ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНА ТЕХНІКА

https://doi.org/10.31649/1997-9266-2019-144-3-40-49

УДК 004.89 + 004.942

В. В. Півошенко[1]
М. С. Кулик[1]
Ю. Ю. Іванов[1]
А. С. Васюра[1]

АНАЛІЗ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МЕТОДУ БЕЗМОДЕЛЬНОГО НАВЧАННЯ З ПІДКРІПЛЕННЯМ

[1]Вінницький національний технічний університет

Розглянуто сучасний метод машинного навчання, який має назву навчання з підкріпленням. У задачах, які розв'язуються на основі взаємодії, найчастіше непрактично намагатися отримувати приклади необхідної поведінки інтелектуального програмного агента, які були б одночасно коректними та доречними для всіх ситуацій, оскільки наявні умови невизначеності, що виникають через неповноту інформації про навколишнє середовище та можливі дії інших ботів або людей. Тому програмний агент повинен навчатися на основі власного досвіду. Важливою перевагою навчання з підкріпленням є можливість навчання бота «з нуля» за рахунок збалансованого поєднання (пошук компромісу) режимів «дослідження» — «застосування» та вивчення стратегій, які дозволяють жертвувати малим на певному етапі заради отримання більшої вигоди в подальшому. Дослідження в області навчання з підкріпленням можна вважати частиною загального процесу, який розвивається в останні роки. Він складається зі взаємодії штучного інтелекту та інженерних дисциплін, тому саме у навчанні з підкріпленням розвиваються ідеї, взяті з теорії оптимального управління, стохастичної оптимізації та апроксимації, прагнучи реалізації загальніших і амбітних цілей штучного інтелекту.

Представлено математичний апарат навчання з підкріпленням із залученням методу безмодельного Q-навчання, показано практичні аспекти його застосування, а також розроблено ефективну стратегію навчання бота у штучному середовищі (комп'ютерній відеогрі). В ролі спостережуваних змінних об'єкта виступає інформація, яку використовує агент, а прихованими змінними є довгострокові оцінки отриманої ним вигоди. Залежно від поточного стану середовища і дій бота розраховується функція вигоди, яку отримає агент у наступний момент часу. З використанням розробленого програмного забезпечення виконано експериментальні дослідження розглянутого методу. У роботі отримано оптимальні параметри налаштування, криві та час навчання бота. Результати дослідження можуть бути корисними для комп'ютерних систем різного функціонального призначення, їх можна застосовувати у моделюванні та проектуванні, в системах автоматичного керування та прийняття рішень, робототехніці, на фондових ринках тощо.

Ключові слова: штучний інтелект, машинне навчання, навчання з підкріпленням, Q-навчання, стратегія навчання, інтелектуальний програмний агент, бот, оптимальні параметри, криві навчання, експериментальні дослідження.

Вступ

Ідея штучного інтелекту з'явилася в 1950-х роках, коли дослідники у галузі комп'ютерних наук почали цікавитись, чи можуть комп'ютери «думати» — це питання є актуальним і все ще інтенсивно досліджується. Штучний інтелект охоплює машинне навчання (*machine learning*), глибоке навчання (*deep learning*), навчання з підкріпленням (*reinforcement learning*), активне навчання (*active learning*), навчання з учителем (*supervised learning*) і без нього (*unsupervised learning*) (рис. 1) [1].

Проблематика та основні концепції машинного навчання пов'язані із питанням: чи може

комп'ютер самостійно навчитися виконувати певне завдання? Це питання відкриває двері в нову парадигму програмування. У класичному програмуванні парадигма символічного штучного інтелекту така: розробники задають правила (програму) та дані, які підлягають обробці відповідно до цих правил і отримують результат (рис. 2а). Використовуючи машинне навчання, розробники задають дані та результат, очікуваний від них, а отримують правила, які можуть застосовуватися до нових даних для отримання нового вихідного результату (рис. 2б).

Рис. 1. Структура підгалузей штучного інтелекту

Рис. 2. Парадигми програмування

Система машинного навчання навчається, а не явно запрограмована. Їй надається багато прикладів, які відносяться до задачі, а вона, в свою чергу, знаходить в цих прикладах закономірності. Це дозволяє системі розробити правила для автоматизації розв'язання завдання. Машинне навчання тісно пов'язане з математичною статистикою, але відрізняється від неї декількома важливими речами. Наприклад, воно дозволяє дослідити складні набори даних великого обсягу (мільйони зображень, що сформовані з десятків тисяч пікселів), для яких класичний статистичний аналіз, такий як байєсівський (*Bayesian inference*), буде непрактичним.

У цій статті розглядається навчання з підкріпленням, яке дозволяє навчити інтелектуального програмного агента діяти у певному штучному середовищі з високою ефективністю, яка оцінюється набраними балами.

Навчання з підкріпленням — це галузь машинного навчання, інспірована біхевіористською психологією. Вона контрастує з іншими галузями машинного навчання, оскільки алгоритм явно не говорить про те, як виконувати завдання, але сам по собі працює над ним [2].

