

*从2015开始，针对深度强化学习的研究就没有停止过，但由于其研究意义原大于商业意义，且难以商业化。不过从如今的形式看来，强化学习跟像深度学习之类新技术的结合还远不止如此，强化学习的框架在未来会层出不穷，每个IT从业者都需要掌握这种为强化学习建立环境框架、编写训练代码、验证结果的能力。就像现在的Django、Flutter\Gin*

Abstract

近几年基于深度特征表达的强化学习方法获得了很多喜人的成绩，很多类似模型都采用了常规神经网络架构，例如卷积神经网络（CNN）、长短期记忆（LSTM）、自编码器（Auto Encoders）等。在这篇文章中，我们介绍了一种新型神经网络结构，这个结构能够用于无模型的强化学习，并且会对未来的RL算法起到深远的影响，我们称为决斗网络（Dueling Network）。注意这个新网络模型并没有创造新的技术类型，仅仅是一种已知技术的模型结构变化。

1. Introduction

经过最近几年的发展，深度学习在可扩展性和性能上都比机器学习有了质的飞跃。尤其让人兴奋的是关于强化学习与控制策略的序列性决策应用，也就是深度强化学习的出现，以及很多基于深度学习的优秀案例，这里就不一一列举了。

决斗网络明确提出了网络之间需要决斗，于是它将每个状态的Q值估计与动作优劣（函数）计算放在两个不同的计算流中，图1就显示了传统的单流计算方式（图1上）与我们的对抗流方式（图1下）。我们可以看到两个计算流在共享卷积网络的特征提取部分，然后通过特殊的聚合层来进行决斗，紧接着模型必须能够生成状态值Q（那个小方块）与动作优势评价。所以决斗网络能够在没有任何监督的前提下，同时估Q 值与动作优势函数。

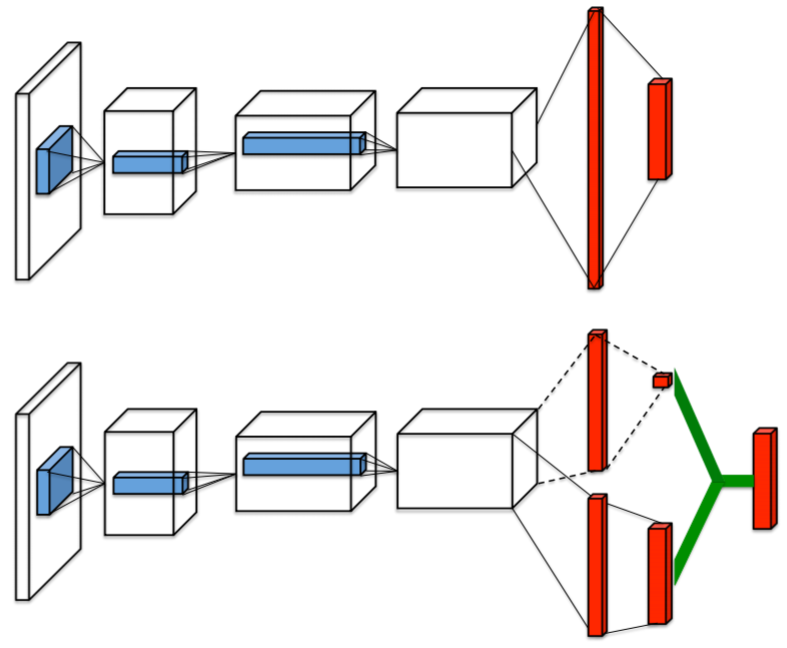


图1

单凭直觉来看，决斗网络似乎能够在某个状态做完动作并学习到一些经验之前，就可以估计该状态的价值，这对于动作与状态相关性低的环境非常有用。两个计算流之间的映射由一个雅可比矩阵完成，正如Simonyan 等人在2013年提出的方法。图2展示了这个映射过程，这是在Atari游戏Enduro上的结果，带点红色的区域就是该计算流的关注点，图2左是状态值计算流，图2右是优势动作计算流。我们可以看到图2左，状态值计算流对远方的路、附近即将接近的车辆与下方分数牌都有关注；图2右则没有对路线与分数牌给予过多的关注，因为决定动作的往往只有“需要避让前方车辆”这样的需求，所以它只对前方车辆感兴趣。

