

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Метод керування технічними об'єктами на основі інтеграції обробки сенсорних даних

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

КВРКІ.016035.20.01.07 ПЗ

Виконав: студент 2 курсу, група КІ1м-20-1


Підпис

Островський І.І.

Керівник: к. т. н, доцент кафедри кібербезпеки


Підпис

Орленко В.С.

Нормоконтролер ст. викладач кафедри кібербезпеки


Підпис

Мостовий С.В.

До захисту допускаю:

Зав. кафедри кібербезпеки, к.т.н., доц


Підпис

Клюц Ю.П.

7 12 2021_р.

Хмельницький, 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХОЛОГІЙ

Кафедра КІБЕРБЕЗПЕКИ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА ПІДГОТОВКИ МАГІСТРА

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц



“ 1 ” 09 2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Островському І.І.

Прізвище, ім'я, по батькові студента

Тема проекту (роботи) Метод керування технічними об'єктами на основі інтеграції обробки сенсорних даних

1. Керівник проекту (роботи) к.т.н., доц. Орленко В.С.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом № _____ ректора університету додаток № _____ від _____

2. Строк подання студентом проекту (роботи) на кафедру 1.12.2021


3. Вихідні дані до проекту (роботи) сенсорні давачі, нейрона мережа, складі технічні об'єкти

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Аналіз особливостей застосування технологій інтелектуального управління. Формалізація задачі інтелектуального управління. Вдоскоалення моделі нейронної мережі для визначення мети та результатів управління. Розробка адаптивного методу інтелектуального керування технічними об'єктами. Вдосконалення методу формування послідовності дій агента на основі інтеграції сенсорних даних та прецедентного підходу. Розробка алгоритмів та програмного забезпечення системи. Експериментальне дослідження розробленого підходу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Тема, мета магістерської роботи, задачі дослідження, наукова новизна, практична цінність, Формалізація задачі інтелектуального управління. Адаптивний підхід до організації керування складними технічними об'єктами. Метод формування послідовності дій. Загальний алгоритм функціонування агента.

Висновки

6. Консультанти розділів дипломного проекту (роботи)

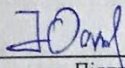
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Відповідальний за оформлення ДП	Мостовий С.В.		

7. Дата видачі завдання «1» лютого 2021 р.

КАЛЕНДАРНИЙ ПЛАН

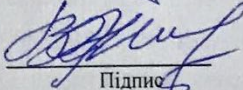
№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КРМ з керівником	2.02.2021	
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	2.03.2021	
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	1.04.2021	
4	Робота над розділом 2 – розробка моделей і методів для вирішення поставленої задачі	1.05.2021	
5	Робота над науковою статтею	1.06.2021	
6	Робота над розділом 3 – розробка алгоритмів та технологій, їх аналіз	1.09.2021	
7	Робота над розділом 4 – проектування ПЗ для вирішення поставленої задачі	1.10.2021	
8	Узгодження отриманих; оформлення пояснювальної записки згідно вимог	1.11.2021	
9	Оформлення графічної частини	11.11.2021	
10	Попередній захист КРМ	15.11.2021	
11	Захист КРМ на засіданні ЕК	8.12.2021	

Студент


Підпис

І.І. Острівський
Ініціали, прізвище

Керівник проекту (роботи)


Підпис

В.С. Орленко
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод керування технічними об'єктами на основі інтеграції обробки сенсорних даних.

Автор роботи: Островський І.І.

Керівник роботи: к.т.н., доц. Орленко В.С.

Пояснювальна записка: 81 с., 25 рис., 5 табл., 3 дод., 35 джерел.

СКЛАДНІ ТЕХНІЧНІ ОБ'ЄКТИ, ІНТЕЛЕКТУАЛЬНЕ КЕРУВАННЯ,
СЕНСОРНІ ДАНІ.

Мета кваліфікаційної роботи полягає в розробці підходу до інтелектуального управління технічними об'єктами, який базується на інтеграції сенсорних даних та їх обробці сучасними нейронними мережами.

Дана кваліфікаційна робота присвячена розробці адаптивного підходу до задачі управління технічними об'єктами, в якому поєднується обробка штучними нейронними мережами внутрішніх рецепторів агенту та сенсорних сигналів з зовнішнього середовища. Запропонований підхід до інтелектуального управління, що враховує класифікацію початкових умов та інтеграцію елементарних дій в послідовності з використанням ланцюгів Маркова.

07.12.2021

Дата



Підпис студента

ANNOTATION

a master's degree work of Ostrovskiy Illia
entitled «The controlling method of technical objects based on the integration of sensory
data processing».

Mentor: Viktoriia Orlenko

Total volume of work: 81 pages, 25 figures, 5 tables, 3 appendices, 35 references.

COMPLEX TECHNICAL OBJECTS, INTELLECTUAL CONTROL,
SENSORY DATA.

The purpose of the qualification work is to develop an approach to intelligent management of technical objects, which is based on the integration of sensory data and their processing by modern neural networks.

This qualification work is devoted to the development of an adaptive approach to the problem of technical object management, which combines the processing of artificial neural networks of internal receptors of the agent and sensory signals from the environment. An approach to intelligent control is proposed, which takes into account the classification of initial conditions and the integration of elementary actions in sequence using Markov chains/

07.12.2024
Date


Signature

ЗМІСТ

ВСТУП.....	3
1. ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ.....	8
1.1. Аналіз предметної області і виявлення наявних проблем і завдань.....	8
1.2. Порівняльний аналіз переваг та недоліків наявних рішень.....	15
1.3. Диспетчеризація та автоматичне планування.....	20
1.4. Постановка задачі.....	27
2. АДАПТИВНИЙ ПІДХІД ДО КЕРУВАННЯ І ЙОГО РЕАЛІЗАЦІЯ.....	29
2.1. Математична модель запропонованого підходу.....	29
2.2. Вдосконалення моделі згорткової нейронної мережі.....	35
2.3. Метод формування послідовності дій з використанням прецедентного підходу.....	42
2.4. Висновки.....	48
3. АЛГОРИТМИ ІНТЕЛЕКТУАЛЬНОГО КЕРУВАННЯ.....	50
3.1. Формування архітектури згорткової нейронної мережі.....	50
3.2. Створення адаптивного класифікатора об'єктів.....	56
3.3. Алгоритм формування дій робота на основі прецедентів.....	57
3.4. Висновки.....	62
4. РЕАЛІЗАЦІЯ АДАПТИВНОГО ПІДХОДУ.....	64
4.1. Оцінка функціонування уточненої моделі нейронної мережі.....	64
4.2. Розробка модулю планування.....	72
4.3. Висновки.....	75
ВИСНОВКИ.....	76
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	78
ДОДАТОК А Лістинг основного компонента модуля планування дій.....	82
ДОДАТОК Б Копія праць по результатам дослідження.....	88
ДОДАТОК В Копія презентації.....	101

ВСТУП

Робототехніка (від робот та техніка; англ. robotics) — прикладна наука, яка опікується проектуванням, розробкою, будівництвом, а також експлуатацією та використанням роботів, спеціалізованих комп'ютерних систем для їх контролю, сенсорного (на основі аналізу вихідних сигналів давачів) зворотного зв'язку і опрацювання інформації автоматизованих технічних систем (роботів) [1]. Перші роботи керувалися простими командами, запозиченими разом із приводами в станках. Для промислових зразків це стало можливим із-за крайнього детермінізму умов промислового виробництва. З розвитком науки і техніки, підтримкою інноваційних технологій робототехніка перетворилася в самостійну наукову сферу. Головною особливістю будь-якого механізму і робота є його корисність. Залежно від корисності автоматизованих технічних систем в тій чи іншій сфері життя заведено виділяти такі різновиди роботів:

- медичні;
- побутові;
- бойові;
- дослідні;
- промислові та будівельні;
- ігрові та ін.

Також їх можна розділити на керовані і автономні; мобільні та стаціонарні.

У сучасній медицині роботи виготовляються серійно, і без багатьох із них складні діагностичні процедури були б майже неможливі. У 1985 році робот Unimation Puma 200 приймав участь у взятті біопсії мозку в пацієнта, що стало значним проривом робототехніки у медицині. Пізніше, через 7 років спеціалізований робот ProBot зробив вперше у світі самостійно операцію

Однак, коли робототехніка розпочинає розповсюджуватися в інших областях з більшою невизначеністю та мінливістю зовнішніх впливів, із

програмного керування довелося перейти до управління від оператора, доповнюючи програмне управління диспетчерським управлінням.

Наступним етапом стала розробка на цій основі адаптивних систем управління з використанням подальших методів штучного інтелекту та переходу до парадигми багатоагентних систем. Останні, крім керування, використовувалися для виконання інших функцій роботів, таких як обробка сенсорної інформації та формування моделей взаємодії навколишнього середовища з оператором.

Багатоагентна система (англ. Multi-agent system) — це система, що утворена декількома взаємодіючими інтелектуальними агентами [2]. Ці системи можуть бути використані для вирішення таких проблем, які складно або неможливо розв'язати за допомогою одного агента чи монолітної системи. Прикладами таких завдань є керування критичною інфраструктурою, онлайн-торгівля, ліквідація надзвичайних ситуацій, та моделювання соціальних структур

Штучний інтелект (англ. artificial intelligence, AI) — розділ інформатики та комп'ютерної лінгвістики, що опікується формалізацією проблем та завдань, що подібні до дій, які виконує людина [3]. Подальша перспектива - технічний розвиток формалізації вмінь та творчих здібностей людини - креативності і її взаємодоповнюваність методами штучного інтелекту. Це дозволить повноцінно відтворити в конкретних прикладних областях розумові здібності людини, які реалізуються двома півкулями нашого мозку - лівим, де зосереджено переважно логічне мислення, і правим, що відповідає за творчі здібності та креативність людини. Зрозуміло, інтелектуальні системи управління значно розширюють можливості агентів з освоєння все більш складних операцій. Однак інтелектуальний агент як і раніше вимагає постійного спостереження з боку людини, особливо в зв'язку з можливістю виникнення аварійних та нештатних ситуацій.

Він не може тривалий час автономно функціонувати в нестационарному середовищі, тому що всі його дії строго формалізовані, і в цих умовах обов'язково вимагається інтуїція, креативність, творчість.

У системах, що працюють в режимі реального часу, особливо в критичній інфраструктурі, агент-робот повинен раціонально вирішувати поставлені перед ним завдання з мінімальними витратами необхідних ресурсів. Передбачається, що робот може виконувати деякі базові дії. Він намагається виконувати свої дії таким чином, щоб поставлена перед ним мета була досягнута. Використання високоточних методів багатокритеріальної оптимізації в реальному часі утруднено, тому агенти зазвичай вирішують задачу апроксимаційними методами з використанням проблемно-орієнтованих евристичних підходів [4].

Проблемами розробки агентів є використовувані ресурсомісткі алгоритми, висока складність і узгодженість моделей. Ці фактори мають негативний вплив на якість агентів, знижують їх обслуговування і продуктивність, а також збільшують витрати на розробку. Додатковим ускладненням є динамічний характер середовища, в якому повинен працювати агент, оскільки стан середовища може значно змінитися в процесі прийняття рішення. Агент повинен адаптуватися до таких змін в найкоротші терміни.

Крім того, досить складно розробити програму, яка була б планом дій для агенту. Тому що зазвичай існує багато способів досягнення цієї мети, і вам потрібно знайти найкращий можливий план. Пошук рішення займає багато часу, тому що це означає повний пошук. Очевидно, що це не раціонально, тому розробляються складні евристики, які можуть зменшити область пошуку до розумного розміру.

Аналіз матеріалів з проблеми менеджменту показує, що створення ефективної системи управління вимагає використання якісно нових підходів до обробки інформації, які повинні ґрунтуватися на виявленні особливостей дій у минулому та їх адаптації на основі ієрархічного представлення. дії. Найбільш перспективним напрямом створення таких систем є використання сучасних

нейронних мереж для класифікації прецедентів та формування нових дій на їх основі.

Метою кваліфікаційної роботи є розробка підходу до інтелектуального управління технічними об'єктами, який базується на інтеграції сенсорних даних та їх обробці сучасними нейронними мережами.

Для досягнення цієї мети необхідно вирішити такі завдання:

1. Аналіз характеристик застосування інтелектуальних технологій управління.

2. Формалізація завдання інтелектуального управління.

3. Уточнення моделі нейронної мережі для виявлення початкових умов та завдань управління.

4. Вдосконалення методу формування послідовності дій агента на основі інтеграції сенсорних даних та прецедентного підходу.

5. Розробка алгоритмів та програмного забезпечення системи.

6. Експериментальне дослідження розробленого підходу.

Об'єктом дослідження в кваліфікаційній роботі є процеси інтелектуального управління технічними об'єктами.

Предметом дослідження в кваліфікаційній роботі є моделі і методи процесу інтелектуального управління технічними об'єктами.

Наукова новизна роботи полягає в наступному:

1. Запропоновано адаптивний підхід до задачі управління технічними об'єктами, в якому поєднується обробка штучними нейронними мережами внутрішніх рецепторів агенту та сенсорних сигналів з зовнішнього середовища,

2. Вдосконалено метод інтелектуального управління, що враховує класифікацію початкових умов та інтеграцію елементарних дій в послідовності з використанням ланцюгів Маркова.

Методика проведення досліджень в кваліфікаційній роботі: методи системного аналізу, теорії множин, апарат мереж Маркова, математичної

статистики, теорії алгоритмів, теорії штучних нейронних мереж і методів імітаційного моделювання

Практична цінність кваліфікаційної роботи. Запропоновані алгоритми формування послідовностей дій можуть бути використані при управлінні агентами, роботами, логістичними системами, соціальними явищами.

1. Публікації. По темі магістерської роботи опубліковано 1 стаття (збірник праць ВІКНУ), та 2 тези доповідей на всеукраїнських конференціях (Тези доповідей Міжнародної науково-практичної конференції "Військова освіта і наука: сьогодення та майбутнє", Тези доповідей Всеукраїнської НПК молодих вчених, ад'юнктів, слухачів, курсантів та студентів "Молодіжна військова наука у Київському національному університеті імені Тараса Шевченка")

1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Агент - це метафора, спочатку прийшла зі штучного інтелекту [12]. Характеристики змінилися, коли основна увага була приділена багатоагентній системі, а не окремим агентам - новий і багатообіцяючий підхід до реалізації і розгортання розподілених систем. Важливо розуміти, що агенти поза ШІ широко використовуються, тому що інструменти і методи, призначені для одного, не обов'язково стануть у пригоді для іншого. В даний час багатоагентні системи мають два основних застосування крім своєї традиційної ролі

1. Проектування розподілених систем з використанням агентно-орієнтованої розробки програмного забезпечення (AOSE) (наприклад агентно-орієнтованих мов програмування).

2. Агентна симуляція і моделювання (ABMS).

Агенти - це суб'єкти, які спостерігають і реагують на своє оточення з метою досягнення своїх цілей [11]. Дві основні властивості агентів - це автономність та ситуативність. Автономія – це риса, яка означає, що агенти мають розумний цикл, який дає їм контроль над своїм внутрішнім станом та поведінкою. Оскільки агенти розташовані, тому що вони можуть "відчувати", сприймати та контролювати середовище, в якому вони працюють. Середовище може бути фізичним або віртуальним, і це розуміється агентами у вигляді (відповідних) даних. Автономія передбачає діяльність, тобто здатність агенту вживати заходів без необхідності робити це для досягнення своїх цілей.

Шохам запровадив агентно-орієнтоване програмування як "спеціалізоване об'єктно-орієнтоване програмування" [13]. Різниця між агентами та статичними об'єктами очевидна. Посилаючись на Wooldridge [10]: (1) об'єкти не контролюють свою поведінку (це можна узагальнити відповідно до популярної приказки «об'єкти роблять це безкоштовно а агенти роблять це тому, що хочуть»), (2)

об'єкти не показати гнучкість, є єдиний потік управління їх поведінкою та (3) а агенти за своєю суттю мають багатопотокові властивості.

Агентно-орієнтована парадигма також відрізняється від моделі акторів (Actor Model) , і від Active Objects [14], Насправді у суб'єктів немає ні цілей, ні завдань, навіть якщо їх специфікація включає процес. Замість цього агенти використовують свій раціональний цикл (як потік управління), можливо, разом з ключовими абстракціями переконань, бажань і намірів (як логіку), для реалізації алгоритмів, таких як процеси дії в їх середовищі, для досягнення своєї мети. Іншими словами, об'єкти «роблять» це безкоштовно, тому що вони дані, агенти - це процеси, і «роблять», тому що вони підходять для своїх завдань.

Середовище, в якому знаходяться агенти, не володіє автономією, типовою для агентів, але також може розвиватися завдяки внутрішнім процесам. Однак його діяльність не направлена на досягнення мети, що робить навколишнє середовище більш активними об'єктами.

Середовище біноміального агента формалізовано за допомогою таких підходів до моделювання, як модель і артефакт [23].

Система, що складається з декількох взаємодіючих агентів, називається багатоагентною системою. На цьому рівні загально визнано, що зазвичай визнається більше абстракцій, ніж організацій і взаємодій, які спрямовані на сприяння значущою і плідною координації автономних і різнорідних агентів в системі. Таким чином, агенти знаходяться не тільки у фізичному середовищі, а й у соціальному середовищі, де вони взаємодіють з іншими агентами і підкоряються правилам середовища, до якої вони належать. Нормативна багатоагентна система - це «багатоагентна система разом з нормативними системами, в якій агенти можуть вирішувати, з одного боку, слід вони явно викладеним нормам, а з іншого боку - нормативні системи визначають, яким чином агенти «можуть змінюватися. норми» [13]. Вплив на раціональний цикл агента полягає в тому, що агенти можуть виправдати соціальні наслідки своїх дій.

Немає єдиного визначення терміну агент і немає єдиного визначення мультиагентної системи (МАС). Найбільш важливі прийняті ухвали визначають загальні характеристики, такі як взаємодія агентів в системі: через загальне середовище, через певні структуровані повідомлення (онтології, протоколи взаємодії). Дійсно, МАС можна визначити щодо взаємодіючих одиниць, зокрема агентів. Комунікація може варіюватися від простих до складних форм. Проста форма комунікації полягає в тому, що вона обмежується простими сигналами з фіксованою інтерпретацією. Цей підхід був використаний Джордом [6] в багатоагентному плануванні, щоб уникнути конфліктів, коли план був спроектований декількома агентами. Складніша форма спілкування - використання структури робочого простору. Робочий простір - це загальний ресурс, який зазвичай ділиться на кілька областей в залежності від різних типів знань або різних рівнів абстракції при вирішенні проблем, які агенти можуть використовувати для читання або запису відповідної інформації для своїх дій. Інша форма зв'язку - це повідомлення, яке передається між агентами.

Автономність - ще одна ключова характеристика агентів при визначенні МАС, також відома як «самоорганізуються», яка дозволяє їм знаходити оптимальне рішення своїх проблем «без зовнішнього втручання». Головною особливістю, досягнутої при розробці багатоагентних систем, є гнучкість, оскільки багатоагентна система може бути додана, змінена і реконструйована без детального переписування програми. МАС також має тенденцію запобігати поширенню дефектів, самовідновлення і бути відмовостійкою, в основному через невідповідність компонентів.

Дуже важливо розрізняти автономні і автоматичні системи. Автоматичні системи повністю запрограмовані і працюють багато разів і незалежно від зовнішніх впливів або елементів управління. Їх можна охарактеризувати як саморегулюючий і здатні слідувати за зовнішнім цільовим шляхом і компенсувати невеликі відхилення, викликані зовнішніми перешкодами. Однак вона не може визначити шлях до конкретного місця призначення або вибрати місце

призначення, яке визначає його шлях. Тому що автономні системи, є самоврядними, тому що їм не потрібен зовнішній контроль, але вони керуються законами і стратегіями, які чітко розрізняють традиційні та багатоагентні системи. При використанні методів машинного навчання, автономні системи можуть розробляти для себе гнучкі стратегії, відповідно до яких вони вибирають свою поведінку.

Ми розглядаємо управління складними технічними об'єктами з точки зору парадигми агентно-орієнтованого моделювання штучного інтелекту.

Парадигма агентно-орієнтованого моделювання полягає в розробці парадигми, в якій агенти являють собою модель активного об'єкта в процесі, який моделюється. Ключове відмінність між агентно-орієнтованим підходом полягає в тому, що агенти є суб'єктами в процесі моделювання і мають бажання, мету і здатність виконувати дії, в той час як об'єктно-орієнтований підхід передбачає, що програма виконує операції з об'єктними об'єктами.

Багатоагентні системи реального часу - ефективний інструмент для моделювання складних процесів, в яких задіяна велика кількість активних автономних одиниць. Ці процеси включають міські транспортні потоки, логістичні системи, соціальні явища і епідемії. Методи багатоагентного моделювання також використовуються для пошуку і обробки даних в інформаційних мережах, автономних системах управління. Перспективним напрямком подальшого розвитку мультиагентних систем є розробка безпілотних літальних апаратів і автомобілів.

У багатоагентних системах реального часу агент повинен раціонально вирішувати поставлені перед ним завдання з мінімумом ресурсів. Використання високоточних методів багатокритеріальної оптимізації в реальному часі утруднено, тому агенти зазвичай вирішують задачу апроксимаційними методами з використанням проблемно-орієнтованих евристик.

Проблеми розробки агента - це алгоритми, які використовують багато ресурсів, висока складність та узгодженість моделей. Ці фактори досить

негативно впливають на якість агентів, знижують їх утримання та продуктивність, збільшують вартість розробки. Додатковим ускладненням є динамічний характер середовища, в якому агент повинен діяти, оскільки стан навколишнього середовища може значно змінюватися під час прийняття рішень. Агент повинен пристосуватися до таких змін у найкоротші терміни.

На рисунку 1.1 зображена функціональна схема агента-робота зі штучним інтелектом. Його основний та обов'язковий компонент - наявність розвиненої пам'яті, основа розумних дій робота та рівень його інтелекту. блок пам'яті пов'язаний з іншими системами, що обробляють інформацію, і він включає базу знань про зовнішнє середовище у вигляді його моделі та базу даних про це середовище, самого робота та операції, які вона може виконувати. Крім того, деякі неспеціалізовані оперативні бази знань та дані, пов'язані з центральною пам'яттю, можуть знаходитися в окремих роботосистемах.

База знань зовнішнього середовища містить інформацію, введену до початку роботи, та оперативну сенсорну інформацію, отриману в процесі сприйняття навколишнього середовища, коли робот виконує різні дії, а також у процесі своїх особливих пізнавальних дій в спробі вивчити це середовище. Сама інформація містить опис геометричних та інших фізичних характеристик об'єктів навколишнього середовища та їх взаємозв'язків. Цей опис має іскладну структуру у вигляді рівнів послідовної систематизації та узагальнення вихідної інформації. Наприклад, опис робочої зони маніпулятора містить набір площин цієї області та її частин, які відрізняються масштабом і точністю, та ступенем узагальнення первинної сенсорної інформації (виділення контурів, об'єктів, поверхонь, груп об'єктів, визначення різних властивостей об'єктів тощо).

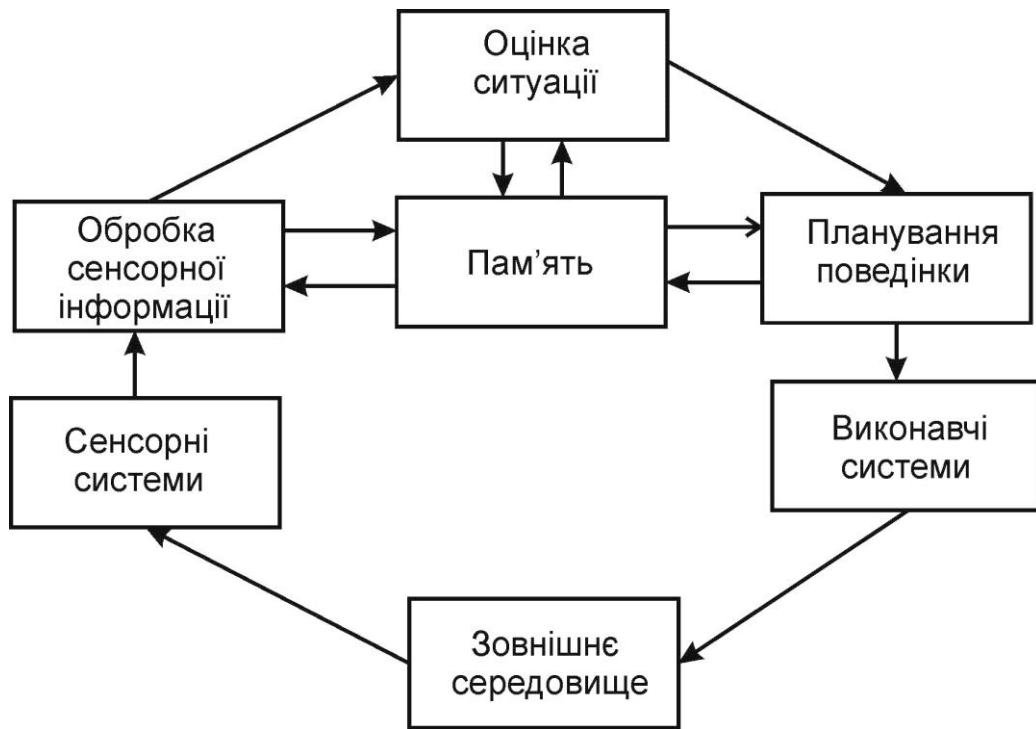


Рисунок 1.1 - Функційна схема інтелектуального агента-робота

Додавання до цих планів часу як параметру, дає зображення зовнішнього середовища в динаміці з урахуванням взаємодії його об'єктів між собою та з роботом. База знань зовнішнього середовища також містить правила, які дозволяють моделювати можливі зміни в цьому середовищі.

Всі інші блоки схеми мають деяку ієрархічну структуру, рівні якої з'єднані між собою вертикально знизу вгору у напрямку узагальнення інформації. У свою чергу, це показано на блок-схемі з'єднання в загальному багатоканальному випадку у вигляді з'єднань між тими ж рівнями по горизонталі.

Блок опрацювання сенсорної інформації отримує від блоку пам'яті екстраполяцію змін стану навколишнього середовища і передає йому виправлення цього стану на рівні прямого сенсорного зображення навколишнього середовища.

Блок оцінки стану та блок планування поведінки зчитують з блоку пам'яті поточну модель середовища та передають їй відповідно свою оцінку за певними критеріями та синтезований план управління рухом робота згідно із завданнями.

Завдання модулю оцінки ситуації також включає швидке коригування цілей та пріоритетів управління. Це найвищий рівень в ієрархії керування роботами.

Загалом, інтерфейсний блок може бути двосторонньо пов'язаний з усіма перерахованими функціональними одиницями. Окрім оператора -людини, він забезпечує спілкування з іншими командами, які працюють разом, включаючи іншу роботу.

У міру вдосконалення систем управління робототехнічними агентами перелік операцій, в яких домінують роботи, постійно розширюється. Однак навіть у сучасній промисловості все ще існує значна кількість технологічних операцій, які не повністю автоматизовані. Особливо це стосується непромислових робіт, таких як екстремальна робототехніка. Крім того, прогресуюче ускладнення технічних систем та функцій, які вони виконують, безперервно загострює цю ситуацію, тому людина продовжуватиме залишатися необхідною складовою роботизованих систем, головним чином вищим рівнем управління. Її завдання - керувати операціями, які неможливо автоматизувати, і навчити роботів виконувати ці операції з поступовим переходом до автоматичного режиму.

В цілому проблема створення "штучного інтелекту" виходить за межі робототехніки як глобальної проблеми, можна сказати, розвитку людської цивілізації. Той факт, що вона найгостріше зіткнулася з робототехнікою, пояснюється самою історією та сутністю робототехніки, для якої людина була еталоном з самого початку.

Робот має двосторонню інформаційну та енергетичну взаємодію із зовнішнім середовищем. І в цьому сенсі вона повинна бути схожою на живих істот і, отже, здатна до самонавчання та вдосконалення особистості.

Загалом, штучний інтелект використовується на робочих місцях для покращення якості таких завдань:

- опрацювання сенсорної інформації;
- оцінка зовнішнього середовища та прийняття рішень щодо відповідної поведінки;

- планування поведінки;
- керування переміщеннями для реалізації цих планів;
- створення інтелектуальних інтерфейсів для взаємодії робота з оператором та іншими роботами.

