DRQN 详解

https://zhuanlan.zhihu.com/p/54898904

一、问题引入与 DQN 的不足

传统的 DQN 有两点局限性： 1. 经验数据存储的内存有限。 2. 需要完整的观测信息。

为了解决上述两个问题，设计了 DRQN 算法，将 DQN 中的全连接层替换为 LSTM 网络。

当时用部分观测数据训练模型，使用完全观测数据评估模型时，模型的效果与观测数据的完整性有关。如果反过来，当使用完全观测数据进行训练，使用部分观测数据进行评估时，DRQN 的效果下降小于 DQN。循环网络在观测质量变化的情况下，具有更强的适应性。

(a) Pong (b) Frostbite (c) Double Dunk

Figure 1: Nearly all Atari 2600 games feature moving objects. Given only one frame of input, Pong, Frostbite, and Double Dunk are all POMDPs because a single observation does not reveal the velocity of the ball (Pong, Double Dunk) or the velocity of the icebergs (Frostbite).

知乎 @ECKai

DeepMind 关于 DQN 的原文中，通常 Atari 等游戏，常常通过将最近的 4 帧画面组成一个状态传入 DQN 中进行学习，这是由于仅凭借 1 帧画面很难判断部分物体的运动方向速度等信息，例如在 Pong 的游戏中，凭 1 帧的画面只能获取球跟球拍的位置，无法获取球将要运动的方向与速度，但是 DRQN 则可使用 1 帧替代之前的 4 帧称为一个状态，进行学习决策。但是如果在某些游戏中，4 帧的画面还是无法满足状态的表达，这时就需要循环网络来辅助记忆。因为无法表达当前状态，就使得整个系统不具有马尔科夫性，其 reward 不仅与这帧画面有关，还与前若干帧画面有关。

在部分可观情况下 MDP 变为 POMDP（部分可观马尔可夫决策过程）。在 POMDP 中，如果对 DQN 引入 RNN（循环神经网络）来处理不完全观测将会取得较好的效果。DQRN 相对于 DQN 能够更好的处理缺失的信息。

二、预备知识

1. DQN

DQN 的思想就是设计一个 $Q(s,a|theta)$ 不断逼近真实的 $Q(s,a)$ 函数。其中主要用到了两个技巧：1. 经验回放。2. 目标网络。该技巧主要用来打破数据之间联系，因为神经网络对数据的假设是独立同分布，而 MDP 过程的数据前后有关联。打破数据的联系可以更好地拟合 $Q(s,a)$ 函数。其代价函数为：

$$L(/theta)+=+E_{s,a,r,s'}[(Q(s,a|/theta)-y)^2],/qquad+/rmwhere/+y=r+/gamma+max_{a'}/overline+Q(s',a'|/overline+/theta)$$

其中 $/overline+Q(s',a'|/overline+/theta)$ 表示目标网络，其参数更新与 $/theta$ 不同步（滞后）。具体可以参看值函数强化学习-DQN、DDQN 和 Dueling DQN 算法公式推导分析。

