МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю. М. Потебні

Кафедра електроніки, інформаційних систем
та програмного забезпечення
(повна назва кафедри)

Кваліфікаційна робота

другий (магістерський)
(рівень вищої освіти)

на тему : *Дослідження нейронної мережі та розробка пристрою і програмного застосунку для розпізнавання образів*

Виконав: студент *II* курсу, групи *8.1763*
спеціальності *176 Мікро- та наносистемна техніка*
(код і назва спеціальності)
освітньої програми *176 Мікроелектронні інформаційні системи*
(код і назва освітньої програми)
Ларченко Олег Ігорович
(ініціали та прізвище)

Керівник *доцент кафедри ЕІСПЗ, доцент, к.т.н., Небеснюк Оксана Юріївна*
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент *інженер-конструктор «Конструкторське бюро Колосова» Григор'єва І.К.*
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя

2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні

Кафедра електроніки, інформаційних систем та програмного забезпечення
Рівень вищої освіти другий (магістерський)
(перший (бакалаврський) рівень
Спеціальність 176 Мікро- та наносистемна техніка
(назва)
Освітня програма Мікроелектронні інформаційні системи
(шифр)

ЗАТВЕРДЖУЮ

Завідувач кафедри Т. В. Критська

«_30_» листопада 2023 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ (СТУДЕНТЦІ)

Ларченко Олегу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи (проєкту) *Дослідження нейронної мережі та розробка пристрою і програмного застосунку для розпізнавання образів*
керівник роботи *Небеснюк О.Ю., к.т.н.,доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ЗНУ від "_28_" лютого 2024 року № 361-с

2.Строк подання студентом роботи 29.11 2024 р.
3. Вихідні дані до роботи: *нейронна мережа, програмне забезпечення MATLAB, C ++, C#, система технічного зору OPEN CV .*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) *1 Нейронні мережі. Види та класифікація 2 Дослідження нейронних мереж та їх застосування для вирішення технічних задач 3 Техніко-економічне обґрунтування 4 Охорона праці та техногенна безпека*

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Л1 Основні задачі та класифікація нейронних мереж Л2 Структура системи розпізнавання образів Л3 Схема декодування символів в середовищі EWB Л4. Схема шифратора в середовищі EWB. Л5 Схема електрична принципова розробленого пристрою введення інформації. Л6 Функціональна схема перетворювача двійкового коду чисел схеми розпізнавання образів в коди семи

сегментного індикатора. Л7 Діаграма варіантів використання Л8 Символ кодування інформації і таблиця декодування символу.

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
I	Небеснюк О.Ю., доц. каф. ЕІСПЗ	23.10.23	20.12.23
II	Небеснюк О.Ю., доц. каф. ЕІСПЗ	25.12.23	20.10.24
III	Небеснюк О.Ю., доц. каф. ЕІСПЗ	23.10.24	30.10.24
IV	Небеснюк О.Ю., доц. каф. ЕІСПЗ	01.11.24	18.11.24

6. Консультанти розділів кваліфікаційної роботи бакалавра

7. Дата видачі завдання *23.10.2023 р.*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів кваліфікаційної роботи бакалавра	Примітка
1	*Аналіз матеріалу за темою кваліфікаційної роботи*	*02.011. -17.11.23*	
2	*Порівняльний аналіз нейронних мереж*	*20.11.23 -20.12.23*	
3	*Розробка та дослідення схеми апаратної реалізації нейронної мережі*	*25.12.23 -26.02.24*	
4	*Схемотехнічний аналіз та написання програмного забезпечення*	*01.03-30.09.24*	
5	*Оформлення другого розділу*	*01.10 -20.10.24*	
6	*Розділ «Техніко-економічне обґрунтування»*	*23.10-30.10.24*	
7	*Розділ «Охорона праці та техногенна безпека»*	*01.11. -18.11.23*	
8	*Оформлення пояснювальної записки, виконання креслень*	*20.11. -27.11.23*	

Студент _____ *Ларченко О.І.*
(підпис) (прізвище та ініціали)

Керівник роботи (проєкту) _____ *Небеснюк О.Ю.*
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер _____ *Верьовкін Л. Л.*
(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота містить 100 сторінок, 42 рисунки, 16 таблиць, 2 додатки, 21 джерело літератури.

Об'єкт проектування – штучні нейронні мережі.

Ціль роботи – дослідження та реалізація нейронної мережі для вирішення задач з розпізнавання образів.

Задачі роботи - розробка та дослідження схеми апаратної реалізації нейрона; розробка пристрою для розпізнавання образів; розробка програмного застосунку для розпізнавання дорожніх знаків.

Методика досліджень – моделювання пристрою розпізнавання образів в програмному середовищі Electronic Workbench, розробка програмного застосунку на мові C++ з використанням бібліотеки компютерного зору Open CV.

Короткий виклад результатів досліджень –розроблено та розраховано пристрій для розпізнавання образів, програмний застосунок TrafficSignRecognition пройшов усі тести, тож можна вважати, що його реалізація є успішною.

Прогнозні пропозиції – запропоновані технічні рішення можуть бути використані для оптимізації та автоматизації процесів в різних галузях науки і техніки.

НЕЙРОННА МЕРЕЖА, ПЕРСЕПТРОН, ПЛІС, РОЗПІЗНАВАННЯ ОБРАЗІВ, КОДУВАННЯ ІНФОРМАЦІЇ, КОМП'ЮТЕРНИЙ ЗІР

Робота виконана на кафедрі електроніки, інформаційних систем та програмного забезпечення в період з 02 вересня по 29 листопада 2024 р.

ЗМІСТ

ВСТУП 6

1 НЕЙРОННІ МЕРЕЖІ 7

1.1 Біологічний прототип і штучний нейрон 7

1.2 Основні положення 10

1.3 Найпростіші нейронні мережі 13

1.4 Вибір структури нейронної мережі 14

1.5 Класифікація нейронних мереж 15

1.6 Задачі для вирішення нейронних мереж 18

2 ДОСЛІДЖЕННЯ НЕЙРОННИХ МЕРЕЖ ТА ЗАСТОСУВАННЯ ЇХ ДЛЯ ВИРІШЕННЯ ТЕХНІЧНИХ ЗАДАЧ 22

2.1 Розпізнавання образів 22

2.2 Методи розпізнавання образів 23

2.3 Загальна характеристика задач розпізнавання образів і їх типи 24

2.4 Розробка математичної моделі нейронної мережі 25

2.5 Види нейронних мереж 32

2.6 Типи нейронних мереж 32

2.7 Апаратна реалізація нейронної мережі 43

2.8 Програмне забезпечення для моделювання нейронних мереж 45

2.9 Розробка системи розпізнавання дорожніх знаків 47

2.9.1 Архітектура запропонованої системи 48

2.9.2 Вимоги до інтерфейсу 50

2.9.3 Проект програмної системи 52

2.9.4 Засоби реалізації та тестування 53

2.10 Розрахунок пристрою розпізнавання образів 53
2.10.1 Схема контролю парності 59
2.10.2 Вибір схеми генератора імпульсів 62
2.10.3 Буферний регістр 62
2.10.4 Пристрій розпізнавання образів 64
2.10.5 Пристрій відображення інформації 65
3 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ 72
3.1 Порівняльний аналіз штучних нейронних мереж за критеріями 72
4 ОХОРОНА ПРАЦІ ТА ТЕХНОГЕННА БЕЗПЕКА 78
4.1 Характеристика потенційно небезпечних та шкідливих виробничих 78
факторів 78
4.2. Заходи з поліпшення умов праці, виробнича санітарія 79
4.3. Електробезпека 82
4.4. Пожежна та техногенна безпека 83
4.5 Дослідження засобів захисту від ураження струмом при дотику до фазового дроту 86
ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ 89
ПЕРЕЛІК ПОСИЛАНЬ 90
ДОДАТОК А 92
ДОДАТОК В 98

ВСТУП

Дослідження в області нейронних мереж почалися в 40-і роки XX століття. Перше систематичне вивчення штучних нейронних мереж було зроблено Маккалокком і Піттсом в 1943 році. Пізніше вони досліджували мережеві парадигми для розпізнавання зображень, що піддаються зсувам і поворотам.

Основні завдання, які ставляться перед нейронними мережами, відносяться до завдань розпізнавання образів. Вони полягають в тому, щоб класифікувати вхідний образ, тобто віднести його до якого-небудь відомого мережі класу. Спочатку мережі даються еталонні образи - такі образи, належність яких до певного класу відома. Потім на вхід мережі подається деякий невідомий образ, і мережа намагається за певним алгоритмом співвіднести його з будь-яким еталонним образом. Можна сказати, що нейронні мережі проводять кластеризацию образів. Необхідність інноваційного розвитку та стимулювання науково-технічного прогресу вимагає від вітчизняної промисловості та усіх секторів економіки пришвидшення технологічної діджиталізації. Використання штучного інтелекту за таких умов дозволить у зручний спосіб та з мінімальними ресурсовитратами вирішити проблему розпізнавання об’єктів, здійснити аналіз діяльності складних господарських систем та вчасно прийняти вірні управлінські рішення на основі одержаних результатів нейромоделювання.

Все вищенаведене свідчить про те, що проблеми дослідження можливостей нейронних мереж і їх розвитку є актуальною на даний момент часу.

1 НЕЙРОННІ МЕРЕЖІ

1.1 Біологічний прототип і штучний нейрон

Для дослідження нервової системи (НС) необхідно мати математичну модель нейрона та нервової системи. Центральна нервова система має клітинну будову. Одиниця - нервова клітина, нейрон. Нейрон має такі основні властивості:

1. Бере участь в обміні речовин і розсіює енергію. Змінює внутрішній стан з плином часу, реагує на вхідні сигнали і формує вихідні, тому є активною динамічною системою.

2. Має безліч синапсів – контактів для передачі інформації.

3. Нейрон взаємодіє шляхом обміну електрохімічними сигналами двох видів: електротонічними (зі загасанням) і нервовими імпульсами (спайками), що поширюються без загасання, стихання.

Існують два підходи до моделювання нерва та нервової системи.

Інформаційний підхід: байдуже, які механізми лежать в основі роботи штучних нейронів, важливо лише, щоб під час розв'язання завдань інформаційні процеси в НС були подібні біологічним.

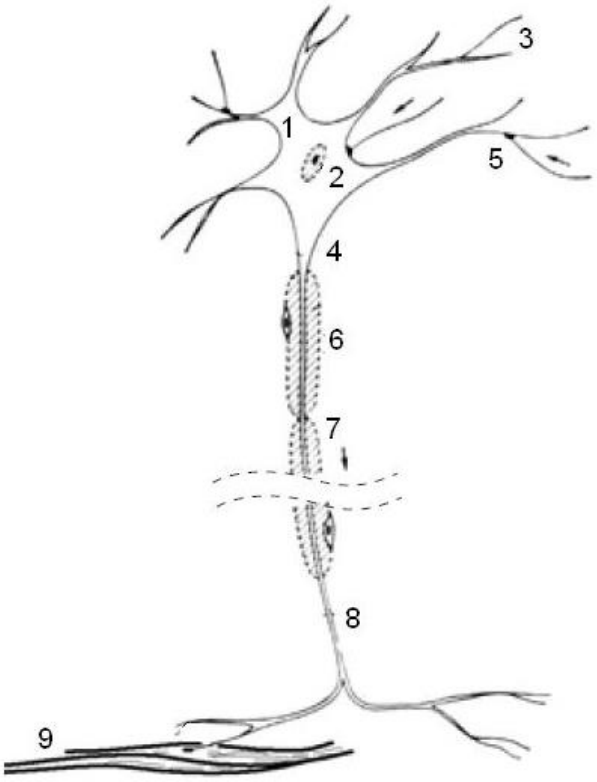
Біологічний: при моделюванні важливо повна біоподоба, і необхідно детально вивчати роботу біологічного нейрона.

Великі роботи в дослідженнях моделювання НС належать Ендрю Хакслі, Алану Ходжкіну, Бернарду Катцу, Джону Екклзу, Стівену Куфлеру.

У центральній нервовій системі людини налічується від 100 до 1000 типів нервових клітин, в залежності від обраного ступеня деталізації. Вони відрізняються картиною дендритів, наявністю і довжиною аксона і розподілом

синапсів біля клітини. Клітини сильно пов'язані між собою. У нейрона може бути більше 1000 синапсів. Близькі за функціями клітини утворюють скупчення, кулясті або паралельні шаруваті. У мозку виділені сотні скупчень. Кора головного мозку – теж скупчення.

Товщина кори – 2мм, площа – близько квадратного метра (0.93 $м^2$). Біологічний нейрон зображено на рисунку 1.1.



1 – тіло клітини, 2 – ядро, 3 – дендрити, 4 – аксонний горбок, 5 – синапс, 6 – шванновські клітини, 7 – перехвати Ранв’є, 8 – аксон, 9 – м’язові волокна

Рисунок 1.1 – Біологічний нейрон

Біологічний нейрон містить такі структурні одиниці:

Тіло клітини (1) – сома: містить ядро (2), мітохондрії (забезпечують клітину енергією), інші органели, що підтримують життєдіяльність клітини.

Дендрити (3) – вхідні волокна, збирають інформацію від інших нейронів. Активність у дендритах змінюється плавно. Довжина їх зазвичай не більше 1мм.

Мембрана – підтримує постійний склад цитоплазми всередині клітини, забезпечує проведення нервових імпульсів.

Цитоплазма – внутрішнє середовище клітини. Відрізняється концентрацією іонів K, Na, Ca та інших речовин у порівнянні з позаклітинним середовищем.

Аксон (8), один або жодного в кожній клітини, – довге, іноді більше метра, вихідне нервове волокно клітини. Імпульс генерується в аксонному горбку (4).

Аксон забезпечує проведення імпульсу і передачу впливу на інші нейрони або м'язові волокна (9). Ближче до кінця аксон часто галузиться.

Синапс (5) – місце контакту нервових волокон – передає збудження від клітини до клітини. Передача через синапс майже завжди односпрямована. Розрізняють предсинаптичні та постсинаптичні клітини – по напрямку передачі імпульсу.

Шванівські клітини (6) – пецифічні клітини, майже повністю складаються з мієліну, органічної ізолюючої речовини. Щільно "обмотують" нервове волокно 250 шарами мієліну. Неізольовані місця нервового волокна між шванівськими клітинами називаються перехопленнями Ранв’є (7). За рахунок мієлінової ізоляції швидкість розповсюдження нервових імпульсів зростає в 5-10 разів і зменшуються витрати енергії на проведення імпульсів. Мієлінізовані волокна зустрічаються тільки у вищих тварин.

Розвиток штучних нейронних мереж надихається біологією. Тобто, розглядаючи мережеві конфігурації і алгоритми, дослідники застосовують терміни, запозичені з принципів організації мозкової діяльності. Але на цьому аналогія закінчується. Наші знання про роботу мозку настільки обмежені, що мало б знайшлося точно доведених закономірностей для тих, хто побажав би керуватися ними. Тому розробникам мереж доводиться виходити за межі сучасних біологічних знань в пошуках структур, здатних виконувати корисні функції. У багатьох випадках це призводить до необхідності відмови від біологічної правдоподібності, мозок стає просто метафорою, і створюються

мережі, неможливі в живої матерії або можуть бути використані неправдоподібно великих припущень про анатомію і функціонуванні мозку.

Незважаючи на те, що зв'язок з біологією слабкий і часто несуттєвий, штучні нейронні мережі продовжують порівнювати з мозком. Їх функціонування часто має зовнішню схожість з людським пізнанням, тому важко уникнути цієї аналогії. На жаль, такі порівняння створюють невиправдані очікування, що неминуче ведуть до розчарування.

1.2 Основні положення

В останні десятиліття в світі бурхливо розвивається нова прикладна область математики, що спеціалізується на штучних нейронних мережах (НМ). Актуальність досліджень в цьому напрямку підтверджується масою різних застосувань НМ. Це автоматизація процесів розпізнавання образів, адаптивне управління, апроксимація функціоналів, прогнозування, створення експертних систем, організація асоціативної пам'яті і багато інших додатків. За допомогою НМ можна, наприклад, передбачати показники біржового ринку, виконувати розпізнавання оптичних або звукових сигналів, створювати саморозвинені системи, здатні керувати автомашиною при парковці або синтезувати мову по тексту. [1].

Широке коло завдань, що вирішується НМ, не дозволяє в даний час створювати універсальні, потужні мережі, змушуючи розробляти спеціалізовані НМ, що функціонують за різними алгоритмами.

По перше, основу кожної НМ складають відносно прості, в більшості випадків - однотипні, елементи (комірки), що імітують роботу нейронів мозку. Далі під нейроном буде матися на увазі штучний нейрон, тобто комірка НМ. Кожен нейрон характеризується своїм поточним станом за аналогією з нервовими клітинами головного мозку, які можуть бути порушені або

загальмовані. Він володіє групою синапсів - односпрямованих вхідних зв'язків, з'єднаних з виходами інших нейронів, а також має аксон - вихідний зв'язок даного нейрона, з якого сигнал (збудження або гальмування) надходить до синапсу наступних нейронів. Загальний вигляд нейрона наведено на рис. 1.2 Кожен синапс характеризується величиною синаптичного зв'язку або її вагою (w_i), який по фізичному змісту еквівалентний електричної провідності.

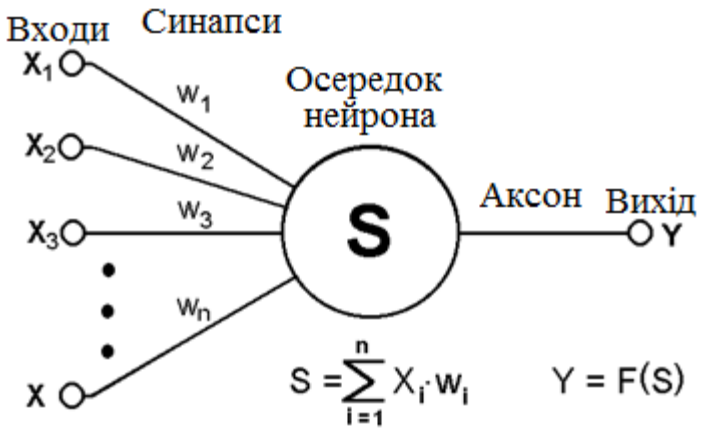


Рисунок 1.2 - Загальний вигляд нейрону

Поточний стан нейрона визначається, як зважена сума його входів:

$$s = \sum_{i=1}^{n} x_i \cdot w_i \qquad (1.1)$$

Вихід нейрона є функція його стану:

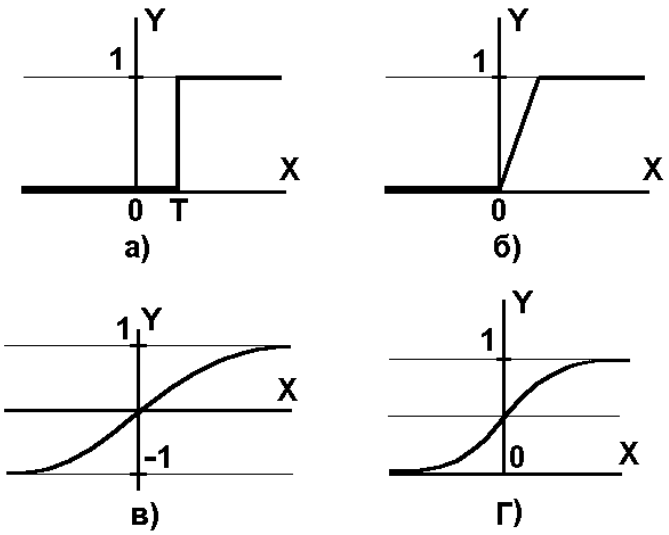
$$y = f(s) \qquad (1.2)$$

Нелінійна функція f називається активаційною, може мати різний вигляд, як показано на рисунку 1.3. Однією з найбільш поширених є нелінійна функція з насиченням, так звана логістична функція або сигмоід (тобто функція S-подібного виду) [2]:

$$f(x) = \frac{1}{1+e^{-\alpha x}} \qquad (1.3)$$

При зменшенні α сигмоід стає більш пологим, в межі при α = 0 вироджуючись в горизонтальну лінію на рівні 0.5, при збільшенні α сигмоид наближається за зовнішнім виглядом до функції одиничного стрибка з порогом T в точці x = 0. З виразу для сигмоіда очевидно, що вихідне значення нейрона лежить в діапазоні [0,1]. Одне з цінних властивостей сігмоідної функції - простий вислів для її похідної, застосування якого буде розглянуто в подальшому.

$$f'(x) = \alpha \cdot f(x) \cdot (1 - f(x)) \qquad (1.4)$$



а) функція одиничного стрибка; б) лінійний поріг (гістерезис); в) сигмоід - гіперболічний тангенс; г) сигмоід - формула (3)

Рисунок 1.3- Функція активації

Слід зазначити, що сигмоідна функція диференційована на всій осі абсцис, що використовується в деяких алгоритмах навчання. Крім того вона має властивість підсилювати слабкі сигнали краще, ніж великі, і запобігає насиченню від великих сигналів, так як вони відповідають областям аргументів, де сигмоід має пологий нахил.

Повертаючись до загальних рис, властивим всім НМ, відзначимо принцип паралельної обробки сигналів, який досягається шляхом об'єднання великого числа нейронів в так звані шари і з'єднання певним чином нейронами різних шарів. В деяких конфігураціях нейронами одного шару між собою, причому обробка взаємодії всіх нейронів ведеться пошарово.

1.3 Найпростіші нейронні мережі

Як приклад найпростішої НМ розглянемо трьохнейронний перцептрон (рис.1.4), тобто таку мережу, нейрони якої мають активаційну функцію у вигляді одиничного стрибка. На n входів надходять деякі сигнали, що проходять по синапсах на 3 нейрона, що утворюють єдиний шар цієї НМ та видають три вихідних сигнали:

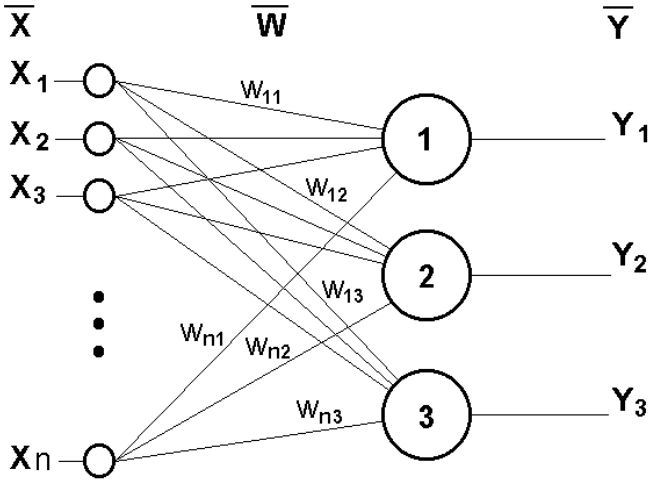


Рисунок 1.4 - Одношаровий перцептрон

$$y_j = f\left[\sum_{i=1}^{n} x_i \cdot w_{ij}\right], \quad j=1...3 \qquad (1.5)$$

Очевидно, що всі вагові коефіцієнти синапсів одного шару нейронів можна звести в матрицю W, в якій кожен елемент W_{ij} задає величину і-ї

синаптичного зв'язку j-ого нейрона. Таким чином, процес, що відбувається в НМ, може бути записаний в матричної формі:

$$Y = F(XW) \qquad (1.6)$$

де X і Y - відповідно вхідний і вихідний сигнальні вектори, F (W) - активаційна функція, застосовувана поелементно до компонентів вектора W.

Теоретично число шарів і число нейронів в кожному шарі може бути довільним, однак фактично воно обмежене ресурсами комп'ютера або спеціалізованої мікросхеми, на яких зазвичай реалізується НМ. Чим складніше НМ, тим масштабніші завдання підвладні їй.

