

一种带探索噪音的深度循环 Q 网络

刘 全^{1),2),3),4)} 闫 岩¹⁾ 朱 斐^{1),4)} 吴 文¹⁾ 张琳琳¹⁾

¹⁾(苏州大学计算机科学与技术学院 江苏 苏州 215006)

²⁾(软件新技术与产业化协同创新中心 南京 210000)

³⁾(吉林大学符号计算与知识工程教育部重点实验室 长春 130012)

⁴⁾(苏州大学江苏省计算机信息处理技术重点实验室 江苏 苏州 215006)

摘 要 结合深度神经网络和强化学习方法的深度 Q 网络在 Atari 2600 游戏平台上取得了巨大成功,相较于深度 Q 网络,深度循环 Q 网络具有记忆历史信息的能力,在部分游戏上显示出了更好的性能.然而在某些复杂的游戏环境中,一方面深度循环 Q 网络需要大量的训练时间,另一方面其在动作空间抖动的策略中不能做出合理决策.针对这些问题,本文提出一种带探索噪音的深度循环 Q 网络(Deep Recurrent Q-Networks with Exploratory Noise, EN-DRQN)模型.与在动作空间的探索方式不同,EN-DRQN 在网络空间注入噪音,引起网络输出变化,然后根据该变化选择动作.这种在网络空间的探索可以在未来多个时间步内造成复杂的改变,并通过循环神经网络记忆多步变化,使智能体(Agent)做出的决策更具有战略性. EN-DRQN 具有以下特点:一是利用带探索性的噪音进行深度探索以弥补传统策略探索的低效性.噪音来自于噪音分布,通过方差驱动探索,这使得 Agent 可以发现大量新状态,提供更加丰富的样本,为决策提供有效信息;二是使用改进的双层门限循环单元来记忆较长时间步的历史信息,使 Agent 能够在延迟奖赏的情况下做出合理的决策.实验结果表明,EN-DRQN 模型在 Atari 2600 游戏平台上的部分战略性游戏以及具有延迟奖赏的游戏上,与动作空间的抖动策略相比,取得了更优的表现.

关键词 深度学习;强化学习;循环神经网络;卷积神经网络;噪音探索

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2019.01588

A Deep Recurrent Q Network with Exploratory Noise

LIU Quan^{1),2),3),4)} YAN Yan¹⁾ ZHU Fei^{1),4)} WU Wen¹⁾ ZHANG Lin-Lin¹⁾

¹⁾(School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006)

²⁾(Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210000)

³⁾(Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012)

⁴⁾(Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, Jiangsu 215006)

Abstract The deep Q network, which combines the deep neural network and the reinforcement learning method, has achieved great success in the Atari 2600 game platform. Compared with the deep Q network, the deep recurrent Q network has the ability of memorizing historical information and shows better performance in some strategic tasks and games with delayed rewards. However, on the one hand, traditional dithering policy cannot make a reasonable decision; on the other hand, in some complex game environments, because of the complex structure and activation

收稿日期:2018-05-22;在线出版日期:2019-03-11. 本课题得到国家自然科学基金(61772355,61472262)、江苏省高等学校自然科学研究重大项目(18KJA520011,17KJA520004)、吉林大学符号计算与知识工程教育部重点实验室资助项目(93K172014K04,93K172017K18)、苏州市应用基础研究计划工业部分(SYG201422,SYG201804)、江苏省级重点实验室(苏州大学)项目(KJS1524)资助. 刘 全,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为智能信息助理、自动推理和机器学习. E-mail: quanliu@suda.edu.cn. 闫 岩,硕士研究生,主要研究方向为强化学习、深度学习、深度强化学习. 朱 斐,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为强化学习、文本挖掘、模式识别. 吴 文,硕士研究生,主要研究方向为强化学习、深度学习、深度强化学习. 张琳琳,硕士研究生,主要研究方向为强化学习、深度强化学习.

function, the deep recurrent Q network requires a lot of training time, which increases the difficulty of the task. To solve these problems, the model of the deep recurrent Q network with exploratory noise (EN-DRQN) is proposed. Different from the way of exploration in the action space, EN-DRQN injects noise directly into the network space, which changes the output of the network, and the agent adjusts its policy through this change. The exploration in the network space can cause complex changes in multiple time steps, and the agent can memory multistep changes through the recurrent neural network memory, making the decision more strategic. Taking the above measures, the decision made by the agent is more strategic. EN-DRQN has the following characteristics: first of all, exploratory noise is used to make up for the inefficiency of traditional policy exploration. The noise comes from the noise distribution and the agent has ability to explore through the variance. If the selected action is the optimal action, the agent will reduce the scale of the noise to reduce the variance of the network. If the selected action is not the optimal action, increasing the variance of the network to improve the possibility of selecting the optimal action. The variables that control the noise scale are trained by the gradient descent method along with other weight parameters in the network. The noise policy in the network space improves the possibility of finding new states. Mining new state can provide more abundant samples for the agent at the learning stage and provide effective information for decision-making. Second of all, we use the improved double gated recurrent units to memorize the historical information of a long time step. This operation enables the agent to make reasonable decisions in the case of delayed rewards. In the deep recurrent Q network, the ability of a single layer LSTM network to memory history information is relatively limited, and it cannot achieve satisfactory performance in a few strategic environments. At the same time, the LSTM unit increases the demand for computing resources. Last but not least, we tested the effectiveness of EN-DRQN in eight games such as AirRaid, BeamRider, Centipede, Freeway and so on. It is worth mentioning that the eight games are all strategic and the rewards given by the environment are delayed. We compared the model by evaluating the average score of each game. Compared with dithering policy in the action space, the experimental results show that EN-DRQN has achieved better performance in some strategy games and games with delayed rewards.

Keywords deep learning; reinforcement learning; recurrent neural network; convolution neural network; exploratory noise

1 引 言

DeepMind 团队于 2013 年公布了基于 Atari 2600 游戏平台的深度强化学习算法^[1]. 这是一项令人瞩目的研究成果,与传统人工智能算法不同,它将具有感知能力的深度学习(Deep Learning,DL)和具有决策能力的强化学习(Reinforcement Learning,RL)相结合并在 Atari 2600 游戏平台上做了大量实验,结果表明:在大部分游戏上,该算法可以取得超越人类玩家水平的成绩.其基本框架如图 1 所示.

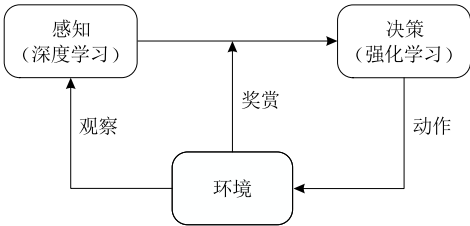


图 1 深度学习与强化学习的关系

深度强化学习在模拟真实场景任务上的优异性能表现,使其成为人工智能领域一个新的研究热点. Mnih 等人^[2]结合卷积神经网络(Convolutional Neural Networks,CNNs)和 Q 学习算法,提出一种

深度 Q 网络模型 (Deep Q-Network, DQN). DQN 在众多基于视觉感知的控制任务中均取得了优异的性能表现,具有强大的通用性. Hasselt 等人^[3]针对 Q 学习中存在的高估动作值问题,提出了一种深度双 Q 网络 (Deep Double Q-Network, DDQN), DDQN 能够很好地抑制高估所带来的误差. Schaul 等人^[4]在训练 DQN 时,采用一种优先级经验重放技术,该技术能够更加高效地利用样本,使 Agent 在部分 Atari 游戏上取得了更高的得分. Fortunato 等人^[5]提出一种利用白噪音探索的机制,弥补了传统抖动策略的不足,并在深度 Q 网络中验证了其猜想. Hausknecht 等人^[6]提出一种基于长短期记忆单元 (Long-Short Term Memory, LSTM) 的深度 Q 网络 (Deep Q-Network with Long-Short Term Memory, LSTM-DQN), 并在 Atari 2600 游戏平台上验证了其可行性与通用性. 实验结果表明, LSTM-DQN 具有一定记忆历史信息的能力,在部分战略性任务上的表现可以超过 DQN. Lample 等人^[7]从任务的特征和动作的特性角度出发,提出一种分层的深度循环 Q 网络 (Hierarchical Deep Recurrent Q-Network, H-DRQN), 在该网络中分别采用 LSTM 网络 and 全连接网络对接收到的特征进行分层处理,并在 FPS 游戏上验证. 翟建伟等人^[8]在深度循环 Q 网络中引入注意力机制 (Deep Recurrent Q-Network with Visual Attention Mechanism, VAM-DRQN) 以聚焦图像中关键信息,并在 Atari 2600 上进行实验,相比于深度循环 Q 网络,取得了更优的表现. 然而在深度循环 Q 网络中,单层的循环网络记忆历史信息的能力相对有限,且 LSTM 单元运算速度较慢,在使用 ϵ -greedy 抖动策略进行探索时,并不能有效地找到合理的行动,这些问题严重地影响 Agent 的表现.

本文的主要贡献如下:

(1) 提出一种带探索噪音的深度循环 Q 网络模型——EN-DRQN. 该模型既保留状态序列之间的相关性,又可以在状态序列之间维持高效的探索. 噪音来源于噪音分布并且是自适应的,与传统的 ϵ -greedy 抖动策略相比,无需额外探索因子便可收敛至最优. EN-DRQN 在噪音帮助下可以有效地探索出新状态,同时还会在历史信息的指引下做出高效决策;

(2) 提出一种改进的简单循环单元以降低计算的复杂度. 由于单层 LSTM 记忆历史信息能力相对有限,在一些连续性不强的环境中并不能取得满意的表现. 同时使用 LSTM 单元增加了网络对计算资

源的要求. 针对此问题,本文引入改进的简单循环单元 (Simple Recurrent Unit, SRU). 这种结构可以记忆多个时间步中的有效信息且进一步加快网络的运算.

本文通过在战略性任务和稀疏性奖赏任务上进行评估以验证提出的猜想. 实验结果表明,本文提出的 EN-DRQN 模型能高效探索,同时减少了训练时间.

2 相关工作

2.1 强化学习

强化学习,又名激励学习,是学习从状态到行为的映射,以获取数值最大的奖励信号^[9]. 强化学习中 Agent 与环境进行交互,其每一个时间步 t , 在状态 s_t 下根据策略 π 从动作集合 A 中执行某一动作 a_t , 接收到环境所反馈的下一状态 s_{t+1} 和标量奖赏 r_{t+1} , 该过程直到环境给出终止状态结束,其中策略 π 表示一个从状态 s_t 到动作 a_t 的映射.