1.2 Порівняльний аналіз переваг та недоліків наявних рішень

Агентно-орієнтоване моделювання – це категорія обчислювальних моделей, які використовують динамічні дії, реакції та протоколи взаємодії між агентами у спільному середовищі для оцінки їх конструкції та продуктивності, а також для кращого розуміння їх поведінки та властивостей [15]. Агенти та багатоагентні системи-це об'єкти, які можна ефективно використовувати для моделювання складних систем, що складаються з об'єктів, що взаємодіють. Ось чому вони були прийняті для вивчення біологічних і хімічних систем, особливо коли системи стали занадто складними для аналітичних засобів, доступних у хімії, фізиці та математичній фізиці. Той факт, що агенти та багатоагентні системи є абстракціями з виконавчими партнерами, які підтримують багато інструментів, сприяв використанню агентної технології для моделювання біологічних та хімічних систем. Розмір модельованих систем та високий рівень точності модельованих явищ потребують спеціальних інструментів, які дозволяють експерту описати моделювання з невеликими знаннями або не вдаючись до інженерних тем, що стосуються розподілених систем. Агентсько-орієнтовані технології надають такі інструменти та допомагають експертам у галузі створювати ефективні розподілені системи з мінімальним акцентом на проблемах розподілених систем. Слід зазначити, що навіть якщо є спеціальні інструменти, існують загальноприйняті інструменти, призначені для створення програмного забезпечення, орієнтованого на агентства, в рамках агентно-орієнтованого моделювання. Зокрема, існує низка агентних інструментів для підтримки моделювання та моделювання складних та / або розподілених систем [16]:

Netlogo - це багатоагентне програмне середовище для моделювання, що дозволяє моделювати природні та соціальні явища. Це особливо добре для моделювання складних систем реального часу. Фактично, інженери можуть давати вказівки сотням або тисячам «агентів», які працюють незалежно один від одного. Це дає можливість вивчити взаємозв'язок між поведінкою мікрорівнів індивідів та моделями макрорівню, що виникають у результаті їх взаємодії. Вона поставляється з великою бібліотекою існуючих моделей, як загальних, так і традиційних, які можна використовувати та змінювати в різних областях, таких як суспільствознавство та економіка, біологія та медицина, фізика та хімія, математика та інформатика.. HubNet додає новий вимір до NetLogo, дозволяючи моделювати не тільки згідно з правилами, але і за безпосередньої участі людини.

FLAME: Це універсальна система моделювання поведінки агентів, яка може використовуватися для розробки додатків у багатьох сферах. Моделі створюються на основі обчислювальної моделі, яка називається кінцевими (розширеними) автоматами стану. Фреймворк автоматично генерує програми моделювання, які можуть ефективно перевіряти моделі на суперкомп'ютерах. Він виробляє комплексні програми, орієнтовані на агентів, які можна компілювати та запускати на більшості комп'ютерних систем-від ноутбуків до суперкомп'ютерів. Крім того, FLAME надає бібліотеку моделей, яка являє собою набір порівняно простих моделей, які ілюструють використання FLAME у різних додатках.

AnyLogK: це інструмент моделювання, який підтримує всі сучасні методології моделювання: системна динаміка, агентно-орієнтоване моделювання та процесно-орієнтоване моделювання (дискретні події). Його візуальне оточення значно прискорює процес розробки.

Repast: Repast Srnte - це безкоштовний програмаз відкритим вихідним кодом, передовим моделюванням та моделюванням платформ, яка розвивалася протягом багатьох років. Repast Smphony-це проста в освоєнні інтерактивна система моделювання на основі Java, призначена для використання на робочих станціях та невеликих обчислювальних кластерах. Розширена версія має назву

Repast for High Performance Computing, що є недостатньо розвинутою та орієнтованою на експертів системою, орієнтованою на C ++, призначеною для використання у великих обчислювальних кластерах та суперкомп'ютерах.

Jason: Це інтерпретатор для розширеної версії AgentSpeak, яка була однією з найвпливовіших абстрактних мов на основі архітектури BDL. Jason реалізує семантику цієї мови. Відомості про переконання використовуються для отримання інформації на рівні цілі та анотацій на етикетках плану. Один з найвідоміших підходів до розвитку когнітивних агентів - це архітектура BD! (Переконання-побажання-наміри). Це надає можливість запускати багатоагентну систему, розповсюджену по мережі.

Framsticks - це проект моделювання 3D життя. Можуть бути проведені різні типи експериментів, включаючи просту оптимізацію, відкриту та спонтанну еволюцію, різні групи генів та популяцій, моделювання видів та екосистем. Користувачі цього програмного забезпечення працюють в галузі еволюційних обчислень, штучного інтелекту, нейронних мереж, біології, робототехніки та моделювання, когнітивної науки, нейронауки, віртуальної реальності, медицини, філософії, графіки та мистецтва.

Це перелік найбільш популярних інструментів моделювання, які можуть бути використані для симуляції та аналізу складних систем шляхом застосування агентно-орієнтованої парадигми.

Інтелектуальна поведінка вимагає адекватної інформації про зовнішнє середовище, в якому працює робот - агент, та його становище в цьому середовищі. Це визначає такі вимоги до сенсорної підтримки розумного робота:

- складне обладнання з різними типами датчиків,
- інтелектуальна обробка первинної інформації з сенсорів для формування моделі середовища.

За властивостями та параметрами сенсорних систем в роботехніці їх можна розділити на три групи:

1) системи, що дають загальну картину навколишнього середовища з подальшим виділенням окремих його об'єктів, які важливі для виконання роботом своїх функцій;

2) системи, які визначають різні фізико - хімічні властивості довкілля та його конкретних об'єктів;

3) системи, які визначають координати розташування робота та параметри його руху, включаючи інформацію про координати щодо об'єктів у зовнішньому середовищі.

Сенсорні системи до першої групи включають системи технічного зору та різні типи локаторів та далекомірів.

Друга група датчикових систем є найрізноманітнішою: вимірювачі щільності, температури, оптичних властивостей, геометричних розмірів, хімічного складу тощо.

Третя група сенсорних систем визначає параметри самого робота: географічні координати в просторі, рух і швидкість навіть по відношенню до окремих об'єктів навколишнього середовища, поки контакт з ними не закріпиться.

В роботі всі ці сенсорні системи насамперед зосереджені на підтримці двох виконавчих систем: керування та руху. Це визначає основні вимоги до датчикових систем: дальність, точність, швидкість тощо.

Сенсорні системи, що обслуговують маніпулятори, також утворюють дві групи: системи, що є частиною ланцюга управління маніпулятора, і рецепторну систему його роботи. Ці останні системи часто включають системи технічного зору та датчики сили, розташовані на робочому тілі маніпулятора.

Системи датчиків, що використовуються в системах переміщення роботів, поділяються на системи, що забезпечують навігацію в просторі, і системи, що забезпечують безпеку руху (запобігання зіткненням з перешкодами, перекиданням на схилах, впливом неприйнятних зовнішніх умов тощо).

Технічні штучні нейронні мережі [8,10]: основний і незмінний засіб роботи з такою інформацією, хоча б тому, що вони, хоча й досі дуже спрощені, але

аналогічні "елементарній основі" мозкової діяльності. Накопичення, зберігання та обробка інформації зображення можуть бути реалізовані за допомогою нейронних мереж, які базуються на формальних нейронних мережах з традиційною пороговою логічною обчислювальною базою. Однак традиційні версії нейронних мереж, такі як багат шарові персептрони, мережі Хопфілда або Кохонена, неефективні при роботі зі складними динамічними зображеннями з невизначеністю, що особливо актуально для систем управління роботами.

Останнім часом були розроблені більш перспективні інструменти нейронної мережі, такі як динамічні нейронні поля, які адаптовані для обробки інформації з динамічних зображень для вирішення проблем робототехніки. Нейрологічні інструменти також розробляються на основі формальних нейронних мереж з гібридними основами, які використовують нечітку та правдоподібну логіку. Починають розроблятися абсолютно нові, так звані нейроморфні, з подібною інформацією, які базуються на спеціалізованих мережах неформальних нейронів. Ці мережі імітують деякі функціональні компоненти мозку, такі як мозочок, гіпокамп та деякі частини кори головного мозку. Ці інструменти можуть ефективно вирішити проблему керування роботами, оскільки вони здатні обробляти образну інформацію так само, як це проходить в біологічній нервовій системі.

Згортова (конволюційна) нейронна мережа будується з локальних мереж, які моделюють окремі ділянки рецептивного поля, в яких виконується однакова процедура послідовного узагальнення відповідних компонентів вихідного зображення.

Пірамідальна нейронна мережа повторює паралельну ієрархічну пірамідальну структуру кори головного мозку людини, здійснюючи розпізнавання суто образним чином. Вона також зберігає топологічну прив'язку до вихідного піксельного зображення, і кожен нейрон у вихідному шарі є певним класом зображень [18].

Наведені приклади нейронних мереж призначені для безпосереднього сприйняття візуальних образів і дозволяють досить ефективно моделювати процеси образного сприйняття візуальної інформації в системі штучного розуму робота.

Розглянемо докладніше, як обробляється візуальна інформація в когнітронній нейронній мережі.

Когнітрон, як і зорова кора людського мозку, складається з декількох шарів, які реалізують різні рівні узагальнення. Нейрон у кожному шарі пов'язаний з обмеженою кількістю нейронів у попередньому шарі (зона зв'язку), а у вихідному шарі кожен нейрон реагує на все поле введення (сприйнятливий) мережі, виявляючи певне зображення.

Шари когнітронів і природна нейронна мережа містять збуджуючі та гальмівні нейрони. Збудливі нейрони в одному шарі мають тенденцію активувати нейрон, зв'язаний з ними в наступному шарі, а гальмуючі нейрони нейтралізують цю зміну. Величина збуджуючого нейронного виходу визначається співвідношенням зважених сум збудливого та гальмівного входів. Когнітрон вчиться без вчителя. Під час навчання мережа організовується шляхом зміни ваги з'єднань.

Останнім часом згорткові нейромережі досягли великих успіхів у розпізнаванні зображень. Важливо, що вони мають багато гнучких реалізованих опцій, і тому ми використовуємо їх модифікацію як частину системи управління.

1.3 Диспетчеризація та автоматичне планування

Диспетчеризація та автоматичне планування (*Automated planning and scheduling*, APS) - це область штучного інтелекту з виконанням стратегії або послідовності дій, як правило, для інтелектуальних агентів, безпілотних апаратів, автономних роботів [33]. Вирішення проблем у цій області є повним,

комплексним і оптимізованим у багатовимірному просторі, на відміну від класичних проблем керування.

У більшості середовищ із існуючими моделями програмування можна виконувати без доступу до Інтернету. Можна знайти та оцінити рішення для його реалізації. У незнайомих умовах стратегію часто потрібно переглянути в Інтернеті. Необхідно адаптувати моделі та політику. Програмні рішення часто вдаються до ітераційного методу проб і помилок, який є поширеним при створенні штучного інтелекту. Він включає динамічне програмування, посилене навчання та комбінаторну оптимізацію. Мови, які використовуються для опису програмування та програмування, часто називають мовами дії.

Типовий планувальник має три входи: опис початкових умов, опис мети та багато можливих дій, наведених формальною мовою, наприклад STRIPS [42]. Планувальник створює послідовність дій, які можна знайти в системі від початкового стану до стану, що задовольняє ціль. Альтернативною формою опису проблем планування є ієрархічна мережа завдань, де з множини завдань кожне може бути виконане за допомогою примітивної дії, або розбита на аналогічну підмножину завдань [25].

Беручи за основу опис можливих початкових станів проблеми, опис бажаних цілей, а також опис набору потенційних дій, задача планування полягає в пошуку плану, який гарантовано створює з будь-якого стану послідовність дій, що ведуть до мети.

Складність планування пропорційно залежить від спрощення використовуваних припущень. Декілька класів завдань планування можна визначити на основі властивостей, які є характерними для проблеми в багатьох вимірах.

Найпростіша проблема планування, більш відома як класична проблема планування, визначається наступним чином:

- єдиний відомий початковий стан,
- детерміновані дії,

що можна зробити лише один раз,
агент.

Найпоширеніші мови для представлення проблеми планування, такі як STRIPS і PDDL для класичного планування, засновані на змінах стану. Кожне з можливих станів світу - це присвоєння значень змінним стану і дії, які визначають, яка з цих змінних зміниться, коли відбудеться запланована дія [41]. Оскільки набір змінних станів створює набір станів, розмір яких у множині є експоненційним, програмування зазнає прокляття розмірності та комбінаторного вибуху.

Часове планування можна вирішити за допомогою методів, подібних до класичного планування [34]. Основна відмінність полягає в тому, що через можливість того, що кілька подій перекриваються в часі, тобто вони відбуваються паралельно, визначення стану має включати інформацію про поточний абсолютний час і те, наскільки просунулася продуктивність кожної діяльності. Крім того, при такому плануванні простір станів може бути нескінченним, на відміну від класичного планування або планування з явним часом. Планування ітераційного методу, але з використанням представлення функцій визначених значень для простору переконань, а не станів часу тісно пов'язане з проблемами планування. Планування часу також можна розуміти в термінах хронометрованих автоматів. Імовірнісне планування може бути розв'язане за допомогою ітераційних методів, таких як ітерація значень і ітерація політики, коли простір станів відносно малий.

Імовірнісне планування вирішується аналогічним чином за допомогою ітераційного методу, але з використанням представлення функцій визначених значень для простору переконань, а не їх станів.

Прикладом імовірнісного планування є марківський процес прийняття рішень MDP (Markov decision process) (Рис.1.2).

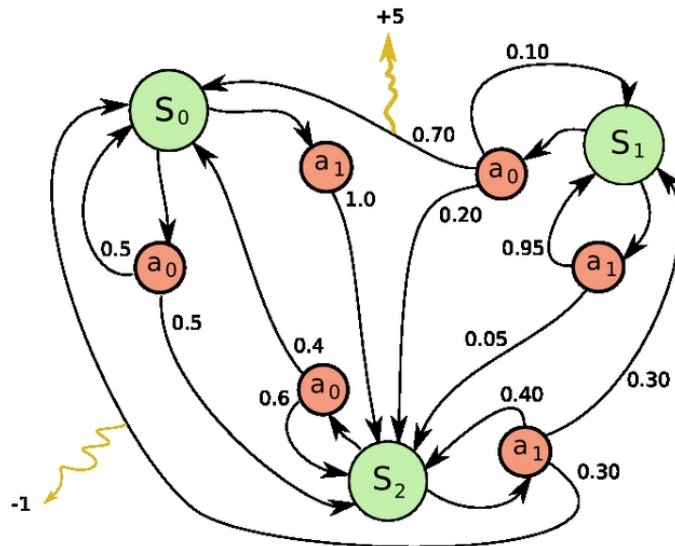


Рисунок 1.2 – Марківський процес прийняття рішень з 3 станами й 2 діями

У системі імовірносного планування BURIDAN розширення класичної моделі планування роботи з невизначеністю середовища здійснюється шляхом введення розподілів ймовірностей на початкові стани та результати дій [37]. Для цієї форми представлення є правильний і повний алгоритм розв'язання задачі та методи оцінки знайденого плану.

Програмні продукти APS активно використовуються в різних видах виробництва. Розробники такого програмного забезпечення постійно вдосконалюють автоматичні системи керування виробництвом, розширюють їх функціональні можливості та усувають невідповідності. Система APS є значно покращеною сучасною версією програмного забезпечення, призначеної для синхронного планування виробництва. Власник бізнесу може швидко створити графік роботи всіх можливостей і компонентів, задіяних у виробничому процесі.

Синхронне планування поділяється на два типи:

- планування виробництва та постачання;
- планування продукції.

Компанія, яка використовує APS для управління процесом постачання та виробництва, також має можливість контролювати велику кількість процесів, на які поширюються обмеження:

- кількість працівників, кваліфікований персонал;
- доступна потужність обладнання, максимальна швидкість його роботи;
- наявність сировини, напівфабрикатів, комплектуючих;
- площа приміщення, в якому відбуваються виробничі процеси.

Ці дані є основою для формування плану виробництва відповідно до умов і особливостей конкретного підприємства.

Системи APS, виконують програмування набагато швидше, ніж аналогічні програмні продукти, а робота розподіляється автоматично. Готовий до використання план виробництва можна отримати за кілька хвилин. Кількість помилок і збоїв мінімальна.

Отже, проблема має дві основні складові: середовище і агент.

Існує два основних підходи до представлення середовища: представлення в просторі станів і представлення в просторі проблем.

Перший підхід будує граф середини, в якому вершини відповідають станам, а дуги — діям, що дозволяє переходити з одного стану в інший.

В другому підході формується ієрархія підзадач, початкова складна задача розкладається на кілька простих, що в деяких випадках значно полегшує вирішення.

Представлення в просторі станів можна розглядати як окремий випадок представлення в просторі задач, коли одна з підзадач є елементарною: перехід до сусіднього стану.

У загальному випадку середовище, в якому діє агент, має власну поведінку, яка може бути відома повністю або частково. Крім того, в процесі реалізації плану агента, як правило, хвилює не реальний стан середовища, а очікуваний на основі наявної у нього інформації.

Для побудови плану та управління його виконанням необхідно побудувати формальний опис (модель) реального середовища. Оскільки в реальній ситуації побудувати повністю повну і правильну модель зазвичай неможливо або вкрай важко, необхідно побудувати якусь спрощену модель середовища, з якої агент формує і перевіряє план.

Основні методи, які використовуються для опису середовища, засновані на методах представлення знань: виробничі моделі, логічні вирази, семантичні мережі, структури фрейму.

В нашій кваліфікаційній роботі зупинимося на марківському середовищі. У марківському середовищі наступний стан системи залежить від поточного стану і не залежить від попередніх станів. У немарківському середовищі (середовищі пам'яті) часто доводиться зберігати інформацію про попередні стани.

Марковські процеси прийняття необхідних рішень з дискретним часом визначаються:

- короткостроковістю дії;
- повним спостереженням;
- недетермінованими імовірнісними діями;
- найбільшою функцією винагороди,
- використанням одного агента.

Другим фундаментальним компонентом у завданні планування є концепція агента, об'єкта, який діє в середовищі з наміром досягнення мети. Агент — це апаратна чи програмна система, що реалізована на основі концепцій, найбільш застосовних до людей, і має такі властивості: автономність, активність, реактивність, комунікативні навички.

Опис існуючих знань про середовище: формалізація представлення середовища в системі планування з урахуванням отриманих властивостей, взаємозв'язків і можливої невизначеності є одним з головних і найскладніших завдань не тільки в плануванні, але і в будь-якому штучному інтелекті системи.

Особливістю сучасних систем планування є активна двостороння взаємодія з підсистемою управління знаннями, тобто не лише використання наявних знань, а й поповнення бази новими знаннями, набутими в результаті навчання.

Основними способами опису знань, які використовуються при плануванні завдань [19-27,32], є:

- логіка предикатів;
- продукційні правила;
- фреймове уявлення;
- семантичні мережі;
- спеціальні мови планування.

Маючи відомі параметри навколишнього середовища та доступні моделі, планування можна виконувати в автономному режимі, навіть на етапі проектування. Можна знайти та оцінити рішення для його реалізації. З невідомими параметрами довкілля стратегія часто потребує перегляду на етапі реалізації плану. Моделі та моделі поведінки повинні адаптуватися. Рішення зазвичай ґрунтуються на методах проб і помилок, що часто використовується в штучному інтелекті, оскільки це вроджений метод людського мислення. Вони включають динамічне програмування, навчання з підкріпленням та комбінаторну оптимізацію. Мови, які використовуються для опису програмування та систем програмування, часто називають мовами дії.

З огляду на опис можливих початкових станів світу, опис бажаних цілей та опис багатьох можливих дій, завдання планування полягає в тому, щоб знайти план, який гарантує (для будь-якого початкового стану) складання послідовності дій, які призведе до одного з цільових станів.

Складність планування зумовлена використанням спрощуючих припущень. Проблеми в деяких задачах програмування виникають через властивості, які мають проблеми в інших середовищах.

1.4 Постановка задачі

Розглянувши підходи до керування технічними об'єктами можна зробити висновок, що вони є сильно формалізовані і спираються при цьому на формальну обробку предикатів. Вони мало інтегровані з реальним середовищем, з яким стикається агент.

Технічні штучні нейронні мережі - основний і незмінний засіб роботи з такою інформацією, хоча б тому, що вони, хоча й досі дуже спрощені, але аналогічні "елементарній основі" мозкової діяльності. Накопичення, зберігання та опрацювання інформації зображення можуть бути реалізовані за допомогою нейронних мереж, які базуються на формальних нейронних мережах з традиційною пороговою логічною обчислювальною базою. Однак традиційні версії нейронних мереж, такі як багат шарові перцептрони, мережі Хопфілда або Кохонена, неефективні при роботі зі складними динамічними зображеннями з невизначеністю, що найбільш актуально для систем управління роботами.

Згортова (конволюційна) нейронна мережа будується з локальних мереж, які моделюють окремі ділянки рецептивного поля, в яких виконується однакова процедура послідовного узагальнення відповідних компонентів вихідного зображення.

Тому метою кваліфікації роботи є розробка підходу до інтелектуального управління технічними об'єктами, який базується на інтеграції сенсорних даних та їх обробці сучасними нейронними мережами.

Для досягнення цієї мети необхідно вирішити такі завдання:

1. Аналіз характеристик застосування інтелектуальних технологій управління.
2. Формалізація завдання інтелектуального управління.
3. Уточнення моделі нейронної мережі для виявлення початкових умов та завдань управління.

4. Вдосконалення методу формування послідовності дій агента на основі інтеграції сенсорних даних та прецедентного підходу.
5. Розробка алгоритмів та програмного забезпечення системи.
6. Експериментальне дослідження розробленого підходу.

2 АДАПТИВНИЙ ПІДХІД ДО КЕРУВАННЯ І ЙОГО РЕАЛІЗАЦІЯ

2.1. Математична модель запропонованого підходу

Біоніка – це прикладна наука про застосування в різноманітних технічних пристроях і системах принципів організації, функцій, властивостей і структури живої природи, тобто форми існуючого живого в природі та його промислових аналогів [17]. Суть цього науково-технологічного рішення є перспективою для того, щоб запозичати в природі цінні ідеї та реалізувати їх у вигляді оригінальних конструкторських та дизайнерських рішень, а також нових інформаційних технологій. Сучасні інтелектуальні системи автоматизації мають можливість реалізації алгоритмів управління складними технологічними системами та об'єктами. Сучасні контролери, створені на основі мікропроцесорних пристроїв, та електронно-обчислювальних машин мають високу швидкість і надійність і можуть виконувати досить складні алгоритми керування. Вони здатні аналізувати та розпізнавати зміни в об'єкті керування та зовнішньому середовищі, яке діє на об'єкт порушення, прогнозувати поведінку як керованого об'єкта, так і системи керування, робити висновки на основі узагальнень, порівнянь, аналізу. і синтез, щоб створити контрольні ефекти. Вони також можуть навчатися під час роботи та використовувати старі знання в нових незнайомих ситуаціях.

В загальному опис завдання планування полягає в тому, що активний елемент (агент) виконує послідовність дій в деякому середовищі і прагне досягти поставленої мети.

У кожен проміжок часу середовище знаходяться в деякому стані, при цьому дії агента можуть змінюють стан середовища.

Завдання планування - знайти послідовність дій, які дозволяють агенту перевести систему з початкового стану в заданий цільовий стан.

У загальному випадку ціль може складатися з кількох станів, досягнення будь-якого з них означає досягнення мети. Також можливо, що жоден з цих станів

є недоступним. Формально в задачі планування дається система агент-середовище [16]:

$$M = (Q, A, q_0, \Gamma_M), \quad (2.1)$$

де Q - множина спостережуваних станів;

A – множина можливих дій;

q_0 – початковий стан;

I - початкові умови;

G - множина цільових станів;

$\Gamma_M : Q \times A \Rightarrow Q$ функція переходу, яка визначає для кожного стану $q \in Q$ і дії $a \in A$ наступний стан $q' = \Gamma_M(q, a)$.

Потрібно знайти план, який є впорядкованою множиною дій $P = \{ a_1, \dots, a_n \}$, який є суперпозицією функцій переходу $\Gamma_M(\Gamma_M(\dots \Gamma_M(\Gamma_M(q_0, a_1) a_2) \dots, a_{n-1}), a_n)$ і належить G при $q_0 \in I$.

Відповідно до формули 1.1 процес керування роботом виглядає так. Існують початкові умови $q_0 \in I$ для ініціалізації процесу керування. Якщо брати до уваги процес управління роботом-агентом, то це може бути, наприклад, розряд батареї робота або технічний зір робота при небезпечному швидкому переміщенні великого об'єкта в напрямку робота.

Початкові умови визначають мету операції робота $g \in G$. У першому випадку мета визначається як отримання заряду акумулятора. У другому випадку метою може бути переміщення робота на безпечну відстань від об'єкта загрози (зменшення розмірів об'єкта в полі зору робота).

Залежно від початкового стану та цілі керування необхідно знайти суперпозицію функцій переходу між станами Γ_M (впорядкована множина дій $P = \{ a_1, \dots, a_n \}$), яка призводить робота в кінцевий стан.

Тобто агент повинен визначити початковий стан. У разі заряду батареї внутрішні сенсори необхідні для визначення ступеня розряду батареї, в останньому випадку зовнішні - для організації технічного зору.

Далі необхідно визначити і класифікувати датчики індикаторів. В організмі людини це здійснюється за допомогою нейронних мереж, як із зовнішнього середовища, так і з індикаторів внутрішнього стану робота-агента.

Використовуючи адаптивний підхід, для ідентифікації початкового стану доцільно використовувати штучні нейронні мережі. На сьогоднішній день розроблено багато типів нейронних мереж. Нам потрібно вибрати серед них дуже схожі на обробку інформації людським мозком, і з їх успішною реалізацією в області обробки зображень. Для цього ми вибираємо згорткові нейронні мережі, які сьогодні досить успішно використовують Google, Facebook, Pinterest та багато інших.

На виході нейронної мережі отримуємо клас початкових умов, за якими можна сформулювати мету і вивести порядок роботи робота.

Тут необхідно робити логічні висновки і планувати дії. Тут також потрібно використовувати аналогію з розумними біологічними системами.

Однак сучасні нейронні мережі все ще не здатні повністю імітувати мислення і в першу чергу виконують завдання класифікації, тобто проводять імітацію роботи частин людського мозку в дуже обмеженому об'ємі. Нейронні мережі, які намагаються відобразити людський мозок більш повним або гіпотетичним способом, наприклад Selfridge Pandemonium [32].

На найнижчому рівні Selfridge Pandemonium розташовані демони даних, на яких покладено функцію сприйняття зовнішнього світу. Демони даних асоціюються з демонами розуміння; Кожному демону розуміння ставиться у відповідність певний клас чи поняття.

Кожен демон розуміння має визначити, наскільки цей шаблон схожий на впізнаваний шаблон. Чим вищий ступінь подібності, тим сильніший сигнал надсилається демону прийняття рішень. Останній обирає найсильніший сигнал. Ця схема часто модифікується таким чином, що обчислювальні демони розміщуються між демонами даних і демонами розуміння, які підкреслюють характеристики об'єктів, які розпізнаються.

Але навіть для роботи тієї ж системи контролю, яка використовує формалізовані знання, потрібен експерт.

Тобто завдання полягає в тому, щоб реалізувати на основі інформації, отриманої датчиками зовнішнього середовища та приймачами внутрішнього стану, сформувавши мету та послідовність дій системи.

Друге завдання в інтелектуальних системах управління - розробити послідовність елементних керуючих впливів на основі поставленої мети та перевірити успішність сформованих впливів.

Необхідно також виявити вихідний стан, визначити на його основі мету дій і сформувавши на їх основі послідовність дій. У разі заряду батареї внутрішні сенсори необхідні для визначення ступеня розряду батареї, в при визначені небезпечного об'єкту, що рухається на зустріч, зовнішні сенсори використовуються для організації технічного зору. В організмі людини це здійснюється за допомогою нейронних мереж, як із зовнішнього середовища, так і за показниками внутрішнього стану.

Сучасні нейронні мережі часто вирішують стандартні завдання класифікації, але методи планування дій відокремлені від проблеми класифікації і є високоформалізованими. Тому актуальним завданням є розробка підходу до інтеграції підсистеми планування з існуючими нейронними мережами.

Якщо розглянути дії інтелектуального робота-агента в небезпечній ситуації, коли великий об'єкт швидко рухається на нього, то нейронна мережа в першу чергу має бути налаштована на ідентифікацію великих об'єктів. Робот повинен розвернутися і якомога швидше відійти від джерела небезпеки, але при цьому необхідно врахувати низький заряд акумулятора робота. Інші параметри менше важливі і можуть бути оброблені пізніше.

При розробці адаптивного підходу ми будемо використовувати аналогії в діяльності людського мозку. Діяльність людського мозку ще далеко не розкрита, але деякі моменти загально відомі і вже використовуються в сучасних системах штучного інтелекту.

