# **通过深度强化学习实现人类水平的控制**

## 摘要

强化学习理论提供了一种规范性的解释，其深刻植根于心理学和神经科学对动物行为的研究视角，阐述了智能体如何优化对环境的控制。然而，要在接近现实世界复杂性的场景中成功应用强化学习，智能体面临着一项艰巨的任务：它们必须从高维感官输入中提取环境的有效表征，并利用这些表征将过去的经历推广到新情境中。令人惊讶的是，人类和其他动物似乎通过强化学习与层次化感官处理系统的和谐结合解决了这一问题，其中强化学习的证据来自于大量神经数据，这些数据揭示了多巴胺能神经元发出的时相信号与时间差分强化学习算法之间的显著相似性。尽管强化学习智能体在多个领域取得了一些成功，但其适用范围此前仅限于能够手工设计有用特征的领域，或完全可观测、低维状态空间的领域。

在此，我们利用近期在深度神经网络训练方面的进展，开发了一种名为深度Q网络（DQN）的新型人工智能体，该智能体能够直接从高维感官输入中通过端到端的强化学习学会成功的策略。我们在经典的Atari 2600游戏这一具有挑战性的领域对该智能体进行了测试。我们证明了深度Q网络智能体仅以像素和游戏得分为输入，便能够在49款游戏中超越所有先前算法的性能，并达到与专业人类游戏测试者相当的水平，且在整个过程中使用了相同的算法、网络架构和超参数。这项工作弥合了高维感官输入与动作之间的鸿沟，实现了首个能够在多样化且具有挑战性的任务中表现出色的人工智能体。

## 介绍

我们致力于开发一种单一算法，使其能够在多种具有挑战性的任务中发展出广泛的技能，这是一般人工智能的核心目标，此前的研究一直未能实现。为了达成这一目标，我们开发了一种新型智能体——深度Q网络（DQN），它能够将强化学习与一类称为深度神经网络的人工神经网络结合起来。值得注意的是，近期深度神经网络的进展使得人工神经网络能够直接从原始感官数据中学习诸如物体类别等概念。我们采用了一种特别成功的架构——深度卷积网络，它利用分层的平铺卷积滤波器来模拟感受野的效果，这一设计灵感来源于Hubel和Wiesel关于初级视觉皮层前馈处理的开创性工作，从而利用了图像中存在的局部空间相关性，并对自然变换（如视角或尺度的变化）具有鲁棒性。

我们考虑的任务是智能体通过一系列的观察、动作和奖励与环境进行交互。智能体的目标是以一种最大化累积未来奖励的方式选择动作。更正式地说，我们使用深度卷积神经网络来近似最优动作价值函数：

*Q*∗(*s*,*a*)≈*π*max​E[*t*=0∑∞​*γtrt*​∣*st*​=*s*,*at*​=*a*,*π*],

其中 *rt*​ 是在每个时间步 *t* 通过折扣因子 *γ* 折现的奖励之和，这一和可以通过行为策略 *π*=*P*(*a*∣*s*) 实现，该策略在进行观察（*s*）和采取动作（*a*）之后选择动作（见方法部分）。

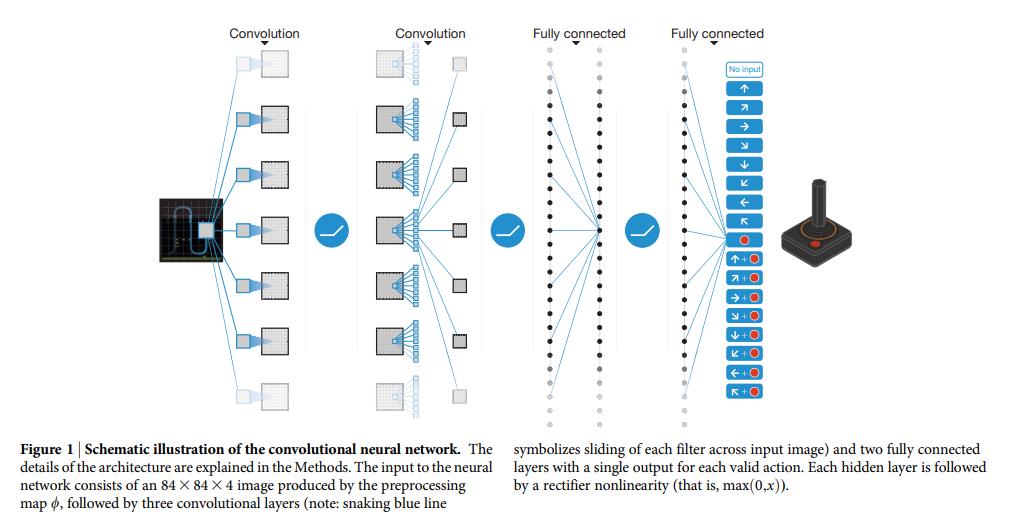
当使用非线性函数逼近器（如神经网络）来表示动作价值（也称为Q）函数时，强化学习已知是不稳定的，甚至可能发散。这种不稳定性有多种原因：观察序列中存在的相关性、对Q的小更新可能会显著改变策略从而改变数据分布，以及动作价值（Q）与目标值 *r*+*γ*max*a*′​*Q*(*s*′,*a*′) 之间的相关性。

我们通过一种新颖的Q学习变体来解决这些不稳定性问题，该变体采用了两个关键思想。首先，我们采用了一种受生物学启发的机制，称为经验回放，它通过对数据进行随机化，从而消除了观察序列中的相关性，并平滑了数据分布的变化（详细内容见下文）。其次，我们采用了一种迭代更新方法，将动作价值（Q）调整为仅定期更新的目标值，从而减少了与目标值的相关性。

