

УДК 004.032.26:004.832

DOI: 10.32626/2308-5916.2019-20.121-130

С. І. Шаповалова, канд. техн. наук,

О. М. Бараніченко, аспірант

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського», м. Київ

РОЗВ'ЯЗАННЯ ЛОГІЧНИХ ЗАДАЧ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

У статті запропоновано спосіб розв'язання логічних задач-головоломок на основі машинного навчання. Спосіб розраховано на попередню формалізацію задач у вигляді опису властивостей та відношень між ними. Оскільки кожна властивість має множину можливих значень, розв'язання задачі методами перебору має комбінаторну складність. При великій кількості властивостей та їх значень час розв'язання стрімко зростає.

В останні роки окремим напрямом досліджень з машинного навчання стало розв'язання логічних задач такого типу. Однак існуючі рішення цього напрямку мають ряд недоліків, насамперед вони не завжди гарантують коректне розв'язання.

У роботі представлено спеціальну мережу зв'язків для навчання розв'язанню логічних задач, а також їх формалізацію для представлення цій мережі. Мережа містить обчислювальні вузли, які відображають відношення між властивостями, та вузли вхідних шарів, які задають значення цих відношень.

Розв'язання кожної задачі відбувається шляхом автоматичного створення мережі зв'язків з її подальшим навчанням до отримання розв'язку. Приведено геометричну інтерпретацію n -мірної мережі зв'язків та її $(n - 1)$ -мірних шарів. Наведено формалізацію представлення навчальної вибірки та алгоритм навчання. Представлено механізм розв'язання логічних комбінаторних задач.

Наведено приклади задач, які є традиційними тестами в системах логічного програмування та продукційних (експертних) системах, а також задач з ресурсу bAbI таких класів: two supporting facts, two arguments relations, positional reasoning.

Експериментально доведено ефективність запропонованого способу.

Визначено перспективи подальших досліджень, які пов'язані зі створенням лексико-синтаксичного аналізатора для автоматичного представлення властивостей, їх значень та відношень між ними.

Запропонований спосіб є універсальним і не залежить від характеристик поточної задачі, таких як кількості властивостей та їх значень.

Ключові слова: логічні головоломки, машинне навчання.

Вступ. В сучасних прикладних програмних системах часто виникає необхідність у розв'язанні логічних задач. Різновидом таких задач є задача підбору значень властивостей на основі опису об'єктів і зв'язків між ними. В таких задачах кожен об'єкт має набір властивостей. Кожна властивість має множину можливих значень. Задача полягає у визначенні такої комбінації значень властивостей всіх об'єктів, що не суперечать заданим умовам. Прикладами таких задач є створення розкладів, логістика, розподіл ресурсів. Як правило, алгоритми розв'язання мають комбінаторну складність, завдяки перебору значень аргументів задач. Зі збільшенням кількості аргументів на порядки зростає час розв'язання. Зі стрімким розвитком нейронних мереж з'явилися дослідження з розв'язання цих задач на основі машинного навчання. Однак існуючі рішення цього напрямку мають ряд недоліків, насамперед вони не завжди гарантують коректне розв'язання. Тому задача автоматичного розв'язання логічних задач на основі машинного навчання є актуальною.

Аналіз останніх досліджень. Розв'язання комбінаторних логічних задач відноситься до одного з історично перших напрямів штучного інтелекту. Для їх розв'язання були створені спеціальні механізми. Одним з найбільш відомих механізмів логічного висновування є Prolog. Для перевірки ефективності його реалізації використовуються тестові задачі (benchmarks), якими є саме логічні задачі зазначеного типу. Колекції таких задач представлено на ресурсах [1, 2].

Інший підхід представлено в роботі [3]. Автор запропонував представлення логічної задачі у вигляді набору булевих виразів. На їх основі будуються і заповнюються таблиці властивостей. Однак метод не дозволяє реалізовувати складні відношення значень властивостей логічної задачі.

У роботі [4] задача представляється таблицею зв'язків між значеннями властивостей. Автори запропонували універсальний алгоритм розв'язання логічних задач на основі розрахунків сум існуючих значень таких зв'язків. Однак цей метод враховує лише наявність зв'язку між атрибутами і не дозволяє призначити додаткові умови.

Сучасною тенденцією досліджень зі штучного інтелекту є спроби розв'язання таких задач на основі машинного навчання. В роботі [5] було представлено систему Sherlock, яка вирішує логічні задачі на основі парадигми об'єднання індуктивного навчання і логічного програмування. Для спрощення логічного висновування як правило задача формалізується в представлення, пристосоване для комп'ютерної обробки. В цій роботі задача представляється за допомогою запропонованої авторами мови з синтаксисом, схожим на Prolog. Одним з недоліків даної системи є залежність від умов задачі.

