УДК 68.13 DOI: 10.20998/2078-9130.2024.2.315342

М. В. НЕКРАСОВА

МЕТОД МОНТЕ-КАРЛО ТА ШТУЧНИЙ ІНТЕЛЕКТ: ВИКОРИСТАННЯ МЕТОДУ МОНТЕ-КАРЛО В НАВЧАННІ З ПІДКРІПЛЕННЯМ

Навчання з підкріпленням - технологія, що найбільш швидко розвивається, застосовується при створенні штучних інтелектуальних систем. На даний момент ця галузь досить швидко розвивається і є надзвичайно затребуваною.. Багато дослідників по всьому світу активно працюють з навчанням з підкріпленням у різноманітних сферах: нейробіології, теорії управління, психології та багатьох інших. Метою даної роботи є обґрунтування можливості застосування методу Монте-Карло в навчанні з підкріпленням. Відомо, що основним у такому навчанні є фіксація аспектів реальної проблеми при взаємодії того, хто навчається з навколишнім світом для досягнення своєї мети. Тобто агент навчання повинен мати мету, пов’язану зі станом навколишнього середовища. Також необхідно мати можливість відчувати середовище та вчиняти дії, що впливають на нього. Формулювання завдання навчання з підкріпленням має враховувати все три аспекти – відчуття, дію та мету – у їх найпростіших формах. В статті показано, що методи Монте-Карло здатні вирішити проблеми навчання із підкріпленням, ґрунтуючись на усередненні результатів вибірки. Не можна використовувати лише перевірені дії або лише шукати нові - в цьому і полягає проблема, бо у стохастичній задачі кожна дія має бути випробувана багато разів, щоб отримати надійну оцінку очікуваної винагороди. Щоб забезпечити доступність чітко визначених результатів, у статті розглядаються методи Монте-Карло лише епізодичних завдань. При цьому показане застосування нестандартного підходу до навчання із заздалегідь невідомими навчальними прикладами, які підбиралися автоматично, у процесі оптимізації. Таким чином, методи Монте-Карло можуть бути успішно інкрементними лише на рівні епізодів.

Ключові слова: навчання з підкріпленням, метод Монте-Карло, стани, модель, прийняття рішень, оптимальність, стратегія, вибірка, цінність, оцінка.

Reinforcement learning is the fastest growing technology used in the creation of artificial intelligence systems. At the moment, this field is developing quite rapidly and is extremely in demand. Many researchers around the world are actively working with reinforcement learning in various fields: neurobiology, control theory, psychology and many others. The purpose of this work is to substantiate the possibility of using the Monte Carlo method in reinforcement learning. It is known that the main thing in such learning is to fix aspects of a real problem during the interaction of the learner with the surrounding world to achieve his goal. That is, the learning agent must have a goal related to the state of the environment. It is also necessary to be able to feel the environment and take actions that affect it. The formulation of the reinforcement learning task should take into account all three aspects - sensation, action and goal - in their simplest forms. The article shows that Monte Carlo methods are able to solve reinforcement learning problems based on averaging the results of the sample. It is not possible to use only proven actions or only search for new ones - this is the problem, because in a stochastic problem each action must be tried many times to get a reliable estimate of the expected reward. To ensure the availability of well-defined results, the article considers Monte Carlo methods only for episodic tasks. In doing so, it shows the use of a non-standard approach to training with previously unknown training examples that were selected automatically during the optimization process. Thus, Monte Carlo methods can be successfully incremental only at the episode level.

Key words: reinforcement learning, Monte Carlo method, states, model, decision making, optimality, strategy, sampling, value, evaluation

Вступ. Якщо задуматися про те, як людина навчається, то, швидше за все, першим, що спадає на думку, буде ідея, що цей процес відбувається при взаємодії людини з довкіллям. Контакти з навколишнім середовищем, поза всякими сумнівами, є джерелом знань як і про саме людину, так і про навколишнє середовище. Причому цей процес триває протягом всього життя людини. Взаємодія з навколишнім середовищем дає повну інформацію про зв'язок причин і наслідків, про послідовність дій, які потрібно виконати, щоб досягти певних цілей. Саме навчання через взаємодію є тією основною ідеєю, на якій базуються майже всі теорії навчання та інтелекту.

