УДК 681.3

# МЕТОД «АКТОР-КРИТИК» В ЗАДАЧАХ УПРАВЛЕНИЯ МОБИЛЬНЫМИ КИБЕРФИЗИЧЕСКИМИ СИСТЕМАМИ

***М.Л. Паткин***

Самарский государственный технический университет
Россия, 443100, г. Самара, ул. Молодогвардейская, 244

*Рассмотрена процедура синтеза устойчивого к стохастическим изменениям среды нейросетевого агента, реализующего управление в задачах преследования для мобильной киберфизической системы в виде машины Дубинса. Синтез нейросетевого агента предложено производить посредством обучения нейронной сети методом «Актор-Критик». Построена компьютерная модель обучения и тестирования агента для различных параметров машины Дубинса и среды. Произведено сравнение предлагаемого метода синтеза нейросетевого агента с методом синтеза с помощью «жадного алгоритма» для различных вариантов поведения жертвы и для различных параметров машины Дубинса. Произведена визуализация выходов нейронной сети Актора, что показало, как нейронная сеть работает и реагирует на окружающую среду.*

***Ключевые слова:*** *обучение с подкреплением, метод «Актор-Критик», машина Дубинса, искусственные нейронные сети, киберфизическая система.*

Искусственные нейронные сети (ИНС) за последнее время показывают выдающуюся эффективность в различных сферах науки и техники. Ученые разрабатывают все новые методы для улучшения качества и ускорения обучения ИНС. Конечно, такая революция в области ИНС не могла пройти мимо такого большого раздела, как обучение с подкреплением (reinforcement learning – RL).

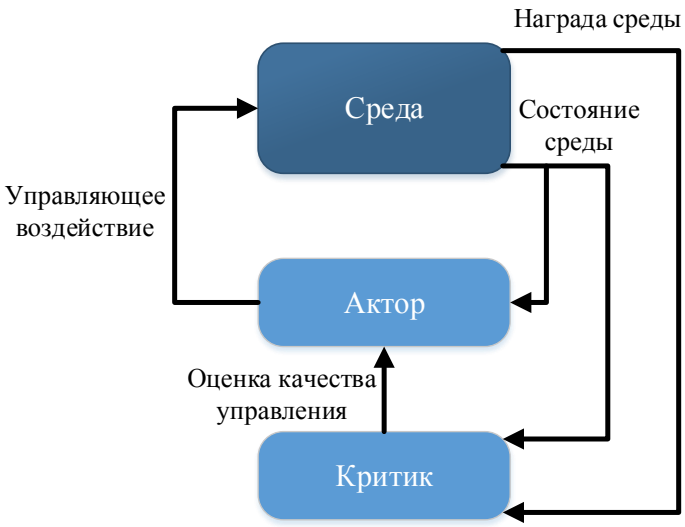



Рис. 1. Диаграмма работы АК-метода

*Михаил Львович Паткин, аспирант.*

Использование методов RL в задачах управления мобильными кибернетическими системами дает существенное преимущество по сравнению с эвристическими методами. При изменении свойств среды достаточно будет обучить агента заново или переобучить предварительно обученного агента для приспособления к новым условиям. В эвристических же методах изменение среды может фатально повлиять на качество работы мобильного агента.

Метод «Актор-Критик» (АК) [1] является одним из самых распространенных методов в RL. Главная идея метода представляет собой структурную независимость блока, отвечающего за выработку управляющего воздействия – Актора от блока, который служит для оценки успешности действий Актора (рис. 1). Метод «Актор-Критик» сочетает в себе преимущества методов, основанных на оценке функции полезности (Value-based) и градиентных (Policy-based) методов [2].

**Описание агента и процесса обучения**

Основой агента являются модули Актор и Критик, представляющие собой глубокие ИНН, конфигурации которых приведены в табл. 1.

