

***写在前面***

*本文是对深度强化学习的记忆体系的研究，文章最早发布于2015年底，也是AlphaGo快要出炉的前夜。在我讲解过的强化学习课程中，前期总是会使用普通的数组来作为AI的记忆空间，因为学生会很容易理解。不过一旦需要使用像Atari这样相对复杂的环境来完成深度强化学习的实验时，数组就会很不靠谱。这种问题通常跟语言有直接的关系，例如Python语言的List，当你不断存入内容时，内存占用会越来越明显，最终系统会疲于写缓存导致卡死。而相较之下C++则对控制内存更有优势，但无论提供怎样合理的内存控制方式，都不如今天这篇文章更加优秀。前文我们描述了一种叫做DQN的深度强化学习模型，该模型在人工智能（AI）玩游戏的领域有所突破，接下来就请看看本文是如何做出改进的。*

Abstract

时至今日深度强化学习已经能够胜任复杂的控制器角色了，但像DQN这样最新的研究所凌驾的控制器其脑容量却非常有限。为了更好的改善这一缺陷，本文为DQN模型加入了循环神经网络元素，使用循环长短期记忆（LSTM）层替代了原DQN模型在卷积之后的那个全连接层。最终实现的模型命名为Deep Recurrent Q-Network.（DRQN），这个模型依旧在每个时间步观看一帧游戏过程，在时间上能够成功整合信息，完美复现了DQN在Atari 游戏上的表现，且一些游戏过程采用了局部观察机制就能够替代对整个屏幕的检测。

更让人惊喜的是，当使用部分观察机制进行训练，然后进行逐步趋于完整的观察机制用于验证时，我们发现DRQN的性能会随着这种观察机制同步变化。相反的，使用完全观察机制进行训练与部分观察机制进行验证，DRQN的性能会低于DQN。因此，给定相同的记忆长度，可循环网络是DQN在输入层直接堆叠历史帧记忆的可行替代方案，而当携带了可循环性的DQN在玩游戏时，这种循环能力不会有作弊嫌疑，这就是网络自身的计算能力，如果验证时观察质量经常发生变化，循环性网络则具备更好的适应能力。

1 INTRODUCTION

DQN是一种能够学习人类级别控制策略的深度强化学习模型，目前已在Atari2600游戏环境获得不错的成绩。正如其名，DQN尝试为每个游戏状态估计做出每个动作选择的价值，称之为Q值。如果给定的Q值具有较高的准确率，那么以DQN模型为大脑的AI程序就能够选择概率最高的动作作为优选动作。所以DQN的功能就是将游戏画面经过处理后制作为游戏状态，与游戏按键的动作空间进行映射，就目前来看这种映射关系使得DQN模型在应对Atari 2600游戏时非常凑效，玩出了目前的最优性能。

尽管如此，由于DQN训练时需要大量累积过去的历史游玩记录，而面对不断累积的历史游玩记录，DQN却只能选取最近期的4帧作为图像输入，所以一旦遇到对玩家记忆要求较高的游戏（例如对打游戏的组合招数通常会大于4帧），DQN的性能就会严重受限。这是因为马尔科夫决策过程并不能处理输入少于输出（未来状态与对应奖励）的依赖关系，就像大自然一样，种下一粒米怎能获得成吨的收入呢？因此对于这种输入不完整的马尔科夫决策过程，我们称为局部视界的马尔科夫决策过程（Partially-Observable Markov Decision Process ，POMDP）

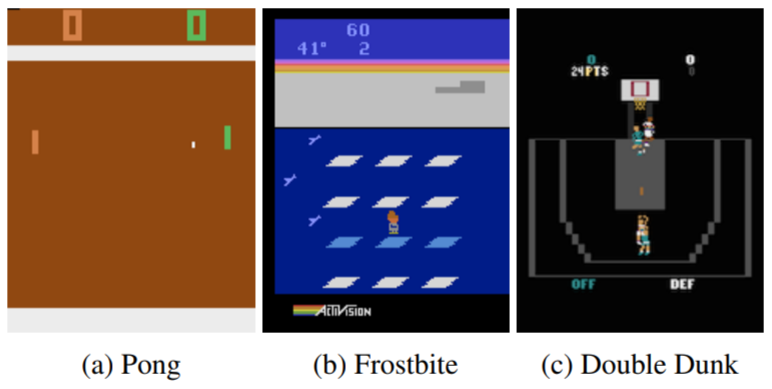


图1

现实世界中的人们的各种活动过程都充斥着特征缺失与噪声状态信息，这些都是由于局部的数据观察方式所导致的，就跟看电视剧一样，不沉下心来追剧，单凭1分钟的观看时间是很难了解剧情的。图1 就显示了众多属于POMDP的Atari 2600游戏。其中一个明显的例子就是乒乓球游戏（图1a），单从画面进行观察仅仅能知道球拍和球的位置，但是对球的速度完全无感。然而根据球的速度对运行轨迹进行估计是这个游戏取胜的关键。