Вивчення того, як зіставити ситуації з діями, щоб отримати максимальну вигоду, називається навчанням із підкріпленням. Той, хто навчається, не знає заздалегідь, які дії необхідно виконати, щоб максимізувати винагороду. Тому він повинен самостійно з'ясувати, які дії потрібно для цього зробити. До того ж, потрібно усвідомлювати, що вибір дії впливає не тільки на винагороду на даному етапі, але і на те, яку вигоду агент навчання хоче отримати надалі. Ці дві характеристики – пошук методом проб та помилок та відкладена винагорода – є двома найбільш важливими відмінними характеристиками навчання з підкріпленням.

Головне при навчанні з підкріпленням - це зафіксувати основні аспекти реальної проблеми при взаємодії учня з навколишнім світом задля досягнення своєї мети. Тобто агент навчання повинен мати мету, пов'язану зі станом довкілля. Також учень повинен мати можливість відчувати середовище і вчиняти дії, що впливають на нього.

Формулювання завдання навчання з підкріпленням має враховувати всі три аспекти – відчуття, дію та мету – у їх найбільш простих формах.

Агент навчання, щоб отримати найбільшу винагороду, як правило, надає перевагу вже перевіреним діям, тобто тим, які виявилися ефективними для нього у минулому і дали йому найкращу винагороду. Але він не зможе знайти такі дії, якщо раніше не мав подібного досвіду. Через це, однією з проблем, що виникає при навчанні з підкріпленням, є компроміс між вивченням та застосуванням. Виходить, що учень має використовувати те, що він вже випробував, щоб отримати винагороду, але він також повинен вивчити нове, щоб зробити найкращий вибір дій у майбутньому [1]. Агент навчання завжди повинен пробувати різні дії, надалі надаючи перевагу тим, які виявилися найкращими. Не можна використовувати лише перевірені дії або лише шукати нові - в цьому і полягає проблема.

У стохастичній задачі кожна дія має бути випробувана багато разів, щоб отримати надійну оцінку очікуваної винагороди. Дилема «вивчення – застосування» інтенсивно вивчається математиками

протягом багатьох десятиліть, але досі залишається невирішеною.

Зазвичай виділяють ще чотири основні елементи крім агента навчання та середовища, що є складовими системи навчання з підкріпленням: 1. Стратегія. 2. Винагорода. 3. Цінність стану. 4. Модель середовища (необов'язково).

Зіставлення станів навколишнього середовища з діями, які мають бути виконані у цих станах, називається стратегією. Простіше кажучи, саме стратегія визначає те, який спосіб поведінки вибере агент навчання у конкретний час. У загальному випадку стратегія може бути випадковою. У деяких випадках вона виражається функцією або таблицею, у складніших варіантах – може навіть включати будь-які обчислення. Як правило, стратегії цілком достатньо для визначення поведінки агента навчання, вона є ядром агента.

Мета та постановка задачі. Метою завдання навчання з підкріпленням є винагорода - число, яке отримує той, хто навчається на кожному кроці середовища. Винагорода несе як позитивний зміст, так і негативний для агента навчання, тому що його основна мета – це максимізація загальної винагороди, яку агент планує отримати у довгостроковій перспективі. Якщо провести аналогію з біологією, то винагороду можна порівняти з досвідом болю чи задоволення. Якщо агент навчання отримав низьку винагороду після дії, обраної якоюсь стратегією, це може бути основою зміни стратегії у майбутньому. У загальному випадку винагороди можуть бути випадковими. Цінність стану – це загальна сума винагороди, яку агент навчання може отримати у майбутньому, починаючи з цього стану. Як видно з визначення, основною відмінністю цінності від винагороди є те, що вона визначає те, що добре у подальшій перспективі. Тобто конкретний стан може давати невелика миттєва винагорода, але за ним можуть йти стани, які приносять високу вигоду, а отже, такий стан матиме високу цінність. Зворотна ситуація так само може бути правдою. Якщо провести людську аналогію, то цінності відповідають тому, наскільки той, хто навчається, задоволений або незадоволений тим, що його оточення перебуває у конкретному стані, тоді як винагороди у чомусь схожі із задоволенням, якщо значення вигоди високе, або з болем, якщо вигода низька.