对于所有状态,如果策略 π 的期望回报大于或等于所有其它策略的期望回报值,那么该策略 π 就被称为最优策略,用 π^* 表示.

在强化学习中,从 t 时刻到 T 时刻的累计奖赏定义为:

$$R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'} \quad (1)$$

其中, R_t 称之为回报或者累计奖赏, $\gamma \in (0, 1]$ 称之为折扣因子. Agent 的目标是最大化每个状态 s_t 下的期望累计奖赏.

状态动作值函数 $Q^\pi(s, a)$ 是指 Agent 在当前状态 s_t 下执行动作 a_t , 并一直遵循策略 π 所获得的累计奖赏:

$$Q^\pi(s, a) = E[R_t | s_t = s, a_t = a, \pi] \quad (2)$$

值得注意的是,最优策略并不唯一,但是最优的状态动作值唯一:

$$Q^*(s, a) = \max_{\pi} E[R_t | s_t = s, a_t = a, \pi] \quad (3)$$

Q 学习是强化学习中应用最为广泛的学习方法之一. 该方法定义了 Q 函数,将得到的数据代入到式(4)、(5),对 Q 函数进行迭代更新,最后 Q 函数会收敛到最优状态动作值:

$$\delta_t = r_{t+1} + \gamma \max_{a'} Q_t(s_{t+1}, a') - Q_t(s_t, a_t) \quad (4)$$

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t \delta_t \quad (5)$$

其中 t 表示时刻, α_t 是学习率, δ_t 表示时间差分 (Temporal Difference, TD) 误差, a' 是下一状态 s_{t+1}

执行的动作. Q 学习使用 TD 误差进行迭代更新. 可以证明, 当满足所有的状态动作都能够被无限次地遍历并且满足 $\sum_{t=0}^{\infty} \alpha_t^2 < \infty, \sum_{t=0}^{\infty} \alpha_t = \infty$ 时, 在时间步 t 趋近于无穷下, 可以得到最优控制策略^[10-11].

然而, 在大规模问题中, 用类似于上述的表格方法是不可取的^[12]. 在这类问题中, 通常利用函数逼近器来近似求解 Q 值: $Q(s, a | \theta) \approx Q^*(s, a)$. 函数逼近器一般可以用线性函数或者非线性函数构造而成^[13].

2.2 深度 Q 网络

深度强化学习中使用深度神经网络作为函数逼近器, 但是使用非线性神经网络与强化学习结合时会导致学习不稳定^[2]. Mnih 等人提出使用目标网络和经验重放机制来解决该问题.

在 DQN 中使用两个独立网络模型来近似表示状态动作值函数^[14], 分别为当前网络和目标网络, 其中参数 θ_t 用于表示当前网络输出 $Q(s, a | \theta_t)$, θ_t^- 用于表示目标网络输出 $Q(s, a | \theta_t^-)$. 在 DQN 中, 通过损失函数来度量预测值和优化目标之间的差异:

$$L_i(\theta_t) = E[(Y_i - Q(s, a | \theta_t))^2] \quad (6)$$

一般用 $Y_i = r + \gamma \max_{a'} Q(s', a' | \theta_t^-)$ 来近似表示值函数的优化目标, 充当监督学习中的标签. DQN 通过反向传播更新网络参数 θ , 更新的梯度由式(6)对参数 θ 求导得到:

$$\nabla_{\theta_t} L(\theta_t) = E_{s, a, r, s'}[(Y_i - Q(s, a | \theta_t)) \nabla_{\theta_t} Q(s, a | \theta_t)] \quad (7)$$

为防止梯度爆炸, 需要将梯度裁剪至 $[-1, 1]$ 区间内, 对于大于 1 的误差将其置为 1, 小于 -1 的将其置为 -1.

在 DQN 中, 保证学习稳定的另一种做法是使用经验重放机制. Agent 在每一个时间步 t 内存储在线得到的经验 $e_t = (s_t, a_t, r_t, s_{t+1})$, 并将其放入经验池 $D = \{e_1, e_2, \dots, e_N\}$ 中. 在训练网络阶段, 每次从经验池中取出固定数量的样本, 使用随机梯度下降(Stochastic Gradient Descent, SGD)进行更新.

2.3 简单循环单元

不同于传统全连接网络(Full Connection Networks, FCNs), 循环神经网络(Recurrent Neural Networks, RNNs)引入了定向循环, 以解决输入数据之间前后关联性问题, 即 RNNs 可以用来处理序列相关性的数据.

在传统 RNNs 模型中, 隐藏层接收来自输入层的序列 $(x_1, x_2, \dots, x_t)$ 以及上一时刻 RNNs 的隐藏

层输出 $(h_1, h_2, \dots, h_t)$, 根据以下公式, 得到当前时刻输出 $(z_1, z_2, \dots, z_t)$:

$$h_t = f(w_{xh} x_t + w_{hh} h_{t-1}) \quad (8)$$

$$z_t = g(W_{zh} h_t) \quad (9)$$

其中, f 和 g 都是非线性的激活函数, w_{ij} 为不同层之间相连接的权重.

传统 RNNs 会随着时间步的增加而出现“梯度弥散”问题, 严重影响 RNNs 对长时历史信息的记忆能力. Hochreiter 等人^[15]提出使用 LSTM 替代传统的循环神经网络单元, 其通过增加三种类型的门(Gate): 输入门(Input Gate)、输出门(Output Gate)、遗忘门(Forget Gate), 来达到有效记忆历史信息的目的.

然而, 在 LSTM 模型中, 各种门的输出依赖于前一时刻的结果, 这严重限制了网络的运算, 破坏网络的独立性和并行性, 拖慢序列的处理速度. 在 EN-DRQN 模型中, RNNs 部分采用一种简单循环单元的变体^[16], 与 LSTM 单元相比, SRU 单元引入重置门(Reset Gate)平衡历史记忆和输入信息的关系, 同时门控操作不再计算前一时刻网络权重与输入矩阵的乘积. SRU 单元计算过程如下:

$$\bar{x}_t = W_x x_t \quad (10)$$

$$f_t = \sigma(W_f x_t + b_f) \quad (11)$$

$$r_t = \sigma(W_r x_t + b_r) \quad (12)$$

$$\tilde{c}_t = f_t \odot c_{t-1} + (1 - f_t) \odot \bar{x}_t \quad (13)$$

$$h_t = r_t \odot g(\tilde{c}_t) + (1 - r_t) \odot x_t \quad (14)$$

其中 $\bar{x}_t, f_t, r_t$ 分别表示 t 时刻的输入门、遗忘门、重置门, c_t 表示 t 时刻长期记忆的内容, h_t 表示 t 时刻隐藏层单元输出, W, b 表示单元权重和偏置, $\odot$ 表示矩阵或向量之间对应元素的相乘. 完整的 SRU 结构如图 2 所示.

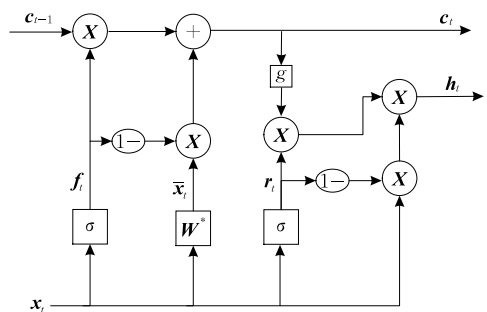


图 2 简单循环单元计算过程图

从式(10)~(12)可以看出 SRU 单元的输入门、遗忘门、重置门的计算已经不需要像 LSTM 门机制那样计算前一时刻的输出与参数矩阵的乘积,

保证了运算的独立性,有效地缓解了网络传播时间过长的问題。此外,SRU 单元比 LSTM 等单元有更少的网络参数,减少了网络运算所需的计算资源。

3 噪音机制的深度循环 Q 网络

本文提出的 EN-DRQN 是基于文献[6]的改进,在其网络结构中,从单一的视频画面映射到行动要经历以下阶段:

(1)卷积神经网络. 将接收到的序列化原始数据作为 CNNs 输入,通过三层卷积操作提取视觉信号中有效特征信息。

(2)循环神经网络. 采用双层 SRU 单元结构,将 RNNs 沿时间轴展开以接收经卷积网处理后的序列化特征,使网络在训练过程中高度关注序列的特征信息以及前后序列的相关性。

(3)带探索噪音处理的全连接网络. 在全连接网络中注入自适应噪音信息,噪音通过方差驱动探索,通过两层全连接网络映射到行动。

本文以 Atari 2600 游戏为例,介绍 EN-DRQN 从接收原始视觉传感信息到做出行为决策的一系列流程,然后将详细阐述预处理、卷积神经网络、循环神经网络、带噪音的全连接网络等部分处理数据的流程及细节,并提出更为一般的带探索噪音的深度循环 Q 网络模型。

3.1 预处理

在训练 Atari 2600 游戏时,需要预处理操作来剔除不重要的信息。对游戏进行预处理本质上是对每一帧画面进行处理。一方面预处理可以减少训练的计算量,另一方面增强了提取特征时的准确性。在 Atari 2600 游戏集中,原始传感输入是尺寸为 $210 \times 160 \times 3$ 的图片。为了计算需要,可以对其进行裁剪。

首先,将原始 RGB 图像转换成 $210 \times 160 \times 1$ 的灰度图,在游戏中使用灰度图像并不会影响 Agent 的表现性能,并且可以减少存储资源、加快网络计算;其次,将 $210 \times 160 \times 1$ 的图片进行降采样处理,本文采用双线性插值法对图像进行缩放,将图片大小缩减为 $110 \times 84 \times 1$;对于降采样后得到的图片,仍存在部分无价值的信息且这种信息集中包含在游戏的边角区域,因此需要进一步裁剪没有价值的像素点以及边界;最终,可以被观测的游戏屏幕区域大小为 $84 \times 84 \times 1$,且被观测的画面足以提供有价值的信息。预处理操作仅需要对图像进行剪裁、降采样和简单的灰度转换,所以并不会影响 Agent 训练游戏的效果,同时将处理后的图片放入 CNNs 中可以大大减少 CNNs 的计算,节省了计算的时间和存储空间,使资源得到有效利用。

在原始的 DQN 中采用距离当前时刻最近的 4 幅经过预处理的图片作为网络的输入。与原始 DQN 不同,EN-DRQN 采用一张预处理的图片作为网络的输入,与 DQN 相比,同一个状态下的计算量缩减至 $1/4$ 。

3.2 CNNs

卷积神经网络可以接收多张图片进行平行化处理^[17],卷积处理过程如下:

$$C=f(\sum_{d=0}^{D-1}X_d \cdot W_d+b)$$
 (15)

