

НЕЙРОАГЕНТНА МОДЕЛЬ ПРИЙНЯТТЯ РІШЕНЬ

© Кравець П., 2013

Досліджено нейроагентну модель прийняття рішень в умовах невизначеності. Розглянуто адаптивні методи навчання штучної нейронної мережі без учителя. Розроблено алгоритм та програмну модель нейроагентного прийняття рішень. Ефективність нейроагентного прийняття рішень підтверджено результатами комп'ютерного експерименту. Досліджено впливи параметрів моделі на швидкість навчання нейроагента.

Ключові слова: прийняття рішень, умови невизначеності, нейроагентна модель, адаптивний метод.

The neuroagent model of decision-making in the conditions of uncertainty is investigated. Adaptive methods of an artificial neural network training without the teacher are considered. The algorithm and program model of neuroagent decision-making are developed. Efficiency of neuroagent decision-making has been confirmed by results of computer experiment. Influences of parameters of model on the neuroagent training rate are investigated.

Key words: decision making, uncertainty conditions, neuroagent model, adaptive method.

Вступ до проблеми прийняття рішень в умовах невизначеності

У повсякденному житті, під час виконання своїх професійних обов'язків та у побуті, люди обдумують та приймають рішення, які забезпечують розв'язування різноманітних задач, досягнення тих або інших цілей, наприклад, мінімізацію зусиль, витрат ресурсів, часу під час виконання робіт, максимізацію прибутку, забезпечення інтелектуального, культурного та кар'єрного росту, підтримання здоров'я, психологічної рівноваги тощо.

Прийняття рішень – це процес раціонального або інтуїтивного вибору альтернативних варіантів дій, метою якого є досягнення поточного або віддаленого у часі результату [1 – 5]. Перелік можливих варіантів обумовлюється навколишнім середовищем та сферою зайнятості особи, яка приймає рішення. Так, діяльність вченого пов'язана з аналізом варіантів вирішення проблеми, праця викладача – з вибором способу подання матеріалу, а праця лікаря – з вибором стратегії лікування хворого.

Раціональний вибір альтернативних рішень складається з таких етапів [6]:

- 1) ситуаційний аналіз проблеми;
- 2) ідентифікація проблеми та формулювання мети;
- 3) пошук необхідної інформації;
- 4) формування альтернатив;
- 5) формування критеріїв для оцінювання альтернатив;
- 6) проведення оцінювання альтернатив;
- 7) вибір найкращої альтернативи;
- 8) реалізація альтернативи;
- 9) розроблення критеріїв для моніторингу впровадження альтернативного рішення;
- 10) моніторинг виконання варіанта рішення;
- 11) оцінювання результатів.

Система прийняття рішень (СПР) складається з об'єкта керування чи застосування рішення та особи, яка приймає рішення. Вважається, що у структурі СПР наявні зворотні зв'язки для оцінювання ефективності прийнятих рішень та коригування стратегії прийняття рішень у майбутні моменти часу. Такий підхід до побудови СПР називається оптимізаційним [7]. На відміну від ситуаційного підходу, коли рішення отримується одразу на основі вхідних даних, в оптимізаційному підході рішення уточнюється через ланку зворотного зв'язку.

Вибір варіантів рішень у проблемно-орієнтованих інформаційних системах та системах керування, зазвичай, здійснюється в умовах невизначеності [8, 9]. Під невизначеністю розуміється ситуація, коли недостатньо інформації для прийняття адекватних рішень або результати прийняття рішень є невідомими апіорі. Невизначеність СПР може бути зумовлена неточністю або неповнотою вхідних даних, стохастичною природою зовнішніх впливів, відсутністю адекватної математичної моделі, нечіткістю сформульованої мети, людським чинником тощо. В умовах невизначеності зростають ризики прийняття неефективних рішень з негативними економічними, технічними та соціальними наслідками. Невизначеності в системах прийняття рішень можуть бути частково компенсовані застосуванням різноманітних методів штучного інтелекту [10 – 12].

Для забезпечення ефективності СПР важливим є розроблення моделі особи, що приймає рішення (ОПР), та її використання на передпроектному етапі або для вироблення рекомендацій під час прийняття рішень. Система, що реалізує таку модель, називається системою підтримання прийняття рішень (СППР) [12 – 15].

Побудова сучасних СППР в умовах невизначеності здійснюється із залученням методів та засобів штучного інтелекту на основі агентно-орієнтованої методології [16 – 19]. Інтелектуальний агент прийняття рішень (АПР) – це модель ОПР, автономна система вироблення рішень з елементами штучного інтелекту. Агент взаємодіє з ОПР як консультаційна підсистема прийняття рішень.

Застосування сучасних методів та засобів штучного інтелекту до проектування СПР, що функціонують в умовах невизначеності інформації, є перспективним напрямком підвищення ефективності процесів прийняття рішень та управління [13].

Життєвий цикл АПР включає: вибір рішення, спостереження реакції середовища, опрацювання реакції для адаптивного формування стратегій прийняття рішень. Спостерігаючи стани системи і опрацюючи поточні виграші, агент повинен знайти такий спосіб прийняття рішень, який забезпечував би максимізацію його середніх виграшів у часі. Для цього він повинен містити механізм для інтегрального запам'ятовування реакцій середовища і адаптивного вироблення на його основі варіантів рішень.

За невизначеності умов функціонування системи для побудови інтелектуального АПР застосовують методи на основі автоматів зі змінною структурою, стохастичних ігор, правил нечіткої логіки, байєсових мереж довіри, прихованих марківських мереж, штучних нейронних мереж та інших адаптивних методів [9, 13, 20 – 22].