У класичних системах навчання з підкріпленням агент B взаємодіє із середовищем через сприйняття та дії (рис. 3). На кожному кроці він в якості вхідного сигналу i отримує деяку індикацію поточного стану середовища s_t та вибирає дію a_t для генерації вихідного сигналу. Вона змінює стан середовища, а цінність переходу до стану повідомляється агенту за допомогою скалярного сигналу підсилення r_t (винагорода). Функція введення або тотожності (*indentity function*) I визначає як агент оглядає стан середовища. Стратегія агента повинна вибирати дії, які мають тенденцію збільшувати суму значень сигналу підсилення r_t. Агент може навчитися робити це шляхом систематичних спроб та помилок, керуючись заданим алгоритмом роботи [3].

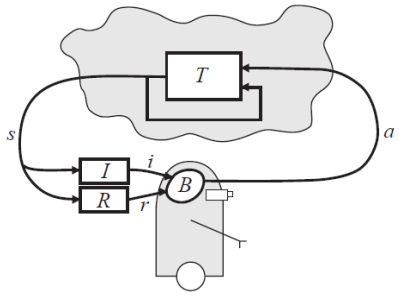

Рис. 3. Графічна інтерпретація навчання з підкріпленням

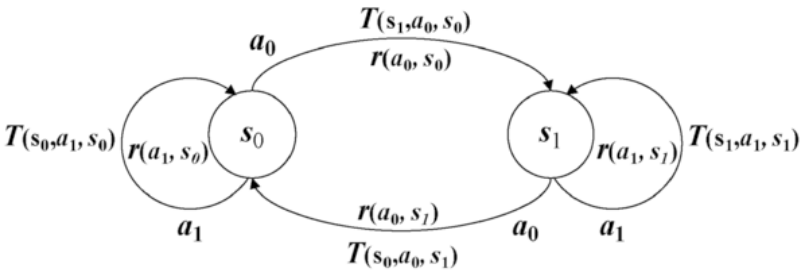

Рис. 4. Схема марковського процесу прийняття рішення

Формально навчання з підкріпленням моделюється як *марковський процес прийняття рішення* [4], [5] і складається з (рис. 4)

– набору станів середовища $s_t \in S$;

– набору дій $a_t \in A$;

– функції винагороди (*reward function*) $R(s,a)$, а r — миттєва винагорода. Вона визначає очікувану миттєву винагороду як функцію поточного стану і дії;

– функції переходу до стану (*state transition function*) $T(s,a,s')$. Вона визначає наступний стан середовища s' як функцію від поточного стану s і дії агента a [6].

Завдання агента: знайти *стратегію* (*policy*) π у відповідності станів та дій, яка максимізує довгострокову міру підсилення. В загальному очікується, що середовище буде недетермінованним, тобто виконання однієї і тієї ж дії в одному і тому ж стані у двох різних випадках може призвести до різних станів або різних значень винагороди. Однак середовище повинно бути стаціонарним, тобто ймовірність виникнення переходів до стану або отримання конкретних сигналів підсилення не зміниться з часом [7].

Навчання з підкріпленням відрізняється від більш вивченої проблеми навчання з вчителем декількома аспектами. Найважливіша відмінність полягає в тому, що не існує уявлення про пари вхідних/вихідних сигналів. Замість них, після вибору дії, агенту повідомляється поточна винагорода та подальший стан, але не вказується, які дії були б кращими для нього. Агент повинен швидко набувати корисного досвіду про можливі стани системи, дії та винагороди, щоб діяти оптимально. Навчання з підкріпленням, у першу чергу, зосереджує свою увагу на тому, як отримати оптимальну стратегію [8]. Методи визначення оптимальної стратегії з урахуванням моделей оптимальної поведінки визначають, як агент повинен враховувати майбутнє у рішеннях, які він приймає, щоб вирішити, як поводитися зараз [9]. Найвідомішими є моделі скінченного горизонту (*finite horizon model*), нескінченного горизонту зі знеціненням (*infinite-horizon discounted model*) та середньої винагороди (*average-reward model*). У цій роботі застосовано другу модель, оскільки в ній враховується довгострокова винагорода агента, але винагороди, отримані в майбутньому, геометрично знецінюються згідно з коефіцієнтом знецінення (*discount factor*) γ $(0 \le \gamma \le 1)$ [2], [8].

Метою роботи є пошук оптимальних параметрів, часу та кривих навчання бота, що дозволить дослідити ефективність методу навчання з підкріпленням у штучному середовищі — грі.

Розв'язання задачі

Для оцінювання оптимальної стратегії використаємо функцію оптимальної цінності (*optimal value function*) $V(s)$. Вона є очікуваною знеціненою сумою винагороди, яку агент отримає, якщо почне діяти в певному стані, застосовуючи оптимальну стратегію. Її можна визначити за допомогою ітераційного процесу, який називається *ітерацією за цінністю* (*value iteration*) [5], [8]

$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')V(s') \right), \ \forall s \in S. \quad (1)$$

Обчислювальна складність алгоритму оцінювання оптимальної стратегії за ітераціями є квадратичною за кількістю станів і лінійної за кількістю дій [10].