1.4 Вибір структури нейронної мережі

Вибір структури НМ здійснюється відповідно до особливостей і складності завдання. Для вирішення деяких окремих типів завдань вже існують оптимальні, на сьогоднішній день, конфігурації, описані, наприклад, в [2], [3], [4] та інших виданнях. Якщо ж завдання не може бути зведене до жодного з відомих типів, розробнику доводиться вирішувати складну проблему синтезу нової конфігурації. При цьому він керується кількома основними принципами: можливості мережі зростають зі збільшенням числа осередків мережі, щільності зв'язків між ними і числом виділених шарів; введення зворотних зв'язків поряд зі збільшенням можливостей мережі піднімає питання про динамічну стійкість мережі; складність алгоритмів функціонування мережі (в тому числі, наприклад, введення декількох типів синапсів - збудливих, припиняючих та ін.) також сприяє посиленню НМ. Питання про необхідні і достатні властивості мережі для вирішення того чи іншого роду завдань є цілий напрям нейрокомп’ютерної науки. Так як проблема синтезу НМ сильно залежить від розв'язуваної задачі, дати загальні докладні рекомендації важко. У

більшості випадків оптимальний варіант виходить на основі інтуїтивного підбору.

Очевидно, що процес функціонування НМ, тобто сутність процесів, які вона здатна виконувати, залежить від величин синаптичних зв'язків, тому, задавшись певною структурою НМ, що відповідає будь-якій задачі, розробник мережі повинен знайти оптимальні значення всіх змінних вагових коефіцієнтів (деякі синаптичні зв'язки можуть бути постійними).

Цей етап називається навчанням НМ, і від того, наскільки якісно він буде виконаний, залежить здатність мережі вирішувати поставлені перед нею проблеми під час експлуатації. На етапі навчання крім параметра якості підбору вагових коефіцієнтів важливу роль відіграє час навчання. Як правило, ці два параметри пов'язані зворотною залежністю і їх доводиться вибирати на основі компромісу.

Навчання НМ може вестися з вчителем або без нього. У першому випадку мережі пред'являються значення як вхідних, так і бажаних вихідних сигналів, і вона по деякому внутрішньому алгоритмі підлаштовує ваги своїх синаптичних зв'язків. У другому випадку виходи НМ формуються самостійно, а ваги змінюються по алгоритму, що враховує тільки вхідні і похідні від них сигнали.

Існує безліч різних алгоритмів навчання, які діляться на два великі класи: детерміновані і стохастичні. У першому з них підстроювання вагових коефіцієнтів являє собою жорстку послідовність дій, у другому - вона виробляється на основі дій, що підкоряються деякого випадкового процесу.

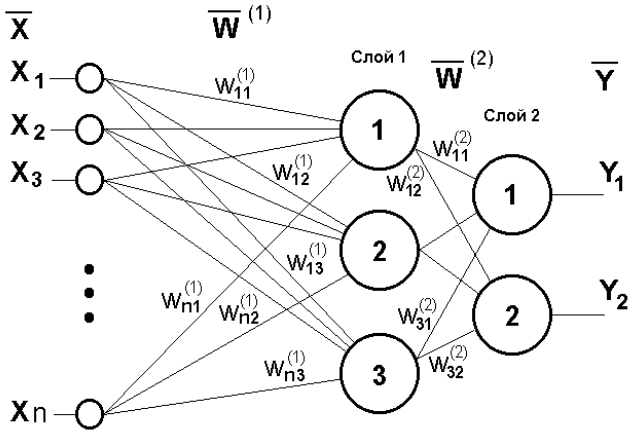
1.5 Класифікація нейронних мереж

Розвиваючи далі питання про можливу класифікації НМ, важливо відзначити існування бінарних і аналогових мереж. Перші з них оперують з двійковими сигналами, і вихід кожного нейрона може приймати тільки два

значення: логічний нуль ("загальмований" стан) і логічна одиниця ("збуджений" стан). До цього класу мереж відноситься і розглянутий вище перцептрон, так як виходи його нейронів, що формуються функцією одиничного стрибка, рівні або 0, або 1. В аналогових мережах вихідні значення нейронів здатні приймати безперервні значення, що могло б мати місце після заміни активаційної функції нейронів перцептрону на сигмоид.

Ще одна класифікація ділить НМ на синхронні і асинхронні [3]. У першому випадку в кожен момент часу свій стан змінює лише один нейрон. У другому - стан змінюється відразу у цілої групи нейронів, як правило, у всього шару. Алгоритмічно хід часу в НМ задається ітераційним виконанням однотипних дій над нейронами. Далі будуть розглядатися тільки синхронні НМ.

Мережі також можна класифікувати за кількістю шарів. На рисунку 1.5 представлений двошаровий перцептрон, отриманий з перцептрону з рисунка 1.4 шляхом додавання другого шару, що складається з двох нейронів.



Рисункок 1.5 - Двошаровий перцептрон

Тут доречно відзначити важливу роль нелінійності активаційної функції, так як, якщо б вона не володіла даними властивістю або не входила в алгоритм роботи кожного нейрона, результат функціонування будь-якої р-шарової НМ з ваговими матрицями W (i), i = 1,2,. ..p для кожного шару і зводився б до перемножування вхідного вектора сигналів X на матрицю

$$W^{(\Sigma)}=W^{(1)}\cdot W^{(2)}\cdot\ldots\cdot W^{(p)} \qquad (1.7)$$

тобто фактично така р-шарова НМ еквівалентна одношарової НМ з ваговою матрицею єдиного шару $W^{(\Sigma)}$:

$$Y=XW^{(\Sigma)} \tag{1.8}$$

Продовжуючи розмову про нелінійність, можна відзначити, що вона іноді вводиться і в синаптичні зв'язки. Більшість відомих на сьогоднішній день НМ використовують для знаходження зваженої суми входів нейрона формулу (1.1), але для деяких програм НМ корисно ввести інший запис, наприклад:

$$s = \sum_{i=1}^{n} x_i^2 \cdot w_i \tag{1.9}$$

Або

$$s = \sum_{i=1}^{n} x_i \cdot x_{((i+1) \bmod n)} \cdot w_i \tag{1.10}$$

Питання в тому, щоб розробник НМ чітко розумів, для чого він це робить, якими цінними властивостями він тим самим додатково наділяє нейрон, і яких позбавляє. Введення такого роду нелінійності, збільшує обчислювальну потужність мережі, тобто дозволяє з меншого числа нейронів з "нелінійними" синапсами сконструювати НМ, що виконує роботу звичайної НМ з великим числом стандартних нейронів і більш складної конфігурації [4].

Різноманіття існуючих структур НМ дозволяє відшукати й інші критерії для їх класифікації.

З рисунку функції одиничного стрибка видно, що граничне значення Т, в загальному випадку, може приймати довільне значення. Більш того, воно повинно приймати якесь довільне, невідоме заздалегідь значення, яке підбирається на стадії навчання разом з ваговими коефіцієнтами. Те ж саме відноситься і до центральної точки сигмоідної залежності, яка може зрушуватися вправо або вліво по осі X, а також і до всіх інших активаційних

функцій. Це, однак, не відображено у формулі (1.1), яка повинна була б виглядати так:

$$s = \sum_{i=1}^{n} x_i \cdot w_i - T \qquad (1.11)$$

Справа в тому, що таке зміщення зазвичай вводиться шляхом додавання до шару нейронів ще одного входу, що порушує додатковий синапс кожного з нейронів, значення якого завжди дорівнює 1. Присвоїмо цьому входу номер 0. Тоді:

$$s = \sum_{i=0}^{n} x_i \cdot w_i \qquad (1.12)$$

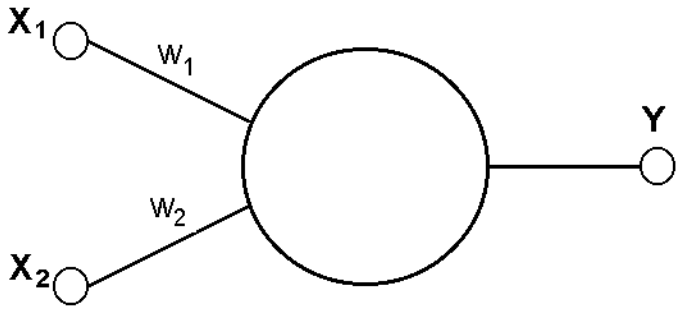
де $w_0 = -T$, $x_0 = 1$.

Очевидно, що відмінність формул (1.1) і (1.12) складається лише в способі нумерації входів.

З усіх активаційних функцій, зображених на рисунку 1.3, одна виділяється особливо. Це гіперболічний тангенс, залежність якого симетрична щодо осі X і лежить в діапазоні [-1,1]. Забігаючи наперед, скажемо, що вибір області можливих значень виходів нейронів багато в чому залежить від конкретного типу НМ і є питанням реалізації, так як маніпуляції з нею впливають на різні показники ефективності мережі, часто не змінюючи загальну логіку її роботи.

1.6 Задачі для вирішення нейронних мереж

Робота всіх мереж зводиться до класифікації (узагальнення) вхідних сигналів, що належать n-мірному гіперпростору, по деякому числу класів. З математичної точки зору це відбувається шляхом розбиття гіперпростору гіперплоскостями (запис для випадку одношарового перцептрона)



Рисункок 1.6 - Однонейронний перцептрон

$$\sum_{i=1}^{n} x_i \cdot w_{ik} = T_k \ , \ \text{k=1...m} \qquad (1.13)$$

Кожна отримана область є областю визначення окремого класу. Число таких класів для однієї НМ перцептроном типу не перевищує 2m, де m - число виходів мережі. Однак не всі з них можуть бути нероздільні даної НМ.

Наприклад, одношаровий перцептрон, що складається з одного нейрона з двома входами, представлений на рисунку 1.4, не здатний розділити площину (двовимірне гіперпространоство) на дві півплощини так, щоб здійснити класифікацію вхідних сигналів по класах А і В (Табл. 1.1).

Таблиця 1.1- Класифікація вхідних сигналів по класах А і В

x_2 \ x_1		
0		
1		

Якщо придивитися до таблиці 1.1, можна помітити, що дане розбиття на класи реалізує логічну функцію виключає АБО для вхідних сигналів.

Неможливість реалізації одношаровим перцептроном цієї функції отримала назву проблеми виключає АБО.

Рівняння мережі для цього випадку:

$$x_1 \cdot w_1 + x_2 \cdot w_2 = T \qquad (1.14)$$

Це рівняння прямої (одновимірної гиперплощини), яка ні за яких умов не може розділити площину так, щоб точки з множини вхідних сигналів, що належать різним класам, опинилися по різні боки від прямої (рис. 1.6).

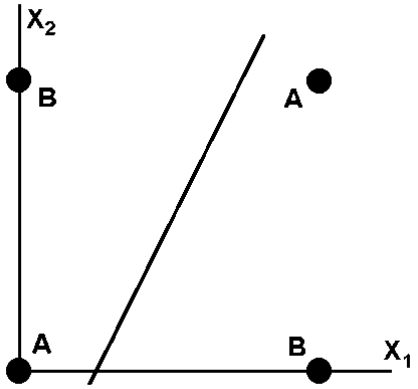


Рисунок 1.7 - Візуальне уявлення роботи НМ з рисунка 1.6

Функції, які не реалізуються одношаровою мережею, називаються лінійно нероздільними [2]. Рішення задач, які підпадають під це обмеження, полягає в застосуванні 2-х і більше шарових мереж або мереж з нелінійними синапсами, однак і тоді існує ймовірність, що коректний поділ деяких вхідних сигналів на класи неможливий.

Тепер ми можемо більш детально розглянути питання навчання НМ, для початку - на прикладі перцептрону з рисунка 1.4.

Розглянемо алгоритм навчання з учителем [5,6].

1. Проініціалізувати елементи вагової матриці (зазвичай невеликими випадковими значеннями).

2. Подати на входи один з вхідних векторів, які мережа повинна навчитися розрізняти, і обчислити її вихід.

3. Якщо вихід правильний, перейти на крок 4.

Інакше обчислити різницю між ідеальним і отриманим значеннями виходу:

$$\delta = Y_I - Y \tag{1.15}$$

Модифікувати ваги відповідно до формули:

$$w_{ij}(t+1) = w_{ij}(t) + \nu \cdot \delta \cdot x_i \tag{1.16}$$

де t и t+1 - номери відповідно поточної та наступної ітерації; ν - коефіцієнт швидкості навчання, 0<νJ1; і - номер входу; j - номер нейрона в шарі. Очевидно, що якщо YI > Y вагові коефіцієнти будуть збільшені і тим самим зменшать помилку. В іншому випадку вони будуть зменшені, і Y теж зменшиться, наближаючись до YI.

4. Цикл з кроку 2, поки мережа не перестане помилятися.

На другому кроці на різних ітераціях по черзі у випадковому порядку пред'являються всі можливі вхідні вектора. На жаль, не можна заздалегідь визначити число ітерацій, які потрібно виконати, а в деяких випадках і гарантувати повний успіх. Ці питання будуть опосередковано впливати у подальшому.

Нейронні мережі застосовуються для вирішення багатьох задач в економічному та технічному напрямку. Проаналізувавши роботу та структуру нейронних мереж, за багатьма принципами було обрано для роботи персептрон.

Метою роботи є дослідження та реалізація нейронних мереж для вирішення задач з розпізнавання образів.

Для досягнення поставленої цілі необхідно було виконати наступні завдання:

- розробити та дослідити схему апаратної реалізації нейрона;
- розробити та розрахувати пристрій розпізнавання образів;
- розробити програмний застосунок для розпізнавання дорожніх знаків.

2 ДОСЛІДЖЕННЯ НЕЙРОННИХ МЕРЕЖ ТА ЗАСТОСУВАННЯ ЇХ ДЛЯ ВИРІШЕННЯ ТЕХНІЧНИХ ЗАДАЧ

2.1 Розпізнавання образів

У 50-60-ті роки 20-го століття була визначена теорія статистичних рішень. В результаті цього визначення знайдені алгоритми, що забезпечують віднесення нового об'єкта до одного з заданих класів, що стало початком планомірного наукового пошуку і практичних розробок. В рамках кібернетики почав формуватися новий науковий напрям, пов'язаний з розробкою теоретичних основ і практичної реалізації пристроїв, а потім і систем, призначених для розпізнавання об'єктів, явищ, процесів. Нова наукова дисципліна отримала назву "Розпізнавання образів".

Розпізнавання образів (об'єктів, сигналів, ситуацій, явищ або процесів) - завдання ідентифікації об'єкта або визначення будь-яких його властивостей по його зображенню (оптичне розпізнавання) або аудіозапису (акустичне розпізнавання) та іншим характеристикам.

Таким чином, базою для вирішення завдань віднесення об'єктів до того чи іншого класу послужили результати класичної теорії статистичних рішень. В її рамках будувалися алгоритми, що забезпечують на основі експериментальних вимірювань параметрів, що характеризують цей об'єкт, а також деяких апріорних даних, що описують класи, визначення конкретного класу, до якого може бути віднесений об'єкт розпізнавання.

Приклади завдань розпізнавання образів:

- розпізнавання букв;

- розпізнавання штрих-кодів;
- розпізнавання автомобільних номерів;
- розпізнавання осіб та інших біометричних даних;
- розпізнавання мови.

2.2 Методи розпізнавання образів

В цілому, можна виділити три методи розпізнавання образів:

Метод перебору. У цьому випадку проводиться порівняння з базою даних, де для кожного виду об'єктів представлені різноманітні модифікації відображення. Наприклад, для оптичного розпізнавання образів можна застосувати метод перебору виду об'єкта під різними кутами, масштабами, зміщеннями, деформаціями і т. д. Для букв потрібно перебирати шрифт, властивості шрифту і т. д. У разі розпізнавання звукових образів, відповідно, відбувається порівняння з деякими відомими шаблонами (наприклад, слово, вимовлене кількома людьми).

Другий підхід – більш детальний аналіз характеристик образу. У разі оптичного розпізнавання це може бути визначення різних геометричних характеристик. Звуковий зразок в цьому випадку піддається частотному, амплітудному аналізу і т. д.

Третій метод - використання штучних нейронних мереж. Цей метод вимагає або великої кількості прикладів завдання розпізнавання при навчанні, або спеціальної структури нейронної мережі, яка враховує специфіку даного завдання. Проте, його відрізняє більш висока ефективність і продуктивність.

2.3 Загальна характеристика задач розпізнавання образів і їх типи

Задачі для розпізнавання мають наступні характерні риси. Це інформаційні завдання, що складаються з двох етапів:

- перетворення вихідних даних до виду, зручного для розпізнавання;
- власне розпізнавання (вказівка приналежності об'єкта певного класу).

У цих задачах можна вводити поняття аналогії або подібності об'єктів і формулювати правила, на підставі яких об'єкт зараховується в один і той же клас або в різні класи. Також можна оперувати набором прецедентів-прикладів, класифікація яких відома і які, у вигляді формалізованих описів, можуть бути пред'явлені алгоритмом розпізнавання для настройки на завдання в процесі навчання.

Загальна структура системи розпізнавання і етапи в процесі її розробки показані на рисунку 2.1.

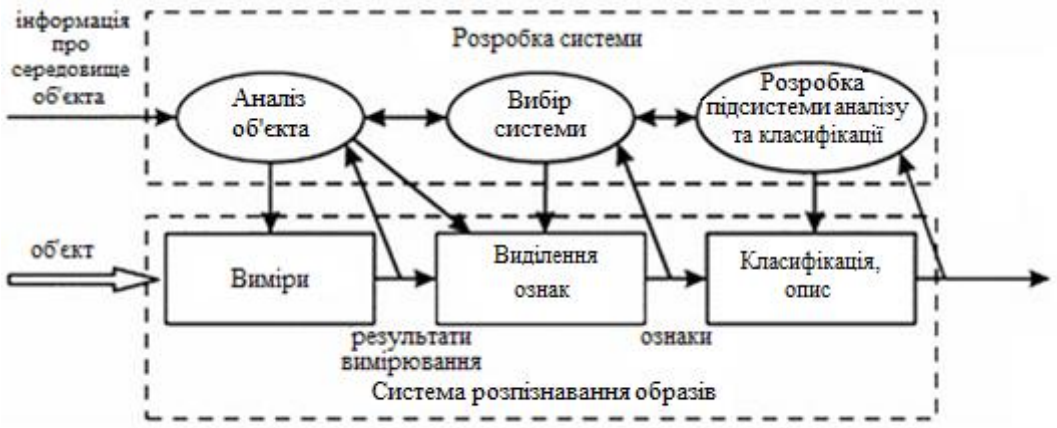


Рисунок 2.1- Структура системи розпізнавання образів

Виділяють наступні типи задач розпізнавання:

- задача розпізнавання - віднесення пред'явленого об'єкта за його описом до одного із заданих класів;

- задача автоматичної класифікації - розбиття множини об'єктів, ситуацій, явищ за їх описами на систему непересічних класів;

- задача вибору інформативного набору ознак при розпізнаванні;

- задача приведення вихідних даних до виду, зручного для розпізнавання;

- задача прогнозування - рішення повинне ставитися до деякого моменту в майбутньому.

Із розглянутих методів розпізнавання образів для даної роботи було обрано метод використання штучних нейронних мереж.

2.4 Розробка математичної моделі нейронної мережі

Нейромережеві методи, засновані на застосуванні різних типів штучних нейронних мереж (НМ), останнім часом набули широкого поширення. Основні завдання, які вирішуються за допомогою нейронних мереж [7,8,9,10] наведені на рисунку 2.2.

Зазвичай архітектура НМ реалізується на паралельних обчислювальних засобах: спеціалізованих мікросхемах, оптичних і квантових комп'ютерах. Це відкриває широкі перспективи застосування НМ в майбутньому. НМ характеризується нечітким розподіленим зберіганням інформації, тобто немає окремого нейрона, що відповідає за будь-яке поняття або ознаку, і видалення або спотворення роботи цього нейрона не приведе до фатальних наслідків.



Рисунок 2.2 - Основні задачі, які вирішуються за допомогою нейронних мереж

Ідея нейронної мережі базується на переконанні, що можна імітувати роботу мозку людини, створивши потрібні зв'язки за допомогою кремнію і провідників таких як у живих нейронів і дендритів. Людський мозок складається з 100 мільярдів нервових клітин, які називаються нейронами. Вони пов'язані з іншими тисячами клітин- аксонами. Подразники із зовнішнього середовища або сигнали від органів почуттів приймаються дендритами. Ці вхідні сигнали створюють електричні імпульси, які швидко переміщаються через нейромережу. Потім нейрон може посилати повідомлення на інші нейрони, які можуть відправити це повідомлення далі або, можуть взагалі її не відправляти після обробки інформації.

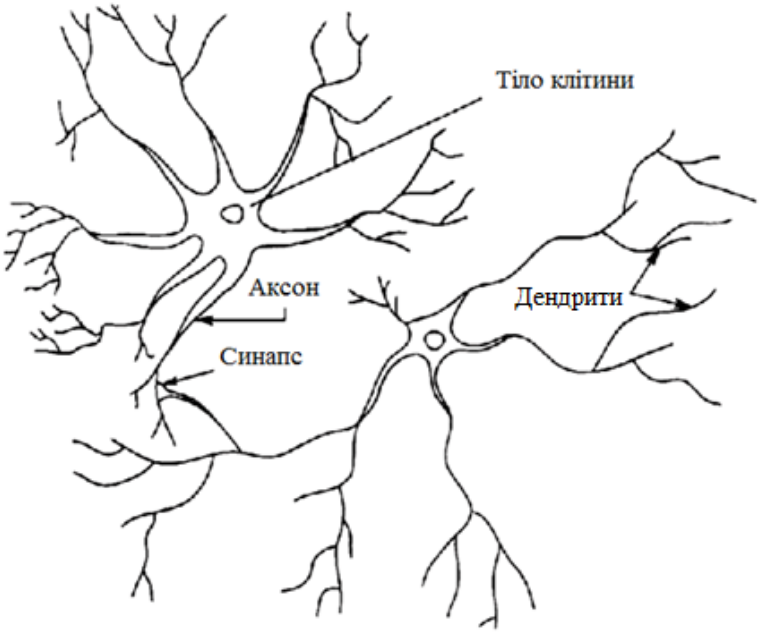


Рисунок 2.3 - Типові біологічні нейрони

Штучні нейронні мережі складаються з декількох вузлів, які імітують біологічні нейрони людського мозку. Нейрони з'єднані між собою і взаємодіють один з одним. Вузли можуть приймати вхідні дані і виконувати найпростіші операції над даними. В результаті цих операцій дані передаються іншим нейронам. Вихідні дані для кожного вузла називаються його активацією [11].