В роботі [6] наведено результати розв'язання логічних задач проекту bAbI [1] за допомогою нейронних мереж. Однак для деяких різновидів задач відсоток коректних відповідей є досить низьким, зокрема для задач Counting, Lists/Sets, Positional Reasoning, Path Find.

Тому необхідно розробити універсальний механізм автоматичного розв'язання логічних задач на основі машинного навчання.

Мета і задачі. Метою роботи є створення обчислювальної структури, яка розв'язує логічні задачі на основі навчання за їх формалізованим описом. Для виконання даної мети необхідно було вирішити такі завдання:

1. Виконати формалізацію логічних комбінаторних задач.
2. Розробити спеціальну мережу для навчання розв'язанню логічних задач.
3. Розробити алгоритм навчання.
4. Провести обчислювальні експерименти на тестових задачах.

Формалізація логічної задачі. Логічна задача складається з набору властивостей, кожна з яких має набір можливих значень. Нехай задача складається з n властивостей, кожна з яких має m значень. Наприклад, для задачі Ейнштейна: $n = 6$ (номер будинку, його колір, національність мешканця, його улюблені напої, цигарки та тварина), $m = 5$ для всіх властивостей (номери будинків — 1, 2, 3, 4, 5; кольори будинків — білий, жовтий, зелений, червоний, синій; національності — англієць, данець, німець, норвежець, швед; напої — вода, кава, молоко, пиво, чай; цигарки — Dunhill, Marlboro, Pall-Mall, Philip Morris, Rothmans; тварини — кішка, кінь, пташка, риба, собака).

Таким чином, кожному властивості P_i (property) можна представити множиною її значень v_i^j (value):

$$P_i = \{v_i^j\},$$

де P_i — i -та властивість, $i = 1..n$, v_i^j — значення j властивості i , $i = 1..n$, $j = 1..m$.

Для задачі Ейнштейна: P_1 — номер будинку, P_2 — колір будинку, ..., P_6 — тварина. Наприклад, $P_3 = \{\text{англієць, данець, німець, норвежець, швед}\}$.

Кожна k -та можлива комбінація значень властивостей C_k (combination), матиме вид:

$$C_k = (x_k^1, x_k^2, \dots, x_k^n), \quad (1)$$

де k — номер поточної комбінації, v_i^k — можливе значення i -ї властивості, $v_i^k \in P_i$.

Наприклад, $C_1 = (1, \text{білий, англієць, вода, Dunhill, кішка})$, $C_2 = (1, \text{білий, англієць, вода, Dunhill, кінь})$, та інші.

Максимальна кількість комбінацій без урахування додаткових умов:

$$q = m^n, \quad (2)$$

де n — кількість властивостей задачі, m — кількість значень властивостей задачі

Рішенням задачі буде множина комбінацій, що не суперечать всім умовам задачі:

$$S = \{C_1, C_2, \dots, C_o\}, \quad (3)$$

де o — кількість коректних комбінацій значень властивостей, $o \leq q$.

Для вирішення задачі використовуються умови відношень між двома значеннями властивостей:

$$v_{i_1}^{j_1} \prec rel \succ v_{i_2}^{j_2} \quad (4)$$

де $v_{i_1}^{j_1}$ — значення j_1 властивості i_1 , $j_1 \in 1..m$, $i_1 \in 1..n$, $v_{i_2}^{j_2}$ — значення j_2 властивості i_2 , $j_2 \in 1..m$, $i_2 \in 1..n$, $\prec rel \succ$ — визначене відношення між властивостями з зазначеними значеннями.

При формалізації задачі всі зв'язки предметної області зводяться до тверджень про наявність/відсутність відповідного зв'язку, що позначається $\prec \epsilon \succ$ / $\prec ne \epsilon \succ$. Наприклад: $1 \prec \epsilon \succ$ *білий*, $2 \prec ne \epsilon \succ$ *зелений*.

Обчислювальна структура розв'язання логічних задач. Для розв'язання логічних задач було розроблено спеціальну структуру — мережу зв'язків. Мережа представляє собою n -мірну «решітку» вузлів. Розмірність мережі n дорівнює кількості властивостей задачі.

Приклад мережі в початковому стані для 2-х властивостей, кожна з яких має по 3 значення, зображено на рисунку 1. Цей приклад відображає частину задачі Ейнштейна для трьох будинків з номерами: 1, 2, 3, та можливими кольорами: синім, червоним, зеленим.

