强化学习：作业三

杨思航 191180166

December 15, 2021

1 作业内容

在 Atari 环境下实现 Deep Q-learning Network 算法。

2 实验环境

- NAME = Ubuntu
- VERSION = 20.04.2 LTS(Focal Fossa)

3 实现过程

1. 超参修改

   (1) 修改原则: 最大化 DQN 性能

   (2) learning_rate: $1e-6 \rightarrow 1e-4$

   (3) learning_interval: $4 \rightarrow 1$

   (4) 优化器:

   - RMSprop $\rightarrow$ Adam
   - eps: $1e-5 \rightarrow 1e-3$
   - 移除参数 weight_decay, momentum, centered

2. DQN (后续代码均仅列出相较于 DQN 而言修改过的部分)

   - Q

```
Q = q_values.gather(1, a).cuda()
```

   - target-Q

```
not_done = 1 - done
next_max_q = self.target_model(s1).max(1)[0].detach().unsqueeze(1).cuda()
next_target_q_values = not_done * next_max_q
target_Q = r + (self.config.gamma * next_target_q_values).cuda()
```

   - loss (loss_func = MSELoss)

```
loss = self.loss_func(Q, target_Q)
```

3. Double DQN

   - target-Q

```
next_q_values = self.model(s1).cuda()
actions = next_q_values.max(1)[1].unsqueeze(1).cuda()
not_done = 1 - done
next_max_q = self.target_model(s1).gather(1, actions).cuda()
next_target_q_values = not_done * next_max_q
target_Q = r + (self.config.gamma * next_target_q_values).cuda()
```

4. Dueling DQN

- value

```
self.value = nn.Sequential(
    nn.Linear(self.features_size(), 512),
    nn.LeakyReLU(),
    nn.Linear(512, 1)
)
```

- advantage

```
self.advantage = nn.Sequential(
    nn.Linear(self.features_size(), 512),
    nn.LeakyReLU(),
    nn.Linear(512, self.num_actions)
)
```

- forward
  - From Lecture 8 Page 14

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a';\theta,\alpha)\right)$$

  - 代码

```
def forward(self, x):
    batch_size = x.size(0)
    x = self.features(x)
    x = x.reshape(batch_size, -1)
    value = self.value(x)
    advantage = self.advantage(x)
    return value + (advantage - advantage.mean())
```