我们在实验中观察到DQN的性能确实会下降，并推测如果搭配循环神经网络（Recurrent Neural Networks，RNN）会让DQN更加从容的处理POMDP形式的环境。因此我们研发了DRQN，这种将LSTM于DQN结合在一起的新模型，与此同时我们在本文证明了DRQN是完全有能力应对局部观察限制的，并且我们还做了使用全面观察进行训练、用局部观察进行评价的实验，结果证明DRQN能够不DQN在这种情况下更好的控制误差。最终，所有的实验都证明了，随着观察质量的下降，循环能力的重要性不断提高。

**Deep Q-Learning**

强化学习是一种让AI与环境进行互动，进而让其学会针对于环境的控制策略的一种机器学习方式。能够进行强化学习的环境通常符合马尔科夫决策过程（Markov Decision Processes ，MDPs），这样与环境的互动可以用4个字来概括就是：S，A，P，R。在每个时间步（游戏的每一帧画面过程），AI会陷入一个状态S，根据已知策略P选择一个合适的动作A，从而得到环境给予的奖励R。

Q学习（Q-Learning）是一个无模型（model-free）与并行策略（off-police），以追求长期最大奖励为目标的强化学习算法，深度Q学习构建的神经网络算法就是DQN，其具体内容在上一篇文章已经讲解过了，在这里就不再赘述了，总结一下就是根据环境计算Q值；根据Q值、目标策略输出值与奖励值计算误差；优化误差以更新神经玩过的训练参数（权重），反复这个过程。

**Partial Observability**

在现实场景中，能够进行完全观测的系统并不常见，而能让AI来完全观察的系统更需要专门定制，可能在未来高速定制技术会越来越普及，但就目前来看为了训练一个AI而花大代价定制环境系统实在不是所有人都能接受的事情。换句话说，MDP在真实世界中可能并不完美存在，但POMDP却处处可见，因为它能更容易的捕捉到真实环境中的状态，也就是更容易获取数据。DQN对环境的完全观测会产生一个4元组（S，A，P，R），那么DRQN对环境的部分观测机制则定义为一个6元组（S，A，P，R，Ω，O）。其中前4个依然代表状态、动作、策略与奖励，而Ω则代表观察集，O代表获得这部分观察数据的概率分布。即现在不是直接从所有状态集合S来生成学习序列数据了，通过概率分布O来按概率生成S的部分集合Ω。

2 DRQN Architecture

为了凸显循环网络的效果，我们对DQN结构做了尽可能微小的调整——仅仅将第一层全连接层替换为LSTM层，并保持尺寸相同（都具有512个神经元）。图2是具体结构描述，我们可以看到新结构依然对84X84像素的图片进行处理，这与DQN完全相同，图片进入网络后经过3个卷积层，之后到达LSTM层，最后由一个线性输出层给出与按键数量一致的Q值（注意根据游戏不同，按键数量会有不同，不一定是18）。训练阶段卷积层与LSTM层的参数同时被更新，针对这种新结构我们做了许多不同的实验，详见原文附录A。

