Computer Science and Application 计算机科学与应用, 2023, 13(4), 779-788
Published Online April 2023 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2023.134077

Hans汉斯

深度强化学习模型轻量化算法研究

安天一，李　宁，王　超

北京信息科技大学计算机学院，北京

收稿日期：2023年3月18日；录用日期：2023年4月17日；发布日期：2023年4月23日

摘　要

针对深度强化学习网络难以部署到资源受限终端设备的问题，本文提出一种深度神经网络优化压缩算法。该算法引入倒残差模块作为主干网络，实现网络的轻量化；采用基于响应的知识蒸馏，以动作策略为蒸馏目标，弥补网络轻量化造成的精度损失；采用基于特征的知识蒸馏，对网络中间层的特征向量进行蒸馏，进一步提升网络精度。实验结果表明，轻量化后的网络参数量为19.79M，参数量为原网络的59.8%，性能提升约12.1%，且在网络轻量化的同时，提升了模型表现，验证了所提算法的有效性。

关键词

深度强化学习，轻量化设计，知识蒸馏

Research on Lightweight Algorithms for Deep Reinforcement Learning

Tianyi An, Ning Li, Chao Wang

School of Computing, Beijing Information Science and Technology University, Beijing

Received: Mar. 18th, 2023; accepted: Apr. 17th, 2023; published: Apr. 23rd, 2023

Abstract

In response to the difficulty of deploying deep reinforcement learning networks on resource- constrained terminal devices, a deep neural network optimization compression algorithm is proposed in this paper. This algorithm introduces an inverse residual module as the backbone network to achieve the lightweight of network; adopts response-based knowledge distillation, with action strategy as the distillation target, to make up for the accuracy loss caused by the lightweight of network; adopts feature-based knowledge distillation to distill the feature vectors in the middle layer of the network, further improving network accuracy. Experimental results show that the parameter size of the lightweight network is 19.79M, the parameter size is 59.8% of the original network, the per-

文章引用: 安天一, 李宁, 王超. 深度强化学习模型轻量化算法研究[J]. 计算机科学与应用, 2023, 13(4): 779-788.
DOI: 10.12677/csa.2023.134077

formance is improved by about 12.1%, and the model performance is improved while the network is lightweight, verifying the effectiveness of the proposed algorithm.

Keywords

Deep Reinforcement Learning, Lightweight Design, Knowledge Distillation

1. 引言

随着深度学习(Deep Learning, DL)技术的不断发展，其凭借深度神经网络强大的特征表达能力，为学术界和工业界解决了许多的难题并取得了众多令人瞩目的研究成果。强化学习(Reinforcement Learning, RL)作为解决序列决策的重要方法，赋予智能体自监督学习能力，能够自主与环境进行交互，通过获得的奖励不断修正策略。而将深度学习引入强化学习则衍生出深度强化学习(Deep Reinforcement Learning, DRL)。近年来，以 DRL 为代表的人工智能技术在各领域取得了重大突破，已广泛应用于优化调度[1]、机器人控制[2]、智能驾驶[3]、机器视觉[4]、游戏[5] [6]和军事作战[7]等领域，被认为是迈向通用人工智能的重要途径[8]。

DRL 在应用过程中，深度模型需要通过与环境高频次大量的交互来获取奖励，以更新智能体的网络参数，最终获得较高水平的表现。由此导致模型的训练开销巨大，且深度模型中含有庞大的参数，这为其在有限资源设备(如嵌入式设备、手机汽车等移动终端)的部署移植带来了困难与挑战。因此，如何辅助模型训练以提升模型的训练效果并且实现深度模型的轻量化，以及减少冗余参数对深度模型压缩研究具有重要理论意义。

本文针对 DRL 因模型存储和计算量大而难移植至嵌入式或移动设备的问题，以雅达利 100k (Atari 100k) [9]数据集为基准，融合知识蒸馏、结构设计的压缩方法，结合蒙特卡洛树搜索算法[10]，基于当前最优算法 EfficientZero [11]验证算法效果，提出一种针对强化学习任务设计的深度神经网络优化压缩算法，以进一步压缩深度神经网络。

