深度 Q 网络（DQN）：原理、公式、应用与实战

2025 年 9 月 21 日

1 引言

深度 Q 网络（Deep Q-Network, DQN）利用深度神经网络近似最优动作价值函数，并通过经验回放与目标网络稳定训练，使强化学习能处理像素级复杂输入。DQN 是深度强化学习迈向高维任务的里程碑。

2 原理与公式

2.1 价值函数逼近

设在线网络 $Q_\theta(s,a)$ 与目标网络 $Q_{\theta^-}(s,a)$，对转移 (s,a,r,s') 的平方 TD 损失为：

$$L(\theta) = \big(r + \gamma \max_{a'} Q_{\theta^-}(s',a') - Q_\theta(s,a)\big)^2. \tag{1}$$

通过从回放缓存 $\mathcal{D}$ 抽样小批量进行梯度下降。

2.2 目标网络更新

目标网络参数周期性或通过 Polyak 平均更新：

$$\theta^- \leftarrow \tau\theta + (1-\tau)\theta^-. \tag{2}$$

缓慢移动的目标缓解了估计震荡问题。

2.3 经验回放

交互产生的转移存入回放缓存，随机抽样打破时序相关性，提高样本利用率。优先级回放则根据 TD 误差加权采样，进一步提升效率。

3 应用与技巧

- **Atari 游戏：**DQN 首次实现像素输入下的超人类表现。
- **机器人与仿真：**离散化动作控制任务。
- **运营优化：**对复杂状态进行决策优化。
- **实用建议：**输入规范化、奖励裁剪、探索率衰减、监控损失和 TD 误差，并使用梯度裁剪保证稳定。

4 Python 实战

脚本 `gen_dqn_figures.py` 在一维连续状态离散动作任务上训练简化 DQN，记录回报曲线与学到的状态-动作价值热力图。

Listing 1: 脚本 $\mathrm{gen}_{d}qn_{f}igures.py$

```
q_target = reward + gamma * np.max(target_network(next_state), axis=0)
q_values = online_network(state)
loss = mse(q_target, q_values[action])
optimizer.zero_grad(); loss.backward(); optimizer.step()
```

5 实验结果

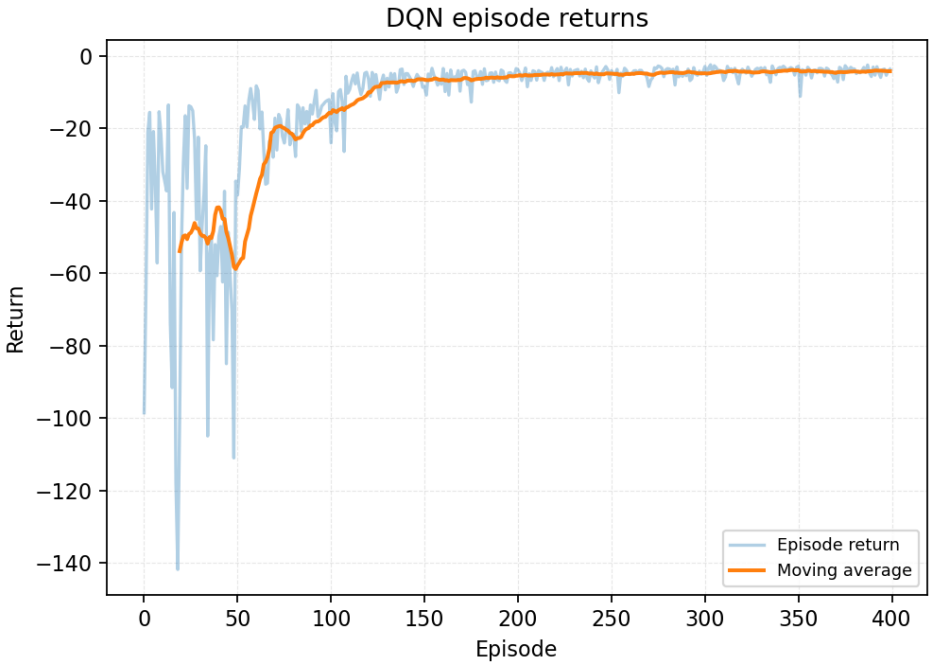

图 1: DQN 训练过程中的回报收敛趋势

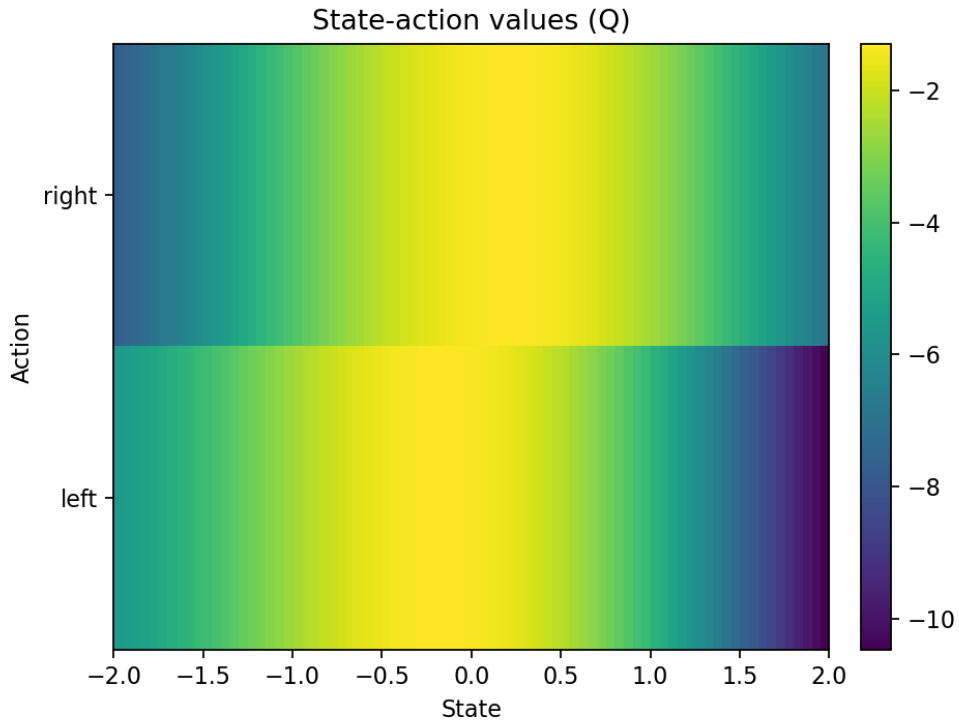

图 2: 学到的状态-动作价值热力图，展示最优策略偏好

6 总结

DQN 通过经验回放与目标网络稳定深度价值学习，适合高维离散动作任务。合理调节学习率、探索率与更新策略是收敛关键。示例说明回报不断提升，且 Q 值热力图反映最优行动偏好。