Для забезпечення високої продуктивності СПР та можливості паралельного опрацювання інформації використовують штучні нейронні мережі [23 – 27], які складаються з множини нейронів та зв'язків між ними. Для отримання необхідної структури зв'язків між нейронами використовується процедура навчання нейромережі, під час якої одні зв'язки підсилюються, а інші – послаблюються.

СПР, реалізована на основі апарата штучних нейронних мереж, називається нейроагентом прийняття рішень. Нейроагентний процес прийняття рішень може відбуватися в автономному режимі функціонування СПР або в інтерактивному режимі взаємодії нейроагента з ОПР.

Нейроні мережі реалізують „м'які” обчислення за зразком процесів, що відбуваються у мозку людини. У силу сучасних знань про організацію та функції мозку, подібність нейромережі з мозком полягає в тому, що інформація, яка надходить у нейромережу з навколишнього середовища, використовується для запам'ятовування і навчання мережі за допомогою коригування синаптичних зв'язків між нейронами.

Нейронні мережі використовуються як моделі об'єктів з невідомими характеристиками. Крім задач підтримки прийняття рішень, їх застосовують для розв'язування задач класифікації та розпізнавання об'єктів, апроксимації функцій за скінченим набором значень, прогнозування послідовностей, фільтрування зашумлених даних, стискання інформації, змістовного пошуку інформації, побудови асоціативної пам'яті, керування динамічними об'єктами тощо.

Використання нейронних мереж забезпечує такі переваги:

- 1) нелінійність – нейронні мережі дають змогу отримувати нелінійну залежність вихідного сигналу від вхідного;
- 2) адаптивність – нейронні мережі мають здатність пристосовувати свої синаптичні ваги до змін навколишнього середовища;
- 3) пластичність та стійкість до відмов – нейронні мережі зберігають інформацію у розподіленому по всіх зв'язках нейронної мережі вигляді і вихід з ладу одного або декількох нейронів не призводить до відмови системи, загалом;
- 4) універсальність – нейронні мережі не потребують спеціального програмування, оскільки вони дозволяють розв'язувати різні задачі опрацювання інформації за однаковими алгоритмами навчання нейронів.

Штучні нейромережі мають більш ніж піввікову історію дослідження: від моделі нейрона У. Мак-Каллока та У. Піттса [28], перцептрона Ф. Розенблатта [29] – до створення сучасних нейрокомп'ютерів [30], електронних машин з функціями, що моделюють роботу живого мозку. Незважаючи на це, питання застосування нейронних мереж для оптимізаційного прийняття рішень в умовах невизначеності у СПР зі зворотними зв'язками є недостатньо вивченим на теперішній час.

Метою цієї роботи є розроблення нейроагентної моделі оптимізаційного прийняття рішень в умовах невизначеності зі здатністю нагромаджувати досвід прийняття рішень та самонавчатися за рахунок адаптивної перебудови синаптичних зв'язків між нейронами.

Постановка задачі прийняття рішень в умовах невизначеності

СПР в умовах невизначеності описується кортежем (U, x, DM) , де $U = \{u[1], u[2], \dots, u[N]\}$ – набір дискретних варіантів рішень, $x(u) \in R^1$ – оцінка якості прийнятого рішення $u \in U$, яка є випадковою величиною з апіорі невідомим розподілом $x(u) \sim Z(v(u), d(u))$, математичним сподіванням $v(u) = E\{x(u)\}$ та обмеженою дисперсією $d(u) < \infty$, DM – метод прийняття рішення, який полягає у виборі одного із варіантів рішень $u \in U$.

Нехай СПР дозволяє повторний вибір варіантів рішень інтелектуальним агентом у моменти часу $t = 1, 2, \dots$. Після вибору варіанта $u_t = DM(x_{t-1}) \in U$ агент отримує випадковий поточний виграш $x_t(u_t)$.

Отримані поточні виграші агентів усереднюються у часі для оцінювання ефективності процесу прийняття рішень:

$$\Xi_t(\{u_t\}) = t^{-1} \sum_{t=1}^t x_t \quad (1)$$

Метою агента є максимізація функції середніх виграшів:

$$\lim_{t \rightarrow \infty} \Xi_t \rightarrow \max_u \quad (2)$$

Отже, на основі спостережень поточних виграшів x_t агент повинен вибрати поточні рішення $u_t = u \in U$ так, щоб з плином часу $t = 1, 2, \dots$ забезпечити максимізацію цільової функції (1).

Нейроагентний метод розв'язування задачі

Відомі адаптивні методи генерування послідовностей $\{u_t\}$, $t = 1, 2, \dots$ для максимізації середніх виграшів ґрунтуються на динамічних розподілах випадкових величин [31]. На відміну від них, розглянемо нейроагентний метод прийняття рішень. Структура нейроагентної системи прийняття рішень зображена на рис. 1.

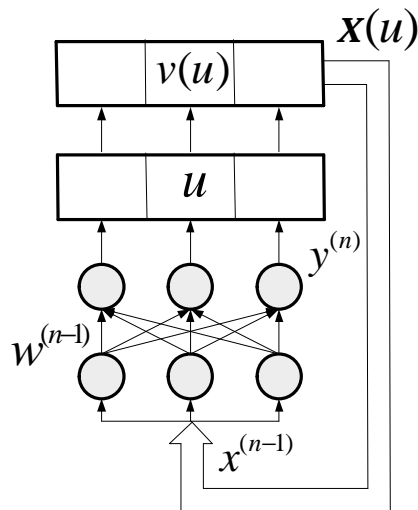


Рис. 1. Структура нейроагентної СПР

Модель середовища прийняття рішень задається вектором математичних сподівань випадкових вигащів v . Кількість елементів вектора дорівнює кількості варіантів рішень: $N = |U|$. На вхід середовища подаються значення варіантів рішень $u \in U$. Виходом середовища є відповідні значення поточних вигащів $x(u)$.