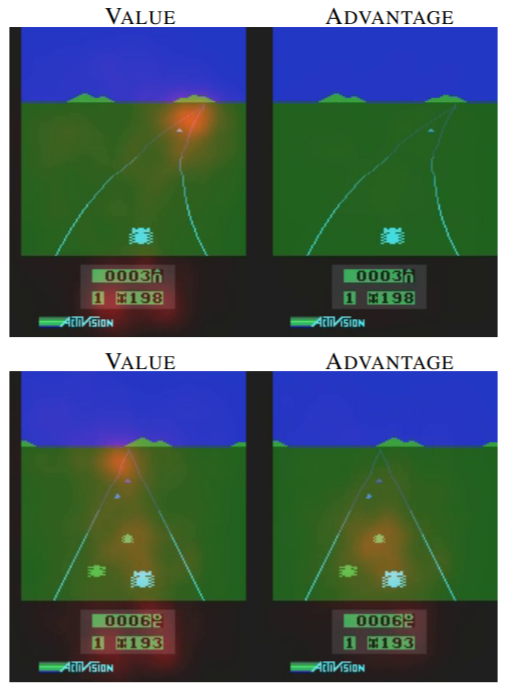


图2

**1.1. Related Work**

这种维持两种计算流的方法可以追溯到1993年Baird的研究——原始优势更新算法（original advantage updating algorithm），其中就有一个Bellman残差更新方程，一个方程分解出两个更新过程：一个用于状态值函数，一个用于其相关的优势函数。1995年Harmon等人的研究证明了优势更新方法在连续状态的环境下，比Q-learning学习更容易收敛。作为优势更新的后代，优势学习算法仅使用单一的优势函数。

决斗网络结构包含了两个函数，V(s)与A(s,a)，得到这两个函数输出后，使用深度神经网络再将其混合为一个Q(s,a)。这种优势学习算法是由网络结构解耦的，所以决斗网络的结构能够用于所有无模型的RL算法。

策略梯度算法与优势学习算法有很长的历史关系，可以从Sutton等人的研究（2000年）作为起始点。作为这一系列研究的最新例子，Schulman等人于2015年研究集中于在线估计优势函数，这样可以减少策略梯度算法的方差。

2. Background

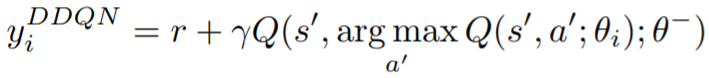
本文所描述的环境依然符合马尔可夫决策过程（MDP），AI就在这样的环境 中。在atari环境中，AI每个时间步t处理M个图像 ，然后AI在动作集 中选择一个动作，并获取环境给予的相应奖励。AI追求最大期望奖励，配以贪心算法，相关计算方式与上一篇论文一致。

**2.1. Deep Q-networks**

详见项目文档《1.DQN.docx》

**2.2. Double Deep Q-networks**

在DQN的基础上，本文吸纳另一篇文章的精华——DDQN（Double DQN）。DQN中选择最优动作与评价动作的最大化操作使用了一样的值，这会使值估计过于乐观。为了修正这个问题，DDQN的目标函数使用下列形式（DDQN与DQN的区别仅仅在这个公式上）：



**2.3. Prioritized Replay**

另一篇文章就是项目文章《3. Prioritized Replay》，它构建于DDQN之上但比DDQN更加优秀。它们的关键技术就是增加了记忆的优先级，使得最有策略的寻找更加快速，具体详见我对那篇文章的翻译。

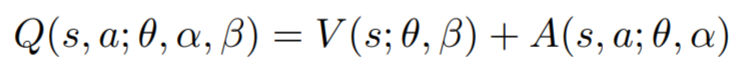
3. The Dueling Network Architecture

其实图2已经能够描述决斗网络的结构了，如果再深度描述一下的话，就是对于许多状态，并不需要去估计每个动作选择的优势。例如在游戏Enduro中，判断是否需要做出向左或向右的动作，只需要在明显看到有障碍的时候，如果前路平坦无车就不需要花时间来做计算了。