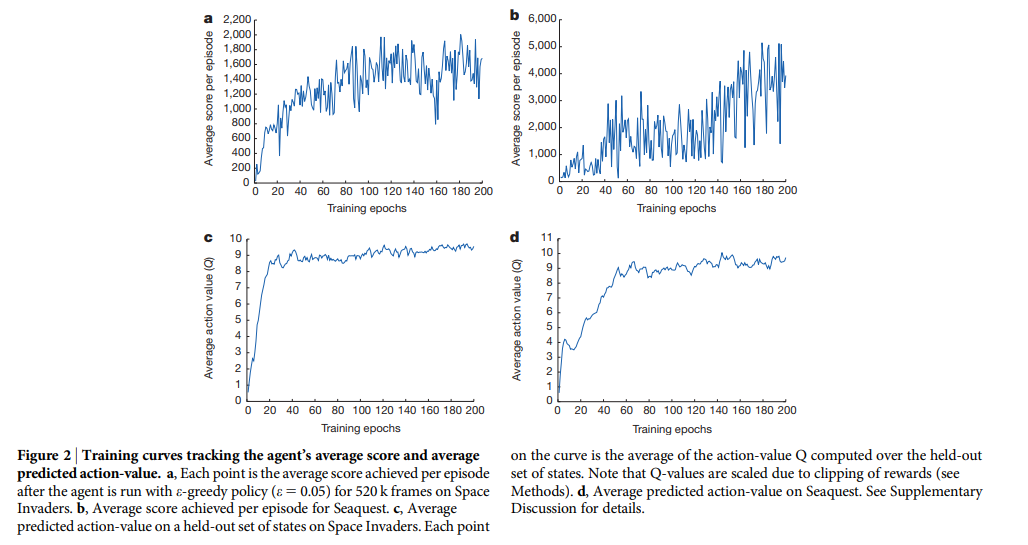
虽然在强化学习设置中训练神经网络还存在其他稳定的方法，例如神经拟合Q迭代，但这些方法需要在数百次迭代中反复从头开始训练网络。因此，与我们的算法不同，这些方法对于大型神经网络来说效率过低，无法成功应用。我们使用图1所示的深度卷积神经网络对近似价值函数 *Q*(*s*,*a*;*θi*​) 进行参数化，其中 *θi*​ 是第 *i* 次迭代时Q网络的参数（即权重）。为了执行经验回放，我们在每个时间步 *t* 将智能体的经验 *et*​=(*st*​,*at*​,*rt*​,*st*+1​) 存储在数据集 *Dt*​={*e*1​,…,*et*​} 中。在学习过程中，我们在从存储样本池中均匀随机抽取的样本（或小批量样本）(*s*,*a*,*r*,*s*′)∈*U*(*D*) 上应用Q学习更新。第 *i* 次迭代的Q学习更新使用以下损失函数：

*Li*​(*θi*​)=E(*s*,*a*,*r*,*s*′)∼*U*(*D*)​[(*r*+*γa*′max​*Q*(*s*′,*a*′;*θi*−​)−*Q*(*s*,*a*;*θi*​))2],

其中 *γ* 是决定智能体视野范围的折扣因子，*θi*​ 是第 *i* 次迭代时Q网络的参数，而 *θi*−​ 是用于计算第 *i* 次迭代目标值的网络参数。目标网络参数 *θi*−​ 每隔 *C* 步才用Q网络参数 (*θi*​) 更新一次，并在两次更新之间保持固定（见方法部分）。

