УДК 004.032.26

Колчин И. А , Егошин А. В

Научный руководитель: Егошин А. В., канд. техн. наук

*Поволжский государственный технологический университет*

**ОБЗОР СУЩЕСТВУЮЩИХ ТЕХНОЛОГИЙ ИНТЕЛЛЕКТУАЛЬНОГО УПРАВЛЕНИЯ ВОЙСКАМИ В СТРАТЕГИЧЕСКИХ ВОЕННЫХ ИГРАХ**

***Аннотация****. В данной статье рассматриваются возможности применения технологий машинного обучения для решения задачи управления мультиагентной системой в многопараметрической системе на графе дорог*

***Ключевые слова****: машинное обучение(Machine learning), обучение с подкреплением(Reinforcement learning), игровой ИИ(Game AI), стратегии реального времени(Real time strategy), карты дорог(Road map).*

Адекватный игровой искусственный интеллект имеет большое значение в современных военных стратегиях, т.к. зачастую игроку очень сложно найти другого игрока для долгой совместной игры.

Ещё большую сложность представляет разработка ИИ, способного адекватно управлять армией из разнородных юнитов, передвигающихся по графу реальных дорог, учитывая множество условий: рельеф местности, ландшафт, время суток и время года, тактико-технические характеристики техники и вооружений, снабжение по дорогам и т. д. Программирование поведения армии, способной нанести поражение игроку-человеку без численного перевеса на основе жёстко заданных алгоритмов сложная задача, кроме того к такому виду ИИ гораздо проще подобрать стратегию, с помощью которой можно всегда выигрывать, такие стратегии называются доминантными.   
 Перспективным видится использование ИНС, способных к «самообучению». Искусственная нейронная сеть(ИНС, в тексте используется сокращение нейросеть) - математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. ИНС представляет из себя набор нейронов, которые представляют из себя функции с несколькими входными переменными, которые передают полученный результат следующему нейрону или наружу. ИНС является ИИ, который представляет из себя самообучающуюся программу. Программы такого типа разрабатываются в рамках машинного обучения.

Реализация игрового интеллекта на базе такой технологии в теории позволит обучить программу не только на играх с человеком (что очень медленно), а на множестве ускоренных по времени играх с самим собой. Также при достаточно длительном обучении можно получить ИИ к которому не удастся подобрать доминантную стратегию и поэтому игроку придётся лучше думать, но это также не означает что будет создан непобедимый ИИ.

Машинное обучение успешно применяется в играх разных жанров. Например

На данный момент наибольших успехов в применении машинного обучения в похожих задачах добились: Per-Arne Andersen, Morten Goodwin, Ole-Christoffer Granmo создав игру Deep RTS для исследования технологий машинного обучения[1]; Kun Shao, Yuanheng Zhu и Dongbin Zhao в своей работе использовали постепенное трансферное обучения для обучения нейросетей управлению боевыми единицами в игре StarCraft[2]; Hendrik Baier, Peter I. Cowling использовали эволюционный вариант алгоритма Монте-Карло для более быстрого обучения нейросети в пошаговой стратегии Hero Academy[3]; компания Deepmind создала нейросеть под названием «AlphaStar» для игры Starcraft 2, которая смогла обыграть двух профессиональных игроков со счётом 5:0[4];

Deep RTS[1] - это высокопроизводительная RTS-игра, созданная специально для исследований в области искусственного интеллекта. В данной стратегии игроку нужно добывать ресурсы, строить базу и армию для победы. Она поддерживает ускоренное обучение, что означает, что она может учиться в 50 000 раз быстрее по сравнению с существующими RTS играми. Такого эффекта разработчики добились за счёт того, что Deep RTS использует краткосрочные конфигурации при обучении и имеет возможность настраивать игровой таймер. Действия в краткосрочной конфигурации непосредственно применяются к среде в течение следующих нескольких игровых кадров. Это делает корреляцию между действием и вознаграждением более наблюдаемой. Долгосрочная конфигурация значительно усложняет пространство состояний, потому что чем больше временной промежуток, тем больше возможных вариантов событий может произойти. Игровой таймер включает в себя множитель, который позволяет регулировать количество тиков, равное секунде, что позволяет, по сути, ускорять или замедлять время в игре, а следовательно и влиять на скорость обучения.