Мета оцінки цінностей – отримання більшої винагороди. Без винагороди не може бути цінностей, тому вони до певної міри первинні, а цінності – вторинні. Але, незважаючи на це, саме цінності цікавіші для прийняття та оцінки рішень. Оскільки цінність розглядає вигоду саме у довгостроковій перспективі, то вибір дій здійснюється саме на основі оціночних суджень, тобто таких дій, які призводять до станів найвищої цінності, а чи не найвищої винагороди. На жаль, визначити винагороди набагато легше, ніж цінності. Цінності необхідно обчислювати знову і знову з усієї послідовності спостережень, тоді винагороди здебільшого можна отримати безпосередньо із самого середовища. Найбільш важливий компонент практично всіх алгоритмів навчання з підкріпленням – це метод ефективної оцінки значень функції цінностей.

І, нарешті, останнім та необов'язковим елементом є модель навколишнього середовища. За допомогою моделі можна робити висновки про те, як поведе себе середовище, тобто завдання моделі – імітувати поведінку самого довкілля. За допомогою моделі можна розглядати можливі майбутні ситуації та, залежно від цього, приймати рішення про курс дій. Тобто моделі використовуються для планування: враховуючи стан та дію, вона може передбачити те, в якому стані виявиться довкілля, і відповідне йому винагороду.

Таким чином, якщо для вирішення завдань навчання з підкріпленням використовуються моделі та планування, то методи вирішення називаються методами, що базуються на моделях. Протилежністю цих методів є більш прості методи, які не використовують моделі, а навчаються методом проб та помилок.

Використання методу Монте-Карло для рішення сформульованої проблеми. Методи Монте-Карло – загальна назва групи чисельних методів. Вони базуються на отриманні якомога більшої кількості реалізацій випадкового процесу, який формується так, щоб його ймовірнісні характеристики збігалися з аналогічними величинами розв'язуваної задачі [2-5, 8,10].

Методи Монте-Карло здатні вирішити проблеми навчання із підкріпленням, ґрунтуючись на усередненні результатів вибірки. Щоб забезпечити доступність чітко визначених результатів, визначимо методи Монте-Карло лише для епізодичних завдань. Передбачається, що дані діляться на епізоди, які, так чи інакше, буде завершено, незалежно від того, які дії вибрано. Тільки після того, як епізод завершиться, може статися оцінювання цінності чи зміна стратегії. Таким чином, методи Монте-Карло можуть бути інкрементними лише на рівні епізодів.

Почнемо з розгляду методів Монте- Карло вивчення функції значення стану для заданої стратегії. Згадаймо, що цінність стану – це майбутня накопичена винагорода, починаючи з цього стану. Таким чином, можна оцінити вигоду, усереднивши результати отриманої вигоди після пройденого стану. Якщо кількість спостережень буде зростати, то кількість значень вигоди також буде збільшуватись. Отже, середнє значення вигоди буде прагнути до очікуваної величини. Ця ідея лежить в основі методів Монте-Карло.

Введемо такі позначення: нехай s – стан, π – стратегія. Враховуючи набір епізодів, які вийшли за допомогою застосування стратегії та проходження через стан, оцінимо цінність стану s за стратегії π. Ця величина буде позначатися як $v_\pi(s)$.