Після того як оптимальна стратегія оцінена її потрібно поліпшити, для цього застосуємо процес *ітерації за стратегією* (*policy iteration*), який є циклом між оцінкою стратегії та її поліпшенням [5], [9]. Тобто, як тільки дізнаємося цінність кожного стану в рамках поточної стратегії, то розглядаємо варіант підвищення цінності, змінюючи першу дію. Цей процес гарантує підвищення ефективності стратегії. Якщо ніяких поліпшень не буде, тоді стратегія є оптимальною. З огляду на функцію оптимальної цінності визначаємо оптимальну стратегію як

$$\pi(s) = \operatorname{argmax}_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')V(s') \right). \quad (2)$$

Оскільки існує не більше $|A|^{|S|}$ різних стратегій, а їхня послідовність поліпшується на кожному кроці, то цей алгоритм описується експоненціальною кількістю ітерацій [4], [10].

У роботі застосуємо алгоритм знаходження оптимальної стратегії:

1. *Ініціалізація масивів*:
   $V(s) \in \mathbb{R}$; $\pi(s) \in A(s)$, довільно, наприклад $V(s) = 0$.
2. *Оцінка стратегії*:
   Повторювати

$\Delta \leftarrow 0$

Для кожного $s \in S$:

$v \leftarrow V(s)$

$V(s) \leftarrow \max_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V(s') \right)$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

Доки $\Delta < \theta$ (невелике позитивне число).

3. *Покращення стратегії*:

policy stable ← *true*

Для кожного $s \in S$:

$a \leftarrow \pi(s)$

$\pi(s) = \text{argmax}_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V(s') \right)$.

Якщо $a \neq \pi(s)$, тоді *policy stable* ← *false*

Якщо *policy stable* ← *true*, тоді зупинити та повернути *V* та π, інакше перейти на крок 2.

Зважаючи на описану математичну основу використаємо метод *безмодельного навчання* або *Q-навчання* (*Q-learning*) [10]—[12], коли модель, яка складається із функцій переходу до стану $T(s,a,s')$ та винагороди $R(s,a)$, невідома заздалегідь [13].

Нехай $Q(s,a)$ — очікуване знецінене підкріплення дії *a* у стані *s*, $V(s)$ — це значення *s*, яке передбачає, що найкраща дія здійснюється спочатку. Тоді формулу (1) запишемо у вигляді

$$V(s) = \max_a Q(s,a). \tag{3}$$

Отже, $Q(s,a)$ представимо рекурсивно

$$Q(s,a) = R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \max_a Q(s',a'). \tag{4}$$

Зауважимо також, оскільки $V(s) = \max_a Q(s,a)$, то $\pi(s) = \text{argmax}_a Q(s,a)$ — оптимальна стратегія.

Оскільки *Q*-функція (*Q-function*) робить дію явною, то оцінимо *Q*-значення (*Q-values*), використовуючи метод *TD*(0) [11], [13], а також визначимо стратегію (дія може бути вибрана простим відбором максимального *Q*-значення для поточного стану). Значення стратегії обчислюється із використанням методу *TD*(0), який є екземпляром більш загального класу методів *TD*(λ) [14]. Правило оновлення значень функції цінності запишемо таким чином:

$$V(s) = V(s) + \alpha \left(r + \gamma V(s') - V(s) \right), \tag{5}$$

тобто кожного разу, коли настає стан *s*, його оцінка оновлюється, щоб бути ближчою до значення $r + \gamma V(s')$. *TD*(0) проглядає простір тільки на один крок вперед при коригуванні оцінок цінностей. Якщо швидкість навчання коригується належним чином (вона повинна бути повільно зменшена) і стратегія незмінна, то *TD*(0) гарантовано буде збігатися до функції оптимальної цінності, але для цього може знадобитися досить багато часу [15].

Загальне правило методу *TD*(λ) подібне формулі (5), але воно застосовується до кожного стану згідно зі значенням *e*(*s*), тобто [16]

$$V(s) = V(s) + \alpha \left(r + \gamma V(s') - V(s) \right) e(s); \tag{6}$$

$$e(s) = \sum_{k=1}^{t} (\lambda\gamma)^{t-k} \delta_{s,s_k}; \quad \delta_{s,s_k} = \begin{cases} 1, s = s_k, \\ 0, s \neq s_k. \end{cases} \tag{7}$$

Обчислювально складніше використовувати метод *TD*(λ), хоча він часто працює значно швидше для великих λ [16].

Отже, врахувавши написане вище, сформулюємо правило для *Q*-навчання

$$Q(s,a) = (1-\alpha) Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') \right). \tag{8}$$

де $\{s, a, r, s'\}$ — набір досвіду; $0 < \alpha \le 1$ — темп (швидкість) навчання; γ — коефіцієнт знецінення.