其中 D 为图片的深度, X_d 为输入的矩阵, W_d 为卷积核, b 为偏置, C 为卷积层输出的特征图 (Feature Maps), f 为激活函数,本文采用线性整流单元 (Rectified Linear Unit, ReLU) 作为激活函数。

EN-DRQN 使用 CNNs 进行图片特征提取。为增强特征提取的准确性,本文采取三层卷积网络,具体细节如表 1 所示。

表 1 卷积网络前向传播细节

网络层	输入	过滤器	步幅	过滤器个数	激活函数	输出
Conv1	$1 \times 84 \times 84$	8×8	4×4	32	ReLU	$20 \times 20 \times 32$
Conv2	$20 \times 20 \times 32$	4×4	2×2	64	ReLU	$9 \times 9 \times 64$
Conv3	$9 \times 9 \times 64$	3×3	1×1	64	ReLU	$7 \times 7 \times 64$

经过三层卷积和非线性变换,将原始的视觉感知输入编码成特征集合,并进行扁平化处理,以便于 RNNs 操作。

值得注意的是,本文在卷积结束后并没有采取池化 (Pooling) 操作,这是由于池化操作具有平移不变性^[18],即网络最终学习到的结果对图像中的物体位置不敏感。但是在许多游戏中,物体位置是十分关

键的信息,这种信息可能对潜在奖励具有十分重要的影响,因此丢掉物体的位置信息并不合适。

3.3 RNNs

在 RNNs 中,如果 N 个参数矩阵共用一份输入数据 x_i ,那么可以将这些矩阵拼接成一个大矩阵以提高 GPU 的利用率。对于善于处理大矩阵计算的 GPU 而言,把多个小矩阵相乘合并为大矩阵相乘要

比循环计算多个矩阵乘法快很多. 在 SRU 单元中, 输入门 $\bar{x}_t$ 、遗忘门 f_t 和重置门 r_t 共用一份输入数据 x_t , 本文将这三种门的参数 W_x, W_f, W_r 拼接成一个矩阵 $[W_x, W_f, W_r]^T$ 以提高计算的利用率. 与 LSTM 单元相比, 在 SRU 单元的门控操作中, 输入门 $\bar{x}_t$ 、遗忘门 f_t 和重置门 r_t 已经不再像 LSTM 单元那样计算前一时刻的输出与参数矩阵的乘积, 时间相关性的历史记忆仅需要变量 c_t, h_t 存储, 且 c_t, h_t 的计算采用矩阵之间元素的点乘操作, 对于 GPU 而言, 计算矩阵之间点乘的时间小于矩阵乘法, 一定程度上降低了计算的复杂度.

不同于 DQN 的 4 张图片作为一个状态, 在 EN-DRQN 中, 以一帧视频信号作为状态. 对于长度为 N 的情节, 定义智能体的决策长度为 $L (L \leq N)$, 若对第 i 帧做决策, 当 $i < L$ 时, Agent 考虑之前的 i 帧做决策, 当 $L \leq i \leq N$ 时, Agent 会对前 L 帧画面进行考虑, 使决策更加具有战略性. 图 3 展示了一个情节长度为 N , Agent 的决策长度 $L=3$ 时的决策图.

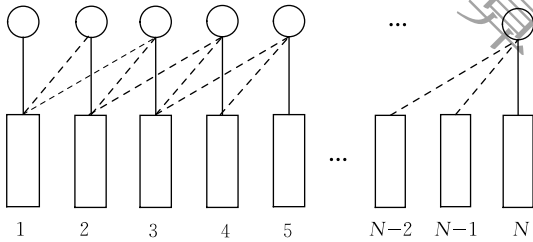


图 3 EN-DRQN 决策示意图

Donahue 等人^[19]的研究表明, 使用双层记忆组件组成的 RNNs 在记忆时序性信息上优于单层和四层的表现, 考虑到网络开销和训练时间成本, 本文采用双层 SRU 单元所组成的 RNNs 网络.

在 DQN 中, 将卷积层得到的特征图直接连接到包含 512 个隐藏单元的下一网络层中. 在 EN-DRQN 中, 增加循环神经网络部分以记忆历史相关信息. 循环层的输入由两个部分组成, 一部分是上一时刻隐藏层激活值 h_{t-1} , 另一部分是卷积层传来的特征序列.

对于情节长度为 N 的特征序列, 需要将 RNNs 沿时间轴展开, 把特征序列输入到 RNNs 的输入层. 记输入序列为 $S = \{s_1, s_2, \dots, s_N\}$, 其中 $s_i \in \mathbf{R}^m$, 表示卷积层得到的特征, 下面详细说明 EN-DRQN 在 RNNs 层处理数据的流程:

(1) 组合当前时刻的输入层序列特征 s_t 和上一时刻隐藏层的激活值 h_{t-1} , 得到输入 $I_t = \{s_t, h_{t-1}\}$,

并将 I_t 作为 RNNs 中第一层的输入.

(2) 将 I_t 输入至第一层隐藏层细胞中, 通过激活函数将激活信息输出. 把第一层激活信息作为第二层输入层的输入信息. 本文设置的隐藏层有 512 个 SRU 单元, 故激活信息可表示为 $h_t = \{h_t^1, \dots, h_t^{512}\}$, 其中 h_t^i 表示 t 时刻第 i 个 SRU 组件的输出.

(3) 按照流程(2)相同的方法输出 RNNs 的最终信息. 此信息一方面记忆了 t 时刻之前数据的相关性, 可以作为沿时间轴上 $t+1$ 时刻的输入信息, 另一方面当作 RNNs 输出层的输出以作为下一个网络部分的输入.

RNNs 具有记忆历史信息特性, 然而记忆历史信息能力要依赖于合理的参数更新. 本文采用自举的随机更新方法: 采样情节随机从经验池中取出, 在情节中随机选择一个时间步, 根据 RNNs 展开的长度从该点进行训练. 例如: 假设一个情节长度为 N , 规定输入训练序列长度为 $L (L < N)$, 从情节中随机选择一个时间点 $P (P+L < N)$, 对于长度为 L 的输入序列, RNNs 展开的时间步长亦为 L , 故从 P 点开始, 把长度为 L 的序列输入网络进行训练. 目标网络更新也是如此. 通过大量反复实验, 保证每个情节中的时间步样本均可被采样. 由于在不同情节下, 序列与序列之间并不存在相关性, 所以只学习记忆序列内部时间步之间的联系, 为此本文采用以下做法: 每进行一次更新, RNNs 的细胞初始化状态值为 0, 网络只记忆输入序列中时间步的相关性而不必记住序列之间的相关性. 更一般地, 如果序列之间存在关系且需要学习序列之间的相关性, 可以将其输入至网络而不必在每次更新时清零. RNNs 网络训练采用通过时间反向传播方法 (Back Propagation Through Time, BPTT). 采用随机自举的更新更好地遵循经验随机采样的特性, 增加了训练可靠性.

3.4 探索噪声

使用循环神经网络, 让 Agent 拥有一定记忆能力, 但是仅考虑前 L 帧是不够的. 行为策略的决策要考虑探索问题, 探索可以为决策提供更多有价值的信息. “探索-利用”的平衡一直是强化学习中研究的热点^[20].

传统的强化学习算法在动作空间采用 ϵ -greedy 进行决策: 以 ϵ 的概率进行探索, 以 $1-\epsilon$ 的概率利用. 然而, 这种抖动策略存在两方面的缺陷: (1) 如何选择 ϵ 探索是具有挑战性的. DQN 的做法是将 ϵ 从 1 开始每隔 50 000 步呈指数级衰减, 直到衰减至

0.00025 后将其固定,但是这种做法不能保证一定学习到有效、合理的 ϵ ,以达到探索的最优;(2) 仅根据状态值函数选择动作即仅在动作空间选择动作并不会引起策略的改变,不能达到深度探索的目的。

本文考虑在网络中加入噪音驱动 Agent 探索.在网络中加入噪音驱动探索的主要动机是在网络中微小的变化会在未来多个时间步内造成持续、内在、复杂的改变^[21],这种改变导致模型输出的策略发生变化,从而可能使策略在下一阶段选择动作时发生很大变化.在一个情节的每个时间步,EN-DRQN 模型均会从噪音分布中采样白噪音输入至网络.当 Agent 与环境交互过程中采样到一整条轨迹时,轨迹中每个时间步输出的 Q 值均受到噪音影响,导致每个时间步的网络输出发生变化,基于 Q 值函数的策略从而也发生了变化.这种在网络空间注入的噪音影响到了整个情节中动作的选择,使探索具备了深度.而在 DQN 等模型中,没有在网络空间引入探索噪音,仅依靠较小的参数 ϵ 保持在动作空间抖动探索,每个情节的网络模型参数并没有发生改变。

Salimans 等人^①研究表明,对神经网络再参数化所引起的抖动对动作选择是有意义的.然而 Salimans 等人对网络的再参数化方法仅局限于进化方法,并不能说明在深度强化学习领域可以获得好的表现.受 Salimans 等人启发,本文提出 EN-DRQN 模型,采用双层改进 SRU 单元结构,同时将探索机制引入至网络中,利用方差驱动网络输出变化以达到探索的目的.这些参数通过损失函数反馈至网络中,与网络中的其它参数一并学习,具体细节见 3.4.1 节及 3.4.2 节。

3.4.1 动作空间噪音与网络空间噪音

对神经网络的再参数化需要设计合理的抖动策略以保证 Agent 能够有效探索.在学习过程中,首先需要考虑探索和利用的问题。

为增强探索,考虑到两种增加探索的方法:在动作空间和网络参数空间注入噪音.在动作空间加入噪音,得到动作 $a_t = \pi_\theta(s_t) + N_{s_t}(0, \sigma^2)$,其中 $N_{s_t}(0, \sigma^2)$ 为人工选取的高斯噪音项.对于每一个状态 s_t ,即使策略给出最优动作 a_t^* ,由于噪音的存在,最优动作 a_t^* 不一定被选择.在深度学习中,深度神经网络陷入局部最优的概率难以被衡量,但如果陷入局部最优,策略在收敛的过程中,噪音项对最优动作的影响逐渐减小,动作空间中均值为 0 的噪音项

