

РАЗЛИЧНЫЕ ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ ПЕРЕВЕРНУТОГО МАЯТНИКА*

В.И. ШИПАГИН

630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, аспирант кафедры автоматики. E-mail: shipagin@gmail.com

В отличие от классического маятника перевернутый находится в неустойчивом положении. Чтобы оставаться в вертикальном состоянии, необходима его балансировка. Поэтому решение этой задачи может быть использовано для тестирования алгоритмов управления, таких как нейронные сети, генетические алгоритмы, алгоритмы нечеткого управления, а также PID-controllers.

В настоящей работе сравнивались результаты тестирования нескольких алгоритмов управления (алгоритм случайного поиска, алгоритм поиска с восхождением к вершине, алгоритм градиента политик с использованием нейронной сети) на примере решения задачи перевернутого маятника по следующим критериям: время обучения, скорость работы, используемая память.

Для программной реализации использовался язык Python (v.3.6). Решение задачи осуществлялось с помощью бесплатного инструмента OpenAI Gym. Эта платформа предназначена для разработки и тренировки алгоритмов с помощью игр и иных испытаний [1]. В OpenAI Gym имеются разные среды для тестирования. В статье будет использоваться среда «CartPole_v0», имитирующая поведение перевернутого маятника. Такая система в каждый момент временного интервала может предоставить координаты положения тележки и угол отклонения стержня маятника [2, 4]. В зависимости от предоставленных данных алгоритм должен принять решение о воздействии на тележку с целью ее перемещения «влево» или «вправо».

Ключевые слова: нейронные сети, перевернутый маятник, искусственный интеллект, алгоритм случайного поиска, функция активации, обучение

ВВЕДЕНИЕ И ПОСТАНОВКА ЗАДАЧИ

Обратный маятник имеет центр масс выше своей точки опоры, закрепленный на конце жесткого стержня. Массу маятника обозначим m (рис. 1) Существует большое количество разновидностей перевернутых маятников. В насто-

* Статья получена 02 апреля 2019 г.

ящей работе будем использовать модель с опорой на тележке, которая может перемещаться по горизонтали. Массу тележки обозначим через M . Сила трения при движении тележки не учитывается. Кроме того, для упрощения решаемой задачи считаем, что стержень не имеет массы. Его длину обозначим через l . Задача системы управления состоит в удержании перевернутого маятника в вертикальном положении за счет смещения тележки [3, 9].

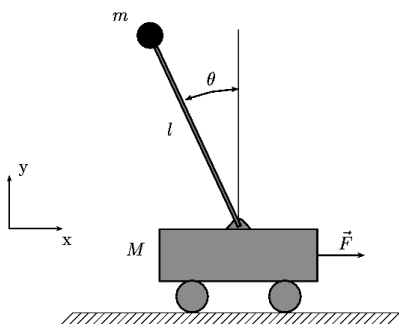


Рис. 1. Схематическое представление обратного маятника на тележке

Таким образом, значениями системы «перевернутый маятник», которые мы будем отслеживать, являются: угол отклонения маятника от вертикали θ и позиция тележки x . Для стабилизации системы будем управлять перемещением тележки, изменяя ее положение x . Сила воздействия на тележку F (будем считать, что сила принимает значения: « -1 » – толкаем влево, « $+1$ » – толкаем вправо). Задачей системы управления является приведение угла наклона маятника к значению, близкому к нулю (устойчивое положение маятника).

Попытка заканчивается, когда угол отклонения маятника достигает значения более чем 15 градусов или когда тележка уходит от изначального положения на 2,4 единицы (покидает область экрана).