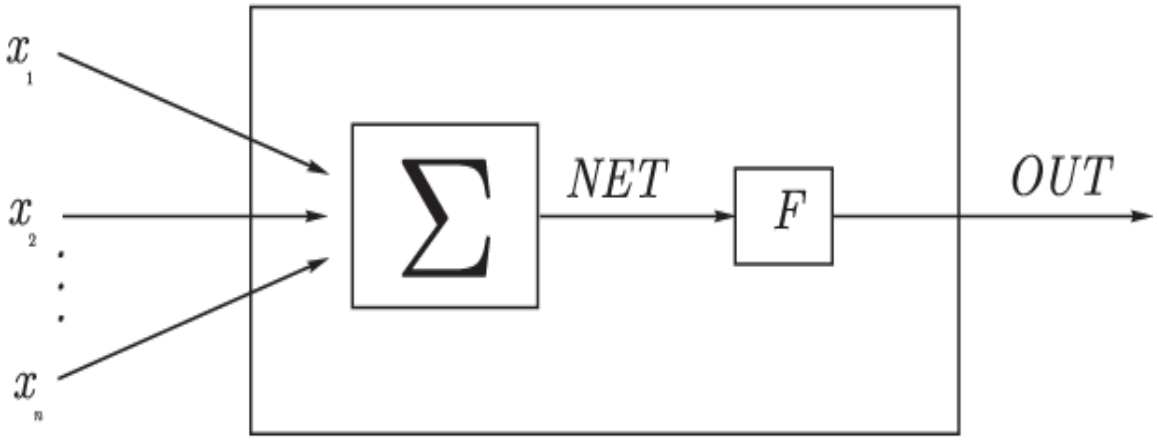


Рисунок 2.4 - Штучний нейрон

На рисунку 2.4 представлена модель, що реалізує цю ідею. Більшість вхідних сигналів, позначених $x_1, x_2, \dots, x_n$ надходить на штучний нейрон. Ці вхідні сигнали, в сукупності позначаються вектором X, що відповідають сигналам, які приходять в синапси біологічного нейрона. Кожен сигнал множиться на відповідний ваговий коефіцієнт $w_1, w_2, \dots, w_n$ і поступає на сумуючий блок, позначений $\sum$. Кожен ваговий коефіцієнт відповідає "силі" одного біологічного синаптичного зв'язку (безліч вагових коефіцієнтів в сукупності позначається вектором W). Сумуючий блок, що відповідає тілу біологічного елемента, складає зважені входи алгебраїчно, створюючи вихід, який ми будемо називати NET. У векторних позначеннях це може бути компактно записане таким чином:

$$X_1 w_1 + X_2 w_2 + \cdots + X_n w_n = \sum_{i=1}^{n} x_i w_i$$

Сигнал NET далі, як правило, перетворюється активаційною функцією F і дає вихідний нейронний сигнал OUT. Активаційна функція може бути звичайною лінійною функцією:

$$\mathrm{OUT} = \mathrm{F}(\mathrm{NET}), \qquad (2.2)$$

де F — порогова функція.

$$\mathrm{OUT} = \begin{cases} 1, \text{якщо } \mathrm{NET} > T; \\ 0, \text{якщо } \mathrm{NET} \leq \mathrm{T}; \end{cases} \qquad (2.3)$$

де T — деяка постійна порогова величина.

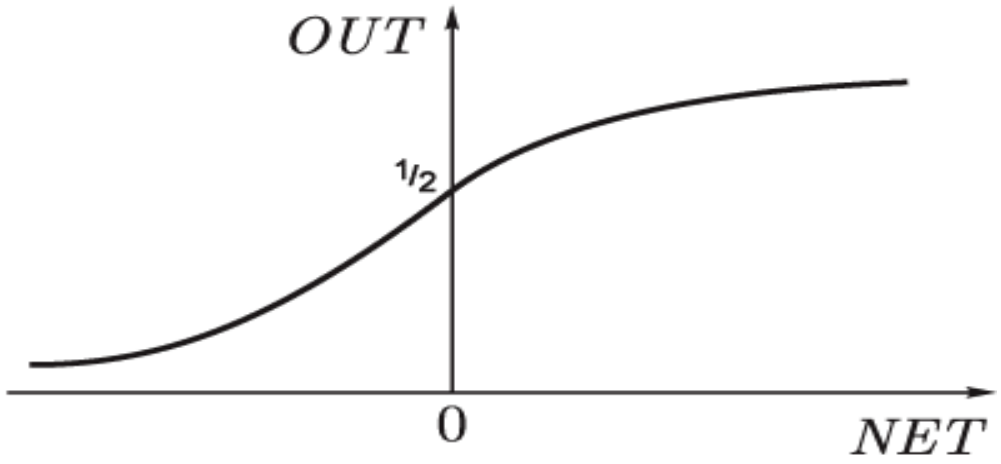


Рисунок 2.5 - Функція F

На рисунку 2.4 блок, позначений F, приймає сигнал NET і видає сигнал OUT. Якщо блок F звужує діапазон зміни величини NET так, що при будь-яких значеннях NET, значення OUT належать деякому кінцевому інтервалу, то F називається "стискаючою" функцією. В якості "стискаючої" функції часто використовується логістична або "сигмоідальна" (S-образна) функція, показана на рисунку 2.5. Ця функція математично виражається як:

$$F(x) = \frac{1}{(1+ e^{-x})} \quad (2.4)$$

Таким чином:

$$OUT = \frac{1}{1+ e^{-NET}} \quad (2.5)$$

За аналогією з електронними системами, активаційну функцію можна вважати нелінійною підсилювальною характеристикою штучного нейрона. Коефіцієнт посилення обчислюється як відношення приросту величини OUT до викликаного нею невеликого приросту величини NET. Він виражається нахилом кривої при певному рівні збудження і змінюється від малих значень, при великих негативних збудженнях (крива майже горизонтальна), до максимального значення, при нульовому збудженні і знову зменшується, коли

збудження стає великим позитивним. С. Гросберг (1973) виявив, що подібна нелінійна характеристика вирішує поставлену їм дилему шумового насичення. Яким чином одна і та ж мережа може обробляти як слабкі, так і сильні сигнали? Слабкі сигнали мають потребу у великому мережевому посиленні, щоб дати придатний до використання вихідний сигнал. Однак підсилювальні каскади з великими коефіцієнтами посилення можуть призвести до насичення виходу шумами підсилювачів (випадковими флуктуаціями), які присутні в будь-який фізично реалізованої мережі. Сильні вхідні сигнали, в свою чергу, також будуть приводити до насичення підсилюючих каскадів, виключаючи можливість корисного використання виходу. Центральна область логістичної функції, що має великий коефіцієнт посилення, вирішує проблему обробки слабких сигналів, в той час як області з падаючим посиленням на позитивному і негативному кінцях підходять для великих збуджень. Таким чином, нейрон функціонує з великим посиленням в широкому діапазоні рівня вхідного сигналу

$$\mathrm{OUT} = \frac{1}{1+ e^{-\mathrm{NET}}} = \mathrm{F(NET)} \qquad (2.6)$$

Іншою функцією, що широко використовується активаційною функцією є гіперболічний тангенс. За формою вона схожа з логістичною функцією і часто використовується біологами як математична модель активації нервової клітини. В якості активаційної функції штучної нейронної мережі вона записується в такий спосіб:

$$\mathrm{OUT} = \mathrm{th(x)} \qquad (2.7)$$

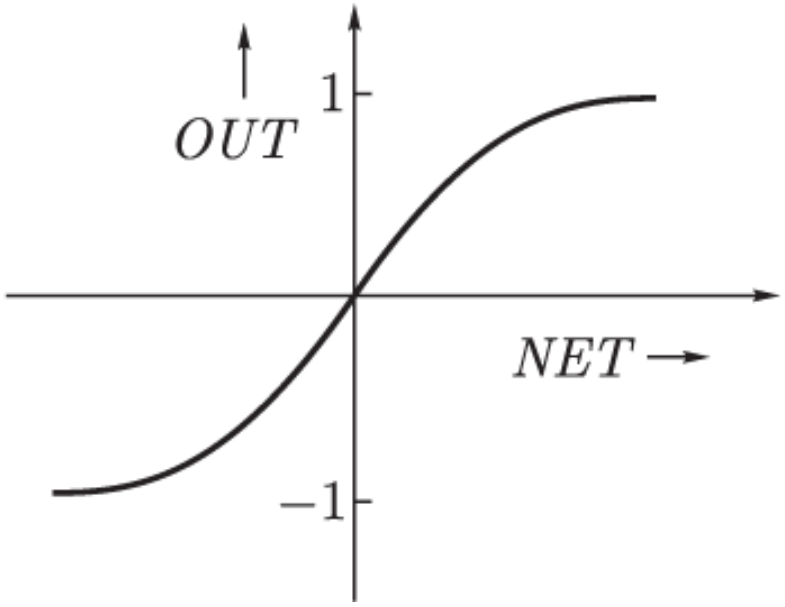


Рисунок 2.6- Функція F гіперболічний тангенс

Подібно логістичної функції гіперболічний тангенс є S-образною функцією, але він симетричний відносно початку координат, і в точці NET = 0 значення вихідного сигналу OUT дорівнює нулю (див. рис. 2.6). На відміну від логістичної функції, гіперболічний тангенс приймає значення різних знаків, і ця його властивість застосовується для цілого ряду мереж.

Розглянута проста модель штучного нейрона ігнорує багато властивостей свого біологічного двійника. Наприклад, ця модель не бере до уваги затримку в часі, яка впливає на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал. І, що більш важливо, вона не враховує впливів функції частотної модуляції або синхронізуючої функції біологічного нейрона, як і ряд дослідників вважають вирішальними в нервовій діяльності природного мозку.

Незважаючи на ці обмеження, мережі, побудовані з таких нейронів, виявляють властивості, які сильно нагадують біологічну систему. Тільки час і дослідження зможуть відповісти на питання, чи є подібні збіги випадковими або ж вони є наслідком того, що в моделі вірно схоплені найважливіші риси біологічного нейрона.

2.5 Види нейронних мереж

В даний час крім багатошарового персептрона існує безліч способів задання структур нейронних мереж. Всі види нейронних мереж можна умовно розділити на мережі прямого поширення і мережі з зворотними зв'язками. На рисунку 2.7 представлена така класифікація найбільш поширених типів НМ.

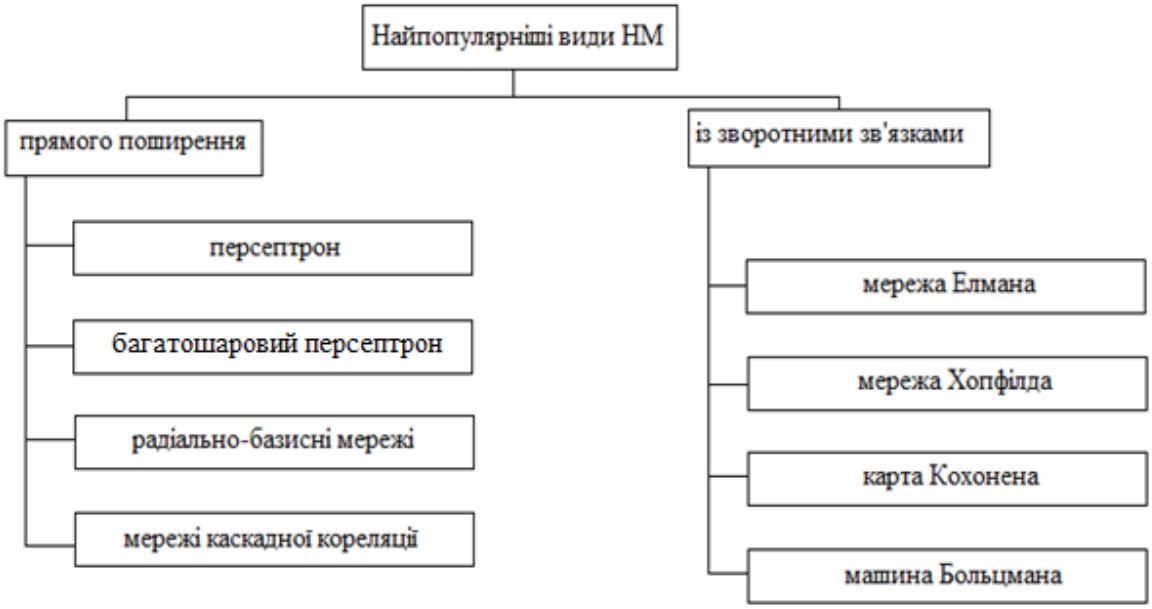


Рисунок 2.7 - Класифікація найпоширеніших видів НМ

2.6 Типи нейронних мереж

Виділяється кілька основних типів архітектур нейронних мереж:

- Мережа прямого поширення (feed-forward network) - має на увазі, що нейрони і зв'язки між ними утворюють ациклічний граф, де сигнали поширюються тільки в одному напрямку. Саме такі мережі є найбільш

популярними і широко вивченими, і їх навчання представляє найменші труднощі.

- Рекурентні нейронні мережі (recurrent neural networks) - в таких мережах, на відміну від мереж прямого поширення, сигнали може передаватися в обох напрямках, і можуть надходити на один і той же нейрон кілька разів в процесі обробки одного вхідного значення. Поширеним різновидом рекурентних нейронних мереж є, наприклад, машина Больцмана. Основною труднiстю в роботі з такими мережами є їх навчання, так як створити ефективний алгоритм для цього в загальному випадку є складним завданням і до сих пір не має універсального рішення.

- Самоорганізуючі карти Кохонена - нейронна мережа, призначена в першу чергу для кластеризації і візуалізації даних.

В історії розвитку нейронних мереж виділяють 3 основних періоди підйому. Перші дослідження в області штучних нейронних мереж відносяться до 40-х років 20 століття. У 1954 році Дж. Маккалок і У. Піттс опублікували роботу «Логічне числення ідей, що відносяться до нервової діяльності» [12], в якій були викладені основні принципи побудови штучних нейронних мереж. У 1949 році була опублікована книга Д.Хебба «Організація поведінки», де автор розглянув теоретичні основи навчання нейронних мереж і вперше сформулював концепцію навчання нейронних мереж як налаштування вагових коефіцієнтів між нейронами. У 1954 році В.Кларк вперше здійснив спробу реалізувати аналог мережі Хебба за допомогою комп'ютера. У 1958 році Ф. Росенблатт запропонував модель персептрона, який представляв собою по суті нейронну мережу з одним прихованим шаром [13]. Принциповий вигляд персептрона Росенблатта представлений на рисунку 2.8.

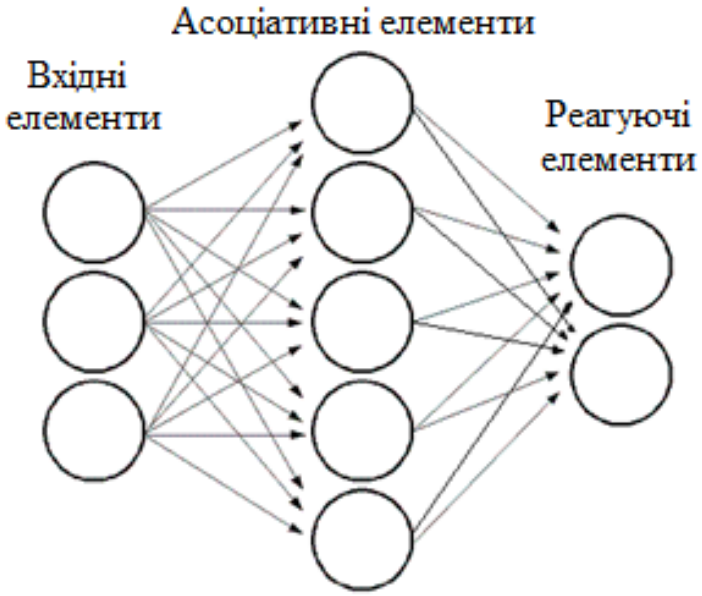


Рисунок 2.8 - Персептрон Росенблата

Дана модель навчалася за допомогою методу корекції помилок, який полягав в тому, що вагові коефіцієнти залишаються незмінними до тих пір, поки вихідне значення персептрона коректно, в разі ж помилки ваговий коефіцієнт зв'язку змінюється на одиницю в бік, зворотний знаку помилки. Даний алгоритм, як було доведено Росенблатом, завжди сходиться. Використовуючи таку модель, вдалося створити комп'ютер, що розпізнає деякі букви латинського алфавіту, що, безсумнівно, було великим успіхом в той час.

Однак інтерес до нейронних мереж істотно знизився після публікації М. Мінським і С. Папертом книги «Персептрони» ("Perceptrons") в 1969 році [14], де вони описали істотні обмеження, які має модель персептрона, зокрема, неможливість подання функції “виключає або”, а також вказали на занадто високі вимоги до необхідної обчислювальної потужності комп'ютерів для навчання нейронних мереж. Так як дані вчені мали дуже високий авторитет в науковому співтоваристві, нейронні мережі були на деякий час визнані

малоперспективною технологією. Ситуація змінилася тільки після створення в 1974 році алгоритму зворотного поширення помилки.

Алгоритм зворотного поширення помилки (backpropagation algorithm) був запропонований в 1974 році одночасно і незалежно двома вченими, П.Вербосом і А. Галушкіним. Даний алгоритм заснований на методі градієнтного спуску. Основна ідея алгоритму полягає в поширенні інформації про помилку від виходів мережі до її входів тобто в зворотному напрямку по відношенню до стандартного проходу.

Цей спад завершився тільки в 2000-х роках 21 століття, коли з'явилося і стало поширюватися поняття deep learning, або глибоке навчання. Відродженню нейронних мереж сприяла поява нових архітектур, таких, наприклад, як згорткові мережі, restricted bolzman machines, стекові автоенкодери і т.д., які дозволили домогтися істотно більш високих результатів в таких сферах машинного навчання, як розпізнавання зображень і мови.

Однією з найбільш важливих проблем, з якими доводиться стикатися при використанні алгоритмів машинного навчання, є проблема вибору правильних ознак, на основі яких проводиться навчання. Особливо значимою стає дана проблема при розгляді таких завдань, як розпізнавання зображень, розпізнавання мови, обробка мови і тому подібних, тобто тих, де відсутній очевидний набір ознак, які можуть бути використані для навчання.

З точки зору теорії машинного навчання, deep learning є підмножиною так званого representation learning. Основна концепція representation learning - якраз таки автоматичний пошук ознак, на підставі яких в подальшому буде працювати деякий алгоритм, наприклад, класифікації.

З іншого боку, ще одна важлива проблема, з якою доводиться стикатися при використанні машинного навчання - це наявність факторів варіації, які можуть зробити істотний вплив на зовнішній вигляд вихідних даних, однак при цьому не мають відношення до самої їх суті, яку дослідник і намагається аналізувати [15]. Так, в задачі розпізнавання зображень такими факторами можуть бути кут, під яким предмет на зображенні повернутий до спостерігача,

час доби, освітлення і т.д. Так, в залежності від точки зору і погоди червона машина може мати на фотографії різний відтінок і форму. Тому для подібних завдань, наприклад, ідентифікації предмета, зображеного на фотографії, виглядає розумним враховувати не конкретні низькорівневі факти, такі як колір певного пікселя, а характеристики вищого рівня абстракції, наприклад, наявність коліс.. Deep learning заснований на наданні вихідного об'єкта у вигляді ієрархічної структури ознак, таким чином, що кожен наступний рівень ознак будується на основі елементів попереднього рівня. Так, якщо мова йде про зображення, як самого нижчого рівня будуть виступати вихідні пікселі зображення, наступним рівнем будуть відрізки, які можна виділити серед цих пікселів, потім - кути і інші геометричні фігури, в які складаються відрізки. На наступному рівні їхніх фігур утворюються вже впізнавані для людини об'єкти, наприклад, колеса, і нарешті, останній рівень ієрархії відповідає за конкретні предмети на зображенні, наприклад, автомобіль.

Для реалізації підходу deep learning в сучасній науці використовуються багатошарові нейронні мережі різних архітектур. Нейронні мережі ідеально підходять для вирішення завдання виявлення з даних і побудови ієрархічної сукупності ознак, так як, по суті, нейронна мережа являє собою сукупність нейронів, кожен з яких активується тільки якщо вхідні дані задовольняють певним критеріям - тобто є певна ознака, при цьому правило активації нейрона - то, що і визначає цей показник - навчається автоматично. Поширенню такого підходу і, в зв'язку з цим, чергового розквіту нейронних мереж, послужило три взаємопов'язаних причини:

- Поява нових архітектур нейронних мереж, заточених для вирішення певних завдань (згорткові мережі, машини Больцмана і т.д.).

- Розвиток і доступність обчислень з використанням gpu і паралельних обчислень в цілому.

- Поява і поширення підходу пошарового навчання нейронних мереж, при якому кожен шар навчається окремо за допомогою стандартного алгоритму backpropagation (зазвичай на нерозмічених даних, тобто по суті відбувається

навчання автоенкодера), що дозволяє виявити суттєві ознаки на даному рівні, а потім всі верстви об'єднуються в єдину мережу і відбувається донавчання мережі вже із застосуванням розмічених даних для вирішення конкретного завдання (fine-tuning).

Розглянемо детальніше основні архітектури нейронних мереж, які як правило, використовуються в контексті глибокого навчання.

- Багатошаровий персептрон - являє собою звичайну повнозв'язну нейронну мережу з великою кількістю шарів. Питання про те, яка кількість шарів вважається досить великою, не має однозначної відповіді, але зазвичай мережі, що мають 5-7 шарів, вже вважаються «глибокими».

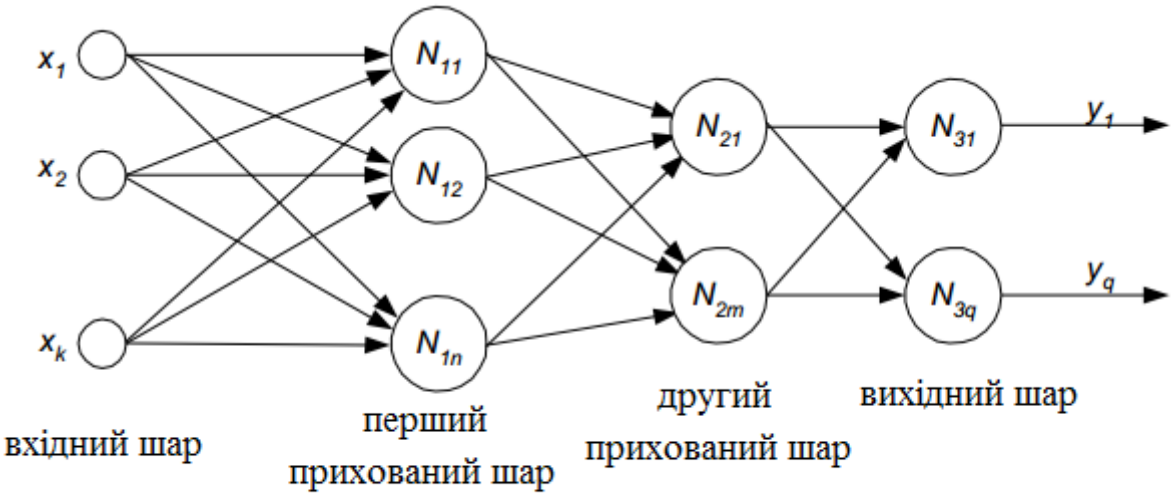


Рисунок 2.9- Багатошаровий персептрон

Дана архітектура нейронних мереж, хоча і не має принципових відмінностей від мереж, які використовувалися раніше до поширення поняття глибокого навчання, може виявитися вельми ефективною в разі успішного вирішення завдання її навчання, що було головною проблемою роботи з такими мережами раніше. В даний час проблема вирішується шляхом використання для навчання мережі графічних карт, що дозволяє прискорити навчання і, відповідно, провести більшу кількість ітерацій навчання, або пошарового

навчання мережі, згаданого раніше. Так, в 2012 році Ciresan з колегами опублікували статтю «Deep big multilayer perceptrons for digit recognition» [14], в якій зробили припущення, що багатошаровий персептрон з великою кількістю шарів, в разі достатньої тривалості навчання (яка досягається за розумний час з використанням паралельних обчислень на gpu) і достатній кількості даних для навчання (яке досягається шляхом застосування різних випадкових трансформацій до вихідної безлічі даних) може показати результативність не гіршу ніж інші, більш складні моделі. Їх модель, що представляє собою нейронну мережу з 5 прихованими шарами, при класифікації цифр з датасета MNIST, показала відсоток помилки 0.35, що краще, ніж опубліковані раніше результати більш складних моделей. Також, шляхом об'єднання кількох навчених таким чином мереж в єдину модель, їм вдалося знизити показник помилки до 0.31%. Таким чином, незважаючи на уявну простоту, багатошаровий персептрон є цілком успішним представником алгоритмів глибокого навчання.