Ми формулюємо основні принципи адаптивного підходу до управління агентами:

1. Нейронна мережа здійснює прийом та класифікацію інформації як від зовнішніх датчиків, так і від внутрішніх сенсорів агента, щоб ідентифікувати початкові умови дії агента, мету дії та визначити на основі мети послідовність дій, що виконує агент.

2. Нейронна мережа може обробляти дані від різних типів датчиків (оптичних, акустичних тощо).

3. Існує перелік елементарних дій, що може виконувати агент.

4. Ці дії можуть виконуватися окремо або об'єднуватися в послідовності дій на основі досягнення мети.

5. Нейронна мережа зчитує характеристики послідовності дій через рецептори для організації зворотного зв'язку та ідентифікації поточного стану.

6. Компоненти штучної нейронної мережі можуть класифікувати різні показники об'єктів навколишнього середовища з різною швидкістю. Ви можете класифікувати розмір і швидкість об'єкта швидше, ніж вказувати, який об'єкт знаходиться в ієрархії, поступово деталізуючи оброблені показники.

7. Початкові умови і мета мають ступінь важливості, яка може змінюватися і безпосередньо впливати на порядок дій.

На рисунку 2.1 наведено загальну схему агента, який працює згідно запропонованого адаптивного методу з врахуванням принципів біонічного підходу.

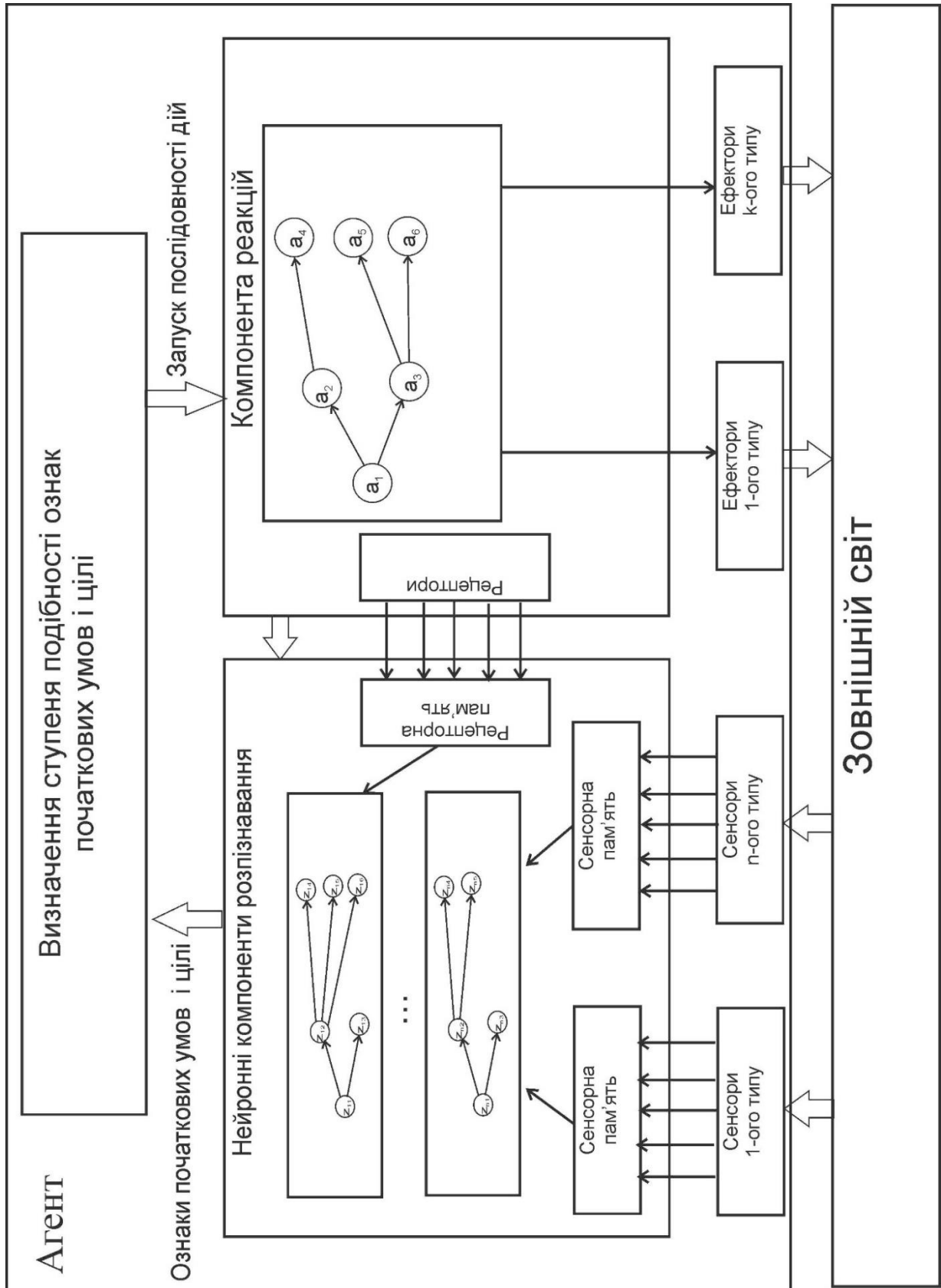


Рисунок 2.1 – Схема організації інтелектуального управління при адаптивному підході

2.2 Вдосконалення моделі згорткової нейронної мережі

Для обробки та класифікації сенсорних індикаторів в кваліфікаційній роботі вибрано згорткові нейронні мережі, які є одними з найуспішніших з точки зору розпізнавання та класифікації зображень.

Згортка нейронна мережа (ЗНМ) (CNN, convolutional neural network) — це тип багатошарової нейронної мережі прямого поширення, яка використовується для розпізнавання зображень.

Мережа отримала свою назву від операції під назвою згортка і часто використовується для обробки зображень. Цей тип нейронної мережі використовує три типи шарів: шар згортки (convolution), субдискретизації (subsampling, pooling), і повнозв'язний шар (fully-connected) [32]. Ще можна виділити вхідний шар: вхідне зображення і кілька колірних каналів.

Коротко розглянемо принцип роботи таких штучних нейронних мереж. На вхідне зображення накладається набір фільтрів із заданим розміром вікна (ядра). Тоді ми отримуємо деякі карти ознак, кількість яких відповідає кількості фільтрів. Ядро всіх фільтрів ініціалізується випадковими значеннями, в результаті кожен фільтр визначає певну властивість зображення.

Для зменшення розмірності зображення використовують шар субдискретизації. Останнім шаром виступає повнозв'язний шар, що формує вихід нейромережі.

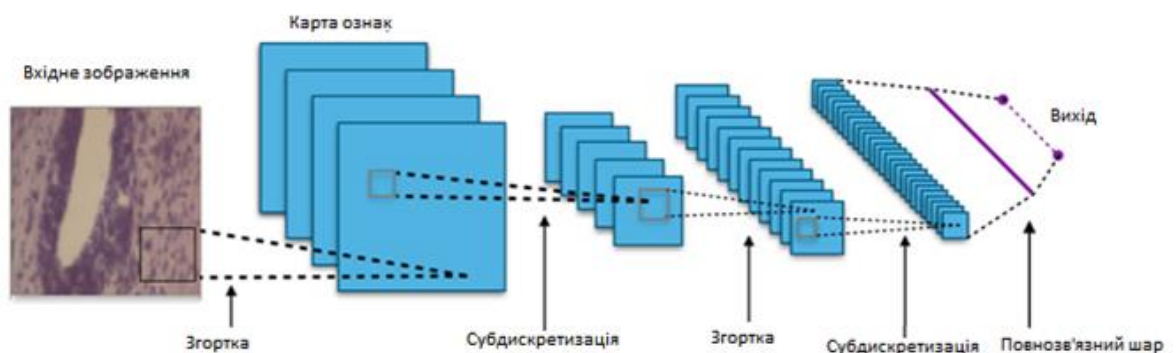


Рисунок 2.2 – ЗНМ

Для формування архітектури класифікаторів згорткових нейронних мереж використовуються наступні рішення:

- Перебір та пошук випадкових конфігурацій мережі: перевірка точності конфігурації кожної моделі на певному наборі даних. Цей метод є найбільш неефективним і рідко може бути використаний для вирішення прикладних завдань.

- Експертний підхід: налаштування параметрів мережі по попередньо успішно вирішених завданнях. Цей підхід використовується більшістю дослідників, які використовують ЗНМ як модель класифікації.

- Автоматичний вибір параметрів з оптимізацією: підбір значень параметрів мережі за допомогою алгоритмів оптимізації, таких як байєсівська оптимізація. Цей підхід використовується майже не використовується і часто може бути неефективним через збільшення часу, витраченого на його реалізацію.

- Проектування індивідуальної архітектури ЗНМ, оптимізованої для вирішення задач певного класу. Найчастіше цей підхід використовується в дослідженнях, безпосередньо пов'язаних з машинним навчанням і розробкою оптимальних моделей ЗНМ.

Аналізуючи наведені вище підходи, можна зробити висновок, що методи побудови архітектури ЗНМ часто є складними або неефективними, а тому для вирішення прикладних завдань дослідники часто використовують нестандартні рішення. Однак через характеристики даних конкретної прикладної задачі рішення з використанням експертного підходу може мати різну ефективність, яка може бути меншою, ніж коли та ж модель використовується в іншій задачі.

У цій кваліфікаційній роботі пропонується використати результати дослідження та досвід розробки найбільш ефективних ЗНМ, які існують на даний момент, для ефективного розробки власного підходу до формування високопродуктивної архітектури класифікатора на основі ЗНМ.

Розглянемо більш детально типи шарів.

Вхідний шар представляється у вигляді тривимірної матриці розміри якої

залежать від вхідного зображення - формула (2.2).

$$I_n = W \times H \times D, \quad (2.2)$$

де I_n - розмірність вхідного шару нейронної мережі,

H - висота зображення на вході,

W - ширина вхідного зображення,

D - глибина чи кількість колірних каналів вхідного зображення.

Згортковий шар ЗНМ є одним з основних і призначений для виділення особливостей зображення та їх перетворення, які, у свою чергу, далі на більш глибокі шари, використовуються для отримання більш складних ознак і в кінцевому підсумку визначення класу об'єкт для розпізнавання.

Основною характеристикою цього шару є фільтри - багатовимірні матриці ваг зв'язків нейронів згорткового шару з нейронами попереднього шару. Їх називають фільтрами, оскільки операція згортки, тобто отримання вихідного сигналу нейрона шару згортки, дуже схожа на фільтрації зображення: кожне значення сигналу нейрона попереднього шару, розташованого в певній області, множиться на відповідне значення ядра фільтра.

1x1	1x0	1x1	0	0	<table border="1"> <tbody> <tr> <td>4</td><td></td><td></td> </tr> <tr> <td></td><td></td><td></td> </tr> <tr> <td></td><td></td><td></td> </tr> </tbody> </table>	4								
4														
0x0	1x1	1x0	1	0										
0x1	0x0	1x1	1	1										
0	0	1	1	0										
0	1	1	0	0										

Рисунок 2.3 – Використання фільтра в ЗНМ

Результатом фільтрації виступає сума всіх отриманих добутків. Процес

створення сигналів нейронів згорткового шару зображено на рисунку 2.4.

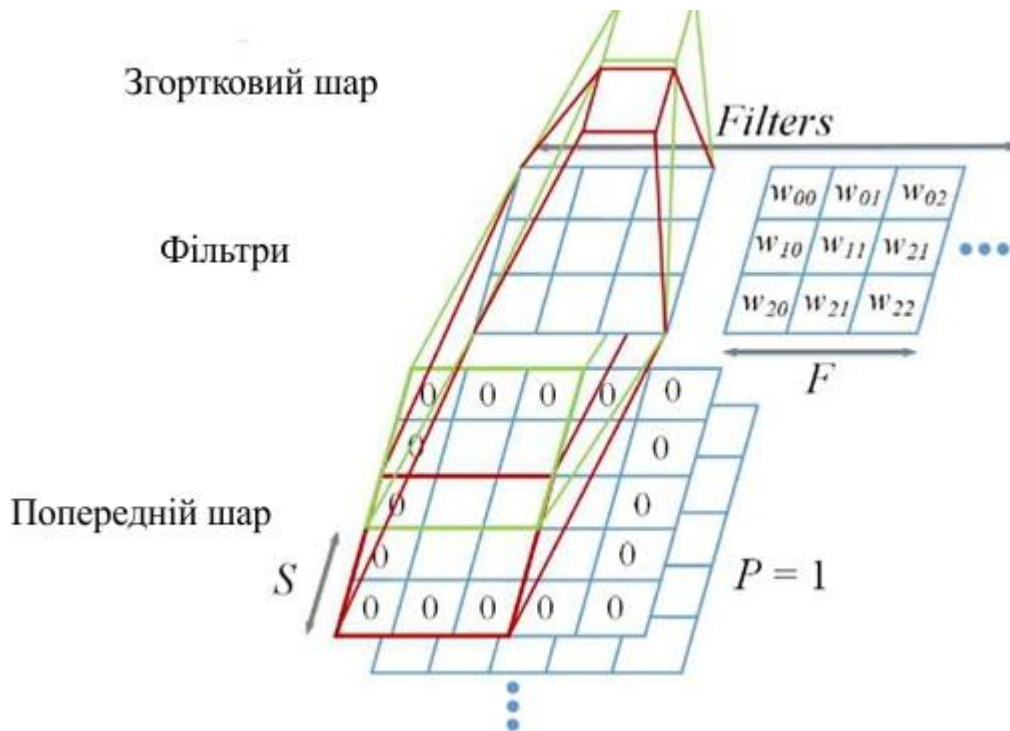


Рисунок 2.4 - Формування згорткового шару ЗНМ

Тип фільтра визначає, для яких характеристик зображення він використовується для (у процесі навчання мережі вагові коефіцієнти фільтрів змінюються, що впливає на роботу фільтру) Зазвичай використовуються квадратні фільтри розміру F . Кількість використовуваних фільтрів визначає глибину шару згортки.

Розмірність згорткового шару визначається за формулами (2.3) і (2.4).

$$W = (W_p - F + P * 2) / S + 1, \quad (2.3)$$

$$H = (H_p - F + P * 2) / S + 1, \quad (2.4)$$

де H і W - висота та ширина згорткового шару нейромережі,

W_p і H_p - ширина та висота попереднього шару нейромережі,

F - розмірність фільтрів згорткового шару,

P -(padding) кількість доданих нейронів з нульовими значеннями по краях попереднього шару, для проведення регулювання розмірності згорткового шару,

S - (stride) зміщення фільтру для формування сигналу наступного нейрона із згорткового шару.

Шар pooling (субдискретизації) згорткової нейронної мережі використовується для зменшення розмірності даних. Логіка роботи така: якщо деякі властивості (характеристики) вже були виявлені в попередній операції згортки, то таке детальне зображення більше не потрібне для подальшої обробки і «згортається» в менш детальне зображення.

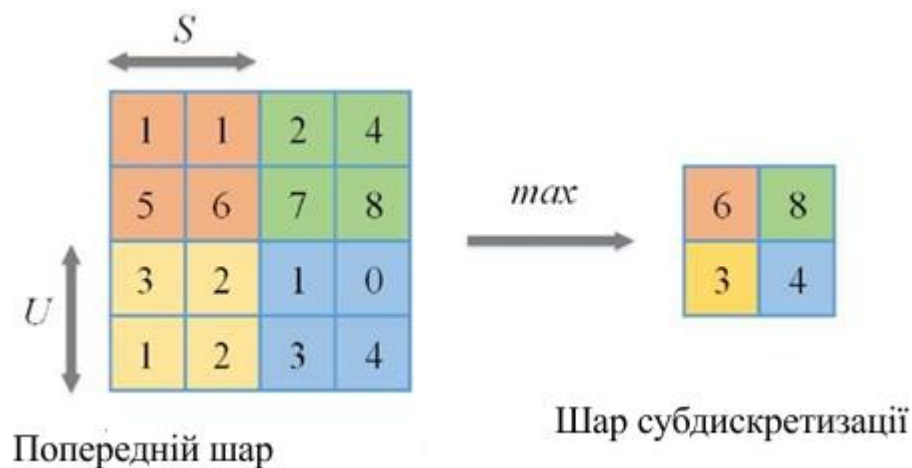


Рисунок 2.5 - Формування шару pooling за допомогою операції max

Розмірність цього шару визначається його параметрами та розмірами попереднього шару мережі, відповідно до формул (2.5) і (2.6).

$$W = (W_p - U) / S + 1, \quad (2.5)$$

$$H = (H_p - U) / S + 1, \quad (2.6)$$

де H і W – висота та ширина шару pooling,

W_p і H_p - ширина та висота попереднього шару,

U - розмір вікна,

S - зміщення вікна шару pooling.

Повнов'язаний шар (fully-connected) є одновимірним і в ньому кожен нейрон з'єднаний на всіх рівнях з кожним нейроном попереднього шару, якщо попередній шар має параметр глибини. Він призначений для перетворення сигналів, одержаних на згорткових рівнях мережі, в одновимірну форму і вибір функцій на одновимірному рівні. Цей шар також можна використовувати як вихідний шар ЗНМ, результатом якого є ймовірність належності вхідного зображення до певного класу. Схема цього шару представлена на рисунку 2.6.

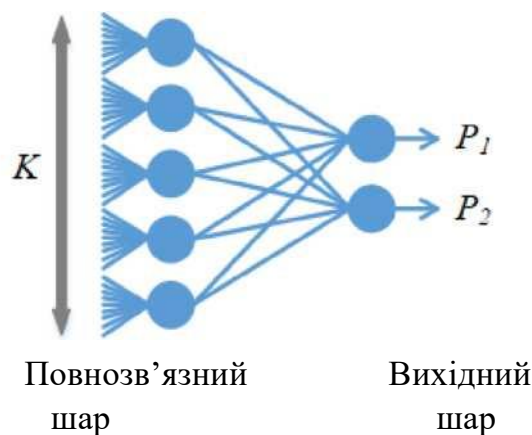


Рисунок 2.6 – Схема fully-connected шару

Кількість нейронів K виступає єдиним параметром даного шару. Зазвичай воно визначається виходячи з розміру попереднього шару та кількості класів (виходів) мережі. Навчанням ЗНМ. називається процес підлаштування значень ваг зв'язків між нейронами мережі.

Типи шарів, описані в цьому розділі, є основними у формуванні архітектури ЗНМ. Однак вони можуть розташовуватися будь-яким чином і мати широкий діапазон значень параметрів.

Аналіз найбільш ефективних і високоточних архітектур ЗНМ (LeNet, AlexNet, Googlenet, Resnet), доступних сьогодні, дозволив визначити архітектуру ЗНМ, які забезпечують найкращу якість виконання роботи [13, 40].

Нижче наведено основні рекомендації щодо вибору параметрів архітектури ЗНМ, які дозволяють забезпечити найвищу ефективність мережі та точність

результатів. Ці рекомендації засновані на аналізі найефективніших доступних архітектур ЗНМ.

Рекомендації щодо вхідного шару мережі:

- вхідні 2D-зображення мають бути квадратними;
- висота та ширина вхідного зображення мають багаторазово ділитися на 2;

Рекомендації щодо шару згортки:

- параметри P і S слід вибрати так, щоб розміри згорткового шару були цілими і, бажано, дорівнювали розмірам попереднього шару, за умови, що попередній шар також не є шаром згортки;

- краще використовувати невеликі фільтри: 3×3 або 5×5

- чим більший розмір вхідного шару мережі, тим більше фільтрів потрібно використовувати в згорткових шарах.

Рекомендації щодо архітектури шару pooling (субдискретизації):

- більш доцільно використовувати вибір максимального значення, а не обчислення середнього значення;

- краще використовувати розмір вікна 2×2 , оскільки більший розмір вікна значно впливає на руйнування ознак, отриманих згортковими шарами мережі.

Процес формування архітектури ЗНМ повинен враховувати параметри глобальної архітектури, показані на рисунку 2.7.

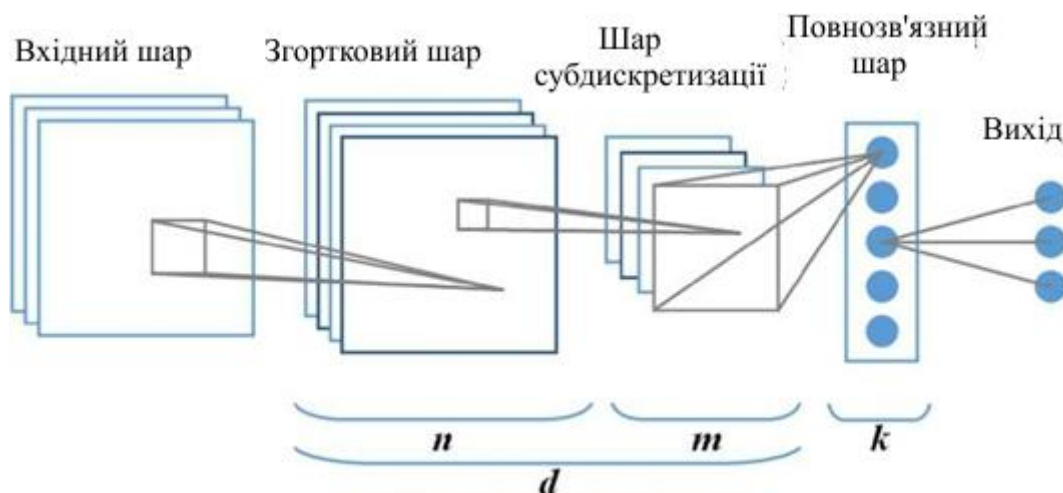


Рисунок 2.7 - Шаблон архітектури ЗНМ

Багато, щоб ЗНМ містила якнайбільшу кількість шарів (більше значення d). Чим складніше зображення та більшого розміру, тим більше значення n .

Рекомендовано використовувати як можна менше значення k .

2.3 Метод формування послідовності дій з використанням прецедентного підходу

Функціонування агента складається з послідовності елементарних дій $a_1, a_2, a_3, \dots, a_n$, що можуть бути використані агентом. Всі елементарні дії в загальному випадку мають певний скінчений набір вхідних параметрів для необхідного налаштування її функціонування x_1, x_2, \dots, x_m . Прикладами елементарних дій робота є рухи робота вперед, назад та повороти, а параметрами можуть виступати швидкість, тривалість руху та ін.

При проектуванні робота виділяються стандартні послідовності дій, що формують бібліотеку і відповідають певним рефлексорним реакціям людини на різні вхідні умови для досягнення мети. Стандартні послідовності дій спрацьовують при активізації певного набору ознак на вихідному шарі мережі.

Послідовність дій агента може бути описана орієнтованим графом на основі ланцюга Маркова, де стани графу відповідають елементарним діям, а ребра відповідають вазі зв'язку між ними. Знаки початкової умови q пов'язані з першою вершиною, з якої починається послідовність дій. Ознаки мети g пов'язані з останньою вершиною послідовності дій. Коли вага зв'язку між вершинами дорівнює одиниці, тоді перехід між вершинами строго визначений. Коли він відмінний від одиниці, перехід у загальному випадку не є строго обумовленим.

Успішне досягнення мети так чи інакше впливає на вагу ребер. Тобто збільшує вагу всіх ребер на шляху при успішному досягненні мети і навпаки.

На рисунку 2.8 наведено граф послідовності дій на основі ланцюга Маркова, яким можна представити безумовний рефлекс.

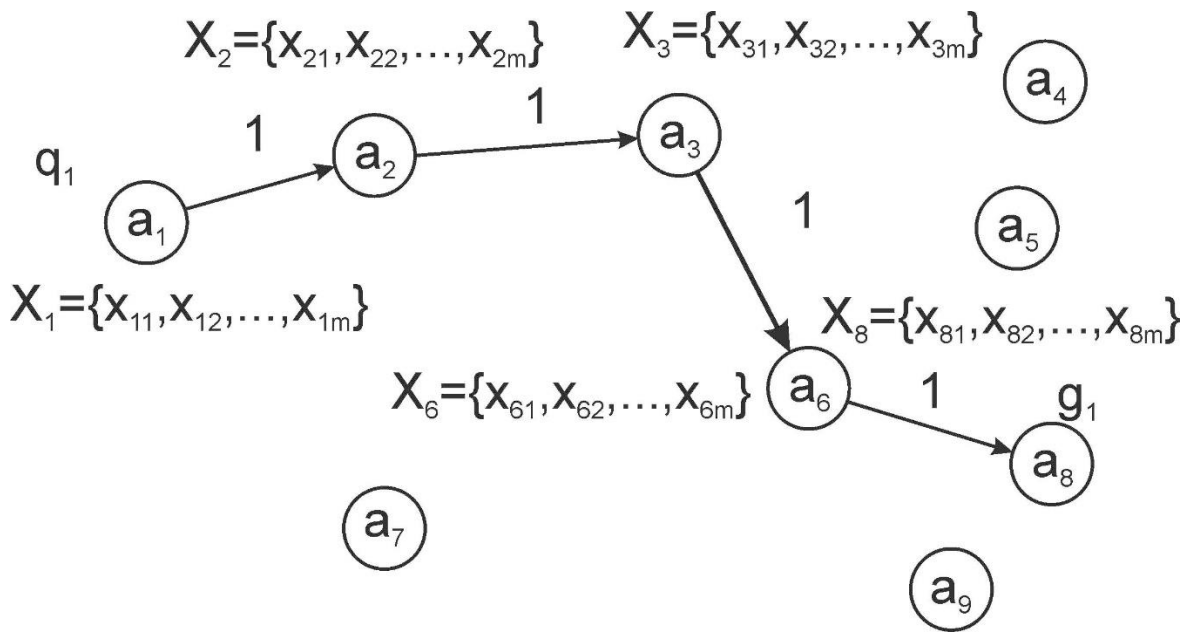


Рисунок 2.8 - Граф, який моделює безумовний рефлекс послідовності дій агенту

Тобто з рисунку чітко видно, що при деякій початковій умові виконується послідовність дій a_1, a_2, a_3, a_6, a_8 , показана на графі.

На рисунку 2.9 наведено граф послідовності дій, яка може відповідати умовному рефлексу агенту. Тобто з вершини може бути кілька сценаріїв, ваги кількох ребер з вершини в сумі повинні дорівнювати одиниці. За замовчуванням вибирається наступна елементарна дія, яка надходить з вершини з найбільшою вагою.

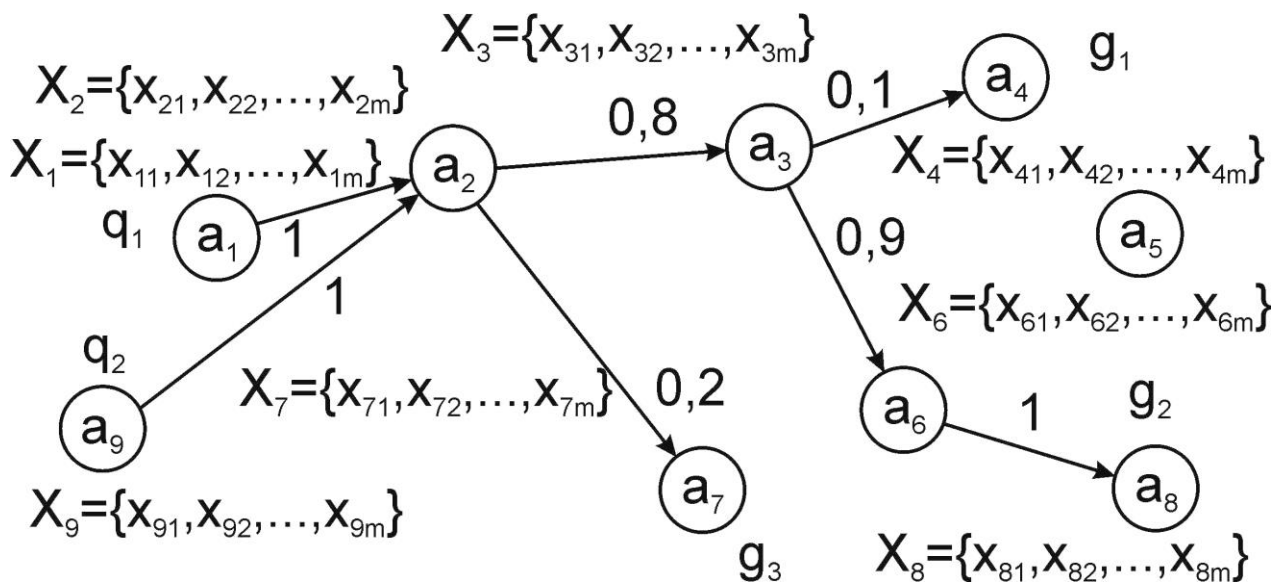


Рисунок 2.9 - Граф, який моделює умовний рефлекс послідовності дій агенту

В цьому випадку при розвитку подій, яка розпочинається з початкової умови q_1 , виконується послідовність дій a_1, a_2, a_3, a_6, a_8 , яка закінчується досягненням мети g_2 .