具体地，考虑到现有深度强化学习模型复杂度高、计算量大、推理速度慢、内存消耗巨大、难以部署在有限资源的终端设备上，因此本文设计了一种基于 MobileNetV2 [12]的轻量化网络，以减少模型参数量，提高模型推理速度；同时，我们发现，这样设计的小模型在 RL 上精度表现较差，本文进一步提出了一种针对强化学习任务设计的知识蒸馏方法。本方法融合了基于输出响应的知识蒸馏与基于特征的知识蒸馏，在常规的模型响应输出的蒸馏基础上，我们进一步提出对强化学习训练过程中关键中间特征进行蒸馏，使得学生模型拥有超越教师模型精度的表现。

本文的主要工作可总结为以下几点：

1) 提出了一种针对强化学习任务设计的深度神经网络优化压缩算法。基于 MobileNetV2 网络对原始模型进行轻量化处理，轻量化后的模型参数量为 19.79M，相较于原始模型，参数量减少了 40.2%。

2) 采用知识蒸馏的思路设计学生网络和教师网络两个神经网络做知识迁移，使得学生网络从教师网络中学习“知识”，再通过对学生网络进行结构设计来缩小学生网络的体量大小。引入知识蒸馏后，学生网络相较于教师网络表现提升了 12.1%。

3) 在 Atari 100k 上对本文算法进行了多个基准测试，并与传统方法进行了实验比较与分析，结果表明，本文方法在多场游戏中的表现优于当前主流强化学习算法，并在显著降低的模型计算量和参数量下，与当前最强基线水平 EfficientZero 相当。

2. 相关工作

近年来，表现优异的深度强化学习模型多数是建立在海量网络参数和庞大计算量的基础上，大部分研究专注于提升模型的采样效率，而很少关心对深度强化学习模型的轻量化方法，因此，如何在保证性能的前提下，降低模型对硬件资源的消耗，使其能够高效部署于资源受限的终端设备上成为本研究关注的焦点。

2.1. 深度强化学习

深度强化学习融合了深度学习和强化学习的优势，能够帮助智能体在复杂的高维状态空间中进行感知和决策[13] [14] [15]。与传统的强化学习方法相比，深度强化学习可以通过训练深度神经网络来学习更复杂的策略和价值函数，从而实现更高效和准确的决策。

2016 年发表于《Nature》上的围棋 AI：AlphaGo [16]创造性地将深度强化学习和蒙特卡罗树搜索相结合，利用价值网络(value network)评估棋局以减少搜索深度，利用策略网络(Policy Network)减少搜索宽度，从而大大提高了搜索效率和胜率估计的精度。在 AlphaGo 的基础上，AlphaGo Zero [17]引入基于残差模块构成的深度神经网络，通过原始状态信息提取相关表示特征，使用神经网络估值函数替换快速走子过程，大幅减少了算法的训练学习和执行走子所需要的时间。MuZero [18]通过将基于树的搜索与学习模型相结合，在不了解游戏规则或环境动态的情况下实现超人类的表现。EfficientZero 通过时序上的环境一致性构建对比学习损失函数，在 Atari 100k 基准测试中实现了 194.3%的平均人类表现和 109.0%的中位数表现，首次在雅达利(Atari)游戏数据上超过同等游戏时长的人类平均水平。

但是，以上模型都存在网络参数庞大，浮点型计算次数过高的问题，网络参数庞大意味着模型需求海量的内存存储空间，浮点型计算次数过高带来训练成本和计算时间的几何式增长，这极大地限制了在资源受限设备上的部署。因此，本文提出一种基于 MobileNetV2 的轻量化网络，以减少模型参数量，提高模型推理速度。

2.2. 深度神经网络压缩

近年来，对深度神经网络计算需求的日益增大，加速了对深度神经网络压缩算法的研究，于是知识蒸馏、轻量化结构设计等压缩算法开始相继出现。

知识蒸馏是一种教师－学生(Teacher-Student)训练结构，通常使用预训练的教师模型提供知识，学生模型通过蒸馏训练来获取知识，以轻微的性能损失为代价，将复杂的深层网络模型向浅层的小型网络模型进行知识迁移，其能够复用现有的模型资源，极大节省了深度神经网络的训练和应用成本。Hinton 等人[19]首次提出知识蒸馏的概念，主要利用神经网络对样本数据的预测中包含的潜在信息，引入与教师网络相关的软目标来促进学生网络训练，达到知识迁移的目的。Romero 等人[20]设计了浅层的 FitNets 网络用使用回归模块来配准部分学生网络和教师网络的输出特征，并对输出特征进行相应处理，通过教师网络中间层的暗示(hints)来引导学生模型向教师模型学习。