Відвідування s називається кожна поява стану s в епізоді. Звичйно, s може відвідати той самий епізод кілька разів. Назвемо перше відвідування в епізоді

першим відвідуванням s. Метод Монте-Карло першого відвідування оцінює $v_\pi(s)$ як усереднення значення винагород, які відповідають першим відвідуванням s, тоді як метод Монте-Карло всіх відвідувань оцінює величину як середнє після всіх відвідувань s в епізодах. Ці два методи Монте-Карло дуже схожі, проте мають різні теоретичні характеристики.

Якщо кількість відвідувань сягає нескінченності, то результат, який виходить при використанні будь-якого з вищезгаданих методів, сходить до $v_\pi(s)$.

У методі Монте-Карло оцінка одного стану жодним чином не базується на оцінці будь-якого іншого стану. Таким чином, ці оцінки є незалежними одна від одної. Цей факт є важливою особливістю методів Монте-Карло.

Також цікавою особливістю даного методу є те, що обчислювальні витрати на оцінку значення одного стану не залежать від кількості станів. Тобто можна створити велику вибірку тільки необхідних для роботи епізодів, не звертаючи уваги на інші. І рахувати середнє значення вигоди лише для цієї вибірки. Така особливість робить методи Монте-Карло дуже корисними, якщо необхідно оцінити цінність тільки одного або деякої підмножини станів.

Якщо модель є, то для визначення стратегій достатньо цінностей стану. Потрібно просто зробити крок і вибрати таку дію, яка призведе до найкращої винагороди. Але якщо модель відсутня, то таких даних буде недостатньо. І в такому разі краще оцінювати значення пар «стан – дія». Щоб значення були корисні під час обирання стратегії, потрібно явно оцінювати цінність кожної дії.

Таким чином, однією з основних цілей для застосування методів Монте-Карло є певна оцінка q*. Щоб досягти цього, спочатку розглянемо завдання оцінки стратегії.

Завдання оцінки стратегії з урахуванням значень дій полягає в тому, щоб оцінити $q_\pi(s,a)$ – очікувану вигоду на початку в стані s, виконуванні дії a та подальшому дотриманні стратегії π. Метод Монте-Карло для цього випадку такий самий, як і для розглянутого раніше випадку для значень стану, за винятком того, що тепер оцінюється пара «стан – дія», а не лише стан. Метод Монте-Карло всіх відвідувань оцінює цінність пари «стан – дія» як середнє значення вигоди, отриманої після всіх відвідувань. Метод Монте-Карло першого відвідування усереднює значення вигод після першого відвідування у кожному епізоді стану s і вибору в ньому дії a. Значення, отримані під час використання цих методів, сходяться квадратично до справжніх значень очікуваних цінностей, оскільки кількість відвідувань кожної пари «стан – дія» наближається до нескінченності.

Єдина складність полягає у тому, що багато пар «стан – дія» можуть ніколи не бути відвідуваними. У випадку, коли π – детермінована стратегія, вигода враховуватиметься тільки для однієї з дій кожного стану. І тоді оцінки для інших дій не будуть покращуватися з досвідом, оскільки значення вигоди не будуть усереднюватися. Згадаймо, що метою вивчення значень цінності дій є наступне: допомога у виборі дій, доступних у кожному стані. Але тоді перелічене вище стає серйозною проблемою, тому що неможливо порівняти дії між собою, щоб вибрати найкращу, оскільки потрібно оцінити цінність всіх дій з кожного стану, а не тільки того стану, який на даний момент є переважним.

Необхідно забезпечити постійне вивчення для того, щоб оцінити стратегію через цінність дії. Щоб гарантувати, що всі пари «стан – дія» будуть відвідані нескінченну кількість разів при нескінченній кількості епізодів, потрібно вказати, що перший крок кожного епізоду починається в парі «стан – дія» і що кожна пара має відмінну від нуля ймовірність бути обраною в якості початку. Це називається припущенням про вивчаючі старти.