Але в цьому випадку агент може проаналізувати поточний стан сенсорів і вибрати перехід по ребрах з меншою вагою, якщо ребра з більшою вагою не приводять до досягнення цілі.

Плануючи послідовність дій, необхідно дозволити агенту використовувати як елементарні дії, так і деяку послідовність цих дій.

Плани дій, представлені попередньою моделлю, повинні бути адаптовані до нової ситуації.

Використання адаптивного методу полягає в підміні мети та початкових умов обраного прецеденту новою задачею і початковими умовами і забезпечені виконання плану після підміни. Деякі кроки початкового плану можуть виявитися непотрібними, так як змінилася мета, досягнення якої вони прагнули.

Досвід розвитку теорії ситуаційного менеджменту показав, що при виникненні нової проблемної ситуації доцільно використовувати метод міркувань на основі прецедентів - Case Based Reasoning (CBR) [34].

Метод CBR був розроблений в результаті досліджень когнітивної науки і, насамперед, дослідження динамічної пам'яті Р. Шенка [8] та його студентів Єльського університету. Було виявлено, що люди часто вирішують проблеми на основі свого попереднього досвіду, згадуючи, як вони вирішували подібні проблеми в минулому. CBR — це підхід, який дозволяє вирішити нову проблему, використовуючи або адаптуючись до вже відомої проблеми. Розуміння універсальності цього підходу призвело до створення та бурхливого розвитку теорії прецедентів. Так, у [22] розглядається розробка ситуаційної моделі управління інформаційною безпекою на оперативному рівні з використанням підходу, заснованого на теорії прецедентів.

Суть методології CBR полягає в наступному. Відповідно до методу CBR слід визначити, що прецедент — це пара «ситуація» і «метод рішення». Ситуації, що виникають з часом, та їх вирішення зберігаються в спеціальній базі прецедентів. Коли виникає нова ситуація на основі прецедентів, ми знаходимо відповідну ситуацію в базі, і спосіб її вирішення адаптується до ситуації, про яку йдеться.

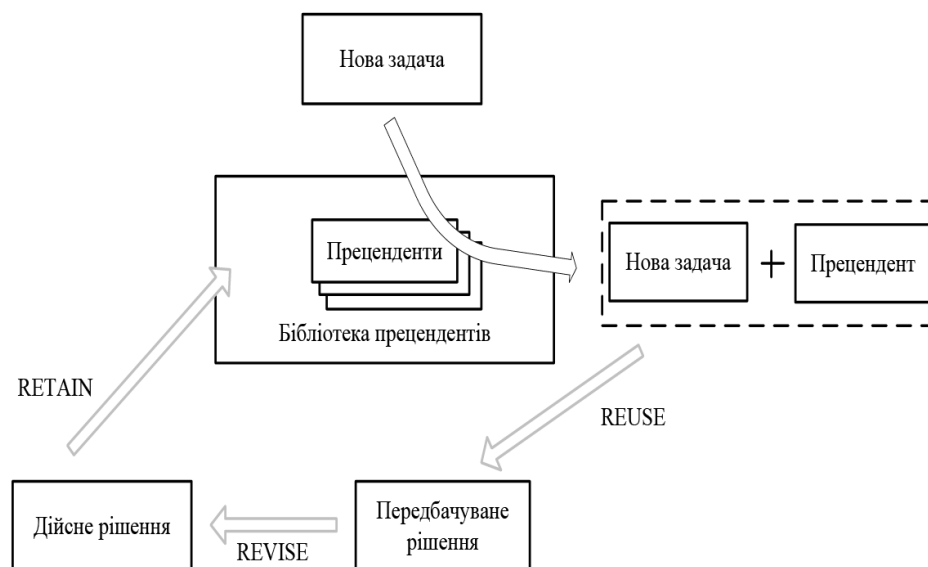


Рисунок 2.10 - Діаграма CBR-циклу

Подія адаптації (як прецедент) може включати інформацію про мету адаптації, що таке початковий план і які методи адаптації використовуються. Для даної нової адаптаційної проблеми проводиться пошук прецеденту адаптації з подібними цілями, і робиться спроба застосувати методи адаптації, знайдені в поточній позиції.

При плануванні послідовності дій на основі адаптивного підходу роботу слід дозволити використовувати як елементарні дії, так і певну послідовність таких дій (підпрограм).

Для створення нових підпрограм робочого процесу можна використовувати кілька варіантів адаптації.

Застосування організованої бібліотеки підпрограм надає можливість використовувати їх при потребі автоматичної генерації послідовності дій на певні загрози, які особливо актуальні в критичній інфраструктурі, системах військового призначення, комплексних системах захисту інформації та спрощує розробку таких протидій фахівцями.

Під час навчання ми створюємо серію вихідних нейронів на виході згорткової нейронної мережі, які відповідають кількості класів розпізнавання.

Вихідні нейрони зазвичай приймають значення від -1 до 1.

Тобто ми отримуємо вектор значень, який нам потрібен для використання готової підпрограми дії або коригування існуючих, які використовуються в подібних випадках.

Основною метою використання прецедентного апарату є надання готового рішення поточної ситуації на основі попередніх прецедентів при управлінні даним об'єктом або процесом.

Існує ряд способів отримання прецедентів та їх модифікацій. Метод визначення найближчого сусіда (NN – Nearest Neighbor) найчастіше використовується для вирішення завдань класифікації, групування, регресії та розпізнавання образів. Основними перевагами цього методу є простота впровадження та універсальність у змісті незалежність від особливостей

конкретної проблемної області. До суттєвих недоліків методу можна віднести складність вибору метрики для знаходження ступеня подібності та пряму залежність необхідних обчислювальних потужностей від розміру БП, а також неефективність при роботі з зашумленими та неповними вихідними даними.

Незалежно від того, чи даний прецедент (C) і поточний проблемний стан (T) у n -вимірному просторі властивостей (параметрів, властивостей), ступінь подібності ($S(C, T)$) можна визначити використовуючи одну з метрик: Евклідова метрика; Манхетенська метрика; Чебишевська відстань; Журавльова відстань; міра подібності Хемінга. Для визначення значення ступеня подібності $S(C, T)$ необхідно знайти максимальну відстань d_{max} у обраній для цього метриці. На жаль, цей метод не дає чітких рекомендацій при наявності кількох прецедентів, рівновіддалених від поточної ситуації, а також неефективних у разі неповноти вихідних даних [6].

З урахуванням особливостей складних об'єктів запропоновано модифікований алгоритм цього методу. Ця модифікація полягає в тому, що вводиться коефіцієнт K , який є пороговим значенням ступеня подібності прецедентів (S) і поточної ситуації. Таким чином, в результаті порівняння вибирається не тільки один найближчий сусід (прецедент), а й деякий набір близьких сусідів, ступінь схожості яких більше або дорівнює пороговому значенню ($S \geq K$). Це дозволяє вирішити проблему, якщо є кілька прецедентів, рівновіддалених від поточної ситуації.

Крім того, модифікований метод враховує можливість роботи з неповною або зашумленою інформацією у вихідних даних та можливість на основі експертних знань враховувати співвідношення значущості параметрів об'єкта.

Якщо в описі прецедентів немає значень параметрів, ми будемо вважати, що прецеденти і поточна ситуація абсолютно однакові для цих параметрів (тобто ці параметри не важливі для цих прецедентів). І навпаки, якщо в описі поточної ситуації немає значень параметрів, ми будемо вважати, що прецеденти і поточна ситуація не збігаються з цими параметрами.

Ваговий коефіцієнт i -го параметра об'єкта позначається w_i , та його значення буде змінюватися в діапазоні від 0 до 1 (від 0 до 100%). Спочатку вага параметра приймається рівною 1, але експерт може встановити необхідні значення для коефіцієнтів значущості параметрів. При отриманні прецедентів з БП і розрахунку ступеня подібності значення параметрів необхідно скоригувати та врахувати шляхом множення значень параметрів на відповідний коефіцієнт ($x_i * w_i$) і врахувати вагові коефіцієнти при розрахунку максимальної відстані $dmax(x_i^{nac} * w_i$ і $x_i^{kon} * w_i$).

2.4 Висновки

У цьому розділі описано формування параметрів архітектури згорткової нейромережі, розробленої на основі аналізу існуючих ЗНМ, таких як LeNet, AlexNet, Googlenet, Resnet та вироблених рекомендацій щодо вибору параметрів архітектури ЗНМ. Запропонований підхід розраховує параметри архітектури на основі характеристик вхідних даних і визначає необов'язкові значення деяких параметрів. Запропоновано також підхід до формування адаптивного класифікатора – набору ЗНМ. Принцип формування набору полягає в тому, що найважливішим для точності архітектурних параметрів є вибір мереж, які істотно відрізняються.

За результатами даної роботи можна зробити висновок, що незважаючи на велику кількість різноманітних параметрів архітектури ЗНМ, на основі вивчення передових робіт у цій галузі ми можемо виділити архітектурні шаблони та найбільш значущі параметри. Важливо, визначити точність модельної класифікації та допоміжної, розрахованої за характеристиками оброблених даних. Тому на основі обраних особливостей архітектури ЗНМ отримано модель, яка забезпечує найвищу точність класифікації та стійкість до змін обробленої інформації.

З урахуванням особливостей складних об'єктів запропоновано модифікований метод найближчого сусіда. Ця модифікація полягає в тому, що застосовується спеціальний коефіцієнт K , що відповідає пороговому значенню ступеня схожості прецедентів і поточної ситуації. В результаті порівняння вибирається не тільки один найближчий сусід (прецедент), а й деякий набір близьких сусідів, ступінь схожості яких більше або дорівнює пороговому значенню ($S \geq K$).

3 АЛГОРИТМИ ІНТЕЛЕКТУАЛЬНОГО КЕРУВАННЯ

3.1 Формування архітектури згорткової нейронної мережі

Архітектура згорткових нейронних мереж розроблена та використовується для ефективного розпізнавання зображень, де згорткові шари чергуються з нелінійними функціями активації (ReLU або гіперболічний тангенс \tanh) і кластерних шарів (pooling layers).

Як відомо, багатошарові нейронні мережі отримують вхідні дані (наприклад, вектор), а потім перетворюють інформацію, пропускаючи її через кілька прихованих шарів. Кожен прихований шар складається з багатьох нейронів, при цьому кожен нейрон має міцний зв'язок з усіма нейронами попереднього шару і де нейрони як шар повністю незалежні один від одного і не мають загальних зв'язків. Останній повністю підключений шар називається вихідним, а в налаштуваннях класифікації він показує кількість класів [24].

Згорткові нейронні мережі (рисунок 3.1) використовують той факт, що вхідні дані складаються із зображень, і обмежують побудову мережі більш розумним чином. На відміну від звичайної нейронної мережі, шари ЗНМ складаються з нейронів, що розташовуються у 3 вимірах: ширина, висота та глибина, тобто у вимірах, які складають об'єм. Як буде описано нижче, нейрони будуть підключені лише до невеликої ділянки шару.[4].

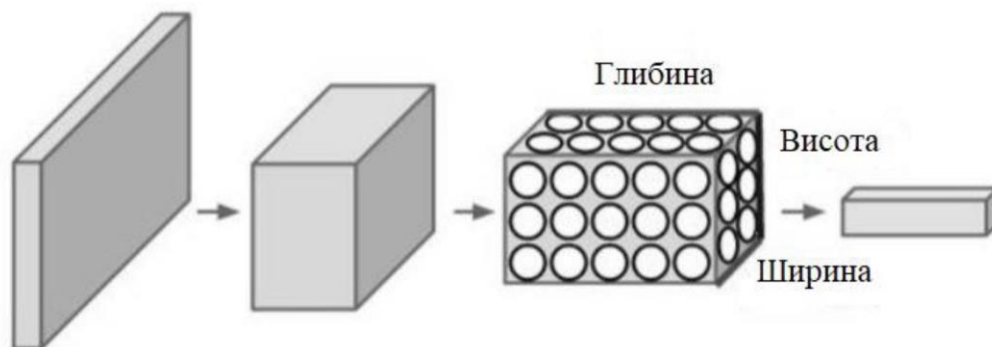


Рисунок 3.1 – Архітектура ЗНМ

В нашому випадку завдання побудови архітектури ЗНМ для проведення класифікації вхідного кольорового зображення потребує «згортання» вхідного шару мережі до шарів які мають найменші розміри, наприклад, $1 \times 1 \times D$ або $2 \times 2 \times D$ де D - глибина шару, і далі перетворенню до C одномірних сигналів вірогідності приналежності об'єкту на вході мережі до одного з наявних C класів.

На основі запропонованого підходу до уточнення ЗНМ розроблено алгоритм, який регулює вибір параметрів архітектури мережі на основі основних характеристик вхідних даних на кожному етапі послідовного формування мережевих шарів. Алгоритм передбачає необов'язкове введення основних параметрів мережі та випадкове значення деяких параметрів, що означає, що результатами роботи алгоритму в результаті його роботи для різних вхідних даних може бути набір архітектур ЗНМ.

В описі алгоритму прийнято наступне позначення:

N - розмір квадрата попереднього шару (для першого згорткового шару — це розмір вхідного шару);

C - кількість класів, до яких належить класифікатор;

D - глибина мережевого вхідного шару;

P - кількість стовпців і рядків, що містять нулі, додані до меж шару, що передуює шару згортки (параметр архітектури шару згортки);

F - може приймати значення 3, 5 або 7, які характеризують розмір квадратних фільтрів згорткового шару;

S - зміщення між фільтрами у формуванні сигналу нейронів згорткового шару / зміщення між вікнами у формуванні сигналу шару піддискретизації;

$Filters$ - кількість фільтрів (глибина шару згортки);

$Filters_p$ - глобальний параметр циклу формування мережевих шарів, (глибина попереднього шару згортки);

U - розмір квадратного вікна субдискретизації;

AF_c - активаційна функція нейронів згорткового шару.

Subf - приймає значення *max* - максимум, або *avg* – середнє, це тип функції субдискретизації (рис. 3.2);

K - кількість нейронів повнозв'язаного шару;

AFfc - це функція для активації повністю зв'язаного шару;

R - повторювана послідовність із *n* згорткових шарів та *m* субдискретизаційних, що складають рівень мережі, приймають цілочисельні значення: $n \in [1; 2]$, $m \in [0; 1]$;

d - глибина мережі, тобто кількість її рівнів мережі.

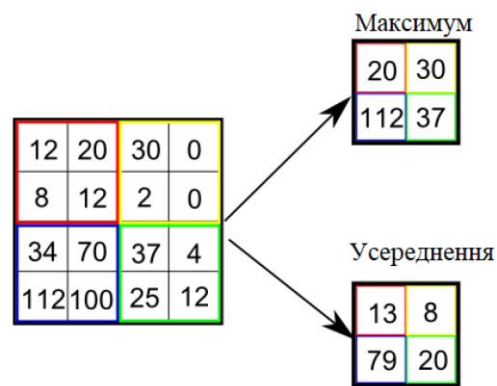


Рисунок 3.2 - Операція підвибірки

Алгоритм формування архітектури ЗНМ наведено на рис. 3.3

Розглянемо основні етапи алгоритму.

При необхідності відбувається введення основних параметрів алгоритму: чим більше функцій потрібно виявити у вибірках даних, тим більше рекомендується встановлювати значення *F*, параметри рівня мережі *n*, *m* - рекомендується встановити в залежності від складності зображення (за замовчуванням $n = 1$, $m = 1$). На наступному етапі відбувається формування параметрів вхідного шару мережі: введення розміру вхідного квадратного зображення *N*, де *N* має ділитися багаторазово на 2 до однозначних чисел. Далі відбувається введення глибини вхідного шару *D* (кількість колірних каналів), та значення кількості класів *C* і ініціалізуються глобальні змінні алгоритму: $d = 0$, $Filters_p = 0$. Та циклу формування згорткових шарів $in = 0$.

F вибирається в залежності від розміру квадрата попереднього шару N за формулою (3.1)

$$F = \begin{cases} 7, \text{if}(N \geq 64) \\ 5, \text{if}(32 \leq N < 64) \\ 3, \text{if}(N < 32) \end{cases} \quad (3.1)$$

$Filters$ вибирається по формулі (3.2).

$$Filters = \begin{cases} 8, \text{if}(F = 7 \ \&\& d = 0) \\ 16, \text{if}(F = 5 \ \&\& d = 0) \\ 24, \text{if}(F = 3 \ \&\& d = 0) \\ Filters_p \ \parallel \ 1,25 * Filters_p, \text{if}(d > 0) \end{cases} \quad (3.2)$$

P і S розраховуються по формулі (3.3):

$$\begin{aligned} S &= (N - F + P * 2) / (N - 1) \\ P &= ((N - 1)S - N + F) / 2 \end{aligned} \quad (3.3)$$

Розрахунок N для вже сформованого згорткового шару проводиться по формулі (3.4):

$$N = (N_p - F + P * 2) / S + 1 \quad (3.4)$$

де N_p - розмір попереднього шару мережі.

$$Af_c = \max(0, x), \quad (3.5)$$

Розрахунок N для вже сформованого шару субдискретизації мережі проводиться по формулі (3.6),

$$N = (N_p - U) / S + 1, \quad (3.6)$$

Формування параметрів повнозв'язного шару мережі відбувається так:

- K вибирається рівним S ;
- сигмоїдна функція використовується в якості функції активації нейронів повнозв'язаного шару.

Далі відбувається формування вихідного шару мережі, який призначений для розрахунку вірогідностей приналежності об'єкту вхідного зображення

існуючим класам С.

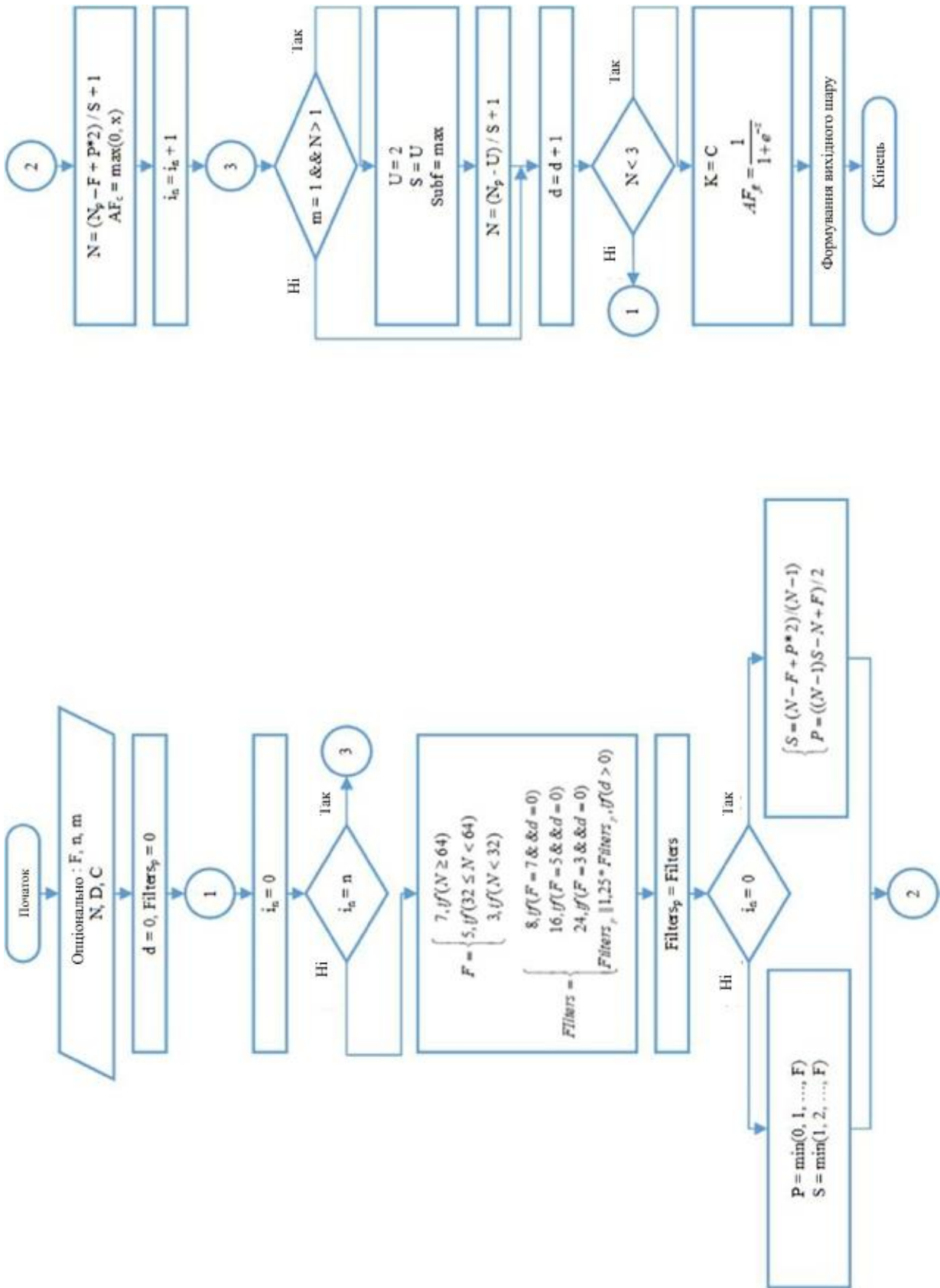


Рисунок 3.3 - Блок-схема алгоритму формування архітектури ЗНМ

Модифікація архітектури мережі потрібно здійснити шляхом її ускладнення та застосування певної послідовності правил:

1. Додайте згортковий шар до одного з рівнів мережі згідно з алгоритмом, починаючи з останнього найглибшого шару.
2. Додайте більше фільтрів (збільшення становить 25%) із згорткового шару мережі, починаючи з останнього найглибшого рівня.
3. Додайте новий шар мережі, який приєднується до останнього, найглибшого рівня утвореної мережі.

В запропонованому підході в якості алгоритму навчання використовується одну з найбільш ефективних і найшвидших модифікацій алгоритму зворотного поширення: Adadelta [17].

Запропонований підхід для формування найбільш ефективної архітектури ЗНМ дозволяє отримати невеликий кінцевий набір архітектур, кожна з яких забезпечує більш високу точність класифікації в задачах розпізнавання зображень.

Одним із недоліків є те, що алгоритм передбачає опціональне введення деяких параметрів мережі і максимальну ефективність цього підходу неможливо забезпечити повністю автоматично. Користувач повинен оцінити складність розпізнавання вибірок даних і введення відповідних значень вхідних змінних алгоритму.

Представлений апарат для формування ефективної архітектури ЗНМ взято за основу запропонованого в цій роботі підходу до створення моделі адаптивного класифікатора на основі набору згорткових нейронних мереж. При відсутності дій користувача використовуються значення за замовчуванням.

3.2 Створення адаптивного класифікатора об'єктів

На основі вивчених досліджень, що описують експериментальні дослідження в області згорткових мережних архітектур та експериментів з різними конфігураціями архітектур ЗНМ, створених із застосуванням описаного в розділі 3.1 підходу, визначено параметри архітектури, які суттєво впливають на результати роботи класифікатора. Нижче наведені списки ідентифікованих характеристик, розбиті за ступенем впливу на результуючу точність класифікації.

Найважливіші параметри:

1. Розмір фільтра F згорткового шару нейронної мережі. Він впливає ефективність виділення ознак різних об'єктів зображення, а тому є однією з найважливіших особливостей архітектури.

2. Тип мережевого рівня, обумовлений типами шарів, що входять до нього і їх кількість - n і m параметри алгоритму. У цій роботі розглядалися рівні двох типів: перший складається із згорткового шару та субдискретного шару, розташованих послідовно ($n = 1, m = 1$); другий, що складається з двох послідовно розташованих згорткових шарів і шару субдискретизації ($n = 2, m = 1$).

Менш значущі параметри:

1. Параметр *Filters* алгоритму, який характеризує кількість фільтрів, що використовуються в згортковому шарі мережі

2. Параметр d алгоритму - кількість рівнів (глибина) мережі є глобальним. Цей параметр повністю визначається іншими параметрами.

Інші параметри, не зазначені в наведених списках, або встановлюються однозначно відповідно до запропонованого, або вони є допоміжними і визначаються зі значень інших параметрів і не мають значного впливу на отримані результати.

З отриманого набору ЗНМ, які створені на основі запропонованого підходу, формується набір моделей (рекомендується обрати не більш трьох архітектур), в яких шляхом усереднення результатів, формується вихідний сигнал.

$$Y_{ensemble} = \frac{1}{T} \sum_{i=1}^T y_i(x), \quad (3.7)$$

де $Y_{ensemble}$ – усереднений вихідний сигнал,

T - кількість архітектур в ансамблі;

$y(x)$ - вихідний сигнал однієї з моделей. Загальна схема набору ЗНМ наведено на рисунку 3.4.

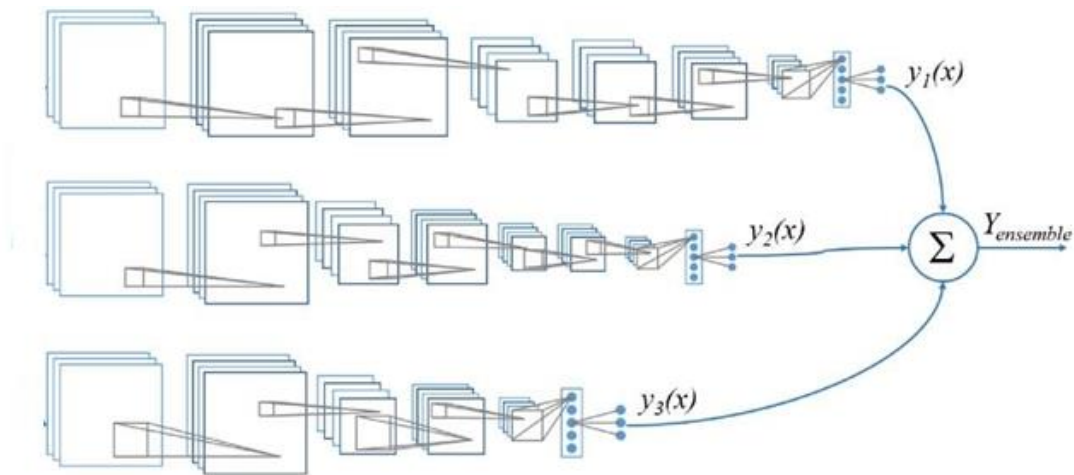


Рисунок 3.4 - Загальна схема набору згорткових нейронних мереж

Одним із підходів до вибору моделі, що оптимально підходить для конкретного завдання детектування зображень, є дослідження точності класифікації на конкретному наборі і виборі для цього набору відповідної моделі.

3.3 Алгоритм формування дій робота на основі прецедентів

На основі описаного в розділі 2.4 модифікованого методу визначення найближчого сусіда (найближчих сусідів) запропоновано алгоритм добування прецедентів, який враховує коефіцієнти важливості параметрів об'єкта та використовують різні метрики [25] для виявлення ступеня близькості прецедентів. Вхідними параметрами є поточна ситуація T , CL – база прецедентів (БП), w_1, \dots, w_n

– вагові коефіцієнти параметрів, m – кількість розглянутих прецедентів і порогове значення подібності K . В результаті ми отримуємо SC (Set of Cases) множину прецедентів, які мають ступінь схожості більшу порогового значення K .

В кваліфікаційній роботі застосовано евклідову метрику, але у подальшому можлива реалізація узагальненого алгоритму отримання прецедентів з використанням різних метрик.

Поведінка агента опишемо кінцевим автоматом, де стан машини описує дії, які виконує агент.

Розробляючи алгоритм роботи нашого агента, ми беремо за основу цілеспрямовану парадигму. Однак цілі агента будуть формулюватися не зовнішнім користувачем, а переважно на основі знання агента про свій стан на основі показників внутрішніх рецепторів.

Функціонування агента складається з послідовності елементарних дій $a_1, a_2, a_3, \dots, a_n$, що можуть бути використані агентом. Всі елементарні дії в загальному випадку мають певний скінчений набір вхідних параметрів для необхідного налаштування її функціонування x_1, x_2, \dots, x_m . Прикладами елементарних дій робота є рухи робота вперед, назад та повороти, а параметрами можуть виступати швидкість, тривалість руху та ін.

При проектуванні робота виділяються стандартні послідовності дій, що формують бібліотеку і відповідають певним рефлекторним реакціям людини на різні вхідні умови для досягнення мети.