- Stacked autoencoder (стековий автоенкодер) - дана модель тісно пов'язана з багатошаровим персептроном і в цілому із задачами навчання глибоких нейронних мереж. Саме з використанням стекового автоенкодера реалізується пошарове навчання глибоких мереж. Однак дана модель використовується не тільки для цілей навчання інших моделей, а часто має велике практичне значення сама по собі. Щоб описати суть стекового автоенкодера, розглянемо спочатку поняття звичайного автоенкодера. Автоенкодер є алгоритмом навчання без вчителя, в якому в якості очікуваних вихідних значень нейронної мережі виступають її ж вхідні значення. Схематично модель автоенкодера представлена на рисунку 2.10.

Очевидно, що завдання навчання подібної моделі має тривіальне рішення. Якщо кількість нейронів в прихованому шарі дорівнює кількості вхідних нейронів - тоді прихованому шару досить просто транслювати свої вхідні значення на вихід. Тому при навчанні автоенкодеров вводяться додаткові обмеження, наприклад, кількість нейронів в прихованому шарі встановлюється

значно менше ніж у вхідному шарі, або застосовуються особливі методики регуляризації, спрямовані на те, щоб забезпечити високу ступінь розрідженості нейронів прихованого шару [15].

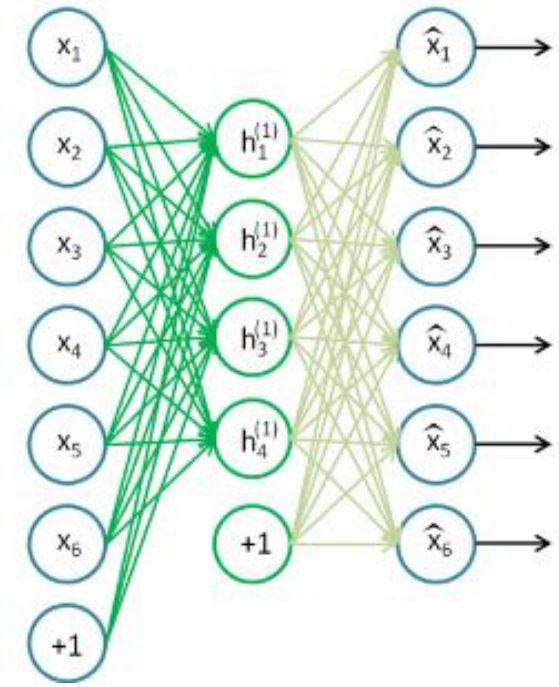


Рисунок 2.10 - Класичний автоенкодер

Одне з найбільш поширених застосувань автоенкодеров в чистому вигляді - це завдання отримання стисненого представлення вихідних даних. Так, наприклад, автоенкодер з 30 нейронами в прихованому шарі, навчений на датасеті MNIST, дозволяє відновити на вихідному шарі вихідні зображення цифр практично без змін, що означає, що насправді кожне з вихідних зображень можна досить точно описати тільки 30 числами. В даному своєму застосуванні автоенкодери часто розглядаються в якості альтернативи методу головних компонент. Стековий же автоенкодер є по суті комбінацію декількох звичайних автоенкодеров, яких навчають шар за шаром. При цьому вихідні значення навчених нейронів прихованого шару першого з автоенкодеров виступають в якості вхідних значень для другого з них, і так далі.

- Згорткові мережі - одна з найбільш популярних останнім часом моделей глибокого навчання, що застосовується в першу чергу для розпізнавання зображень [15]. Концепція згортальних мереж побудована на трьох основних ідеях:

1- Локальна чутливість (local receptive fields) - якщо говорити про завдання розпізнавання зображень, це означає, що на розпізнавання того чи іншого елемента на зображення має в першу чергу впливати його безпосереднє оточення, в той час як пікселі, що знаходяться в іншій частині зображення, швидше за все з даним елементом ніяк не пов'язані і не містять інформації, яка допомогла б правильно його ідентифікувати.

2- Розділені вагові коефіцієнти (shared weights) - наявність в моделі розділених вагових коефіцієнтів фактично уособлює припущення, що один і той же об'єкт може бути знайдений в будь-якій частині зображення, при цьому для його пошуку у всіх частинах зображення застосовується один і той же патерн (набір вагових коефіцієнтів)

3- Сабсемплінг (subsampling) - концепція, що дозволяє зробити модель більш стійкою до незначних відхилень від шуканого патерну - в тому числі, пов'язаних з дрібними деформаціями, зміною освітлення і т.д. Ідея сабсемплінга полягає в тому, що при зіставленні з патерном враховується не точне значення для даного пікселя або області пікселів, а його агрегація в деякому околі, наприклад, середнє або максимальне значення.

З математичної точки зору, основою згортальних нейронних мереж є операція матричної згортки, яка полягає в поелементному перемножуванні матриці, що представляє собою невелику ділянку вихідного зображення (наприклад, 7 * 7 пікселів) з матрицею того ж розміру, званої ядром згортки, і подальше підсумовування отриманих значень . При цьому ядро згортки є по суті деякий шаблон, а отримане в результаті підсумовування число характеризує ступінь схожості даної області зображення на цей шаблон.

$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} * I_{x+i-1,y+j-1} \quad (1.8)$$

На рисунку 2.11 схематично зображена вищевказана формула, а також представлений результат застосування операції згортки (з двома різними ядрами) до зображення з метою виділити контури об'єкта.

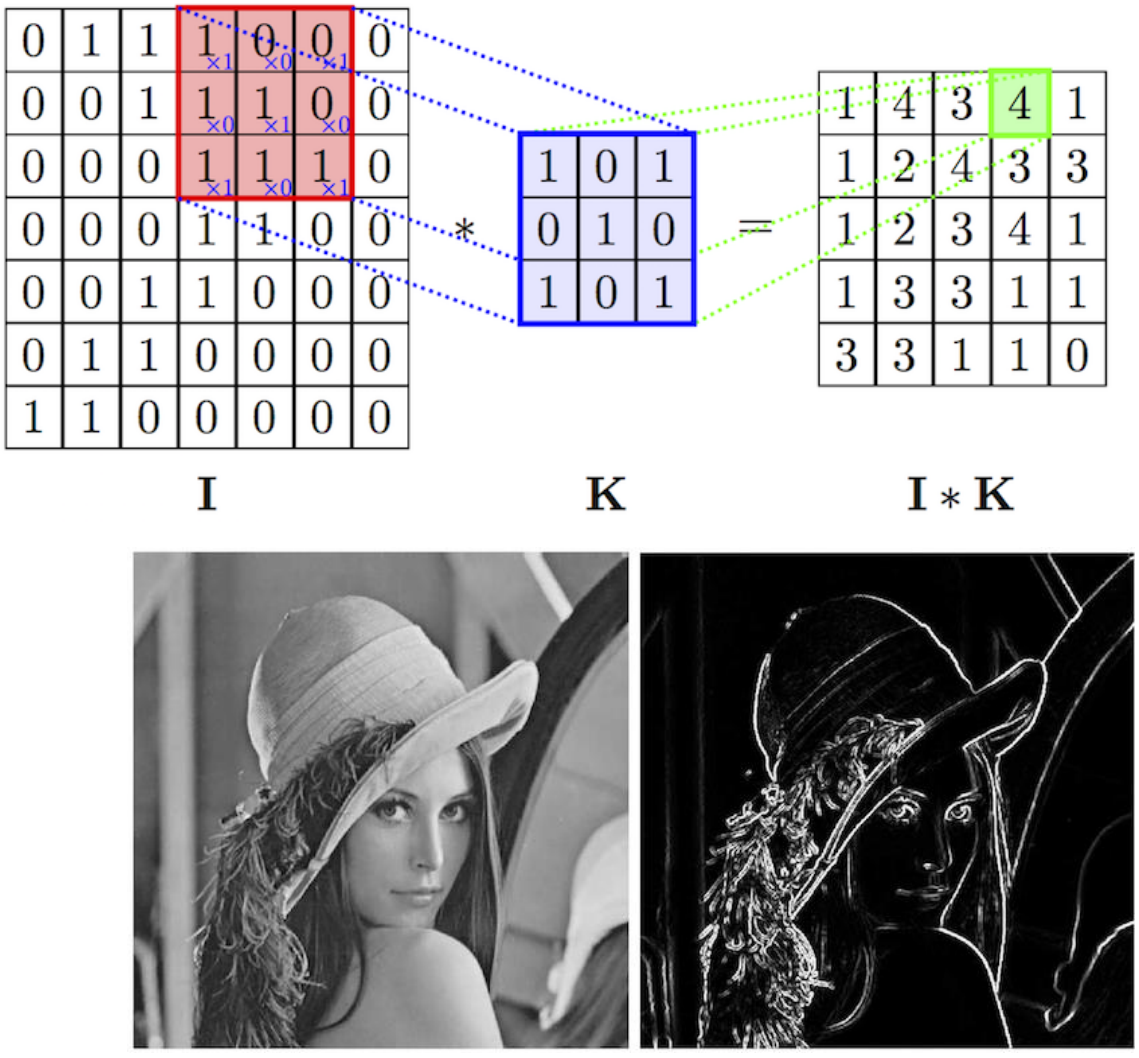


Рисунок 2.11 – Результат операції матричної згортки

Відповідно, кожен шар згортальної мережі складається з певної кількості шаблонів, і завдання навчання мережі полягає в підборі правильних значень в цих шаблонах - так, щоб вони відображали найважливіші характеристики вихідних зображень. При цьому кожен шаблон порівнюється послідовно з усіма частинами зображення - саме в цьому знаходить вираз ідея розподілу вагових

коефіцієнтів. Шари такого типу в згортковій мережі називаються шарами згортки. Крім шарів згортки, в згортальних мережах присутні шари сабсемплінга, які замінюють невеликі області зображення одним числом, тим самим одночасно зменшуючи розмір зразка для роботи наступного шару і роблячи мережу більш стійкою до невеликих змін в даних. В останніх же шарах в згортковій мережі зазвичай використовується один або кілька повнозв’язних шарів, яких навчають для виконання безпосередньо класифікації об'єктів. В останні роки використання згортальних мереж стало фактично стандартом при класифікації зображень і дозволяє добути найкращих результатів в цій області.

- Обмежені машини Больцмана (Restricted Boltzmann Machines) - ще один різновид моделей глибокого навчання, на відміну від згортальних мереж, що застосовуються в першу чергу для задачі розпізнавання мови. Машина Больцмана в своєму класичному розумінні являє собою неорієнтований граф, ребра в якому відображають залежності між вузлами (нейронами). При цьому частина нейронів є видимими, а частина прихованими. З точки зору нейронних мереж машина Больцмана являє собою по суті рекурентну нейронну мережу, з точки зору статистики - випадкове Марківське поле. Важливими поняттями для машин Больцмана є поняття енергії мережі і стану рівноваги. Енергія мережі залежить від того, яка кількість сильно пов'язаних між собою нейронів одночасно знаходяться в активованому стані, завдання ж навчання такої мережі полягає в її сходженні до стану рівноваги, при якому її енергія мінімальна. Основним недоліком подібних мереж є великі проблеми з навчанням їх у загальному вигляді. Для вирішення даної проблеми Дж. Хінтоном з колегами була запропонована модель обмеженої машини Больцмана, яка накладає обмеження на структуру мережі, представляючи її у вигляді дводольного графа, в одній частині якого знаходяться тільки видимі нейрони, а в іншій - тільки приховані, відповідно, зв'язки присутні тільки між видимими і прихованими нейронами. Дане обмеження дозволило розробити ефективні алгоритми для навчання мереж такого виду, завдяки чому було здійснено суттєвий прогрес у

вирішенні завдань розпізнавання мови, де дана модель практично витіснила популярну раніше модель прихованих мереж Маркова.

Серед розглянутих архітектур нейронних мереж, можна зробити висновок, що для розпізнавання образів більш підходящим варіантом буде обрати згорткову нейронну мережу. Саме цей тип нейронних мереж застосовується в першу чергу для розпізнавання зображень.

2.7 Апаратна реалізація нейронної мережі

Сучасні штучні нейронні мережі(ШНМ) в силу своїх здібностей до моделювання нелінійних процесів, узагальнення, роботі зашумленими даними і т.д. знаходять застосування в багатьох областях. При цьому все більше зростає інтерес апаратних нейромережевих реалізацій. Це пов'язано з вимогами до забезпечення високих швидкостей вирішення складних задач з використанням масового паралелізму [16], з якими поки не справляються сучасні комп'ютери.

Однією з перспективних апаратних реалізацій ШНМ є побудова їх на основі ПЛІС - матричних структур, що забезпечують паралельну обробку інформації і володіють високим ступенем інтеграції, масштабованість, розвиненими засобами проектування, реалізації виконання алгоритмів навчання[16].

Більш прийнятною в практиці моделлю ШНМ є модель, що складається з нейронів з безперервною (як правило, сигмоідальною) функцією активації[16]. При цьому простір рішень, в якому працює така мережа, не обмежується дискретними значеннями і складається з повного спектру можливих входів і їх значень, завдяки чому мережа здатна апроксимувати практично будь-яку функціональну залежність. Крім того, сигмоідальна функція активації має властивість підсилювати слабкі сигнали і запобігає насиченню нейрона від великих сигналів[12].

Необхідно відзначити, що в ШНМ з пороговими елементами, навчання проводитися лише на її моделі, заносячи отримані вагові коефіцієнти в пам'ять кристала на етапі проектування. Використання в апаратних реалізаціях багатошарових нейронних мереж нейронів з сигмоідальною функцією активації може дозволити виробляти навчання нейромережі на кристалі в реальному масштабі часу.

Один з варіантів архітектурного рішення штучного нейрона на основі ПЛІС представлено на рисунку 2.12.

Структура нейрона складається з різних обчислювальних підблоків[16]. Для визначення значення функції активації використовується таблиця (LUT), що зберігається в блоці пам'яті RAM. У структурі міститься один перемножувач (MUL) і один суматор (ADD). Вхідні сигнали (input) вводиться паралельно до ROM1. CONTROL UNIT, керуючи стартовими сигналами (start), кінцевими сигналами (reset) і лічильником сигналів нейрона (num) координує функції блоків, забезпечує синхронізацію вхідних сигналів і відповідних їм вагових коефіцієнтів. Перший вхідний сигнал і відповідний йому ваговий коефіцієнт зберігатися в ROM2, після чого блок MUL послідовно їх перемножує і заносить в ROM3. Та ж операція проводиться і для інших вхідних сигналів, після чого суматор ADD сумує зважені входи (sum) і додає до них значення зміщення BIEST. Результат підсумовування (sum_out) надходить в таблицю визначення значень сигмоидальної функції LUT, після чого отримане значення зберігається в блоці OUT, який подає сигнал про закінчення процесу.

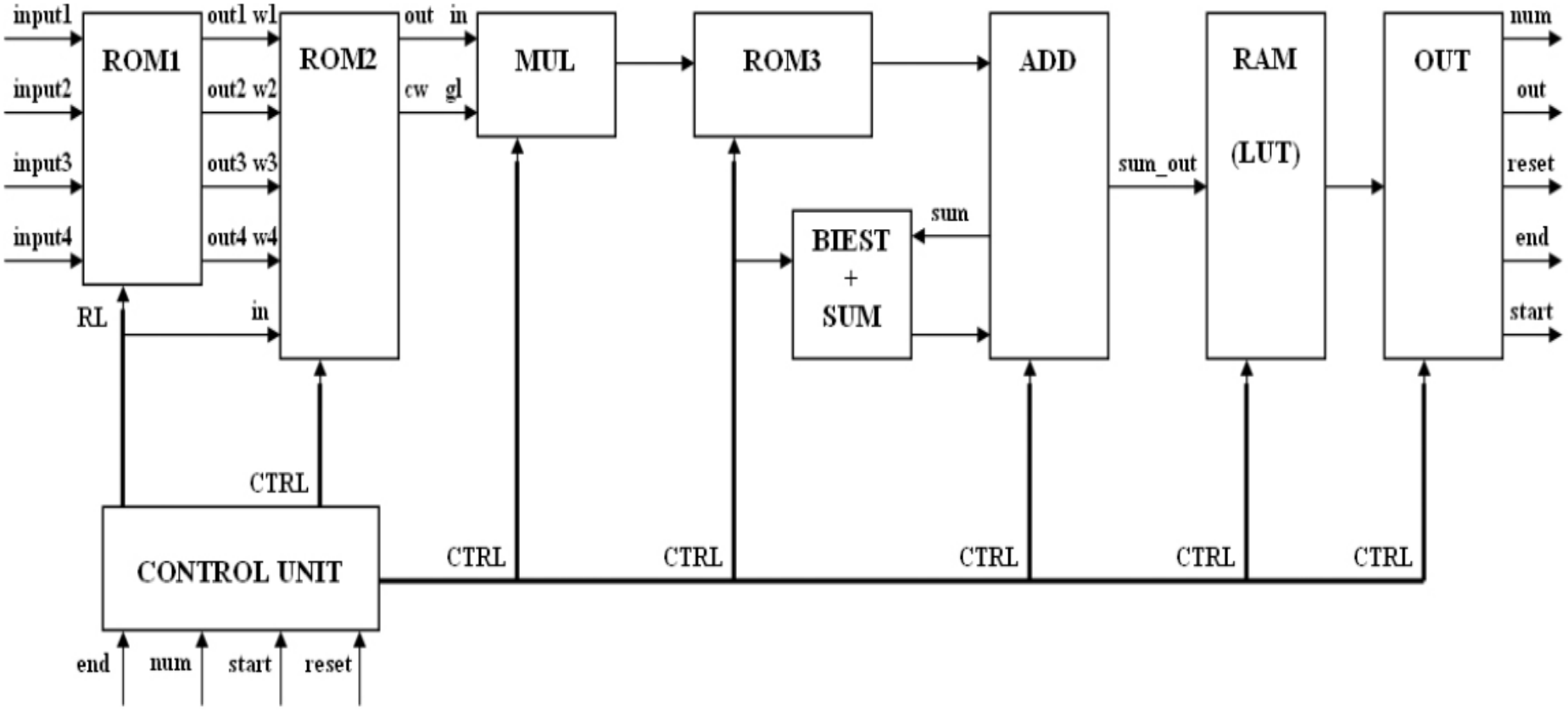


Рисунок 2.12 – Структурна схема апаратної реалізації нейрона на основі ПЛІС

2.8 Програмне забезпечення для моделювання нейронних мереж

В даний час вирішення основної частини практичних завдань за допомогою НМ забезпечується використанням комерційного програмного забезпечення на основі методів нейрологіки, виконання яких відбувається за допомогою центрального процесора. Це цілком логічно, головним чином через те, що це відносно нова галузь, а створення ПО - досить гнучкий процес, що дозволяє тестувати і впроваджувати з малими витратами деякі експериментальні методи.

Розрізняють універсальні та прикладні програмні продукти для моделювання нейронних мереж. Універсальні або об'єктно-інваріантні середовища дозволяють синтезувати оптимальні нейронні мережі, що застосовуються для вирішення широкого класу задач, з пропозицією різних парадигм і алгоритмів навчання. Прикладні середовища моделювання орієнтовані для синтезу нейронних мереж, що застосовуються в тій чи іншій галузі, прикладної області або іншої специфічної задачі.

Серед найважливіших властивостей нейромережевих симуляторів - здатність синтезувати код програми результуючої нейронної мережі на алгоритмічній мові високого рівня (найчастіше - C, C ++, C #, Delphi). Такий код згодом легко інтегрувати в призначену для користувача програму. Наприклад Програмний продукт TRAJAN компанії TRAJAN Software Co (Великобританія) є симулятором повного циклу і призначений для моделювання в середовищі Windows різних видів нейронних мереж і алгоритмів навчання. Симулятор включає широкі можливості для графічного і статистичного контролю в процесі моделювання параметрів і характеристик ефективності синтезованих нейронних мереж. У таблиці 2.1 представлені особливості найбільш поширених універсальних програмних середовищ для моделювання нейронних мереж [13].

Таблиця 2.1- Програмні засоби моделювання нейронних мереж

Назва	Розробник	Плат-форма	Підтримувані парадигми і алгоритми навчання	Інтерфейс	Ціна	Коментарі
Matlab Neural network Toolbox 3.0	MathWorks, США	Win 95,98	Персептрон, мережі Ельмана, мережі Хопфілда, Хебб, самоорганізовані карти.	GUI для Ms Windows	----	Генерує ANSI-сумісний код
Trajan	Trajan software Ltd., Великобританія	Win 3.x,9x, NT	Багатошаровий персептрон, карти Кохонена, зворотне поширення	GUI для Ms Windows	620$	Демо-версія програми доступна на сайті виробника
X-Sim	Іспанія (Мадрид)	Unix, Linux	Багатошаровий персептрон, карти Кохонена, зворотне поширення	Режим командної строки для ОС типу Unix	безкоштовно	Демо-версія
Aspirin/ MIGRAINES	Mitre Corp.	Unix	Зворотне поширення	GUI для Ms Windows	безкоштовно	Зберігає данні НМ на диску в доступному форматі
Neuron	Університет Дюка, США	Unix	Зумовлені мережі	GUI для Ms Windows	безкоштовно	Демо-версія
Sankom	Дортмунський Университет, Германія	Unix	Карти Кохонена	Режим командної строки Shell	безкоштовно	Демо-версія

2.9 Розробка системи розпізнавання дорожніх знаків

Досить часто початківці і вже досвідчені водії стикаються з низкою проблем під час керування транспортним засобом. Водієві необхідно встигати стежити за розміткою на дорозі і бачити всі дорожні знаки, щоб уникнути аварійних і непередбачуваних ситуацій. Для полегшення життя автомобілістів запропоновано систему розпізнавання дорожніх знаків.

Система розпізнавання дорожніх знаків являє собою програмне забезпечення, яке сканує вказане зображення і розпізнає дорожній знак. У більшості випадків такі системи розпізнають лише знаки обмеження швидкості.

Аналогічною системою близькою за тематикою є програмне забезпечення для автомобілів від компаній BMW та Mercedes, які вперше з’явилися у цих авто в 2008 році.

Переваги:

- відкритий код;
- малі системні вимоги;

Недоліки:

- розпізнавання не всіх дорожніх знаків;
- менше оцінка розпізнавання, яка залежить від якості зображення;

Точність розпізнавання вищевказаної системи залежить від якості зображення, тому водіям не слід повністю довіряти таким системам, так як вони є лише додатковою опцією, яка може стати надійним помічником у деяких ситуаціях на дорозі.

Був розроблений програмний застосунок на мові програмування C++ з використанням бібліотеки комп'ютерного зору OpenCV, який надає наступні можливості:

- завантажувати зображення;
- аналізувати та виділяти знайдений дорожній знак;
- зберігати оброблене зображення;

Візуальну частину було розроблено на мові програмування C#.

Програмний застосунок потребує додатково встановленого програмного забезпечення у вігляді бібліотек комп’ютерного зору OpenCV.

Для коректної роботи програми має бути встановлена операційна система Windows.

Оновлення списку процесів та кількості потоків повинно відбуватися в окремому потоці, щоб переглядати інформацію в режимі реального часу.

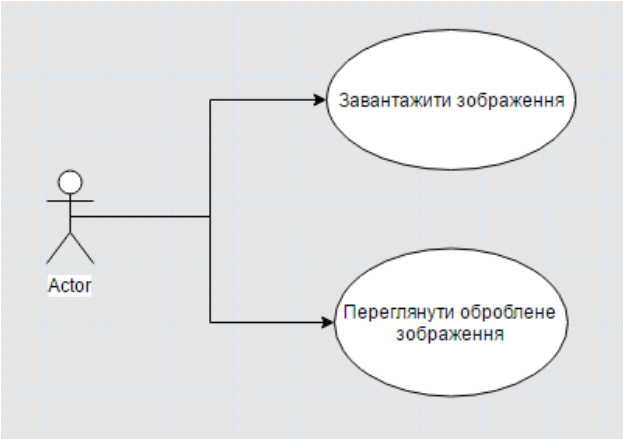


Рисунок 2.13 – Діаграма варіантів використання

2.9.1 Архітектура запропонованої системи

Розроблений програмний застосунок містить один клас, який відповідає за аналіз зображення.