Нехай нейроагент задається штучною мережею з $n = 2$ шарами нейронів. Кількість елементів кожного шару є однаковою, що дорівнює кількості варіантів рішень N . На вхід нейроагента подається вектор параметрів $x^{(n-1)}$, обчислених на основі виходів середовища $x(u)$. Виходами нейроагента є вектор параметрів $y^{(n)}$, на основі якого визначаються варіанти рішень $u \in U$. Ваги $w^{(n-1)}$ позначають величину синаптичних зв'язків між нейронами. Додатні значення ваг зв'язків відповідають збуджуючим, а від'ємні – гальмівним синапсам. Нульове значення ваг означає відсутність зв'язку між нейронами.

Функціонування нейроагента здійснюється за одним із адаптивних алгоритмів навчання без учителя, наприклад, Хебба, Кохонена або за іншим [27]. Навчання без учителя, або самонавчання, за своєю природою найближче до свого біологічного прототипу – мозку. Самонавчання не орієнтується на наявність правильних виходів нейромережі. Алгоритм самонавчання самостійно виявляє внутрішню структуру вхідних даних, перебудовуючи ваги синаптичних зв'язків так, щоб близькі (за деякою метрикою) набори вхідних сигналів викликали достатньо близькі вихідні набори сигналів. Фактично, процес самонавчання нейромережі розв'язує задачу кластеризації даних, виявляючи статистичні властивості навчальних множин та групуєчи подібні вихідні множини у кластери. Подаючи на вхід навченої нейромережі вектор із заданого класу, отримаємо характерний для цього класу вихідний вектор. Вихідний вектор наперед не відомий. Його формування обумовлене структурою навчальної вибірки, випадковим розподілом початкових значень ваг зв'язків між нейронами та комбінацією збуджених нейронів вихідного шару нейромережі.

Вибір варіантів рішень $u \in U$ здійснюється детермінованим або стохастичним методом.

Детермінований метод полягає у визначенні варіантів рішень на основі максимальних значень виходів $y_i^{(n)}$:

$$u_i = \left(u[i] \mid i = \arg \max_{j=1..N} y_i^{(n)}[j] \right). \quad (3)$$

Застосування цього методу може призвести до того, що деякі з нейронів не братимуть участі у навчанні.

На відміну від попереднього, стохастичний метод розширює свободу вибору нейроагента. Для цього будується допоміжний вектор умовних імовірностей вибору варіантів рішень проектуванням вектора виходів $y_t^{(n)}$ на одиничний N -вимірний e -симплекс:

$$p(u_t | u_t, x_t, t = 1, 2, \dots, t-1) = p_e^N \{y_t^{(n)}\}, \quad (4)$$

де p_e^N – проєктор на одиничний e -симплекс $S_e^N \subseteq S^N$ [31]. Параметр e регулює швидкість розширення e -симплексу S_e^N до одиничного симплексу S^N і може бути використаний як додатковий фактор керування збіжністю нейроагентного методу прийняття рішень.

Отриманий вектор імовірностей використовується для побудови емпіричного розподілу дискретних випадкових величин, на основі якого здійснюється вибір варіантів рішень:

$$u_t = \left\{ u[i] \mid i = \arg \left(\min_i \sum_{j=1}^i p_t(u_t[j]) > w \right), i = 1..N \right\}, \quad (5)$$

де $w \in [0, 1]$ – дійсне випадкове число з рівномірним розподілом.

Реакцією середовища прийняття рішень на вибраний варіант є значення випадкової величини з невідомим розподілом Z , яка інтерпретується як поточний вигравш агента:

$$x(u_t) \sim Z(v(u_t), d(u_t)),$$

де $v(u_t)$ – математичне сподівання, $d(u_t)$ – дисперсія.

Отриманий поточний вигравш $x_t(u_t)$ подається на входи нейроагента

$$x^{(n-1)} = e x(u_t), \quad (6)$$

де $e = (1 | \forall u \in U)$ – вектор, усі елементи якого дорівнюють одиниці; $c() \in \{0, 1\}$ – індикаторна функція події.

За потреби здійснюється нормалізація елементів вектора $x^{(n-1)}$, наприклад, така:

$$x^{(n-1)} = e x(u_t) / |x_{\max}|,$$

де x_{\max} – максимальне значення поточних вигравшів. Нормалізація може призвести до зменшення кількості кроків, необхідних для навчання нейроагента.

Сумарні входи $x^{(n)}$ нейронів n -го шару обчислюються на основі виходів $y^{(n-1)}$ нейронів $(n-1)$ -го шару:

$$x^{(n)}[j] = \sum_{i=1}^N w_i^{(n-1)}[i, j] y^{(n-1)}[i], \quad j = 1..N, \quad (7)$$

де $w_i^{(n-1)}[N, N]$ – матриця ваг зв'язків між вузлами нейромережі, обчислених у момент часу t . Тут $w_i^{(n-1)}[i, j]$ позначає вагу зв'язку між i -м вузлом $(n-1)$ -го шару та j -м вузлом n -го шару.

Для обчислення виходів $y^{(n)}$ нейроагента використовується передатна функція $j()$ нейрона:

$$y^{(n)}[j] = j(x^{(n)}[j]), \quad (8)$$

де $j = 1..N$.

Залежно від розв'язуваної задачі та виду нейромережі передатна функція $j()$ може бути пороговою, лінійною з насиченням, сигмоїдальною, синусоїдальною, радіально-симетричною тощо [27].