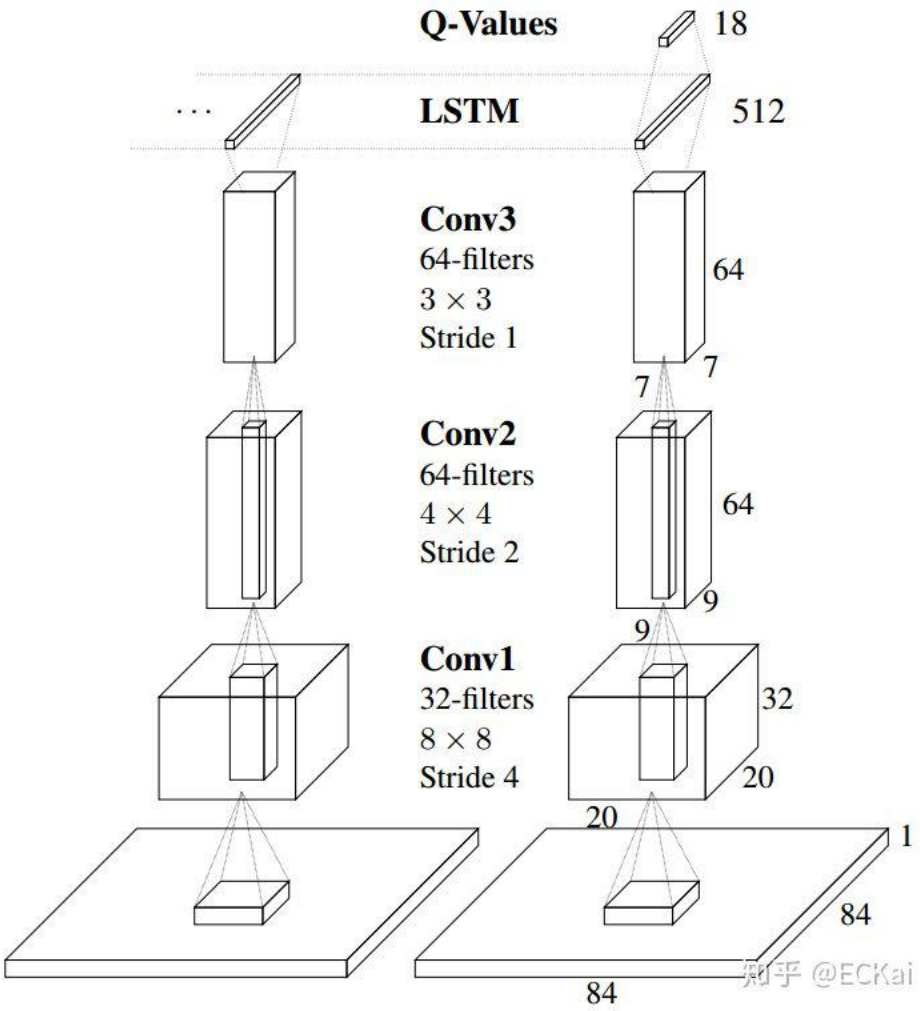
2. 部分可观性

在实际环境中，智能体很少能获得完整的状态信息。因此也就失去了马尔科夫性。部分可观马尔可夫决策过程（POMDP）就能很好的表达这种状态无法完全获取的动态特性，其定义观测 o 为状态 s 的观测值，其可以用一个函数表示为 $o_t \sim \mathcal{O}(s_t)$。一个 POMDP 可以被表示为 (S,A,P,R,Ω,O)，S,A,P,R 分别表示状态、动作、状态转移概率、奖励，智能体在每一步不在接收状态 s 而是收到观测 o，观测是底层的系统状态经过概率分布 $o_t \sim \mathcal{O}(s_t)$ 得到的。如果在 POMDP 中使用 DQN 将不能很好地逼近 Q 函数，这是由于 $Q(o,a|\theta) \neq Q(s,a|\theta)$。通过文章中的实验能够观察出，引入了 RNN 的 DRQN 能够更好地处理部分可观的情况，DRQN 能够更好的逼近实际的 $Q(s,a|\theta)$ 以至于学习到更优秀的策略。

三、DRQN 设计

1. 结构设计

DRQN 最小程度的修改 DQN 的结构，只将卷基层后一层的全连接层替换为了 LSTM 网络，最终输出结果为每个动作 a 对应的 $Q(s,a)$ 值。在训练的过程中，卷积部分与循环网络部分一同更新迭代学习。其结构如下图所示：



2. 更新方式

每次更新循环网络，需要包含一段时间连续的若干观测 与奖励值 。此外，在每次训练时，LSTM 隐含层的初始状态可以是 0，也可以从上一次的值继承过来。因此具有两种更新学习方式：

a. Bootstrapped 序列更新

从经验回放内存中随机选取一次游戏过程（episode），从这次游戏过程的开始一直学习到游戏结束。在每一个时刻 t，其目标状态值还是通过目标网络 /overline + Q(/overline + /theta) 来获取。在一次游戏过程中，每一时刻 LSTM 隐含层的状态值从上一时刻继承而来。

b. Bootstrapped 随机更新

从经验回放内存中随机选取一次游戏过程（episode），再在这个游戏过程中随机选择一个时刻点，再选择若干步进行学习（可以是一步）。在每一个时刻 t，其目标状态值还是通过目标网络 /overline + Q(/overline + /theta) 来获取。在每一次训练前需要将 LSTM 隐含层的状态置为 0。