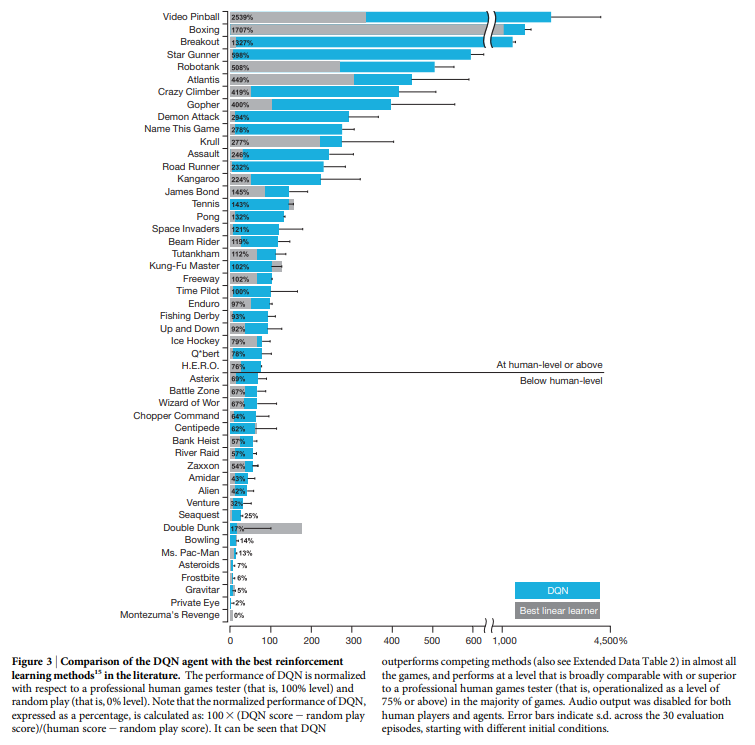


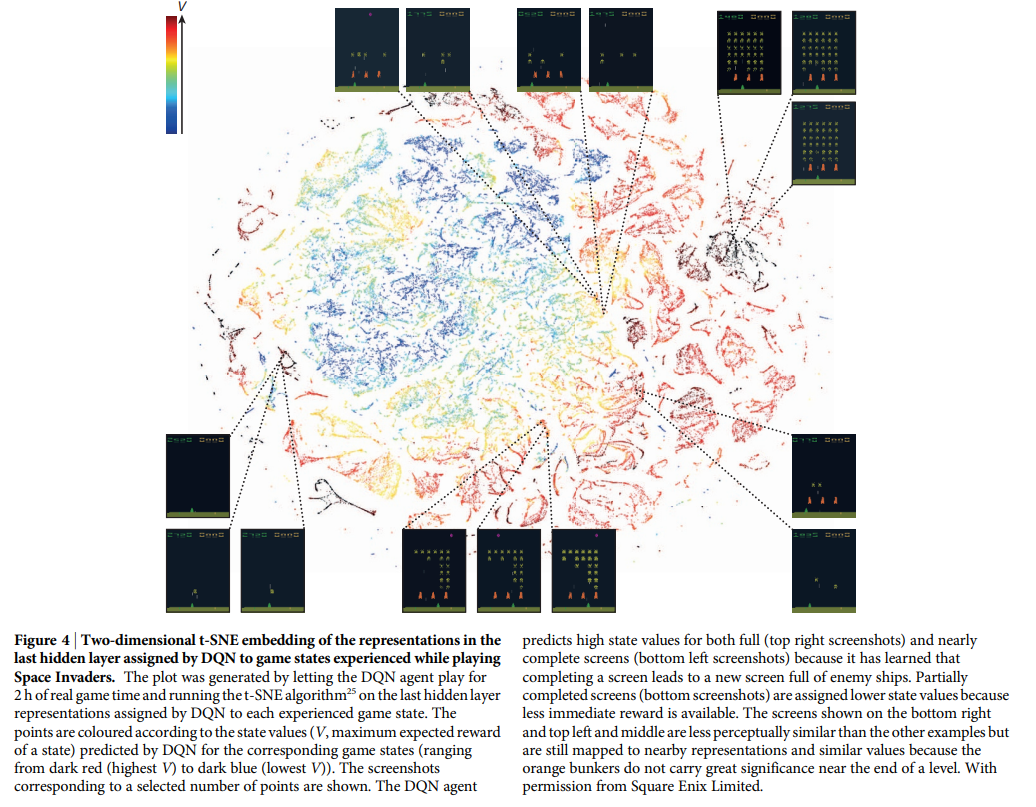
为了评估我们的深度Q网络（DQN）智能体，我们利用了Atari 2600平台，该平台提供了49种多样化且具有挑战性的任务，这些任务被设计为对人类玩家具有较高的难度和吸引力。我们在整个实验中使用了相同的网络架构、超参数值（见扩展数据表1）和学习过程——输入数据为高维的210×160像素、60Hz刷新率的彩色视频——以证明我们的方法能够仅基于感官输入（几乎没有任何先验知识，即仅输入为视觉图像，以及每款游戏中可用的动作数量，但不包括其对应关系；详见方法部分）在多种游戏中稳健地学习到成功的策略。值得注意的是，我们的方法能够以稳定的方式使用强化学习信号和随机梯度下降训练大型神经网络，这一点通过两个学习指标的时间演变得到了体现（智能体的平均每局得分和平均预测Q值；详见图2和补充讨论部分）。



我们将DQN与强化学习文献中在49款游戏中表现最佳的方法进行了比较（这些游戏中已有结果可供参考）。除了学习到的智能体外，我们还报告了在受控条件下进行游戏的专业人类游戏测试者的得分，以及随机选择动作的策略的得分（扩展数据表2和图3，分别用y轴上的100%（人类）和0%（随机）表示；详见方法部分）。我们的DQN方法在43款游戏中优于现有的最佳强化学习方法，而无需使用其他方法（例如文献12和15）中针对Atari 2600游戏所采用的额外先验知识。此外，我们的DQN智能体在49款游戏的整体表现上达到了与专业人类游戏测试者相当的水平，在超过一半的游戏（29款游戏）中达到了人类得分的75%以上（见图3、补充讨论和扩展数据表2）。在额外的模拟实验中（见补充讨论以及扩展数据表3和4），我们通过禁用DQN智能体的各个核心组件（即回放记忆、独立的目标Q网络以及深度卷积网络架构），展示了这些组件对性能的不利影响，从而证明了它们的重要性。

接下来，我们通过“t-SNE”技术（一种用于高维数据可视化的技术）来研究DQN在游戏《太空侵略者》中所学习到的表征，这些表征是智能体成功表现的基础（有关DQN表现的演示，请参阅补充视频1，见图4）。正如预期的那样，t-SNE算法倾向于将感知上相似的状态的DQN表征映射到相邻的点。有趣的是，我们还发现了一些实例，其中t-SNE算法为预期奖励相近但感知上不相似的状态的DQN表征生成了类似的嵌入（见图4右下角、左上角和中部），这与网络能够从高维感官输入中学习支持适应性行为的表征的观点一致。此外，我们还展示了DQN所学习到的表征能够泛化到其自身策略之外的数据——在模拟实验中，我们将人类和智能体游戏过程中经历的游戏状态输入到网络中，记录了最后一层隐藏层的表征，并通过t-SNE算法可视化生成的嵌入（扩展数据图1和补充讨论）。扩展数据图2进一步说明了DQN所学习到的表征如何使其能够准确预测状态和动作的价值。





值得注意的是，DQN表现出色的游戏在性质上极为多样，从横向卷轴射击游戏（例如《Riverraid》）到拳击游戏（例如《Boxing》）以及三维赛车游戏（例如《Enduro》）。事实上，在某些游戏中，DQN能够发现相对长期的策略。例如，在《Breakout》中，智能体学会了最优策略，即首先在墙的侧面挖一条隧道，使球能够绕到墙后方，从而摧毁大量砖块（有关DQN在训练过程中性能发展的演示，请参阅补充视频2）。然而，对于需要更长时间跨度规划策略的游戏，仍然是包括DQN在内的所有现有智能体面临的主要挑战，例如《Montezuma's Revenge》。

在本研究中，我们证明了一种单一架构能够在多种不同环境中成功地学习控制策略，仅需极小的先验知识。该架构仅以像素和游戏得分为输入，并在每款游戏中使用相同的算法、网络架构和超参数，仅依赖于人类玩家所拥有的输入信息。与以往的研究不同，我们的方法采用了“端到端”的强化学习，利用奖励信号持续塑造卷积网络内的表征，使其向环境中有助于价值估计的关键特征靠拢。这一原理借鉴了神经生物学的证据，即在知觉学习过程中，奖励信号可能会影响灵长类动物视觉皮层内表征的特征。

