RL_exercise

Environment: CartPole-v0 from gym

Agent algorithm: Deep Q-Learning

Algorithm 1: deep Q-learning with experience replay.

Initialize replay memory D to capacity N

Initialize action-value function Q with random weights θ

Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$

For episode = 1, M **do**

  Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$

  For t = 1,T **do**

    With probability ε select a random action a_t

    otherwise select $a_t = \text{argmax}_a Q(\phi(s_t), a; \theta)$

    Execute action a_t in emulator and observe reward r_t and image x_{t+1}

    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$

    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in D

    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from D

    Set $y_j = \begin{cases} r_j & \text{if episode terminates at step j+1} \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^{-}) & \text{otherwise} \end{cases}$

    Perform a gradient descent step on $\left(y_j - Q(\phi_j, a_j; \theta)\right)^2$ with respect to the network parameters θ

    Every C steps reset $\hat{Q} = Q$

  End For

End For

Q-learning is a model-free algorithm that learns optimal Q(s,a) action value functions from the agent's history of interaction with the environment.

In Deep Q-learning, at each time step the agent memorizes some experience (state, action, reward, next_action, done), and learns by replaying a batch of experience from its memory. This is done to reduce the effect of correlations between sequence of observations which would have made neural network behave poorly.

Current Results

Currently this is just a straightforward implementation of Deep Q-learning algorithm. The agent can consistently pass the given task after around 150 episodes of learning. A higher number of episodes (around 300) allow the agent to always get maximum rewards(200) from each game. Further hyper parameters turning could possibly improve the result.

Reference

Deep Q-Learning Nature

Mxnet Tutorial