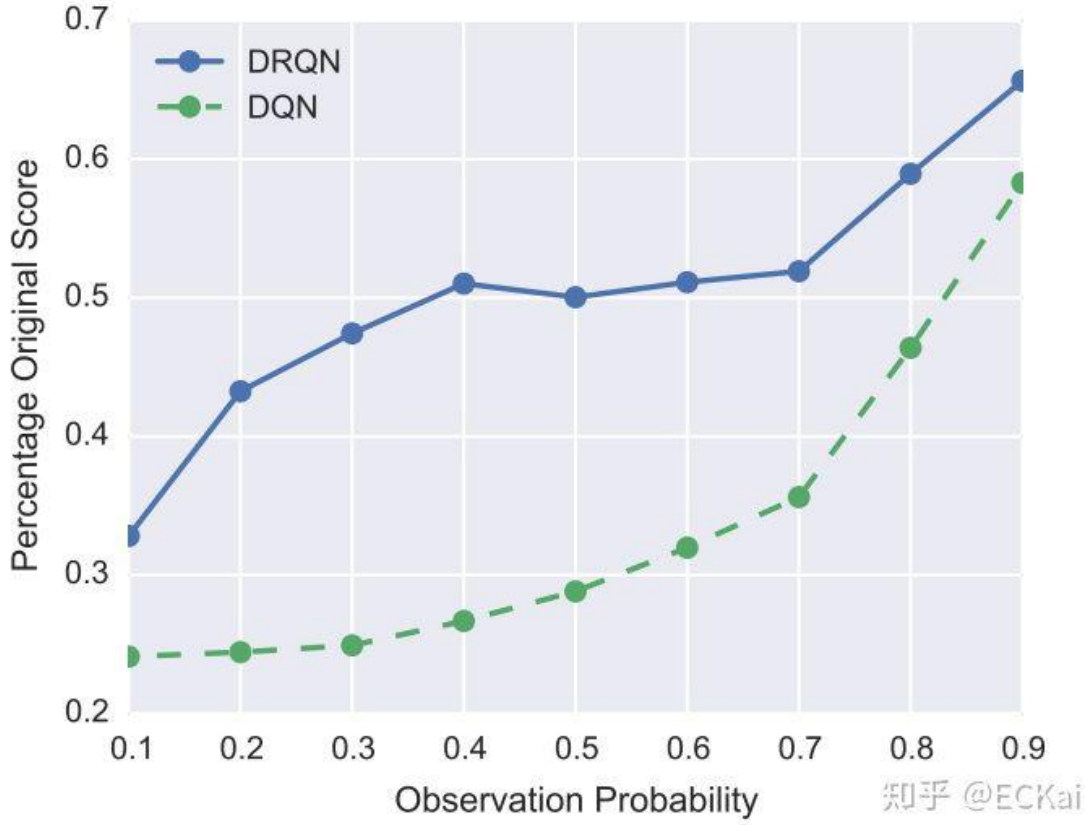
序列更新能够更好的让 LSTM 学习一次游戏过程的所有时间序列记忆，更有利于时序推理。但是由于序列化采样学习，违背了 DQN 随机采样的策略（因为神经网络要求学习数据独立同分布，由于时序数据之间有马尔科夫性，则会损害神经网络的学习效果）。

随机更新更符合 DQN 的随机采样策略，但是需要每次更新前将 LSTM 隐含层状态置为 0，这将损害 LSTM 的记忆能力。实验证明这两种更新方法都能得到收敛的策略以及相似的效果。原文中主要采用随机更新的方法。

在仿真阶段，原文采用 0.5 的概率对画面进行模糊处理来模拟部分可观的情景。对比实验为，DQN 输入为连续的 4 帧画面，而 DRQN 输入为 1 帧画面。DRQN 更善于利用循环记忆来完善部分观测信息，推理出完整的状态信息。因此，DRQN 可以是一种 DQN 输入多帧的一种替代算法。

四、MDP 到 POMDP 的一般化过程

原文作者想要测试使用完全观测数据训练 DQN 与 DRQN，然后再使用部分观测数据评估 DQN 与 DRQN。通过在 9 个游戏中进行测试得出平均结果如下图所示



DRQN 在信息逐渐缺失的情况下，其效果下降小于 DQN，说明其对缺失信息更具鲁棒性。