值得注意的是，强化学习与深度网络架构的成功整合，关键依赖于我们引入的一种回放算法，该算法涉及对最近经历的转换进行存储和表示。越来越多的证据表明，海马体可能在哺乳动物大脑中支持这一过程的物理实现，即在离线期间（例如清醒休息时）对最近经历的轨迹进行时间压缩的再激活，这可能是通过与基底神经节的相互作用高效更新价值函数的一种潜在机制。

在未来的研究中，探索将经验回放的内容偏向于显著事件的可能性将非常重要，这种现象在经验观察到的海马体回放中具有特征性，并且与强化学习中的“优先扫描”概念相关。总体而言，我们的工作展示了将最先进的机器学习技术与受生物学启发的机制相结合的强大潜力，从而创造出能够掌握多样化且具有挑战性任务的人工智能体。

## 方法

**数据预处理**

直接处理原始的Atari 2600帧（210×160像素图像，128种颜色）在计算和内存需求方面可能较为繁重。因此，我们采用了一种基本的预处理步骤，旨在降低输入维度，并处理Atari 2600模拟器中的一些伪影。  
首先，为了编码单帧图像，我们取当前帧和前一帧中每个像素颜色的最大值。这一操作是为了消除游戏中存在的闪烁现象，即某些物体仅在偶数帧中出现，而另一些物体仅在奇数帧中出现，这是由于Atari 2600同时显示精灵数量有限而导致的伪影。  
其次，我们从RGB帧中提取Y通道（即亮度通道），并将其缩放为84×84的图像。算法1中描述的函数 *w* 对最近的 *m* 帧应用这种预处理，并将它们堆叠起来以产生输入到Q函数的输入，其中 *m*=4，尽管算法对不同的 *m* 值（例如3或5）具有鲁棒性。

**代码可用性**

源代码可在 <https://sites.google.com/a/deepmind.com/dqn> 获取，仅限非商业用途。

**模型架构**

使用神经网络参数化Q函数有多种方式。由于Q函数将历史-动作对映射到标量的Q值估计，一些先前的方法将历史和动作作为神经网络的输入。这种架构的主要缺点是需要为每个动作单独进行前向传播，从而导致计算成本随动作数量线性增长。  
我们采用了一种不同的架构，即为每个可能的动作设置一个独立的输出单元，而只有状态表示作为神经网络的输入。输出对应于输入状态下各个动作的预测Q值。这种架构的主要优点是能够在单次网络前向传播中计算给定状态下所有可能动作的Q值。  
具体的架构如下：神经网络的输入是一个由预处理函数 *w* 生成的84×84×4图像。第一隐藏层使用32个8×8的滤波器对输入图像进行卷积，步长为4，并应用ReLU非线性激活函数。第二隐藏层使用64个4×4的滤波器进行卷积，步长为2，同样使用ReLU非线性激活。第三卷积层使用64个3×3的滤波器进行卷积，步长为1，后接ReLU激活。最后一层隐藏层是全连接层，包含512个ReLU单元。输出层是一个全连接的线性层，每个有效动作对应一个输出。我们考虑的游戏中的有效动作数量在4到18之间变化。

**训练细节**

我们在49款Atari 2600游戏上进行了实验，这些游戏的结果可用于所有其他可比方法。每款游戏都训练了一个不同的网络：所有游戏均使用相同的网络架构、学习算法和超参数设置（见扩展数据表1），这表明我们的方法足够鲁棒，能够在多种游戏中工作，同时仅需极小的先验知识（见下文）。  
尽管我们在未修改的游戏中评估了我们的智能体，但在训练期间，我们对游戏的奖励结构进行了一个修改。由于不同游戏的得分范围差异很大，我们将所有正奖励限制为1，所有负奖励限制为-1，而0奖励保持不变。这种奖励裁剪方式限制了误差导数的范围，并使得我们可以跨多款游戏使用相同的 学习率。同时，这也可能影响我们智能体的性能，因为它无法区分不同大小的奖励。  
对于有生命计数器的游戏，Atari 2600模拟器还会发送游戏中剩余的生命数量，这在训练期间被用来标记一个回合的结束。  
在这些实验中，我们使用了RMSProp算法（见[多伦多大学计算机科学系](http://www.cs.toronto.edu/)的讲座幻灯片），小批量大小为32。训练期间的行为策略为ε-贪婪策略，其中ε在前100万帧内从1.0线性退火到0.1，之后保持在0.1。我们总共训练了5000万帧（即大约38天的游戏经验），并使用了包含最近100万帧的回放记忆。  
遵循之前玩Atari 2600游戏的方法，我们还使用了一种简单的帧跳过技术。具体来说，智能体每 *k* 帧观察和选择一次动作，而不是每帧都进行操作，而其上一次动作会在跳过的帧中重复执行。由于运行模拟器向前一步所需的计算量远小于智能体选择动作所需的计算量，这种技术允许智能体在不显著增加运行时间的情况下多玩大约 *k* 倍的游戏。我们在所有游戏中使用 *k*=4。  
所有超参数和优化参数的值是通过对Pong、Breakout、Seaquest、Space Invaders和Beam Rider这五款游戏进行非正式搜索得出的。由于计算成本较高，我们没有进行系统的网格搜索。这些参数随后在所有其他游戏中保持不变。所有超参数的值和描述均在扩展数据表1中提供。

**实验设置中的先验知识**

我们的实验设置仅使用了以下极小的先验知识：输入数据为视觉图像（这促使我们使用卷积深度网络）、游戏特定的得分（未进行修改）、动作数量（尽管不是它们的对应关系，例如“向上”按钮的规范）以及生命计数。