На жаль, на припущення про вивчаючі старти не можна покладатися в цілому, оскільки стартові умови не завжди можуть бути корисними. Наприклад, при навчанні безпосередньо на основі фактичної взаємодії з навколишнім середовищем, варіантом для забезпечення появи всіх пар «стан – дія» може бути підхід, який полягає у розгляді лише стохастичних стратегій з ненульовою ймовірністю вибору всіх дій у кожному стані.

Розглянемо, як можна використовувати оцінку методом Монте-Карло для апроксимації оптимальних стратегій.

Стратегія покращується у кілька разів, щоб наблизитись до функції цінності. Але й функція цінності, у свою чергу, постійно змінюється, щоб найбільш точно наблизитись до поточної стратегії. Кожен із цих двох видів створює постійно мінливу мету один для одного, тобто певної міри працюють один проти одного. Але, незважаючи на це, вони наближають до оптимальності і цінність, і стратегію.

Спочатку розглянемо метод класичної ітерації за стратегіями. Виконуватимемо кроки, що чергуються: спочатку повну її оцінку, потім повне поліпшення стратегії. Почнемо з довільної стратегії π_0, а закінчимо оптимальною стратегією та оптимальною функцією цінності рис. 1.

$$\pi_0 \xrightarrow{E} q_{\pi o} \xrightarrow{I} \pi_1 \xrightarrow{E} q_{\pi 1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi_* \xrightarrow{E} q_*$$

Рис. 1 – Схема методу

E→ позначає повну оцінку стратегії, а I→ – повне покращення стратегії. Реалізується багато епізодів, де приблизна функція цінності дії асимптотично наближається до істинної функції. Припустимо, що спостерігатимемо нескінченну кількість епізодів і що, крім того, вони генеруватимуться за допомогою вивчаючих стартів. При цих припущеннях методи Монте-Карло будуть точно обчислювати кожне $q_{\pi k}$ для довільного π_k.

Стратегію можна покращити, зробивши її «жадібною» стосовно поточної функції цінності. Тоді будемо мати функцію «дія-цінність», отже, щоб побудувати «жадібну» стратегію, модель не знадобиться.

Для будь-якої функції цінність дії q, відповідної «жадібної» стратегії, це така стратегія, що для кожного s ∈ S обирає дію з максимальною цінністю:

$$\pi(S) = \arg\max_a q\,(s, a).$$

Потім можна покращити стратегію, побудувавши кожну π_{k+1} як жадібну по відношенню до $q_{\pi k}$. Для всіх s ∈ S

$$q_{\pi k}\big(s, \pi_{k+1}(s)\big) = q_{\pi k}\left(s, \arg\max_a q_{\pi k}(s, a)\right) = \max_a q_{\pi k}(s, a) \geq q_{\pi k}\big(s, \pi_k(s)\big) \geq v_{\pi k}(s).$$

Тоді кожна стратегія π_{k+1} краща, ніж π_k, або дорівнює їй у тому випадку, якщо вони обидві є оптимальними стратегіями. Це, у свою чергу, гарантує, що весь процес сходиться до оптимальної стратегії та оптимальної функції цінності.

Таким чином, методи Монте-Карло можна використовувати для знаходження оптимальних стратегій, враховуючи лише вибірку епізодів, за відсутності інших знань про динаміку довкілля.

Практичне моделювання і результати. Щоб зрозуміти, як працює метод Монте-Карло практично, розглянемо наступний приклад. Зіграємо у карткову гру блекджек та обчислимо функцію цінності.