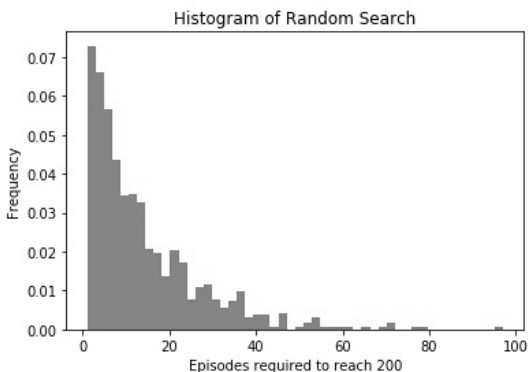
Принятие решения о воздействии на тележку (движение тележки влево или вправо) принимается после суммирования значений, полученных в результате наблюдения, умноженных на соответствующие им веса. Если конечный результат меньше нуля, тележка движется влево, иначе – вправо.

Для того чтобы понимать, насколько хорошо подобран вектор весов, введем оценочный показатель. Будем считать, что для каждого интервала времени, пока угол отклонения маятника не достиг порогового значения (пока маятник находится в состоянии равновесия), добавляем $+1$ к нашему оценочному показателю [5].

РЕШЕНИЕ ЗАДАЧИ

Перейдем к одному из методов численной оптимизации – случайному поиску. Одной из ключевых особенностей данного метода является отсутствие потребности в вычислении градиента для решения задач оптимизации. Методом случайного перебора пробуются различные веса и выбирается наилучший [6, 10, 11, 14].

Гистограмма эффективности использования данного алгоритма показана на рис. 2.



13.662

Рис. 2. Гистограмма эффективности использования алгоритма случайного поиска

Для сбора статистики запустим алгоритм 1000 раз. Результаты работы алгоритма представлены на гистограмме (рис. 2). По оси абсцисс представлено количество эпизодов, необходимых для обучения (маятник не падает в течение 200 шагов и более); по оси ординат показана частота, с которой это решение встречается.

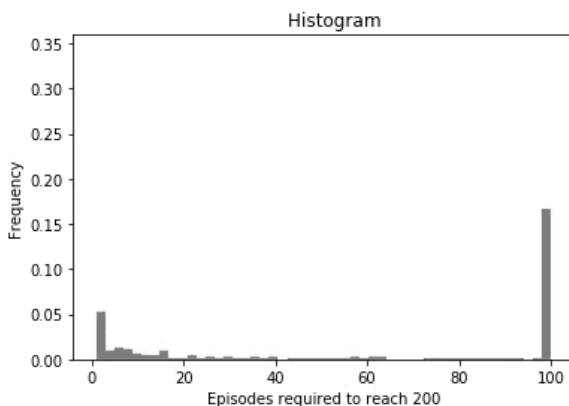
На гистограмме видно, что среднее число эпизодов, необходимое для обучения (поиска оптимальных параметров), составляет 13,662.

В отличие от первого метода, в алгоритме поиска с восхождением к вершине веса изменяются не скачкообразно, а постепенно.

К недостаткам данного метода по сравнению с предыдущим можно отнести возможное зависание параметров на некоем небольшом локальном максимуме или «плато». То есть такой случай, когда небольшое изменение весов (шум) не позволяют оценить, ближе ли данное решение к максимуму или нет.

Для исключения данной ошибки необходимо внимательно подходить к размеру шума. Если размер будет слишком большим, то мы получим алгоритм случайного поиска. Если слишком малым, то высока вероятность войти в стагнацию параметров [7, 12]. Размер шума был подобран эмпирически и равен 0,1.

Гистограмма эффективности использования данного алгоритма представлена на рис. 2.



7.3813

Рис. 3. Гистограмма эффективности алгоритма поиска с восхождением к вершине

Как видно из графика, довольно большое число решений было отрицательным (т. е. за 100 шагов не было найдено таких параметров, которые позволили бы удерживать маятник в состоянии равновесия 200 итераций и более). Несмотря на это, среднее число эпизодов, необходимых для решения задачи, составило 7,38.

В методе градиентного спуска меняем правило с детерминированного подхода (воздействие на тележку влево, если сумма умножения весов на параметры меньше нуля, и вправо – в противном случае) на стохастический (вероятностный). Каждое воздействие (движение тележки влево или вправо) описывается своей линейной комбинацией. С помощью функции активации выбираем предпочтительное действие для данного набора параметров системы.