轻量化结构设计通过调整神经网络架构，使得其只需要以较少的参数就能获得同等量级网络的精度，并达到压缩神经网络的目的。轻量化结构设计主要针对卷积网络设计一种更高效、计算复杂度更低的方法，在不损失网络精度的情况下，减少每秒浮点运算频率，降低模型参数量。Iandola 等人[21]提出一种

轻型网络 SqueezeNet，在与 AlexNet 精度持平的情况下，参数量只有 AlexNet 的 50%，MobileNetV1 [22]提出深度可分离卷积代替原来的传统卷积进行计算，将滤波器的参数大大降低，MobileNetV2 在 MobileNetV1 的基础上设计了 Inverted Residuals 模块，以减少推理时间，MobileNetV3 [23]通过神经结构搜索获得子网络，并在 MobileNetV2 的 block 中添加了 SENet [24]，大大增强了网络特征提取能力，获得了更高的模型精度。ShuffleNetV1 [25]使用组卷积(Group Convolution)降低模型参数大小，使用通道混排(Channel Shuffle)增强各特征图的连接，ShuffleNetV2 [26]对 ShuffleNetV1 进行了改进，提升了模型精度和运行速度。

神经网络压缩技术在实际应用中经常结合对抗生成网络[27]、神经架构搜索[28]、图卷积[29]、集成学习[30]等主流技术，以求在低算力情况下获得更好的性能，但很少研究针对强化学习的任务提出网络压缩算法。因此，本文提出了一种针对强化学习任务的神经网络压缩算法，该算法基于轻量化神经网络设计，融合了基于输出响应和基于特征的知识蒸馏方法。

3. 深度强化学习算法

本文提出的通用强化学习模型轻量化方法适用于多种强化学习算法，本文以当前最优的 EfficientZero 算法为基准验证算法效果。

3.1. EfficientZero

EfficientZero 是一种基于蒙特卡洛树搜索(MCTS)算法的策略学习方法，它致力于通过提高算法采样效率来解决强化学习算法在现实世界场景中环境模型难以建立、环境数据量受限等问题。

EfficientZero 的网络架构由三部分组成，即表征网络 H (Representation Network)、动态网络 G (Dynamics Network)和预测网络(Prediction Network)，其中预测网络部分由结构相似的三部分组成，分别对奖励、价值和策略进行预测。网络执行一步的训练流程如公式所示：

$$\begin{aligned} s_t &= H\left(o_t\right) \\ s_{t+1} &= H\left(o_{t+1}\right) \\ \hat{s}_{t+1}\left(t+1\right) &= G\left(s_t, a_t\right) \\ v_t &= V\left(s_t\right) \\ p_t &= P\left(s_t\right) \\ r_t, h_{t+1} &= R\left(\hat{s}_{t+1}, h_t\right) = R\left(G\left(s_t, a_t\right), h_t\right) \end{aligned} \tag{1}$$

式中，V 为价值预测网络，P 为策略预测网络，R 为奖励预测网络，表征网络 H 对当前状态的观测结果 o_t (通常为当前状态的表征向量或图片等信息)进行特征编码，得到状态 s_t；动态网络 G 接收表征网络输出的状态 s_t，基于一个候选动作 a_t，将状态 s_t 映射到下一个状态 $\hat{s}_{t+1}$；预测网络中价值预测网络 V 和策略预测网络 P 均将状态 s_t 作为输入，预测价值 V_t 以及策略 p_t。奖励预测网络以动态网络输出的下一状态 s_{t+1} 以及当前状态的隐藏状态 h_t (神经网络循环过程中产生的隐藏状态)作为输入，预测奖励值 r_t 以及下一隐状态 h_{t+1}。

网络总体结构如图 1 所示，通过 MCTS 启发式地对环境进行探索和利用，得到每个动作的观测结果和对应的评价用于训练强化学习模型。在 EfficientZero 模型中，对每个观测结果的输入，首先使用表征网络对观测结果进行特征编码，再将特征编码及候选动作输入到动态网络得到下一时刻的隐状态及网络对当前状态－动作对的预测奖励，同时将特征编码输入到预测网络中，得到基于当前状态的策略预测及价值预测，最后，重复将动态网络输出的隐状态输入到自身和预测网络中。网络训练最终主要有三个目

标，第一个是最小化预测策略和 MCTS 得到的策略之间的误差；第二个是最小化预测价值与 MCTS 得到的价值之间的误差；第三个是最小化预测奖励和观察到的奖励之间的误差。