所驱动的探索将意义甚微.同时由于噪音项依赖于当前状态,不会对后续状态产生影响,所以并不是一种有效的探索方式.相反,在网络参数中加入噪音可以引起网络抖动,使驱动策略做出不同的行为模式.根据抖动网络所学习到的当前策略 $\tilde{\pi}_\theta$,在状态 s_t 下所做出的最优动作 $a^* = \tilde{\pi}_\theta(s_t)$ 一定被选择.加入至网络空间中的噪音是均值为 0,方差为 σ 的高斯白噪音,其中 $\sigma \neq 0$.由于网络参数的微小变化会影响至后续多个状态,这种白噪音所引起的微小抖动使探索具有了深度.探索可以通过白噪音的方差进行学习:Agent 在策略的指导下,若选择的动作为当前策略下最优动作,Agent 会减小方差规模以提高最优动作被选中的概率,由于方差不会为 0,对于次优动作依然有被选中的可能性,以保证探索.若该动作不是最优动作,提高方差,加大最优动作被选择的可能性.故本文选择在网络中注入噪音机制。

在网络参数中注入探索噪音需要控制合适的噪音规模.对于模型的全连接层部分,不同的网络层对噪音的敏感性可能不同,因此本文对每层均设置了控制噪音规模的参数因子 σ , σ 负责对探索噪音值进行合理缩放以使网络的方差控制在合理的范围内.因子 σ 在训练过程中通过反向传播算法自适应修正.在网络收敛阶段, σ 学习到了合适的值,此时加入的噪音并不影响最优动作的选择;与其它动作相比,最大的 Q 值所对应的动作的相对性并没有发生改变。

图 4 展示了在动作空间和网络空间中注入噪音的比较。

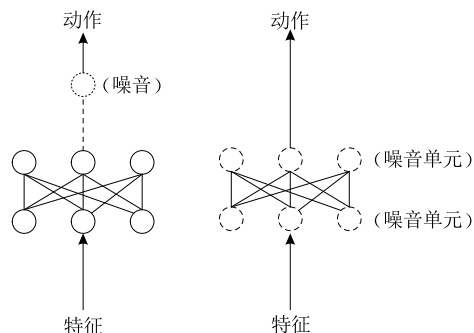


图 4 在动作空间和网络参数空间的噪音(图中实线圆形代表网络的神经单元,虚线圆形代表注入噪音的神经单元,实线代表网络的连接,虚线代表预处理操作)

3.4.2 带有噪音的网络

对于网络的全连接部分,DQN 中通过两层全连

① <http://arxiv.org/abs/1703.03864>

接来映射动作,而在 EN-DRQN 中,本文对全连接层加入可学习的噪声以增强其探索性. 噪声参数来自于噪声分布,这些参数通过梯度下降训练. 具体的:可以把噪声神经网络抽象为 $y=f_{\theta}(x)$,对于输入向量 x 和输出向量 y , θ 表示已噪音化的网络参数: $\theta=\mu+\sigma\odot\psi$,其中规定 $\zeta=(\mu,\sigma)$ 是一系列可学习的网络参数, ψ 是一个均值为 0 的噪声向量, $\odot$ 表示向量中元素之间的乘法运算. 训练网络就是优化可学习的参数 ζ .

对于有 p 个输入和 q 个输出的全连接层可以抽象表示为:

$$y=W \cdot x+b \quad (16)$$

其中 $x \in \mathbf{R}^p$ 表示输入向量, $W \in \mathbf{R}^{p \times q}$ 表示隐藏层全连接的参数矩阵, $b \in \mathbf{R}^q$ 表示偏置. 为利用噪声以驱动探索,做出如下改进:将 W 替代为 $\mu^w + \sigma^w \odot \psi^w$, 将 b 替代为 $\mu^b + \sigma^b \odot \psi^b$, 其中 $\mu^w \in \mathbf{R}^{p \times q}$, $\mu^b \in \mathbf{R}^q$, $\sigma^w \in \mathbf{R}^{p \times q}$, $\sigma^b \in \mathbf{R}^q$ 是网络中可学习的参数, $\psi^w \in \mathbf{R}^{p \times q}$, $\psi^b \in \mathbf{R}^q$ 是随机噪声,那么 y 可以表示成 $y = (\mu^w + \sigma^w \odot \psi^w)x + \mu^b + \sigma^b \odot \psi^b$.

随机噪声产生方式有如下两种选择:

(1) 独立的高斯噪声. 权重和偏差的噪声是独立的. 对于随机噪声矩阵 ψ^w 中的每一个元素 $\psi_{i,j}^w$ 都是从一个高斯分布中独立产生的, ψ^b 亦是如此. 那么,则带有噪声的全连接层有 $p \times q + q$ 个噪声变量.

(2) 可分解的高斯噪声. 可分解的高斯噪声减少了噪声变量的个数. 生成有 p 个元素的噪声向量 ψ_i , 有 q 个元素的噪声向量 ψ_j , 并进行如下分解:

$$\phi_{i,j}^w = f(\psi_i) f(\psi_j) \quad (17)$$

$$\psi_j^b = f(\psi_j) \quad (18)$$

其中 $\phi_{i,j}^w$ 表示两层全连接之间噪声权重矩阵的每个噪声值. ψ_j^b 表示输出层偏置矩阵的每个噪声值, f 是一个分解函数, 本文使用 $f = \text{sgn}(x) \sqrt{|x|}$. 那么, 网络有 p 个输入单元, q 个输出单元, 只需要 $p+q$ 个噪声变量, 大幅度减少了计算代价.

网络每一次前向计算时, Agent 从噪声分布中采样噪声, 通过分布的方差进行探索. 在优化之中, Agent 根据当前固定的网络参数执行动作, 保证 Agent 可以根据当前噪声分布中采样的噪声做出决策.

传统的 DQN 中, Agent 根据 ϵ -greedy 选择动作, 本文做出如下改进: 在 EN-DRQN 中, 由于网络已经具有探索的能力, 故不再使用 ϵ -greedy 策略, 而是使用贪婪策略; 其次在网络的全连接层注入噪声, 通过方差驱动探索. 本文采用可分解的高斯噪声以减小噪声参数, 加快网络的计算. 由于高频的噪声采样会导致较大的方差, 本文从目标网络中选取动作以减小状态动作值函数的扰动. 完整的 EN-DRQN 如图 5 所示.

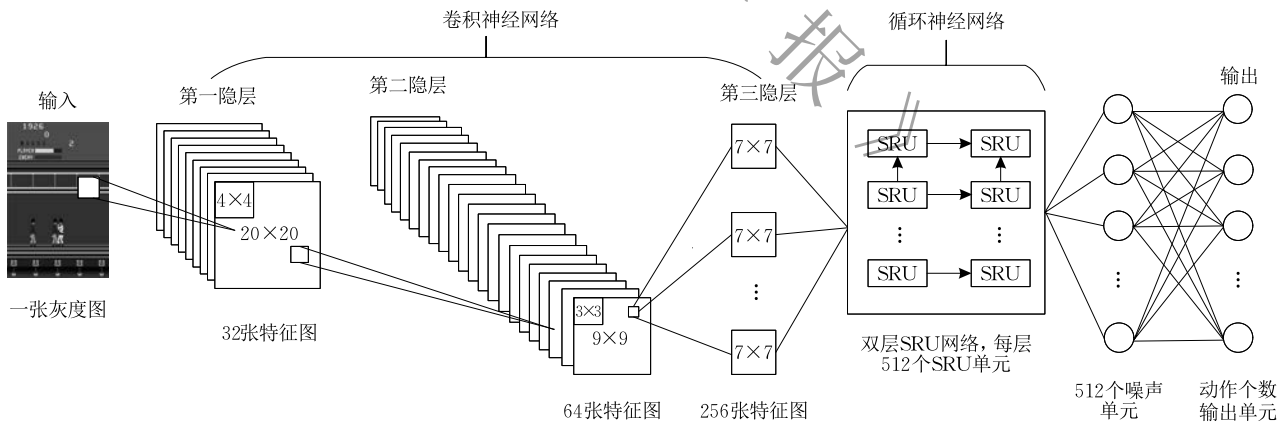


图 5 EN-DRQN 模型图(EN-DRQN 采用三层卷积网络、两层 SRU 单元的循环网络和两层带探索噪声的全连接神经网络. 网络输入为一张图片, 网络输出的单元数由任务定义的动作个数决定)

3.5 广义的带探索噪声的深度循环探索 Q 网络

对于更一般的任务, 若任务的输入为原始的图片, 可以先对图片进行裁剪、降采样等预处理操作. 假设预处理完成后的图片大小为 $M \times N \times C$, 其中 M 、 N 、 C 分别为图片的长度、宽度以及通道数. 可以通过设计卷积神经网络进一步提取有效特征, 网络中卷积层个数为 L_c , 卷积核大小为 $F \times F \times C$, 并根

据实际需要决定是否设计池化操作; 将卷积后得到的特征按照任务要求的序列组合输入至 SRU 循环神经网络, 并设计循环网络的层数为 L_r 层, 每层 SRU 单元数为 K ; 最后设计 L_f 层全连接网络, 并对其噪音化. 噪音分布设计为高斯分布且设置均值为 0, 方差为 var , 噪音可以选择独立的高斯噪音或可分解的高斯噪音.

3.6 模型训练

EN-DRQN 是一个平滑的模型,其三大模块 CNNs、RNNs、FCNs 均是可微的. 本文采用自适应学习率的随机梯度下降方法对网络参数进行更新.

在 DQN 中使用神经网络逼近状态动作值函数,其中最优状态动作值函数通过最小化神经网络的参数 θ 而获得:

$$L_i(\theta_i) = E_{s,a,r,s'}[(y_i^{\text{DQN}} - Q(s,a;\theta_i))^2] \quad (19)$$

其中 y_i^{DQN} 为标签,可表示为 $y_i^{\text{DQN}} = (r_i + \gamma \max_{a'} Q(s', a', \theta^-))$. $L_i(\theta_i)$ 为当前参数 θ_i 期望均方误差, θ^- 代表目标网络的参数, θ 代表当前网络参数. 每隔一定步数 C , 将当前网络参数复制给目标网络, 引入目标网络可以使学习更加稳定, $\gamma \in (0, 1]$ 表示折扣因子, Agent 采用 ϵ -greedy 选择动作,即以 ϵ 鼓励探索动作,以 $1-\epsilon$ 的概率选择贪心动作,式(19)的梯度如下:

$$\nabla_{\theta_i} L_i(\theta_i) = E_{s,a,r,s'}[(y_i^{\text{DQN}} - Q(s,a;\theta_i)) \nabla_{\theta_i} Q(s,a;\theta_i)] \quad (20)$$

在 DL 中要求训练数据必须是相互独立的,而 RL 与环境交互而产生的样本之间是高度相关的,为使 RL 的方法可以在 DL 中成功实践,采用经验重放技术以打破序列样本的相关性. 在经验重放技术中使用样本池 $D = \{e_1, \dots, e_t\}$ 来存储经验,经验样本 $e_t = (s_t, a_t, r_t, s_{t+1})$ 以先进先出(First In First Out, FIFO)的形式加以存储,每次更新随机取出 Mini-Batch 数量进行随机梯度下降更新,一定程度上打破了样本之间的相关性.