Так как у нас нет базы знаний, содержащей информацию по управляющему воздействию для каждого состояния системы, то метод обучения «с учите-

лем» не подходит. Однако, для каждого воздействия на систему мы можем получить обратную связь – вознаграждение [8, 13, 15].

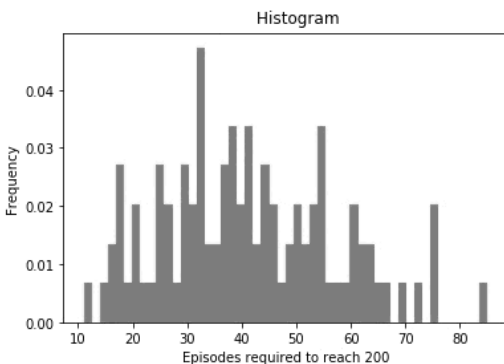
В качестве отправной точки создадим эталонную базу результатов работы алгоритма с равновероятностными воздействиями для каждого события системы. Для этого будем использовать нейронную сеть с одним скрытым слоем.

Для «обучения» данной сети запускаем алгоритм многократно. На каждой итерации работы алгоритма записываем первоначальное положение параметров маятника, значения весов управляющего воздействия и результат, к которому данное воздействие привело.

На следующем этапе будем учитывать улучшения от каждого изменения параметров и на основании этого обновляем нейронную сеть. Учитывать будем среднее значение награды по всем действиям, а не конкретное действие, взятое от каждого состояния.

Оптимизацию нейронной сети можем осуществлять благодаря случайно найденным значениям при различных состояниях системы. Выбираем изменения, при которых средняя награда по всем состояниям оказалась выше текущей. Преимуществом будет являться разница текущего и среднего значений. Будем учитывать преимущество для обновления политики.

На рис. 4 представлена гистограмма эффективности использования данного алгоритма.



41.27

Рис. 4. Гистограмма эффективности использования метода градиентного спуска

Как видно из графика, в среднем алгоритму градиентного спуска требуется 41,27 итерации для поиска решения задачи.

ЗАКЛЮЧЕНИЕ

Алгоритм градиента политик по времени обучения уступает алгоритмам случайного поиска и поиска с восхождением к вершине. Это связано с временными затратами на обучение нейронной сети. Агент не может делать правильные предположения о воздействии на систему без обучения функции полезности. Для первых двух случаев улучшения наступают сразу.

Метод градиентных политик подходит для систем с многомерными пространствами состояний. Этот подход гарантирует движение к наиболее эффективным параметрам. И, напротив, для более простых случаев подходят алгоритмы случайного поиска и поиска с восхождением к вершине, но в некоторых случаях они не могут гарантировать поиск наилучшего решения.

Алгоритм	Среднее число эпизодов, необходимых для обучения	Время обучения (с)	Память (байт)
Алгоритм случайного поиска	13,53	10	192
Алгоритм поиска с восхождением к вершине	7,43	6	356
Алгоритм градиента политик	41,27	71	443

Сравнительный анализ показал преимущества и недостатки каждого метода. Так алгоритмы случайного поиска и поиска с восхождением к вершине показали хорошую скорость работы и низкие требования к занимаемой памяти на компьютере. Но работа этих алгоритмов не гарантирует поиск оптимального решения в некоторых случаях.

Алгоритм политики градиентов с использованием нейронной сети хорошо справляется с задачей поиска глобального максимума, но требователен к ресурсам компьютера.