为了做对比，正如图1所描述的，我们也搭建了一个单独的DQN模型。对决网络的模型卷积部分与DQN相同，卷积之后单DQN是衔接一个全连接层，我们则是分流为两个全连接层，分流单独计算之后再合并为一个Q函数，以输出针对每个动作选择的Q值。

由于决斗网络的输出是一个Q函数，所以通过很多算法可以对其进行训练，比如DDQN与SARSA，顺带说一句，这种训练方式有很多途径进行改进，包括记忆优化、贪心策略优化、内在动机优化等。

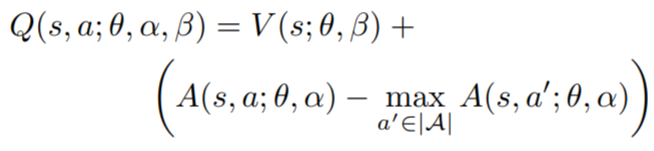
对决斗网络中两个计算流进行合并的技术必须经过精心的设计。就像图1描述的对决网络结构，其中一个计算流的输出为 ，而另一个计算流的输出是一个动作域维度的向量 。这些公式里面， 是卷积网络的参数，而 是各自全连接网络的参数，所以对决模型的聚合部分可以写为以下的形式：



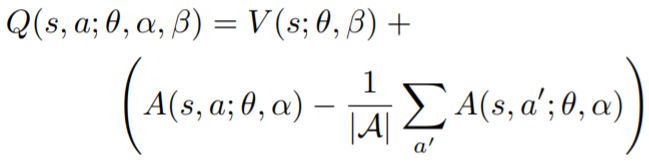
注意上式是针对所有（s,a）实例对的，因为A是个向量而V是个值，矩阵运算自然会把V广播到A的维度。同时要时刻明白Q(s, a; θ, α, β) 仅仅是对真值Q的参数化估计，单纯的认为V已经是一个很好的状态值估计了或者A已经提供了有效的优势估计，很可能这样的想法是不正确的。

上式同样也是一个不可逆的过程，即已知Q，并不能得到固定的V与A。给V加一个固定的常数，同样也给A减去这个常数，得到的Q是不变的。当直接使用公式时，这种不可识别性体现在性能的差异上。

为了解决这一问题，我们通过让A向量减去最优势动作值，将A向量削弱在0及以下，这样当动作选择最优时，Q就等于V，而其他情况Q则比V更小。这样相当于从视觉过来的V在进行正面影响，而动作优势计算流A则在进行反面影响。



另一类方法就是使用平均惩罚来代替最大值惩罚，这种做法也能够解决上述问题，因为A仍然被稳定下来了，如果结果发生明显变化时大多数情况会由V引起。我们也实验过最大值惩罚配合Softmax的情况，不过结果看起来跟下面的公式差不多，所以本文接下来讲解的所有实验均采用下方公式：



在此公式基础上，仍可以实施像贪心算法这样的技术，当AI需要做出动作时，按A计算流的估计结果执行即可。有一点非常重要，就时上式的使用过程是整个决斗网络的一部分，并不是单独计算的。整个架构就包含了传统Q网络、V和A的计算流，新加入的这两个计算流的优化过程也包含在反向传播中，并不需要单独的监督算法或优化算法。由于决斗网络与传统Q网络的输入输出形式一致，所以任何适用于Q网络的训练方法也都适用于决斗网络。