Послідовність дій агента може бути описана орієнтованим графом на основі ланцюга Маркова, де стани графу відповідають елементарним діям, а ребра відповідають вазі зв'язку між ними.

Моделювання даних забезпечує найбільший ефект при застосуванні на ранніх стадіях життєвого циклу розробки. Модель надає важливу інформацію для розуміння функціональних меж проекту на ітераційних фазах розробки. Початок етапу впровадження без чіткого розуміння вимог до даних може призвести до того, що проект швидко виходить за межі виділеного бюджету або закінчить своє

існування на купі незавершених бета-версій. Моделювання даних є складним і відповідальним завданням, оскільки від якості моделі даних залежить, в кінцевому рахунку, ефективність і продуктивність інформаційної системи.

Представимо графові моделі з підрозділу 2.3 на основі матриці суміжностей [31].

У процесі виконання елементарних дій відбувається їх групування в підпрограми. В одну підпрограму краще об'єднувати елементарні дії, які виконуються, наприклад, одна за одною. На основі графа (рис. 2.8) будується матриця суміжності T . Розглянемо матрицю суміжності, побудовану за графом (рис. 3.5), де $a1-a9$ – це набір елементарних дій робота.

	a1	a2	a3	a4	a5	a6	a7	a8	a9
a1	0	1	0	0	0	10	0	0	0
a2	10	0	1	0	0	0	0	0	0
a3	0	0	0	0	0	1	0	0	0
a4	0	0	0	0	0	0	0	0	0
a5	0	0	0	0	0	0	0	0	0
a6	0	0	0	0	0	0	0	1	0
a7	0	0	0	0	0	0	0	0	0
a8	0	0	0	0	0	0	0	0	0
a9	0	0	0	0	0	0	0	0	0

Рисунок 3.5 - Матриця суміжності T , який моделює безумовний рефлекс послідовності дій агента

	a1	a2	a3	a4	a5	a6	a7	a8	a9
a1	0	1	0	0	0	10	0	0	0
a2	0	0	0,8	0	0	0	0,2	0	0
a3	0	0	0,1	0	0,9	0	0	0	0
a4	0	0	0	0	0	0	0	0	0
a5	0	0	0	0	0	0	9	0	0
a6	0	0	0	0	0	0	0	1	0
a7	0	0	0	0	0	0	0	0	0
a8	0	0	0	0	0	0	0	0	0
a9	0	1	0	0	0	0	0	0	0

Рисунок 3.6 - Матриця суміжності T, який моделює умовний рефлекс послідовності дій агенту

Ми використовуємо наступні підходи для отримання дій:

- 1) модифікація готових до використання підпрограм з схожими ознаками (планування за прецедентом);
 - 2) проєктування меташаблонів експертами;
- Алгоритм функціонування агента показаний на рис. 3.7.

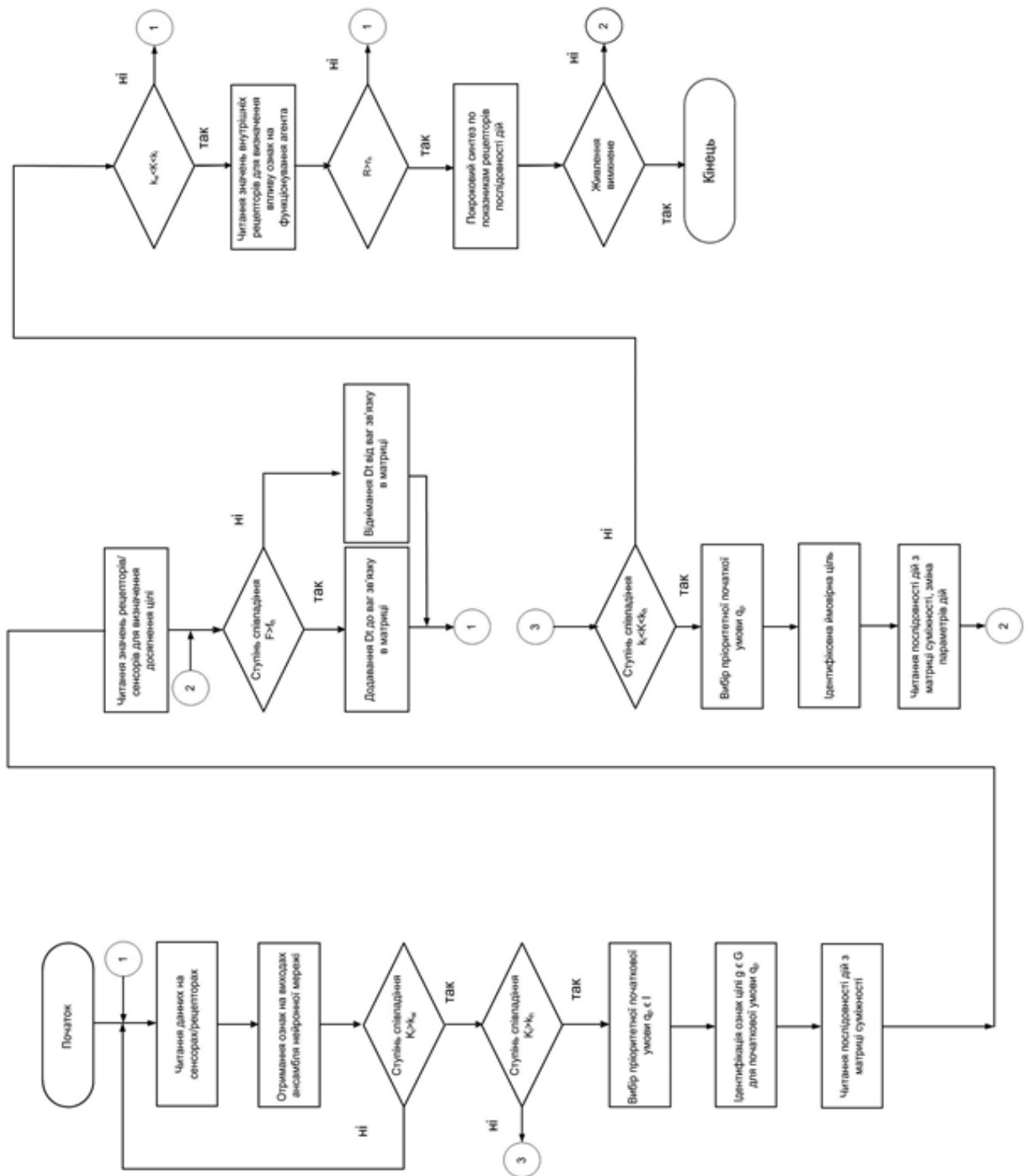


Рисунок 3.7 - Алгоритм функціонування агента

Нейронні мережі отримують інформацію на рецепторах і видають на виході вектор ознак для початкових умов. На підставі цих даних формується вектор ознак цілі, враховуючи існуючі прецеденти. Набір ознак, який надходить з сенсорів має поріг k_w , щоб агент не реагував на інформаційний шум. Якщо інформація з

сенсорів має більше ступеня подібності ознак k_h то агент виконує послідовність дій, які відповідають цим ознакам.

Щоб агент не реагував на шум інформації та непотрібний набір характеристик, інформація має встановлений нижній поріг характеристик k_w , які отримуються з сенсорних і приймальних входів.

Також є верхній поріг знаків k_h . Якщо інформація отримана і має певний ступінь подібності до знаків k_h , агент посилається на певну послідовність дій, що відповідає ознакам. Також є коефіцієнт k_l - границя межі подібності. При $k_l < K < k_h$ робот намагається по існуючим прецедентам синтезувати необхідну послідовність дій. При $k_w < K < k_l$ аналізується показники тільки внутрішніх рецепторів і вважається, що інформація на зовнішніх сенсорах мало важлива.

Якщо робот має негайні внутрішні проблеми (розрядний акумулятор, пошкодження елементів, розрядка акумулятора тощо), вони негайно відображаються в давачах, і нейронна мережа визначає їх як ціль агенту. Після ідентифікації дії їй може бути присвоєний вищий пріоритет. Ваги зв'язків в кожній послідовності дій коректуються після завершення, в залежності від успішності досягнення цілі.

3.4 Висновки

Запропонований підхід для формування найбільш ефективної архітектури ЗНМ дозволяє отримати невеликий кінцевий набір архітектур, кожна з яких забезпечує більш високу точність класифікації в задачах розпізнавання зображень.

Представлений апарат для формування ефективної архітектури ЗНМ взято за основу запропонованого в цій роботі підходу до створення моделі адаптивного класифікатора на основі набору згорткових нейронних мереж. При відсутності дій користувача використовуються значення за замовчуванням.

На основі запропонованого модифікованого методу визначення найближчого сусіда було розроблено алгоритм отримання прецедентів із використанням різних метрик для визначення ступеня схожості прецедентів із поточною ситуацією. і з урахуванням коефіцієнтів, що характеризують важливість параметрів об'єкта.

Агент-робот чекає дані, які дають встановлений нижній поріг сигналів, отриманих від входів давачів. Якщо вони надходять, робот виконує дії, які відповідають характеристикам, або намагається синтезувати послідовність дій відповідно до прецедентів.

Якщо робот має негайні внутрішні проблеми (розряджений акумулятор, пошкодження елементів тощо), вони негайно відображаються в давачах, і нейронна мережа визначає їх як ціль агенту. Після ідентифікації дії їй може бути присвоєний вищий пріоритет. Ваги зв'язків в кожній послідовності дій коректуються після завершення, в залежності від успішності досягнення цілі.

4 РЕАЛІЗАЦІЯ АДАПТИВНОГО ПІДХОДУ

4.1 Оцінка функціонування уточненої моделі нейронної мережі

На сьогоднішній день при реалізації нейронних мереж важливу роль відіграє реалізація архітектур, встановлених у конфігурації, як правило, спеціалізованих пакетів, що надають вже повністю реалізовані базові структури та алгоритми для легкої розробки. Найпоширенішими популярними бібліотеками є Theano, Tensorflow і Torch, серед інших [29]. Ці пакети реалізують програмний інтерфейс для реалізації низькорівневих обчислень, тому прийняття рішень повинно базуватися переважно на суперечностях продуктивності та простоти розробки. Варто відзначити, що Torch реалізує інтерфейс до мови Lua, яка не дуже популярна серед програмістів, що значно знижує зручність і швидкість розробки. Згідно з дослідженнями, Theano і Torch демонструють приблизно однакову продуктивність на одному наборі даних [18].

Крім описаного вище недоліку використання бібліотеки Torch, є ще один, також пов'язаний із простотою розробки. Два інших інструменти, згадані вище, можуть працювати з бібліотекою Keras верхнього рівня [24], яка надає загальні попередньо визначені алгоритми та структури для Tensorflow [25] і Theano. За результатами аналізу документації бібліотек було прийнято рішення використовувати для програмної реалізації згорткової нейронної мережі бібліотека ConvNetJS — це бібліотека JavaScript, спеціально розроблена для вивчення моделей глибокого навчання, які включають нейронні мережі. Великою перевагою цієї бібліотеки є те, що її можна повністю використовувати у браузері без спеціальних програмних вимог, таких як графічні процесори, компілятори тощо. ConvNetJS має параметри для нейронних мереж, задач класифікації та регресії, орієнтованих на зображення згорткових мереж та навчання. модуль покращення, який знаходиться на стадії експерименту. [31], реалізована мовою програмування Javascript і розповсюджувана за open source ліцензією.

ConvNetJS — це бібліотека JavaScript, спеціально розроблена для вивчення моделей глибокого навчання, які включають нейронні мережі. Великою перевагою цієї бібліотеки є те, що її можна повністю використовувати у браузері без спеціальних програмних вимог, таких як графічні процесори, компілятори тощо. ConvNetJS має параметри для нейронних мереж, задач класифікації та регресії, орієнтованих на зображення згорткових мереж та навчання. Модуль покращення, який знаходиться на стадії експерименту, передбачає можливість імпорту та експорту конфігурації у форматі Javascript Object Notation (JSON) [22].

Програмна система для перевірки роботи згорткових нейронних мереж, що мають задані архітектурні параметри, являє собою сторінку html, що містить скрипти на мові програмування Javascript. Конфігурація параметрів архітектури мережі та конфігурація навчання здійснюється в текстовому полі на веб-сторінці за текстовим описом (рис. 4.2).

```

layer_defs = [];
layer_defs.push({type:'input', out_sx:32, out_sy:32, out_depth:3});
layer_defs.push({type:'conv', sx:5, filters:16, stride:1, pad:2, activation:'relu'});
layer_defs.push({type:'pool', sx:2, stride:2});
layer_defs.push({type:'conv', sx:5, filters:20, stride:1, pad:2, activation:'relu'});
layer_defs.push({type:'pool', sx:2, stride:2});
layer_defs.push({type:'conv', sx:5, filters:20, stride:1, pad:2, activation:'relu'});
layer_defs.push({type:'pool', sx:2, stride:2});
layer_defs.push({type:'fc', num_neurons:2, activation:'sigmoid'});
layer_defs.push({type:'softmax', num_classes:2});
net = new convnetjs.Net();
net.makelayers(layer_defs);
trainer = new convnetjs.SGDTrainer(net, {method:'adadelta', batch_size:4, l2_decay:0.0001});

```

Рисунок 4.1 - Опис архітектури мережі й параметрів навчання в програмі для проведення експериментів

У програмі архітектура мережі являє собою масив з об'єктів Javascript, які задають опис типу та параметрів ЗНМ. Параметри навчання мережі, як видно зі скріншота на рисунку 4.2, задаються в змінній trainer. В роботі використовується алгоритм навчання Adadelta [23]. AdaDelta (root mean square propagation) - це алгоритм оптимізації, що є спадкоємцем ідей, закладених у AdaGrad .

Нагадаємо, головний недолік чистого адаптивного градієнтного спуску - це безконтрольне накопичення квадратів градієнтів, яке призводить до постійного

зменшення коефіцієнта навчання η , в результаті, до паралічу самого процесу навчання.

Перший принцип AdaDelta – замість повної суми оновлень $G_t G_t$ використовуватиметься усереднений квадрат градієнта. Метод нагадує принцип, що використовується в MomentumSGD - метод експоненційно загасаючого середнього, що біжить.

Для оцінки роботи мережі за певними критеріями використовується набір даних, у нашому випадку буде використовуватися набір даних у вигляді сховищ класифікованих зображень CIFAR-10 (англ. Canadian Institute for Advanced Research 10). Набір даних містить загалом 60 000 кольорових зображень, ці зображення розміром $32 * 32$, розділені на 10 категорій, кожна категорія містить 6 000 зображень. Існує 50 000 навчальних зображень, що становлять 5 навчальних пакетів, кожен з яких складається з 10 000 зображень. Дані тестової партії беруться з кожної з 10 категорій, і з кожної категорії випадковим чином вибирається 1000 аркушів. Решта організовані випадковим чином, щоб сформувати тренувальну групу. Зауважте, що кількість зображень у навчальному пакеті не обов'язково однакова. Що стосується навчального пакету в цілому, то в кожній категорії є 5000 зображень.

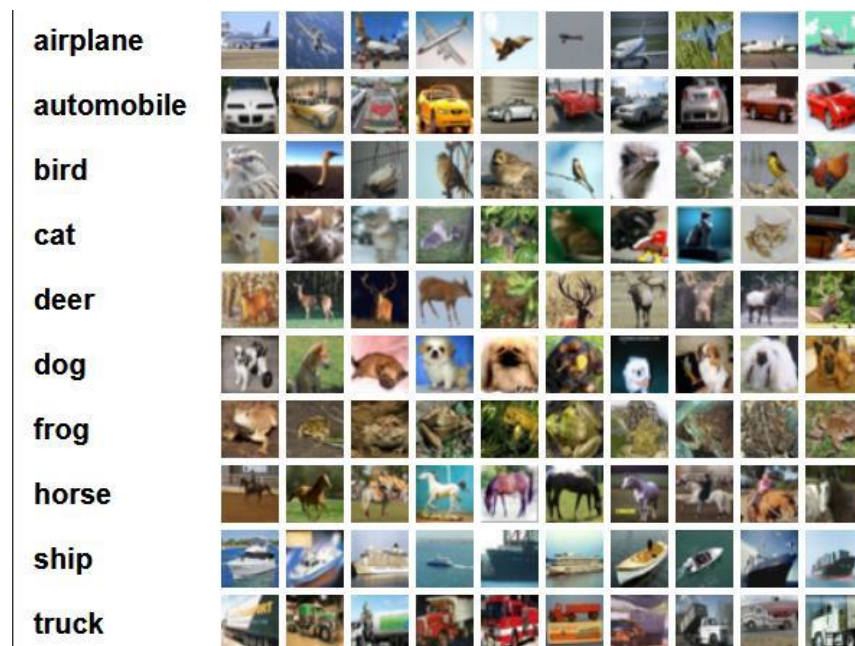


Рисунок 4.2 - CIFAR-10

Недоліком алгоритму зворотного поширення є те, що він не дозволяє досягти глобального мінімуму в загальному випадку. Тому виникають великі труднощі в навчанні нейронних мереж, які є методами подолання локальних мінімумів.

Епоха: пряме та зворотне проходження у всіх навчальних прикладах.

Розмір серії (паketу) – кількість навчальних прикладів для ітерації прямих і зворотних проходів. Кількість ітерацій - кількість проходів: кожен прохід використовує приклади (паket). Один пас еквівалентний прямому пасу та зворотному пасу. Тобто, маючи 1000 прикладів, паket = 500, нам потрібно буде виконати дві ітерації, щоб завершити одну еру.

З математичної точки зору навчання нейронної мережі є багатопараметричною задачею нелінійної оптимізації.

Процес навчання мережі описується за допомогою графіка кривої, де горизонтальна вісь відображає кількість навчальних вибірок, отриманих мережею, а вертикальна вісь відображає значення помилки, отримане при класифікації вхідних даних.

На рисунку 4.3 показано дві криві: крива значень помилок у навчальному наборі (чис значення встановлюється горизонтально) і крива значень помилок у тестовому або тестовому наборі (яка має фіксований розмір). Чим більше даних використовується для навчання, тим більше помилок буде створюватися архітектурою мережі, що відповідає навчальним даним. У той же час навчальні дані все більше нагадують фактичний розподіл даних, які будуть записані в навчальні дані. У якийсь момент похибка в навчальному та тестовому наборах повинна бути приблизно однаковою [12].

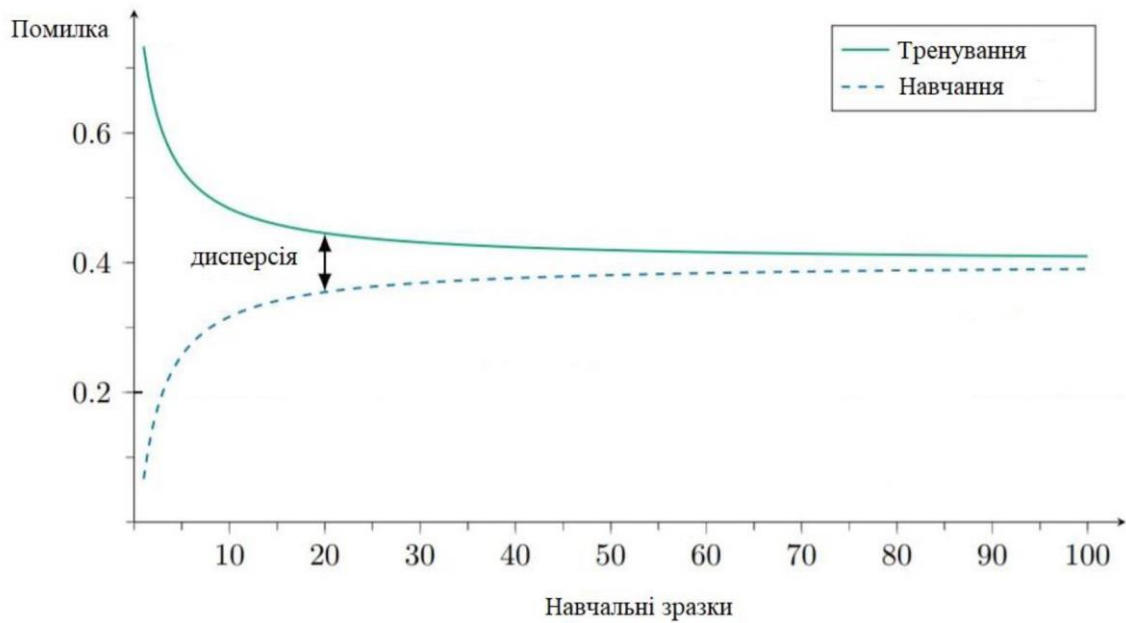
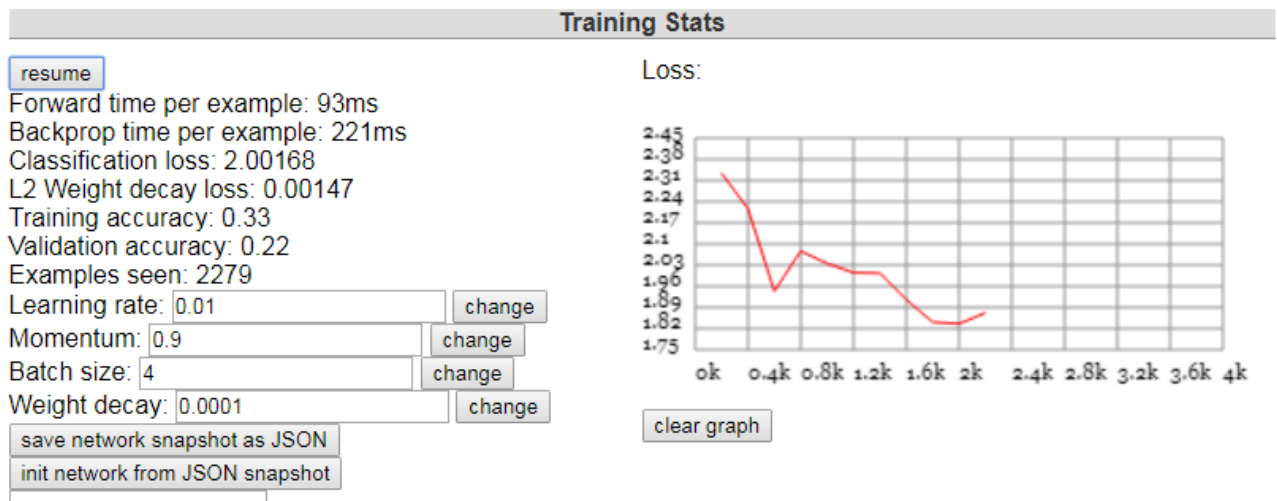


Рисунок 4.3 – Процес навчання ЗНМ

У процесі навчання вибірка, яка не повторюється протягом епохи навчання, випадковим чином вибирається з навчальної вибірки і надсилається до входу в мережу. Для управління перенавчанням мережі програма враховує динамічну побудову кривої помилок (рис. 4.4).



Кількість переглянутих прикладів

Рисунок 4.4 – Крива помилок Receiver Operating Characteristic

Крива процесу навчання є показником коректності мережі, якщо є велика кількість незмінних навчальних даних, що в свою чергу підвищує ефективність роботи мережі. Побудувавши криву процесу навчання, можна оцінити, чи здатна запропонована модель мережі відповідати даним для бажаної класифікації помилок. Значення помилки в тестовому наборі ніколи не повинно бути набагато менше значення помилки в навчальному наборі. Якщо помилка в навчальному наборі занадто велика, збільшення даних не допоможе вирішити проблему. Натомість стане зрозуміло, що необхідно коригувати архітектуру самої моделі або алгоритм її навчання. Якщо крива навчального набору на графіку значно перевищує криву набору тестових (пробних) даних, для вирішення цієї проблеми необхідно зменшити роздільну здатність зображень, ввести додаткові навчальні вибірки або збільшити регуляризацію.

Індикатори мережі використовуються для обчислення та відображення гіперпараметрів (наприклад, освітньої доби). Показники можуть бути представлені у вигляді діаграм гіперпараметрів по горизонтальній осі та значень якості на вертикальній осі. Точність розпізнавання, представлена втрата, є типовим показником якості. У випадку, коли основним гіперпараметром розглядається кількість епох навчання, показники виступають індикатором тривалого часу навчання та продуктивності мережі. Побудовуючи графіки значень помилок у навчальному наборі, а також значень помилок у тестовому наборі, також можна оцінити, чи є перенавчання загрозливим [16] (Рис 4.5).

Кількість епох навчання виступає як гіперпараметр мережі, а оцінка якості розпізнавання виражається через значення помилки. Чим більше ви вивчаєте мережу, тим краще ви звикнете до навчального набору даних. У якийсь момент мережа добре розпізнає лише навчальні дані, і ви втратите здатність узагальнювати. На цьому етапі криві якості навчання та тестові зразки на графіку розходяться. Поки класифікатор продовжує покращувати показник якості в навчальному наборі, він погіршує якість розпізнавання під час використання тестового набору. У ситуації, коли показник якості не покращується при достатній

кількості епох, необхідно змінити значення

Нейронні ваги ініціалізуються на початку мережі або застосовують нормалізацію моделі.

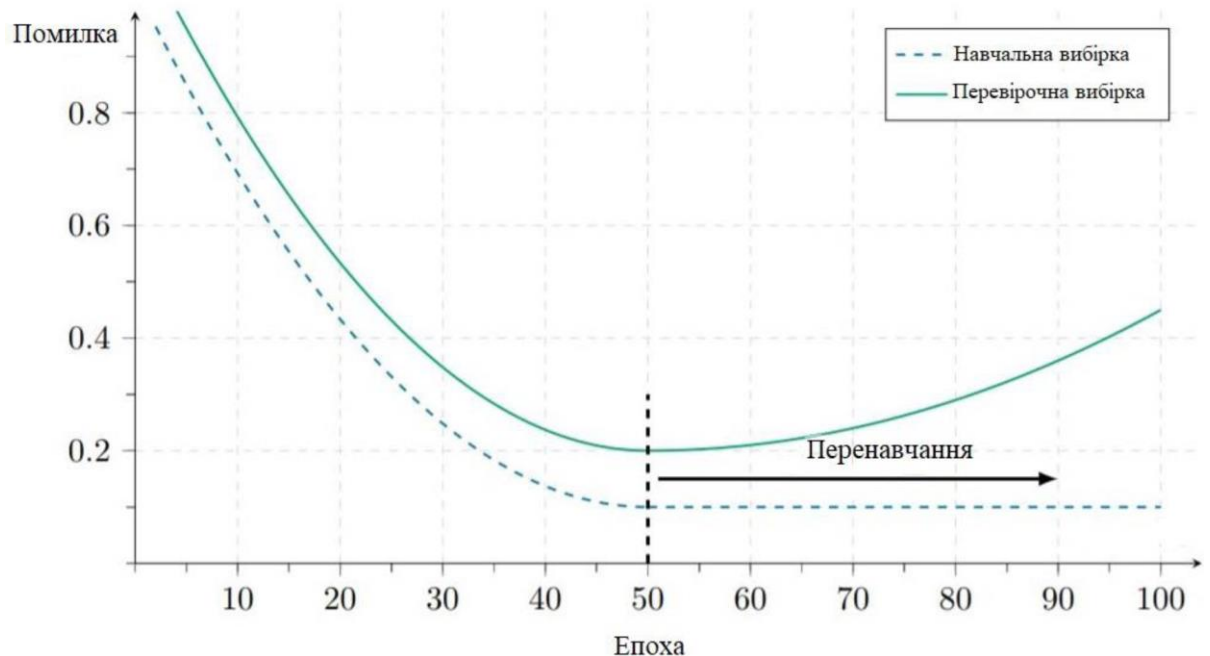


Рисунок 4.4 – Проблема перенавчання мережі

В роботі проводився експеримент на трьох видах моделей: класичної моделі, що складається з одного згорткового шару та одного шару субдискретизації; моделі, яка отримана при використанні запропонованого алгоритму з встановленими параметрами за замовчуванням (Рис. 4.6); та набору згорткових мереж, де вихідний сигнал набору розраховується як середнє арифметичне вихідних сигналів мереж.

використанням алгоритму Adadelta протягом десяти періодів, а тестування відбувалося на одному спільному наборі.

Таблиця 4. 2 - Аналіз трьох архітектур при навчанні на 10000 зразках

Модель класифікатора	Чутливість класифікатора
Архітектура 1	0,76
Архітектура 2	0,85
Архітектура 3	0,89

Запропонований в роботі підхід на основі набору ЗНМ продемонстрував найкращі показники якості, хоча результати, отримані за допомогою вдосконаленого в роботі алгоритму формування архітектур згорткових нейронних мереж теж показав гарний результат.