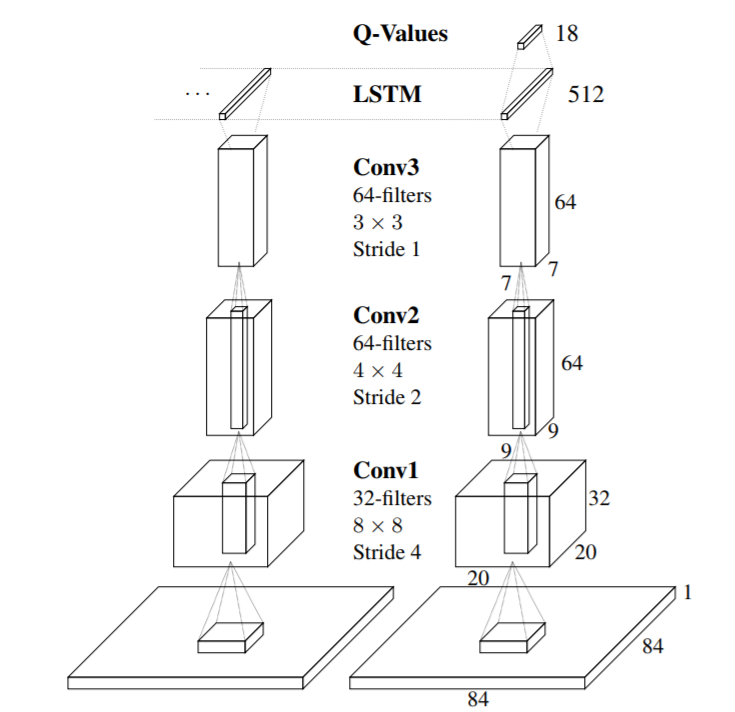


图2

3 Stable Recurrent Updates

更新一个循环+卷积网络，需要反向传播过程计算包含多个时间步的游戏截图与目标值。另外，LSTM的初始隐含状态可以全零起步或继承前任值，我们为这个新模型的更新考虑了这两种更新方式：

**Bootstrapped Sequential Updates**:首先从记忆序列中随机选择游戏经历，然后从这次游戏经历的开头进行学习并更新，一直学习到这次游戏结束为止，LSTM状态值在所有前向传播过程中进行覆盖。每个时间步计算所使用的目标值来自于目标Q网络（target Q-network）。

**Bootstrapped Random Updates**：首先从记忆序列中随机选择游戏经历，然后从这次游戏经历的随机时刻进行学习，仅在完整运行的那次迭代对LSTM的状态值进行覆盖（例如拥有反向传播的计算过程），每个时间步所使用的目标值依然来自于target Q-network，不过LSTM的初始状态为零。

序列更新的优势在于能将LSTM状态很好的从游戏开始保持到游戏结束，提供了较为完整的记忆，不过通常DQN都需要从大量的历史记录中随机选择记忆然后用来训练，所以如果使用序列更细则会违背这一随机形式。

随机更新是非常符合随机选取记忆的方式的，但同样由于LSTM的状态值在每次游戏开始时必须被清零，所以可能会导致网络对长期游戏过程不敏感，毕竟按照算法要相隔好一段时间步才开始做一次反向传播呢。

然而实验表明，这两种类型的更新都是可行的，并且在一组游戏中产生具有相似性能的融合策略。因此，为了限制复杂性，本文中所有实验结果都使用自举随机更新法。我们希望所呈现的结果都能够合理推广到自举序列更新法。

在描述清楚DRQN的结构和更新方法之后，接下来就展示一下它在具有局部观察环境下的实验结果。

4 Atari Games: MDP or POMDP?

Atari 2600游戏的所有游戏状态由128字节的控制台RAM描述。然而人类和AI只观察控制台生成的游戏屏幕。对于许多游戏，单从屏幕截图不足以确定系统的状态。通过选取最后四个游戏截图来推断Atari游戏的完整状态，DQN就是依靠这个方法将许多游戏从POMDP变成了MDP。在Mnih等人于2015年调查的49个游戏中，连游戏作者在仅给予最后四帧图像的情况下都无法识别改游戏术语POMDP还是MDP。给定4帧的方法适用于MDP，那么面对POMDP我们就需要另一种方法来给予DRQN足够的输入帧数量。

5 Flickering Atari Games

为了解决上述问题，我们改造了游戏，例如讲乒乓球游戏改为闪烁版乒乓球游戏。在闪烁版的乒乓球游戏过程中，每一帧的游戏画面都有50%的概率变化为模糊状态。这样就导致了但凡对模糊的画面产生了记忆，那么这种记忆就是来源于不完整的观察。

面对闪烁版乒乓球难题，对游戏帧的信息整合就显得尤为重要，毕竟只有相对完整的信息才能准确推断出乒乓球与球拍的即时位置与速度。所以当接近一半的帧记忆被模糊时，成功的AI必须能够对这种模糊输入产生鲁棒性。

能解决这个难题的核心方案也许就藏在卷积运算中，你觉得呢？图3所示的游戏过程是对不同卷积层所输出的最大游戏动作选择进行了可视化，其时间范围截取为10帧，我们可以看到如果给予10帧的数据，确实能够估计速度