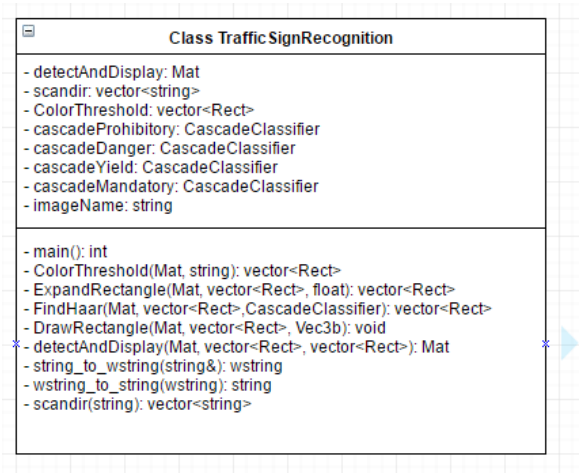


Рисунок 2.14 – Діаграма класів

Важливу роль відіграє метод ColorThreshold, який виконує бінарізацію зображення по заданому кольору(червоний або синій):

```
std::vector<cv::Rect>    ColorThreshold(cv::Mat    frame,
std::string color)
{
    cv::Mat        frameThreshold,        frameThreshold1,
frameThreshold2, frameHSV;
    cv::cvtColor(frame, frameHSV, CV_BGR2HSV);
    if (color == "red")
    {
        inRange(frameHSV,    cv::Scalar(140,    30,    5),
cv::Scalar(179, 255, 255), frameThreshold1);
        inRange(frameHSV,    cv::Scalar(0,    50,    5),
cv::Scalar(12, 255, 255), frameThreshold2);
        cv::bitwise_or(frameThreshold1,  frameThreshold2,
frameThreshold);
    }
    else
        if (color == "blue")
            inRange(frameHSV,  cv::Scalar(103,  100,  5),
cv::Scalar(130, 255, 255), frameThreshold);
        else
        {
            printf("--(!)Error,    can't    get    color
threshold\n");
        }
    //морфологическое открытие
    cv::erode(frameThreshold,              frameThreshold,
cv::getStructuringElement(cv::MORPH_ELLIPSE,  cv::Size(3,
3)));
    cv::dilate(frameThreshold,             frameThreshold,
cv::getStructuringElement(cv::MORPH_ELLIPSE,  cv::Size(3,
3)));
    //морфологическое закрытие
    cv::dilate(frameThreshold,             frameThreshold,
cv::getStructuringElement(cv::MORPH_ELLIPSE,  cv::Size(3,
3)));
    cv::erode(frameThreshold,              frameThreshold,
cv::getStructuringElement(cv::MORPH_ELLIPSE,  cv::Size(3,
3)));
```

```
    //Находим связные контура, выделяем их прямоугольниками и оставляем только достаточно большие прямоугольники, близкие к квадрату
    std::vector<std::vector<cv::Point> > contours;
    std::vector<cv::Rect> signs;
    cv::findContours(frameThreshold, contours, cv::RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE);
    for (size_t i = 0; i < contours.size(); ++i)
    {
        cv::Rect r = cv::boundingRect(contours[i]);
        if ((r.height / (float)r.width > 0.7) && (r.height / (float)r.width < 1.3) && (r.width > 20) && (r.height > 20))
            signs.push_back(r);
    }
    for (size_t i = 0; i < signs.size(); i++)
    {
        rectangle(frameThreshold, signs[i], cv::Scalar(255, 0, 255), 2, 8, 0);
    }
    return signs;
}
```

2.9.2 Вимоги до інтерфейсу

Програмний застосунок виконаний у вигляді Windows-інтерфейсу. Всі функції інтерфейсу повинні бути доступні при використанні комп’ютерної мишки та клавіатури.

На рисунку 2.15 показано, що користувач може обрати функції «Завантажити зображення» та «Зберегти зображення», а також рядок стану операції, у якому будуть появлятися повідомлення про успішне завершення операції або помилку під час виконання.

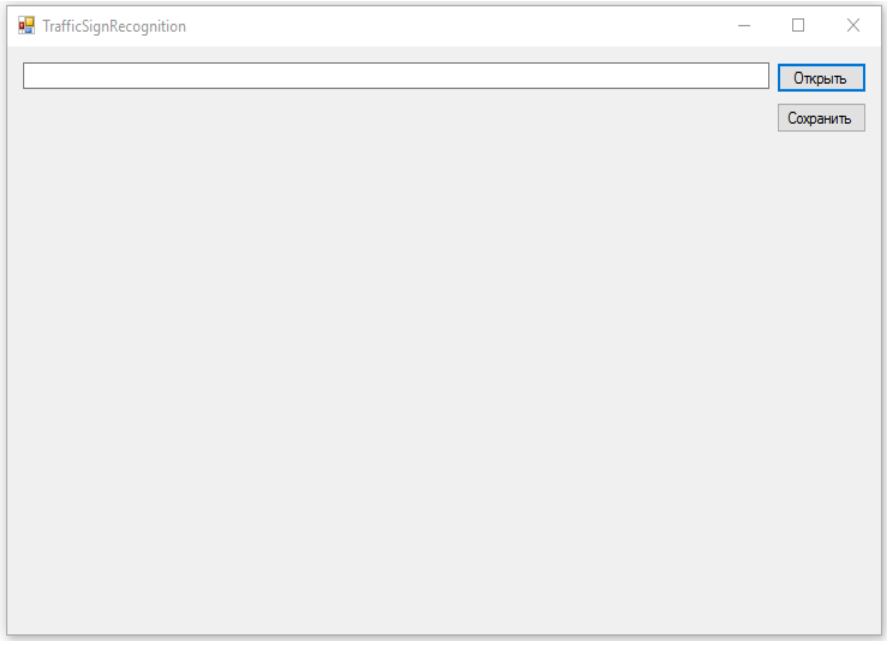


Рисунок 2.15 – Вигляд головного інтерфейсу програмного застосунку

Якщо користувач обере функцію «Завантажити зображення», то відкриється додаткове вікно, в якому потрібно буде вказати шлях до зображення, яку необхідно проаналізувати.

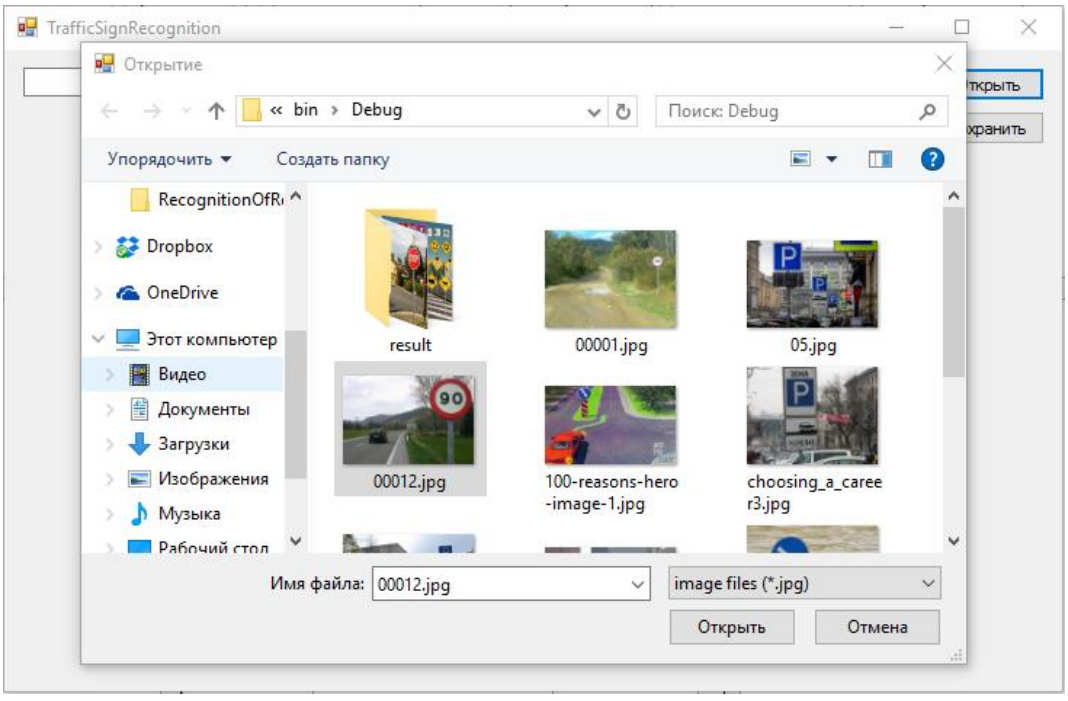


Рисунок 2.16 – Завантаження зображення



Рисунок 2.17 – Виділення розпізнаного дорожнього знаку

2.9.3 Проект програмної системи

Система складається з двох застосунків:

Застосунок у проводить усю роботу над аналізом зображення та збереженню результатів, виконує роль обробника винятків від системи, оновлює інформацію аналізу в окремому потоці та передає її до головного вікна, при цьому не потребує від користувача додаткових дій.

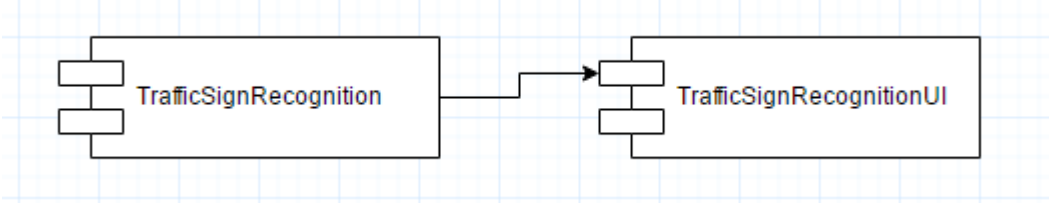


Рисунок 2.18 – Діаграма компонентів

Застосунок TrafficSignRecognitionUI відповідає за роботу з користувачем, а саме за обробку подій, оповіщення про успішність операцій у рядку стану.

2.9.4 Засоби реалізації та тестування

Для реалізації програмного застосунку було використано:

- мову програмування Visual C++ та C#;
- технологію Win Forms;
- середовище розробки Visual Studio;
- бібліотеку OpenCV.

Тестування проводилось по принципу «чорного ящику». Під час тесту перевірялись усі можливі збої в кожній функції, спроби завантажити зображення дуже великого об’єму, зображення різних форматів тощо.

Остання версія програмного застосунку пройшла усі зазначені тести, тому можна вважати, що реалізація є успішною. Основний лістинг програмного коду наведено у додатку А.

2.10 Розрахунок пристрою розпізнавання образів

Робота пристрою заснована на принципі функціонування перцептрона – математичній, комп'ютерній або логічній моделі сприйняття інформації. Заданий пристрій розпізнає цифри від 0 до 9, закодовані в символі, представленому на рисунку 2.19.

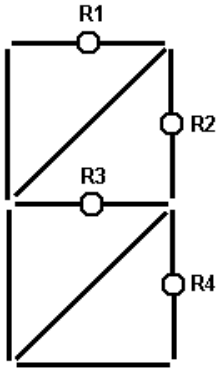


Рисунок 2.19 – Символ кодування інформації

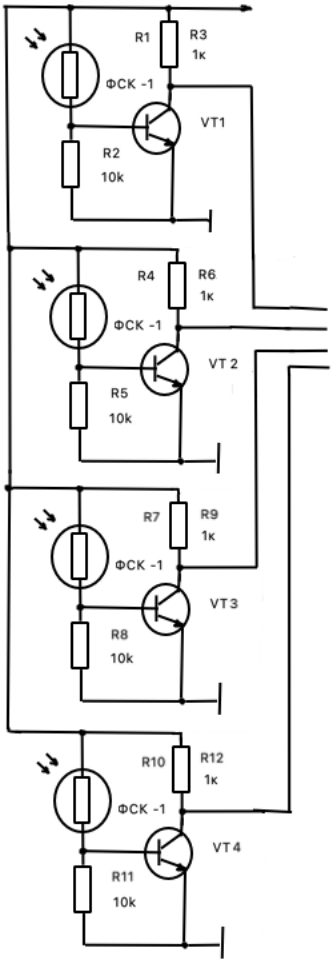


Рисунок 2.20 – Блок з фоторезисторів

Зняття інформації виробляється з фоторезисторів розташованих в сітці. Декодування інформації виробляється у відповідності таблиці. 2.2.

Таблиця 2.2 – Таблиця декодування інформації поштового символу

R1	R2	R3	R4	X
1	1	0	1	0
0	1	0	1	1
1	1	0	0	2
1	0	1	0	3
0	1	1	1	4
1	0	1	1	5
0	0	1	1	6
1	0	0	0	7
1	1	1	1	8
1	1	1	0	9

Схема декодування символів представлена на рисунку 2.21.

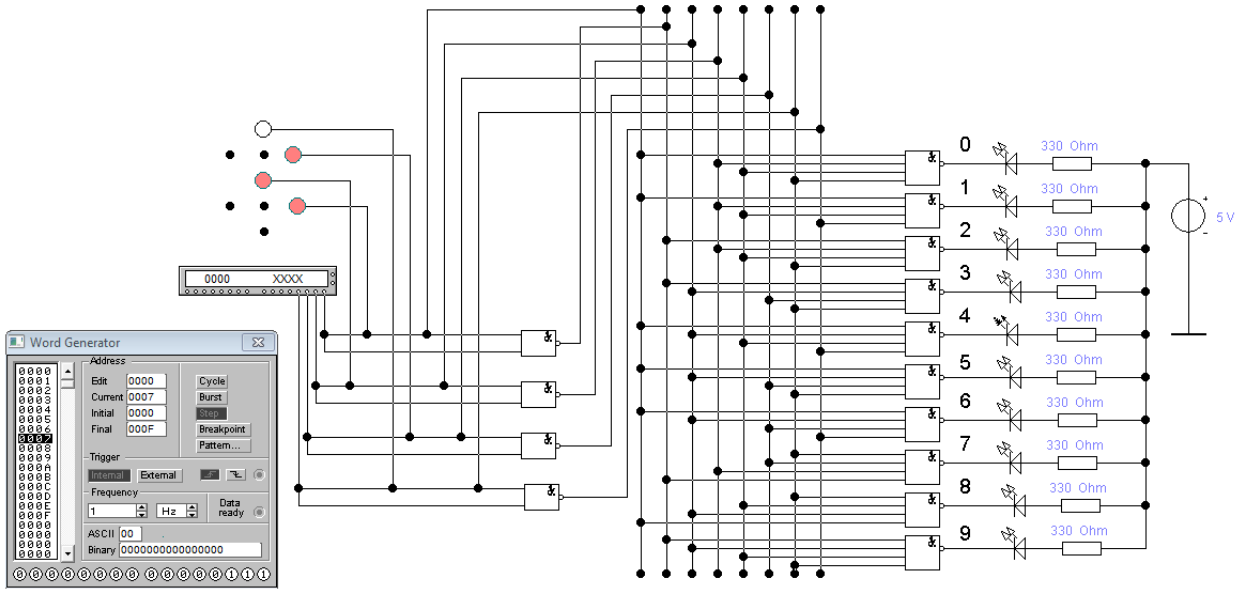


Рисунок 2.21 – Схема декодування символів в середовищі EWB

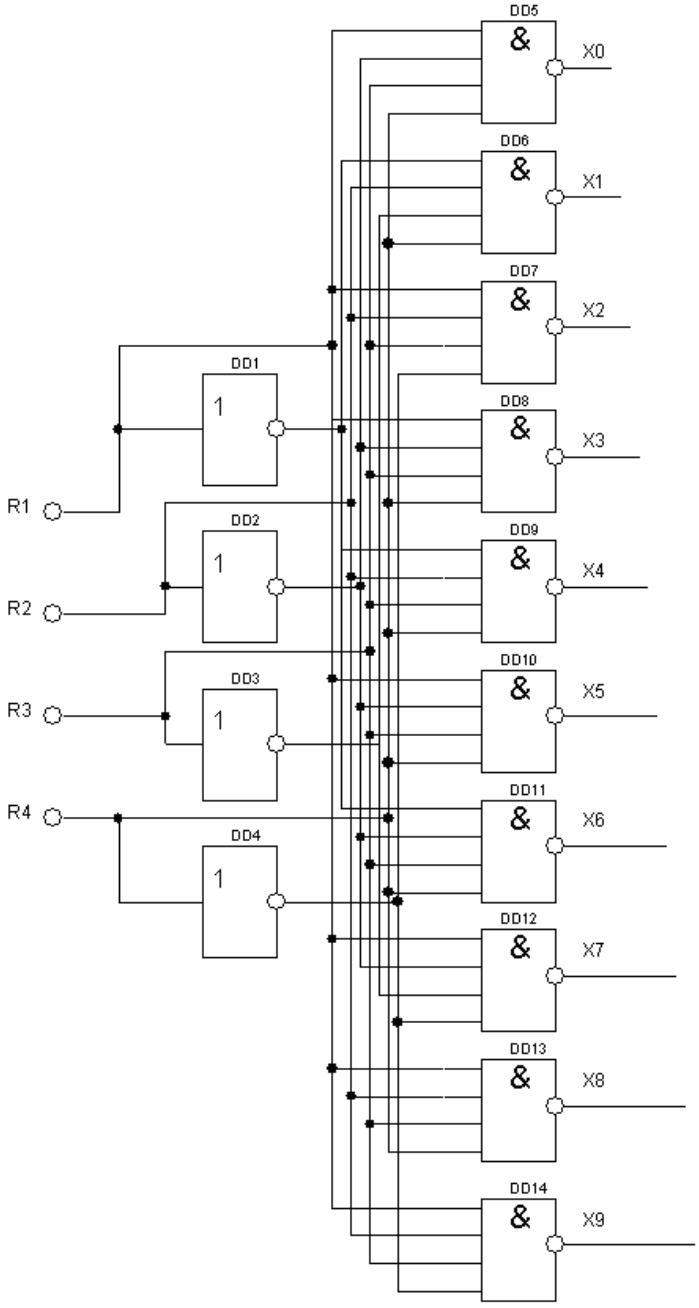


Рисунок 2.22 - Схема декодування символів

Декодовану інформацію необхідно представити у двійковому коді згідно з таблиці 2.3.

Таблиця 2.3 – Таблиця функціонування шифратора

X	Y4	Y3	Y2	Y1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Згідно таблиці функціонування шифратора побудуємо рівняння функціонування.

Y1=X1+X3+X5+X7+X9; Y2=X2+X3+X6+X7;

Y3=X4+X5+X6+X7; Y4=X8+X9;

Згідно з рівняннями структурна схема функціонування шифратора представлена на рисунку 2.23.

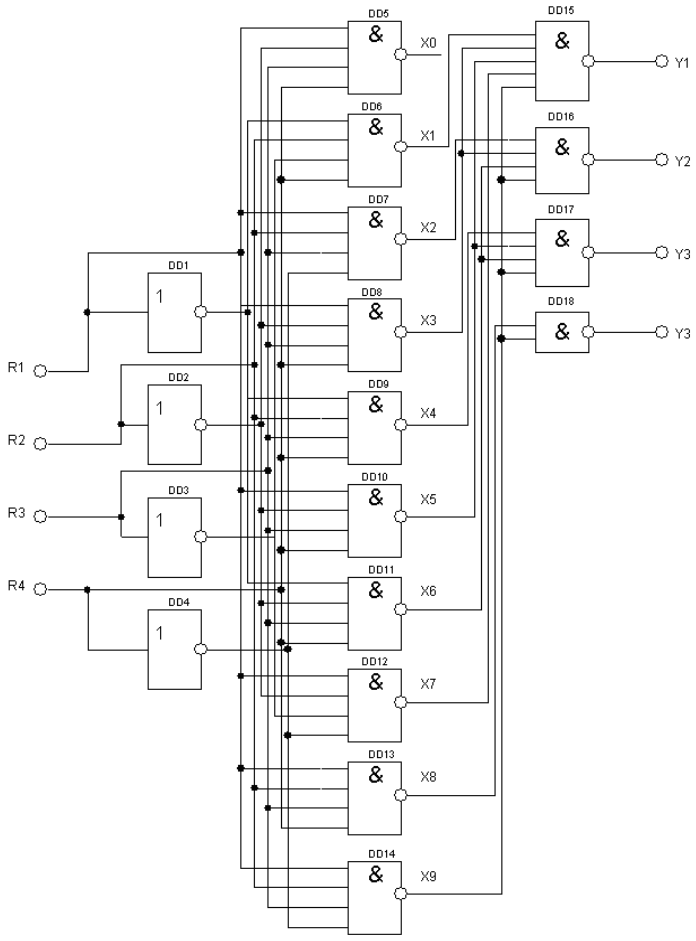


Рисунок 2.23 – Схема перетворювача коду поштових символів в двійковий код

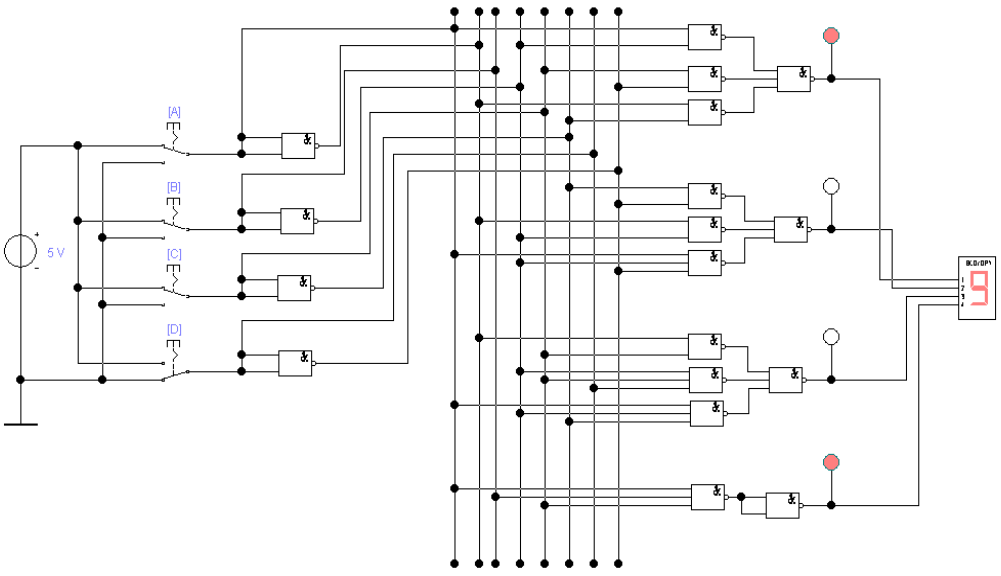


Рисунок 2.24 – Схема шифратора в середовищі EWB

Формувач сигналу готовності виконано на базі D – тригерів (мікросхема DD6 К155ИЕ5), тактованих генератором імпульсів. При декодуванні відповідної цифри пристрій фіксує код, формує сигнал «Готовий».

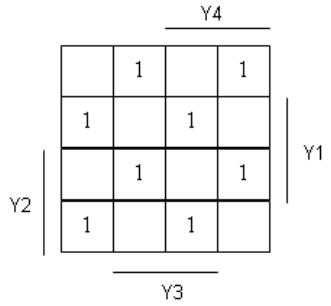
2.10.1 Схема контролю парності

Схема контролю парності застосовується для виявлення одиночних помилок, викликаних перешкодами в лінії зв'язку або в блоках пам'яті. Метод заснований на підрахунку числа одиниць в переданих в лінію або переданих в пам'ять на зберігання порції інформації, причому, якщо число одиниць парне - функція парності P(Parity) дорівнює нулю. Для чотирирозрядного двійкового числа таблиця станів схеми контролю парності представлена в таблиці. 2.4.

