УДК 004.852

ИСПОЛЬЗОВАНИЕ ГЛУБОКОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПЛАНИРОВАНИЯ ПУТИ ПОКРЫТИЯ

Луканов Сергей Юрьевич, аспирант,
Псковский государственный университет

APPLICATION OF DEEP REINFORCEMENT LEARNING TO THE PROBLEM OF COVERAGE PATH PLANNING

Lukanov Sergey, postgraduate, lukanovysergey@gmail.com
Pskov State University

В статье рассмотрен способ получения модели нейронной сети для решения задачи планирования пути покрытия с помощью глубокого обучения с подкреплением. Приведены результаты обучения с различными функциями вознаграждения, размерами полносвязных нейронных сетей и гиперпараметрами.

Ключевые слова: *глубокое обучение с подкреплением, управление, нейронная сеть, планирование пути покрытия.*

The article presents a method for developing a neural network model to solve the task of coverage path planning using deep reinforcement learning. It provides training results using various reward functions, sizes of fully connected networks and hyperparameters.

Keywords: *deep reinforcement learning, control, neural network, coverage path planning.*

Задача планирования пути покрытия актуальна в различных гражданских сферах, где необходимо провести обследование или осуществить мониторинг местности: сельское хозяйство, охрана окружающей среды, геодезия, спасательные операции и др.

Задача формулируется следующим образом: дана целевая зона, агент с ограниченной областью наблюдений, карта местности и ограничения. Требуется найти траекторию агента, такую, что вся зона будет полностью покрыта, а заданный критерий ограничения оптимизируется. Задача является NP-трудной и для решения могут быть использованы различные эвристические и вероятностные методы. Одним из применяемых методов является жадная стратегия, при которой агент перемещается к ближайшей непокрытой точке.

Методы глубокого обучения с подкреплением получили значительный толчок в развитии и позволяют решать сложные задачи управления. Достижения компании Google DeepMind, связанные с их моделями AlphaGo, AlphaZero и AlphaFold, вдохновили новые исследования в области, продемонстрировав свою эффективность.

Глубокое обучение с подкреплением использует идеи динамического программирования, итеративно обновляя веса нейронной сети, высчитывая градиент на основе наград по траекториям. Система в данном случае состоит из агента и среды. Агент находится в состоянии s, наблюдает за средой, принимает решения о дальнейшем действии a на основе стратегии π и получает в ответ награду r. Проблемы обучения, помимо прочего, связаны с задержанным по времени сигналом обратной связи, разреженностью наград, необходимостью эффективно исследовать пространство возможных состояний.

На практике популярен подход актер-критик. Критик аппроксимирует ценность состояния V или ценность пары состояние-действие Q, а актер обновляет параметры стратегии, опираясь на эту оценку.

Для получения модели управления, была разработана среда на платформе Unity. Выбор платформы мотивирован удобством разработки и наличием легко настраиваемого обучения с плагином Unity ML Agents [1]. Среда представляет собой набор из заранее обработанных спутниковых карт, где цель агента – покрыть все водоемы. Также разработана многоагентная версия для n агентов и версии совместимые с gym и pettingzoo интерфейсами.

Водоемы и агенты распределены по отдельным цветовым каналам, земля маскируется черным цветом. За обработку визуальных наблюдений отвечает трехслойная сверточная нейронная сеть, позволяющая учитывать пространственные взаимосвязи в данных. Агент получает награду за каждый покрытый пиксель водоема, может двигаться в восьми направлениях, ограничение по времени формируется за счет введение штрафа на каждом шаге в функцию вознаграждения. Полученные стратегии сравниваются друг с другом по скорости покрытия зоны. Также к сравнению добавлена реализация простого жадного алгоритма.

Для обучения среды с одним агентом используются алгоритмы SAC и PPO, для обучения среды с несколькими агентами к ним добавляется алгоритм POCA.

В SAC (Soft Actor-Critic) в целевой функции учитывается вознаграждение и энтропия, таким образом алгоритм стремится максимизировать суммарное вознаграждение и простимулировать эффективное исследование среды. Обучаются стохастическая стратегия и две Q-сети [2].

PPO (Proximal Policy Optimization) оптимизирует параметры стратегии напрямую на основе оценки функции преимущества A. Функция преимущества оценивает, насколько выбор действия a лучше среднего результата начиная из текущего состояния. В PPO изменения за шаг ограничены специальным параметром, что способствует более стабильному обучению [3].

POCA (Posthumous Credit Assignment) направлен на учет групповых взаимосвязей в работе нескольких агентов. Используется механизм внимания, что позволяет динамически изменять количество активных агентов [4].

Для различных сценариев обучения было произведено по три обучающих прогона, из которых был выбран лучший по графикам средней суммарной награды и средней длины эпизода. Далее для

полученной модели нейронной сети строился исполняемый файл, записывающий прогресс покрытия карт. Всего использовано сорок различных карт местности с двумя вариантами размеров. Для полученного набора прогрессов выводятся сравнительные матрицы эффективности и графики зависимости площади покрытой зоны от времени.

Для изучения влияния на качество модели различных функций вознаграждения, были выбраны следующие награды за покрытие одного пикселя: 1 и 2; и следующие штрафы за временной шаг: 1, 3 и 5. Всего получено шесть комбинаций, представляющих различные соотношения награды и штрафа. Результаты показали, что комбинация с наградой 2 и штрафом 1 является самой эффективной. На рис. 1-2 приведен пример сравнительной матрицы и графика. Ячейка матрицы показывает в скольких случаях результаты модели, указанной в строке, лучше, чем у модели в столбце.

