开悟比赛-皮皮鸥队技术整理与分享

谢铭扬 华南理工大学软件学院

范峻铭 华南理工大学软件学院

张怡 华南理工大学生物科学与工程学院

指导老师：刘飞

一、 简介

在 2022 年 9 月-2023 年 4 月举办的腾讯第三届开悟 MOBA 多智能体强化学习大赛中，我们队伍（皮皮鸥）在初赛中有幸获得了第 9 名的成绩。这得益于实验室老师与腾讯官方的大力支持， 也得益于我们队伍本身的技术探索与积累。本文首先会简单叙述比赛的基本情况，随后从各关键模块出发，简要介绍本队伍在开悟比赛中的探索历程与心得体会。

二、 参赛概况

我们队伍共有 3 名成员，均为华南理工大学研究生，来自软件学院和生物科学与工程学院。得益于刘飞老师和腾讯官方的交流合作，我们很荣幸有机会参加此次比赛。我们队伍的 3 名成员在校研究的课题均与强化学习有关，故希望通过参加本次比赛，在实践中加强对经典强化学习算法的理解，学习大规模分布式强化学习训练中系统的框架设计，了解最新的多智能体强化学习算法，提升动手能力和实际运用能力，积累相关经验。在初赛阶段，我们平均每天投入在比赛上的时间约为 0.5~1 小时，每周进行一次讨论，总结经验并明确之后的改进方向。

三、网络设计

我们队伍神经网络的设计主要沿用论文[1]中的架构，如图 1 所示，未作过多的改动，以下描述均是参考论文[1]对比赛已有工作的描述。具体地，输入网络的特征包括图像特征、我方和敌方英雄的特征、小兵特征、塔特征和全局游戏信息特征等，以上各特征包含的具体信息在说明文档中均有描述，此处不再赘述。输出的动作组成与论文[1]中的描述一致，包括确定动作类型的 button，决定位置方向的 move 和 offset，以及释放技能的目标 target。对于输入特征的处理方式，与论文[1]中描述的设计基本一致。

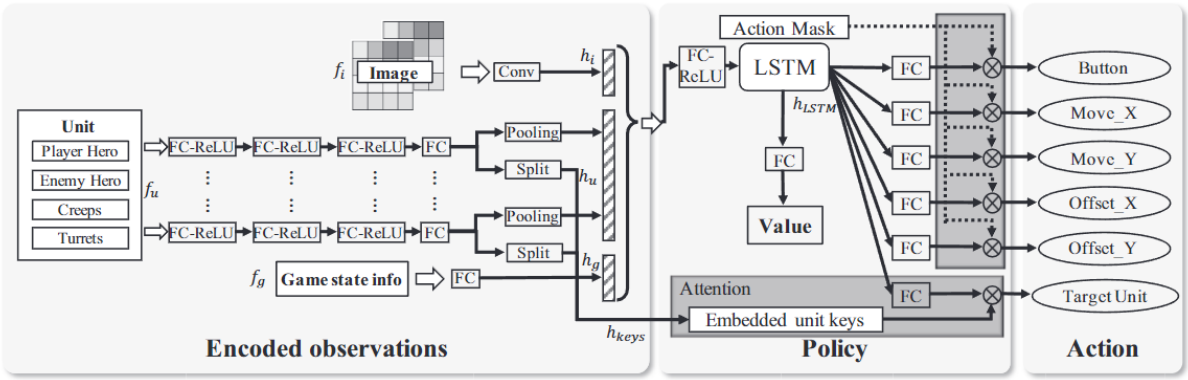

图 1 神经网络架构设计

在 RL Learner 中，采用了如上图所示的 actor-critic 网络结构，其中 state 包括了三种类型的信息：local image info（图像信息），observable unit attributes （英雄类型、英雄血量等）以及 observable game state info（游戏时间、已经破坏的塔的数量等）。特别的，编码后的 observable unit attributes 将被分为两个部分 :the representation of the unit，the attention keys of our target。此外，为了处理 unit 的数量不确定问题，相同类型的 units 将通过 max-pooling 映射到一个固定长度的特征向量中。

对于攻击目标的选择，文章采取了注意力机制进行预测，将 LSTM 的输出再经过一层 FC 作为 query，所有 unit 的编码（即上文描述的 the attention keys of our target）作为 keys，具体公式如下：

$$p(t \mid a) = Softmax\left(FC(h_{LSTM}) \cdot h_{\text{keys}}^{T}\right)$$

其中$p(t \mid a)$的维度为 unit 的数量。

四、奖励体系

奖励的设计和初始默认权重值由表 1 给出。在初赛中，我们未对奖励的计算方法进行改动，但会根据智能体在实战中的表现动态地修改奖励的权重值，以期通过奖励的修改引导智能体修正、改进其行为。奖励的计算方式，沿用零和奖励的设计方案：以当前决策帧和上一决策帧的相关数值差作为智能体的奖励，两个智能体的同类奖励项相减作为最终奖励，最终多种奖励项加权求和作为最终的奖励返回。

在训练初期，我们增大了击杀的奖励，增大死亡的惩罚，增加金币和经验的奖励。修改后平均经济有提升，击败对方英雄数目提高显著，英雄较快地学会了打野和击杀小兵。在训练的中后期，我们逐步调小了 money 和 exp 等稠密奖励的权重，调大了稀疏奖励的权重，引导智能体调整行为策略，更加关注长期和与胜负密切相关的奖励，智能体的性能有进一步的提升。

总的来说，对于奖励权重值的调整，我们均参照默认值的数量级按一定比例增大或减小。对于奖励的修改对模型表现的影响，我们未做相关的对比试验，仅从经验和直觉上作出相应的调整。在训练的后期，我们发现奖励的调整对模型性能的提高作用不明显，便沿用默认的奖励权