Найчастіше для моделювання штучної нейронної мережі використовують лінійну

$$y_i^{(n)}[j] = \begin{cases} 0, & \text{if } x_i^{(n)}[j] \leq q, \\ h(x_i^{(n)}[j] - q), & \text{if } x_i^{(n)}[j] > q, \end{cases} \quad (9)$$

та біполярну сигмоїдальну

$$y_i^{(n)}[j] = -0.5 + 1 / (1 + e^{-h(x_i^{(n)}[j] - q)}) \quad (10)$$

передатні функції. Параметр $h > 0$ задає тангенс кута нахилу прямої для лінійної передатної функції та рівень крутості для сигмоїдальної передатної функції. Параметр $q \geq 0$ визначає поріг активації нейрона.

Навчання нейроагента здійснюється зміною ваг $w_i^{(n-1)}$ синаптичних зв'язків між нейронами. Перерахунок ваг зв'язків здійснюється за сигнальним методом Хебба, методом Кохонена або за іншим методом навчання без учителя.

Навчання за методом Хебба призводить до підсилення зв'язків між збудженими нейронами:

$$w_i^{(n-1)}[i, j] = w_{i-1}^{(n-1)}[i, j] + g_i (y_{i-1}^{(n-1)}[i] * y_{i-1}^{(n)}[j]), \quad i = 1..N, \quad j = 1..N. \quad (11)$$

Збудженими називаються нейрони, для яких значення сумарного входу $x^{(n)}$ перевищує поріг активації q .

Навчання за диференційним методом Хебба призводить до підсилення зв'язків між тими нейронами, виходи яких змінилися найбільше:

$$w_i^{(n-1)}[i, j] = w_{i-1}^{(n-1)}[i, j] + g_i (y_i^{(n-1)}[i] - y_{i-1}^{(n-1)}[i]) * (y_i^{(n)}[j] - y_{i-1}^{(n)}[j]), \quad i = 1..N, \quad j = 1..N. \quad (12)$$

Навчання за методом Кохонена ґрунтується на механізмі конкуренції, суть якого полягає у мінімізації різниці між вхідними сигналами нейрона-переможця, що надходять з виходів нейронів попереднього шару, та ваговими коефіцієнтами його синапсів:

$$w_i^{(n-1)}[i, k] = w_{i-1}^{(n-1)}[i, k] + g_i (y_{i-1}^{(n-1)}[i] - w_{i-1}^{(n-1)}[i, k]), \quad i = 1..N, \quad (13)$$

де k – індекс нейрона-переможця.

На відміну від методу Хебба, в якому одночасно можуть збуджуватися декілька нейронів одного шару, у методі Кохонена нейрони одного шару змагаються між собою за право активації [23]. Це правило відоме у літературі з машинного навчання під назвою „переможець забирає все”.

У методі Кохонена перебудова ваг зв'язків здійснюється тільки для нейрона-переможця. Переможцем є той нейрон, значення синапсів якого максимально подібні на вхідний образ.

Визначають нейрон-переможець обчисленням відстані між векторами $y_{i-1}^{(n)}$ та $w_{i-1}^{(n-1)}$:

$$D_{i-1}[j] = \sqrt{\sum_{i=1}^N (y_{i-1}^{(n-1)}[i] - w_{i-1}^{(n-1)}[i, j])^2}, \quad j = 1..N.$$

Перемагає нейрон з найменшою відстанню:

$$k = \underset{j=1..N}{\text{index min}}(D_{i-1}[j]).$$

Інший спосіб визначення нейрона-переможця полягає у максимізації виходів $y_{i-1}^{(n)}$ нейронів n -го шару згідно з (3). У цьому випадку індекс нейрона-переможця є порядковим номером вибраного варіанта рішення u_{i-1} :

$$k = \text{index}(u[j] | c(u[j] = u_{i-1}), \quad j = 1..N). \quad (14)$$

Навколо нейрона-переможця можна задавати радіус навчання R у просторі векторів ваг нейронів:

$$r_i[j] = \|w_i^{(n-1)}[k] - w_i^{(n-1)}[j]\|, \quad j = 1..N,$$

де $w_i^{(n-1)}[k]$ – вектор ваг нейрона-переможця; $\|*\|$ – евклідова норма вектора.

Кожен нейрон, відстань від вектора ваг якого до вектора ваг нейрона-переможця є меншою від радіуса навчання ($r_i[j] < R$), бере участь у перерахунку ваг синапсів. Ваги нейронів, які знаходяться за межами радіуса навчання, не змінюються. Радіус навчання зменшується у часі так, щоб у кінці процесу навчання коригувати ваги зв'язків міг тільки один нейрон-переможець.

Параметри g_i у (11) – (13) та e_i у (4) визначають швидкість навчання нейроагента. Для забезпечення збіжності процесу навчання нейроагента ці параметри задаються як додатні монотонно спадні величини:

$$g_i = g_0 / t^a, \quad e_i = e_0 / t^b, \quad (15)$$

де $g_0, a > 0$; $e_0, b > 0$.

Вибір варіантів рішень продовжується до заданої кількості кроків t_{\max} , або до виконання умови точності навчання:

$$d_t = \|w_t^{(n-1)} - w_{t-1}^{(n-1)}\| < e, \quad (16)$$

де e – точність навчання нейроагента, яка визначається евклідовою нормою зміни ваг зв'язків між нейронами за два послідовні моменти часу.

Якість СПР оцінюється досягнутим під час оптимізації значенням функції середніх виграшів Ξ_t (1) та похибкою вибору оптимального варіанта прийняття рішень.

Оптимальним вважатимемо варіант прийняття рішень з максимальним значенням математичного сподівання виграшу:

$$u^* = \arg v(u^*),$$

де $v(u^*) = \max_{u \in U} v(u)$.