Тоді запишемо псевдокод алгоритму *Q-learning*, який застосовано під час експериментального дослідження:

1. *Необхідно*:
   набір станів $S = \{1, \ldots, s_t\}$
   набір дій $A = \{1, \ldots, a_t\}$
   функція винагороди $R(s, a)$
   функція переходу до стану $T(s', a, s)$
   темп навчання $\alpha \in [0,\ 1]$
   коефіцієнт знецінення $\gamma \in [0,\ 1]$
2. *Функція Q-learning* (*S, A, R, T, α, γ*):
   Ініціалізувати Q довільно
   Доки Q не збігається, виконувати
   Початок в стані $s \in S$
   Доки стан s не кінцевий, виконувати
   Обчислити $\pi(s) \leftarrow \text{argmax}_a Q(s, a)$
   $a \leftarrow \pi(s)$
   $r \leftarrow R(s, a)$
   $s' \leftarrow T(s, a)$
   $Q(s, a) = (1 - \alpha) Q(s, a) + \alpha\left(r + \gamma \max_{a'} Q(s', a')\right)$
   $s \leftarrow s'$
   Повернути *Q*.

Якщо кожна дія виконується в кожному стані нескінченну кількість разів на нескінченному прогоні, то *Q*-значення будуть збігатися до оптимальних *Q* із імовірністю $p = 1$. Коли *Q*-значення майже сходяться до оптимальних значень, для агента доцільно діяти з «жадібністю» (*greedy policy*), приймаючи в кожній ситуації дію з найвищим значенням *Q*. Слід зазначити, що *Q*-навчання не залежить від «розвідки» (*exploration*). Це означає, що *Q*-значення будуть збігатись до оптимальних значень незалежно від того, як поводить себе агент під час отримання винагороди. З цих причин *Q*-навчання є найпопулярнішим і найефективнішим методом безмодельного навчання. Однак, він може досить повільно наближатися до оптимальної стратегії [17], [18].

Експериментальне дослідження

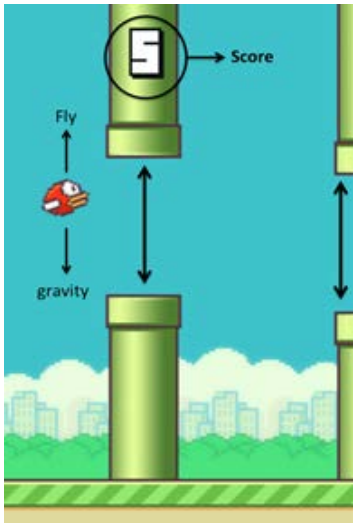

Рис. 5. Процес навчання бота у штучному середовищі

Середовище навчання. Дослідження роботи методу *Q-learning* з розробленими правилами навчання проводилося у штучному середовищі, яке представляло собою комп'ютерну 2D гру «*Flappy Bird*» з незмінним геймплеєм (рис. 5). Мета гри полягає в управлінні польотом пташки, яка безперервно пересувається між рядами зелених труб (перешкод) на координатній сітці. Гравець може виконувати тільки одну дію — ривок пташки (*fly*) на певну величину. За відсутності ривків або зіткненні з перешкодами пташка падає (*fall by gravity*), а гра відповідно вважається програною. Бали (*scores*) набираються з кожним успішним проходженням перешкоди.

Модель навчання. Розроблений для навчання бот вибирає: робити ривок у середовищі або ні. Його швидкість *v* змінюється з кроком 1 від –10 до 10 умовних одиниць. У якості стану використовувалась відстань від пташки до нижньої перешкоди на координатній сітці (*x, y*), а також її швидкість (*v*) за віссю ординат. *Q*-матриця представляла собою словник з ключем у вигляді (*x, y, v*), елементами якого є масиви з вагами можливих дій бота. За кожну правильну дію він отримував винагороду (+1 пункт за крок гри за умови живої

пташки, +10 пунктів за успішне подолання перешкоди), а за помилку (смерть пташки) — покарання на 1000 пунктів. Прийняття рішення ботом відбувається 60 разів у секунду, оскільки гра розраховує 60 фреймів у секунду. Мета бота — набрати найбільшу кількість балів у заданому штучному середовищі.

Стратегія навчання. У ході розробки бота застосовано *жадібну стратегію Q-навчання*, яка полягає в тому, що вибрано дію з найкращим *Q*. Причина її вибору полягає в тому, що без використання жадібної стратегії у більшості випадків агент не проходив навіть першу пару перешкод.

Експеримент 1. Побудувати криві навчання та знайти оптимальні параметри методу *Q*-навчання (коефіцієнти знецінення γ та темпу навчання α), завдяки яким агент буде отримувати високі бали у середовищі.

Для знаходження значень параметрів бот навчався у середовищі впродовж 30 епох по 5000 ітерацій, потім параметри методу $\gamma \in [0,0;\ 0,2;\ 0,5;\ 0,8;\ 1,0]$ та $\alpha \in [0,0;\ 0,2;\ 0,5;\ 0,7;\ 1,0]$ змінювалися, а процес повторювався. Результати навчання агента (набрані бали) аналізуються на останніх 100 ітераціях навчання, оскільки на них процес стабілізується (табл. 1).