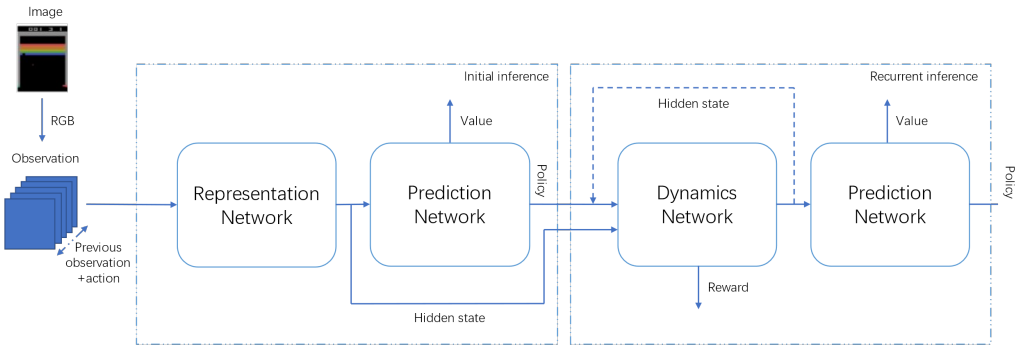

Figure 1. Network architecture overall structure diagram
图 1. 网络结构总体结构图

3.2. 蒙特卡洛树搜索

蒙特卡罗树搜索是一种用于决策过程的启发式搜索算法，它在决策空间中抽取随机样本，根据结果构建搜索树，最后在给定域中寻找最优决策[31]。为了找到高质量的决策，扩展过程必须在探索与利用之间平衡，即在扩展具有更多访问次数但表现较差的节点与访问次数较少但表现更好的节点之间进行平衡。MCTS 采用 UCT [32] [33]方程对每个节点进行打分，并将得分最高的节点作为下一次探索的节点，从而逐层探索得到叶子节点，即强化学习下一步需要进行的动作。在第 k 层蒙特卡洛树的扩展步骤中，UCT 将按照以下方式选择一个节点：

$$a^k = \arg\max_a \left\{ Q(s,a) + P(s,a) \frac{\sqrt{\sum_b N(s,b)}}{1+N(s,a)} \left(c_1 + \log\left(\frac{\sum_b N(s,a) + c_2 + 1}{c_2} \right) \right) \right\} \tag{2}$$

其中，Q(s, a)是当前 Q 值的估计，P(s, a)是选择此动作的当前神经网络策略，帮助 MCTS 优先探索树中前景较好的部分。在训练时间，P(s, a)通常会添加噪声以允许探索。N(s, a)表示在树搜索中访问此状态－动作对的次数，N(s, b)表示 a 的兄弟节点的访问次数。在扩展节点一定次数后，MCTS 将返回根节点下每个动作的访问次数。

MCTS 算法需要一个环境模型、一个先验策略函数和一个值函数，即 2.1 中提到的动态网络 G，策略预测网络 P 以及价值预测网络 V，MCTS 使用这些函数扩展新的子节点。预测的策略 $p_t = acts$ 用作节点上动作的先验搜索，它帮助 MCTS 在扩展节点时导向高收益的动作。价值函数 V 用于测量节点 s_t 的预期回报，为叶节点提供长期的评估。最终，MCTS 输出一个基于根节点的动作概率分布 π_t。

3.3. 损失函数

网络损失函数如公式所示：

$$\begin{aligned} &L_{similarity}\left(s_{t+1}, \hat{s}_{t+1}\right) = L_2\left(sg\left(P_1\left(s_{t+1}\right)\right), P_2\left(P_1\left(\hat{s}_{t+1}\right)\right)\right) \\ &L_t = L\left(u_t, r_t\right) + \lambda_1 L\left(\pi_t, p_t\right) + \lambda_2 L\left(z_t, v_t\right) + \lambda_3 L_{similarity}\left(s_{t+1}, \hat{s}_{t+1}\right) \\ &L = \frac{1}{l_{unroll}} \sum_{i=0}^{l_{unroll}-1} L_{t+i} \end{aligned} \tag{3}$$