Похибка вибору оптимального варіанта рішення може бути визначена за відхиленням $\Delta(\Xi_t)$ значення функції середніх виграшів від прогнозованого оптимального значення або похибкою $\Delta(p_t)$ імовірності вибору оптимального варіанта прийняття рішень.

Відхилення значення функції середніх виграшів від прогнозованого оптимального значення:

$$\Delta(\Xi_t) = |\Xi_t - v(u^*)|. \quad (17)$$

Усереднена у часі похибка імовірності вибору оптимального варіанта рішення:

$$\Delta(p_t) = \frac{1}{t} \sum_{t=1}^t \|p_t - e^*\|, \quad (18)$$

де e^* – одиничний вектор-індикатор оптимального варіанта рішення.

Елементи вектора e^* визначаються так:

$$e^*[i] = \begin{cases} 0, & \text{if } i \neq \arg \max_{j=1..N} v[j] \\ 1, & \text{if } i = \arg \max_{j=1..N} v[j] \end{cases}, \quad i = 1..N. \quad (19)$$

Для спрощення дослідження збіжності алгоритму навчання нейроагента припустимо, що варіант оптимального рішення єдиний, тобто:

$$h = v(u^*) - \max_{u \in U, u \neq u^*} v(u) > 0. \quad (20)$$

Алгоритм Кохонена функціонування нейроагента

1. Задати початкові значення параметрів:

$t = 0$ – початковий момент часу;

N – кількість варіантів рішень;

$U = \{u[1], u[2], \dots, u[N]\}$ – множина варіантів рішень;

$v = (v_1, v_2, \dots, v_N)$ – вектор математичних сподівань виграшів;

$d = (d_1, d_2, \dots, d_N)$ – вектор дисперсій виграшів;

$w_0^{(n-1)}[N, N]$ – матриця початкових ваг зв'язків між вузлами нейромережі;

h, q – параметри передатної функції нейронів;

g_0 – параметр кроку навчання нейронів;

$a \in [0, 1]$ – порядок кроку навчання нейронів;

e_0 – параметр e -симплекса;

b – порядок швидкості розширення e -симплекса;

t_{\max} – максимальна кількість кроків методу;

e – точність навчання.

2. Виконати детермінований вибір варіанта рішення u_t згідно з (3), або стохастичний вибір згідно з (4) – (5).

3. Отримати значення поточних виграшів, як випадкову величину з нормальним розподілом $x(u_t) \sim Normal(v(u_t), d(u_t))$. Нормально-розподілені виграші обчислюються за допомогою суми випадкових величин $w \in [0,1]$ з рівномірним розподілом

$$x(u_t) = v(u_t) + \sqrt{d(u_t)} \left(\sum_{j=1}^{12} w_t[j] - 6 \right).$$

4. Обчислити входи нейроагента $x^{(n-1)}$ згідно з (6) та відповідні виходи $y^{(n-1)}$ нейронів $(n-1)$ -го шару згідно з (8). У якості передатної функції вибрано лінійну функцію (9).

5. Обчислити сумарні входи нейронів $x^{(n)}$ (7) та відповідні виходи $y^{(n)}$ (8) для нейронів n -го шару.

6. Обчислити значення параметра g_t згідно з (15).

7. Визначити індекс k нейрона-переможця згідно з (14).

8. Обчислити ваги зв'язків для нейрона-переможця $w_t^{(n-1)}[j,k]$, $j=1..N$ згідно з (13).

9. Обчислити характеристики якості прийняття рішень Ξ_t (1), $\Delta(\Xi_t)$ (17) та $\Delta(p_t)$ (18).

10. Задати наступний момент часу $t := t + 1$.

11. Якщо $t < t_{\max}$, то перейти на крок 2, інакше – кінець.

Результати комп'ютерного моделювання

Працездатність нейроагентної моделі щодо можливості навчатися вибирати оптимальні варіанти рішень підтверджено результатами комп'ютерного експерименту.

На рис. 2 у логарифмічному масштабі зображено графіки функції середніх виграшів Ξ_t та похибок $\Delta(\Xi_t)$ і $\Delta(p_t)$ вибору оптимального варіанта керування для середовища з нормально розподіленими виграшами $x(u) \forall u \in U$ з математичними сподіваннями $v = (0,3;0,9;0,1;0,5)$ та дисперсіями $d = (0,01;0,01;0,01;0,01)$. Нейроагент отримує поточні випадкові виграші як реакції середовища за вибір одного із $N = |U| = 4$ варіантів рішень.

Початкові значення ваг зв'язків $w_0^{(n-1)}$ між нейронами є випадковими величинами, рівномірно розподіленими в інтервалі $[0; 1]$. Результати отримано для алгоритму навчання нейроагента (13) з параметрами: $g = 1$; $a = 0,1$; $e = 0,999/N$; $b = 1$.

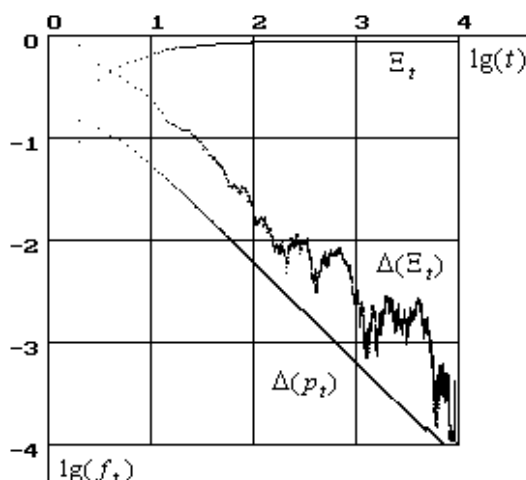


Рис. 2. Характеристики навчання нейроагента

Зростання Ξ_t до значення $v(u^*)$ і відповідні зменшення $\Delta(\Xi_t)$ і $\Delta(p_t)$ у часі свідчать про збіжність нейроагентного методу прийняття рішень у сенсі виконання цільової функції (2). Спадання графіків функцій $\Delta(\Xi_t)$ і $\Delta(p_t)$ демонструє те, що нейроагентний метод (13) навчається вибирати оптимальний варіант рішення з найбільшим значенням математичного сподівання виграшу.