Таблиця 1

Діапазон усереднених результатів навчання бота на останніх 100 ітераціях

γ \ α	0,0	0,2	0,5	0,7	1,0
1,0	[0; 0]	[21; 71]	[126; 422]	[165; 480]	[0; 1]
0,8	[0; 0]	[26; 78]	[26; 59]	[15; 43]	[6; 15]
0,5	[0; 0]	[6; 18]	[6; 25]	[5; 15]	[3; 10]
0,2	[0;0]	[0; 3]	[1; 3]	[1; 3]	[1; 4]
0,0	[0; 0]	[0; 0]	[0; 0]	[0; 0]	[0; 0]

Як випливає з табл. 1, оптимальною парою значень параметрів є $[\gamma;\alpha] = [1,0;\ 0,7]$, оскільки з ними бот отримує великі бали за дії у середовищі-грі. Також можна помітити, що зі збільшенням коефіцієнта знецінення γ, якщо темп навчання знаходиться у проміжку $\alpha \in [0,5;\ 0,7]$, результат навчання поліпшується.

На рис. 6а, б червоними крапками зображено отримані ботом бали за проходження гри, а синім кольором показано криву його навчання, яка наочно демонструє крапки — «викиди». Це говорить про те, що бот продовжує навчатись (отримані бали не зростають рівномірно), але існують шляхи поліпшення моделі та стратегії його навчання. На рис. 6в, г криві навчання бота зображено у збільшеному вигляді.

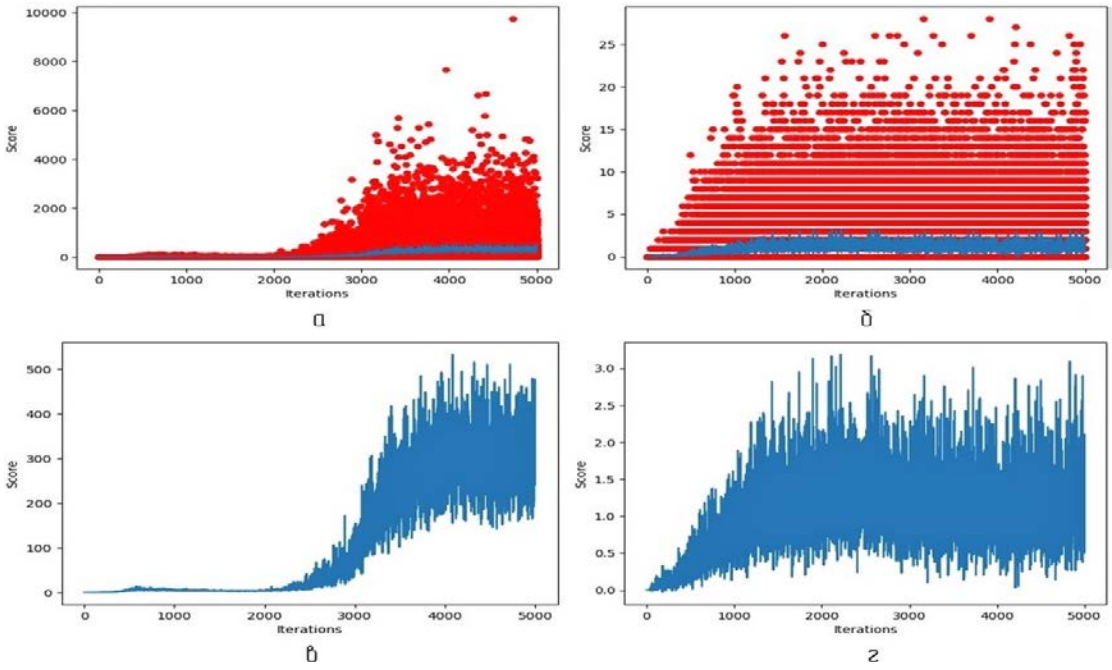

Рис. 6: а, б — отримані бали; в, г — криві навчання для параметрів $[\gamma;\alpha] = [1,0;\ 0,7]$ та $[\gamma;\alpha] = [0,2;\ 0,2]$

Також під час експерименту виявлено аномальну пару параметрів $[\gamma;\alpha]=[1,0;1,0]$. Саме за її використання агент починає активно набирати бали на початкових ітераціях, після чого результат стрімко падає (рис. 7).