4.2 Розробка модулю планування

Модуль планування відповідає за створення плану дій, необхідного для досягнення кінцевої мети. На його вхід надходять данні про ознаки початкового стану, що ідентифікується всім набором згорткових нейронних мереж, і мета, яка була досягнута в результаті дії агента. Опрацювавши всю необхідну інформацію, модуль планування для початкового стану знаходить відповідний рядка в матриці суміжності і починає роботу з пошуку маршруту з цього стану до кінцевої мети. Програма написана на мові C++.

Основні переваги мови C++:

- підтримка об'єктно-орієнтованого програмування за допомогою класів;
- підтримка узагальненого програмування за допомогою шаблонів;
- доповнення до стандартної бібліотеки;

- додаткові типи даних;
- обробка винятків;
- простори імен;
- вбудовані функції;
- перевантаження оператора;
- перевантаження імені функції;
- вільно розподілені прив'язки та оператори керування пам'яттю.

Нейронні мережі отримують інформацію на вхідних рецепторах і видають на виході вектор характерних ознак початкових умов. На них формується вектор характерних ознак кінцевої цілі до якої повинні привести дії агента в цих вихідних умовах. Модуль визначення подібності ознак порівнює ознаки початкових умов з уже вивченими прецедентами.

Модуль планування працює в нескінченному циклі очікування інформації на сенсорних входах.

```
template <class Element> Element * Marcov <Element> :: Next (int
StartElement = -1)
{
if (Marcov <Element> :: Initiated)
{
if (StartElement == -1)
StartElement = Marcov <Element> :: Current;
std :: random_device rd;
std :: mt19937 gen (rd ());
std :: discrete_distribution <> dicr_distr (Marcov <Element> ::
AdjacencyMatrix.at (Current) .begin (),
Marcov <Element> :: AdjacencyMatrix.at (Current) .end ());next =
dicr_distr (gen);
if (next == Marcov <Element> :: size ())return NULL;
```

Всім початковим умовам формується матриця суміжності, яка описує певні дії. Рядки матриці суміжності вказують на ймовірність переходу до іншої дії, а стовпці вказують кількість нових дій.

```
int main(int argc, char *argv[])
{
    float act[9][9]={{0,1,0,0,0,0,0,0,0},
                    {0,0,0.8,0,0,0,0,0,0.2},
                    {0,0,0.1,0,0,0.9,0,0,0},
                    {0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,1,0},
                    {0,0,0,0,0,0,0,0,0},
                    {45,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,0}};

    int (*mas[9])(int);
    int jmax,imax,i,j;
    float amax;
```

Дії описуються як масив функцій, кожна з яких має однакову кількість параметрів. При моделюванні безумовного рефлексу програма знаходить по номеру рядка матриці відповідну функції з одновимірного масиву шаблонів. Далі фіксується номер стовпця, у якому знаходиться максимальна ймовірність. Номер стовпця i є ідентифікатором наступної дії.

Потім по рядку виконується пошук максимальної в ньому ймовірності переходу до наступної дії. Для цього зчитується номер стовпця, де знаходиться максимальна ймовірність. Номер стовпця є номером наступної дії.

Після порівняння ознак цілі та ознак реального образу, який отриманий після виконання дій, модуль планувальника коректує ймовірності переходів в матриці. Після завершення всіх дій, за допомогою рецепторів перевіряється їх успішність.

4.3 Висновки

У цьому розділі були проведені експерименти для оцінки ефективності моделей класифікаторів.

Результати експериментів показали, що модель, отримана з адаптивного класифікатора на основі набору згорткових нейронних мереж, забезпечує високу якість розпізнавання та перевищує чутливість класифікації звичайної згорткової нейронної мережі на 13%. А характеристики побудови множин запропонованим підходом, дозволяють адаптуватися до даних, що змінюються без істотної втрати точності.

На основі отриманих результатів можна зробити висновок, що запропонований підхід підвищив точність виявлення зображень.

Розроблено також програмну утиліту, яку можна використовувати як модуль планування, що є частиною інтелектуальної системи управління агентами.

Модуль планування призначений за створення плану дій, необхідного для досягнення кінцевої мети.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи була виконано завдання розробки підходу до інтелектуального управління технічними об'єктами, який базується на інтеграції сенсорних даних та їх обробці сучасними нейронними мережами.

У процесі реалізації роботи були вирішені наступні завдання:

1. Запропоновано адаптивний підхід до задачі управління технічними об'єктами, в якому поєднується обробка штучними нейронними мережами внутрішніх рецепторів агенту та сенсорних сигналів з зовнішнього середовища.

2. Вдосконалено модель згорткової нейронної мережі для визначення початкових умов керування та класифікації цілей керування.

3. З урахуванням особливостей складних об'єктів запропоновано модифікований метод найближчого сусіда. Ця модифікація полягає в тому, що застосовується спеціальний коефіцієнт K , що відповідає пороговому значенню ступеня схожості прецедентів і поточної ситуації. В результаті порівняння вибирається не тільки один найближчий сусід (прецедент), а й деякий набір близьких сусідів, ступінь схожості яких більше або дорівнює пороговому значенню ($S \geq K$).

4. На основі запропонованого модифікованого методу визначення найближчого сусіда було розроблено алгоритм отримання прецедентів із використанням різних метрик для визначення ступеня схожості прецедентів із поточною ситуацією. і з урахуванням коефіцієнтів, що характеризують важливість параметрів об'єкта.

5. Вдосконалено метод інтелектуального управління, що враховує класифікацію початкових умов та інтеграцію елементарних дій в послідовності з використанням ланцюгів Маркова.

6. Програмно реалізовано керуючу систему на основі розроблених методу і алгоритмів.

7. Результати експериментів показали, що модель, отримана з адаптивного класифікатора на основі набору згорткових нейронних мереж, забезпечує високу якість розпізнавання та перевищує чутливість класифікації звичайної згорткової нейронної мережі на 13%.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

2. Акинин, М.В. Нейросетевые системы искусственного интеллекта у задачах обработки изображений / М.В. Акинин, М.Б. Никифоров, А.И. Таганов. - М.: ГЛТ , 2016. - 152 с.
3. Бойчук В. О. Метод формування послідовності дій систем реального часу / В. О. Бойчук, М. В. Бойчук, С. М. Жовнір // Наука і економіка : наук.-теорет. журн. / Хмельниц. екон. ун-т. – Хмельницький, 2018. – Вип. 4 (48). – С. 133-137
4. Братко И. Алгоритмы искусственного интеллекта на языке Пролог. — М.: Вильямс, 2004. — 640 с.
5. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.:Техносфера, 2005. - 1072 с.
6. Глибовець М.М., Олецький О.В. Системи штучного інтелекту. — К.: КМ Академія, 2002. — 366 с.
7. Дорогов А.Ю. Быстрые нейронные сети. /А.Ю. Дорогов –СПб.: Издательство Санкт-Петербургского университета, 2002.- 80 с.
8. Девятков В.В. Системы искусственного интеллекта. — М.: Изд-во МГТУ им. Баумана, 2001. — 352 с.
9. Дубровін В.І. Методи оптимізації та їх застосування у задачах навчання нейронних мереж: Навчальний посібник/ В.І.Дубровін, С.О Субботін.- Запоріжжя: ЗНТУ, 2003.- 136 с.
10. Каллан Р. Основные концепции нейронных сетей. The Essence of Neural Networks First Edition. — 1-ше. — «Вильямс», 2001. — С. 288. — ISBN 5-8459-0210-X. (рос.) Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание. / Пер. с англ. - М.: Издательский дом "Вильямс", 2003, - 864 с.
11. Люгер Дж. Искусственный интеллект. Стратегии и методы решения сложных проблем. — М.: Вильямс, 2003. — 864 с.

12. Методи захисту від кіберзагроз у бездротових мережах / І. В. Гурман, В.Ю. Тітова, І.І. Островський, Л.О. Ряба // Тези доповідей Всеукраїнської НПК молодих вчених, ад'юнктів, слухачів, курсантів та студентів "Молодіжна військова наука у Київському національному університеті імені Тараса Шевченка". – К. : ВІКНУ, 2021. – С. 105-106.
13. Муляр І.В. Біокомп'ютери та методики їх програмування. / В. О. Бойчук, .І. В. Муляр, Ю. О. Царьов // Сучасна спеціальна техніка. - 2014. - № 2(37). - С. 54-60.
14. Проблеми класифікації додатків компютерних мереж / І. В. Муляр, І.І. Островський, // Тези доповідей XVII Міжнародної науково-практичної конференції " Військова освіта та наука: сьогодення та майбутнє". – К. : ВІКНУ, 2021. – С. 66-67.
15. Представление и использование знаний / Под ред. Уэно Х., Исидзука М. — М.: Мир, 1989. — 220 с.
16. Рассел, С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. - М.: Вильямс, 2016. - 578 с.
17. Робототехніка [Електронний ресурс] Режим доступу: <https://uk.wikipedia.org/wiki/Робототехніка>. Дата звернення: 08.05.2021
18. Руденко О. Г., Бодянський Є. В. Штучні нейронні мережі: Навчальний підручник. — Харків: ТОВ "Компанія СМІТ", 2006. — 404 с.
19. Смолин Д.В. Введение в искусственный интеллект: конспект лекций. — М.: ФИЗМАТЛИТ, 2004. — 208 с.
20. Слэйгл, Дж. Искусственный интеллект / Дж. Слэйгл. - М.:Мир, 2016. - 320 с.
21. Субботін С.О. Подання та обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Запоріжжя: ЗНТУ, 2008. — 341 с.
22. Ayyadevara, V. K. Convolutional Neural Network. In: Pro Machine Learning Algorithms / V. K Ayyadevara - Berkeley, CA : Apress, 2018. - С. 179–215.

23. A.M. Aibinu, A.J. Onumanyi, M. Ipinoyomi, T.A. Folorunso, A.P. Adedigba, M.J.E. Salami, «De-velopment of hybrid artificial intelligent based handover decision algorithm», *Engineering Science and Technology an International Journal*, 2017. – V.20 (2), pp. 381–390
24. Abar, S., Theodoropoulos, G.K., Lemarinier, P., OHare, G.M., 2017. Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review* 24, 13-33.
25. Deep Learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville. – 2016. – 800 p
26. Graves, A. Long Short-Term Memory. In: *Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence*,. Springer, Berlin, Heidelberg, 2021 (Vol. 385)
27. Hiai, Fumio; Lin, Minghua (February 2017). "On an eigenvalue inequality involving the Hadamard product".
28. J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281-305, Feb. 2012. [Электронный ресурс] Режим доступа: <http://jmlr.csail.mit.edu/papers/volume13/bergstra12a/bergstra12a.pdf>
29. J. Zhao, M. Mathieu et al., “Stacked what-where auto-encoders,” *arXiv preprint arXiv:1506.02351*, Jun. 2015. [Электронный ресурс] Режим доступа: <https://arxiv.org/abs/1506.02351v1>
30. Kravari K., Bassiliades N. A Survey of Agent Platforms. *Journal of Artificial Societies and Social Simulation*. 2015. Vol. 18, no. 1. P. 1–18.
31. Mariya Yao *Applied Artificial Intelligence: A Handbook For Business Leaders* Kindle Edition, Publisher: TOPBOTS, 2018, 246 p.
32. M. Thoma, “The HASYv2 dataset,” *arXiv preprint arXiv:1701.08380*, Jan. 2017. [Электронный ресурс] Режим доступа: <https://arxiv.org/abs/1701.08380>
33. N. Srivastava, G. E. Hinton et al., “Dropout: a simple way to prevent neural networks from overfitting.” *Journal of Machine Learning Research*, vol. 15, no. 1, pp.

1929-1958, 2014. [Электронный ресурс] Режим доступа:
<https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf>

34. Stephen Marsland. Machine Learning: An Algorithmic Perspective / Stephen Marsland. – 2015. – 452 p.,

35. Y. LeCun, B. Boser, J. S. Denker, R. E. Howard, D. Henderson, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4):541 -551, Winter 1989

ДОДАТОК А

(обов'язковий)

Лістинг основного компонента модуля планування дій

```

template <class Element> Element * Markov <Element> :: Next (int
StartElement = -1)
{
if (Markov <Element> :: Initiated) // якщо матриця суміжності створена
{

if (StartElement == -1) // якщо стартовий елемент за замовчуванням
StartElement = Markov <Element> :: Current; // то продовжуємо (в
конструкторі Current = 0)

std :: random_device rd;
std :: mt19937 gen (rd ());
std :: discrete_distribution <> dicr_distr (Markov <Element> ::
AdjacencyMatrix.at (Current) .begin ()),
Markov <Element> :: AdjacencyMatrix.at (Current) .end ()); //
ініціалізуємо контейнер для генерації числа на основі розподілу
ймовірності

int next = dicr_distr (gen); // генеруємо наступну вершину
if (next == Markov <Element> :: size ()) // якщо розподіл ймовірностей
нульовий, то він повертає кількість елементів
return NULL;

Markov <Element> :: Current = next; // міняємо поточну вершину
return & (Markov <Element> :: elems.at (next)); // повертаємо значення
в вершині
}
return NULL;
}
#include <iostream>

```

```
#include "Marcov.h"
#include <string>
#include <fstream>

using namespace std;

int main ()
{
    Marcov <string> chain;
    ofstream outs;
    outs.open ( "out.txt");
    ifstream ins;
    ins.open ( "matrix.txt");

    int num;
    double Prob = 0;
    (Ins >> num) .get (); // кількість вершин
    string str;
    for (int i = 0; i <num; i ++)
    {
        getline (ins, str);
        chain.AddElement (str); // додаємо вершину
    }
    if (chain.InitAdjacency ()) // ініціалізуємо матрицю нулями
    {
        for (int i = 0; i <chain.size (); i ++)
        {
            for (int j = 0; j <chain.size (); j ++)
            {
                (Ins >> Prob) .get ();
                if (! chain.PushAdjacency (i, j, Prob)) // вводим матрицю
                {
                    cerr << "Adjacency matrix write error" << endl;
                }
            }
        }
    }
}
```

```

}

outs << chain.At (0) << ""; // виводимо 0-ю вершину
for (int i = 0; i < 20 * chain.size () - 1; i ++) // генеруємо 20
ланцюжків
{
string * str = chain.Next ();
if (str != NULL) // якщо попередня не кінцева
outs << (* str) .c_str () << ""; // виводимо значення вершини
else
{
outs << std :: endl; // якщо кінцева, то починаємо з початку
chain.Current = 0;
outs << chain.At (0) << "";

}
}

chain.UninitAdjacency (); // зрозуміло
}
else
cerr << "Can not initialize Adjacency matrix" << endl ;;

ins.close ();
outs.close ();
cin.get ();
return 0;

#include <stdlib.h>
#include <stdio.h>

int func1(int a)
{
    return a;
};

```

```
int func2(int a)
{
    return a+1;
};
int func3(int a)
{
    return a+2;
};
int func4(int a)
{
    return a+3;
};
int func5(int a)
{
    return a+4;
};
int func6(int a)
{
    return a+5;
};
int func7(int a)
{
    return a+6;
};
int func8(int a)
{
    return a+7;
};
int func9(int a)
{
    return a+8;
};
int main(int argc, char *argv[])
{
    float act[9][9]={{0,1,0,0,0,0,0,0,0},
```

```

                                {0,0,0.8,0,0,0,0,0.2,0},
                                {0,0,0.1,0,0,0.9,0,0,0},
                                {0,0,0,0,0,0,0,0,0},
                                {0,0,0,0,0,0,0,0,0},
                                {0,0,0,0,0,0,0,1,0},
                                {0,0,0,0,0,0,0,0,0},
                                {45,0,0,0,0,0,0,0,0},
                                {0,0,0,0,0,0,0,0,0}};

int (*mas[9])(int);
int jmax,imax,i,j;
float amax;

mas[0] = func1;
mas[1] = func2;
    mas[2] = func3;
mas[3] = func4;
mas[4] = func5;
mas[5] = func6;
    mas[6] = func7;
mas[7] = func8;
mas[8] = func9;
i=2;
printf(" %d\n",mas[i](1));
for(;;)
{ amax=act[i][0];
  if(amax>1) goto a1;
  for(j=0;j<9;j++)
  {
  if(act[i][j]>amax)
  {
      jmax=j;
      amax=act[i][j];
      imax=i;
  }
}
}

```

```
    }  
  
    printf(" %d\n",mas[jmax](1));  
    i=jmax;  
}  
  
a1:return 0;  
};
```

ДОДАТОК Б

(обов'язковий)

Копія праць по результатам дослідження

к.т.н., доц. Муляр І.В. (ХмНУ)

Островський І.І. (ХмНУ)

Класифікація додатків комп'ютерних мереж

Штучний інтелект - це методи роботи з інформацією, засновані на швидкому використанні знань. Подальша перспектива - технічний розвиток формалізації вмінь та творчих здібностей людини - креативності і її взаємодоповнюваність методами штучного інтелекту. Зрозуміло, інтелектуальні системи управління значно розширюють можливості агентів з освоєння все більш складних операцій. Однак інтелектуальний агент як і раніше вимагає постійного спостереження з боку людини, особливо в зв'язку з можливістю виникнення аварійних та нештатних ситуацій.

У системах, що працюють в режимі реального часу, агент-робот повинен раціонально вирішувати поставлені перед ним завдання з мінімальними витратами необхідних ресурсів. Використання високоточних методів багатокритеріальної оптимізації в реальному часі утруднено, тому агенти зазвичай вирішують задачу апроксимаційними методами з використанням проблемно-орієнтованих евристичних підходів [1].

Проблемами розробки агентів є використовувані ресурсомісткі алгоритми, висока складність і узгодженість моделей. Ці фактори мають негативний вплив на якість агентів, знижують їх обслуговування і продуктивність, а також збільшують витрати на розробку. Додатковим ускладненням є динамічний характер середовища, в якому повинен працювати агент, оскільки стан середовища може значно змінитися в процесі прийняття рішення. Агент повинен адаптуватися до таких змін в найкоротші терміни.

Крім того, досить складно розробити програму, яка була б планом дій для агенту. Тому що зазвичай існує багато способів досягнення цієї мети, і вам потрібно знайти найкращий можливий план. Пошук рішення займає багато часу, тому що це означає повний пошук.

Аналіз матеріалів з проблеми керування показує, що створення ефективної системи управління вимагає використання якісно нових підходів до обробки інформації, які повинні ґрунтуватися на виявленні особливостей дій у минулому та їх адаптації на основі ієрархічного представлення дії [2]. Найбільш перспективним напрямом створення таких систем є використання сучасних нейронних мереж для класифікації прецедентів та формування нових дій на їх основі.

Метою дослідження є розробка теоретичних принципів та практичних підходів до управління технічними об'єктами, характерною рисою яких є інтеграція сенсорної обробки даних сучасними нейронними мережами з інтелектуальним управлінням на їх основі послідовністю дій.

Список використаних джерел:

1. Abar, S., Theodoropoulos, G.K., Lemarinier, P., OHare, G.M., 2017. Agent based modelling and simulation tools: A review of the state-of-art software. Computer Science Review 24, 13-33.

УДК 681.5

к.т.н., доц. Муляр І.В.. (ХМНУ),
к.т.н., доц. Орленко В.С. (ХМНУ),
Островський І.І. (ХМНУ)

АДАПТИВНИЙ МЕТОД КЕРУВАННЯ АВТОМАТИЗОВАНИМИ ТЕХНІЧНИМИ СИСТЕМАМИ

У статті запропоновано підхід до процесу керування технічними об'єктами на основі інтеграції обробки сенсорних даних.

Інтелектуальні методи управління знайшли застосування в різних завданнях, зокрема в робототехніці. У системах, що працюють в режимі реального часу, робот-агент повинен раціонально вирішувати задачі, які перед ним поставлені з мінімальними витратами ресурсів. У реальному часі використання високоточних багатокритеріальних методів оптимізації утруднене, тому агенти часто вирішують задачу наближеними методами з використанням проблемно-орієнтованої евристики. Аналіз матеріалів з проблеми управління показує, що створення ефективної системи управління потребує використання якісно нових підходів до обробки інформації, які мають базуватися на пошуку особливостей дій у минулому, їх адаптації на основі ієрархічного представлення дій. Найбільш перспективним напрямком у створенні таких систем є використання сучасних нейронних мереж для класифікації прецедентів та формування нових дій на основі підходу прецедентів.

При розробці моделі поведінки агента пропонується розширити стандартні рішення, використавши елементи біологічного підходу в штучному інтелекті. Для цього агентська діяльність здійснюється через взаємодію класифікаційної та виконавчої сторін. В якості класифікаційної частини використовуються сучасні типи штучних нейронних мереж, згорткових в нашому випадку. За виконавчу частину відповідають моделі на основі ланцюгів Маркова.

Штучна нейронна мережа отримує та класифікує інформацію від різних типів зовнішніх сенсорів і внутрішніх рецепторів агента, щоб ідентифікувати початкові умови дії агента, мету дії та визначити на їх основі послідовність дій, які виконує агент.

Використання адаптивного методу полягає в підміні мети та початкових умов обраного прецеденту новою задачею і початковими умовами і забезпечені виконання плану після підміни. Деякі кроки початкового плану можуть виявитися непотрібними, так як змінилася мета, досягнення якої вони прагнули. Основні дії можна виконувати окремо або комбінувати в послідовності в залежності від успішності поставленої мети.

Ключові слова: інтелектуальне керування, багатоагентні системи, сенсорна інформация

Вступ. Робототехніка (від [робот](#) та [техніка](#); [англ. robotics](#)) — прикладна [наука](#), яка опікується проєктуванням, розробкою, будівництвом, а також експлуатацією та використанням роботів, спеціалізованих комп'ютерних систем для їх контролю, сенсорного (на основі аналізу вихідних сигналів [давачів](#)) зворотного зв'язку і опрацювання інформації [автоматизованих](#) технічних систем ([роботів](#)) [1]. Перші роботи керувалися простими командами, запозиченими разом із приводами в станках. Для промислових зразків це стало можливим із-за крайнього детермінізму умов промислового виробництва. З розвитком науки і техніки, підтримкою інноваційних технологій робототехніка перетворилася в самостійну наукову сферу. Головною особливістю будь-якого механізму і робота є його корисність. Залежно від корисності [автоматизованих](#) технічних систем в тій чи іншій сфері життя заведено виділяти такі різновиди роботів:

- медичні;
- побутові;
- бойові;
- дослідні;
- промислові та будівельні;
- ігрові та ін.

Також їх можна розділити на керовані і автономні; мобільні та стаціонарні.

У сучасній медицині роботи виготовляються серійно, і без багатьох із них складні діагностичні процедури були б майже неможливі. У 1985 році робот Unimation Puma 200 приймав участь у взятті біопсії мозку в пацієнта, що стало значним проривом робототехніки у медицині. Пізніше, через 7 років спеціалізований робот ProBot зробив вперше у світі самостійно операцію

Однак, коли робототехніка розпочинає розповсюджуватися в інших областях з більшою невизначеністю та мінливістю зовнішніх впливів, із програмного керування довелось перейти до управління від оператора, доповнюючи програмне управління диспетчерським управлінням.

Наступним етапом стала розробка на цій основі адаптивних систем управління з використанням подальших методів штучного інтелекту та переходу до парадигми багатоагентних систем. Останні, крім керування, використовувалися для виконання інших функцій роботів, таких як обробка сенсорної інформації та формування моделей взаємодії навколишнього середовища з оператором.

Багатоагентна система ([англ. Multi-agent system](#)) — це система, що утворена декількома взаємодіючими [інтелектуальними агентами](#). Ці системи можуть бути використані для вирішення таких проблем, які складно або неможливо розв'язати за допомогою одного агента чи [монолітної системи](#). Прикладами таких завдань є керування критичною інфраструктурою, онлайн-торгівля, ліквідація надзвичайних ситуацій, та моделювання соціальних структур [2].

Штучний інтелект ([англ. artificial intelligence, AI](#)) — розділ [інформатики](#) та [комп'ютерної лінгвістики](#), що опікується формалізацією проблем та завдань, що подібні до дій, які виконує людина. Подальша перспектива - технічний розвиток формалізації вмінь та творчих здібностей людини - креативності і її взаємодоповнюваність методами штучного інтелекту. Це дозволить повноцінно відтворити в конкретних прикладних областях розумові здібності людини, які реалізуються двома півкулями нашого мозку - лівим, де зосереджено переважно логічне мислення, і правим, що відповідає за творчі здібності та креативність людини. Зрозуміло,

інтелектуальні системи управління значно розширюють можливості агентів з освоєння все більш складних операцій. Однак інтелектуальний агент як і раніше вимагає постійного спостереження з боку людини, особливо в зв'язку з можливістю виникнення аварійних та нештатних ситуацій [3, 4].

Він не може тривалий час автономно функціонувати в нестационарному середовищі, тому що всі його дії строго формалізовані, і в цих умовах обов'язково вимагається інтуїція, креативність, творчість.

У системах, що працюють в режимі реального часу, особливо в критичній інфраструктурі, агент-робот повинен раціонально вирішувати поставлені перед ним завдання з мінімальними витратами необхідних ресурсів. Передбачається, що робот може виконувати деякі базові дії. Він намагається виконувати свої дії таким чином, щоб поставлена перед ним мета була досягнута. Використання високоточних методів багатокритеріальної оптимізації в реальному часі утруднено, тому агенти зазвичай вирішують задачу апроксимаційними методами з використанням проблемно-орієнтованих евристичних підходів [5].

Проблемами розробки агентів є використовувані ресурсомісткі алгоритми, висока складність і узгодженість моделей. Ці фактори мають негативний вплив на якість агентів, знижують їх обслуговування і продуктивність, а також збільшують витрати на розробку. Додатковим ускладненням є динамічний характер середовища, в якому повинен працювати агент, оскільки стан середовища може значно змінитися в процесі прийняття рішення. Агент повинен адаптуватися до таких змін в найкоротші терміни.

Крім того, досить складно розробити програму, яка була б планом дій для агента. Тому що зазвичай існує багато способів досягнення цієї мети, і вам потрібно знайти найкращий можливий план. Пошук рішення займає багато часу, тому що це означає повний пошук. Очевидно, що це не раціонально, тому розробляються складні евристики, які можуть зменшити область пошуку до розумного розміру.

Аналіз наукових досліджень та постановка задачі. Аналіз матеріалів з проблеми керування за допомогою ШІ показує, що створення ефективної системи управління вимагає використання якісно нових підходів до обробки інформації, які повинні ґрунтуватися на виявленні особливостей дій у минулому та їх адаптації на основі ієрархічного представлення дії [5, 6]. Найбільш перспективним напрямом створення таких систем є використання сучасних нейронних мереж для класифікації прецедентів та формування нових дій на їх основі.

Розглянувши підходи до керуванні технічними об'єктами можна зробити висновок, що більшість з них сильно формалізовані і спираються при цьому на формальну обробку предикатів. Вони практично не інтегровані з реальним середовищем, з яким стикається агент.

Технічні штучні нейронні мережі: основний і незмінний засіб роботи з такою інформацією, хоча б тому, що вони, хоча й досі дуже спрощені, але аналогічні "елементарній основі" мозкової діяльності. Накопичення, зберігання та обробка інформації зображення можуть бути реалізовані за допомогою нейронних мереж, які базуються на формальних нейронних мережах з традиційною пороговою логічною обчислювальною базою. Однак традиційні версії нейронних мереж, такі як багатошарові перцептрони, мережі Хопфілда або Кохонена, неефективні при роботі зі складними динамічними зображеннями з невизначеністю, що особливо актуально для систем управління роботами [7, 8].

Згорткова (конволюційна) нейронна мережа складається з локальних мереж, які моделюють окремі ділянки рецептивного поля, в яких виконується однакова процедура послідовного узагальнення відповідних компонентів вихідного зображення [9].

Тому метою роботи є розробка підходу до інтелектуального управління технічними об'єктами, який базується на інтеграції сенсорних даних та їх обробці сучасними нейронними мережами.

Основна частина. Ми розглядаємо управління складними технічними об'єктами з точки зору парадигми агентно-орієнтованого моделювання штучного інтелекту.

Ключове відмінність між агентно-орієнтованим підходом полягає в тому, що агенти є суб'єктами в процесі моделювання і мають бажання, мету і здатність виконувати дії, в той час як об'єктно-орієнтований підхід передбачає, що програма виконує операції з об'єкними об'єктами.

Багатоагентні системи реального часу - ефективний інструмент для моделювання складних процесів, в яких задіяна велика кількість активних автономних одиниць. Ці процеси включають міські транспортні потоки, логістичні системи, соціальні явища і епідемії. Методи багатоагентного моделювання також використовуються для пошуку і обробки даних в інформаційних мережах, автономних системах управління. Перспективним напрямком подальшого розвитку мультиагентних систем є розробка безпілотних літальних апаратів і автомобілів.