Вхідним шаром мережі зв'язків є вузли, які представляють собою значення властивостей задачі. На рисунку 1 властивість «номер будинку» зі значеннями 1, 2, 3 представляється вузлами $v_1^1 - v_1^3$ відповідно, а властивість «колір будинку» зі значеннями синій, червоний, зелений — вузлами $v_2^1 - v_2^3$.

Після визначення вхідних шарів необхідно побудувати внутрішню структуру мережі, яка містить обчислювальні вузли.

Кожен обчислювальний вузол фактично представляє C_k комбінацію значень всіх властивостей (2). Загальна кількість вузлів складає q (3). На кожному етапі обчислювальний вузол містить власне значення Nv (Node value), отримане при ініціалізації, або внаслідок навчання. Вузол може приймати значення 1, тобто комбінація C_k можлива, або 0 — комбінація неможлива.

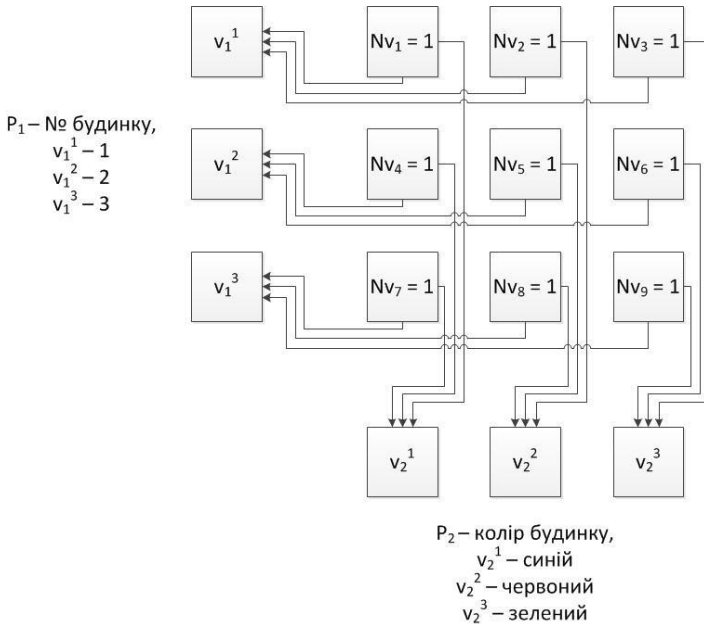


Рис. 1. Приклад мережі зв'язків

Приклад наведено для двомірної мережі зв'язків для наочності графічного представлення. Але принципи побудови можна розповсюдити на n -мірну мережу.

В алгоритмі використовується поняття $(n - 1)$ -мірного шару. Такий шар представляє деякий зріз з фіксованим значенням однієї властивості. В наведеному прикладі $(n - 1)$ -мірним шаром для фіксованого значення v_1^2 буде набір вузлів зі значеннями Nv_4, Nv_5, Nv_6 , для v_1^1 — зі значеннями Nv_1, Nv_4, Nv_7 . Виходячи з геометричних представлень $(n - 1)$ -мірний шар мережі при:

- 1) $n = 2$, є рядком або стовбцем;
- 2) $n = 3$, є площиною;
- 3) $n = 4$, є паралелепіпедом.

Запропонована мережа може мати будь-яку розмірність, однак для випадків $n > 3$ це важко інтерпретувати геометрично.

Для отримання розв'язку задачі побудована мережа повинна пройти навчання.

Навчання мережі зв'язків. Навчання полягає у встановленні значень Nv для всіх вузлів мережі. Критерієм закінчення навчання є відсутність змін значень Nv всіх вузлів по відношенню до поперед-

ної епохи. Епохою вважається представлення мережі всіх навчальних прикладів з подальшим уточненням значень вузлів. Навчальним прикладом є відношення між значеннями двох властивостей (4).

Представлення прикладу полягає в послідовному перерахунку значень всіх вузлів мережі. Перерахунок значення поточного вузла здійснюється за таким алгоритмом:

1. Визначення кількості збігів значень властивостей в поточній комбінації C_k з поточним прикладом.
2. Оновлення значення вузла:

$$N_v^i = N_v \times F(S, R),$$

де N_v^i — нове значення вузла, N_v — значення, встановлене після представлення попереднього прикладу, S — кількість збігів значень властивостей, R — відношення між значеннями властивостей, $F(S, R)$ — функція, яка залежить від кількості збігів значень властивостей і визначається в таблиці 1.