**评估过程**

训练后的智能体通过在每款游戏中运行30次、每次最多5分钟、在不同的初始随机条件下（“无操作”；见扩展数据表1）以及使用ε-贪婪策略（ε=0.05）来评估。这一过程旨在最小化评估期间过拟合的可能性。  
随机智能体作为基线比较，以10Hz的频率选择随机动作，即每六帧选择一次动作，并在中间帧中重复其上一次动作。10Hz大约是人类玩家选择“发射”按钮的最快速度，将随机智能体设置为这一频率可以避免在少数游戏中出现虚假的基线分数。我们还评估了以60Hz（即每帧）选择动作的随机智能体的性能。这种改变的影响极小：在六款游戏（Boxing、Breakout、Crazy Climber、Demon Attack、Krull和Robotank）中，归一化的DQN性能变化超过5%，而在这些游戏中，DQN的性能都显著优于专业人类玩家。  
专业人类测试者使用与智能体相同的模拟器引擎，并在受控条件下进行游戏。人类测试者不允许暂停、保存或重新加载游戏。与原始Atari 2600环境一样，模拟器以60Hz运行，并且音频输出被禁用：因此，人类玩家和智能体的感官输入是相同的。人类的表现是每款游戏大约20个回合的平均奖励，每个回合最多持续5分钟，这些游戏在大约2小时的练习后进行测试。

**算法**

我们考虑的任务是智能体与环境（在这种情况下是Atari模拟器）通过一系列动作、观察和奖励进行交互。在每个时间步，智能体从合法的游戏动作集合 *A*={1,…,*K*} 中选择一个动作 *at*​。动作被传递给模拟器，并修改其内部状态和游戏得分。一般来说，环境可能是随机的。模拟器的内部状态对智能体是不可见的；相反，智能体观察模拟器输出的图像 *xt*​∈R*d*，这是一个表示当前屏幕的像素值向量。此外，智能体还会收到一个奖励 *rt*​，表示游戏得分的变化。需要注意的是，游戏得分通常取决于之前整个动作和观察序列；关于某个动作的反馈可能只有在经过数千个时间步后才会收到。

由于智能体仅观察当前屏幕，任务是部分可观测的（即许多模拟器状态在感知上是等价的，仅从当前屏幕 *xt*​ 无法完全理解当前情况）。因此，算法的输入是动作和观察的序列 *st*​=(*x*1​,*a*1​,*x*2​,…,*at*−1​,*xt*​)，算法根据这些序列学习游戏策略。假设模拟器中的所有序列都在有限的时间步内终止，这构成了一个庞大但有限的马尔可夫决策过程（MDP），其中每个序列是一个独立的状态。因此，我们可以通过简单地使用完整的序列 *st*​ 作为时间 *t* 的状态表示，应用标准的MDP强化学习方法。

智能体的目标是通过选择动作与模拟器交互，以最大化未来的奖励。我们假设未来的奖励以每时间步 *γ* 的因子进行折扣（实验中 *γ* 被设置为0.99），并定义时间 *t* 的未来折扣回报为 *Rt*​=∑*t*′=*tT*​*γt*′−*trt*′​，其中 *T* 是游戏终止的时间步。我们定义最优动作价值函数 *Q*∗(*s*,*a*) 为在观察到某个序列 *s* 后采取某个动作 *a* 所能实现的最大期望回报，即 *Q*∗(*s*,*a*)=max*π*​E[*Rt*​∣*st*​=*s*,*at*​=*a*,*π*]，其中 *π* 是一个将序列映射到动作（或动作分布）的策略。

最优动作价值函数满足一个重要的恒等式，称为贝尔曼方程。其直观含义是：如果已知下一个时间步所有可能动作 *a*′ 的序列 *s*′ 的最优值 *Q*∗(*s*′,*a*′)，那么最优策略是选择最大化 *r*+*γQ*∗(*s*′,*a*′) 期望值的动作 *a*′：

*Q*∗(*s*,*a*)=E*s*′,*a*′​[*r*+*γa*′max​*Q*∗(*s*′,*a*′)∣*s*,*a*]

许多强化学习算法的基本思想是使用贝尔曼方程作为迭代更新来估计动作价值函数，即 *Qi*+1​(*s*,*a*)=E*s*′,*a*′​[*r*+*γ*max*a*′​*Qi*​(*s*′,*a*′)∣*s*,*a*]。这种价值迭代算法会收敛到最优动作价值函数 *Qi*​→*Q*∗。然而，在实践中，这种方法是不切实际的，因为动作价值函数是针对每个序列单独估计的，没有泛化能力。相反，通常使用函数逼近器来估计动作价值函数 *Q*(*s*,*a*;*θ*)≈*Q*∗(*s*,*a*)。在强化学习领域，这通常是一个线性函数逼近器，但有时也会使用非线性函数逼近器，例如神经网络。我们将带有权重 *θ* 的神经网络函数逼近器称为Q网络。通过调整Q网络在第 *i* 次迭代中的参数 *θi*​，以减少贝尔曼方程中的均方误差，其中最优目标值 *r*+*γ*max*a*′​*Q*∗(*s*′,*a*′) 被替换为近似目标值 *y*=*r*+*γ*max*a*′​*Q*(*s*′,*a*′;*θi*−​)，使用来自前一次迭代的参数 *θi*−​。

这导致了一个随每次迭代 *i* 变化的损失函数序列 *Li*​(*θi*​)：

*Li*​(*θi*​)=E*s*,*a*,*r*,*s*′​[(*y*−*Q*(*s*,*a*;*θi*​))2]

需要注意的是，目标值依赖于网络权重。这与监督学习中使用的固定目标值形成对比，监督学习的目标值在学习开始之前就已经确定。在优化的每个阶段，我们在优化第 *i* 次损失函数 *Li*​(*θi*​) 时，固定前一次迭代的参数 *θi*−1​，从而得到一系列明确定义的优化问题。最终项是目标值的方差，它不依赖于我们当前正在优化的参数 *θi*​，因此可以忽略不计。