*Таблица 1*

**Структуры Актора и Критика**

| | *Структура Актора* | *Структура Критика* |
|---|---|---|
| 1 | Полносвязный слой: 256 нейрона | Полносвязный слой: 256 нейрона |
| 2 | Регуляризация Дропаут: 20 % | Регуляризация Дропаут: 20 % |
| 3 | Активационный слой: РЕЛУ [3] | Активационный слой: РЕЛУ |
| 4 | Полносвязный слой: 256 нейрона | Полносвязный слой: 256 нейрона |
| 5 | Регуляризация Дропаут: 20 % | Регуляризация Дропаут: 20 % |
| 6 | Активационный слой: РЕЛУ | Активационный слой: РЕЛУ |
| 7 | Полносвязный слой: 3 нейрона | Полносвязный слой: 1 нейрона |

В табл. 1 Дропаут (Dropout) 20 % – слой, в котором обнуляется 20 % выходных значений. РЕЛУ (ReLU – Rectified Linear Unit) – слой нелинейности по функции $f(x) = \max(0, x)$.

Обучение агента делится на две большие части – обучение Критика и Актора. Схематично обучение представлено на рис. 2.

**Обучение Критика**

Задачей Критика является аппроксимация *Q*-функции – функции полезности управления. ИНН Критика обучается с использованием градиентов, которые происходят от TD-ошибки (ошибки временной разности между оценками полезности на предыдущем и текущем шаге [4])

$$y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'(s_{i+1}\left|\theta^{\mu'}\right)\right|\theta^{Q'}\Big), \quad (1)$$

где $r$ – награда среды;

$s$ – состояние среды;

$\gamma$ – весовой коэффициент;

$\mu'$ – функция Актора;

$\theta^{\mu'}$ – веса нейронной сети Актора;

$\theta^{Q'}$ – веса нейронной сети Критика.

Веса нейронной сети критика обучаются с помощью градиентов, полученных от среднеквадратичной функции потерь *L:*

$$L = \frac{1}{N}\sum_i \left( y_i - Q\left(s_i, a_i \middle| \theta^{Q'}\right)^2 \right), \tag{2}$$

где *N* – размер минибатча данных. Как видно из (2), *L* является среднеквадратичной функцией потерь.

**Обучение Актора**

Обучение Актора осуществляется по методу DDPG (deep deterministic policy gradient – глубокий градиентный метод). В [4] показано, что функцию DPG можно свести к SPG (stochastic policy gradient – стохастический градиентный метод). Функция DPG рассчитывается согласно алгоритму

$$\nabla_{Q^\mu} \mu \approx \mathrm{E}_{\mu'}\left[ \nabla_a Q\left(s, a \middle| \theta^Q\right)\Big|_{s=s_t, a=\mu(s_t)} \nabla_{Q^\mu} \mu\left(s \middle| \theta^\mu\right)\Big|_{s=s_t} \right], \tag{3}$$

где $\mathrm{E}_{\mu'}$ – отношение между наградой, полученной Актором, и предсказанной наградой.

В качестве алгоритма оптимизации и в случае Актора, и в случае Критика был выбран RMSProp [5, 6].

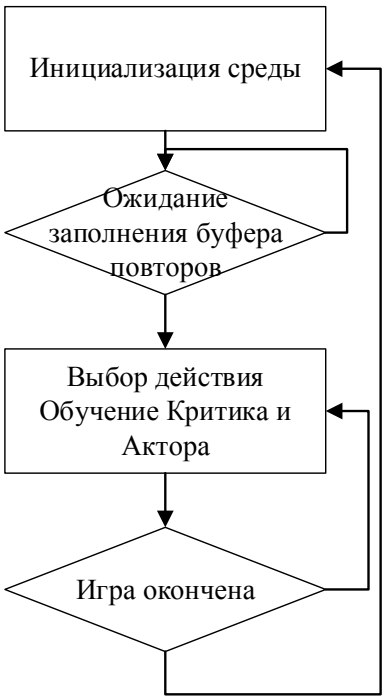



Рис. 2. Процесс обучения агента

Обучение проводилось на компьютере с процессором Intel i7 3,4 GHz и видеокарте Nvidia Titan Black и заняло 30 минут.