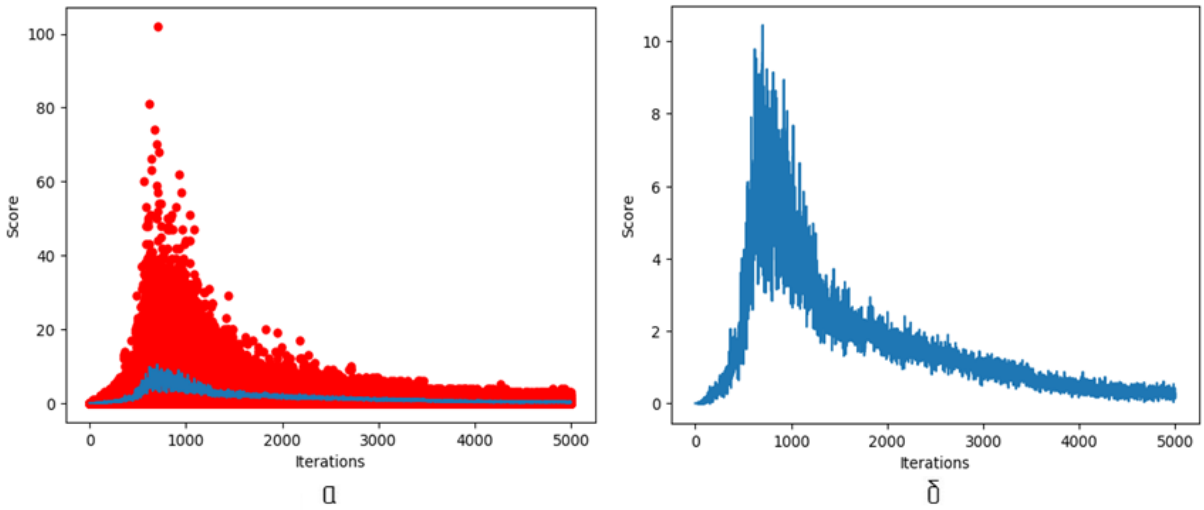

Рис. 7: а — отримані бали; б — крива навчання для параметрів $[\gamma;\alpha]=[1,0;1,0]$

Експеримент 2. Дослідження швидкодії l методу Q-навчання залежно від значень коефіцієнтів знецінення γ та темпу навчання α.

Отримані результати в числовому вигляді наведено в табл. 2 та 3. Експериментальні залежності $l = f(\alpha,\gamma)$ показано на рис. 8.

Таблиця 2

Загальний час *t* (у секундах) навчання бота (30 епох по 5000 ітерацій)

γ \ α	0,0	0,2	0,5	0,7	1,0
1,0	88,757	1178,597	5277,394	8313,158	317,805
0,8	91,619	954,167	1322,496	1027,943	542,778
0,5	90,810	637,078	660,601	619,258	574,978
0,2	93,228	315,397	315,168	337,673	446,378
0,0	94,976	216,502	218,726	216,881	216,805

Таблиця 3

Середня тривалість $t_{сер}$ (у секундах) навчання бота

γ \ α	0,0	0,2	0,5	0,7	1,0
1,0	2,958	39,287	175,913	277,105	10,594
0,8	3,053	31,806	44,083	34,265	18,093
0,5	3,027	21,236	22,020	20,642	19,166
0,2	3,107	10,513	10,505	11,256	14,879
0,0	3,165	7,216	7,290	7,229	7,226

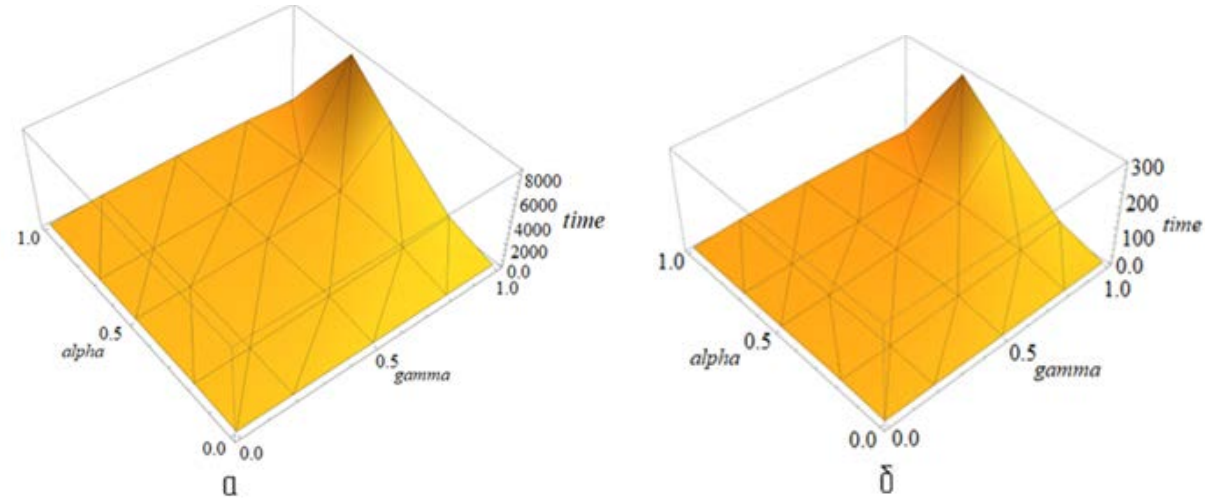

Рис. 8. Графічна інтерпретація: а — загальної; б — середньої тривалості навчання бота

Отримані тривимірні поверхні зміни тривалості навчання бота за варіації параметрів (рис. 8) показують, що час навчання збільшується з наближенням значень α та γ до оптимальних.

Висновки

Проаналізовано математичну основу навчання з підкріпленням. Проведено експериментальні дослідження методу *Q-learning* з використанням розробленого бота у штучному середовищі — грі «*Flappy Bird*». Побудовано криві навчання та визначено, що оптимальними значеннями коефіцієнтів знецінення γ та темпу навчання α. для методу є $[\gamma;\alpha]=[1,0;0,7]$, оскільки з ними бот отримує найбільші бали. Дослідження швидкодії методу показало, що тривалість навчання збільшується з наближенням значень α та γ до оптимальних. Наприклад, для параметрів $[\gamma;\alpha]=[1,0;0,7]$ загальний час навчання склав приблизно 2 год 19 хв (у середньому 4,6 хв на епоху). Результати цих досліджень є внеском у подальший розвиток методів машинного навчання. На практиці їх можна застосовувати у моделюванні і проектуванні, в системах автоматичного керування та прийняття рішень, робототехніці, на фондових ринках тощо.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] O. Hernández-Lerma, J. Hennet, and J. Lasserre, “Average Cost Markov Decision Processes: Optimality conditions,” *Journal of Mathematical Analysis and Applications*, vol. 158, no. 2, pp. 396-406, 1991.