对损失函数关于权重求导，我们得到以下梯度：

∇*θi*​​*L*(*θi*​)=E*s*,*a*,*r*,*s*′​[(*r*+*γa*′max​*Q*(*s*′,*a*′;*θi*−1​)−*Q*(*s*,*a*;*θi*​))∇*θi*​​*Q*(*s*,*a*;*θi*​)]

通常情况下，计算上述梯度中的完整期望在计算上是低效的。因此，我们通常使用随机梯度下降来优化损失函数。在这个框架下，通过在每个时间步更新权重、用单个样本来替代期望，并设置 *θi*−1​=*θi*​，可以恢复到熟悉的Q学习算法（Q-learning）。需要注意的是，该算法是无模型的（model-free）：它直接使用模拟器的样本解决强化学习任务，而无需显式估计奖励和转移动态 *P*(*r*,*s*′∣*s*,*a*)。此外，它也是离策略的（off-policy）：它学习关于贪婪策略 *a*=argmax*a*′​*Q*(*s*,*a*′;*θ*) 的知识，而实际遵循的行为分布确保了对状态空间的充分探索。在实践中，行为分布通常由ε-贪婪策略选择，以概率 1−*ϵ* 遵循贪婪策略，以概率 *ϵ* 选择随机动作。

**深度Q网络的训练算法**

深度Q网络的完整训练算法如算法1所示。智能体根据基于Q值的ε-贪婪策略选择并执行动作。由于将任意长度的历史作为神经网络的输入可能会很困难，我们的Q函数改为使用上述函数 *w* 生成的固定长度的历史表示。该算法通过两种方式改进标准的在线Q学习，以使其适合于训练大型神经网络而不发散。

首先，我们使用一种称为“经验回放”（experience replay）的技术，将智能体在每个时间步的经验 *et*​=(*st*​,*at*​,*rt*​,*st*+1​) 存储在一个数据集 *Dt*​={*e*1​,…,*et*​} 中，并将这些数据汇总到一个回放记忆中（一个回合的结束是指到达终止状态）。在算法的内循环中，我们对从存储的样本池中随机抽取的经验样本 (*s*,*a*,*r*,*s*′)∈*U*(*D*) 应用Q学习更新（或小批量更新）。与标准在线Q学习相比，这种方法具有以下优势：

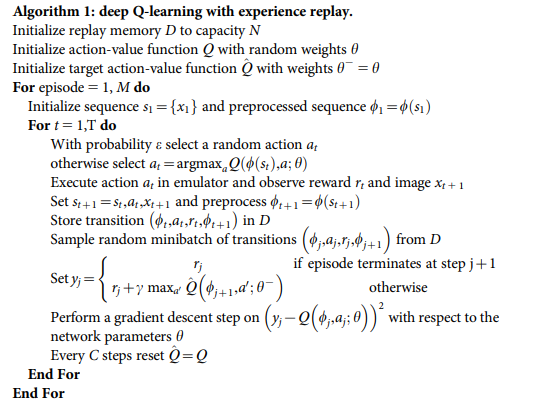
1. 每个经验样本可能会被用于多次权重更新，从而提高了数据的利用效率。
2. 直接从连续样本中学习是低效的，因为这些样本之间存在强相关性。随机化样本可以打破这些相关性，从而减少更新的方差。
3. 当在线策略学习时，当前参数决定了参数将要训练的下一个数据样本。例如，如果最大化动作是向左移动，那么训练样本将主要来自左侧；如果最大化动作切换到向右，那么训练分布也会随之切换。很容易看出，这种情况下可能会产生不良的反馈循环，参数可能会陷入一个较差的局部最小值，甚至可能灾难性地发散。通过使用经验回放，行为分布会在其许多先前状态之间进行平均，从而平滑学习过程，避免参数的振荡或发散。

需要注意的是，当通过经验回放进行学习时，必须进行离策略学习（因为当前参数与生成样本时使用的参数不同），这促使我们选择Q学习。

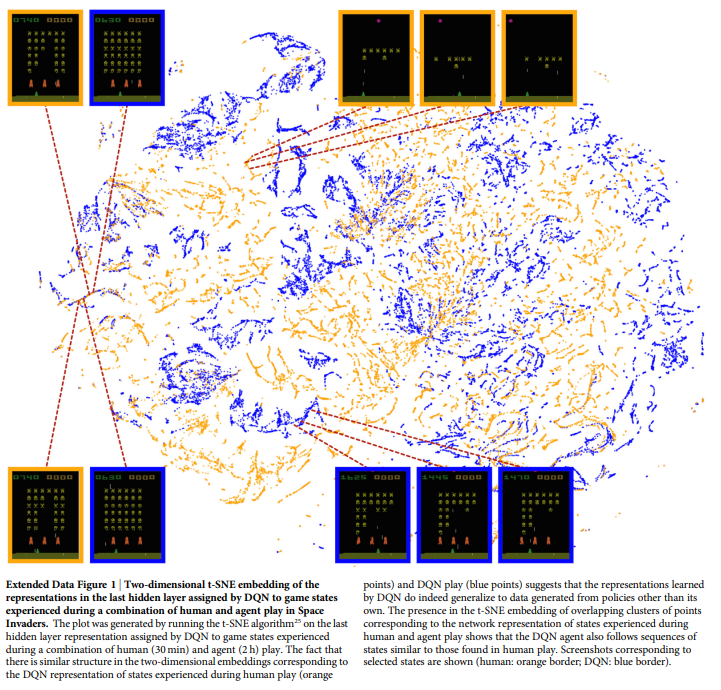
在实践中，我们的算法仅在回放记忆中存储最近的 *N* 个经验元组，并在执行更新时从 *D* 中均匀随机采样。这种方法在某些方面存在局限性，因为记忆缓冲区不会区分重要的转换，并且由于有限的记忆大小 *N*，总是用最近的转换覆盖旧的转换。同样，均匀采样会给予回放记忆中的所有转换相同的权重。一种更复杂的采样策略可能会强调那些我们能从中学习最多的转换，类似于强化学习中的优先扫描（prioritized sweeping）。