Суть гри блекджек полягає в наступному: необхідно зібрати карти таким чином разом, щоб сума була максимальною, але при цьому не перевищувала 21. Король, дама, валет мають значення 10, туз приймає значення 1 чи 11, тоді він називається граючим. Інші карти мають значення відповідно до свого номіналу. Гравець грає зі здаючим, незалежно від інших учасників. На початку гри їм обом дають по дві карти, одна з розданих карт відкривається. Гравець може взяти собі ще одну карту, або він може зупинитися. Якщо він зупиняється, то той, хто здає, бере собі карти з колоди доти, доки їх сума не виявиться більшою або дорівнює 17. Якщо гравець або здаючий отримує в сумі більше 21, то він програє. В інших випадках виграє той, у кого сума карт виявиться більше, ніж в іншого. У випадку рівної суми – нічия.

Для імітації довкілля скористаємося середовищем Blackjack бібліотеки Gym [6, 9, 11-15]. Вона описується так:

1. Кожен епізод є марківським процесом прийняття рішень, на початку якого обидва учасники отримують свої дві карти, при цьому одна карта здаючого є відкритою.
2. Епізод закінчується у разі, якщо хтось виграє або гра завершується нічиєю. Винагорода нараховується наприкінці епізоду: 1, якщо гравець виграв; 0 – нічия; -1 якщо гравець програв.
3. У кожному раунді гравець має дві можливі дії: отримати ще одну карту (1) або більше не брати карти (0).

Подивимося, як працює це середовище. Для початку підключимо бібліотеки PyTorch та Gym і створимо екземпляр навколишнього середовища Blackjack [7]:

```
Import torch
import gym
env = gym.make('Blackjack-v0')
```

Потім переведемо середовище у вихідний стан командою env.reset() та отримаємо наступний результат на рис. 2:

(10, 2, False)

Рис. 2 - Початковий стан

Повертаються три змінні, які визначають:

1. Кількість очок у гравця – у цьому випадку 10.
2. Кількість очок у того, хто здає – у цьому випадку 2.
3. Наявність граючого туза у гравця – у разі відсутня.

Можна попросити ще одну картку командою env.step(1). Отримаємо результат на рис. 3:

((19, 2, False), 0.0, False, { })

Рис. 3 – Результат роботи команди

Після виконання цієї команди повертаються три змінні стани (19, 2, False), винагорода, що дорівнює нулю в даному випадку, і ознака завершення епізоду - False. Після цього гравець перестає брати карти за допомогою команди env.step(0).

Після цього до дій приступає здаючий, і в цьому випадку гравець програє рис. 4.

((19, 2, False), -1.0, True, { })

Рис. 4 – Завершення гри

Тепер перейдемо до передбачення цінності для простої стратегії, коли гравець перестає брати карти, якщо він набрав 19 очок.

Для початку напишемо функцію, яка імітує епізод Blackjack під час проходження простий стратегії:

```
def run_episode(env, hold_score):
state = env.reset()
rewards = []
states = [state]
is_done = False
while not is_done:
action = 1 if state[0] < hold_score else 0
state, reward, is_done, info = env.step(action)
states.append(state)
rewards.append(reward)
if is_done:
break
return states, rewards
```

Тепер визначимо функцію, яка оцінює просту стратегію методом Монте-Карло першого відвідування:

```
from collections import defaultdict
def mc_prediction_first_visit(env, hold_score, gamma, n_episode):
V = defaultdict(float)
N = defaultdict(int)
for episode in range(n_episode):
states_t, rewards_t = run_episode(env, hold_score)
return_t = 0
G = {}
for state_t, reward_t in zip(states_t[1::-1], rewards_t[::-1]):
return_t = gamma * return_t + reward_t
G[state_t] = return_t
for state, return_t in G.items():
if state[0] <= 21:
V[state] += return_t
N[state] += 1
for state in V:
V[state] = V[state] / N[state]
return V
```

Ця функція виконує такі дії : 1. Проганяє n_episode епізодів, слідуючи простий стратегії. 2. Обчислює доходи при першому відвідуванні кожного стану у кожному епізоді. 3. Усереднює доходи, отримані при першому відвідуванні кожного стану по всім епізодам, обчислюючи цим цінність. Стани, в яких гравець набрав більше 21 очка, ігноруються, тому що винагорода в них дорівнює -1.

