

Report & Notes

I implemented the code from DQN lesson. For the model and agent files, I used the mostly the same code except hyper-parameters and some the references to other files. In the agent and model files; regarding hyper parameters, I ;

- increased the learning rate as I observed that it increased the learning speed.
- decreased the gama to 0.9 in order to give more importance to the next moves (to consider blues behind yellows).
- increased the depth of neural layers to 256 for better approximation of q values.
- also increased the batch size to 128 for better learning.

In the main training algorithm, I ;

- reached to the target score 13 in 533 episodes. Similarly I reached to the 13 in 571 episodes in my computer locally.
- set target score to 13 as required in rubric.
- set n_episodes=1000 and time steps 500 which was ideal for training.
- kept eps_start=0.5 and eps_end=0.01 in order to decrease the exploring decrease faster as the environment doesn't require complex exploration.

You can see the plotted rewards over episodes.

You can also see test results of a trained agent in 250 time steps.

I also uploaded the videos to the youtube. You can watch those at below links:

Training : <https://youtu.be/J3B5zx8BLB8>

Trained Agent: <https://youtu.be/yNvRt8H9X1s>

Future Ideas:

- Currently I am trying to train the agent with less number of actions by decreasing the action space to 3. I already observed agent is able to get score around 5 very quickly with only 3 actions, forward,backward and left. It is to be investigated further.
- Prioritizing the important expriences by sampling more from them can be helpful for faster training. It is to be investigated.
- Alternative neural architectures can be tried. For example LSTM or Convolutional layers can be used for sequence pixel inputs.

Best, Volkan