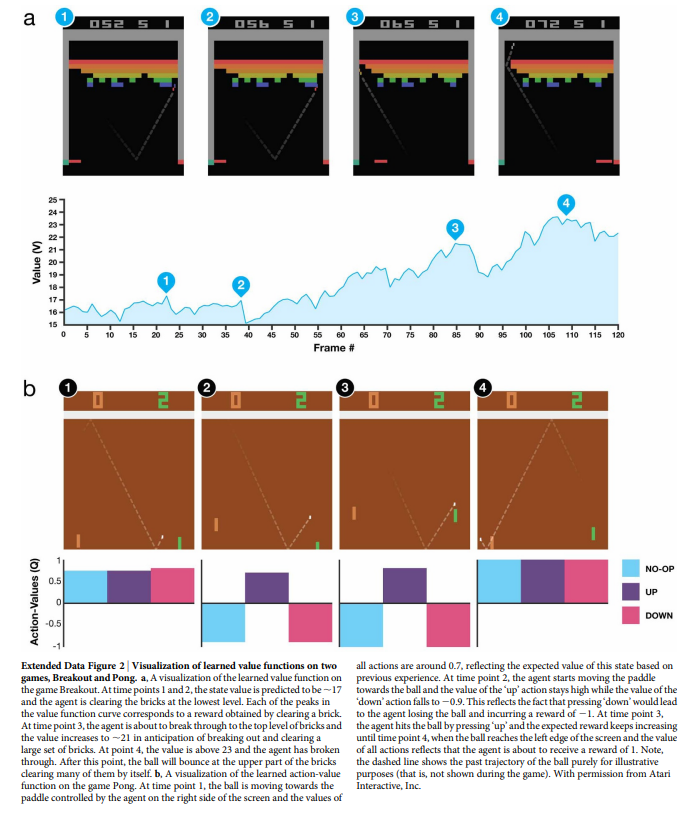
为了进一步提高使用神经网络时算法的稳定性，我们对在线Q学习进行了第二个修改，即使用一个独立的网络来生成Q学习更新中的目标值 *yj*​。具体来说，每经过 *C* 次更新，我们克隆网络 *Q* 以获得目标网络 *Q*′，并使用 *Q*′ 生成接下来 *C* 次更新到 *Q* 的Q学习目标 *yj*​。这种修改使得算法比标准的在线Q学习更稳定，因为在标准在线Q学习中，增加 *Q*(*st*​,*at*​) 的更新通常也会增加 *Q*(*st*+1​,*a*) 对于所有的 *a*，从而也增加了目标 *yj*​，这可能导致策略的振荡或发散。使用一组较旧的参数生成目标值，增加了更新 *Q* 和更新影响目标 *yj*​ 之间的时间延迟，使得发散或振荡的可能性大大降低。

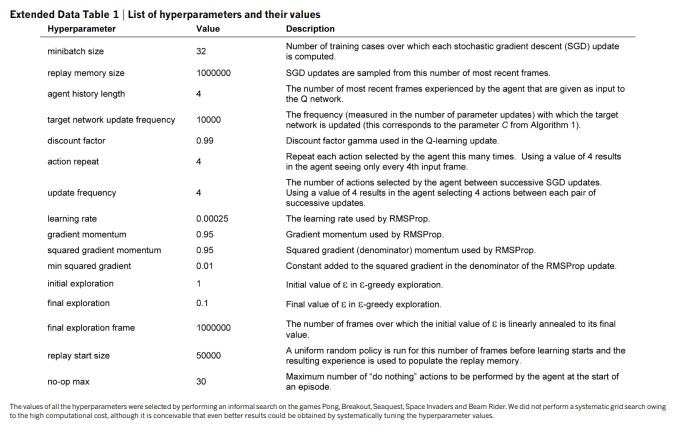
我们还发现将更新中的误差项 *r*+*γ*max*a*′​*Q*(*s*′,*a*′;*θi*−1​)−*Q*(*s*,*a*;*θi*​) 限制在 −1 和 1 之间是有帮助的。因为绝对值损失函数 ∣*x*∣ 对于所有负值的 *x* 的导数为 −1，对于所有正值的 *x* 的导数为 1，将平方误差限制在 −1 和 1 之间对应于对于 (−1,1) 区间外的误差使用绝对值损失函数。这种误差裁剪形式进一步提高了算法的稳定性。