**Описание среды**

Среда (рис. 3), в которой действует мобильный агент (охотник), представляет собой поле шириной 1000 пикселей и высотой 700. В среде находятся жертва, препятствия и мобильный агент. Задача агента – приблизиться к жертве на расстояние не менее 0,5 м (что соответствует 100 пикселям на экране) относительно центра жертвы, при этом не задев препятствия и границы среды. У мобильного агента есть 4 сенсора: 3 сонара и компас, который показывает относительный угол между осью движения мобильного агента и центром жертвы. Управление мобильным агентом осуществляется посредством выбора одной из трех команд: налево, прямо, направо. Управление скоростью робота не предполагается в данной задаче. Инициализация робота происходит всегда в одной позиции, но угол каждый раз выбирается случайно. Также среда поддерживает режим, при котором жертва может двигаться случайно по плоскости.

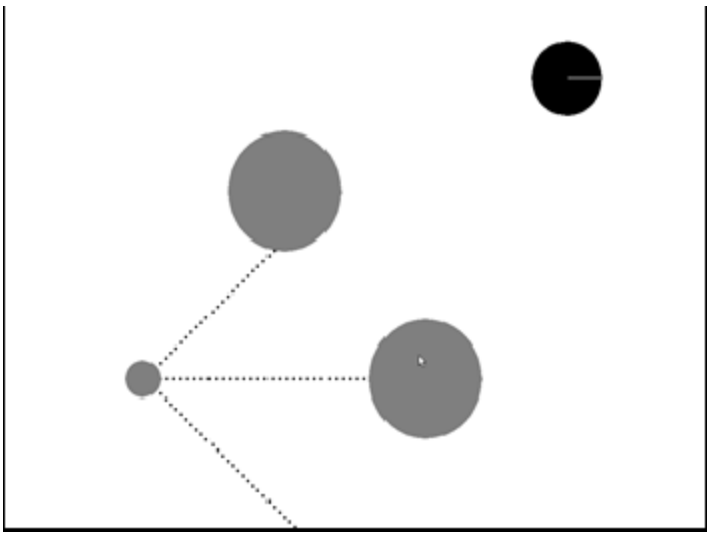


Рис. 3. Визуализация среды, объекты: маленький серый – хищник, черные точки – линии действия сонаров, два больших серых – препятствия, черный – жертва

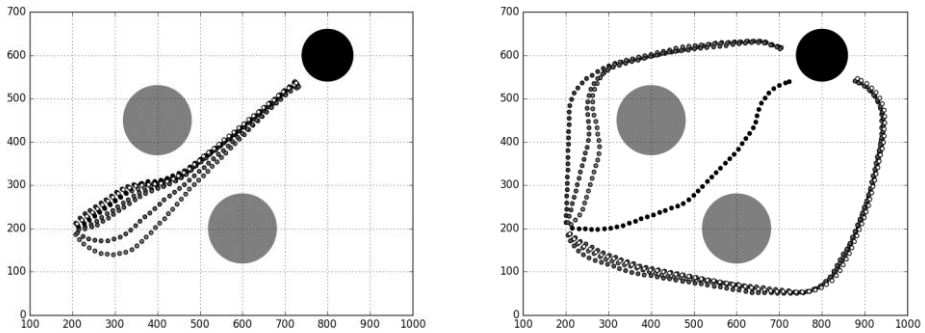
**Тестирование**

Тестирование проводилось для случаев со статической жертвой, со стохастической жертвой, при котором жертва выбирала направление движения случайно, а также с убегающей жертвой, при котором жертва всегда убегает под углом +/-90° относительно приближающегося хищника. И в случае со стохастической жертвой, и с убегающей скорость жертвы равна скорости хищника. Также тестирование проводилось для случаев с минимальными радиусами разворота машины Дубинса, равными 50 и 100 пикселей. Качество агента сравнивалось с «жадным алгоритмом», его схема работы:

– если угол до цели меньше -3°, то выбираем поворот налево. Если угол больше 3°, то выбираем поворот направо. В остальных случаях выбираем движение прямо;

– если расстояние до препятствия по данным левого сонара меньше 100 пикселей, то выбираем поворот направо. Если расстояние до препятствия по данным правого и центрального сенсора меньше 100 пикселей, то выбираем поворот налево.