重值继续进行训练。

表 1 默认奖励权重值设计

reward	默认权重	类型	描述
hp_point	2	dense	the rate of health point of hero
tower_hp_point	5	dense	the rate of health point of tower
money (gold)	0.006	dense	the total gold gained
ep_rate	0.75	dense	the rate of mana point
death	-1	sparse	being killed
kill	-0.6	sparse	killing an enemy hero
exp	0.006	dense	the experience gained
last_hit	0.5	sparse	the last hit for soldier

五、特征与规则

我们在初赛中暂未对原始特征进行修改，未使用后处理规则对动作进行处理。

六、 强化学习算法

我们未对比赛中所提供的强化学习算法框架进行修改，以下对比赛中算法的描述均参考自论文[1]。比赛所使用的强化学习算法是包含经验、金钱、击杀、助攻、推塔在内的多目标函数，并使用 dual-clip PPO 进行网络更新。由于训练需要大量的比赛数据且属于 off-policy 训练范式，因此采样策略可能与训练策略有着较大的差异。当 ratio 较大而 Advantage 又小于 0 时会带来较大的训练方差，因此除了使用较大的 batch size 外还利用了 Proximal Policy Optimization 的拓展算法 Dual-clip PPO。

$$\mathcal{L}^{CLIP}(\theta) = \widehat{E_t}\left[\min\left(r_t(\theta)\widehat{A_t}, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\widehat{A_t}\right)\right]$$

不同于 PPO 算法中，只使用一个裁剪因子来控制策略梯度的更新，来避免更新步长过大而导致训练不稳定。在 Dual-clip PPO 算法中，使用两个裁剪因子来控制策略梯度的上下限更新，从而确保训练过程中策略的稳定性。同时，使用两个裁剪因子还可以提高算法的效率，使得算法更快地收敛到最优策略。

我们观察到在王者荣耀游戏中，往往难以对一个动作不同标签之间的联系进行显式建模，如技能的类型与技能的方向之间的联系。因此选择直接将不同标签独立处理，来解耦它们之间的依赖关系。具体而言，解耦之前的目标函数为：

$$\max_{\theta} \hat{E}_{s,a\sim\pi_{\theta_{\text{old}}}} \left[\frac{\pi_{\theta}(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \widehat{A_t} \right]$$

而解耦后则变为了如下形式，

$$\max_{\theta} \sum_{i=0}^{N_a-1} \hat{E}_{s,a\sim\pi_{\theta_{\text{old}}}} \left[\frac{\pi_{\theta}\left(a_t^{(i)} \mid s_t\right)}{\pi_{\theta_{\text{old}}}\left(a_t^{(i)} \mid s_t\right)} \widehat{A_t} \right]$$

这样解耦不但简化了策略网络的结构，策略网络可以不再考虑动作不同属性之间的联系，在之后再进行处理，还由于每一个属性都有单独的输出通道，增加了动作的多样性。

此外，我们还参考德州扑克对于动作空间的处理[2]，为了加速训练，采用动作掩码来进行强化学习探索上的剪枝。具体来说，动作掩码用于消除以下不合理、受限制的行为：

1. 物理上禁止的行为，例如向有墙的方向移动，即长时间位移但是没有改变自身坐标；
2. 技能不可用的行为，例如在冷却时间内的技能；
3. 无意义的行为，例如在周围没有敌人时使用召唤师技能；
4. 其他特定限制下的行为，例如被敌方所控制时不采取行为。

七、 系统工程架构

调整熵损失函数的权重，进行学习率退火。在观察到离线数据积累较高时同步 GPU 速率与数据缓存器大小以减小数据复用率。

八、 模型迭代过程

模型的奖励在训练的过程中是根据当前英雄与 baseline 或者天梯赛中与其他队伍的英雄对战的情况是不断进行调整的。初期主要是通过增加稠密奖励，比如击杀小兵获得的经济，提升其对应的的基础技能。后期主要通过提高其对应的稀疏奖励，比如击杀等，提升其攻击性。期间也在不断地观察其效果，适时地增大或者减小相应的奖励。

九、 训练效果分析

由于赛题主要考查探索模型泛化性和通用性，让同一个模型，控制狄仁杰、公孙离、后羿、鲁班七号、马可波罗等五位英雄进行墨家机关道 1v1 对战。而同为射手也存在普通攻击和技能输出的派系区别，而我们的模型在马可波罗英雄熟练度较低，为此我们有意增加了马可波罗对局数量。但实际收益不大，反而使得其他英雄的表现下降。同时还观察到：模型在阵营 B 时表现较在阵营 A 时差，在学习时提高了阵营 B 对局的数量，加强训练。

除了监控“reward”，“total hurt to hero”指标外，我们还通过 ABSTools 观察具体对局情况，在英雄过于好斗不注重发育时逐步调小奖励中 money 和 exp 的权重，调大其他项；在英雄学会发育时，精进其对线技巧，逐步调小奖励中 money 和 exp 的权重，最后的模型效果稍微有所提升。

十、 总结与展望

每位成员都深度参与了本次初赛，学习到了有关强化学习训练的知识、奖励的设置、如何利用有限的资源最大化训练收益等。接下来，我们会更加投入地参加决赛，争取取得好的成绩。

参考文献

[1] Ye D, Liu Z, Sun M, et al. Mastering complex control in moba games with deep reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(04): 6672-6679.

[2] Zhao E, Yan R, Li J, et al. AlphaHoldem: High-Performance Artificial Intelligence for Heads-Up No-Limit Poker via End-to-End Reinforcement Learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2022, 36(4): 4689-4697.