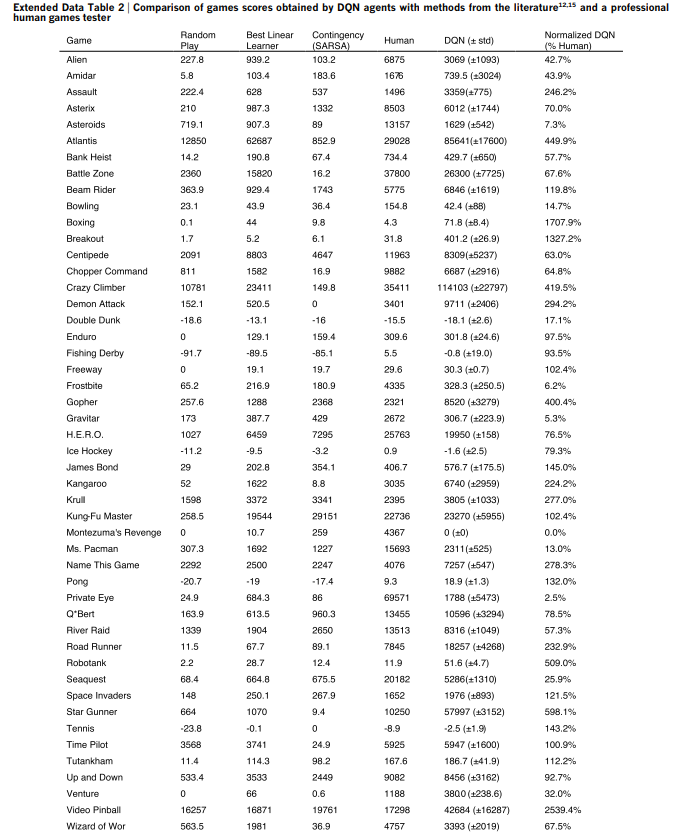


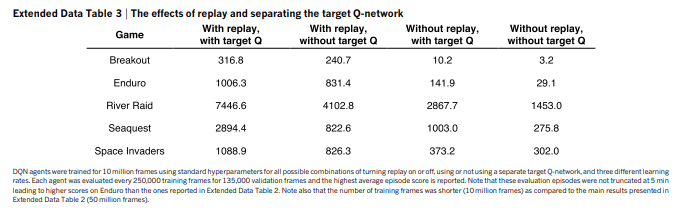
## 扩展数据

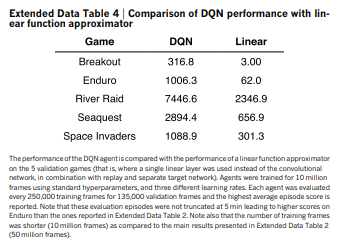












## 参考文献

1. Sutton, R. & Barto, A. Reinforcement Learning: An Introduction (MIT Press, 1998).

2. Thorndike, E. L. Animal Intelligence: Experimental studies (Macmillan, 1911).

3. Schultz, W., Dayan, P. & Montague, P. R. A neural substrate of prediction and reward. Science 275, 1593–1599 (1997).

4. Serre, T., Wolf, L. & Poggio, T. Object recognition with features inspired by visual cortex. Proc. IEEE. Comput. Soc. Conf. Comput. Vis. Pattern. Recognit. 994–1000 (2005).

5. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 36, 193–202 (1980).

6. Tesauro, G. Temporal difference learning and TD-Gammon. Commun. ACM 38, 58–68 (1995).

7. Riedmiller, M., Gabel, T., Hafner, R. & Lange, S. Reinforcement learning for robot soccer. Auton. Robots 27, 55–73 (2009).

8. Diuk, C., Cohen, A. & Littman, M. L. An object-oriented representation for efficient reinforcement learning. Proc. Int. Conf. Mach. Learn. 240–247 (2008).

9. Bengio, Y. Learning deep architectures for AI. Foundations and Trends in Machine Learning 2, 1–127 (2009).

10. Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst.25, 1106–1114 (2012).

11. Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. Science 313, 504–507 (2006).

12. Bellemare, M. G., Naddaf, Y., Veness, J. & Bowling, M. The arcade learning environment: An evaluation platform for general agents. J. Artif. Intell. Res. 47, 253–279 (2013).

13. Legg, S. & Hutter, M. Universal Intelligence: a definition of machine intelligence. Minds Mach. 17, 391–444 (2007).

14. Genesereth, M., Love, N. & Pell, B. General game playing: overview of the AAAI competition. AI Mag. 26, 62–72 (2005).

15. Bellemare, M. G., Veness, J. & Bowling, M. Investigating contingency awareness using Atari 2600 games. Proc. Conf. AAAI. Artif. Intell. 864–871 (2012).

16. McClelland, J. L., Rumelhart, D. E. & Group, T. P. R. Parallel Distributed Processing: Explorations in the Microstructure of Cognition (MIT Press, 1986).

17. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324 (1998).

18. Hubel, D. H. & Wiesel, T. N. Shape and arrangement of columns in cat’s striate cortex. J. Physiol. 165, 559–568 (1963).

19. Watkins, C. J. & Dayan, P. Q-learning. Mach. Learn. 8, 279–292 (1992).

20. Tsitsiklis, J. & Roy, B. V. An analysis of temporal-difference learning with function approximation. IEEE Trans. Automat. Contr. 42, 674–690 (1997).

21. McClelland, J. L., McNaughton, B. L. & O’Reilly, R. C. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. Psychol. Rev. 102, 419–457 (1995).

22. O’Neill, J., Pleydell-Bouverie, B., Dupret, D. & Csicsvari, J. Play it again: reactivation of waking experience and memory. Trends Neurosci. 33, 220–229 (2010).

23. Lin, L.-J. Reinforcement learning for robots using neural networks. Technical Report, DTIC Document (1993).

24. Riedmiller, M. Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. Mach. Learn.: ECML, 3720, 317–328 (Springer, 2005).

25. Van der Maaten, L. J. P. & Hinton, G. E. Visualizing high-dimensional data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605 (2008).

26. Lange, S. & Riedmiller, M. Deep auto-encoder neural networks in reinforcement learning. Proc. Int. Jt. Conf. Neural. Netw. 1–8 (2010).

27. Law, C.-T. & Gold, J. I. Reinforcement learning can account for associative and perceptual learning on a visual decision task. Nature Neurosci. 12, 655 (2009).

28. Sigala, N. & Logothetis, N. K. Visual categorization shapes feature selectivity in the primate temporal cortex. Nature 415, 318–320 (2002).

29. Bendor, D. & Wilson, M. A. Biasing the content of hippocampal replay during sleep. Nature Neurosci. 15, 1439–1444 (2012).

30. Moore, A. & Atkeson, C. Prioritized sweeping: reinforcement learning with less data and less real time. Mach. Learn. 13, 103–130 (1993).

31. Jarrett, K., Kavukcuoglu, K., Ranzato, M. A. & LeCun, Y.What is the bestmulti-stage architecture for object recognition? Proc. IEEE. Int. Conf. Comput. Vis. 2146–2153 (2009).

32. Nair, V. & Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. Proc. Int. Conf. Mach. Learn. 807–814 (2010).

33. Kaelbling, L. P., Littman, M. L. & Cassandra, A. R. Planning and acting in partially observable stochastic domains. Artificial Intelligence 101, 99–134 (1994)