4 实验效果

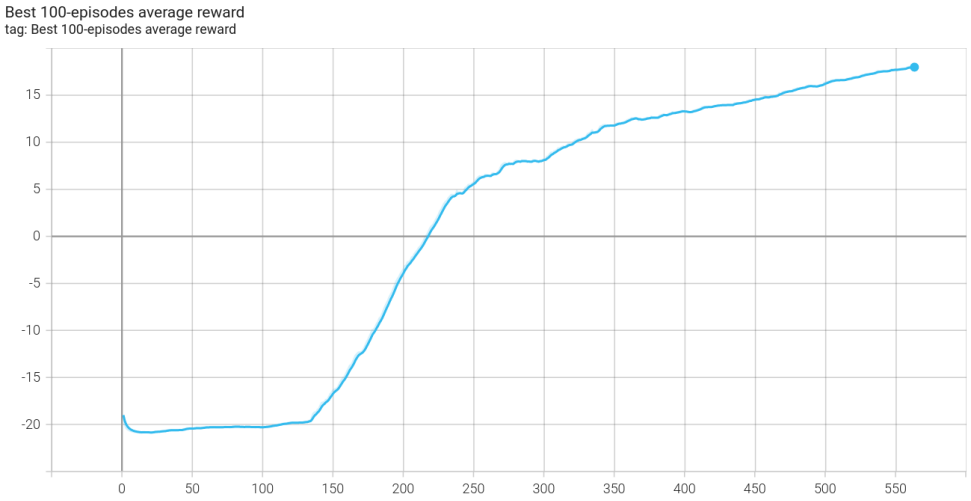
1. DQN

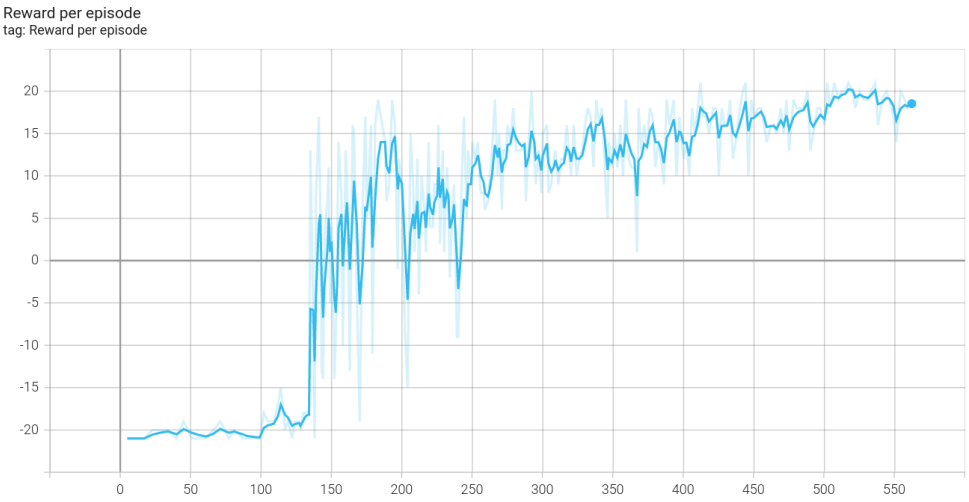
- train

(a) 运行结果

```
Ran 563 episodes best 100-episodes average reward is 18.020000. Solved after 463 trials ✓
```

(b) tensorboard





- test

```
avg reward: 1.140000
```

- gif 见 dqn.gif

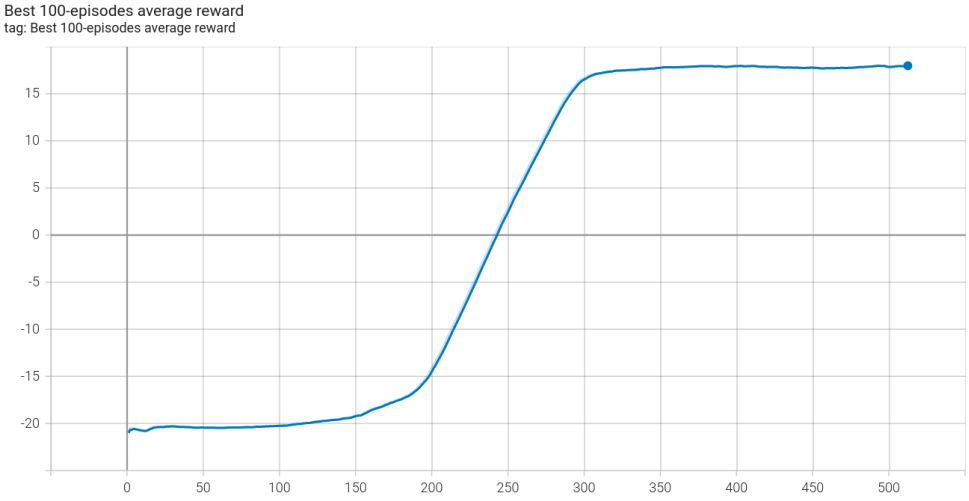
2. Double DQN

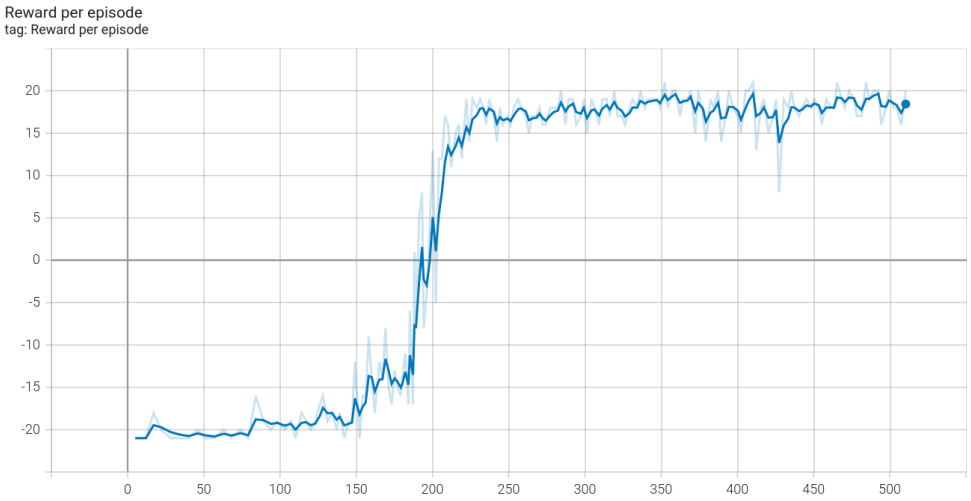
- train

  (a) 运行结果

```
Ran 512 episodes best 100-episodes average reward is 18.020000. Solved after 412 trials ✓
```

