, again followed by a rectifier nonlinearity. Convolutional outputs are fed to the fully connected LSTM layer. Finally, a fully connected linear layer outputs a Q-Value for each possible action.

Stable Recurrent Updates:

Bootstrapped Sequential Update:

Episodes are selected randomly from the replay memory and updates begin at the beginning of the episode and proceed forward through time to the conclusion of the episode. The targets at each timestep are generated from the target Q-network. The RNN’s hidden state is carried forward throughout the episode.