Таблиця 1

Значення функції оновлення

Вхідні параметри		Значення функції
R	S	F
$\langle \epsilon \rangle$	0	1
$\langle \epsilon \rangle$	1	0
$\langle \epsilon \rangle$	2	1
$\langle \text{не } \epsilon \rangle$	0	1
$\langle \text{не } \epsilon \rangle$	1	1
$\langle \text{не } \epsilon \rangle$	2	0

Після представлення всіх прикладів здійснюється *уточнення значень мережі* за таким алгоритмом:

1. Виокремлення всіх можливих $(n - 1)$ -мірних шарів.
2. Обчислення суми встановлених значень всіх вузлів поточного s -го шару Sl_s (Sum of layer).
3. Оновлення значень за схемою:

3.1. Якщо сума $Sl=1$ — формується додаткова навчальна вибірка зі всіх відношень « ϵ », які відображають поточний вузол:

$$x_k^i \langle \epsilon \rangle x_k^j, i = 1..n, j = 1..n, i \neq j,$$

де k — номер поточного вузла, n — кількість властивостей задачі.

Після цього здійснюється представлення всіх цих прикладів за вищеописаним алгоритмом.

- 3.2. Якщо $SI \neq 1$ — здійснюється обробка наступного $(s + 1)$ -го шару поточного вузла.
4. Перевірка закінчення алгоритму уточнення:
 - 4.1. Якщо є необроблені шари поточного вузла, здійснюється обробка наступного шару — повернення до пункту 2 поточного алгоритму.
 - 4.2. Якщо всі шари поточного вузла оброблено, здійснюється перехід до обробки наступного вузла — повернення до пункту 1.

Навчання відбувається за схемою:

1. Ініціалізація вузлів мережі: в кожному вузлі встановлюється значення 1.
2. Навчання за поточною епохою: представлення всіх навчальних прикладів з подальшим уточненням мережі за кожним з них.
3. Перевірка закінчення навчання: якщо умова закінчення навчання не виконується — перехід до наступної епохи. В іншому випадку — отримання розв'язку задачі у вигляді (5).

Обчислювальні експерименти. Для апробації запропонованого алгоритму було використано представлення n -мірної мережі зв'язків у вигляді одновимірного масиву. Представлення n -мірності організовано за допомогою вирахування індексів.

На рисунку 2 наведено фрагмент інтерфейсу програмної системи з формалізацією задачі Ейнштейна за властивостями та їх значеннями.

Номер будинку	
Колір	
Напій	
Тварина	
Цигарки	
Національність	

Назва значення	Зелений
<input type="button" value="Створити"/> <input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>	

Білий
Жовтий
Зелений
Червоний
Синій

<input type="button" value="Створити"/>	<input type="button" value="Редагувати"/>
---	---

Рис. 2. Формалізація задачі Ейнштейна

Навчальна вибірка для задачі Ейнштейна представлена на рисунку 3.



Рис. 3. Навчальна вибірка для задачі Ейнштейна

Розв'язок задачі представлено на рисунку 4.

Номер будинку	Колір	Напій	Тварина	Цигарки	Національність
1	Жовтий	Вода	Кішка	Dunhill	Норвежець
2	Синій	Чай	Кінь	Rothmans	Датчанин
3	Червоний	Молоко	Пташка	Pall Mall	Англієць
4	Зелений	Кава	Риби	Marlboro	Німець
5	Білий	Пиво	Пес	Phillip Morris	Швед

Рис. 4. Розв'язок задачі Ейнштейна

У випадку, якщо задача має декілька розв'язків, система видає рішення як перелік коректних комбінацій значень параметрів S_k . Такий приклад з варіантами розв'язання задачі Ейнштейна при видавленні першої умови наведено на рисунку 5.

Номер будинку	Колір	Напій	Тварина	Цигарки	Національність
1	Жовтий	Вода	Кішка	Dunhill	Норвежець
1	Жовтий	Вода	Риби	Dunhill	Норвежець
1	Жовтий	Вода	Пес	Dunhill	Швед
1	Жовтий	Чай	Кішка	Dunhill	Датчанин
1	Жовтий	Чай	Риби	Dunhill	Датчанин
1	Зелений	Кава	Кішка	Marlboro	Німець
1	Зелений	Кава	Кішка	Rothmans	Норвежець
1	Зелений	Кава	Пташка	Pall Mall	Норвежець
1	Зелений	Кава	Риби	Marlboro	Німець
1	Зелений	Кава	Риби	Rothmans	Норвежець
1	Зелений	Кава	Пес	Rothmans	Швед
1	Червоний	Вода	Кішка	Rothmans	Англієць
1	Червоний	Вода	Пташка	Pall Mall	Англієць

Рис. 5. Можливі комбінації для задачі Ейнштейна з неповною навчальною вибіркою

Розроблене програмне забезпечення реалізовано на платформі .Net Framework 4.6, мовою C#. Тестування та апробація системи відбувалися під керуванням операційної системи Windows 10 x64.