Качество агента оценивалось как доля успешных запусков, то есть случаев, когда машина поймала жертву. В табл. 2 приведены результаты тестирования «жадного алгоритма» и модели «Актор-Критик», которое проводилось на 1000 повторений. На рис. 4 представлены примеры траекторий для «жадного алгоритма» и метода «Актор-Критик».



*а* *б*

Рис. 4. Статическая жертва: *а* – траектория при «жадном алгоритме»; *б* – траектория при методе «Актор-Критик»

*Таблица 2*

**Доли успешных запусков для «жадного алгоритма» и модели «Актор-Критик»**

| Показатель | «Жадный алгоритм» | «Актор-Критик» |
|---|---|---|
| *Статическая цель/Минимальный радиус 50* | *0,372* | *0,788* |
| *Статическая цель/Минимальный радиус 100* | *0,832* | *0,911* |
| *Динамическая цель/Минимальный радиус 50* | *0,399* | *0,735* |
| *Динамическая цель/Минимальный радиус 100* | *0,757* | *0,762* |
| *Убегающая цель/Минимальный радиус 50* | 0,370 | 0,766 |
| *Убегающая цель/Минимальный радиус 100* | 0,697 | 0,758 |

Как видно из табл. 2, во всех случаях метод «Актор-Критик» выигрывает, либо совсем немного, либо существенно.

**Сравнение с животным миром**

В статье [7] было высказано предположение, что в животном мире за направление и за близость к жертве у хищника в мозге отвечают разные группы нейронов.

Была проверена гипотеза о том, что Актор после обучения имеет нечто похожее на то, что происходит в головном мозге хищника, а именно разные акти-

вации групп нейронов в разных ситуациях. Однако после проведения ряда экспериментов и визуализации выходов второго полносвязного слоя было замечено (рис. 5, где 256 нейронов этого слоя изображены в виде квадрата 16 на 16), что яркие активации (значения выходов, значительно отличающихся от нуля) происходят в моменты, когда агенту нужно принять решение об изменении направления, чтобы не врезаться в препятствие (рис. 5, *в*, *г*). В моменты, когда агенту не нужно принимать срочного решения о смене траектории движения, ярких активаций немного (рис. 5, *а*, *б*).