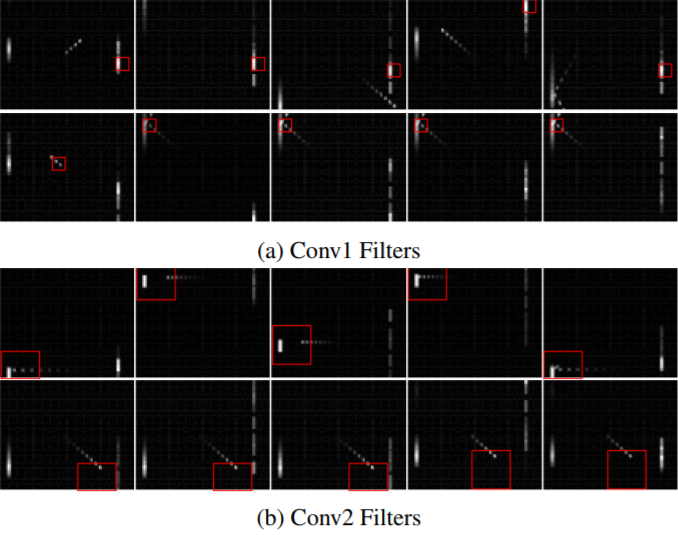


图3上半部分

令人惊叹的是，DRQN面对这样的游戏，如果每次仅给予1帧画面，也能玩下去。当给予AI仅1帧画面时，卷积层就无法解析到速度了，因为它获得的仅仅是静止的单独画面，重任全部压在LSTM的肩头。现在的局面就是每个单独的时间会有1帧的数据，图3d很好的证明了LSTM能够从时间线上重组残破的数据，盯着模糊画面与处理着1帧数据在努力进行游戏。这种数据重组过程能够有效检测一些事件，例如丢球了、球拍击中球了、球在墙面上反弹了。

经过我们的实验，发现使用10帧数据的DQN和1帧数据的DRQN效果类似（局部观测机制下），于是这里就产生了两个选择，究竟哪个方法更好呢？我们唯一可以解答的就是，DRQN能够在时间线上重组数据，所以不需要使用给帧做打包的那个步骤了，速度上更具优势。

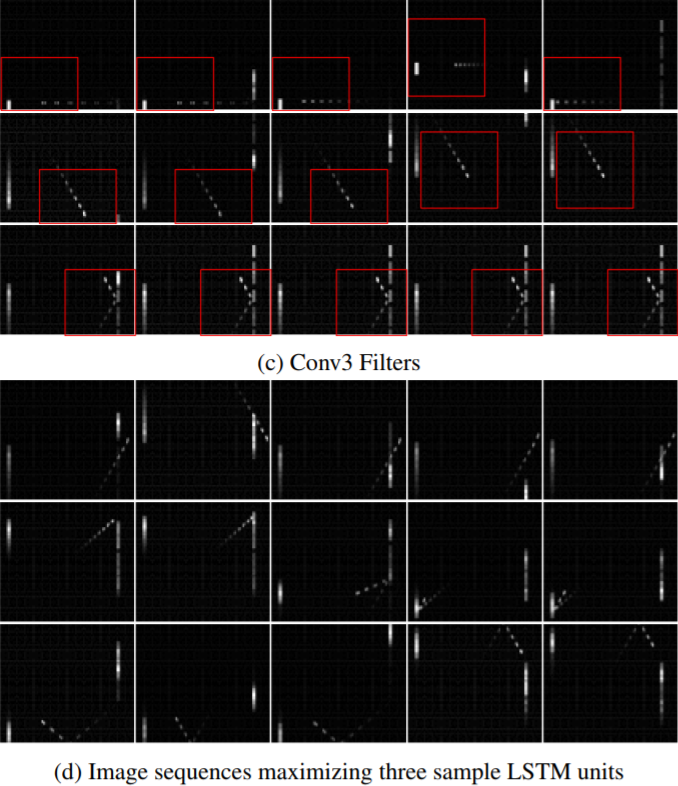


图3下半部分

6 Evaluation on Standard Atari Games

我们选取了下列9个游戏作为评估项目：

Asteroids; Double Dunk; Beam Rider; Centipede; Chopper Command; Frostbite; Hockey; Bowling; Ms Pacman

这一次实验我们将帧数锁定为4帧，方便与DQN进行比较。表1说明了从平均数据看来，DRQN是略优于DQN的。尤其在Frostbite与Double Dunk这两个游戏中优势比较明显，但是在Beam Rider游戏的表现上比较逊色，具体见图4的描述。