Таблиця 2.4 – Таблиця станів схеми контролю парності

Входи				Вихід
Y4	Y3	Y2	Y1	P
0	0	0	0	x
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
0	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Карта Карно:



Рівняння функціонування:

$$P = \overline{Y4}\,\overline{Y3}\,\overline{Y2}Y1 + \overline{Y4}\,\overline{Y3}Y2\overline{Y1} + \overline{Y4}Y3\overline{Y2}\,\overline{Y1} + \overline{Y4}Y3Y2Y1 +$$
$$+ Y4\overline{Y3}\,\overline{Y2}\,\overline{Y1} + Y4\overline{Y3}Y2Y1 + Y4Y3\overline{Y2}Y1 + Y4Y3Y2\overline{Y1} =$$
$$= \overline{Y4}\,\overline{Y3}(\overline{Y2}Y1 + Y2\overline{Y1}) + \overline{Y4}Y3(\overline{Y2}\,\overline{Y1} + Y2Y1) + Y4\overline{Y3}(\overline{Y2}\,\overline{Y1} +$$
$$+ Y2Y1) + Y4Y3(\overline{Y2}Y1 + Y2\overline{Y1}) = \overline{Y4}\,\overline{Y3}(Y2 \oplus Y1) +$$
$$+ \overline{Y4}Y3(\overline{Y2 \oplus Y1}) + Y4\overline{Y3}(\overline{Y2 \oplus Y1}) + Y4Y3(Y2 \oplus Y1) =$$
$$= (Y2 \oplus Y1)(\overline{Y4 \oplus Y3}) + (\overline{Y2 \oplus Y1})(Y4 \oplus Y3) = (Y2 \oplus Y1) \oplus (Y4 \oplus Y3)$$

Схемна реалізація і умовне позначення приведені на рисунку 2.25.

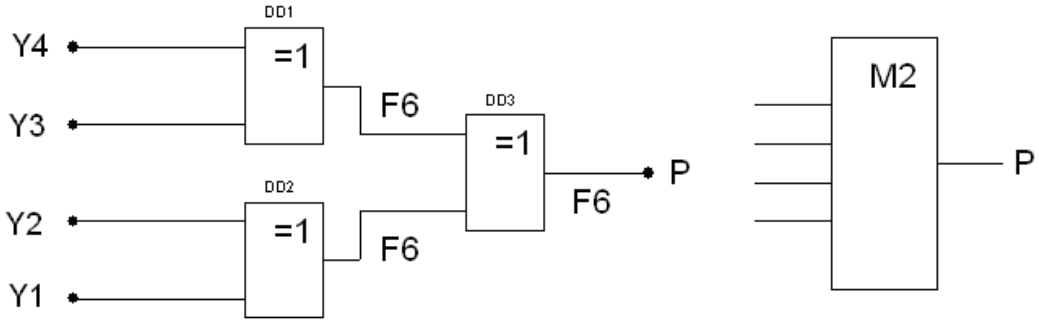


Рисунок 2.25 – Схема контролю парності

Символом М2 позначена операція - "сума по модулю два ".

По 4-дротяній лінії зв'язку (рис.2.26) передається паралельний двійковий код Y4Y3Y2Y1, а приймається код X4’X3’X2’X1’.

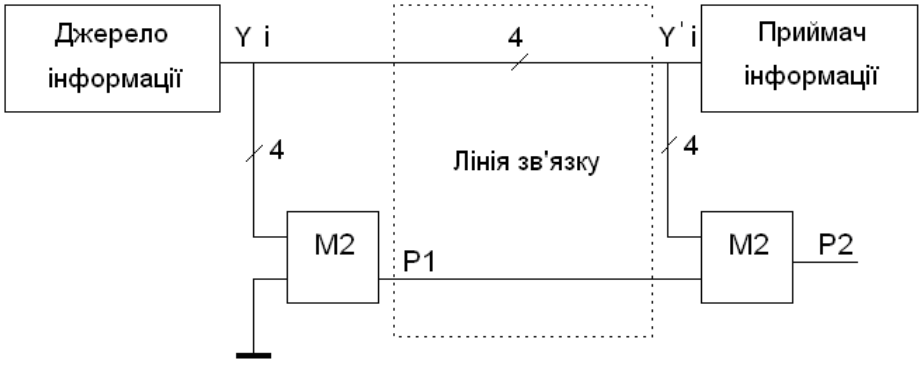


Рисунок 2.26 – Передача двійкового паралельного коду

Тоді величина P1 = (Y4⊕ Y3) ⊕ (Y2 ⊕ Y1). На приймальному кінці лінії зв'язку P2= (Y4'⊕ Y3’) ⊕ (Y2’ ⊕ Y1’).

Підставляючи в останню формулу вираження для P1 і групуючи змінні в однойменні пари, отримаємо: P2 = (Y4 ⊕ Y4’) ⊕ (Y3 ⊕ Y3’) ⊕ (Y2 ⊕ Y2’)⊕ (Y1 ⊕ Y1’). З останнього вираження виходить, що якщо передача пройшла без спотворень то $Y_i=Y'_i$ и $Y_i \oplus Y_i'=0$, а P2=0. При спотворенні одного і в загальному випадку непарного числа біт функція P2=1. Експериментальна схема контролю парності представлена на рисунку 2.27.

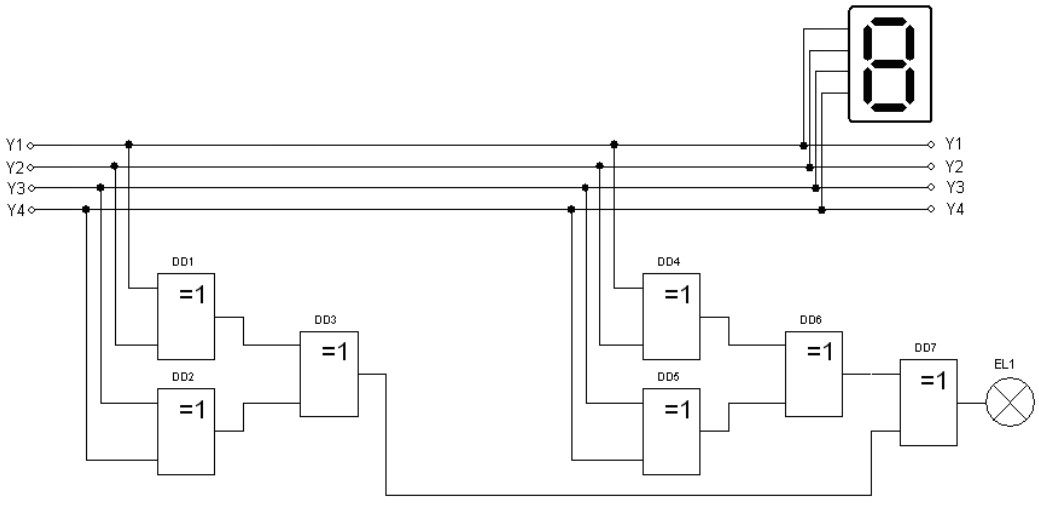


Рисунок 2.27 – Експериментальна схема контролю парності

2.10.2 Вибір схеми генератора імпульсів

Генератор імпульсів призначений для узгодження роботи всіх вузлів пристрою, що розробляється.

Широка зміна частоти імпульсів (близько 50 тисяч разів), що генеруються, забезпечує генератор, зібраний за схемою, що представлена на рисунку 2.28. Мінімальна частота імпульсів тут близько 0,25 Гц. Тривалість імпульсів регулюється резистором R1. Частоту дотримання можна визначити по формулі:

$$f = \frac{1}{2} R1C1, \tag{2.1}$$

Де f – частота, Гц; R1 – опір, Ом; C1 – ємність, Ф.

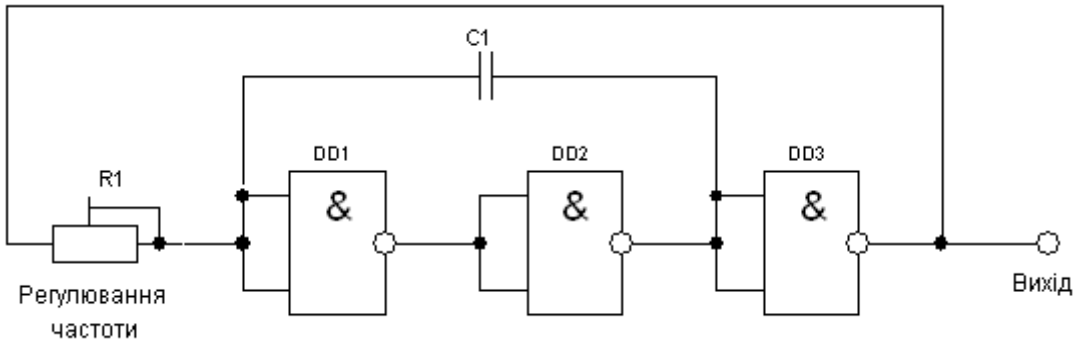


Рисунок 2.28 – Схема генератора імпульсів

2.10.3 Буферний регістр

Буферний регістр призначений для зберігання проміжної інформації і може бути реалізований за схемою регістра з паралельним введенням і паралельним виведенням інформації на тактованих D-тригерах (рис. 2.29).

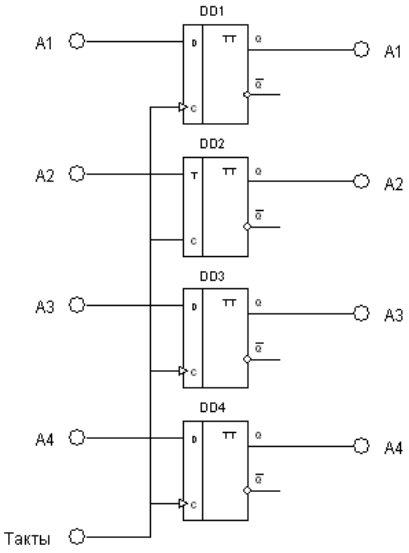


Рисунок 2.29 – Регістр з паралельним введенням і паралельним виведенням інформації на тактованих D-тригерах

Схема електрична принципова буферного регістру зібрана на мікросхемі К155ТМ2. (рис. 2.30).

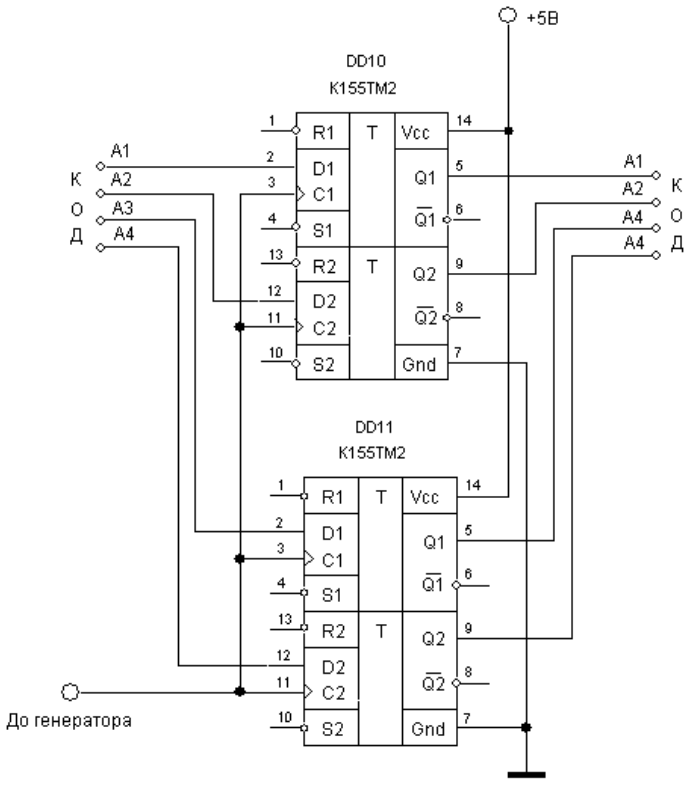


Рисунок 2.30 - Схема електрична принципова буферного регістру

2.10.4 Пристрій розпізнавання образів

Розроблений пристрій розпізнавання образів представлений на рисунку 2.31.

Декодування символьної інформації виконується схемою зібраній на мікросхемах DD1 – DD6 К155ЛА1. Шифратор DD7 – DD8 (К155ЛА1, К155ЛА2) перетворить десятковий код чисел 0 – 9 в їх двійковий еквівалент.

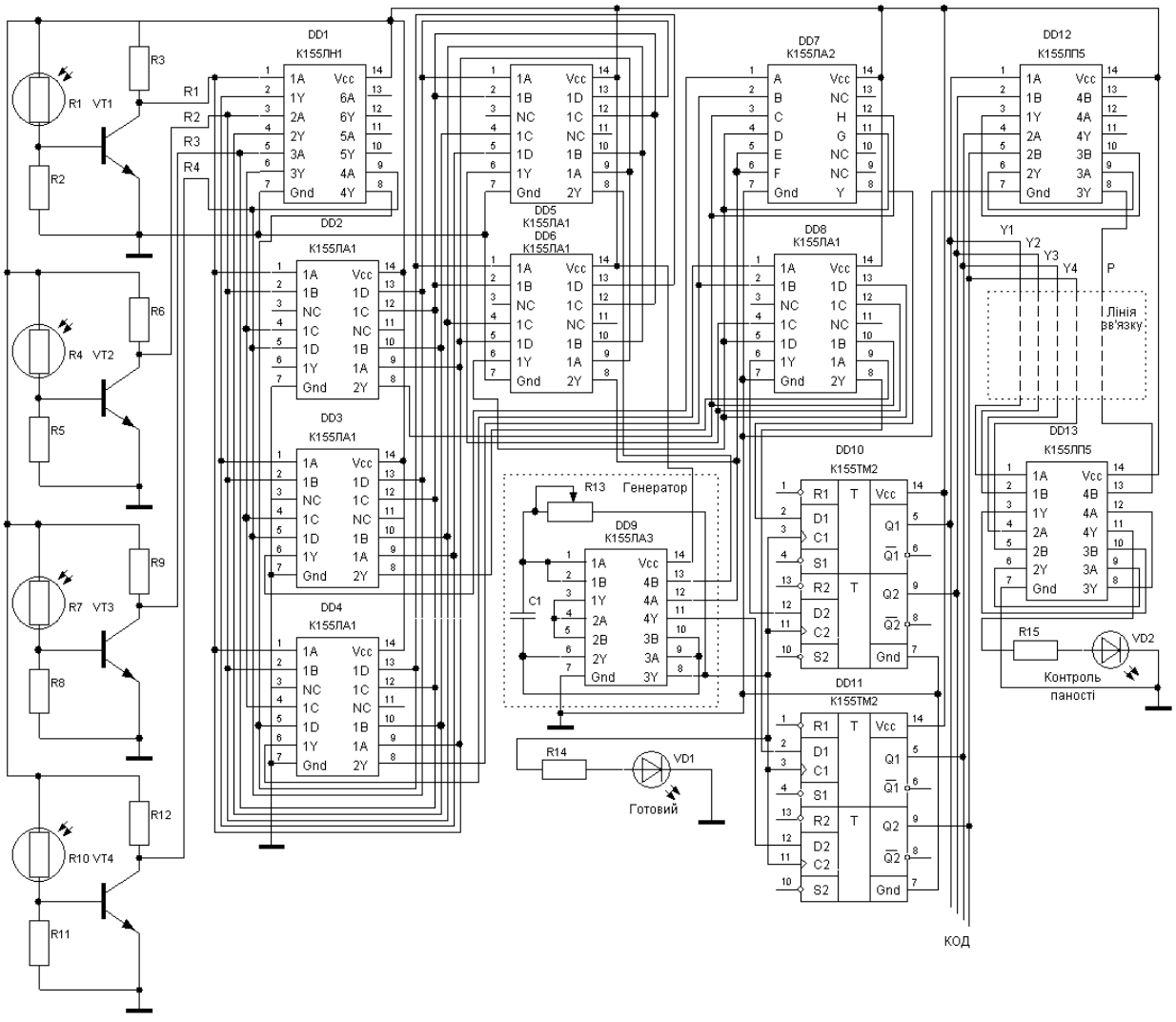


Рисунок 2.31 – Схема електрична принципова розробленого пристрою введення інформації

Буферний пристрій DD10 – DD11 (К155ТМ2) служить для зберігання отриманого коду і спрацьовує по тактових імпульсах, що поступають від генератора зібраного на мікросхемі DD9 К155ЛА3. Під час вступу коду на буферний пристрій виробляється сигнал «Готовий». Схема контролю парності DD12 – DD13 (К155ЛП5) перевіряє правильність інформації, що передається по лінії зв'язку.

2.10.5 Пристрій відображення інформації

Результати обчислень необхідно представити на двох семисегментних індикаторах (рис. 2.32).

Побудуємо пристрій перетворення двійкового коду чисел в код семисегментних індикаторів. Таблиця функціонування такого пристрою представлена в таблиці. 2.5.

Таблиця 2.5 - Таблиця функціонування перетворювача двійкового коду чисел в код семисегментних індикаторів

№	A	B	C	D	a'	b'	c'	d'	e'	f'	g'	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0
2	0	0	1	0	1	1	1	1	1	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1
4	0	1	0	0	1	1	1	1	1	1	0	0	1	1	0	0	1	1
5	0	1	0	1	1	1	1	1	1	1	0	1	0	1	1	0	1	1
6	0	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1
10	1	0	1	0	0	1	1	0	0	0	0	1	1	1	1	1	1	0
11	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0
12	1	1	0	0	0	1	1	0	0	0	0	1	1	0	1	1	0	1
13	1	1	0	1	0	1	1	0	0	0	0	1	1	1	1	0	0	1
14	1	1	1	0	0	1	1	0	0	0	0	0	1	1	0	0	1	1
15	1	1	1	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1

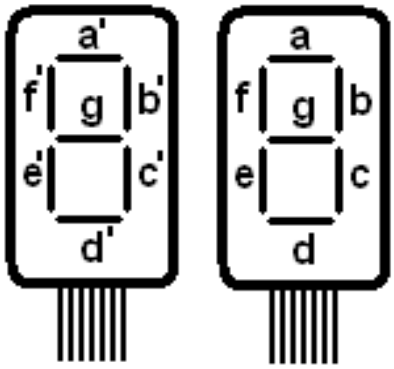
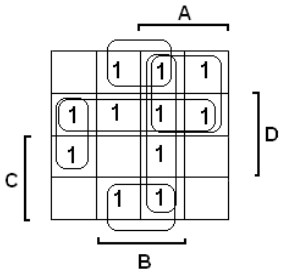


Рисунок 2.32 – Семисегментний індикатор

З таблиці виводимо рівняння функціонування пристрою перетворення двійкового коду чисел в код семисегментних індикаторів.

$$g = \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} +$$
$$+ A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + AB\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D} +$$
$$+ ABCD$$

Карта Карно:

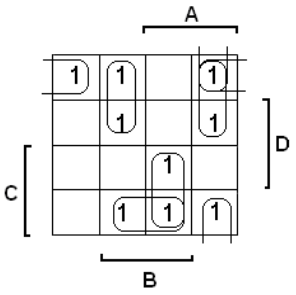


Спрощене рівняння:

$$g = \overline{C}D + A\overline{C} + AB + B\overline{D} + \overline{A}\overline{B}D$$

$$f = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + A\overline{B}\overline{C}\overline{D} +$$
$$+ A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + ABC\overline{D} + ABCD$$

Карта Карно:

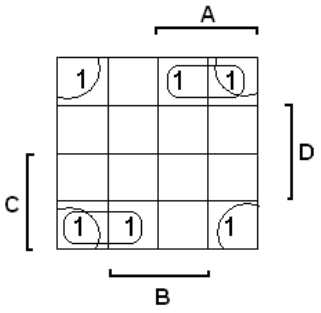


Спрощене рівняння:

$$f = \overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC + BC\overline{D} + A\overline{B}\overline{D}$$

$$e = \overline{A}B\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}BC\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} + AB\overline{C}\overline{D}$$

Карта Карно:

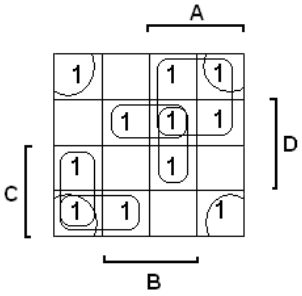


Спрощене рівняння:

$$e = \overline{B}\overline{D} + A\overline{C}\overline{D} + \overline{A}C\overline{D}$$

$$d = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} +$$
$$+ A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + AB\overline{C}\overline{D} + AB\overline{C}D +$$
$$+ ABCD$$

Карта Карно:

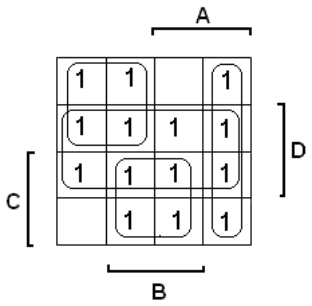


Спрощене рівняння:

$$d = A\overline{C} + \overline{B}\,\overline{D} + B\overline{C}D + ABD + \overline{A}\overline{B}C + \overline{A}C\overline{D}$$

$$c = \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D +$$
$$+ \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\overline{C}D +$$
$$+ A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}D + ABC\overline{D} + ABCD$$

Карта Карно:

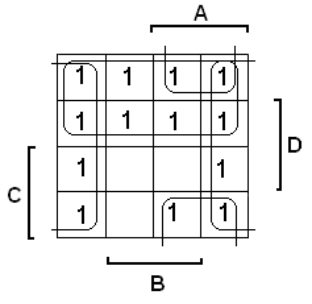


Спрощене рівняння:

$$c = D + \overline{A}\overline{C} + BC + A\overline{B}$$

$$b = \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\,\overline{D} +$$
$$+ \overline{A}BCD + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD$$
$$+ AB\overline{C}\,\overline{D} + AB\overline{C}D + ABC\overline{D}$$

Карта Карно

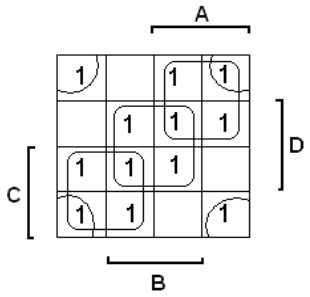


Спрощене рівняння:

$$b = \overline{C} + \overline{B} + A\overline{D}$$

$$a = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD$$
$$+ A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + AB\overline{C}\overline{D} + AB\overline{C}D + ABCD$$

Карта Карно:



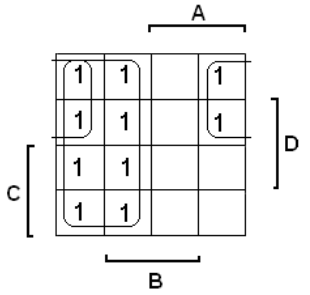
Спрощене рівняння:

$$a = A\overline{C} + BD + \overline{A}C + \overline{B}\overline{D}$$

$$g' = 0$$

$$f' = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} +$$
$$+ \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D$$

Карта Карно:



Спрощене рівняння:

$$f = \overline{A} + \overline{B}\overline{C}$$

$$e' = \overline{A} + \overline{B}\overline{C}$$

$d' = \overline{A} + \overline{B}\overline{C}$

$c' = 1$

$b' = 1$

$a' = \overline{A} + \overline{B}\overline{C}$

Мікросхему перетворювача двійкового коду чисел в код семи сегментного індикатора представлено на рисунку 2.33.

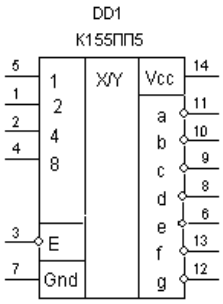


Рисунок 2.33 – Мікросхема перетворювача двійкового коду чисел в код семи сегментного індикатора

Функціональна схема перетворювача двійкового коду чисел в код семи сегментного індикатора представлено на рисунку 2.34.