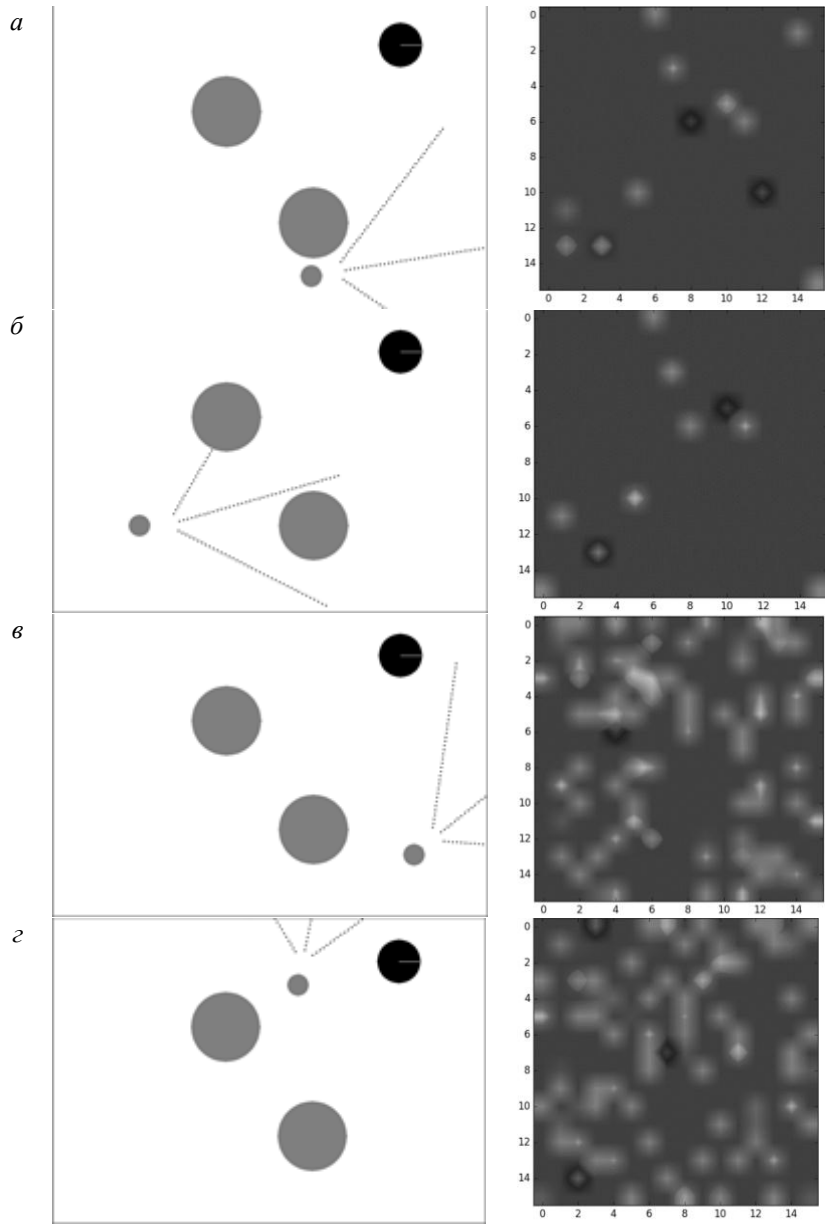



Рис. 5. Карта активации выхода 2-го полносвязного слоя Актора

**Выводы**

Были созданы среда и модель машины Дубинса, для которой синтезирован и обучен агент на основе метода «Актор-Критик», который показал лучшие результаты в задачах преследования в сравнении с агентом на основе «жадного алгоритма». Следующим этапом готовится модификация среды и алгоритма для мультиагентного случая.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК


1. *Konda V.R., Tsitsiklis J.N.* Actor-Critic Algorithms // NIPS. – 1999. – Т. 13. – С. 1008-1014.
2. *Bhatnagar S. et al*. Natural actor-critic algorithms //Automatica. – 2009. – Т. 45. – №. 11.
3. *Nair V., Hinton G.E*. Rectified linear units improve restricted boltzmann machines // Proceedings of the 27th international conference on machine learning (ICML-10). – 2010. – С. 807-814.
4. *Silver D. et al*. Deterministic Policy Gradient Algorithms // Proceedings of The 31st International Conference on Machine Learning. – 2014. – С. 387-395. *Witten I. H. et al.* Data Mining: Practical machine learning tools and techniques. – Morgan Kaufmann, 2016.
5. *Tieleman T., Hinton G*. Lecture 6.5-RMSProp, COURSERA: Neural networks for machine learning // University of Toronto, Tech. Rep. – 2012.
6. *Sarel A. et al.* Vectorial representation of spatial goals in the hippocampus of bats // Science. – 2017. – Т. 355. – № 6321. – С. 176-180.



*Статья поступила в редакцию 15 января 2017 г.*


## ACTOR-CRITIC METHOD IN MOBILE CYBER-PHYSICAL SYSTEMS CONTROL PROBLEM


***M.L. Patkin***

Samara State Technical University
244, Molodogvardeyskaya st., Samara, 443100, Russian Federation



*A synthesis procedure of neural network agent that is resistant to stochastic environmental changes that implement the control in the problem of prosecution for mobile cyber-physical system as Dubins Car is considered. Synthesis of neural network agent asked to produce by training "Actor-Critic" neural network. A computer model of training and testing agent for various parameters Dubins car and environment is developed. A comparison between the suggested method and the greedy algorithm for the various cases of victims' parameters as well as for the cases with different Dubins Car parameters is made. The visualization of outputs of the Actor neural network that showed how a responsive neural network on the environment is performed.*



***Keywords****: reinforcement learning, Actor-Critic method, Dubins car, artificial neural networks, cyber-physical system.*



*Mihail L. Patkin, Postgraduate Student.*