СПИСОК ЛИТЕРАТУРЫ

1. OpenAI [Электронный ресурс] // Википедия: web-сайт. – 2019. – URL: <https://ru.wikipedia.org/wiki/OpenAI> (дата обращения: 18.12.2019).
2. Getting Started with Gym [Electronic resource] // OpenAI: website. – 2019. – URL: <https://gym.openai.com/docs/> (accessed: 18.12.2019).
3. Обратный маятник [Электронный ресурс] // Википедия: web-сайт. – 2019. – URL: https://ru.wikipedia.org/wiki/Обратный_маятник (дата обращения: 18.12.2019).
4. OpenAI Universe. Открытая платформа для тренировки сильного ИИ [Электронный ресурс]. – 2016. – URL: <https://geektimes.ru/post/283384/> (дата обращения: 18.12.2019).
5. Frans K. Simple reinforcement learning methods to learn CartPole [Electronic resource] // Kevin Frans blog. – URL: <http://kvfrans.com/simple-algorithms-for-solving-cartpole/> (accessed: 18.12.2019).
6. *Растрюгин Л.А.* Адаптация сложных систем. – Рига: Зинатне, 1981. – 375 с.
7. Поиск восхождением к вершине [Электронный ресурс] // Википедия: web-сайт. – 2019. – URL: https://ru.wikipedia.org/wiki/Поиск_восхождением_к_вершине (дата обращения: 18.12.2019).
8. Нейронные сети: обучение без учителя [Электронный ресурс] // Курс лекций по предмету «Основы проектирования систем с искусственным интеллектом» / сост. С.Л. Сотник. – Днепропетровск, 1997–1999. – Гл. 3.4. – URL: http://www.codenet.ru/progr/alg/ai/htm/gl3_4.php (дата обращения: 18.12.2019).
9. *Федосов Б.Т.* Управление неустойчивыми объектами. Обратный маятник [Электронный ресурс] // Model.Exponenta.Ru: сетевое сообщество. – 2009. – URL: http://model.exponenta.ru/bt/bt_152_Inv_Pend_control_1.htm (дата обращения: 18.12.2019).
10. *Сушков Ю.А.* Метод, алгоритм и программа случайного поиска. – Л.: ВНИИТрансМаш, 1969. – 43 с.
11. *Жиглявский А.А., Жилинскас А.Г.* Методы поиска глобального экстремума. – М.: Наука, 1991. – 248 с.
12. *Гилл Ф., Мюррей У., Райт М.* Практическая оптимизация. – М.: Мир, 1985. – 509 с.
13. *Гасников А.В.* Современные численные методы оптимизации. Метод универсального градиентного спуска: учебное пособие. – М.: МФТИ, 2019. – 267 с.
14. *Жадан В.Г.* Методы оптимизации: в 3 ч. – М.: МФТИ, 2015–2017. – 3 ч.

15. Николенко С., Кадуринов А., Архангельская Е. Глубокое обучение: погружение в мир нейронных сетей. – СПб.: Питер, 2018. – 480 с.

Шипагин Виктор Игоревич, аспирант кафедры автоматизации Новосибирского государственного технического университета. Основное направление научных исследований – нейронные сети. E-mail: shipagin@mail.com

DOI: 10.17212/2307-6879-2019-2-18-27

Different approaches to solving the inverted pendulum problem*

V.I. Shipagin

Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, postgraduate of the automation department. E-mail: shipagin@mail.com

Unlike the classical pendulum, the inverted one is in an unstable position. To stay upright, you need to balance it. Therefore, the solution of this problem can be used to test control algorithms (such as neural networks, genetic algorithms, fuzzy control algorithms, and PID-controllers).

In this paper, we compared the results of testing several control algorithms (random search algorithm, search algorithm with climbing to the top, algorithm of the gradient of policies using a neural network) on the example of solving the inverted pendulum problem according to the following criteria: training time; speed; memory used.

For the software implementation uses the Python language (v.3.6). The solution of the problem was carried out with the help of the free tool OpenAI Gym. This platform is designed to develop and train algorithms through games and other tests.

OpenAI Gym in a different environment for testing. I will use the environment "CartPole_v0", simulating the behavior of an inverted pendulum. Such a system at any time of the time interval can provide the coordinates of the position of the beam and the angle of deflection of the pendulum rod. Depending on the data provided, the algorithm must decide on the impact on the trolley in order to change it "to the left" or "to the right".

Keywords: neural networks, inverted pendulum, artificial intelligence, random search algorithm, activation function, learning

REFERENCES

1. OpenAI. *Wikipedia*: website. (In Russian). Available at: <https://ru.wikipedia.org/wiki/OpenAI> (accessed 18.12.2019).