У багатоагентних системах реального часу агент повинен раціонально вирішувати поставлені перед ним завдання з мінімумом ресурсів. Використання високоточних методів багатокритеріальної оптимізації в реальному часі утруднено, тому агенти зазвичай вирішують задачу апроксимаційними методами з використанням проблемно-орієнтованих евристик [10].

Проблеми розробки агенту - це алгоритми, які використовують багато ресурсів, висока складність та узгодженість моделей. Ці фактори досить негативно впливають на якість агентів, знижують їх утримання та продуктивність, збільшують вартість розробки. Додатковим ускладненням є динамічний характер середовища, в якому агент повинен діяти, оскільки стан навколишнього середовища може значно змінюватися під час прийняття рішень. Агент повинен пристосуватися до таких змін у найкоротші терміни.

На рисунку 1 зображена функціональна схема агента-робота зі штучним інтелектом [11]. Його основний та обов'язковий компонент - наявність розвиненої пам'яті, основа розумних дій робота та рівень його інтелекту. блок пам'яті пов'язаний з іншими системами, що обробляють інформацію, і він включає базу знань про зовнішнє середовище у вигляді його моделі та базу даних про це середовище, самого робота та операції, які вона може виконувати. Крім того, деякі неспеціалізовані оперативні бази знань та дані, пов'язані з центральною пам'яттю, можуть знаходитися в окремих роботосистемах.

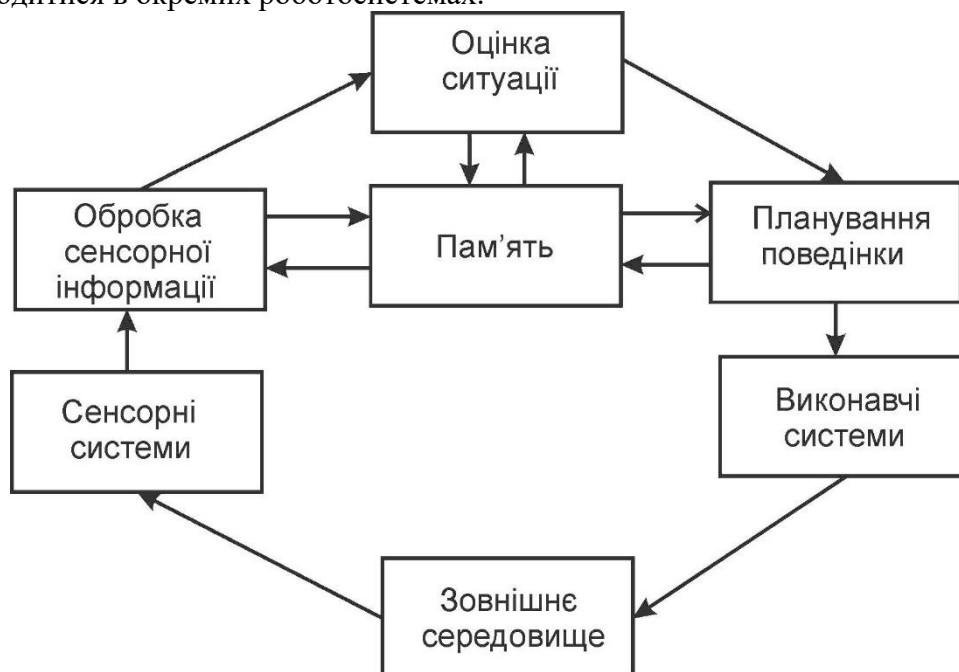


Рисунок 1 - Функційна схема інтелектуального агента-робота

База знань зовнішнього середовища містить апіорну інформацію, введену до початку роботи, та оперативну сенсорну інформацію, отриману в процесі сприйняття навколишнього

середовища, коли робот виконує різні дії, а також у процесі своїх особливих пізнавальних дій вивчити це середовище. Сама інформація містить опис геометричних та інших фізичних характеристик об'єктів навколишнього середовища та їх взаємозв'язків. Цей опис має ієрархічну структуру у вигляді рівнів послідовного узагальнення вихідної інформації. Наприклад, опис робочої зони маніпулятора містить набір площин цієї області та її частин, які відрізняються масштабом і точністю, та ступенем узагальнення первинної сенсорної інформації (виділення контурів, об'єктів, поверхонь, груп об'єктів, визначення різних властивостей та фізико-хімічних властивостей цих об'єктів тощо).

Додавання до цих планів часу як параметру, дає зображення зовнішнього середовища в динаміці з урахуванням взаємодії його об'єктів між собою та з роботом. База знань зовнішнього середовища також містить правила, які дозволяють моделювати можливі зміни в цьому середовищі.

Всі інші блоки схеми мають деяку ієрархічну структуру, рівні якої з'єднані між собою вертикально знизу вгору у напрямку узагальнення інформації. У свою чергу, це показано на блок-схемі з'єднання в загальному багатоканальному випадку у вигляді з'єднань між тими ж рівнями по горизонталі.

Блок опрацювання сенсорної інформації отримує від блоку пам'яті екстраполяцію змін стану навколишнього середовища і передає йому виправлення цього стану на рівні прямого сенсорного зображення навколишнього середовища.

Блок оцінки стану та блок планування поведінки зчитують з блоку пам'яті поточну модель середовища та передають їй відповідно свою оцінку за певними критеріями та синтезований план управління рухом робота згідно із завданнями. Завдання модулю оцінки ситуації також включає швидке коригування цілей та пріоритетів управління. Це найвищий рівень в ієрархії керування роботами.

Загалом, інтерфейсний блок може бути двосторонньо пов'язаний з усіма перерахованими функціональними одиницями. Окрім оператора - людини, він забезпечує спілкування з іншими командами, які працюють разом, включаючи іншу роботу.

У міру вдосконалення систем управління робототехнічними агентами перелік операцій, в яких домінують роботи, постійно розширюється. Однак навіть у сучасній промисловості все ще існує значна кількість технологічних операцій, які не повністю автоматизовані. Особливо це стосується непромислових роботів, таких як екстремальна робототехніка. Крім того, прогресуюче ускладнення технічних систем та функцій, які вони виконують, безперервно загострює цю ситуацію, тому людина продовжуватиме залишатися необхідною складовою роботизованих систем, головним чином вищим рівнем управління. Її завдання - керувати операціями, які неможливо автоматизувати, і навчити роботів виконувати ці операції з поступовим переходом до автоматичного режиму [12].

В цілому проблема створення "штучного інтелекту" виходить за межі робототехніки як глобальної проблеми, можна сказати, розвитку людської цивілізації. Той факт, що вона найгостріше зіткнулася з робототехнікою, пояснюється самою історією та сутністю робототехніки, для якої людина була еталоном з самого початку.

Робот має двосторонню інформаційну та енергетичну взаємодію із зовнішнім середовищем. І в цьому сенсі вона повинна бути схожою на живих істот і, отже, здатна до самонавчання та вдосконалення особистості.

В загальному опис завдання планування полягає в тому, що активний елемент (агент) виконує послідовність дій в деякому середовищі і прагне досягти поставленої мети.

У кожен проміжок часу середовище знаходяться в деякому стані, при цьому дії агента можуть змінюють стан середовища.

Завдання планування - знайти послідовність дій, які дозволяють агенту перевести систему з початкового стану в заданий цільовий стан.

У загальному випадку ціль може складатися з кількох станів, досягнення будь-якого з них означає досягнення мети. Також можливо, що жоден з цих станів є недоступним. Формально в задачі планування дається система агент-середовище [12]:

$$M = (Q, A, q_0, G_M), \quad (1)$$

де Q - множина спостережуваних станів;

A – множина можливих дій;

q_0 – початковий стан;

I - початкові умови;

G - множина цільових станів;

$\Gamma_M : Q \times A \Rightarrow Q$ функція переходу, яка визначає для кожного стану $q \in Q$ і дії $a \in A$ наступний стан $q' = \Gamma_M(q, a)$.

Потрібно знайти план, який є впорядкованою множиною дій $P = \{ a_1, \dots, a_n \}$, який є суперпозицією функцій переходу $\Gamma_M(\Gamma_M(\dots \Gamma_M(\Gamma_M(q_0, a_1) a_2) \dots, a_{n-1}), a_n)$ і належить G при $q_0 \in I$.

Відповідно до формули 1.1 процес керування роботом виглядає так. Існують початкові умови $q_0 \in I$ для ініціалізації процесу керування. Якщо брати до уваги процес управління роботом-агентом, то це може бути, наприклад, розряд батареї робота або технічний зір робота при небезпечному швидкому переміщенні великого об'єкта в напрямку робота.

Початкові умови визначають мету операції робота $g \in G$. У першому випадку мета визначається як отримання заряду акумулятора. У другому випадку метою може бути переміщення робота на безпечну відстань від об'єкта загрози (зменшення розмірів об'єкта в полі зору робота).

Залежно від початкового стану та цілі керування необхідно знайти суперпозицію функцій переходу між станами Γ_M (впорядкована множина дій $P = \{ a_1, \dots, a_n \}$), яка призводить робота в кінцевий стан.

Тобто агент повинен визначити початковий стан. У разі заряду батареї внутрішні сенсори необхідні для визначення ступеня розряду батареї, в останньому випадку зовнішні - для організації технічного зору.

Далі необхідно визначити і класифікувати датчики індикаторів. В організмі людини це здійснюється за допомогою нейронних мереж, як із зовнішнього середовища, так і з індикаторів внутрішнього стану робота-агенту.

Використовуючи адаптивний підхід, для ідентифікації початкового стану доцільно використовувати штучні нейронні мережі. На сьогоднішній день розроблено багато типів нейронних мереж. Нам потрібно вибрати серед них дуже схожі на обробку інформації людським мозком, і з їх успішною реалізацією в області обробки зображень. Для цього ми вибираємо згорткові нейронні мережі, які сьогодні досить успішно використовують Google, Facebook, Pinterest та багато інших [6].

На виході нейронної мережі отримуємо клас початкових умов, за якими можна сформулювати мету і вивести порядок роботи робота.

Тут необхідно робити логічні висновки і планувати дії. Тут також потрібно використовувати аналогію з розумними біологічними системами.

Сучасні нейронні мережі часто вирішують стандартні завдання класифікації, але методи планування дій відокремлені від проблеми класифікації і є високоформалізованими. Тому актуальним завданням є розробка підходу до інтеграції підсистеми планування з існуючими нейронними мережами.

Якщо розглянути дії інтелектуального робота-агенту в небезпечній ситуації, коли великий об'єкт швидко рухається на нього, то нейронна мережа в першу чергу має бути налаштована на ідентифікацію великих об'єктів. Робот повинен розвернутися і якомога швидше відійти від

джерела небезпеки, але при цьому необхідно врахувати низький заряд батареї робота. Інші параметри менш важливі і можуть бути оброблені пізніше.

При розробці адаптивного підходу ми будемо використовувати аналогії в діяльності людського мозку. Діяльність людського мозку ще далеко не розкрита, але деякі моменти загально відомі і вже використовуються в сучасних системах штучного інтелекту.

Ми формулюємо основні принципи адаптивного підходу до управління агентами:

1. Нейронна мережа здійснює прийом та класифікацію інформації як від зовнішніх датчиків, так і від внутрішніх сенсорів агента, щоб ідентифікувати початкові умови дії агента, мету дії та визначити на основі мети послідовність дій, що виконує агент.

2. Нейронна мережа може обробляти дані від різних типів датчиків (оптичних, акустичних тощо).

3. Існує перелік елементарних дій, що може виконувати агент.

4. Ці дії можуть виконуватися окремо або об'єднуватися в послідовності дій на основі досягнення мети.

5. Нейронна мережа зчитує характеристики послідовності дій через рецептори для організації зворотного зв'язку та ідентифікації поточного стану.

6. Компоненти штучної нейронної мережі можуть класифікувати різні показники об'єктів навколишнього середовища з різною швидкістю. Наприклад, ви можете класифікувати розмір і швидкість об'єкта швидше, ніж вказувати, який об'єкт знаходиться в ієрархії, поступово деталізуючи оброблені показники.

7. Початкові умови і мета мають ступінь важливості, яка може змінюватися і безпосередньо впливати на порядок дій.

На рисунку 2 наведено загальну схему агента, який працює згідно запропонованого адаптивного методу з врахуванням принципів адаптивного підходу.

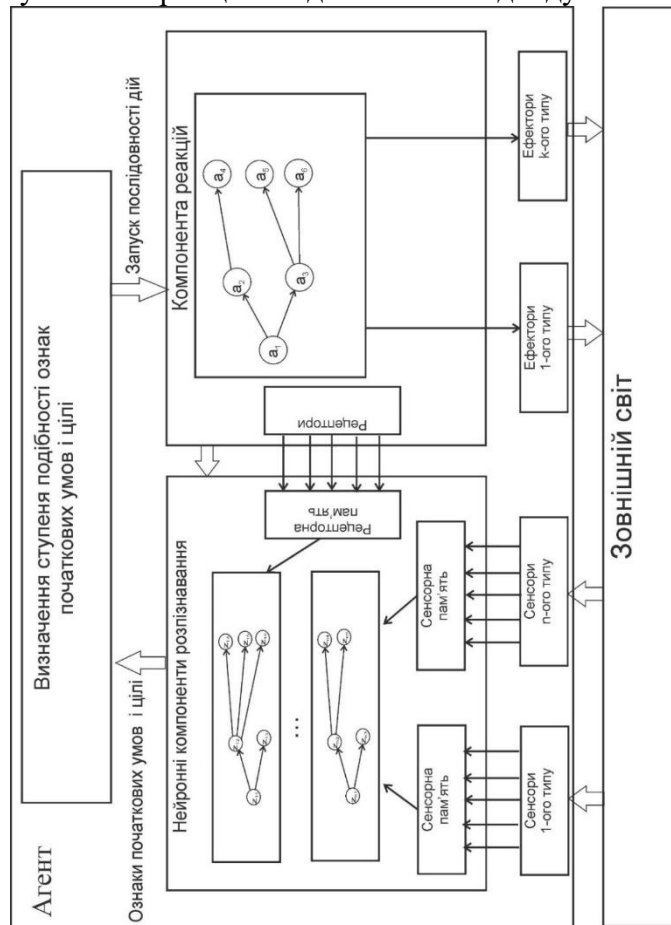


Рисунок 2 – Схема організації інтелектуального управління при адаптивному підході

Функціонування агента складається з послідовності елементарних дій $a_1, a_2, a_3, \dots, a_n$, що можуть бути використані агентом. Всі елементарні дії в загальному випадку мають певний скінчений набір вхідних параметрів для необхідного налаштування її функціонування x_1, x_2, \dots, x_m . Прикладами елементарних дій робота є рухи робота вперед, назад та повороти, а параметрами можуть виступати швидкість, тривалість руху та ін.

При проектуванні робота виділяються стандартні послідовності дій, що формують бібліотеку і відповідають певним рефлекторним реакціям людини на різні вхідні умови для досягнення мети. Стандартні послідовності дій спрацьовують при активізації певного набору ознак на вихідному шарі мережі.

Послідовність дій агенту може бути описана орієнтованим графом на основі ланцюга Маркова, де стани графу відповідають елементарним діям, а ребра відповідають вазі зв'язку між ними [13]. Знаки початкової умови q пов'язані з першою вершиною, з якої починається послідовність дій. Ознаки мети g пов'язані з останньою вершиною послідовності дій. Коли вага зв'язку між вершинами дорівнює одиниці, тоді перехід між вершинами строго визначений. Коли він відмінний від одиниці, перехід у загальному випадку не є строго обумовленим.

Успішне досягнення мети так чи інакше впливає на вагу ребер. Тобто збільшує вагу всіх ребер на шляху при успішному досягненні мети і навпаки.

На рисунку 3 наведено граф послідовності дій на основі ланцюга Маркова, яким можна представити безумовний рефлекс.

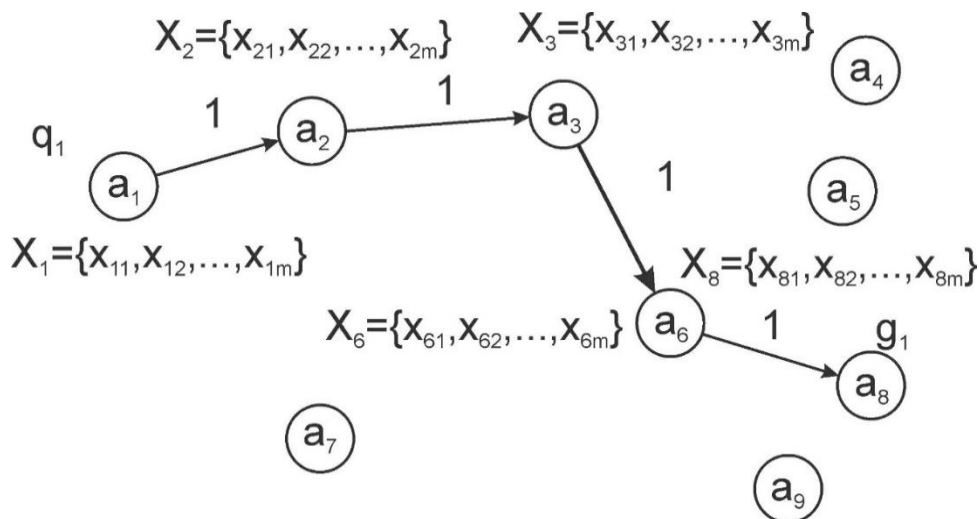


Рисунок 3 - Граф, який моделює безумовний рефлекс послідовності дій агенту

Тобто з рисунку чітко видно, що при деякій початковій умові виконується послідовність дій a_1, a_2, a_3, a_6, a_8 , показана на графі.

На рисунку 4 наведено граф послідовності дій, яка може відповідати умовному рефлексу агенту. Тобто з вершини може бути кілька сценаріїв, ваги кількох ребер з вершини в сумі повинні дорівнювати одиниці [12]. За замовчуванням вибирається наступна елементарна дія, яка надходить з вершини з найбільшою вагою.

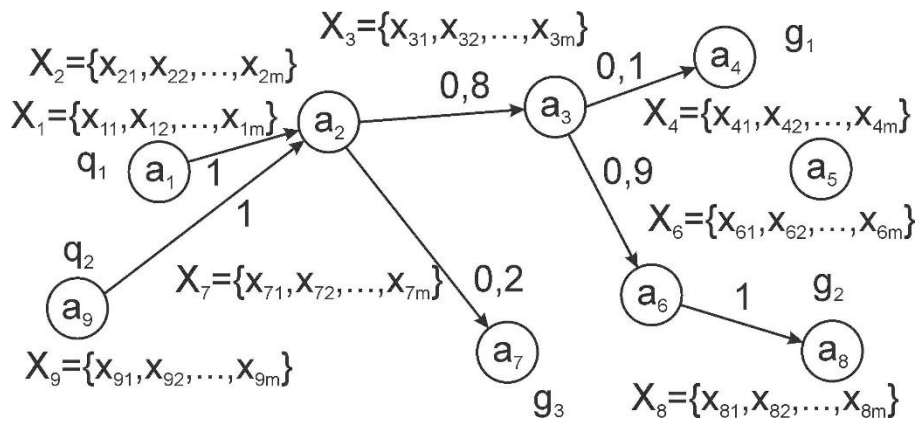


Рисунок 4 - Граф, який моделює умовний рефлекс послідовності дій агента

В цьому випадку при розвитку подій, яка розпочинається з початкової умови q_1 , виконується послідовність дій a_1, a_2, a_3, a_6, a_8 , яка закінчується досягненням мети g_2 .

Але в цьому випадку агент може проаналізувати поточний стан сенсорів і вибрати перехід по ребрах з меншою вагою, якщо ребра з більшою вагою не приводять до досягнення цілі.

Плануючи послідовність дій, необхідно дозволити агенту використовувати як елементарні дії, так і деяку послідовність цих дій.

Плани дій, представлені попередньою моделлю, повинні бути адаптовані до нової ситуації.

Використання адаптивного методу полягає в підміні мети та початкових умов обраного прецеденту новою задачею і початковими умовами і забезпечені виконання плану після підміни. Деякі кроки початкового плану можуть виявитися непотрібними, так як змінилася мета, досягнення якої вони прагнули.

Висновки. При розробці моделі поведінки агента доцільно розширити стандартні рішення, використавши елементи біологічного підходу в штучному інтелекті. Для цього агентська діяльність здійснюється через взаємодію класифікаційної та виконавчої сторін. В якості класифікаційної частини використовуються сучасні типи штучних нейронних мереж, згорткових в нашому випадку. За виконавчу частину відповідають моделі на основі ланцюгів Маркова.

Штучна нейронна мережа отримує та класифікує інформацію від різних типів зовнішніх сенсорів і внутрішніх рецепторів агента, щоб ідентифікувати початкові умови дії агента, мету дії та визначити на їх основі послідовність дій, які виконує агент.

Штучна нейронна мережа складається з багатьох паралельних компонентів, які генерують класифікатори з різним ступенем деталізації та швидкості обробки для визначення початкових умов роботи.

Існує багато основних дій, які може виконувати агент. Основні дії можна виконувати окремо або комбінувати в послідовності в залежності від успішності поставленої мети.

ЛІТЕРАТУРА:

1. Робототехніка [Електронний ресурс] Режим доступу: <https://uk.wikipedia.org/wiki/Робототехніка>. Дата звернення: 08.05.2021
2. Stephen Marsland. Machine Learning: An Algorithmic Perspective / Stephen Marsland. – 2015. – 452 р.,
3. Deep Learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville. – 2016. – 800 p
4. Mariya Yao Applied Artificial Intelligence: A Handbook For Business Leaders Kindle Edition, Publisher: TOPBOTS, 2018, 246 p.
5. В. О. Бойчук, І. В. Муляр, Ю. О. Царьов. Біокомп'ютери та методики їх програмування. [Сучасна спеціальна техніка](#). - 2014. - № 2(37). - С. 54-60.

6. Рассел, С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. - М.: Вильямс, 2016. - 578 с.
7. Слэйгл, Дж. Искусственный интеллект / Дж. Слэйгл. - М.: Мир, 2016. - 320 с.
8. Акинин, М.В. Нейросетевые системы искусственного интеллекта в задачах обработки изображений / М.В. Акинин, М.Б. Никифоров, А.И. Таганов. - М.: ГЛТ, 2016. - 152 с.
9. Ayyadevara, V. K. Convolutional Neural Network. In: Pro Machine Learning Algorithms / V. K. Ayyadevara - Berkeley, CA : Apress, 2018. - С. 179–215.
10. A.M. Aibinu, A.J. Onumanyi, A.P. Adedigba, M. Ipinyomi, T.A. Folorunso, M.J.E. Salami, «Development of hybrid artificial intelligent based handover decision algorithm», Engineering Science and Technology an International Journal, 2017. – V.20 (2), pp. 381–390
11. Graves, A. Long Short-Term Memory. In: Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence., Springer, Berlin, Heidelberg, 2021 (Vol. 385)
12. Бойчук В. О. Метод формування послідовності дій систем реального часу / В. О. Бойчук, М. В. Бойчук, С. М. Жовнір // Наука й економіка : наук.-теорет. журн. / Хмельницьк. екон. ун-т. – Хмельницький, 2018. – Вип. 4 (48). – С. 133-137
13. Kravari K., Bassiliades N. A Survey of Agent Platforms. Journal of Artificial Societies and Social Simulation. 2015. Vol. 18, no. 1. P. 1–18.

REFERENCES:

1. Robototekhnika [Elektronnyi resurs] Rezhym dostupu: <https://uk.wikipedia.org/wiki/Robototekhnika>. Data zvernennia: 08.05.2021
2. Stephen Marsland. Machine Learning: An Algorithmic Perspective / Stephen Marsland. – 2015. – 452 p.,
3. Deep Learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville. – 2016. – 800 p
4. Mariya Yao Applied Artificial Intelligence: A Handbook For Business Leaders Kindle Edition, Publisher: TOPBOTS, 2018, 246 p.
5. Boichuk V. O., Muliar I. V., Tsarov Yu. O. (2014), “Biokompiutery ta metodyky yikh prohramuvannia” [Biocomputers and methods of their programming]. Modern special equipment. No 2(37). - pp. 54-60.
6. Rassel, S, Norvyh. P. (2016), “Yskusstvenniy intellekt: sovremenniy podkhod” [Artificial Intelligence: A Modern Approach] Vyliams, Moskva, 578 p.
7. Slэйгл, Dzh. (2016), “Yskusstvenniy intellekt” [Artificial Intelligence] Myr, Moskva, 320 p.
8. Akynyn, M.V., Nykyforov M.B., Tahanov A.Y. Akynyn, M.V., Nykyforov M.B., Tahanov A.Y. (2016), “Neirosetevii systemi iskusstvennoho intellekta v zadachakh obrabotky izobrazheniy” [Artificial intelligence neural network systems in image processing tasks] GLT, Moskva, 152 p
9. Ayyadevara, V. K. (2018). Convolutional Neural Network. In: Pro Machine Learning Algorithms. Apress, Berkeley, CA, 179–215.
10. A.M. Aibinu, A.J. Onumanyi, A.P. Adedigba, M. Ipinyomi, T.A. Folorunso, M.J.E. Salami, (2017) «Development of hybrid artificial intelligent based handover decision algorithm», Engineering Science and Technology an International Journal, V.20(2), - pp. 381–390
11. Graves, A. (2012). Long Short-Term Memory. In: Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence, (Vol. 385). Springer, Berlin, Heidelberg
12. Boichuk V. O., Boichuk, M. V., Zhovnir S. M. (2018), “Metod formuvannia poslidoynosti dii system realnoho chasu” [The method of forming a sequence of actions of real-time systems]. Nauka i ekonomika : nauk.-teoret. zhurn. / Khmelnyts. ekon. un-t. – Khmelnytskyi, No 4(48). - pp. 133-137.
13. Kravari K., Bassiliades N. (2015.) A Survey of Agent Platforms. Journal of Artificial Societies and Social Simulation. Vol. 18, no. 1. - pp. 1–18.

Ph.D. Ihor Muliar,

Ph.D. Viktoriia Orlenko, Illia Ostrovskiy

ADAPTIVE METHOD OF CONTROLLING AUTOMATED TECHNICAL SYSTEMS

The article proposes an approach to the process of managing technical objects based on the integration of sensor data processing.

Intelligent control methods have found application in various tasks, in particular in robotics. In systems that work in real time, the robot agent must rationally solve the tasks that are set before him with minimal resource costs. In real time, the use of high-precision multicriteria optimization

methods is difficult, so agents often solve the problem by approximate methods using problem-oriented heuristics. Analysis of materials on the problem of management shows that the creation of an effective management system requires the use of qualitatively new approaches to information processing, which should be based on finding features of action in the past, their adaptation based on hierarchical representation. The most promising direction in the creation of such systems is the use of modern neural networks for the classification of precedents and the formation of new actions based on the precedent approach.

When developing a model of agent behavior, it is proposed to expand the standard solutions, using elements of the biological approach in artificial intelligence. To do this, agency activities are carried out through the interaction of the classification and executive parties. As a classification part modern types of the artificial neural networks convoluted in our case are used. Models based on Markov chains are responsible for the executive part.

An artificial neural network receives and classifies information from different types of external sensors and internal receptors of the agent to identify the initial conditions of the agent, the purpose of the action and determine on their basis the sequence of actions performed by the agent.

The use of the adaptive method is to replace the purpose and initial conditions of the selected precedent with a new task and initial conditions and ensure the implementation of the plan after the replacement. Some of the steps in the original plan may not be necessary, as the goal they have sought has changed. The main actions can be performed separately or combined in sequence depending on the success of the goal.