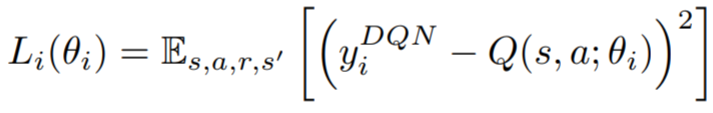
4. Experiments

接下来我们就一起看看决斗网络的实际性能。先从一个简单的策略评估开始，然后逐步扩大到能泛化于Atari游戏的学略学习结果。

**4.1. Policy evaluation**

一个简单的策略评估能够直观验证网络结构的有效性，它避免了贪心算法中探索概率衰减的策略影响，也减弱了策略更新与策略评估之间的相互影响，当消除了这些混淆因素后得到的评估结果就直接与网络结构相关。在这个实验中，我们使用了时间差分学习方法来学习Q值。更确切的说，当给定行为策略π时直接计算得到估计值 ，然后与下列目标进行误差计算。





为了验证学习到的Q值，我们选择了一个相对简单的环境——一个由3个走廊相连的走廊，别问我为什么这句话这么难懂，老外的逻辑就时这么不靠谱，具体请见下图。AI从左下方开始行动，目标是抵达右上方，这样就可以获得奖励。动作集包含5个动作：上、下、左、右、无动作。我们能够自由添加任意数量的无操作动作（搞得好像会有人限制你们似的），给予环境的设定是两个垂直走廊各有10个状态，一个水平走廊有50个状态。

我们在行为策略计算（离线算法一共两个策略，行为策略与目标策略）中使用了贪心算法，探索概率值锁定为0.001。

为了增加对比项目，我们通过添加“无动作”的动作数量，来使环境的动作总数变为5、10与20三种。评价指标选择针对人判断的真值，并计算方差。传统DQN模型最后三个全连接层是每层具有50个神经元的MLP，对决网络模型第一层也是50个神经元，然后第二层分道扬镳为两个25神经元的隐含层，第三层再做汇聚。详细环境与比试结果参见图3。

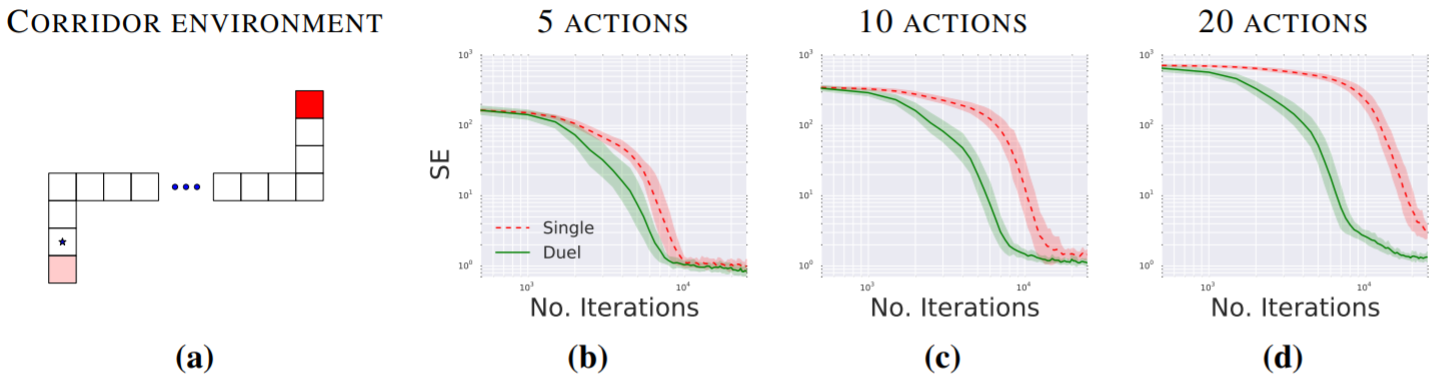


图3

我们可以看到，在5个动作（图3b）的环境里，两个模型以差不多相同的速度收敛。而当环境中的可用动作增加时，决斗网络就的性能就会逐渐优于传统DQN。上图中第一个就时走廊环境，星号所在的位置就时起始点。