Далі задаються початкові параметри ігри: кількість очок, при яких гра зупиняється, що дорівнює 19; коефіцієнт знецінення 1; кількість епізодів 500000:

```
hold_score = 19,
gamma = 1,
n_episode = 500000.
```

Виконаємо передбачення методом Монте-Карло з даними параметрами, роздруковуємо функцію цінності, що вийшла, виведемо кількість станів:

```
value = mc_prediction_first_visit(env, hold_score, gamma, n_episode)
print('Функція цінності, обчислена методом МК першого відвідування:\n',
value)
print('Кількість станів:', len(value))
```

В результаті отримуємо кількість станів 280.

Висновок. Проведене дослідження дало змогу показати, наскільки ефективно можна визначити функцію цінності 280 станів у середовищі BlackJack за допомогою передбачення методом Монте-Карло. При цьому було застосовано нестандартний підхід до навчання із заздалегідь невідомими навчальними прикладами, які підбиралися автоматично, у процесі оптимізації. Наведено можливі шляхи покращення стратегії:

− Збільшення числа оптимізованих параметрів.

− Застосування інших способів винагороди агента.

− Створення кількох конкуруючих між собою агентів збільшення простору варіантів.

Список літератури

1. *Sutton R., Barto A.* Reinforcement Learning: An Introduction. MIT Press; second edition, 2018. 552 p. P. 115–124.
2. *da Silva, W B.; Dutra, J. C.; Knupp, D. C.; Abreu, L. A.; Silva Neto, A. J.* Estimation of timewise varying boundary heat flux via Bayesian filters and Markov Chain Monte Carlo method. In Computational Intelligence in Emerging Technologies for Engineering Applications; Springer: Cham, Switzerland, 2020; pp. 137–153ttps://doi.org/10.1007/978-3-030-34409-2_8.
3. *Andrade, J.; Duggan, J.* An evaluation of Hamiltonian Monte Carlo performance to calibrate age-structured compartmental SEIR models to incidence data. Epidemics 2020, 33, 100415. ttps://doi.org/10.1016/j.epidem.2020.100415
4. *Jin, Y. F.; Yin, Z. Y.; Zhou, W. H.; Horpibulsuk, S.* Identifying parameters of advanced soil models using an enhanced transitional Markov chain Monte Carlo method. Acta Geotech. 2019, 14, 1925–1947. https://doi.org/10.1007/s11440-019-00847-1
5. *Durmus, A.; Moulines, É.; Pereyra, M.* A Proximal Markov Chain Monte Carlo Method for Bayesian Inference in Imaging Inverse Problems: When Langevin Meets Moreau. SIAM Rev. 2022, 64, 991–1028. https://doi.org/10.1137/22M1522917
6. *Chollet, F.* Deep learning with PYTHON. Second edition, M a n n i n g SHELTER ISLAND, 2021, P.504.
7. *Subramanian, V.* Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch 1788626079, 9781788626071. Poct, 2018, P.262.
8. *Hellweger V, Fischer J-T, Kofler A, Huber A, Fellin W, Oberguggenberger M* (2016) Stochastic methods in operational avalanche simulation—from back calculation to prediction. In: Paper presented at the international snow science workshop 2016 proceedings, Colorado, USA
9. *Півошенко В. В.* Аналіз та експериментальне дослідження методу безмодельного навчання з підкріпленням / В. В. Півошенко, М. С. Кулик, Ю. Ю. Іванов, А. С. Васюра // Вісник Вінницького політехнічного інституту. 2019. № 3. С. 40-49.
10. *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.
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
12. *M. Lapan* Deep Reinforcement Learning Hands-On, Packt Publishing Ltd, 2024, 716 p.
13. *Marc J Bellemare, Will Dabney, Mark Rowland* Distributional reinforcement learning, MIT Press, 2023. https://doi.org/10.7551/mitpress/14207.001.0001
14. *Kayakökü, Hakan & Guzel, Mehmet & Bostanci, Gazi Erkan & Medeni, Ihsan & Mishra, Deepti.* (2021). A Novel Behavioral Strategy for RoboCode Platform Based on Deep Q-Learning. Complexity. 2021. pp. 1-https://doi.org/10.1155/2021/9963018
15. *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