Keywords: *control inteligente, multi-agent system, información sensorial*

Муляр Ігор Володимирович, кандидат технічних наук, доцент кафедри кібербезпеки Хмельницького національного університету (Хмельницький, Україна)

Номер ORCID <http://orcid.org/0000-0002-6659-605X>

Орленко Вікторія Сергіївна, кандидат технічних наук, доцент кафедри кібербезпеки Хмельницького національного університету (Хмельницький, Україна)

Номер ORCID <https://orcid.org/0000-0001-9601-1916>

Островський Ілля Ігорович, магістр кафедри кібербезпеки Хмельницького національного університету (Хмельницький, Україна)

Номер ORCID <https://orcid.org/0000-0003-1307-6082>

Муляр Ігорь Владимирович, к.т.н., доцент кафедри кібербезпеки Хмельницького національного університету (Хмельницький, Україна)

Номер ORCID <http://orcid.org/0000-0002-6659-605X>

Орленко Вікторія Сергеевна, к.т.н., доцент кафедри кібербезпеки Хмельницького національного університету (Хмельницький, Україна)

Номер ORCID <https://orcid.org/0000-0001-9601-1916>

Островський Ілля Ігорович, магістр кафедри кібербезпеки та комп'ютерних систем і мереж Хмельницького національного університету (Хмельницький, Україна)

Номер ORCID <https://orcid.org/0000-0003-1307-6082>

Muliar . Ihor, PhD, assistant professor of Cybersecurity of Computer Systems and Network of Khmelnytsky National University (Khmelnitsky, Ukraine)

Number ORCID <http://orcid.org/0000-0002-6659-605X>

Orlenko Viktoriia, PhD, assistant professor of Cybersecurity of Computer Systems and Networks of Khmelnytsky National University (Khmelnitsky, Ukraine)

Number ORCID <https://orcid.org/0000-0001-9601-1916>

Ostrovskiy Illia, PhD, assistant professor of Cybersecurity of Computer Systems and Networks of Khmelnytsky National University (Khmelnitsky, Ukraine)

Number ORCID <https://orcid.org/0000-0003-1307-6082>

к.т.н., доц. Гурман І.В. (ХмНУ)
к.т.н., доц. Тітова В.Ю. (ХмНУ)
Островський І.І. (ХмНУ)
Ряба Л.О. (ВІКНУ)

Методи захисту від кіберзагроз в бездротових мережах

Бездротові мережі стали невід'ємними у всіх сферах життя. Сучасні бездротові мережі за пливністю з'єднання не поступаються мережам, які використовують фізичне середовище. Навіть назвами мають перевагу — можуть підключити більшу кількість користувачів. Загалом кількість інтернет-користувачів у світі у 2020 році досягла 4,5 млрд. З них 3,8 млрд використовують смартфони для доступу до соцмереж [1]. Статистика підтверджує популярність бездротових мереж.

Сфери ІТ технологій та бізнесу мають велику потребу в бездротових з'єднаннях. Адже саме такі технології забезпечують гнучкість та мобільність співробітників, що в результаті розширює можливості компаній та працівників.

Але з широким поширенням бездротових мереж гострою стає проблема захисту переданої інформації. Забезпечити безпеку для бездротової мережі важче ніж для провідної. Адже очевидно, щоб дістатись до провідної мережі зловмисник повинен спочатку отримати фізичний доступ до кабельної системи. В бездротових мережах така умова не актуальна, адже дані передаються «по повітрю», тому для отримання доступу необхідний звичайний приймач.

IEEE 802.11 — це базовий стандарт, що визначає набір протоколів для передачі даних в бездротових мережах, який постійно доповнюється та оновлюється.

WPA2 — це найпростіший спосіб захисту від криптографічних атак. WPA / WPA2 здатен підвищити безпеку бездротової мережі, але додатковий захист відбувається за рахунок додаткової складності протоколу. WPA на високому рівні поділяють на дві категорії: атаки аутентифікації та атаки шифрування [2].

Реалізація протоколу безпеки WPA2, є кращим вирішенням проблем безпеки бездротової мережі. Але також важливими рекомендаціями є такі як приховування паролів, використання MAC фільтрів для підключених пристроїв та генерування складних паролів.

Отже, значне збільшення користувачів бездротових мереж призвело до розробки механізмів безпеки, які були подолані зловмисниками, тому необхідність отримання комплексного рішення для захисту мережі досі є доцільною. Для цього необхідно впровадити технологію WPA2 в усіх типах бездротових мереж. А також

використовувати методологію, яка включає в себе: створення паролю та плану забезпечення безпеки і захисту програмного забезпечення.

ДОДАТОК В
(обов'язковий)
Копія презентації

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Островський Ілля Ігорович

**МЕТОД КЕРУВАННЯ ТЕХНІЧНИМИ ОБ'ЄКТАМИ НА ОСНОВІ
ІНТЕГРАЦІЇ ОБРОБКИ СЕНСОРНИХ ДАНИХ**

**Науковий керівник
к.т.н., доцент Орленко В.С.**

кафедра кібербезпеки

Мета магістерської роботи полягає в розробці підходу до інтелектуального управління технічними об'єктами, який базується на інтеграції сенсорних даних та їх обробці сучасними нейронними мережами

Об'єкт дослідження: процеси інтелектуального управління технічними об'єктами

Предмет дослідження: моделі і методи процесу інтелектуального управління технічними об'єктами

Задачі досліджень у роботі формулюються наступним чином:

1. Аналіз характеристик застосування інтелектуальних технологій управління.
2. Формалізація завдання інтелектуального управління.
3. Уточнення моделі нейронної мережі для виявлення початкових умов та завдань управління.
4. Вдосконалення методу формування послідовності дій агенту на основі інтеграції сенсорних даних та прецедентного підходу.
5. Розробка алгоритмів та програмного забезпечення системи.
6. Експериментальне дослідження розробленого підходу.

Наукова новизна роботи полягає в наступному:

1. Запропоновано адаптивний підхід до задачі управління технічними об'єктами, в якому поєднується обробка штучними нейронними мережами внутрішніх рецепторів агенту та сенсорних сигналів з зовнішнього середовища,
2. Вдосконалено метод інтелектуального управління, що враховує класифікацію початкових умов та інтеграцію елементарних дій в послідовності з використанням ланцюгів Маркова.

Методика проведення досліджень: методи системного аналізу, теорії множин, апарат мереж Маркова, математичної статистики, теорії алгоритмів, теорії штучних нейронних мереж і методів імітаційного моделювання

Практична цінність. Запропоновані алгоритми формування послідовностей дій можуть бути використані при управлінні агентами, роботами, логістичними системами, соціальними явищами.

Публікації. По темі магістерської роботи опубліковано 1 стаття (збірник праць ВІКНУ), та 2 тези доповідей на всеукраїнських конференціях (Тези доповідей Міжнародної науково-практичної конференції "Військова освіта і наука: сьогодення та майбутнє" [Текст] / за заг. редакцією Ігоря Толока.– К. : ВІКНУ, 2021)

ФОРМАЛІЗАЦІЯ ЗАДАЧІ ІНТЕЛЕКТУАЛЬНОГО УПРАВЛІННЯ

У загальному випадку ціль може складатися з кількох станів, досягнення будь-якого з них означає досягнення мети. Також можливо, що жоден з цих станів є недоступним.

$$M = (Q, A, q_0, \Gamma_M),$$

де M – множина агентів;

Q - множина спостережуваних станів;

A – множина можливих дій;

q_0 – початковий стан;

I - початкові умови;

G - множина цільових станів;

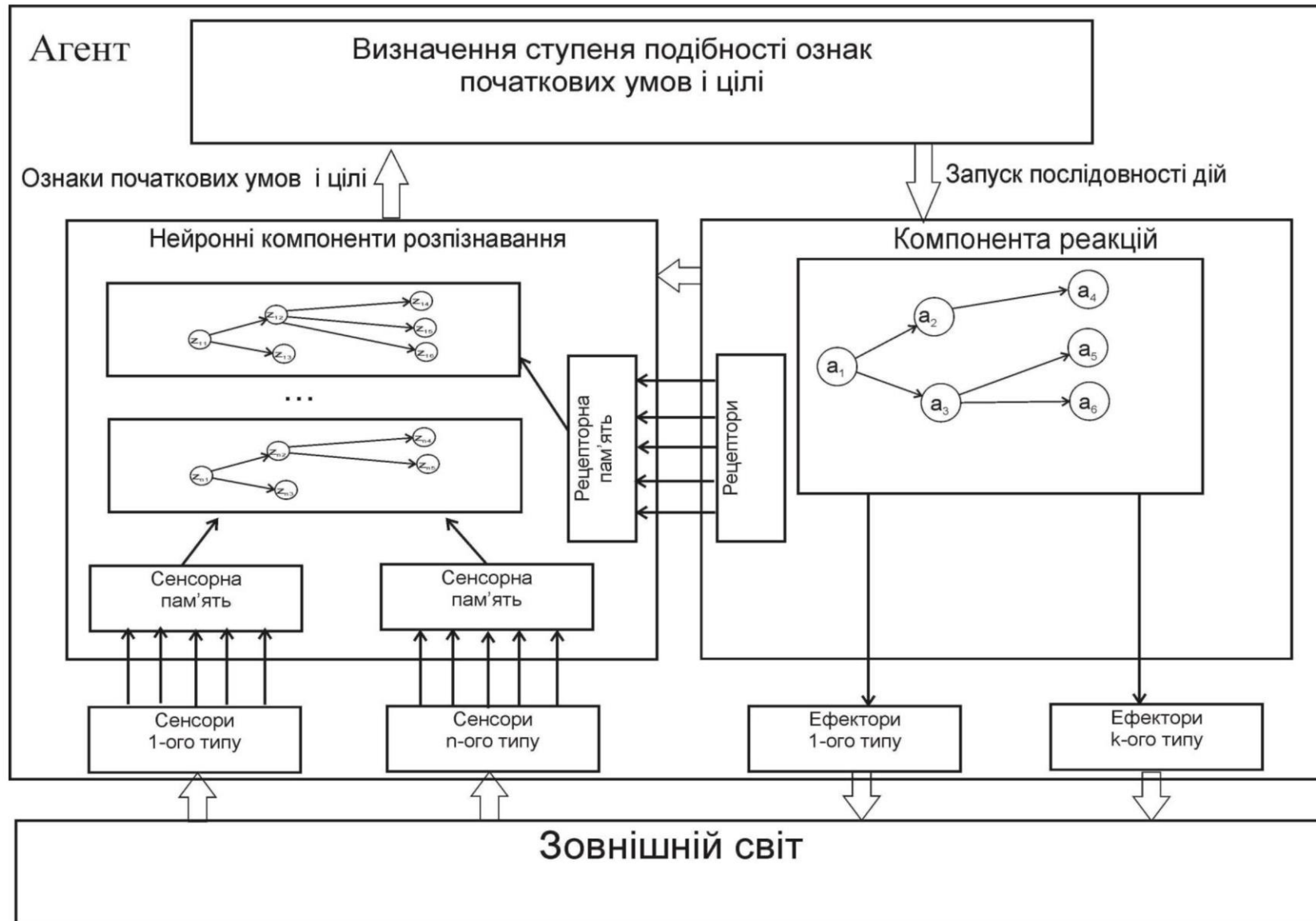
$\Gamma_M : Q \times A \Rightarrow Q$ функція переходу, яка визначає для кожного стану $q \in Q$ і дії $a \in A$ наступний стан $q' = \Gamma_M(q, a)$.

Потрібно знайти план, який є впорядкованою множиною дій $P = \{ a_1, \dots, a_n \}$, який є суперпозицією функцій переходу $\Gamma_M(\Gamma_M(\dots \Gamma_M(\Gamma_M(q_0, a_1) a_2) \dots, a_{n-1}), a_n)$ і належить G при $q_0 \in I$.

Існують початкові умови $q_0 \in I$ для ініціалізації процесу керування. Якщо брати до уваги процес управління роботом-агентом, то це може бути, наприклад, розряд батареї робота або технічний зір робота при небезпечному швидкому переміщенні великого об'єкту в напрямку робота.

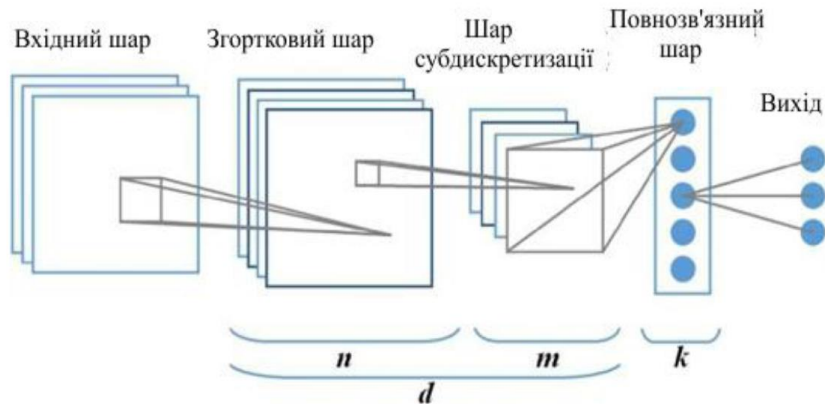
АДАПТИВНИЙ ПІДХІД ДО ОРГАНІЗАЦІЇ КЕРУВАННЯ СКЛАДНИМИ ТЕХНІЧНИМИ ОБ'ЄКТАМИ

(перший науковий результат)



УТОЧНЕНА МОДЕЛЬ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ

Параметри згорткової нейронної мережі



n - розмір квадратного попереднього шару мережі (для першого згорткового шару є розміром вхідного шару).

d - глибина вхідного шару мережі

C - кількість класів, приналежність до яких визначається класифікатором.

P - кількість рядків і стовпців, що містять нулі, що додаються до границями шару, що передує згортковому шару (параметр архітектури згорткового шару).

S - зсув між фільтрами при формуванні сигналів нейронів згорткового шару / зсув між вікнами при формуванні сигналів шару субдискретизації.

F - розмір квадратних фільтрів згорткового шару, причому F може приймати значення 3, 5 або 7.

F_{ilters} - глибина згорткового шару (кількість фільтрів).

F_{ilters}_p - глибина попереднього згорткового шару (глобальний параметр циклу формування шарів мережі).

AF_c - функція активації нейронів згорткового шару.

U - розмір квадратного вікна для шару субдискретизації.

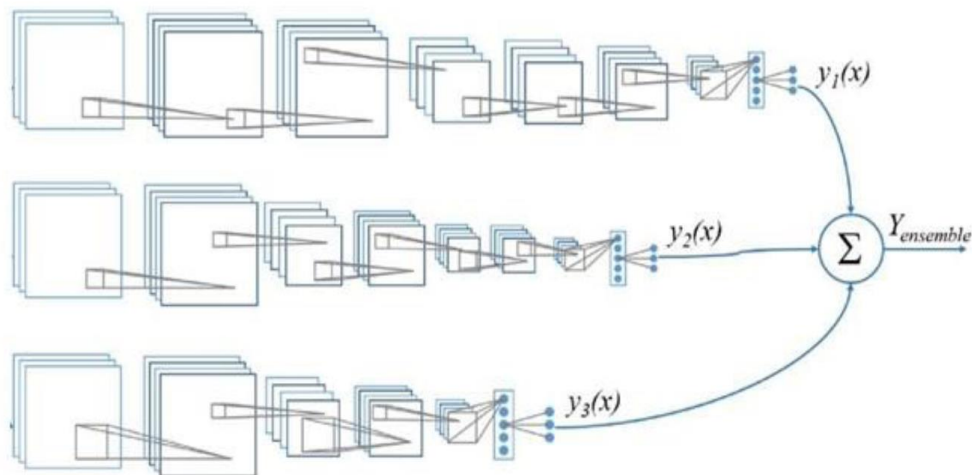
$Subf$ - тип функції шару субдискретизації (max - максимум, або avg розрахунок середнього).

k - кількість нейронів у повнозв'язаному шарі.

AF_{fc} - функція активації повнозв'язаного шару

d - глибина мережі - кількість рівнів мережі (глобальний параметр циклу формування шарів мережі).

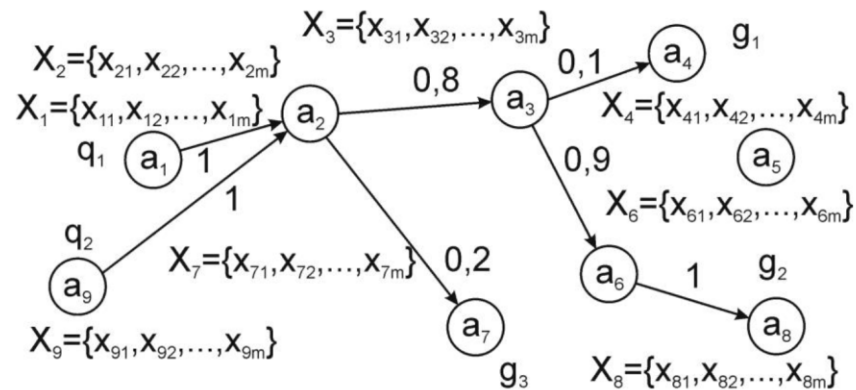
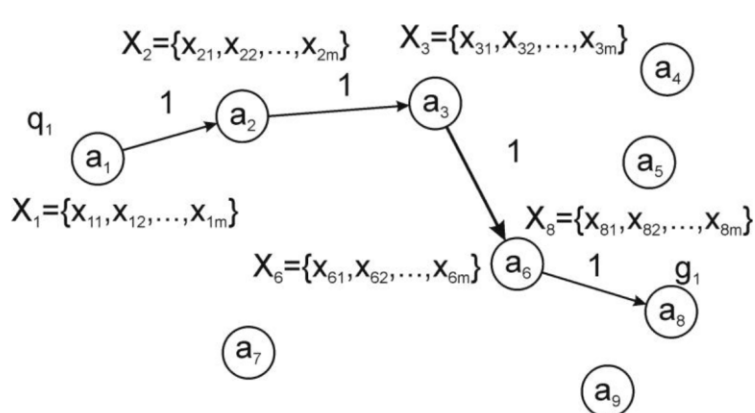
Ансамбль згорткових нейронних мереж



МЕТОД ФОРМУВАННЯ ПОСЛІДОВНОСТІ ДІЙ

(другий науковий результат)

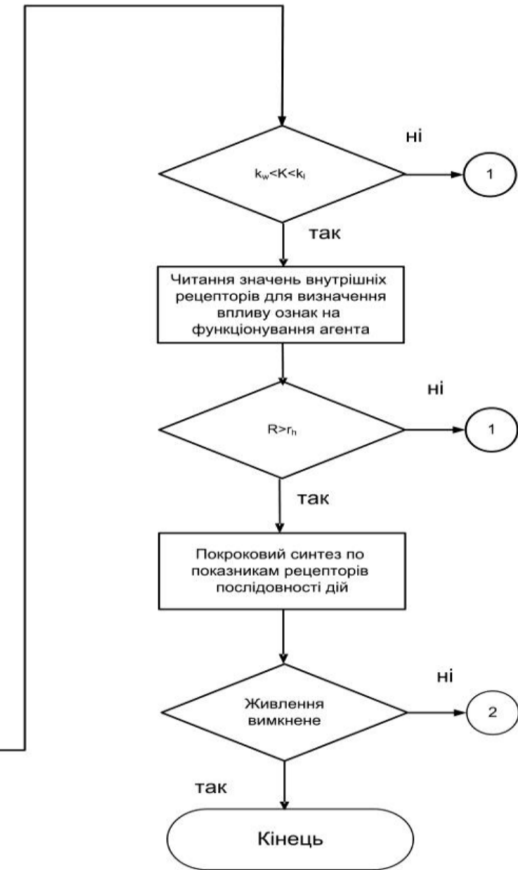
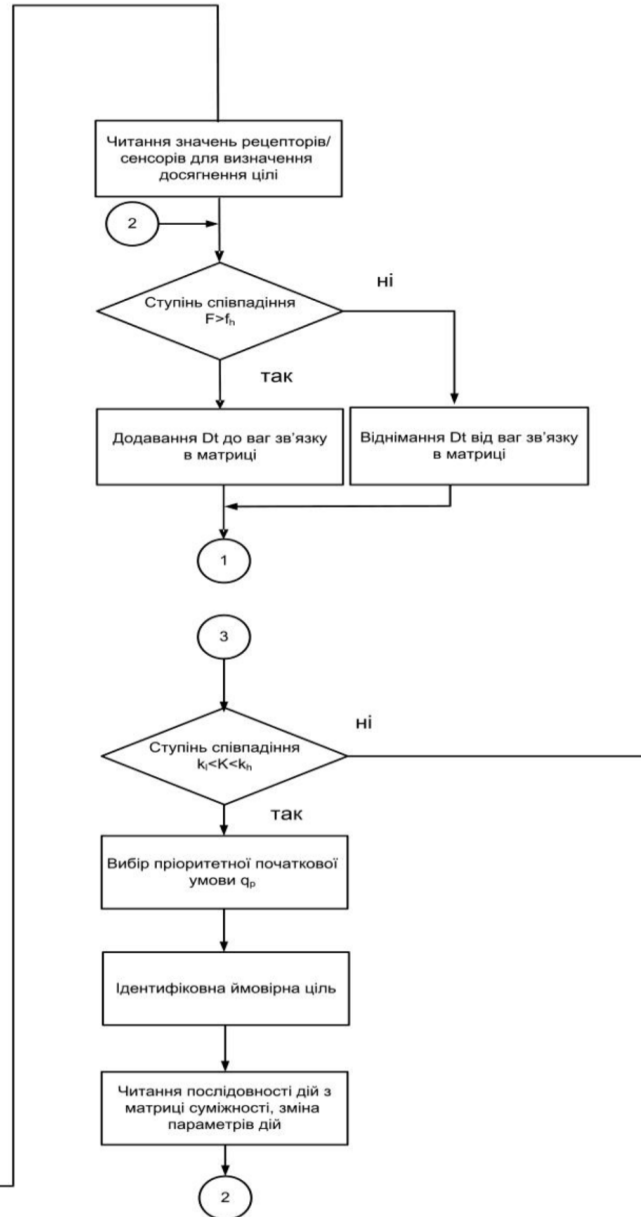
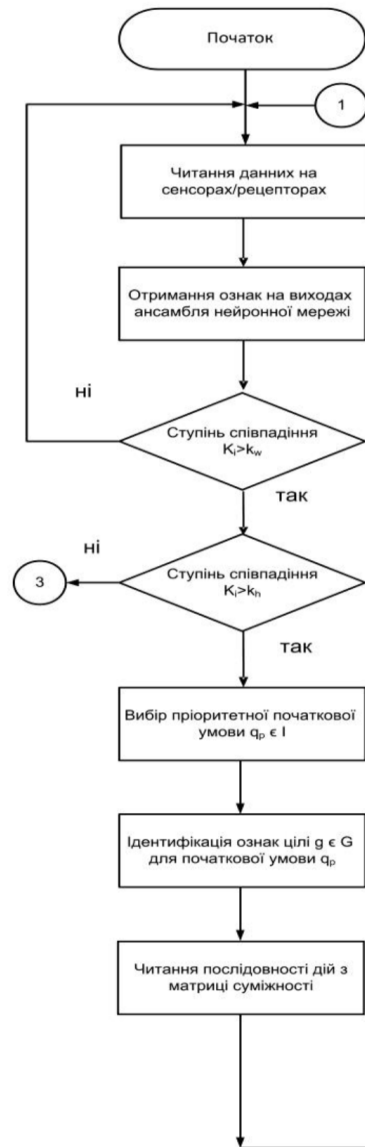
$q \in I$ - початкові стани, $g \in G$ - цілі дії, a_1, \dots, a_9 - елементарні дії



	a1	a2	a3	a4	a5	a6	a7	a8	a9
a1	0	1	0	0	0	0	0	0	0
a2	0	0	1	0	0	0	0	0	0
a3	0	0	0	0	0	1	0	0	0
a4	0	0	0	0	0	0	0	0	0
a5	0	0	0	0	0	0	0	0	0
a6	0	0	0	0	0	0	0	1	0
a7	0	0	0	0	0	0	0	0	0
a8	0	0	0	0	0	0	0	0	0
a9	0	0	0	0	0	0	0	0	0

	a1	a2	a3	a4	a5	a6	a7	a8	a9
a1	0	1	0	0	0	0	0	0	0
a2	0	0	0,8	0	0	0	0,2	0	0
a3	0	0	0,1	0	0,9	0	0	0	0
a4	0	0	0	0	0	0	0	0	0
a5	0	0	0	0	0	0	0	0	0
a6	0	0	0	0	0	0	0	1	0
a7	0	0	0	0	0	0	0	0	0
a8	0	0	0	0	0	0	0	0	0
a9	0	1	0	0	0	0	0	0	0

ЗАГАЛЬНИЙ АЛГОРИТМ ФУНКЦІОНУВАННЯ АГЕНТА



ВИСНОВКИ

У процесі реалізації роботи були вирішені наступні завдання:

1. Запропоновано адаптивний підхід до задачі управління технічними об'єктами, в якому поєднується обробка штучними нейронними мережами внутрішніх рецепторів агенту та сенсорних сигналів з зовнішнього середовища.

2. Вдосконалено модель згорткової нейронної мережі для визначення початкових умов керування та класифікації цілей керування.

3. З урахуванням особливостей складних об'єктів запропоновано модифікований метод найближчого сусіда. Ця модифікація полягає в тому, що застосовується спеціальний коефіцієнт K , що відповідає пороговому значенню ступеня схожості прецедентів і поточної ситуації. В результаті порівняння вибирається не тільки один найближчий сусід (прецедент), а й деякий набір близьких сусідів, ступінь схожості яких більше або дорівнює пороговому значенню ($S \geq K$).

4. На основі запропонованого модифікованого методу визначення найближчого сусіда було розроблено алгоритм отримання прецедентів із використанням різних метрик для визначення ступеня схожості прецедентів із поточною ситуацією. і з урахуванням коефіцієнтів, що характеризують важливість параметрів об'єкта.

5. Вдосконалено метод інтелектуального управління, що враховує класифікацію початкових умов та інтеграцію елементарних дій в послідовності з використанням ланцюгів Маркова.

6. Програмно реалізовано керуючу систему на основі розроблених методу і алгоритмів.

7. Результати експериментів показали, що модель, отримана з адаптивного класифікатора на основі набору згорткових нейронних мереж, забезпечує високу якість розпізнавання та перевищує чутливість класифікації звичайної згорткової нейронної мережі на 13%.

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «магістр»

Магістр Островський І.І.

Тема Метод керування технічними об'єктами на основі інтеграції обробки сенсорних даних

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «магістр»:

кількість листів креслень 9; кількість сторінок записки 81

1. Короткий зміст кваліфікаційної роботи та прийнятих рішень в рамках роботи розроблено адаптивний підхід до задачі управління технічними об'єктами, в якому поєднується обробка штучними нейронними мережами внутрішніх рецепторів агента та сенсорних сигналів з зовнішнього середовища. Запропонований підхід до інтелектуального управління, що враховує класифікацію початкових умов та інтеграцію елементарних дій в послідовності з використанням ланцюгів Маркова.

2. Висновок про відповідність кваліфікаційної роботи завданню Кваліфікаційна робота ОС «магістр» у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині дипломної роботи.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі висвітлюється актуальність теми роботи, дається аналіз досліджуваної проблеми і обґрунтовується застосований підхід до її вирішення, формулюються цілі і завдання дослідження, описується наукова новизна і практична значимість отриманих результатів. У першому розділі розглядаються питання особливостей застосування технологій інтелектуального управління. Наступні розділи присвячені розробці моделі і методу формування послідовності дій агента на основі інтеграції сенсорних даних та прецедентного підходу. Розглянуто питання застосування розробленого методу.

4. Позитивні сторони роботи Кваліфікаційна робота містить ряд інноваційних рішень, зокрема запропонований підхід забезпечує високу якість розпізнавання та перевищує чутливість класифікації звичайної згорткової нейронної мережі на 13%

5. Негативні сторони роботи Використання розробленого методу передбачає опціональне введення окремих початкових даних

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми кваліфікаційної роботи з дотриманням стандартів. В загальному графічне оформлення виконане якісно. Пояснювальна записка відповідає нормам для її оформлення.

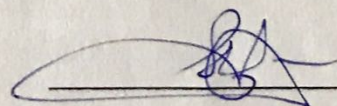
7. Відгук про роботу в цілому В загальному кваліфікаційної роботи заслуговує позитивної оцінки. Весь матеріал дипломної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики кваліфікаційної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої задачі.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку визначно (4.7/5)

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) Ф.Т.Н., проф.
Мисенко Сергій Михайлович, професор кафедри
Комп'ютерної інженерії та Інформаційних систем
Київського Національного університету

« 7 » грудня 2021.



(підпис)

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 8%**

ID: 98040 Название: Метод керування технічними об'єктами на основі інтеграції обробки сенсорних даних Добавлено в БД: 2021-12-05 Авторы: Островський І.І. Руководители: Орленко В.С, Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	104706	807	3900 (4%)	55 (7%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1009527925

Дата перевірки:
05.12.2021 16:03:57 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
05.12.2021 16:05:22 EET

ID користувача:
100008300

Назва документа: **Робота_Островський_плагіат**

Кількість сторінок: 86 Кількість слів: 14772 Кількість символів: 116921 Розмір файлу: 3.45 MB ID файлу: 1009538689

5% Схожість

Найбільша схожість: 1.36% з Інтернет-джерелом (<https://www.doccity.com/ru/inzheneriya-programnogo-zabezpechenny>).

3.23% Джерела з Інтернету 92 Сторінка 88

2.62% Джерела з Бібліотеки 80 Сторінка 89

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 26