Вивчимо залежність часу навчання нейроагента від основних параметрів алгоритму. Час навчання визначимо як мінімальну кількість кроків, необхідних для навчання нейроагента із заданою точністю $\epsilon > 0$:

$$t_{out} = (t = t_{min} | d_t < \epsilon),$$

де поточна точність навчання d_t обчислюється згідно з (16).

Для алгоритму з випадковим вибором варіантів рішень необхідно виконати усереднення часу навчання нейроагента для різних послідовностей випадкових величин:

$$\bar{t} = \frac{1}{k_{exp}} \sum_{j=1}^{k_{exp}} t_{out}[j],$$

де k_{exp} – кількість експериментів.

Середня кількість кроків навчання \bar{t} залежить від параметрів алгоритму навчання нейроагента та параметрів середовища прийняття рішень.

Графік залежності середнього часу \bar{t} навчання нейроагента від параметра α зображено на рис. 3 у логарифмічному масштабі. Параметр $\alpha \in (0,1]$ визначає порядок монотонного спадання величини $g_t > 0$ (15), яка регулює швидкість навчання нейроагента. Із збільшенням α значення g_t зменшується. Наведені результати отримано для середовища прийняття рішень з параметрами $N = |U| = 4$, $v = (0,3; 0,9; 0,1; 0,5)$ та $d = (0,01; 0,01; 0,01; 0,01)$.

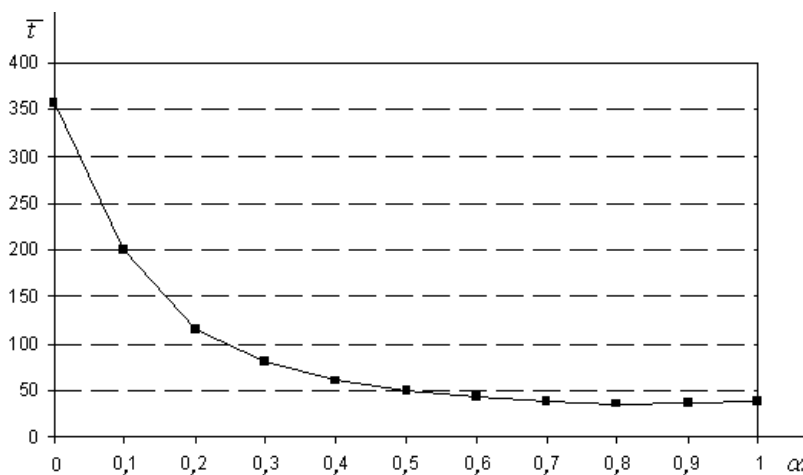


Рис. 3. Вплив параметра α на час навчання нейроагента

Точність навчання нейроагента дорівнює $\epsilon = 10^{-3}$. Дані усереднено за $k_{exp} = 100$ експериментами. В усіх експериментах навчання із заданою точністю забезпечує правильну локалізацію оптимального варіанта рішення, який визначається умовою (19).

Як видно із результатів моделювання, збільшення параметра α призводить до зменшення середньої кількості кроків \bar{t} навчання нейроагента.

Залежність середньої кількості кроків \bar{t} навчання нейроагента від дисперсії d стохастичного середовища (дисперсії оцінок варіантів рішень) подано на рис. 4.

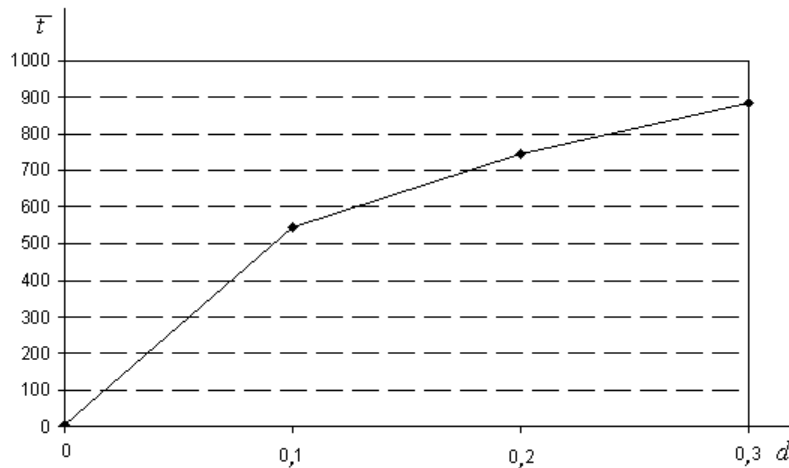


Рис. 4. Вплив дисперсії оцінок варіантів рішень на час навчання нейроагента

Із зростанням дисперсії зростає середня кількість кроків, необхідних для навчання нейроагента. Зазначимо, що надмірне зростання дисперсії може призвести до неправильного визначення оптимального варіанта рішення.