其中，λ_1是策略函数损失系数，λ_2是价值函数损失系数，λ_3是自监督一致损失函数系数，u_t是实际得到的奖励，z_t是采样一次的总价值，π_t是实际采取的动作，l_{unroll}是序列数据的展开步数。式 n 中，第一项 L 是展开的 l_{unroll} 步的奖励预测网络总损失，对网络预测的奖励 r_t 进行监督，第二项是策略预测网络损失函数，对网络预测的策略 p_t 进行监督，第三项是价值预测网络损失函数，对网络预测的价值v_t进行监督，第四项 $L_{similarity}$ 是自监督一致性损失函数(Self-supervised Consistency Loss)，通过时序上的环境一致性构建对比学习损失函数，加强动态网络的训练信号，对动态网络进行自我监督，确保预测状态 $\hat{s}_{t+1}$ 的一致性。L_1是交叉熵损失函数，L_2是负余弦相似度损失函数，P_1和P_2分别为 3 层和 2 层的多层感知机(MLP)，$sg(P_1)$表示停止梯度回传。

4. 强化学习模型压缩

本文提出了一种针对强化学习任务设计的强化学习模型压缩方法，该方法在对原模型进行轻量化处理的基础上，融合了基于输出响应的知识蒸馏与基于特征的知识蒸馏，使得学生模型在低复杂度的情况下也能拥有超越教师模型精度的表现。

4.1. 基于强化学习的轻量化模型设计

EfficientZero 算法的表征网络及动态网络主要由卷积神经网络和残差神经网络构成，其中残差网络模块层次较深且参数较多，模型参数量较大，会占用过多存储空间，增加了计算处理难度。因此，为降低模型复杂度，我们将表征网络及动态网络的残差模块(Residual Block)替换为 MobileNetV2 中的倒残差模块(Inverted Residual Block)。

MobileNetV2 是一种轻量级卷积神经网络，主要用于移动端设计和嵌入式视觉应用，具备参数量少、时延低等特性。MobileNetV2 网络结构中最强力的创新在于使用了倒残差结构(Inverted Residuals)，倒残差结构不同于标准残差结构，其结构呈“梭”型，使用深度可分离卷积(Depthwise Separable Convolution)，将标准卷积拆分为逐通道卷积(Depthwise Convolution)和逐点卷积(Pointwise Convolution)，这种特殊结构让 MobileNetV2 在保证模型精度的同时，大幅度地减少了神经网络参数和计算量。表征网络及动态网络中有多个残差模块，并且每个残差网络模块计算复杂度较高，因此，将其中的残差模块使用倒残差模块进行替换，模型的计算量和参数数量将大幅减少，推理速度也将得到提升。

4.2. 基于强化学习的知识蒸馏算法

模型在经过轻量化之后，参数量和计算量在显著减少的同时，也伴随着一定的精度下降。为弥补由于模型轻量化导致的性能损失，本文融合基于响应的知识蒸馏以及基于特征的知识蒸馏方法与轻量化后的网络模型相结合，对模型特定层的特征及预测结果进行知识蒸馏，以实现模型精度的提升。

4.2.1. 基于响应的知识蒸馏算法

神经网络的响应是网络输出层向量，基于响应的知识蒸馏主要思想是学生网络直接模仿教师网络的输出，即通过平滑 SoftMax 层的输出，最大化学生网络和教师网络之间的输出相似性，使得学生网络获得泛化性更强的预测能力。

预测网络作为网络的输出终端，其作用是预测模型在对应状态下的动作策略和状态值函数，它可以指导蒙特卡洛树搜索选择最优的动作，并可以反馈给神经网络进行更新。动作策略作为一种高层次和抽象化的知识表示，可以捕捉到环境中重要和有价值的信息，并且它只需要一个向量来表示每个动作出现在当前状态下的概率，因此它也是一种更易于传递和比较的知识形式。由此，本文将动作策略作为基于响应的知识蒸馏的目标标签，使其可以直接指导学生模型如何在给定状态下做出最优决策，增强学生模

型与教师模型的相似度，提升学生模型在实际任务中的表现。

综上所述，本文采用基于响应的知识蒸馏，原始模型作为教师网络，轻量化后的模型作为学生网络，通过学习教师模型的动作策略，让轻量化后的学生模型掌握教师模型的推理方式，从而达到提升模型精度的目的。