* Received 02 April 2019.

2. Getting Started with Gym. *OpenAI*: website. Available at: <https://gym.openai.com/docs/> (accessed 18.12.2019).
3. Obratnyi mayatnik [Inverted pendulum]. *Wikipedia*: website. (In Russian). Available at: https://ru.wikipedia.org/wiki/Обратный_маятник (accessed 18.12.2019).
4. *OpenAI Universe. Otkrytaya platforma dlya trenirovki sil'nogo II* [OpenAI Universe. Open space for gym strong AI]. 2016. (In Russian). Available at: <https://geektimes.ru/post/283384/> (accessed 18.12.2019).
5. Frans K. Simple reinforcement learning methods to learn CartPole. *Kevin Frans blog*, 2016. Available at: <http://kvfrans.com/simple-algoritms-for-solving-cartpole/> (accessed 18.12.2019).
6. Rastrigin L.A. *Adaptatsiya slozhnykh sistem* [Adaptation of complex system]. Riga, Zinatne Publ., 1981. 375 p.
7. Poisk voskhozhdeniem k vershine [Search with climbing to the top]. *Wikipedia*: website. 2019. (In Russian). Available at: https://ru.wikipedia.org/wiki/Поиск_восхождением_к_вершине (accessed 18.12.2019).
8. Sotnik S.L., comp. Neironnye seti: obuchenie bez uchitelya [Neural network. Learning without teacher]. *Kurs lektsii po predmetu "Osnovy proektirovaniya sistem s iskusstvennym intellektom"* [Lecture course on the subject "Fundamentals of designing systems with artificial intelligence"]. Dneprodzerzhinsk, 1997–1999. (In Russian). Available at: http://www.codenet.ru/progr/alg/ai/htm/gl3_4.php (accessed 18.12.2019).
9. Fedosov B.T. *Upravlenie neustoiichivymi ob"ektami. Obratnyi mayatnik* [Control of unstable objects. Reverse pendulum]. *Model.Exponenta.Ru: network community*, 2009. (In Russian). Available at: http://model.exponenta.ru/bt/bt_152_Inv_Pend_control_1.htm (accessed 18.12.2019).
10. Sushkov Yu.A. *Metod, algoritm i programma sluchainogo poiska* [Method, algorithm and program of random search]. Leningrad, VNIITransMash Publ., 1969. 43 p.
11. Zhiglyavskii A.A., Zhilinskas A.G. *Metody poiska global'nogo ekstremuma* [Method of search global extremum]. Moscow, Nauka Publ., 1991.
12. Jill P.E., Murray W., Wright M.H. *Practical optimization*. London, Academic Press, 1981 (Russ. ed.: Gill F., Myurrei U., Rait M. *Prakticheskaya optimizatsiya*. Moscow, Mir Publ., 1985. 509 p.).
13. Gasnikov A.V. *Sovremennye chislennyye metody optimizatsii. Metod universal'nogo gradientnogo spuska* [Modern numerical methods of optimization. Universal method of gradient descent]. Moscow, MFTI Publ., 2019. 267 p.
14. Zhadan V.G. *Metody optimizatsii*. V 3 ch. [Methods of optimization. In 3 pt.]. Moscow, MFTI Publ., 2015–2017.

15. Nikolenko S., Kadurin A., Arkhangel'skaya E. *Glubokoe obuchenie: pogru-zhenie v mir neuronnykh setei* [Deep learning. Immersion in the world of neural network]. St. Petersburg, Piter Publ., 2018. 480 p.

Для цитирования:

Шипагин В.И. Различные подходы к решению задачи перевернутого маятника // Сборник научных трудов НГТУ. – 2019. – № 2 (95). – С. 18–27. – DOI: 10.17212/2307-6879-2019-2-18-27.

For citation:

Shipagin V.I. Razlichnye podkhody k resheniyu zadachi perevernutoy mayatnika [Different approaches to solving the inverted pendulum problem]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2019, no. 2 (95), pp. 18–27. DOI: 10.17212/2307-6879-2019-2-18-27.