EN-DRQN 中,由于采用噪音机制,可学习的噪音用 ζ 表示,故损失函数可表示成在噪音之上的期望,其梯度可表示为:

$$\nabla \bar{L}(\zeta) = \nabla E[L(\zeta)] = E[\nabla_{\mu,\Sigma} L(\mu + \Sigma \odot \psi)] \quad (21)$$

对于式(21),本文采用蒙特卡洛逼近的方法进行逼近以保证损失的无偏性,即:

$$\nabla \bar{L}(\zeta) \approx \nabla_{\mu,\Sigma} L(\mu + \Sigma \odot \psi) \quad (22)$$

具体细节如下:

令 $y_i = r + \gamma \max_{a'} Q(s', a', \psi'; \zeta^-)$ 得到:

$$\bar{L}(\theta_i) = E_{\zeta}[E_{s,a,r,s'}[y_i - Q(s,a,\psi;\zeta)^2]] \quad (23)$$

其中外部期望针对的是 $Q(s,a,\psi;\zeta)$ 和 $Q(s',a',\psi';\zeta^-)$ 中的噪音向量 ψ 和 ψ' 的分布.

4 实验及结果分析

本节介绍了实验所使用的平台以及实验过程中使用的参数,随后对 Atari 2600 游戏集中的部分战略性游戏以及带有稀疏奖赏的游戏进行了评估. 评估时采用 DQN、EN-DQN、LSTM-DQN、EN-DRQN 4 个模型进行比较,其中 DQN、LSTM-DQN 使用在动作空间抖动的探索方式. 与 DQN 相比,LSTM-DQN 增加了 LSTM 门机制以记忆历史信息. EN-DQN 是在 DQN 基础上,在网络空间使用探索噪音的模型, EN-DRQN 是本文提出的基于简单循环单元的带探索噪音机制网络. 不同模型在训练过程中使用相同的参数集合.

4.1 实验平台描述

本文采用 OpenAI Gym^① 中 Atari 2600 游戏环境作为实验环境. OpenAI Gym 是开发和对比强化学习算法的一个开源工具包,其提供了各种类型的 Atari 2600 游戏环境接口,涉及到体育竞技、桌游、射击以及战略性的策略游戏,为人工智能研究者提供了各种具有挑战性的任务.

为了有效评估模型的表现,本文选择 gym 平台中具有战略性和稀疏奖赏性的 8 个游戏进行实验. 游戏的简要介绍已列于表 2,可以看出游戏的类型繁多,Agent 也需要完成不同的目标任务,最终学习策略也不同.

表 2 游戏的简要介绍

英文名	中文名	游戏类型	游戏动作数	任务目标
AirRaid	空袭	射击类	6	Agent 避开敌人攻击并且适时击落目标
BeamRider	波速骑手	射击类	9	Agent 避开子弹并且要击中移动的敌人
Carnival	狂欢	策略类	6	Agent 在规定时间内躲避并击杀掉下的敌人
Centipede	蜈蚣博弈	策略类	18	Agent 避障且合理击打上方蜈蚣形状的敌人
Freeway	过马路	策略类	3	Agent 合理避开连续、移动的车辆
StarGunner	星际炮手	射击类	18	Agent 要做到避障和打掉敌人
TimePilot	时空飞机	射击类	10	Agent 驾驶飞机,避障同时攻击敌人
MontezumaRevenge	蒙特祖玛的复仇	冒险类	18	Agent 避开敌人以及机关获取宝藏

4.2 实验参数设置

本文采用一些方法以减小在训练过程中出现的不稳定现象. 首先,Agent 根据评估的 Q 值选择动

作的时间大于网络传播的时间,所以采用跳帧技术来缓解二者之间的计算^[22]. 本文选取跳帧参数为 4,

① <https://github.com/openai/gym>

在跳帧期间采取相同的动作,以 4 帧相同动作下的累积奖赏作为总奖赏,其中跳帧期间如果出现中止状态则结束游戏;其次,为提高算法的稳定性,防止出现大的波动而导致 Q 值发散,将损失函数进行分段裁剪处理.如果损失的绝对值在 $[-1,1]$ 之间,则采用式(6)所示的均方误差函数,否则使用绝对值函数;最后,由于游戏的奖赏区间范围波动较大,会导致不同游戏得分差异较大,实验时将所有正的奖赏设置为 1,负的奖赏设置为 -1 ,零奖赏保持不变,以方便各算法之间的比较.

为进行有效对比,在不同模型中采用相同的参数集合,使用均方根随机梯度下降方法(Root Mean Square Propagation,RMSProp)更新网络参数,动量系数设置为 0.95,折扣因子 γ 统一设置为 0.99.经验重放的样本池大小设置为 5 000 000. EN-DRQN 的决策长度为 4.在训练初期,样本经验池中采样的数据达到 50 000 才进行训练以防止样本太少导致学习效果较差.当前网络每隔 10 000 步将参数复制给目标网络.在经验重放时,从样本池 D 中随机抽取 Mini-Batch 为 32 的样本数据.在 EN-DRQN 模型中,采样的噪音分布是均值为 0,方差为 1 的高斯分布,全连接参数 μ 初始值在 $[-1/\sqrt{p},1/\sqrt{p}]$ 区间内采样, σ 初始值设置为 $0.4/\sqrt{p}$,其中 p 为输入层单元数.

对于 LSTM-DQN 网络,本文采取三层卷积、两层 LSTM 网络和两层全连接网络.对于卷积层部分,LSTM-DQN 网络的参数与 EN-DRQN 参数一致;LSTM 网络部分每层 512 个 LSTM 单元,全连接层部分隐藏层有 512 个神经元,输出单元的个数是游戏定义的动作个数.在 LSTM-DQN 中,所有的激活函数选用 Relu 函数.

4.3 实验评估与结果分析

在 RL 中,Agent 在每个时间步 t 内根据策略 π 选择行动 a_t 并与环境交互获得奖赏 r_{t+1} ,通过从情节开始到结束所获得的累计奖赏来评价学习过程的好坏.DL 中,一般通过分阶段训练网络模型,并在训练集和测试集验证模型的学习效果.在 DRL 中结合两种评估方式,在训练集上,通过分阶段训练,比较每个阶段中平均每个情节的累计奖赏进行评估模型的好坏,在测试集上,通过测试每个情节的累计奖赏评估模型学习的效果.

在实验中采用 100 个训练阶段(Epoch)作为训练周期,其中每个阶段 200 000 步进行训练网络参数,在测试过程中取 100 个情节,每个情节 10 000 步,比较每局游戏的得分.

本文首先比较 LSTM-DQN、EN-DRQN 2 种循环网络训练表 2 中游戏时,使用 Ubuntu 16.04 操作系统,在 NVIDIA Tesla P40 GPU 上的计算时间,结果如表 3 所示.

表 3 RNNs 模型训练时间

环境	模型	每时间步耗费时间/s	训练总时间/h
AirRaid	LSTM-DQN	0.001 72	95.82
	EN-DRQN	0.001 70	93.54
BeamRider	LSTM-DQN	0.0141	78.4
	EN-DRQN	0.0137	76.1
Carnival	LSTM-DQN	0.0315	174.87
	EN-DRQN	0.0285	158.34
Centipede	LSTM-DQN	0.0245	130.2
	EN-DRQN	0.0243	128.4
Freeway	LSTM-DQN	0.0192	106.5
	EN-DRQN	0.0174	96.7
StarGunner	LSTM-DQN	0.0122	67.8
	EN-DRQN	0.0121	67.3
TimePilot	LSTM-DQN	0.0307	170.68
	EN-DRQN	0.0304	169.37
MontezumaRevenge	LSTM-DQN	0.0139	77.43
	EN-DRQN	0.0130	72.51

表 3 展示各模型在玩不同游戏时,在 GPU 上的训练时间.其中,训练 Carnival 游戏时,二者之间的差异最明显,EN-DRQN 在训练总时间上比 LSTM-DQN 减少了 16.53 个小时.

虽然 LSTM、SRU 都是根据门控操作来处理状态之间存在的依赖关系,但是 LSTM 的门控机制以及循环结构较为复杂.基于 LSTM 的深度 Q 网络在处理序列数据时,门机制以及记忆细胞 c_t 依赖上一时刻运算结果 h_{t-1} 、 c_{t-1} ,导致网络训练较慢,而在基于 SRU 机制的深度循环 Q 网络中,门机制已经不再需要计算前一时刻的输出与参数矩阵的乘积,且长时刻的历史记忆通过矩阵元素之间的乘法操作计算,降低了循环单元的复杂度.表 3 充分说明新模型在训练时间上的优势.

本文在接下来比较 DQN、EN-DQN、LSTM-DQN、EN-DRQN 4 种模型在表 2 描述的难度系数大的 8 款游戏上,各阶段平均每情节获得的奖赏.实验结果如图 6 所示.