基于响应的知识蒸馏损失函数如式(4)所示：

$$L = L_{\text{response}} + L_{\text{task}} \tag{4}$$

其中，L_{response} 为学生网络基于响应的知识蒸馏的损失函数，L_{task} 为教师网络的原任务的损失函数。

4.2.2. 基于特征的知识蒸馏算法

神经网络中间特征是深度神经网络的中间层部件所提取出的高维特征，基于特征的知识蒸馏主要思想是利用教师网络提取的更具表征能力的高维特征指导学生网络进行训练。由于教师网络参数量庞大，网络结构复杂，只依赖预测网络输出的动作策略作为软目标无法有效地将知识迁移至学生网络，同时学生模型仅通过输出差异也很难有效衡量样本的异常程度。因此，本文不仅使用基于响应的知识蒸馏算法，还采用了基于中间层特征的知识蒸馏算法作为对算法的改进。

本文选用动态网络输出的隐藏奖励(Reward Hidden)和隐状态(hidden_state)两个特征向量进行蒸馏。EfficientZero 的动态网络为解决在预测价值时的状态混叠(State Aliasing)问题时，引入了 LSTM 网络架构来预测多步状态变化下的回报和，即价值前缀(prefixvalue)，而 LSTM 网络的输入除了被编码的当前状态 s，实际执行动作 a，还有 reward_hidden 即 LSTM 网络的隐状态与细胞状态，由于隐状态和细胞状态包含了 LSTM 网络对序列数据的重要信息，并且相对 reward 预测的标量具有较高的维度，非常适合将其作为特征蒸馏的蒸馏目标，并且相较于只使用最后一个时间步上的输出作为蒸馏目标，对隐状态和细胞状态的蒸馏可以在每个时间步上进行监督，可以增加训练信号和反馈强度。隐状态 hidden_state 作为对真实环境状态的抽象，是动态网络连接预测网络的中介值，并在网络训练循环中也是动态网络的输入，使用 hidden_state 进行蒸馏可以帮助学生网络更好地理解和预测环境状态的隐含信息。

因此，本文将中间层特征的损失函数 L_{distill} 定义为教师－学生网络中间层特征之间的均方误差(MSE)，损失函数如式(5)所示，其中，$L_{\text{feature}[0]}$为隐状态特征，$L_{\text{feature}[1]}$为细胞状态特征。

$$L_{\text{feature}} = L_{\text{feature}[0]} + L_{\text{feature}[1]} + L_{\text{task}} \tag{5}$$

5. 算法实验与结果分析

5.1. 实验配置

实验使用 Atari 100k 基准进行测试，Atari 100k 最初由 SimPLe [34]方法提出，本文抽取其中的 14 个 Atari 游戏环境进行测试，每个环境中的智能体只允许执行 100k 个动作。这个约束大约相当于 2 小时的人类游戏时间。相比之下，不受限制的 Atari 智能体通常训练 5000 万步，经验增加了 500 倍。

实验环境为 I9-9700X 处理器，64 GB-RAM，RTX 3090 独立显卡，并使用 Pytorch 作为深度学习框架。

实验中将本文提出算法与多种方法进行了对比，包括 SimPLe、OTRainbow [35]、CURL [36]、DrQ [37]、SPR [38]、MuZero 等。实验为验证算法的有效性，将原始“教师模型”在所有 Atari 100k 的 26 个游戏环境中预训练，并根据所提出的通用强化学习模型轻量化方法对“教师模型”进行训练。

5.2. 实验结果

本文所提算法与当前主流算法对比结果如表 1 所示。

Table 1. Comparison of scores of mainstream reinforcement learning algorithms
表 1. 主流强化学习算法得分对比

Game	SimOLe	OT Rainbow	CURL	Drq	SPR	MuZero	EfficientZero	Ours
Alien	616.9	824.7	558.2	771.2	801.5	530.0	697.5	773.1
Assault	527.2	351.9	600.6	452.4	571.0	500.1	1427.1	**1508.7**
Asterix	1128.3	628.5	734.5	603.5	977.8	1734.0	5745.3	5578.1
Battle Zone	5184.4	4060.6	14870.0	12954.0	16651.0	7687.5	12312.5	**13250**
Breakout	16.4	9.8	4.9	16.1	17.1	48.0	353.3	**393.9**
Crazy Climber	62583.6	21327.8	12146.5	20516.5	42923.6	56937.0	64237.5	**66728.1**
Demon Attack	208.1	711.8	817.6	1113.4	545.2	3527.0	6272.8	**8091.0**
Hero	2656.6	6458.8	6279.3	3736.3	7019.2	3095.0	12256.4	**13457.0**
Kangaroo	323.1	605.4	872.5	940.6	3276.4	62.5	918.8	**1068.8**
Krull	4539.9	3277.9	4229.6	4018.1	3688.9	4890.8	7047.8	**7176.25**
Ms. Pac-Man	1480.0	941.9	1465.5	960.5	1313.2	1265.6	1287.8	1176.2
Private Eye	58.3	100.0	218.4	−13.6	124.0	56.3	100	100
Qbert	1288.8	509.3	1042.4	854.4	669.1	3952.0	13865.6	**15337.5**
UpND	3350.3	2847.6	2955.2	3180.8	28138.5	2896.9	8931.25	**64405.0**