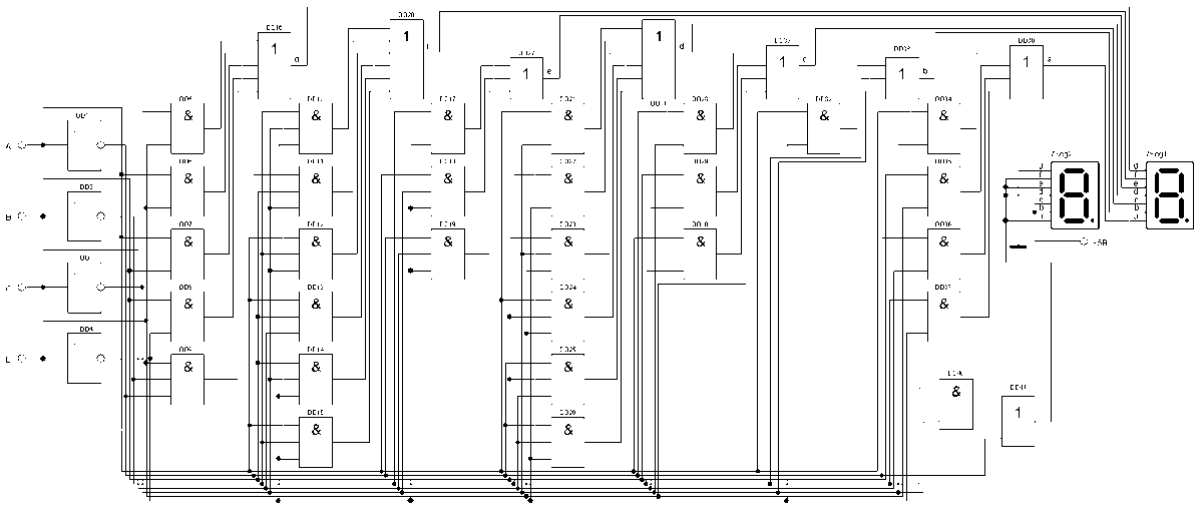


Рисунок 2.34 - Функціональна схема перетворювача двійкового коду чисел схеми розпізнавання образів в коди семи сегментного індикатору

Схема електрична принципова може бути виконана на мікросхемах ТТЛШ логіки.

Таким чином було проведено дослідження різних видів нейронних мереж, принципу їх роботи та реалізація різних підходів до їх використання.

Розроблено програмний застосунок TrafficSignRecognition, який працює на основі нейронної мережі з використанням мов програмування C++, та C# для розпізнавання дорожніх знаків. Під час роботи над застосунком досліджено можливості бібліотеки та систему комп'ютерного зору OpenCV .

В програмному середовищі Electronic Workbench розроблено та побудовано пристрій розпізнавання образів. Пристрій складається з двох електричних схем: схеми електричної принципової для введення інформації, та функціональної схеми розпізнавання образів в коди семи сегментного індикатору.

Запропоновано схему апаратної реалізації нейрона, яка побудована на основі ПЛІС - матричних структур, що забезпечують паралельну обробку інформації і володіють високим ступенем інтеграції, масштабованістю, розвиненими засобами проектування, реалізації виконання алгоритмів навчання.

3 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

Актуальність досліджень підтверджується масою різних застосувань нейронних мереж. Це автоматизація процесів розпізнавання образів, адаптивне управління, апроксимація функціоналів, прогнозування, створення експертних систем, організація асоціативної пам'яті і багато інших додатків.

Дипломна робота «Дослідження схеми апаратної реалізації нейронної мережі» присвячена дослідженню та обробці функціоналу за різними параметрами нейронних мереж.

3.1 Порівняльний аналіз штучних нейронних мереж за критеріями

Штучні нейронні мережі, що засновані на різних функціональних процесах: Багатошаровий персептрон, мережа Елмана, , Згорткові мережі, карти Кохонена, кожна має свої позитивні та негативні якості.

Тому для вибору кращої з цих мереж проаналізуємо ієрархії чотирьох варіантів (табл. 3.1) враховуючи шкалу відносної важливості (табл.3.2) [17]

Таблиця 3.1 – Варіанти видів штучних нейронних мереж

Технологія	Короткий опис (види НМ)
A	Персептрон
B	Мережа Елмана
C	Згорткові мережі
D	карти Кохонена

Таблиця 3.2 – Шкала відносної важливості

Інтенсивність відносної важливості	Визначення
1	рівна важливість
3	помірна перевага
5	сильна перевага
7	значна перевага
9	дуже сильна перевага
2,4,6,8	проміжні судження

Вибір робимо за критеріями, наведеними в таблиці 3.3, встановлюємо відносну вагу кожного критерію на основі матриці попарних порівнянь для обраних критеріїв.

Таблиця 3.3 – Попарне порівняння критеріїв

Критерій	1	2	3	4	5	6	$\sqrt[6]{\prod_{i=1}^{6}\omega_i}$	X_i
1. Швидкість	1	1	1/3	1/7	7	3	1,003	0,127
2. Вартість	1	1	5	7	5	7	3,255	0,412
3. Надійність	3	1/5	1	1/5	3	1/5	0,646	0,082
4. Експлуата-ційні режими	7	1/7	5	1	5	1	1,706	0,216
5. Безпека	1/7	1/5	1/3	1/5	1	1/5	0,270	0,034
6. Навчання	1/3	1/7	5	1	5	1	1,027	0,129
∑							7,907	1,00

Далі аналогічно складаємо 6 матриць попарних порівнянь альтернатив стосовно кожного критерію (таблиці 3.4, 3.5, 3.6, 3.7, 3.8, 3.9). Оскільки тепер порівнюються 4 технології по одному критерію , то i = 1, 2, 3, 4;

$$X_i = \frac{\sqrt[4]{\prod_{i=1}^{4} \omega_i}}{\sum_{i=1}^{4} \sqrt[4]{\prod_{i=1}^{4} \omega_i}}; \sum \text{ - сума по стовпці } \sqrt[4]{\prod_{i=1}^{4} \omega_i}.$$

Таблиця 3.4 – Порівняння альтернатив стосовно критерію «Швидкість»

Технологія	A	B	C	D	$\sqrt[4]{\prod_{i=1}^{4} \omega_i}$	X_i
A	1	5	7	5	3,64	0,47
B	1/5	1	1/3	1	1,26	0,16
C	1/7	3	1	3	1,63	0,21
D	1/5	1	1/3	1	1,26	0,16
∑					7,79	1,00

Таблиця 3.5 – Порівняння альтернатив стосовно критерію «Вартість»

Технологія	A	B	C	D	$\sqrt[4]{\prod_{i=1}^{4} \omega_i}$	X_i
A	1	7	5	7	2,11	0,34
B	1/7	1	1	3	1,50	0,23
C	1/5	1	1	3	1,51	0,24
D	1/7	1/3	1/3	1	1,16	0,19
∑					6,28	1,00

Таблиця 3.6 – Порівняння альтернатив стосовно критерію «Надійність»

Технологія	A	B	C	D	$\sqrt[4]{\prod_{i=1}^{4}\omega_i}$	X_i
A	1	7	5	5	2,06	0,33
B	1/7	1	1	3	1,50	0,25
C	1/5	1	1	1	1,34	0,22
D	1/5	1/3	1	1	1,26	0,20
∑					6,16	1,00

Таблиця 3.7 – Порівняння альтернатив стосовно критерію «Експлуатаційні режими»

Технологія	A	B	C	D	$\sqrt[4]{\prod_{i=1}^{4}\omega_i}$	X_i
A	1	3	5	7	2,00	0,31
B	1/3	1	3	5	1,75	0,27
C	1/5	1/3	1	5	1,59	0,25
D	1/7	1/5	1/5	1	1,11	0,17
∑					6,45	1,00

Таблиця 3.8 – Порівняння альтернатив стосовно критерію «Безпека»

Технологія	A	B	C	D	$\sqrt[4]{\prod_{i=1}^{4}\omega_i}$	X_i
A	1	3	7	3	1,93	0,30
B	1/3	1	5	1	1,65	0,26
C	1/7	1/5	1	5	1,59	0,25
D	1/3	1	1/5	1	1,26	0,19
∑					6,43	1,00

Таблиця 3.9 – Порівняння альтернатив стосовно критерію «Навчання»

Технологія	A	B	C	D	$\sqrt[4]{\prod_{i=1}^{4}\omega_i}$	X_i
A	1	1/3	1/3	1/5	1,17	0,19
B	3	1	3	3	1,78	0,29
C	3	1/3	1	1	1,51	0,25
D	5	1/3	1	1	1,65	0,27
∑					6,11	1,00

Глобальний пріоритет для кожної альтернативи обчислюється як сума добутків кожного локального пріоритету на його ваговий коефіцієнт.

Таблиця 3.10 – Глобальний пріоритет для кожної альтернативи

Пріоритети	№1	№2	№3	№4	№5	№6	Глобальний
Вага	0,127	0,412	0,082	0,216	0,034	0,129	
Персептрон	0,47	0,34	0,33	0,31	0,30	0,19	0,328
Мережа Елмана	0,16	0,23	0,25	0,27	0,26	0,29	0,240
Згорткові мережі	0,21	0,24	0,22	0,25	0,25	0,25	0,238
карти Кохонена	0,16	0,19	0,20	0,17	0,19	0,27	0,193

З порівняння глобальних пріоритетів різних видів нейронних мереж (табл. 3.10) видно, що найбільшим є пріоритет у Персептрона.

Висновки: За допомогою методу аналізу ієрархій проведене порівняння чотирьох видів нейронних мереж за наступними критеріями: 1) Швидкість ; 2) Вартість; 3) Надійність; 4) Експлуатаційні режими; 5) Безпека; 6) Навчання. Найбільший локальний пріоритет у критерію «Швидкість» (дивись табл. 3.3). За даними таблиць глобальний пріоритет за багатьма критеріями є найвищим для Персептрона. Тому перевага віддається цій мережі, і саме вона й буде використовуватися у проектуванні завдання до розрахунку пристрою розпізнавання образів.

4 ОХОРОНА ПРАЦІ ТА ТЕХНОГЕННА БЕЗПЕКА

4.1 Характеристика потенційно небезпечних та шкідливих виробничих факторів

На робочому місці інженер-електронщик піддається впливу наступних несприятливих умов:

— недостатнє освітлення;

— шум від працюючих машин (комп’ютерів, робочих стендів). Джерелом шуму є робочі стенди. Шум діє на робітника протягом 70% робочого часу;

— електромагнітне випромінювання, джерелом якого є персональні комп'ютери з системними блоками і моніторами;

— виділення надлишків теплоти;

— підвищена запиленість.

Розвитку стомлюваності на робочому місці сприяють такі чинники:

— характер протікання праці. Трудовий процес організований таким чином, що інженер-електронщик змушений з перших хвилин робочого дня вирішувати найбільш складні і трудомісткі задачі, у той час як у перші хвилини роботи функціональна рухливість нервових клітин мозку низька. Важливе значення має чергування праці та відпочинку, зміна одних форм роботи іншими.

— неправильна ергономічна організація робочого місця, нераціональні зони розміщення обладнання по висоті від підлоги, по фронту від осі симетрії;

Джерела небезпечних та шкідливих чинників на інженера – електронщика:

— рівень шуму в приміщенні протягом робочого дня непостійний. При цьому протягом однієї години діє шум з рівнем звуку 83 дБА, протягом наступних двох годин з рівнем звуку 86 дБА, останній час 81 дБА при тому що норматив повинен бути не більше 70 дБА. Значить, робоче місце по показнику рівня шуму відноситься до класу умов праці 3.2 – шкідливий.

— основним джерелом електромагнітного випромінювання приміщення є персональні комп'ютери з системними блоками і моніторами.

— іонізація повітря приміщення. Рівні позитивних і негативних аероіонів в повітрі приміщень, де розташовані ПЕОМ, повинні відповідати санітарно-епідеміологічним нормативам, що діють.

Так як у приміщенні присутнє лабораторне устаткування, комп’ютери, шафи та робоча документація, то спостерігається деяка запиленість. За нормою запиленість в приміщенні не повинна перевищувати 4 мг/м³[18].

4.2. Заходи з поліпшення умов праці, виробнича санітарія

Для запобігання дещо підвищеної запиленості рекомендується встановлювати витяжні пристрої, а також проводити вологе прибирання приміщення лабораторного приміщення.

Раціональне колірне оформлення приміщення направлено на поліпшення санітарно-гігієнічних умов праці, підвищення її продуктивності та безпеки. Забарвлення приміщень впливає на нервову систему людини, його настрій і в кінцевому рахунку на продуктивність праці. Основні виробничі приміщення доцільно офарблювати відповідно до кольору технічних засобів. Освітлення приміщення і устаткування має бути м'яким, без блиску [18].

Температура в приміщеннях є одним з провідних чинників, що визначають метеорологічні умови виробничого середовища. Високі температури надають негативну дію на здоров'я людини. Робота в умовах

високої температури супроводжується інтенсивним потовиділенням, що приводить до обезводнення організму, втрати мінеральних солей і водорозчинних вітамінів, викликає серйозні і стійкі зміни в діяльності серцево-судинної системи, збільшує частоту дихання.

При низькій температурі висока відносна вологість збільшує тепловтрати організму в результаті інтенсивного поглинання водяними парами енергії випромінювання людини. Це веде до переохолодження організму – гіпотермії. Низька вологість викликає пересихання слизистих оболонок дихальних доріг того, що працює.

В даному приміщенні допустима температура повітря в теплий період для роботи складає 22-24ºС (в холодний період 21-23ºС), відносна вологість 40-60 %, швидкість руху повітря 0,1-0,2 м/с.

Фактичні параметри: в теплий період температура – 23-25 ºС, відносна вологість 45-60%, швидкість руху повітря 0,15-0,2 м/с, категорія робіт – 1а, в холодний період – температура 18-24^{0}С, відносна вологість 40-60 %, швидкість руху повітря 0,15-0,2 м/с, категорія робіт 1б[19], що відповідає нормам.

У приміщеннях, обладнаних ПЕВМ, повинна проводиться щоденне, вологе прибирання і систематичне провітрювання після кожної години роботи на ПЕВМ.

Світло є природною умовою життя людини, необхідною для збереження здоров'я і високої продуктивності праці, і заснованим на роботі зорового аналізатора. Недостатнє освітлення робочого місця утрудняє тривалу роботу, викликає підвищене стомлення і сприяє розвитку короткозорості. Дуже низькі рівні освітленості викликають апатію, сонливість, а в деяких випадках сприяють розвитку відчуття тривоги.

За правилами, світло при роботі з комп’ютером повинно падати зліва. Крім того, крісло слід відрегулювати так, щоб очі були на одному рівні з центром монітора. Саме очі найбільш страждають при роботі з комп’ютером. Виявляється, коли довго дивишся на екран, перестаєш

моргати. Тому очі червоніють, сльозяться, а значить, знижується зір. Невелика відстань до екрану, дрібний шрифт, мерехтіння, різне освітлення призводять, в остаточному підсумку, до короткозорості.

З метою зниження мерехтіння екрану рекомендується встановлювати частоту кадрів не менше 75 Гц для ЕЛТ-моніторів, та для ЖК-моніторів достатньою є мінімальна частота кадрів у 60 Гц.

Екран відеомонітора повинен знаходитися від очей на оптимальній відстані 700мм, але не ближче 500 мм. Перед початком роботи необхідно переконатися, що монітори комп'ютера мають антіблоковое покриття з коефіцієнтом відбиття не більше 0,5. Покриття повинне також забезпечувати зняття електростатичного заряду з поверхні екрана, іскріння і накопичення пилу. Корпус монітора повинен забезпечувати захист від іонізуючих та неіонізуючих випромінювань[18].

Для зниження рівня шуму стіни і стеля приміщень, де встановлені комп'ютери, можуть бути фанеровані звукопоглинальними матеріалами, а також застосування різних звукопоглинальних пристроїв. Зниження шуму в джерелі випромінювання можна забезпечити застосуванням м'яких килимків з синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, прокладки з м'якої гуми завтовшки 6-8 мм.

У зв'язку з усіма небезпеками досить чітко регламентовані розміри столу і стільця для роботи з комп'ютером. Адже "закам'яніла" постава шкідливо впливає на скелетно-м'язову систему. Стіл повинен бути просторим, із спеціальною підставкою для ніг, а робочий стілець – мати відрегульовану висоту, певний кут нахилу сидіння і спинки. Висота робочої поверхні столу повинна регулюватися а межах 680-800 мм, при відсутності такої можливості висота робочої поверхні столу Робота за комп'ютером повинна бути 725 мм.

Робота за комп'ютером без регламентованої перерви не повинна перевищувати двох годин. Під час регламентованих перерв з метою зниження нервово-емоційної напруги, стомлення зорового аналізатора, усунення впливу

гіподинамії і гіпокінезії, запобігання розвитку захворювань спини треба виконувати комплекси гімнастичних вправ.

Клавіатуру слід розташовувати на поверхні столу на відстані 100-300 мм від краю, зверненого до користувача.

Необхідно перевірити робоче положення комп'ютера та відстань між стіною з віконними прорізами і столом (має бути не менше 0,8 м). При невеликій кількості робочих місць бажано розташовувати столи біля протилежної стіни щодо віконних прорізів. Також відстань між робочими столами повинна бути не менш ніж 1,2 м. Не допускається знаходження другого робочого місця з боку задньої стінки комп’ютера.

Для забезпечення оптимальної працездатності і збереження здоров’я, протягом часу роботи з комп’ютером, повинні встановлюватися регламентовані перерви. Під час регламентованної перерви, з метою зниження нервово-емоційного напруження, стомлення зорового аналізатора, усунення впливу гіподинамії та гіпокінезії та запобігання розвитку познотопіческого втоми, доцільно виконувати комплекси вправ.

Щоб відновити загальний баланс атмосфери у приміщенні необхідно, щоб до робочого місця за комп’ютером проникало свіже повітря.

Таким чином, для безпечної роботи людини, лабораторне приміщення, де знаходиться персональний комп’ютер, повинно відповідати діючим санітарно-епідеміологічним нормам [18].

4.3. Електробезпека

Електричні установки, до яких відноситься практично все обладнання ПЕОМ, представляють для людини велику потенційну небезпеку, тому що в процесі експлуатації або проведенні профілактичних робіт людина може торкнутися частин, що знаходяться під напругою. Специфічна небезпека

електроустановок: струмоведучі провідники, корпуси стійок ПЕОМ і іншого устаткування, що опинився під напругою в результаті пошкодження (пробою) ізоляції, не подають будь-яких сигналів, які попереджають людину про небезпеку[20].

У лабораторному приміщенні розрядні струми статичної електрики частіше за все виникають завдяки дотику до будь-якого з елементів ПЕОМ. Такі розряди небезпеки для людини не представляють, але крім неприємних відчуттів вони можуть привести до виходу з ладу ПЕОМ. Для зниження величини виникаючих зарядів статичної електрики в приміщенні покриття технологічних підлог слід виконувати з одношарового полівінілхлоридного антистатичного лінолеуму. Іншим методом захисту є нейтралізація заряду статичного електрики іонізованим газом. У промисловості широко застосовуються радіоактивні нітралізатори. До загальних заходів захисту від статичної електрики в лабораторному приміщенні можна віднести загальні і місцеве зволоження повітря[20].

Щоб захистити людину від ураження електричним струмом, захисне заземлення має задовольняти ряду вимог, викладених у «Правилах безпеки електроустановок для споживачів» (ПЕУС), НПАОП 40.1-1.21-98[21]. Електробезпека. Захисне заземлення. Занулення». Ці вимоги залежать від напруги електроустановок та потужності джерела живлення. В електроустановках змінного струму напругою до 1000 В у мережі з ізольованою нетралю або ізольованим виводом джерела однофазного струму опір заземлювального пристрою не повинен перевищувати 4 Ом [20].

4.4. Пожежна та техногенна безпека

В лабораторії найбільш небезпечна ситуація виникає при пожежах.

Пожежі пов'язані з великими матеріальними втратами. Характерна особливість лабораторних приміщень – невеликі робочі площі. Як відомо пожежа може виникнути при взаємодії горючих речовин, окислення і джерел запалювання. У даному приміщеннях присутні всі три основні чинника, необхідні для виникнення пожежі.

Горючими компонентами в кімнаті є: будівельні матеріали для акустичної і естетичної обробки приміщень, перегородки, двері, підлоги, перфокарти і перфострічки, ізоляція кабелів та ін.

Протипожежний захист – це комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобігання пожежі, обмеження її розповсюдження, а також на створення умов для успішного гасіння пожежі.

Джерелами запалювання в лабораторному приміщенні можуть бути електронні схеми від ПЕОМ, прилади, застосовувані для технічного обслуговування, пристрої електроживлення, кондиціонування повітря, де в результаті різних порушень утворюються перегріті елементи, електричні іскри і дуги, здатні викликати загоряння горючих матеріалів.

В сучасних ПЕОМ дуже висока щільність розміщення елементів електронних схем. У безпосередній близькості один від одного розташовуються сполучні дроти, кабелі. При протіканні по них електричного струму виділяється значна кількість теплоти. При цьому можливо оплавлення ізоляції. Для відводу надлишкової теплоти відП ЕОМ служать системи вентиляції та кондиціонування повітря. При постійній дії ці системи представляють собою додаткову пожежну небезпеку.

Однією з найбільш важливих завдань пожежного захисту є захист будівельних приміщень від руйнувань та забезпечення їх достатньої міцності в умовах впливу високих температур при пожежі. Враховуючи високу вартість електронного обладнання, а також категорію його пожежної небезпеки, будівлі

для лабораторного приміщення і частини будівлі іншого призначення, в яких передбачено розміщення ПЕОМ повинні бути 1 та 2 ступеня вогнестійкості.

Для виготовлення будівельних конструкцій використовуються, як правило, цегла, залізобетон, скло, метал та інші негорючі матеріали. Застосування дерева повинно бути обмежено, а в разі використання необхідно просочувати його вогнезахисними складами.

Для гасіння пожеж на початкових стадіях широко застосовуються вогнегасники. В лабораторних приміщеннях застосовуються головним чином вуглекислотні вогнегасники (ВВК3-5), перевагою яких є висока ефективність гасіння пожежі, збереження електронного устаткування. Діелектричні властивості вуглекислого газу, що дозволяє використовувати ці вогнегасники навіть у тому випадку, коли не вдається знеструмити електроустановку відразу. Вогнегасник (ВВК3-5) переносний з місткістю балона 5 літрів (3,5 кілограма), призначений для гасіння електроустановок, що знаходяться під напругою не більш 10кВ. В даному приміщенні знаходиться один такий вогнегасник [19].

При виявленні пожежі кожен працівник зобов'язаний:

— Негайно викликати пожежну команду за телефоном 101;

— Вимкнути світлову та силову електромережу;

— Видалити з вогнища пожежі джерело запалення;

— Зменшити доступ повітря в зону горіння, ізолюючи вогнище вогнезахисної тканиною.

При виникненні пожежі на електроустановці потрібно негайно знеструмити обладнання і застосувати неелектропровідних вогнегасні засоби (пісок, порошкові вогнегасники, вогнезахисну тканина).

При гасінні одягу на людині потрібно діяти з урахуванням конкретних обставин. Не можна допускати різких рухів і бігати в палаючому одязі, полум'я від цього тільки посилюється. При видаленні тліючих залишків одягу провести екстрені заходи по охолодженню ділянок тіла; не менше 10 хвилин витримати під струменем води, накласти товстий

шар тканини, змоченою водою. До лікарської допомоги поверхню опіку ізолювати асептичною пов'язкою або чистою м'якою тканиною, змоченою спиртом.

При виникненні пожежі або іншої надзвичайної ситуації необхідно здійснити негайну евакуацію людей з об’єкту. Евакуація людей здійснюється по шляхах евакуації через евакуаційні виходи. Евакуаційні шляхи і виходи утримуються вільними. Двері на шляху евакуації відчиняються в напрямку виходу з будівлі. Шляхи евакуації постійно освітлюються електричним світлом. Евакуаційні виходи позначені світловими покажчиками з написом "Вихід".