Kun Shao, Yuanheng Zhu и Dongbin Zhao в своей работе[2] использовали комбинацию двух подходов трансферного обучения(Transfer Learning) и постепенного обучения(Curriculum Learning) для более быстрого обучения нейронной сети, которая управляла армией в игре Starcraft. Суть трансферного обучения заключается в том, что для создания нового слоя нейронов мы берём копию другого слоя, который выполнял похожую задачу, и обучаем его. В результате новый слой обучается в 10 раз быстрее, чем если бы мы создавали его заново(300 тренировок против 3000). Постепенное обучение представляет из себя обучение определённой последовательности постепенно усложняющихся задач, которые помогут в достижении конечной цели. Совмещение двух подходов позволило разработчикам достаточно быстро обучить нейросеть управлять различными видами боевых единиц и кроме того вероятность победы достаточна высока при произвольном количестве боевых единиц(выше 80%)

Hendrik Baier и Peter I. Cowling в своей работе[3] рассмотрели улучшенный вариант алгоритма поиска дерева решений Монте-Карло(Monte-Carlo tree search, MCTS) — это эволюционный MCTS (evolutionary MCTS, EMCTS). MCTS используется для поиска оптимального решения и регулирования параметров нейронной сети. EMCTS отличается от классического MCTS тем, что он использует эволюционные алгоритмы для более быстрого поиска решения, суть которых заключается в том что выбирается действие давшее больший результат в текущем состоянии. MCTS строит полное дерево решений и только после этого оценивает полезность каждого узла, а EMCTS в процессе построения строит дерево, в котором остаются только те узлы, которые принесли наибольшую выгоду. EMCTS согласно результатам исследования хорошо масштабируется и показывает себя лучше чем MCTS на таких сложных играх как стратегии. EMCTS тестировался только на пошаговой стратегии Hero Academy. Авторы отмечают, что данный алгоритм этот алгоритм может хорошо себя показать в более сложных играх, таких как XCOM или Civilization.

Компания Deepmind создала нейросеть под названием «AlphaStar»[4]. AlphaStar использует мультиагентный процесс обучения с подкреплением. Суть этого подхода в том, что сначала создаётся несколько агентов и они обучаются друг на друге какое-то время, затем на их основе создаются новые. Из новых агентов формируется лига внутри которой они сражаются между собой. Для большего охвата различных стратегий, каждому новому агенту ставили свою цель(например, научиться обыгрывать конкретного агента или группу агентов). Скомбинировав различных агентов с наиболее эффективными стратегиями, которые получились при применении такого подхода, разработчикам удалось добиться победы ИИ над игроками.

Целью данного исследования является исследование применения современных технологий машинного обучения для решения задачи управления мультиагентной системой в многопараметрической системе на графе дорог. Для того, чтобы выполнить эту цель необходимо разработать нейронную сеть, которая будет решать описанную ранее задачу.

Представленная нейросеть будет первым применением технологии машинного обучения в задача такого типа. Нейросеть будет управлять различными видами войск: пехота, артиллерия, мотострелковые подразделения, танки, конвои грузовиков для снабжения войск и поезда для переброски на дальние расстояния. У каждого подразделения имеется запас сил, боеприпасов, еды, воды, кроме этого при ведении боя учитываются боевые характеристики(боевой дух, боевой опыт, дальность атаки, скорость передвижения). Все эти характеристики и особенности графа дорог(множество путей, разветвлённость и плотность дорог) будут учитываться нейронной сетью при достижении цели.

В рассмотренных работах рассматривались игры, в которых действия происходят на графах похожих на шахматную доску. Такие графы структурно похожи на графы дорог, с тем отличием что графы вида «шахматная доска» не такие плотные и разветвлённые, и в качестве вершин у них клетки вместо мест стыковок дорог. Из этого следует что рассмотренные решения могут быть применены для решения задач на графе дорог.

Литература

1. Per-Arne Andersen, Morten Goodwin, Ole-Christoffer Granmo «Deep RTS: A Game Environment for Deep Reinforcement Learning in Real-Time Strategy Games» [Электронный ресурс] - Режим доступа: <https://arxiv.org/abs/1808.05032>, свободный
2. Kun Shao, Yuanheng Zhu, Member, IEEE and Dongbin Zhao, Senior Member «StarCraft Micromanagement with Reinforcement Learning and Curriculum Transfer Learning» [Электронный ресурс] - Режим доступа: https://arxiv.org/pdf/1804.00810.pdf, свободный
3. Hendrik Baier, Peter I. Cowling «Evolutionary MCTS for Multi-Action Adversarial Games» [Электронный ресурс] - Режим доступа: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwjqpb-r-qbgAhWI8ywKHdj5B5YQFjAAegQICRAC&url=https%3A%2F%2Fhendrikbaier.jimdo.com%2Fapp%2Fdownload%2F13604570727%2Fevolutionary%2520mcts%2520for%2520multi-action%2520adversarial%2520games.pdf%3Ft%3D1532703212&usg=AOvVaw3M9xX0AbJl4h5uYAy8GBBh, свободный
4. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II [Электронный ресурс] - Режим доступа: https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/ , свободный