从表 1 中可以看出，本文方法在多项游戏中得分较高，相比当前的 SoTA，性能平均高出 12.1%，并且该模型参数量为 19.79M，参数量为原网络的 59.8%，在网络轻量化的同时，提升了模型表现，验证了所提算法的有效性。

6. 结束语

针对传统深度强化学习网络的局限性，本文提出通过融合模型轻量化设计与知识蒸馏方法优化上述问题。并将提出的算法与传统算法进行对比，结果表明，本文提出的算法能够在网络轻量化的同时，提升模型的性能表现，验证了所提算法的有效性。

参考文献

[1] Mao, H., Schwarzkopf, M., Venkatakrishnan, S.B., *et al*. (2019) Learning Scheduling Algorithms for Data Processing Clusters. *Proceedings of the ACM Special Interest Group on Data Communication*, Beijing, 19-23 August 2019, 270-288. https://doi.org/10.1145/3341302.3342080

[2] Long, P., Fan, T., Liao, X., *et al*. (2018) Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning. 2018 *IEEE International Conference on Robotics and Automation* (*ICRA*), Brisbane, 21-25 May 2018, 6252-6259. https://doi.org/10.1109/ICRA.2018.8461113

[3] Li, D., Zhao, D., Zhang, Q., *et al*. (2019) Reinforcement Learning and Deep Learning Based Lateral Control for Autonomous Driving. *IEEE Computational Intelligence Magazine*, **14**, 83-98. https://doi.org/10.1109/MCI.2019.2901089

[4] Liao, X., Li, W., Xu, Q., *et al*. (2020) Iteratively-Refined Interactive 3D Medical Image Segmentation with Multi-Agent Reinforcement Learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, 13-19 June 2020, 9394-9402. https://doi.org/10.1109/CVPR42600.2020.00941

[5] Vinyals, O., Babuschkin, I., Czarnecki, W.M., *et al*. (2019) Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning. *Nature*, **575**, 350-354. https://doi.org/10.1038/s41586-019-1724-z

[6] Ye, D., Chen, G., Zhao, P., *et al*. (2020) Supervised Learning Achieves Human-Level Performance in Moba Games: A

Case Study of Honor of Kings. *IEEE Transactions on Neural Networks and Learning Systems*, **33**, 908-918. https://doi.org/10.1109/TNNLS.2020.3029475

[7] 李琛, 黄炎焱, 张永亮, 等. Actor-Critic 框架下的多智能体决策方法及其在兵棋上的应用[J]. 系统工程与电子技术, 2021, 43(3): 755-762.

[8] Wang, P. and Goertzel, B. (2007) Introduction: Aspects of Artificial General Intelligence. *Proceedings of the* 2007 *Conference on Advances in Artificial General Intelligence*: *Concepts, Architectures and Algorithms*, Washington DC, June 2007, 1-16.

[9] Bellemare, M.G., Naddaf, Y., Veness, J., *et al*. (2013) The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, **47**, 253-279. https://doi.org/10.1613/jair.3912

[10] Coulom, R. (2007) Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. *Computers and Games*: 5*th International Conference*, Turin, 29-31 May 2006, 72-83. https://doi.org/10.1007/978-3-540-75538-8_7

[11] Ye, W., Liu, S., Kurutach, T., *et al*. (2021) Mastering Atari Games with Limited Data. *Annual Conference on Neural Information Processing Systems* 2021, *NeurIPS* 2021, 6-14 December 2021, 25476-25488.

[12] Sandler, M., Howard, A., Zhu, M., *et al*. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-22 June 2018, 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[13] 赵冬斌, 邵坤, 朱圆恒, 等. 深度强化学习综述: 兼论计算机围棋的发展[J]. 控制理论与应用, 2016, 33(6): 701-717.

[14] Li, Y. (2017) Deep Reinforcement Learning: An Overview. ArXiv: 1701.07274.

[15] Arulkumaran, K., Deisenroth, M.P., Brundage, M., *et al*. (2017) A Brief Survey of Deep Reinforcement Learning. ArXiv: 1708.05866.