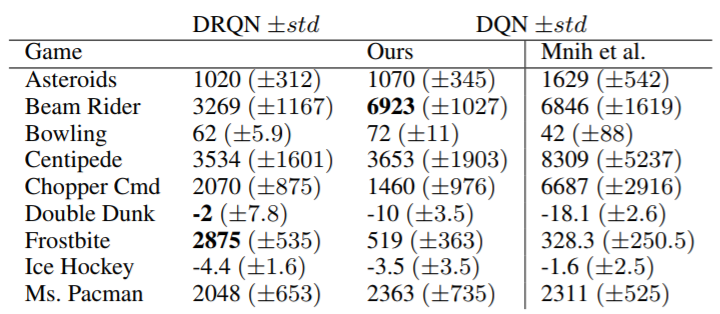


表1

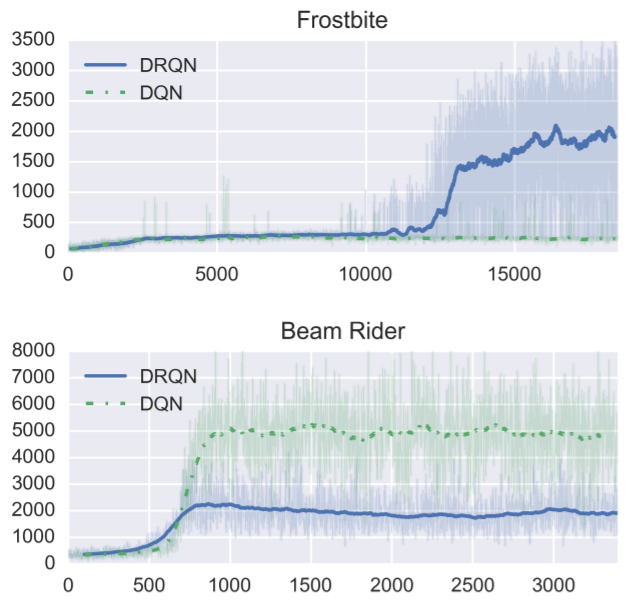


图4

7 MDP to POMDP Generalization

如果一个DRQN模型在训练时给予清晰图像，而评估其性能时给予模糊图像会发生什么事情呢？图5就是答案，横轴是观察完整度的概率，纵轴就是分数降比后的结果。可以看到观察完整度在40%时DRQN具有很强的优势。

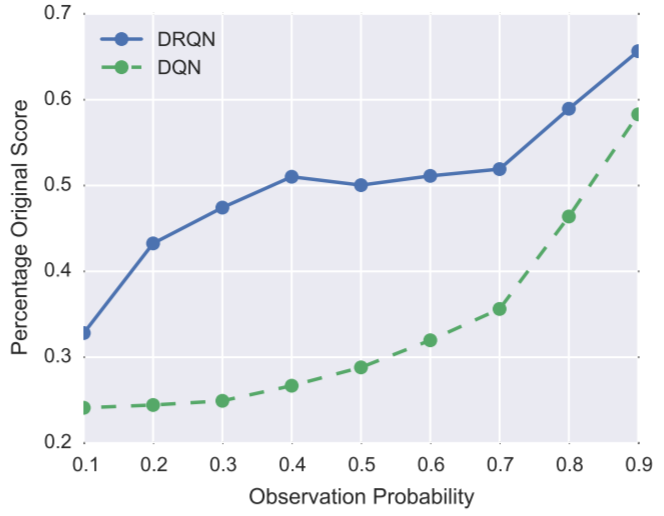


图5

8 Related Work

迄今为止，LSTM网络可以被证明为能够解决POMDP问题的有效方法，当然还有没提到的隐含英雄们，强化学习中的策略梯度下降与时间差分更新方法，结合这些所有的方法与卷积网络，我们的AI已经越来越像一个人类，它仅仅接受图像的输入，甚至是不太好的图像，然后就能够玩游戏了！

与此同时也有人将强化学习与LSTM结合来玩文本游戏了，虽然领域不同但有异曲同工之妙，文本游戏的状态总量非常少，远低于视频游戏。

9 Discussion and Conclusion

现实世界中形成POMDP的原因大多是不完整的数据特征与复杂的数据噪音，即便我们能在现实社会里看尽众生相，也看不完所有的相面。相信在未来会有更多以DRQN为原型的创造，让AI能更加完善的生存在这个不完整的世界里。