**4.2. General Atari Game-Playing**

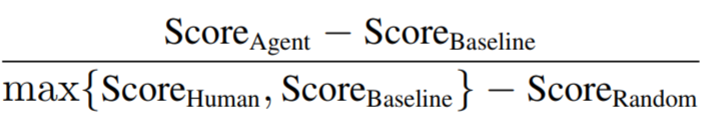
我们采用的Atari游戏环境与DQN是一样的，进行的挑战也是使用统一的模型结构、算法与超参数，来适配尽量多类型的游戏，游戏环境仅提供外在观察的视频流内容。

对于模型的卷积网络部分，与DQN是相同的。3层卷积，其中第一层拥有32个8x8的卷积核，步长为4；第二层拥有64个4x4的卷积核，步长为2；第三层拥有64个3x3的卷积核，步长为1。而对于全连接层，决斗网络在第三个卷积后将数据复制为两份，在各自的计算流中都使用了512个神经元的隐含层，紧接着输出各自长度的矢（或标量），最后使用上文中描述的求和公式进行合并，所有层之间均使用整流非线性函数作为激活函数（Rectifier non-linearities）。

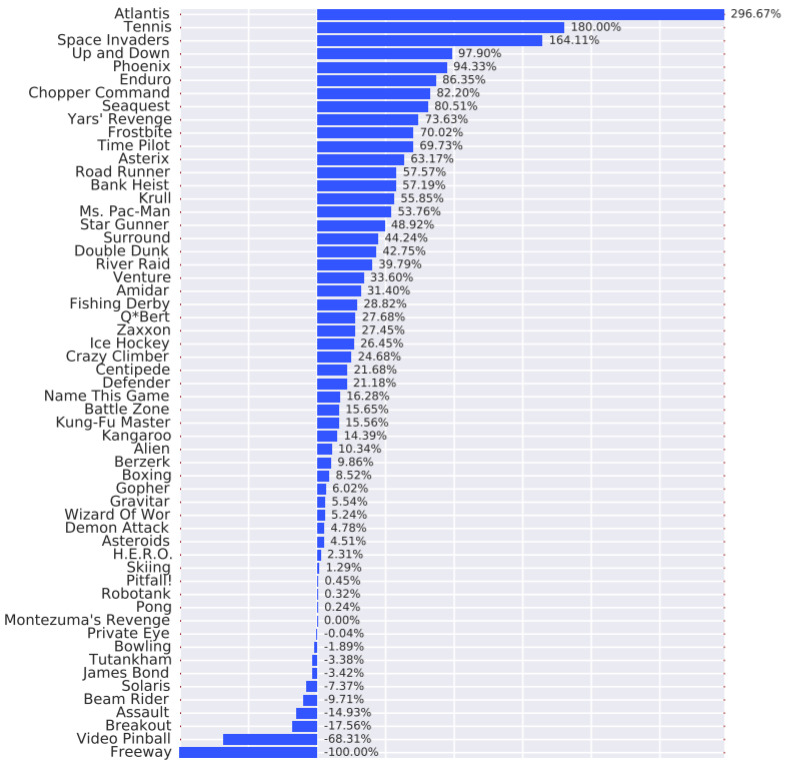
决斗网络的反向传播过程有个问题，因为反向传播的时候，第三个卷积层会接收到后方传来的两个数据流的残差项，如果直接相加则会导致更新内容太多，导致前后更新的差异进而造成网络不稳定。为了解决这个问题我们把第三个卷积层接收到的两个反向传播项都缩放为 ，这样网络就会相对稳定很多。另外我们还修剪梯度，使其范数保持在10以内，这个技巧在强化学习中并不常见，但却经常出现在循环网络中。

前文说过我们会参照DDQN与DQN这两个模型，以这两个模型为基础我们构建了两个版本的对比模型，一个是通过对DDQN进行再训练（re-train），并为全连接层配置1024个神经元的版本（完全参照DDQN配置），我们称为Single Clip；而另一个版本则以原始DQN为基石完整训练构建，我们称为Single。

根据van Hasselt等人在2015年的论文，我们针对每个游戏都先填入30帧的“无动作”，这样可以让游戏先“发展”一会。然后找到一个随机的点来开始游戏，这样可以让AI每次在游戏开头都觉得自己好像玩了一个全新的版本。我们还设计了一个对游戏得分的评估机制：