References (transliterated)

1. *Sutton R., Barto A.* Reinforcement Learning: An Introduction. MIT Press; second edition, 2018. 552 p. P. 115–124.
2. *da Silva, W. B.; Dutra, J. C.; Knupp, D. C.; Abreu, L. A.; Silva Neto, A. J.* Estimation of timewise varying boundary heat flux via Bayesian filters and Markov Chain Monte Carlo method. In Computational Intelligence in Emerging Technologies for Engineering Applications; Springer: Cham, Switzerland, 2020; pp. 137–153ttps://doi.org/10.1007/978-3-030-34409-2_8.
3. *Andrade, J.; Duggan, J.* An evaluation of Hamiltonian Monte Carlo performance to calibrate age-structured compartmental SEIR models

to incidence data. Epidemics 2020, 33, 100415. ttps://doi.org/10.1016/j.epidem.2020.100415
4. *Jin, Y. F.; Yin, Z. Y.; Zhou, W. H.; Horpibulsuk, S.* Identifying parameters of advanced soil models using an enhanced transitional Markov chain Monte Carlo method. Acta Geotech. 2019, 14, 1925–1947. https://doi.org/10.1007/s11440-019-00847-1
5. *Durmus, A.; Moulines, É.; Pereyra, M.* A Proximal Markov Chain Monte Carlo Method for Bayesian Inference in Imaging Inverse Problems: When Langevin Meets Moreau. SIAM Rev. 2022, 64, 991–1028. https://doi.org/10.1137/22M1522917
6. Chollet, F. Deep learning with PYTHON. Second edition, M a n n i n g SHELTER ISLAND, 2021, P.504.
7. *Subramanian, V.* Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch 1788626079, 9781788626071. Poct, 2018, P.262.
8. *Hellweger V, Fischer J-T, Kofler A, Huber A, Fellin W, Oberguggenberger M* (2016) Stochastic methods in operational avalanche simulation—from back calculation to prediction. In: Paper presented at the international snow science workshop 2016 proceedings, Colorado, USA
9. *Pivoshenko V. V.* Analiz ta eksperymentalne doslidzhennia metodu bezmodelnoho navchannia z pidkriplenniam / V. V. Pivoshenko, M. S. Kulyk, Yu. Yu. vanov, A. S. Vasiura // Visnyk Vinnytskoho politekhnichnoho instytutu. 2019. № 3. pp. 40-49.
10. *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.
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
12. *M. Lapan* Deep Reinforcement Learning Hands-On, Packt Publishing Ltd, 2020, 716 p.
13. *Marc J Bellemare, Will Dabney, Mark Rowland* Distributional reinforcement learning, MIT Press, 2023. https://doi.org/10.7551/mitpress/14207.001.0001
14. *Kayakökü, Hakan & Guzel, Mehmet & Bostanci, Gazi Erkan & Medeni, Ihsan & Mishra, Deepti.* (2021). A Novel Behavioral Strategy for RoboCode Platform Based on Deep Q-Learning. Complexity. 2021. pp. 1- https://doi.org/10.1155/2021/9963018
15. *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

Надійшла (received) 12.11.2024

Відомості про авторів/ About the Authors

Некрасова Марія Володимирівна – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут»; тел.: (057)-707-64-54; e-mail: masha12dec@gmail.com https://orcid.org/0009-0006-9285-0740

Nekrasova Mariia Volodymyrivna – Candidate of Technical Sciences, Dozent, National Technical University "Kharkiv Polytechnic Institute"; tel.: (057)-707-60-58; e-mail: masha12dec@gmail.com