[16] Silver, D., Huang, A., Maddison, C.J., *et al*. (2016) Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature*, **529**, 484-489. https://doi.org/10.1038/nature16961

[17] Silver, D., Schrittwieser, J., Simonyan, K., *et al*. (2017) Mastering the Game of Go without Human Knowledge. *Nature*, **550**, 354-359. https://doi.org/10.1038/nature24270

[18] Schrittwieser, J., Antonoglou, I., Hubert, T., *et al*. (2020) Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *Nature*, **588**, 604-609. https://doi.org/10.1038/s41586-020-03051-4

[19] Hinton, G., Vinyals, O. and Dean, J. (2015) Distilling the Knowledge in a Neural Network. ArXiv: 1503.02531.

[20] Romero, A., Ballas, N., Kahou, S.E., *et al*. (2014) Fitnets: Hints for Thin Deep Nets. ArXiv: 1412.6550.

[21] Iandola, F.N., Han, S., Moskewicz, M.W., *et al*. (2016) SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and < 0.5 MB Model Size. ArXiv: 1602.07360.

[22] Howard, A.G., Zhu, M., Chen, B., *et al*. (2017) Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv: 1704.04861.

[23] Howard, A., Sandler, M., Chu, G., *et al*. (2019) Searching for MobileNetV3. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, 27 October-2 November 2019, 1314-1324. https://doi.org/10.1109/ICCV.2019.00140

[24] Hu, J., Shen, L. and Sun, G. (2018) Squeeze-and-Excitation Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745

[25] Zhang, X., Zhou, X., Lin, M., *et al*. (2018) ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 6848-6856. https://doi.org/10.1109/CVPR.2018.00716

[26] Ma, N., Zhang, X., Zheng, H.T., *et al*. (2018) ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. *Proceedings of the European Conference on Computer Vision* (*ECCV*), Munich, 8-14 September 2018, 116-131. https://doi.org/10.1007/978-3-030-01264-9_8

[27] Goodfellow, I., Pouget-Abadie, J., Mirza, M., *et al*. (2020) Generative Adversarial Networks. *Communications of the ACM*, **63**, 139-144. https://doi.org/10.1145/3422622

[28] Zoph, B. and Le, Q.V. (2016) Neural Architecture Search with Reinforcement Learning. ArXiv: 1611.01578.

[29] Kipf, T.N. (2020) Deep Learning with Graph-Structured Representations. Ph.D. Thesis, Universiteit van Amsterdam, Amsterdam, 164 p.

[30] Schapire, R.E. (1990) The Strength of Weak Learn Ability. *Machine Learning*, **5**, 197-227. https://doi.org/10.1007/BF00116037

[31] Fu, M.C. (2019) Simulation-Based Algorithms for Markov Decision Processes: Monte Carlo Tree Search from Alphago to Alphazero. *Asia-Pacific Journal of Operational Research*, **36**, Article ID: 1940009. https://doi.org/10.1142/S0217595919400098

[32] Rosin, C.D. (2011) Multi-Armed Bandits with Episode Context. *Annals of Mathematics and Artificial Intelligence*, **61**, 203-230. https://doi.org/10.1007/s10472-011-9258-6

[33] Kocsis, L. and Szepesvári, C. (2006) Bandit Based Monte-Carlo Planning, Machine Learning. *ECML* 2006: 17*th European Conference on Machine Learning*, Berlin, 18-22 September 2006, 282-293. https://doi.org/10.1007/11871842_29

[34] Kaiser, L., Babaeizadeh, M., Milos, P., *et al*. (2019) Model-Based Reinforcement Learning for Atari. ArXiv: 1903.00374.

[35] Kielak, K.P. (2020) Do Recent Advancements in Model-Based Deep Reinforcement Learning Really Improve Data Efficiency? ArXiv: 2003.10181.

[36] Laskin, M., Srinivas, A. and Abbeel, P. (2020) CURL: Contrastive Unsupervised Representations for Reinforcement Learning. *International Conference on Machine Learning PMLR*, Vienna, 13-18 July 2020, 5639-5650.

[37] Kostrikov, I., Yarats, D. and Fergus, R. (2020) Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. ArXiv: 2004.13649.

[38] Schwarzer, M., Anand, A., Goel, R., *et al*. (2020) Data-Efficient Reinforcement Learning with Self-Predictive Representations. ArXiv: 2007.05929.