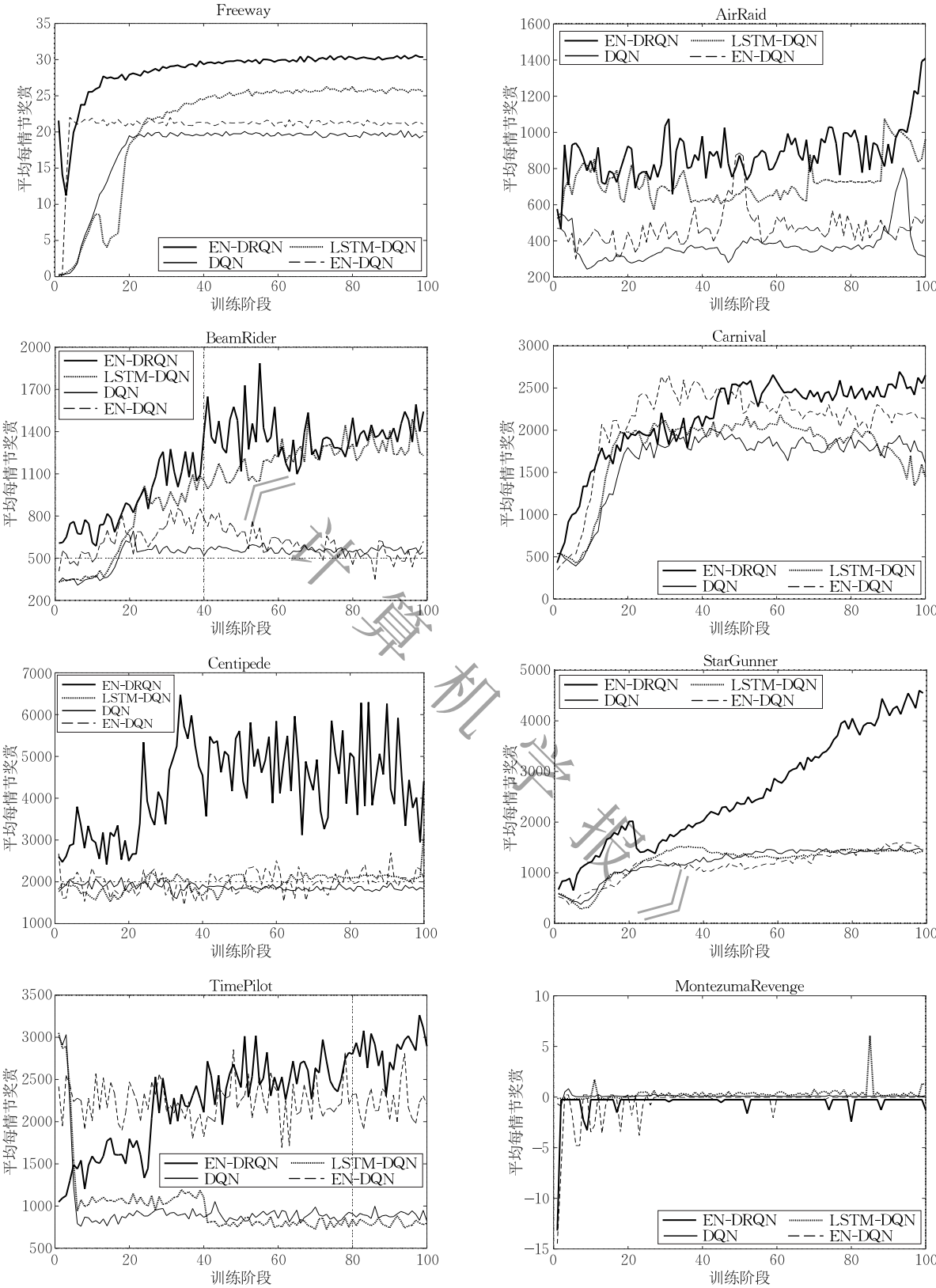


图 6 不同模型实验效果

Busoniu 等人^[23]表明:当强化学习算法与非线性函数逼近器相结合时,会导致算法的收敛性难以得到保证;深度强化学习算法采用大型的深度神经

网络作为非线性函数逼近器来学习值函数或者状态动作值函数,因而算法的收敛性无法从理论上得到保证.然而,Mnih 等人^[6-7]提出了 DQN,引入了经验

重放缓冲区和单独的目标网络,使得算法的稳定性得到显著的改善。

下面以 Freeway 游戏为例,重点分析实验结果。

在 Freeway 游戏中,Agent 要穿过马路到达另一端,期间会出现车辆进行阻碍。Agent 如果被车撞到并不会终止游戏而是后退若干步,每通过一次马路获得一分,比较在规定时间内成功到达目标的次数(分数)。由于 Agent 要避过许多障碍到达终点才可以拿到一分,期间所做的动作获得奖赏均为 0,故该游戏的奖赏是稀疏的,需要多个时间步后才可以获得有效奖赏。

如果输入状态是离当前时刻最近的 4 幅连续画面(DQN),Agent 根据这 4 张图做出动作,并未考虑到 4 张图的连续性和相关性,同时 Agent 依据 ϵ -greedy 策略采取行动,由于浅层的探索并没有使 Agent 找到合理的动作,故 DQN 收敛至 20 分左右。

EN-DQN 中通过噪音机制学习到的策略 $\pi_{\theta}(s_t)$ 可以探索到新的状态,使 Agent 具有获得学习到更多奖赏的潜力。但是探索仅在连续的 4 张图中,未考虑到这连续 4 张图之间的因果关系,做出的策略缺乏一定的战略性,探索出更多状态却无法得到合理的利用。

LSTM-DQN 输入信息是当前画面的一帧,通过输入具有序贯性的画面使其可以根据历史信息做出有效的决策,但是历史信息 h_t 直到 h_{t-1} 计算完毕才可以被计算,这使 LSTM-DQN 需要大量的训练时间。

EN-DRQN 保留了 LSTM-DQN 的优势,在考虑到历史信息做出决策的同时,使用双层的 SRU 细胞来记忆更长时间步内有效的历史信息,有效缓解了当环境奖赏是稀疏情况时回报延时高的问题,其次增加噪音探索机制使 Agent 在根据历史信息作出决策的同时探索新的状态,为决策提供更多的信息。相对于 LSTM-DQN,SRU 单元的计算不必等到 h_{t-1} 时刻结束,加速训练的时间,使 Agent 可以更快地学习到一个有效的策略。

从 Agent 在 Freeway 游戏上的表现可以看出各模型平均每个情节的得分:EN-DRQN 最优,游戏可以得到 30 分左右的分数,即在系统规定的时间内度过马路 30 次以上,明显好于 LSTM-DQN、DQN、EN-DQN。4 种模型在最后均表现稳定,LSTM-DQN 在 Freeway 游戏环境下的得分约为 25 分,优于 EN-DQN 的 22 分和 DQN 的 20 分。在 DQN 中,动作的决策

依赖于连续 4 张图片,然而学习到前后图片的关系合理地规避车辆可以缩短到达目标的时间。但规避期间所获得奖赏均为 0。DQN 因无法区分不同状态下的序贯性,因此得分较低。在 EN-DQN 中,通过高频次抖动,造成 Agent 在某片区域内来回穿梭,这样是有意义的,因为车辆在行驶,出现了更多新的状态,随着训练次数的增加,逐渐地学习到在新的状态下所应该执行的动作。EN-DRQN 结合了 LSTM-DQN 中根据历史信息做出决策的优势,同时加入探索机制,可以根据历史序列做出探索,使 Agent 在长时间步下做出最优决策。

对于表 2 中其它的游戏,从图 6 可以看出本文提出的 EN-DRQN 模型比其余 3 个模型获得了更好的效果,在 Centipede、StarGunner 上表现更加明显。

在 Centipede 游戏中,Agent 不仅需要躲避蜘蛛攻击,而且需要进攻有攻击性的蜈蚣形状的敌人,具有一般游戏不具有的战略性,因此属于较为复杂的游戏类型。该类型使用传统模型效果并不令人满意。LSTM-DQN 虽然考虑了序列的相关性,然而探索性不足,无法探索到更多有价值的样本,导致仅能躲避蜘蛛的攻击,却无法进攻蜈蚣,无法获得更高的得分,故 LSTM-DQN 的效果并不理想。EN-DRQN 在考虑状态相关性的同时进一步在序列间进行探索,这种探索可能会在若干步之后获得奖赏,探索出的新的状态增加了 Agent 获取高分的可能性,所以得分远高于其它模型。

然而在 BeamRider 等游戏中总体性能上提升并不是很显著。一方面游戏操作难度大,Agent 面对复杂的游戏环境在学习过程难以捕捉到有效信息;另一方面游戏环境是随机变化的,前后关联性不大。即使是在这种环境中,EN-DRQN 模型依然是 4 种模型中平均奖赏值最高的一种,但是其优势不明显。

值得注意的是,在 TimePilot 游戏中,DQN 与 LSTM-DQN 的评估曲线随着训练的进程不升反降,这是由于 TimePilot 游戏奖赏稀疏。对于 DQN 与 LSTM-DQN 的探索策略而言,Agent 在和环境交互过程中在动作空间探索样本, ϵ -greedy 探索策略并不像噪音策略那样可以维持高效率的探索,仅依靠在动作空间的抖动导致在该环境下采样到大量无意义的样本,这些样本的指导最终导致了 Agent 学习恶化。

然而,在 MontezumaRevenge 游戏中,EN-DRQN 模型相比于其它模型并没有取得优势。这是由于游

戏的巨大难度导致. 在游戏的第一周目, Agent 需要翻过梯子、利用绳子翻越并躲过怪物的追击才能获得钥匙, 该钥匙需要在第三周目躲过障碍才能使用, 期间获得的奖赏均为 0 且 Agent 容易死亡. 由于游戏难度过大、奖赏过于稀疏, Agent 在执行数百个动作之后依然无法获得奖赏. 即使是在本游戏中效果略好于 EN-DRQN 的其它模型, 相较于人类玩家,

也存在巨大差距.

在 EN-DRQN 模型中, 循环层的 SRU 单元个数将会影响 Agent 记忆历史信息的能力. 本文选取 BeamRider、Freeway、StarGunner、TimePilot 等 4 个游戏, 分别使用含有 512、1024 个 SRU 单元的 EN-DRQN 来比较 Agent 的学习效果, 实验结果如图 7 所示.