  (b) tensorboard





- test

```
avg reward: 1.060000
```

- gif 见 ddqn.gif

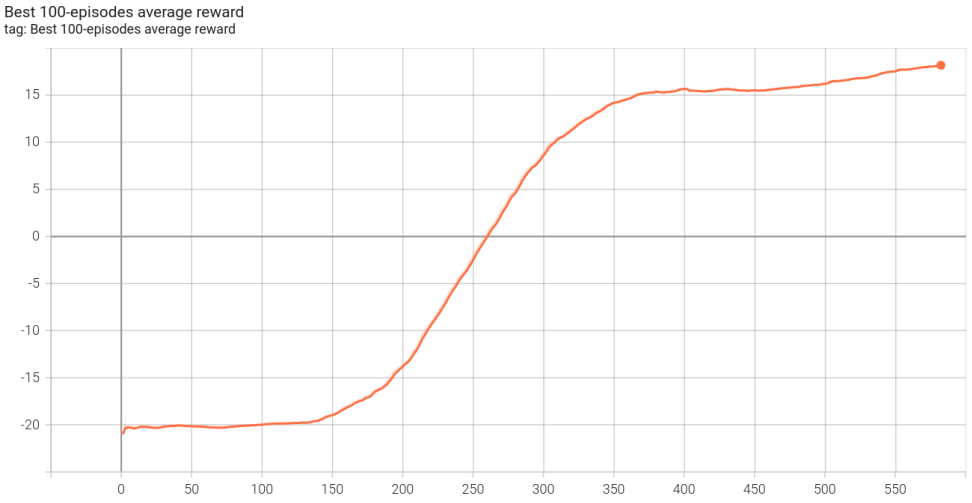
3. Dueling DQN

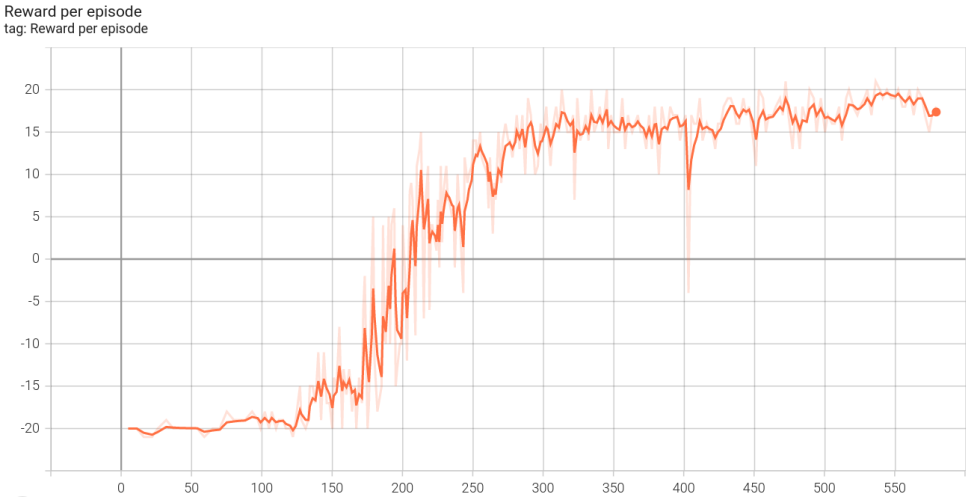
- train

  (a) 运行结果

```
Ran 582 episodes best 100-episodes average reward is 18.230000. Solved after 482 trials ✓
```

(b) tensorboard





- test

- gif 见 dueldqn.gif

5 复现实验

- 为方便助教复现, 我已将代码整合成 3 份, 助教可于各文件夹内通过 tensorboard 或 −−test 进行复现
- DQN
  - tensorboard

```
/dqn$ tensorboard --logdir model
```

  - test

```
/dqn$ python atari_ddqn.py --test --model_path model/model_best.pkl
```

- Doublu DQN
  - tensorboard

```
/ddqn$ tensorboard --logdir model
```

  - test

```
/ddqn$ python atari_ddqn.py --test --model_path model/model_best.pkl
```

- Dueling DQN
  - tensorboard

```
/dueldqn$ tensorboard --logdir model
```

  - test

```
/dueldqn$ python atari_ddqn.py --test --model_path model/model_best.pkl
```

6 小结

- 本次作业帮助我掌握了 tensorflow 与 pytorch 的基本使用, 加深了对深度强化学习的理解
- weight_decay 设置过大会造成模型训练时过于“守旧”, 无法习得新能力, 导致训练效果不良
- 从实验结果可以看出: 相较于 DQN, Double DQN 虽然加速了训练, 但是训练曲线震荡更剧烈, 具体测试表现也稍弱于 DQN; 而 Dueling DQN 则在稳定性上有较大突破, 但是在我设置 (专为 DQN 调整) 的超参下训练稍慢于 DQN