其中Human是人类玩家的得分，Baseline就是原始DDQN的模型得分，Agent是不加随机的得分，Random是加随机效果后的得分。图4就是Single版的决斗网络相比DDQN所带来的得分加成，注意大部分都是正面加成，有一些游戏获得了性能上的弱化。



***Robustness to human starts***

30个“无动作”的动作安排可能会让AI觉得猝不及防，而让AI每次都从头开始则容易陷入单一习惯性的序列，这两种方式对模型的鲁棒性都没啥帮助。所以在Nair等人在2015的论文中，就使用了让人类专家提前录制100帧的游戏开头，让AI可以随机从这100帧来选择游戏的起点。这种方法称为Human Starts。

表1显示了本次实验的各种模型的成绩，对57个游戏玩耍后，按上面的得分公式计算的平均分数与中位数得分，且区分了AI在30个“无动作”开头与人类专家随机点开头的得分。

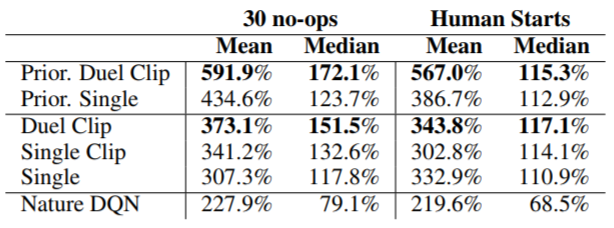


表1

***Combining with Prioritized Experience Replay***

表1 中最上方的两个模型加了前缀Prior. 是因为这两个模型加入了记忆优化技术，我们可以看到加了记忆优化的决斗网络（Prior. Duel Clip）是及其优秀的。针对这个及其优秀的模型我们又做了相同的实验，这次数据更加乐观，如图5所示。

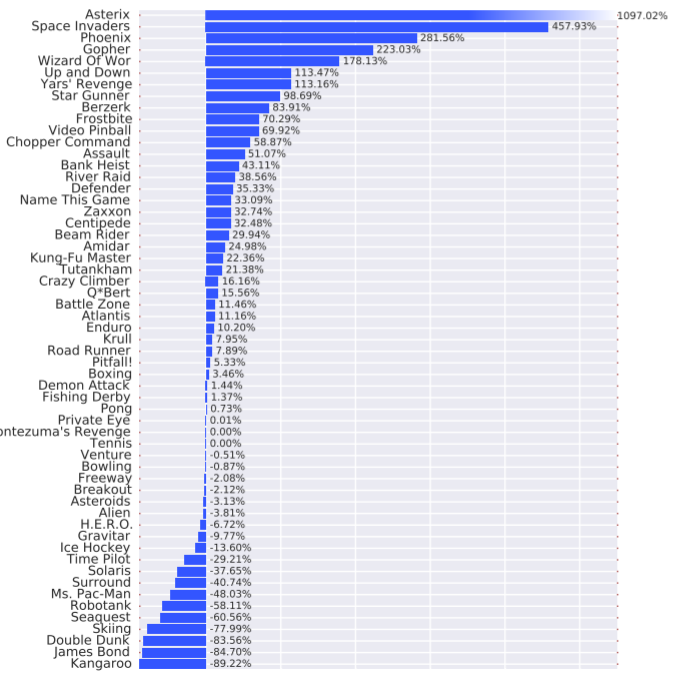


图5

***Saliency maps***

Saliency maps就是显著性映射，就是我们在图2中看到的红色部分的上色过程。这是Simonyan等人在2013年论文中所描述的方法。具体方法就是为V值计算流的输出与动作计算流的输出都再计算一个雅可比矩阵的绝对值。通过改变通道，将灰度输入放置在RGB绿色通道中以显示绿色路面，将雅可比矩阵计算的结果放置在红色通道，达到同时渲染画面的效果。

5. Discussion

总的来说，这一次模型的改进还是具有非凡的意义，感谢作者团队为我们提供了这么好的模型与思路，激励后来的研究者继续开拓。希望你看的开心，并进行实践。