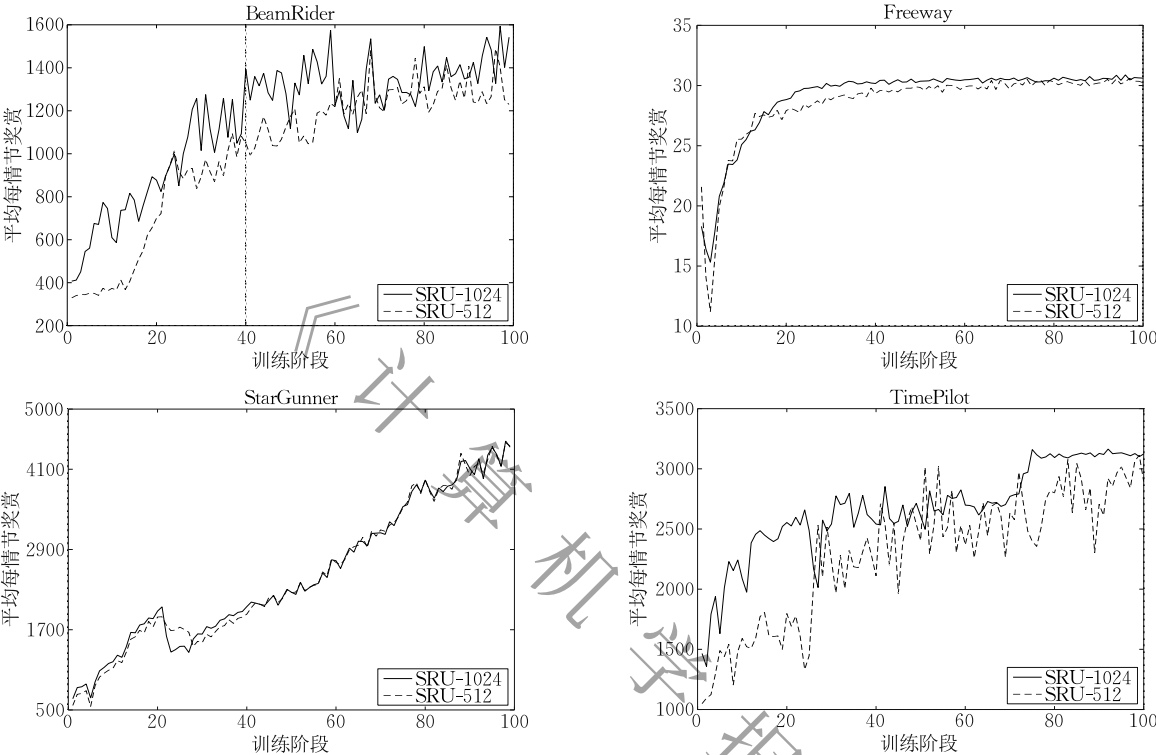


图 7 不同 SRU 单元个数实验效果对比图

从图 7 可以看出, SRU 的个数对不同游戏有不同的影响. BeamRider、TimePilot 两个游戏需要较多的 SRU 单元数量以增强历史记忆, 1024 个 SRU 单元的 EN-DRQN 效果优于 512 个 SRU 单元的效果; 在 Freeway 和 StarGunner 两个游戏中, 512 个 SRU 的 EN-DRQN 足够满足任务需要, 所以 1024 个 SRU 单元的 EN-DRQN 并没有很大的效果提升.

EN-DRQN 采用噪音进行探索, 噪音来自于均值为 0 的高斯分布, 高斯分布的方差将会影响网络的波动. 为进一步探索噪音分布的方差对训练结果的影响, 本文选取均值为 0, 方差为 1 的高斯噪音分布和均值为 0, 方差为 0.1 的高斯噪音分布, 在 Carnival、StarGunner、TimePilot、BeamRider 等 4 个游戏上分别进行评估, 结果如图 8 所示.

从图 8 可以看出不同的游戏对噪音的敏感程度不同. 在 Carnival、StarGunner、BeamRider 游戏中,

方差为 1 的 EN-DRQN 效果略优于方差为 0.1 的 EN-DRQN, 说明在这些环境中, 需要大量的探索以保证 Agent 可以充分的学习. 在 TimePilot 环境中方差为 1 的 EN-DRQN 效果远不如方差为 0.1 的 EN-DRQN 模型, 说明在这种游戏环境下, 过大的方差虽然可以让 Agent 探索出更多的行为轨迹, 但是这些轨迹对 Agent 的学习并没有太大帮助.

在网络中注入噪音以驱动探索, 然而网络参数出现的随机性可能会对输出的映射关系造成的影响, 使 Agent 做决策时产生一定的差异. 本文以 Freeway 游戏为例, 选择噪音网络初始权重服从 $[-1/\sqrt{p}, 1/\sqrt{p}]$ 的均匀分布, 其中 p 为输入层单元数; 以及均值为 0, 方差为 1 的高斯截断分布, 独立试验三次以说明同一种分布在学习过程中随机波动并不会影响学习到的最终策略, 实验结果如图 9 中的 (a)、(b) 图所示.

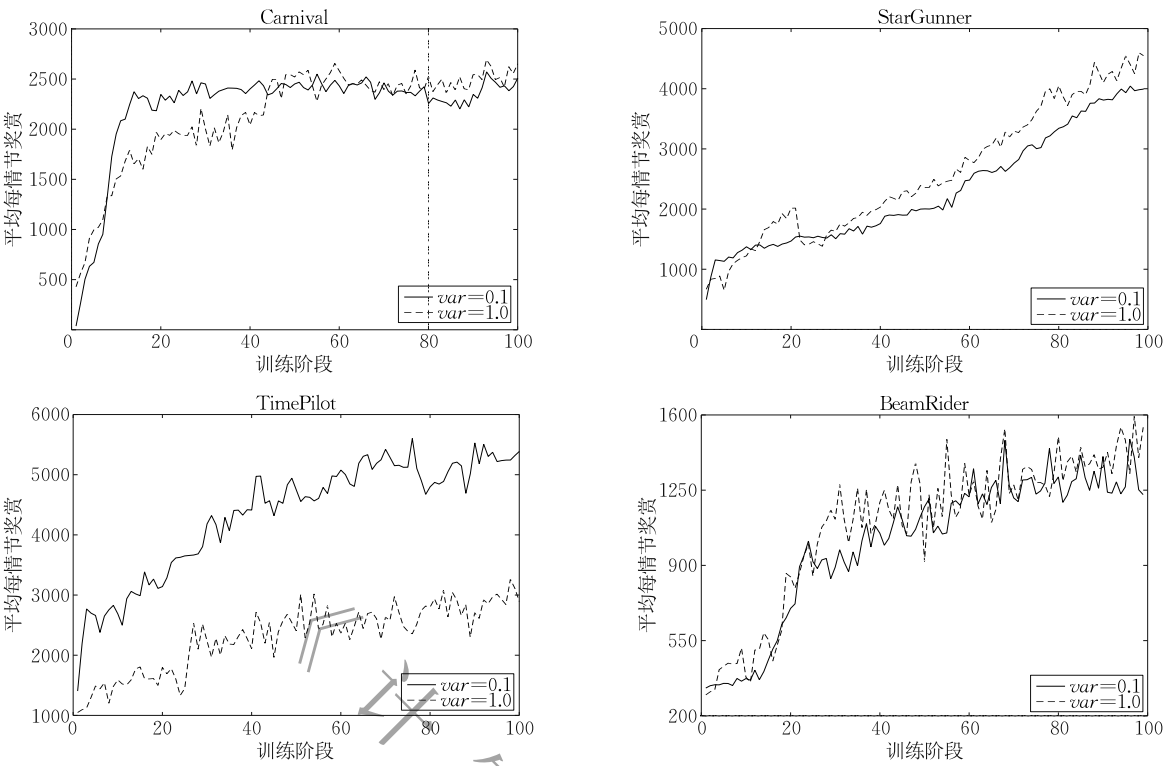


图 8 噪声分布方差对比图

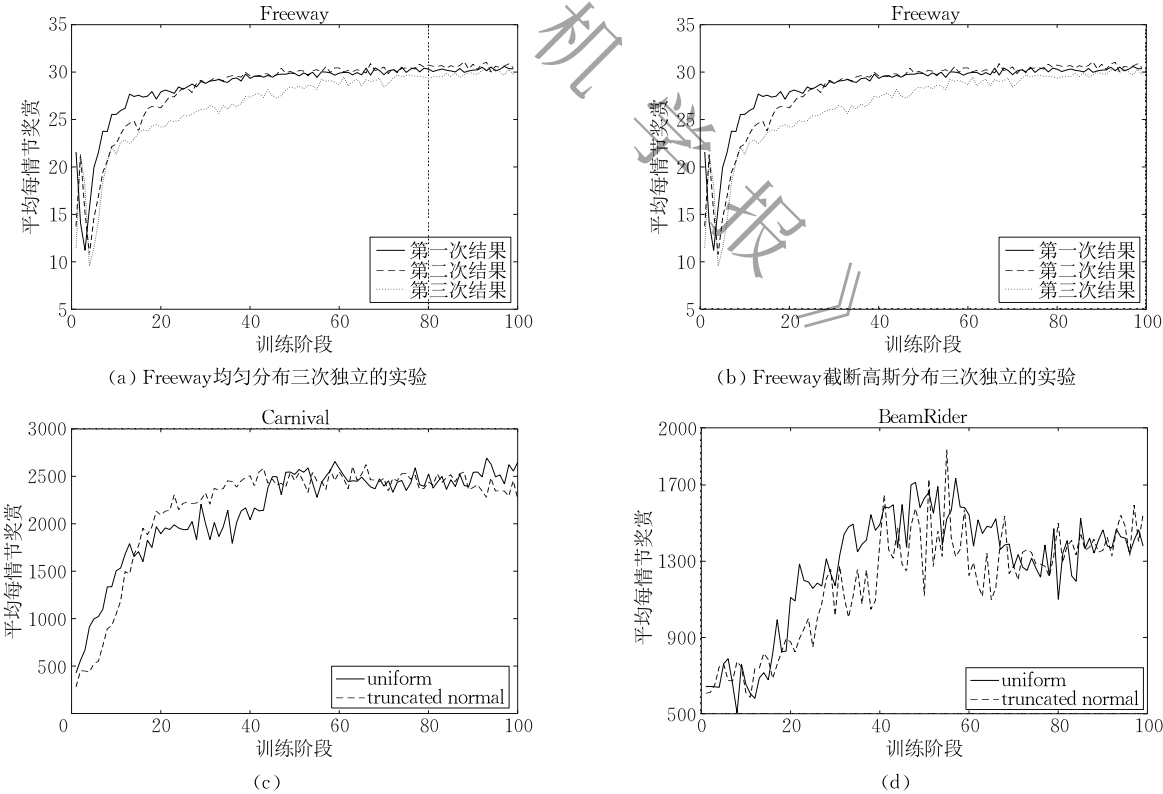


图 9 网络权重初始化对比

此外,本文还在 Carnival、BeamRider 2 个游戏上使用上述两个分布进行评估,以说明噪声网不同的权重初始化分布并不影响 Agent 的学习,实验结

果如图 9 中(c)、(d)所示。

从图 9 中可以看出,模型的训练曲线图虽然出现不同情况的波动,但总体趋势不变,导致该情况的

原因有两种:一方面噪音虽然引起了网络抖动,但是控制噪音的规模变量通过梯度下降方法进行了自适应调整,使网络的抖动在合理的范围内. 另一方面,加入的噪音一定程度上并不影响最优动作的选择,最优动作所对应的 Q 值相较于次优动作对应的 Q 值的相对性并没有发生改变. 故通过方差驱动探索并不会出现随机性和偶然性.