Вплив кількості варіантів прийняття рішень $N = |U|$ на швидкість навчання нейроагента зображено на рис. 5. Середовище прийняття рішень задається параметрами:

$$v(u) = 0,1 \quad u \in U, \quad u \neq u^* ;$$

$$v(u^*) = 0,9 ;$$

$$d(u) = 0,01 \quad \forall u \in U .$$

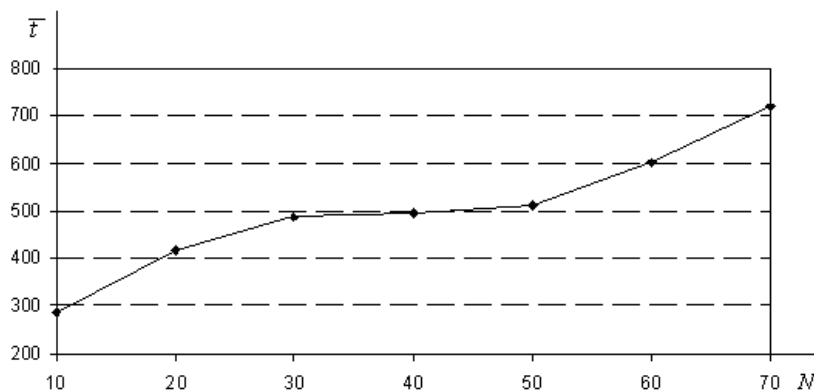


Рис. 5. Залежність часу навчання нейроагента від кількості варіантів рішень

Зростання кількості варіантів прийняття рішень призводить до сповільнення швидкості збіжності алгоритму навчання нейроагента.

Експериментально встановлено, що збіжність методу залежить від величини h інтервалу розділення математичних сподівань виграшів (20).

Графіки функції $\Delta(p_t)$, отримані для різних значень параметра h , зображено на рис. 6 . Випадкове середовище прийняття рішень задається параметрами:

$$N = |U| = 4 ;$$

$$v(u) = 0,1 \quad u \in U, \quad u \neq u^* ;$$

$$v(u^*) = v(u) + k * 0,1, \quad k = 1, 2, \dots ;$$

$$d(u) = 0,01 \quad \forall u \in U .$$

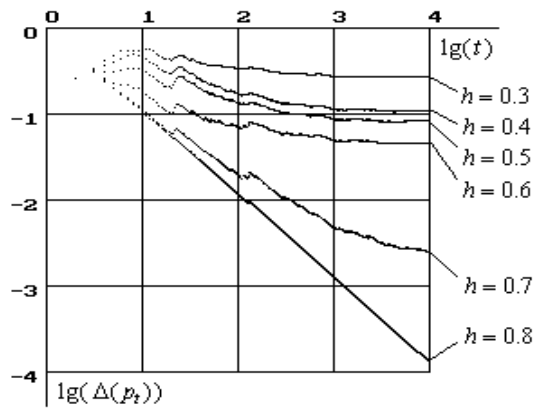


Рис. 6. Залежність швидкості навчання нейроагента від інтервалу розділення оцінок варіантів рішень

Як видно на рис. 6, із збільшенням h збіжність нейроагентного методу покращується, що виявляється у зростанні імовірності визначення оптимального значення $v(u^*)$ і, відповідно, спаданні $\Delta(p_t)$ під час повторень комп'ютерного експерименту.

Порядок швидкості збіжності нейроагентного методу прийняття рішень можна оцінити за значенням крутості графіка функції $\Delta(p_t)$, яке можна обчислити як тангенс кута лінійної апроксимації функції $\Delta(p_t)$ з віссю часу у логарифмічному масштабі. Зростання крутості графіків $\Delta(p_t)$ свідчить про збільшення швидкості навчання нейроагента.

Висновки

Запропоновано нову нейроагентну модель та метод адаптивного прийняття рішень в умовах стохастичної невизначеності, оснований на штучній нейронній мережі зі зворотними зв'язками з навчанням без учителя. Поточний варіант рішення вибирається на основі виходів нейромережі детермінованим або випадковим способом. Детермінований вибір оснований на визначенні максимального значення виходу нейромережі. Випадковий вибір передбачає визначення імовірностей вибору варіантів рішень оптимальним проектуванням виходів нейромережі на одиничний симплекс. Після вибору варіанта рішення визначається реакція середовища прийняття рішень як значення поточного виграшу нейроагента. Поточний виграш по ланці зворотного зв'язку скеровується на входи двошарової нейромережі. Далі відбувається навчання нейроагента шляхом модифікації ваг зв'язків між нейронами за одним із алгоритмів навчання без учителя. Процес навчання повторюється у реальному масштабі часу до моменту стабілізації ваг зв'язків між нейронами із заданою точністю. Хід навчання спрямований на максимізацію середніх виграшів нейроагента.

Розроблена програмна модель підтверджує збіжність адаптивного нейроагентного методу (13) прийняття рішень. Ефективність методу оцінено за допомогою характеристичних функцій середніх виграшів та похибок вибору оптимального варіанта керування. Збіжність нейроагентного методу залежить від кількості варіантів рішень та співвідношень параметрів методу і середовища прийняття рішень. Результатом зростання кількості варіантів рішень або дисперсії їх оцінок є зменшення швидкості збіжності нейроагентного методу. Розширення інтервалу розділення математичних сподівань виграшів призводить до покращення швидкості збіжності процесу навчання нейроагента.

Достовірність отриманих результатів підтверджено повторювальністю значень розрахованих характеристик нейроагентного методу прийняття рішень для різних послідовностей випадкових величин.

1. Кини Р.Л. Принятие решений при многих критериях: предпочтения и замещения: Пер. с англ. под ред. И. Ф. Шахнова / Р.Л. Кини, Х. Райфа. – М.: Радио и связь, 1981. – 560 с. 2. Евланов Л. Г.