Рисунок 1. – Сравнительная матрица эффективности моделей

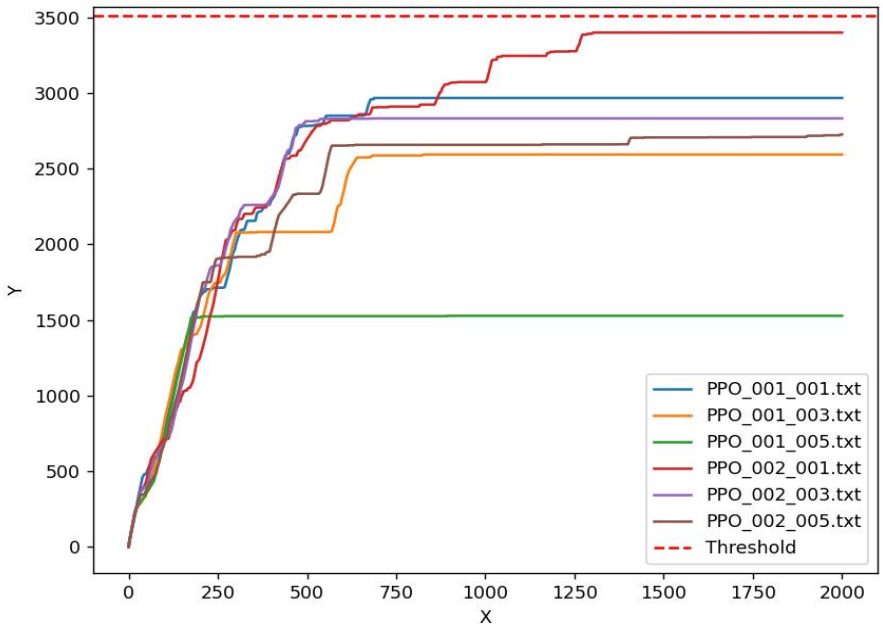

Рисунок 2. – График скорости покрытия

Среди протестированных размеров полносвязных нейронных сетей комбинации с 256, 512, 1024 нейронов в слое с количеством слоев 2 или 3. Лучшей комбинацией оказались 256 нейронов и 3 слоя.

Размер пакета для вычисления градиента влиял на стабильность процесса обучения. Большие значения приводят к меньшим колебаниям на графике суммарной награды, но к более редким обновлениям параметров и требуют больше времени на сбор данных.

Подбор параметра λ в алгоритме PPO для вычисления взвешенного преимущества показал значительную разницу в скорости обучения и эффективности итоговой модели. Из протестированных значений 0.95, 0.97 и 0.99 лучшим оказалось значение в 0.99.

Горизонт времени протестирован на значениях в 64, 128 и 256. Лучшим оказался вариант в 128 шагов.

Для повышения точности позиционирования агента протестирован прием добавления двух дополнительных каналов с координатами для более эффективной обработки пространственной информации [5]. Полученные результаты не показали статистически значимого повышения эффективности, что может быть связано с небольшим размером карт и выделением отдельных каналов под агентов и водоемы.

Многоагентная среда характеризуется своей не статичностью. Агенты могут быть обучены конкурентно, тогда решение стремится к равновесию Нэша, или же кооперативно, что требует изменений в целевой функции и учет вклада каждого агента. Обучение проводилось для четырех агентов.

Как в случае с одним агентом, так и в случае с несколькими, лучше всего себя показал алгоритм PPO, отличившись более стабильным обучением, быстрой сходимостью и устойчивостью в среде с несколькими агентами. Если сравнивать с жадным алгоритмом, то полученные модели существенно уступают ему в эффективности. Однако, анализ данных показал, что это связано с проблемами на поздних этапах покрытия. Если обрезать целевую площадь на 20-30%, то видно, что полученные модели опережают жадный алгоритм. Если дообучить модель управления на разреженных картах, скорректировав функцию вознаграждения, то можно получить модель управления, точность которой не будет страдать в конце эпизода.

Список использованных источников

1. Juliani A. et al. Unity: A general platform for intelligent agents //arXiv preprint arXiv:1809.02627. – 2018. [Электронный ресурс]. – Режим доступа: https://doi.org/10.48550/arXiv.1809.02627 / – Дата доступа: 07.04.2025

2. Haarnoja T. Et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor //International conference on machine learning. – Pmlr, 2018. – С. 1861-1870. [Электронный ресурс]. – Режим доступа: https://arxiv.org/abs/1801.01290 / – Дата доступа: 07.04.2025

3. Schulman J. Et al. Proximal policy optimization algorithms //arXiv preprint arXiv:1707.06347. – 2017. [Электронный ресурс]. – Режим доступа: https://arxiv.org/abs/1707.06347 / – Дата доступа: 07.04.2025

4. Cohen A. et al. On the use and misuse of absorbing states in multi-agent reinforcement learning //arXiv preprint arXiv:2111.05992. – 2021. [Электронный ресурс]. – Режим доступа: https://arxiv.org/abs/2111.05992 / – Дата доступа: 07.04.2025

5. Liu R. et al. An intriguing failing of convolutional neural networks and the coordconv solution //Advances in neural information processing systems. – 2018. – Т. 31. [Электронный ресурс]. – Режим доступа: https://arxiv.org/abs/1807.03247 / – Дата доступа: 07.04.2025