最优策略反映了 Agent 最终学到的效果,在最优策略指导下,Agent 在环境中可以获得最大累计奖赏. 在本文的 4 个模型中,最优策略表现为网络的参数. 当训练完毕时,Agent 得到可复用的策略模型. 本文通过测试最优策略的质量评估模型学习效果. 测试中采用 100 个情节进行评估. 为测试最优策略的有效性,选取不同的随机初始点:在情节开始时 Agent 与环境随机进行 0~30 次无动作操作. 虽然 Agent 不做动作,但是环境却得到了变化,例如敌人会在 Agent 不做动作期间逼近. 每个情节 Agent 最多与环境进行 10 000 步交互,策略的最终得分为 100 个情节得分的平均值. 本文在表 2 中的游戏环境做了测试,结果如表 4 所示.

表 4 模型的测试结果

环境	模型	平均奖赏	标准差	最高分
AirRaid	DQN	34. 70	20. 75	429. 25
	EN-DQN	508. 60	33. 61	540. 92
	LSTM-DQN	900. 52	50. 58	972. 91
	EN-DRQN	1267. 20	196. 54	1409. 56
BeamRider	DQN	553. 98	37. 49	573. 73
	EN-DQN	540. 60	40. 12	620. 73
	LSTM-DQN	1326. 17	96. 83	1487. 68
	EN-DRQN	1794. 10	43. 74	1877. 46
Centipede	DQN	1844. 98	26. 52	1879. 05
	EN-DQN	2146. 41	151. 60	3153. 72
	LSTM-DQN	2291. 97	425. 34	2303. 78
	EN-DRQN	3515. 71	542. 81	4538. 30
Freeway	DQN	19. 87	0. 44	20. 10
	EN-DQN	21. 07	0. 21	21. 38
	LSTM-DQN	25. 82	0. 22	25. 89
	EN-DRQN	30. 34	0. 38	30. 61
StarGunner	DQN	1536. 40	48. 48	1595. 88
	EN-DQN	1449. 83	20. 09	1479. 71
	LSTM-DQN	1413. 20	17. 99	1423. 57
	EN-DRQN	4478. 65	134. 05	4598. 65
Carnival	DQN	1771. 25	218. 91	1937. 93
	EN-DQN	2176. 70	64. 72	2206. 38
	LSTM-DQN	1499. 95	288. 02	1699. 13
	EN-DRQN	2552. 89	183. 70	2651. 46
TimePilot	DQN	2152. 84	273. 89	2306. 31
	EN-DQN	795. 73	467. 50	832. 79
	LSTM-DQN	889. 93	121. 34	949. 52
	EN-DRQN	3008. 24	350. 48	3263. 33
MontezumaRevenge	DQN	0. 09	0. 06	0. 15
	EN-DQN	-0. 86	3. 56	0. 33
	LSTM-DQN	0. 68	1. 16	1. 33
	EN-DRQN	-0. 52	0. 86	-0. 2655

从表 4 平均得分可以看出,与其余 3 种模型相比,除了 MontezumaRevenge 外,每个训练完毕的 EN-DRQN 在指导 Agent 玩游戏时均获得了一定程度上的提高. 从最大值可以看出,训练完成的 EN-DRQN 在最优表现中也明显优于其它模型,尤其在 Centipede 和 StarGunner 游戏中表现更为明显. 这说明完成训练的模型不仅在游戏得分上获得了更优的效果,而且能在多次游戏中保持该效果的优异性.

5 结束语

深度 Q 网络由于自身结构的原因无法记忆多个时间步内有效依赖关系,依靠循环神经网络可以有效解决这个问题. 然而,传统循环神经网络中的门机制严重依赖于上一时间步的输出信息,增加网络的计算成本,同时传统算法依靠 ϵ -greedy 探索,难以寻找到合适的 ϵ 达到探索与利用的平衡. 本文提出一种在网络中加入噪音驱动深度循环 Q 网络探索的方法,采用改进双层的简单循环单元以加强循环网络的记忆能力,同时加快网络的训练速度. 本文选取 8 个带有战略性以及稀疏奖赏性的游戏验证模型的有效性,同时对这 8 个游戏进行测试,结果表明本文提出的模型平均在每个情节中获得的奖赏最高,说明了模型的优异性.

然而,使用噪音网络探索会导致训练表现有较大方差,下一步研究重点是考虑如何在降低方差的情况下驱动网络更加有效地探索. Anschel 等人^[24]提出的 Average DQN 方法在降低方差的前提下提高了训练的表现. 类似的,为进一步提高 Agent 性能,对噪音进行处理,以达到减小方差,增加稳定性的目的.

参 考 文 献

[1] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with deep reinforcement learning//Proceedings of the Neural Information Processing Systems, Deep Learning Workshop. Lake Tahoe, USA, 2013: 10-18

[2] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529-533

[3] van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 2094-2100

- [4] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay//Proceedings of the International Conference on Learning Representations. San Juan, Puerto Rico, 2016: 53-73
- [5] Fortunato M, Azar M G, Piot B. Noisy networks for exploration //Proceedings of the International Conference on Learning Representations. Vancouver, Canada, 2018: 532-552
- [6] Hausknecht M, Stone P. Deep recurrent Q-learning for partially observable MDPs//Proceedings of the Association for the Advance of Artificial Intelligence Fall Symposium. Ferrara, Italian, 2015: 29-37
- [7] Lample G, Chaplot D S. Playing FPS games with deep reinforcement learning//Proceedings of the AAAI Conference on Artificial Intelligence. California, USA, 2017: 2140-2146
- [8] Zhai Jian-Wei, Liu Quan, Zhang Zong-Zhang, et al. A deep recurrent Q-network based on visual attention mechanism. Chinese Journal of Computers, 2017, 37(6): 1353-1366 (in Chinese)
(翟建伟, 刘全, 章宗长等. 一种基于视觉注意力机制的深度循环 Q 网络模型. 计算机学报, 2017, 37(6): 1353-1366)
- [9] Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 1998
- [10] Jaakkola T, Jordan M I, Singh S P. On the convergence of stochastic iterative dynamic programming algorithms. Neural Computation, 1994, 6(6): 1185-1201
- [11] Tsitsiklis J N. Asynchronous stochastic approximation and Q-learning. Machine Learning, 1994, 16(3): 185-202
- [12] Tsitsiklis J N, Van R B. An analysis of temporal-difference learning with function approximation. IEEE Transactions on Automatic Control, 1997, 42(5): 674-690
- [13] Fu Qi-Ming, Liu Quan, Wang Hui, et al. A novel off policy $Q(\lambda)$ algorithm based on linear function approximation. Chinese Journal of Computers, 2014, 37(3): 677-686 (in Chinese)
(傅启明, 刘全, 王辉等. 一种基于线性函数逼近的离策略 $Q(\lambda)$ 算法. 计算机学报, 2014, 37(3): 677-686)
- [14] Liu Quan, Zhai Jian-Wei, Zhang Zong-Zhang. An overview of deep reinforcement learning. Chinese Journal of Computers, 2018, 41(1): 1-27 (in Chinese)
(刘全, 翟建伟, 章宗长. 深度强化学习综述. 计算机学报, 2018, 41(1): 1-27)
- [15] Hochreiter S. Long short-term memory. Neural Computation, 1997, 9(8): 1735-1780
- [16] Lei T, Zhang Y, Artzi Y. Training RNNs as fast as CNNs//Proceedings of the Neural and Evolutionary Computing. California, USA, 2017: 1709-1723
- [17] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural network//Proceedings of the 26th Annual Conference on Neural Information Processing Systems. Nevada, USA, 2012: 1097-1105
- [18] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324
- [19] Donahue J, Anne H L, Guadarrama S, et al. Long-term recurrent convolutional networks for visual recognition and description//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, USA, 2015: 2625-2634
- [20] Gao Yang, Chen Shi-Fu, Lu Xin. Research on reinforcement learning technology: A review. Acta Automatica Sinica, 2004, 30(1): 86-100 (in Chinese)
(高阳, 陈世福, 陆鑫. 强化学习研究综述. 自动化学报, 2004, 30(1): 86-100)
- [21] Osband I, Russo D J, Wen Z, et al. Deep exploration via randomized value functions//Proceedings of the Neural Information Processing Systems. California, USA, 2017: 1703-1763
- [22] Lakshminarayanan A S, Sharma S, Ravindran B. Dynamic frame skip deep Q network//Proceedings of the International Joint Conference on Artificial Intelligence Workshop on Deep Reinforcement Learning. California, USA, 2016: 365-371
- [23] Busoniu L, Babuska R, De Schutter B, et al. Reinforcement Learning and Dynamic Programming Using Function Approximators. Oxford, UK: CRC press, 2010
- [24] Anschel O, Baram N, Shimkin N. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning//Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia, 2017: 176-185



LIU Quan, Ph. D., professor, Ph.D. supervisor. His main research interests include intelligence information processing, automated reasoning and machine learning.

YAN Yan, M. S. candidate. His main research interests include reinforcement learning, deep learning and deep reinforcement learning.

ZHU Fei, Ph. D., associate professor. His main research interests include reinforcement learning, text mining, and pattern recognition.

WU Wen, M. S. candidate. His main research interests include reinforcement learning, deep learning and deep reinforcement learning

ZHANG Lin-Lin, M. S. candidate. Her main research interests include reinforcement learning and deep reinforcement learning.

Background

Deep reinforcement learning combined with deep learning and reinforcement learning was a great success on the Atari 2600. The deep Q network using convolutional neural networks and recurrent neural networks face enormous challenges in remembering historical information and exploring problems. The deep Q network, which combines the deep neural network and the reinforcement learning method, has achieved great success in the Atari 2600 game platform. Compared with the deep Q network, the deep recurrent Q network has the ability of memorizing historical information and shows better performance in some strategic tasks and games with delayed rewards. However, on the one hand, traditional dithering policy cannot make a reasonable decision; on the other hand, in some complex game environments, because of the complex structure and activation function, the deep recurrent Q network requires a lot of training time, which increased the difficulty of the task. It is worth mentioning that the eight games are all

strategic and the rewards given by the environment are delayed. We compared the model by evaluating the average score of each game. Compared with the dithering policy in the action space, the experimental results show that EN-DRQN has achieved better performance in some strategy games and games with delayed rewards.

This paper is partially supported by the National Natural Science Foundation of China (61772355, 61472262), the Major Natural Science Research Project of Jiangsu Higher Education Institutions (18KJA520011, 17KJA520004), the Funded Project of Key Laboratory of Symbolic Computation and Knowledge Engineering of Jilin University (93K172014K04, 93K172017K18), the Suzhou Application Foundation Research Project Industrial Section (SYG201422, SYG201804), and the Jiangsu Provincial Key Laboratory (Suzhou University) Project (KJS1524).