用 DQN 学习 pong 游戏

20214969 凌笑铃

1 实现思路

DQN 的设计思路主要参考[1]中的 Algorithm 1，每帧做一个动作，取临近四帧的图像作为输入状态，用 openai 提供的 baselines 模块中的 atari_wrappers.py 对帧进行预处理。

训练过程的实现分为 4 个 python 文件。

deep_q_network.py 基于 keras 框架实现了 DQN 的功能，包括输入当前状态获得动作函数 get_action、复制网络参数函数 copy_from、训练函数 train、预测函数 predict 等。

memory_buffer.py 实现了保存样例和随机选择过去样例的功能。

utils.py 主要实现了 epsilon_greedy 函数，用于判断当前状态是随机选择一个 action、还是选择 dqn 给出的最佳 action。

train.py 主要是参照[1]中的 Algorithm 1 实现了训练过程。核心代码如下：

```
next_frame, reward, done, _ = env.step(action)
next_state = np.array(next_frame)
buf.push(state, action, reward, next_state, done)
state = next_state
cur_episode_reward += reward
if buf.size() > MIN_BUFFER:
    states, actions, rewards, next_states, dones = buf.sample(MINI_BATCH)
    next_state_action_values = np.max(target_dqn.predict(next_states / 255.0), axis=1)
    y_true = dqn.predict(states / 255.0)  # y_true.shape: (MINI_BATCH, num_actions), i.e., (32, 6)
    y_true[range(MINI_BATCH), actions] = rewards + GAMMA * next_state_action_values * np.invert(done
s)
    dqn.train(states / 255.0, y_true)
```

2 实验设置

表 1 参数设置

参数	值
minibatch size	32
replay memory size	10001
agent history length	4
target network update frequency	1000
learning rate	2.5e-4
optimizer	adam
initial exploration	1
final exploration	0.01
final exploration frame	10000
replay start size	10000

3 实验结果

3.1 Reward 学习曲线

图 1 的左图为每个 episode 对应的 reward 曲线，右图为取 100 个 episode 的平均 reward 曲线。由图 1 可以看到，模型性能可以达到 18 分左右。

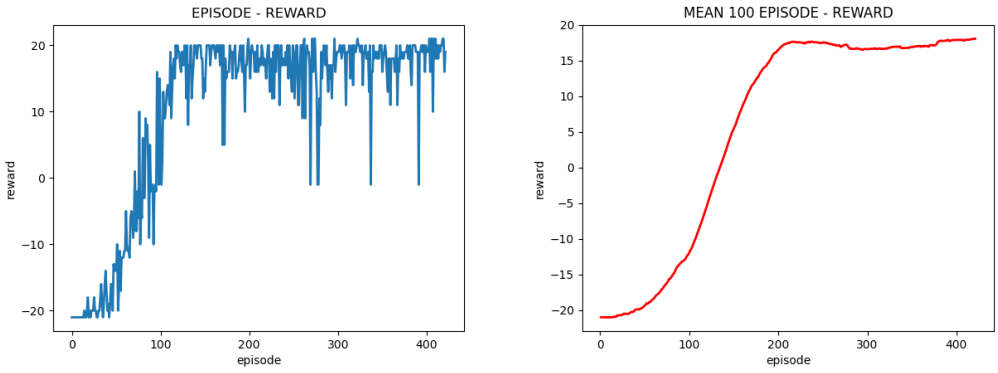

图 1 episode-reward 曲线图和 mean 100 episode - reward 曲线图

3.2 超参数对照试验

将 batch_size 分别设置为[16,32,64]进行对比实验，得到如下的结果。由图

2，可以看到当 batch_size=16 时，DQN 在运行 1000 个 episode 之后仍然没有收敛迹象。作为对比，batch_size=32（图 1）和 batch_size=64（图 3）时，在不到 150 个 episode 之后，DQN 就能达到 20 分左右。因此，在 DQN 的训练过程中，batch size 的影响不可忽视，batch size 选一个太小的值 DQN 可能无法收敛。

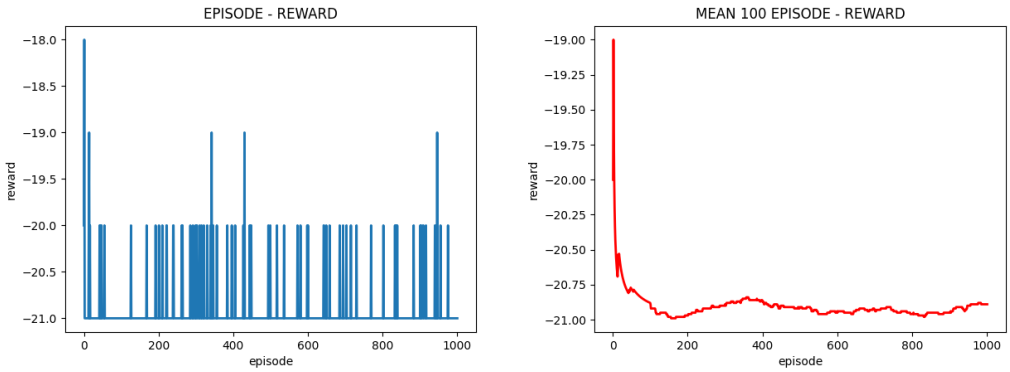

图 2 batch_size=16 时的 episode-reward 曲线图和 mean 100 episode - reward 曲线图

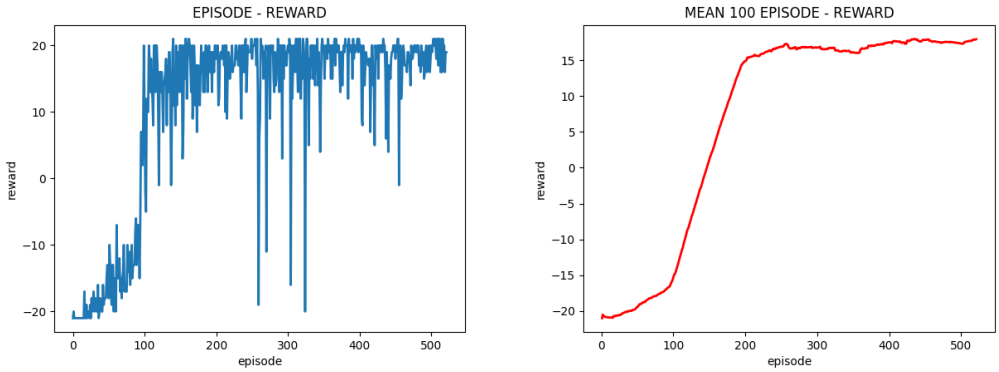

图 3 batch_size=64 时的 episode-reward 曲线图和 mean 100 episode - reward 曲线图

参考文献

[1] Volodymyr M , Koray K , David S , et al. Human-level control through deep reinforcement learning[J]. Nature, 2019, 2015 年 518 卷 7540 期(7540):529-33 页.