Теория и практика принятия решений / Евланов Л. Г. – М.: Экономика, 1984. – 176 с. 3. Лескин А. А. Системы поддержки управленческих и проектных решений / А. А. Лескин., В. Н. Мальцев. – Л.: Машиностроение. Ленингр. отд-е, 1990. – 167 с. 4. Литвак Б. Г. Разработка управленческого решения / Б. Г. Литвак. – М.: Изд-во “Дело”, 2004 г. – 392 с. 5. Орлов А. И. Теория принятия решений: учебник / А. И. Орлов. – М.: Экзамен, 2006. – 573 с. 6. Теория принятия решений / Википедия.mht: [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/../> Теория_принятия_решений. 7. Дубовой В. М. Модели прийняття рішень в управлінні розподіленими динамічними системами. Монографія / В. М. Дубовой, О. О. Ковалюк. – Вінниця: УНІВЕРСУМ-Вінниця, 2008. – 185 с. 8. Кондратенко Ю.П. Оптимізація процесів прийняття рішень в умовах невизначеності: Навчальний посібник / Ю.П. Кондратенко. – Миколаїв: Вид-во МДГУ ім. Петра Могили, 2006. – 96 с. 9. Адаптивна система підтримки прийняття рішень з використанням методів нечіткого логічного висновку / Р. Н. Кветний, В. Ю. Коцюбинський, Л. М. Кислиця, Н. В. Казимірова, Г. О. Кириленко // Інформаційні технології та комп'ютерна техніка: Наукові праці ВНТУ. – 2011. – № 3. – С. 1 – 10. 10. Дюк В. Data Mining: учебный курс / В. Дюк, А. Самойленко. – СПб: Изд. Питер, 2001. – 368 с. 11. Литвак Б. Г. Экспертные оценки и принятие решений / Литвак Б. Г. – М.: Патент, 1996. – 271 с. 12. Геловани В. А. Интеллектуальные системы поддержки принятия решений / В.А. Геловани, А.А. Баишыков, В.Б. Бритков, Е.Д. Вязилов. – М.: Эдиториал УРСС, 2001. – 304 с. 13. Кондратенко Ю.П. Системы підтримки прийняття рішень на основі нечіткого логічного виводу / Ю.П. Кондратенко // Комп'ютерні технології: Наукові праці Миколаївського державного гуманітарного університету ім. Петра Могили. – Том 35. – Вип. 22. – С. 71 – 80. 14. Ситник В. Ф. Системы підтримки прийняття рішень: навч. посіб. / В.Ф. Ситник. – К.: КНЕУ, 2004. – 614 с. 15. Верес О.М. Компоненти концептуальної моделі системи підтримки прийняття рішень / О.М. Верес // Комп'ютерні науки та інформаційні технології. – Львів : Вісн. Нац. ун-ту “Львівська політехніка”. – № 686. – 2010. – с. 103 – 111. 16. Wooldridge M. An Introduction to Multiagent Systems / M. Wooldridge. – John Wiley & Sons, 2002. – 366 p. 17. Городецкий В.И. Информационные технологии и многоагентные системы / В.И. Городецкий // Проблемы информатизации. – 1998. – Вып. 1. – С. 3 – 14. 18. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика / В.Б.Тарасов. – М.: Эдиториал УРСС, 2002. – 352 с. 19. Швецов А.Н. Агентно-ориентированные системы: от формальных моделей к промышленным приложениям / А.Н. Швецов. – Вологодский гос. технич. унив. – 101 с.: [Электрон. ресурс]. – <http://www.ict.edu.ru/ft/005656/62333e1-st20.pdf>. 20. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы: пер. с польск. И.Д. Рудинского / Данута Рутковская, Мачей Пилиньский, Лешек Рутковский. – М. : Горячая линия – Телеком, 2007. – 452 с. 21. Таха Хемди А. Теория игр и принятия решений. Глава 14 / Хемди А. Таха // Введение в исследование операций = Operations Research: An Introduction. – 7-е изд. – М.: Вильямс, 2007. – С. 549–594. 22. Згуровський М. З. Системна методика побудови байесових мереж / М. З. Згуровський, П.І. Бідюк., О.М. Терентьєв // Наукові вісті НТУУ “КПІ”. – 2007. – № 4. – С. 47–61. 23. Уоссермен Ф. Нейрокомп'ютерна техніка / Ф. Уоссермен. – М. : Мир, 1992. – 240 с. 24. Крисилов В.А. Применение нейронных сетей в задачах интеллектуального анализа информации / В.А. Крисилов, Д.Н. Олешко, А.В. Трутнев // Тр. Одесского политехн. ун-та, Вып.2 (8). – 1999. – 134 с. 25. Барский А. Б. Нейронные сети: распознавание, управление, принятие решений / А. Б. Барский. – М.: Финансы и статистика, 2004. – 176 с. 26. Каширина И.Л. Искусственные нейронные сети: Учебное пособие / И.Л. Каширина. – Воронеж: Изд-во ВГУ, 2005. – 51 с. 27. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1104 с. 28. Мак-Каллок У. Логические исчисления идей, относящихся к нервной активности / У. Мак-Каллок, У. Питтс. – Сб. “Автоматы”. – М.: ИЛ, 1956. 29. Розенблатт Ф. Принципы нейродинамики (Перцептроны и теория механизмов мозга) / Ф. Розенблатт. – М.: Мир, 1965. – 480 с. 30. Злобин В.К. Нейросети и нейрокомпьютеры: учеб. пособие для студ. вузов / В. К. Злобин, В. Н. Ручкин. – СПб. : БХВ-Петербург, 2011. – 256 с. 31. Назин А.В. Адаптивный выбор вариантов: Рекуррентные алгоритмы / А.В. Назин, А.С. Позняк. – М.: Наука, 1986. – 288 с.