Для експериментів було обрано задачі, які є традиційними тестами в системах логічного програмування та продукційних системах: задача Ейнштейна, розміщення за столом. Крім того, було розв'язано задачі з ресурсу bAbI [1] таких класів: two supporting facts, two arguments relations, positional reasoning.

В таблицю 2 зведено інформацію з тестових задач, а також характеристики їх розв'язання.

Таблиця 2

Результати обчислювальних експериментів

№	Назва задачі	Параметри задачі		Час розв'язання, с	Кількість епох	Кількість коректних комбінацій o
		Кількість властивостей n	Кількість значень m			
1	Задача Ейнштейна	6	5	5.71	6	5
2	Розміщення за столом	3	4	1.12	3	4
3	Розміщення за столом	4	3	1.53	3	3
4	bAbI, two supporting facts	2	3	0.76	3	3
5	bAbI, two arguments relations	2	3	0.83	3	3
6	bAbI, positional reasoning	3	4	1.78	4	4

Таким чином, представлено механізм розв'язання логічних комбінаторних задач. Розв'язання кожної задачі відбувається шляхом автоматичного створення мережі зв'язків з її подальшим навчанням до отримання розв'язку.

Висновки.

1. Розроблено формалізацію логічних комбінаторних задач як набору властивостей та їх значень.
2. Розроблено представлення задачі у вигляді мережі зв'язків для навчання розв'язанню логічних задач.
3. Розроблено алгоритм навчання.
4. Експериментально доведено ефективність запропонованого способу.

Перспективою подальших досліджень є створення системи з лексико-синтаксичним аналізатором для автоматичного представлення властивостей, їх значень та відношень між ними.

Список використаних джерел:

1. bAbI — Facebook research // Facebook research. — 2015. — Access mode: <https://research.fb.com/downloads/babi/>.

2. Einstein's riddle and grid puzzles. — 2012. — Access mode: <http://brainden.com/einsteins-riddles.htm>.
3. Filho W. Solving the Zebra Puzzle with Boolean Algebra / W. Filho. — 2017. — Access mode: <https://code.energy/solving-zebra-puzzle/>.
4. Application of «Einstein's riddle» in solving construction machine allocation problems / B. Dasović, M. Čorak, M. Galić, U. Klanšek // E-gfos. — 2016. — P. 12–22.
5. Zhang Q. Sherlock — A Neural Network Software for Automated Problem Solving. — 2011. — Access mode: <http://www.macs.hw.ac.uk/~ek19/QK.pdf>.
6. Towards AI-complete question answering: a set of prerequisite toy tasks / [J. Weston, A. Bordes, S. Chopra a.o.] // arXiv. — 2015. — Access mode: <https://arxiv.org/pdf/1502.05698.pdf>.

LOGICAL PUZZLES SOLVING BASED ON MACHINE LEARNING

The article proposes a method of solving logical puzzles on the basis of machine learning. The method is designed for the preliminary formalization of tasks in the form of description of properties and relations between them. Because each property has a set of possible values, the solution of the puzzle by the methods of search has a combinatorial complexity. With a large number of properties and their values, the time of the solving is rapidly increasing.

In recent years, a separate area of research in machine learning has been the solution to logical tasks of this type. However, existing solutions to this area have a number of shortcomings, first and foremost, they do not always guarantee a correct solution.

The paper presents a special network of connections for learning the solution of logical puzzles, as well as their formalization for the representation of this network. The network contains computing nodes that represent the relationship between properties, and the nodes of the input layers that specify the values of these relationships.

Every task is solved by automatically creating a network of links with its further training until the solution is obtained. The geometric interpretation of the n -dimensional network of bonds and its $(n-1)$ -dimensional layers is given. The formalization of the presentation of the study sample and the learning algorithm are presented. The mechanism of solving logical combinatorial problems is presented.

The article presents examples of tasks that are traditional tests in systems of logical programming and production (expert) systems, as well as tasks from the resource bAbI of such classes: two supporting facts, two argument relations, positional reasoning.

The efficiency of the proposed method has been experimentally proved.

The prospects of further researches, which are connected with the creation of a lexical-syntactic analyzer for automatic representation of properties, their values and relations between them, are determined.

The proposed method is universal and does not depend on the characteristics of the current task, such as the number of properties and their values.

Key words: *logical puzzles, machine learning.*

Отримано: 13.08.2019