[2] R. Bellman, “A Markovian Decision Process,” *Indiana University Mathematics Journal*, vol. 6, no. 4, pp. 679-684, 1957.

[3] L. Busoniu, R. Babuska, B. Schutter, and D. Ernst, “Reinforcement Learning and Dynamic Programming Using Function Approximators,” *Automation and Control Engineering*, pp. 55-88, 2010.

[4] А. С. Васюра, Т. Б. Мартинюк, та Л. М. Куперштейн, *Методи та засоби нейроподібної обробки даних для систем керування.* Вінниця, Україна: Універсум-Вінниця, 2008.

[5] C. J. C. H. Watkins, and P. Dayan, *Reinforcement Learning, Technical Note*, 1992, pp. 55-68.

[6] F. Chollet, *Deep learning with Python*. Shelter Island. NY: Manning Publications Co., 2018, pp. 27-38.

[7] J. Gläscher, N. Daw, P. Dayan, and J. P. O’doherty, “States versus Rewards: Dissociable Neural Prediction Error Signals Underlying Model-Based and Model-Free Reinforcement Learning,” *Neuron*, vol. 66, no. 4, pp. 585-595, 2010.

[8] R. S. Sutton, and A. G. Barto, *Reinforcement learning: an introduction*. Cambridge: The MIT Press, 2015, pp. 143-160.

[9] Т. М. Боровська, А. С. Васюра, та В. А. Северілов, *Моделювання та оптимізація систем автоматичного управління.* Вінниця, Україна: ВНТУ, 2009.

[10] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. Jordan, "Is Q-learning Provably Efficient?", *arXiv.org*, 2018. [Electronic resource]. Available: https://arxiv.org/pdf/1807.03765.pdf . Accessed: Jul. 10, 2018.

[11] J. Dornheim, N. Link, and P. Gumbsch, “Model-Free Adaptive Optimal Control of Sequential Manufacturing Processes Using Reinforcement Learning,” *arXiv.org*, 2019. [Electronic resource]. Available: https://arxiv.org/abs/1809.06646v1 . Accessed: Jan. 07. 2019.

[12] W. Haskell, and W. Huang, "Stochastic Approximation for Risk-Aware Markov Decision Processes", *Arxiv.org*, 2018. [Electronic resource]. Available: https://arxiv.org/pdf/1805.04238.pdf. Accessed: May. 17, 2018.

[13] R. Bellman, “Dynamic programming and stochastic control processes,” *Information and Control*, vol. 1, no. 3, pp. 228-239, 1958.

[14] C. J. C. H. Watkins, *Learning from delayed rewards*. University of Cambridge, 1989, pp. 55-68.

[15] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “An Introduction to Reinforcement Learning,” *The Biology and Technology of Intelligent Autonomous Agents*, 1995, pp. 90–127.

[16] M. Rahman and H. Rashid, “Implementation of Q Learning and Deep Q Network for Controlling a Self-Balancing Robot Model,” *ArXiv.org*, 2018. [Electronic resource]. Available: https://arxiv.org/pdf/1807.08272.pdf . Accessed: Jul. 22, 2018.

[17] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, no. 3-4, pp. 279-292, 1992.

[18] E. Even-Dar and Y. Mansour, “Learning Rates for Q-Learning,” *Lecture Notes in Computer Science Computational Learning Theory*, 2001, pp. 589–604.

Рекомендована кафедрою автоматизації та інтелектуальних інформаційних технологій ВНТУ

Стаття надійшла до редакції 7.05.2019

Півошенко Володимир Володимирович — студент факультету комп’ютерних систем і автоматики, e-mail: volodymyr.pivoshenko@gmail.com ;

Кулик Максим Сергійович — студент факультету комп’ютерних систем і автоматики, e-mail: mxx888777@gmail.com ;

Іванов Юрій Юрійович — канд. техн. наук, старший викладач кафедри автоматизації та інтелектуальних інформаційних технологій, e-mail: Yura881990@i.ua ;

Васюра Анатолій Степанович — канд. техн. наук, професор, професор кафедри автоматизації та інтелектуальних інформаційних технологій, e-mail: vasanat@i.ua .

Вінницький національний технічний університет, Вінниця

V. V. Pivoshenko[1]
M. S. Kulyk[1]
Yu. Yu. Ivanov[1]
A. S. Vasiura[1]

Analysis and Experimental Research of Model-Free Reinforcement Learning Method

[1]Vinnytsia National Technical University

In this article there has been considered a modern method of machine learning, which is called reinforcement learning. In tasks, that are solved based on interaction, is often impractical to try to get the desired behavior examples of an intellectual software agent, that would be both correct and appropriate for all situations, since the uncertainty conditions exist, arising from incomplete information about an environment and possible actions of other bots or humans. Therefore, the software agent should be trained on the basis of its own experience. An important advantage of the reinforcement learning is the possibility of learning a bot «from scratch» by the balanced combination (search of the compromise) of the «exploration» «exploitation» modes and learning of the strategies, which allow to sacrifice some scores at this stage for the sake of greater benefit in the future.Researches in the field of the reinforcement learning can be considered as a part of the overall process, that developed over a last few years. It consists of an interaction of an artificial intelligence and other engineering disciplines that is why reinforcement learning develops ideas drawn from the optimal control theory, stochastic optimization and approximation, following common and ambitious goals of the artificial intelligence.

In this work there has been presented the mathematical apparatus of reinforcement learning with the usage of the model-free Q-learning method, practical aspects of its application have been shown, also an effective strategy for the bot learning in an artificial environment (computer video game) has been developed. The role of the observed object variables is accepted by the information used by the agent, and the hidden variables are long-term estimates of the benefit it gainsDepending on the current status of the environment and bot activities is calculated the benefit function, which is received by the agent at the next time moment. With the usage of the developed software, experimental researches of the considered method have been performed. The optimal setting parameters, curves and time learning of the bot have been obtained.The research results may be useful for computer systems of various functional purposes; they can be used in modeling and design, in automatic control and decision making systems, in robotics, in stock markets, etc.

Keywords: artificial intelligence, machine learning, reinforcement learning, Q-learning, learning strategy, intellectual software agent, bot, optimal parameters, learning curves, experimental researches.

Pivoshenko Volodymyr V. — Student of the Department of Computer Systems and Automation, e-mail: volodymyr.pivoshenko@gmail.com ;

Kulyk Maksym S. — Student of the Department of Computer Systems and Automation, e-mail: mxx888777@gmail.com ;

Ivanov Yurii Yu. — Cand. Sc. (Eng.), Senior Lecturer of the Chair of Automatization and Intellectual Information Technologies, e-mail: Yura881990@i.ua ;

Vasiura Anatolii S. — Cand. Sc. (Eng.), Professor, Professor of the Chair of Automatization and Intellectual Information Technologies, e-mail: vasanat@i.ua

В. В. Пивошенко[1]
М. С. Кулик[1]
Ю. Ю. Иванов[1]
А. С. Васюра[1]

Анализ и экспериментальное исследование метода безмодельного обучения с подкреплением

[1]Винницкий национальный технический университет

Рассмотрен современный метод машинного обучения, названый «обучение с подкреплением». В задачах, которые решаются на основе взаимодействия, чаще всего непрактично пытаться получать примеры необходимого поведения интеллектуального программного агента, которые были бы одновременно корректными и уместными для всех ситуаций, поскольку существуют условия неопределенности, возникающие из-за неполноты информации об окружающей среде и возможных действиях других ботов или людей. Поэтому программный

агент должен учиться на основе собственного опыта. Важным преимуществом обучения с подкреплением является возможность обучения бота «с нуля» за счет сбалансированного сочетания (поиск компромисса) режимов «исследование» — «применение» и изучения стратегий, которые позволяют на определенном этапе жертвовать малым ради получения большей выгоды в дальнейшем. Исследования в области обучения с подкреплением можно считать частью общего процесса, который развивается в последние годы. Он состоит из взаимодействия искусственного интеллекта и других инженерных дисциплин, поэтому именно в обучении с подкреплением развиваются идеи, взятые из теории оптимального управления, стохастической оптимизации и аппроксимации, стремясь к реализации более общих и амбициозных целей искусственного интеллекта.

Представлен математический аппарат обучения с подкреплением с применением метода безмодельного Q-обучения, показаны практические аспекты его применения, а также разработана эффективная стратегия обучения бота в искусственной среде (компьютерной видеоигре). В качестве наблюдаемых переменных объекта выступает информация, которую использует агент, а скрытыми переменными являются долгосрочные оценки полученной им выгоды. В зависимости от текущего состояния среды и действий бота рассчитывается функция выгоды, которую получит агент в следующий момент времени. С использованием разработанного программного обеспечения выполнены экспериментальные исследования рассматриваемого метода. В работе получены оптимальные параметры настройки, кривые и время обучения бота. Результаты исследования могут быть полезными для компьютерных систем разного функционального назначения, их можно применять в моделировании и проектировании, в системах автоматического управления и принятия решений, робототехнике, на фондовых рынках.

Ключевые слова: искусственный интеллект, машинное обучение, обучение с подкреплением, Q-обучение, стратегия обучения, интеллектуальный программный агент, бот, оптимальные параметры, кривые обучения, экспериментальные исследования.

Пивошенко Владимир Владимирович — студент факультета компьютерных систем и автоматики, e-mail: volodymyr.pivoshenko@gmail.com ;

Кулик Максим Сергеевич — студент, факультета компьютерных систем и автоматики, e-mail: mxx888777@gmail.com ;

Иванов Юрий Юриевич — канд. техн. наук, старший преподаватель кафедры автоматизации и интеллектуальных информационных технологий, e-mail: Yura881990@i.ua ;

Васюра Анатолій Степанович — канд. техн. наук, профессор, профессор кафедры автоматизации и интеллектуальных информационных технологий, e-mail: vasanat@i.ua