4.5 Дослідження засобів захисту від ураження струмом при дотику до фазового дроту

Визначимо напругу дотику та силу струму, що пройде крізь тіло людини, при доторканні до фазового дроту та нульового дроту мережі одночасно. Напруга фазового дроту змінна, U=220В. Частота струму промислова, f=50 Гц. Приймемо власний опір дротів як r=2 Ом, індуктивність мережі L=0,5 Гн, ємність мережі С=1,5 мкФ. Схема дотику представлена на рисунку 3.2. Опір тіла людини Rh=1 кОм.

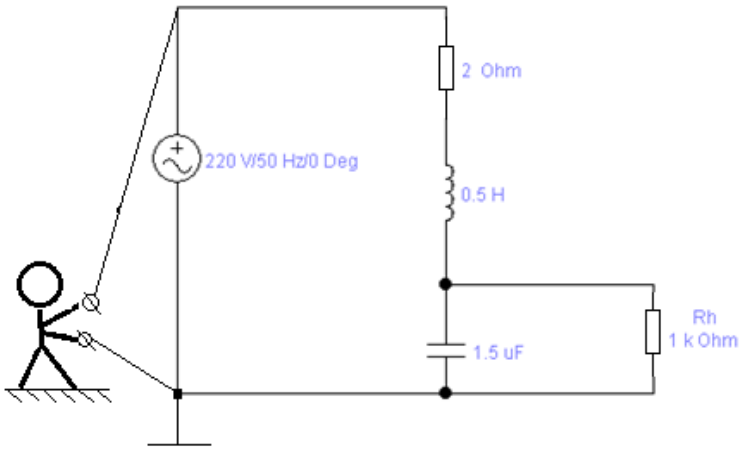


Рисунок 4.1 – Схема дотику

Визначимо індуктивний та ємнісний опір:

$$X_L = wL = 2\pi fL = 2 \cdot 3{,}14 \cdot 50 \cdot 0{,}5 = 157 \text{ Ом}$$

де w – циклічна частота;

$$X_C = \frac{1}{wC} = \frac{1}{2\pi fC} = \frac{1}{2 \cdot 3{,}14 \cdot 50 \cdot 1{,}5 \cdot 10^{-6}} = 800 \text{ Ом}$$

Опір r=2 Ом незначний, у зрівнянні з X_L та X_C, тому практично не вплине на значення напруги дотику. Тобто цим опором можна знехтувати.

Теж можна і сказати про опір людини Rh, так як паралельне під'єднання опору до конденсатору майже не впливає на проходження струму в ланцюзі. Таким чином, отримуємо силу струму через тіло людини:

$$I = \frac{U}{X_c - X_L} = \frac{220}{800 - 157} = 0{,}344 \text{ А}$$

Тоді, напруга дотику дорівнює:

$$U_{\text{дот}} = IX_C = 0{,}344 \cdot 800 = 276 \text{ В}$$

Таким чином отримуємо,що напруга дотику за рисунком 4.1 буде більше, ніж напруга у ланцюзі. Це явище зв’язано з наявністю у ланцюзі послідовно з’єднаних індуктивності та ємності, за рахунок яких може відбутися резонанс напруг.

Отримана напруга дотику 276 В є набагато більшою за безпечну напругу 42 В. Отриманий струм 344 мА, що набагато перевищує значення смертельного струму 100 мА, тому є дуже небезпечним.

Проаналізувавши даний випадок можна зробити висновок, що при роботі з ланцюгами, де є послідовно з’єднані ємність та індуктивність необхідно використовувати індивідуальні засоби заходу (діелектричні рукавички, тощо), напруга пробою яких більша, ніж напруга ланцюга, тому що через резонансні явища є ризик отримати електричну травму або електричний удар

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

Проведено дослідження різних видів нейронних мереж, принципу їх роботи та реалізація різних підходів до їх використання. За багатьма критеріями перевагу надано персептрону.

Розроблено програмний застосунок TrafficSignRecognition, який працює на основі нейронної мережі з використанням мов програмування C++, та C# для розпізнавання дорожніх знаків. Під час роботи над застосунком досліджено можливості бібліотеки та систему комп’ютерного зору OpenCV .

В програмному середовищі Electronic Workbench розроблено та побудовано пристрій розпізнавання образів. Пристрій складається з двох електричних схем: схеми електричної принципової для введення інформації, та функціональної схеми розпізнавання образів в коди семи сегментного індикатору.

Запропоновано схему апаратної реалізації нейрона, яка побудована на основі ПЛІС - матричних структур, що забезпечують паралельну обробку інформації і володіють високим ступенем інтеграції, масштабованістю, розвиненими засобами проектування, реалізації виконання алгоритмів навчання.

Запропоновані технічні рішення можуть бути використані для оптимізації та автоматизації процесів в різних галузях науки і техніки.

ПЕРЕЛІК ПОСИЛАНЬ

1. С.В.Ткаліченко. Штучні нейронні мережі : навч. посіб. Кривий Ріг : Державний університет економіки і технологій, 2023. 150 с.

2. Дорогий, Я.Ю. Вплив параметрів згорточних нейронних мереж на якість розпізнавання людини за фотопортретом. *Сучасні інформаційні системи і технології*: тези доповіді Першої міжнародної науково-практичної конференції, м. Суми, 15-18 травня 2012 р. / Суми : СумДУ, 2012. С. 37-38.

3. Терейковський І.А., Бушуєв Д.А., Терейковська Л.О. Штучні нейронні мережі : базові порложення: навч. посіб. Київ: Національний технічний університет України «Київський політехнічний інститут», 2022. 123 с.

4. Субботін С. О. Нейронні мережі : теорія та практика : навч. посіб. Житомир : Вид. О. О. Євенок, 2020. 184 с.

5. Charu C. Aggarval Neural Networks and Deep Learning.- Springer Nature, 2018. 512 с.

6. Троцько В.В. Методи штучного інтелекту: навчально-методичний і практичний посібник. Київ : Університет економіки та права «КРОК», 2020. 86 с.

7. OCR. URL: http://surl.li/itxrav (дата звернення: 11.09. 2024).

8. Переяславська, С., Шевченко, В., Смагіна, О. Аналіз підходів до розпізнавання текстової інформації у технології OCR. URL: http://surl.li/jeevgg (дата звернення: 25.09.2024).

9. Методи розпізнавання тексту. URL: http://surl.li/yaohhc (дата звернення: 30.09.2024).

10. Neural Networks. URL: https://www.ibm.com/cloud/learn/neuralnetworks (дата звернення: 02.10.2024).

11. OCR model. URLhttp://surl.li/viwnko (дата звернення: 07.10.2024).

12. How to use PIL. URL: https://pillow.readthedocs.io/en/stable/ (дата звернення: 09.10.2024).

13. How to use OpenCv. URL: https://pypi.org/project/opencv-python/ (дата звернення: 14.10.2024).

14. RNN neural network. URL: http://surl.li/xtpuhq (дата звернення: 18.10.2024).

15. Згорткові нейронні мережі. URL: http://surl.li/rmtagr (дата звернення: 21.10.2024).

16. Огляд та дослідження основних видів нейронних мереж. URL: http://surl.li/miduha (дата звернення: 25.10.2024).

17. Техніко-економічне обґрунтування проектно-конструкторських робіт. Методичні вказівки до самостійної роботи та виконання дипломних і контрольних робіт для студентів спеціальності 7.0908.04 «Фізична і біомедична електроніка» / Укл.: Є.Я. Швець, О.П. Головко -Запоріжжя: ЗДІА, 2010. – с.

18. Ткачук К.Н. Охорона праці та промислова безпека: навч. посіб. / К.Н. Ткачук та ін. Київ: Основа, 2014. 823с.

19. Зеркалов Д. В. Охорона праці в галузі. Київ: Основа. 2011. 551с.

20. Кепич Т.Ю. Охорона праці в галузі / Т.Ю. Кепич та ін. Київ: КНУ, 2013. 255с.

21. ДСТУ 7238:2011 Система стандартів безпеки праці. Засоби колективного захисту працюючих. Чинний від 14.03.2011. Вид. офіц. Київ: УкрНДНЦ, 2011. 78 с.

ДОДАТОК А

Повний лістинг коду програмного застосунку TrafficSignRecognition

```
// OpenCV
#include "opencv2/objdetect.hpp"
#include "opencv2/highgui.hpp"
#include "opencv2/imgproc.hpp"

// C++
#include <iostream>
#include <fstream>
#include <sstream>
#include <stdio.h>
#include <windows.h>
#include <string>



cv::Mat detectAndDisplay(cv::Mat frame, std::vector<cv::Rect> signsRed, std::vector<cv::Rect> signsBlue);
std::vector<std::string> scandir(std::string p);
std::vector<cv::Rect> ColorThreshold(cv::Mat frame, std::string color);

cv::CascadeClassifier cascadeProhibitory, cascadeDanger, cascadeYield, cascadeMandatory;
std::string imageName;

int main(int argc, char** argv)
{
	cv::Mat frame;
	imageName = argv[1];
	std::ifstream file(imageName);
	if (file.is_open()) //Проверка на наличие файла
	{
		file.close();
		frame = cv::imread(imageName, CV_LOAD_IMAGE_COLOR);
		//Осуществляем бинаризацию по цветовым порогам
		std::vector<cv::Rect> signsRed = ColorThreshold(frame, "red");
		std::vector<cv::Rect> signsBlue = ColorThreshold(frame, "blue");
		//Загружаем классификаторы и осуществляем поиск в найденных на предыдущем этапе областях
		if (!cascadeProhibitory.load("prohibitory.xml")
			|| !cascadeDanger.load("danger.xml")
			|| !cascadeYield.load("yield.xml")
```

```
                || !cascadeMandatory.load("mandatory.xml"))
            {
                printf("--(!)Error loading cascade\n");
                return -1;
            };
            frame = detectAndDisplay(frame, signsRed, signsBlue);
            cv::imwrite("result\\r_" + imageName, frame);
        }
        else
        {
            std::cout << "--(!)Error loading image\n" << std::endl;
            std::cout << imageName << std::endl;
            return -1;
        }
        return 0;
}

/** @Осуществляем бинаризацию изображения по выбранному цвету (красному или синему) */
std::vector<cv::Rect> ColorThreshold(cv::Mat frame, std::string color)
{
        cv::Mat frameThreshold, frameThreshold1, frameThreshold2, frameHSV;
        cv::cvtColor(frame, frameHSV, CV_BGR2HSV);
        if (color == "red")
        {
            inRange(frameHSV, cv::Scalar(140, 30, 5), cv::Scalar(179, 255, 255), frameThreshold1);
            inRange(frameHSV, cv::Scalar(0, 50, 5), cv::Scalar(12, 255, 255), frameThreshold2);
            cv::bitwise_or(frameThreshold1, frameThreshold2, frameThreshold);
        }
        else
            if (color == "blue")
                inRange(frameHSV, cv::Scalar(103, 100, 5), cv::Scalar(130, 255, 255), frameThreshold);
            else
                printf("--(!)Error, can't get color threshold\n");
        //морфологическое открытие
        cv::erode(frameThreshold, frameThreshold, cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(3, 3)));
        cv::dilate(frameThreshold, frameThreshold, cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(3, 3)));
        //морфологическое закрытие
        cv::dilate(frameThreshold, frameThreshold, cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(3, 3)));
        cv::erode(frameThreshold, frameThreshold, cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(3, 3)));
```

```
	//Находим связные контура, выделяем их прямоугольниками и оставляем только достаточно большие прямоугольники, близкие к квадрату
	std::vector<std::vector<cv::Point> > contours;
	std::vector<cv::Rect> signs;
	cv::findContours(frameThreshold, contours, cv::RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE);
	for (size_t i = 0; i < contours.size(); ++i)
	{
		cv::Rect r = cv::boundingRect(contours[i]);
		if ((r.height / (float)r.width > 0.7) && (r.height / (float)r.width < 1.3) && (r.width > 20) && (r.height > 20))
			signs.push_back(r);
	}
	for (size_t i = 0; i < signs.size(); i++)
	{
		rectangle(frameThreshold, signs[i], cv::Scalar(255, 0, 255), 2, 8, 0);
	}
	return signs;
}

/** @Расширяем rectangle на заданное значение */
std::vector<cv::Rect> ExpandRectangle(cv::Mat frame, std::vector<cv::Rect> r, float coef)
{
	for (size_t i = 0; i < r.size(); i++)
	{
		cv::Size deltaSize(r[i].width * coef, r[i].height * coef);
		cv::Point offset(deltaSize.width / 2, deltaSize.height / 2);
		r[i] += deltaSize;
		r[i] -= offset;
		if (r[i].x < 0) r[i].x = 0;
		if (r[i].x + r[i].width > frame.cols) r[i].width = frame.cols - r[i].x;
		if (r[i].y < 0) r[i].y = 0;
		if (r[i].y + r[i].height > frame.rows) r[i].height = frame.rows - r[i].y;
	}
	return r;
}

/** @К требуемым прямоугольным областям применяем поиск каскадным классификатором */
std::vector<cv::Rect> FindHaar(cv::Mat frame_gray, std::vector<cv::Rect> signs, cv::CascadeClassifier cascade)
{
	std::vector<cv::Rect> haarSigns, findSigns;
	for (size_t i = 0; i < signs.size(); i++)
```

```
	{
		cv::Mat signsROI = frame_gray(signs[i]);
		cascade.detectMultiScale(signsROI, haarSigns, 1.1, 2, 0 |
cv::CASCADE_SCALE_IMAGE, cv::Size(20, 20));
		for (size_t j = 0; j < haarSigns.size(); j++)
		{
			cv::Rect r;
			r.x = signs[i].x + haarSigns[j].x;
			r.y = signs[i].y + haarSigns[j].y;
			r.width = haarSigns[j].width;
			r.height = haarSigns[j].height;
			findSigns.push_back(r);
		}
	}
	return findSigns;
}

/** @Нарисовать прямоугольники на заданном изображении в заданных
вектором областях и заданным цветом */
void	DrawRectangle(cv::Mat	image,	std::vector<cv::Rect>
rectangles, cv::Vec3b color)
{
	for (size_t i = 0; i < rectangles.size(); i++)
		rectangle(image, rectangles[i], color, 2, 8, 0);
}

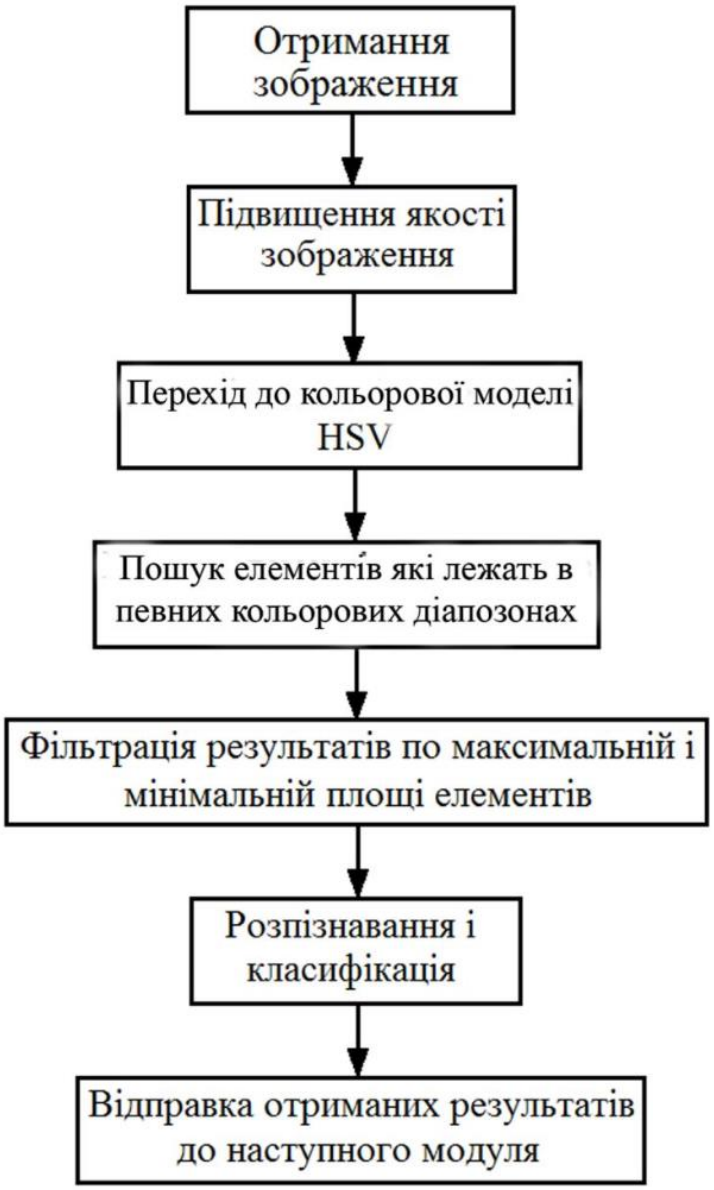
/** @Применяем  к ранее найденным контурам поиск четырьмя разными
каскадными классификаторами */
cv::Mat	detectAndDisplay(cv::Mat	frame,	std::vector<cv::Rect>
signsRed, std::vector<cv::Rect> signsBlue)
{
	cv::Mat frame_gray;
	cvtColor(frame, frame_gray, cv::COLOR_BGR2GRAY);
	equalizeHist(frame_gray, frame_gray);
	//Расширяем область для поиска на 40%
	signsRed = ExpandRectangle(frame, signsRed, 0.4);
	signsBlue = ExpandRectangle(frame, signsBlue, 0.4);
	//Ищем знаки в расширенных областях
	std::vector<cv::Rect> signsProhibitory = FindHaar(frame_gray,
signsRed, cascadeProhibitory);
	DrawRectangle(frame, signsProhibitory, cv::Vec3b(0, 0, 255));
	std::vector<cv::Rect>	signsDanger	=	FindHaar(frame_gray,
signsRed, cascadeDanger);
	DrawRectangle(frame, signsDanger, cv::Vec3b(0, 255, 255));
	std::vector<cv::Rect>	signsYield	=	FindHaar(frame_gray,
signsRed, cascadeYield);
	DrawRectangle(frame, signsYield, cv::Vec3b(255, 0, 255));
	std::vector<cv::Rect>	signsMandatory	=	FindHaar(frame_gray,
signsBlue, cascadeMandatory);
	DrawRectangle(frame, signsMandatory, cv::Vec3b(255, 0, 0));
	return frame;
```

```
}

std::wstring string_to_wstring(std::string& s)
{
	std::ofstream ofs("temp.txt", std::ofstream::out);
	if (!ofs) { std::cerr << "don't open file: 'temp.txt1' " <<
std::endl; exit(1); }
	ofs << s;
	ofs.close();
	std::wstring s1;
	std::wifstream wifs("temp.txt", std::wifstream::in);
	if (!wifs) { std::cerr << "don't open file: 'temp.txt2'" <<
std::endl; exit(1); }
	wifs >> s1;
	wifs.close();
	return s1;
}

std::string wstring_to_string(std::wstring s)
{
	std::wofstream wofs("temp.txt", std::wofstream::out);
	if (!wofs) { std::cerr << "don't open file: 'temp.txt3'" <<
std::endl; exit(1); }
	wofs << s;
	wofs.close();
	std::string s1;
	std::ifstream ifs("temp.txt", std::ifstream::in);
	if (!ifs) { std::cerr << "don't open file: 'temp.txt3'" <<
std::endl; exit(1); }
	ifs >> s1;
	ifs.close();
	return s1;
}
/** @Получение списка файлов из директории */
std::vector<std::string> scandir(std::string p)
{
	std::wstring Path = string_to_wstring(p);
	WIN32_FIND_DATA FindFileData;
	HANDLE hf;
	hf	=	FindFirstFile((string_to_wstring(p)).c_str(),
&FindFileData);
	std::vector<std::string> v;
	if (hf != INVALID_HANDLE_VALUE)
	{
		do
		{
			std::wcout << "FindFileData.dwFileAttributes= " <<
FindFileData.dwFileAttributes << std::endl;
			if	(FindFileData.dwFileAttributes	!=
FILE_ATTRIBUTE_DIRECTORY)
```

```
        v.push_back(wstring_to_string(FindFileData.cFileName));
             } while (FindNextFile(hf, &FindFileData) != 0);
        }
        return v;
}
```



Блок-схема роботи програмного застосунку

ДОДАТОК В

«ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Науково-дослідна частина
Наукове товариство
студентів, аспірантів і докторантів
Рада молодих вчених

ЗБІРНИК
наукових праць студентів,
здобувач PhDів, докторантів і молодих вчених

«МОЛОДА НАУКА-2024»

Том V

«Секції Інженерного навчально-наукового інституту ім. Ю.М. Потебні Запорізького національного університету»

Запоріжжя
2024

2024=УДК: 001 (06)
М 754

РЕДАКЦІЙНА КОЛЕГІЯ

Фролов М.О. *– в. о. ректор ЗНУ;*
Васильчук Г.М. *– проректор з наукової роботи;*
Арабаджиєв Д.Ю. *– начальник науково-дослідної частини;*
Бєлоконь К.В. *– заступник директора з наукової роботи Інженерного навчально-наукового інституту ім. Ю.М. Потебні ЗНУ;*
Білівненко С.М*. – заступник декана з наукової роботи історичного факультету;*
Бобришева К.О*. – голова НТ САД факультету іноземної філології;*
Будько Д.З. *– голова НТ САД економічного факультету;*
Вербицька Т.А. *– голова НТ САД біологічного факультету;*
Євтушенко Д.С. *– заступник декана з наукової роботи юридичного факультету;*
Єгорова В.С. *– голова НТ САД математичного факультету;*
Залужна М.В. *– заступник декана з наукової роботи факультету іноземної філології;*
Зубцова Ю.С. *– заступник декана з наукової роботи факультету соціальної педагогіки та психології;*
Кайшева Д.О.*– голова НТ САД факультету менеджменту;*
Караулова С.І. *– заступник декана з наукової роботи факультету фізичного виховання, здоров'я та туризму;*
Ковпак В.А. *– заступник декана з наукової роботи факультету журналістики;*
Копійка В.В. *– заступник декана з наукової роботи біологічного факультету;*
Леонтьева В.В. *– заступник декана з наукової роботи математичного факультету;*
Лепська Н.В. *– заступник декана з наукової роботи факультету соціології та управління;*
Линенко А.В. *– заступник декана з наукової роботи економічного факультету;*
Ляхов Г.І. *– голова НТ САД юридичного факультету;*
Меркулова О.В. *– заступник декана з наукової роботи філологічного факультету;*
Олійник О.М. *– заступник декана з наукової робота факультету менеджменту;*
Рибка Д.О. *– голова НТ САД факультету журналістики;*
Руднік О.Р. *– голова НТ САД ЗНУ;*
Саражина Є.В. *– голова НТ САД факультету фізичного виховання здоров'я та туризму;*
Сінельнікова П.Д. *– голова НТ САД факультету соціальної педагогіки та психології;*
Тарабан Є.В.*– голова НТ САД Інженерного навчально-наукового інституту ім. Ю.М. Потебні ЗНУ;*
Хавер Д.І.*– фахівець центру наукової роботи студентів НДЧ;*
Харчева Д.Р. *– голова-НТ САД філологічного факультету;*
Ходаковська А.В. *– заступник директора з навчально-методичної роботи ЕПФК;*
Шумська А.Ю. *– голова НТ САД історичного факультету;*
Ясьмо О.В. *– голова НТ САД факультету соціології та управління.*

150

Ларченко Олег
магістрант 1 курсу ІННІ ім. Ю.М. Потебні
Наук. кер.: канд. техн. наук, доц. Небеснюк О.Ю.

ПРИСТРІЙ ДЛЯ РОЗПІЗНАВАННЯ ОБРАЗІВ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ

Дослідження в області нейронних мереж почалися в 40-і роки XX століття. Перше систематичне вивчення штучних нейронних мереж було зроблено Маккалокком і Піттсом в 1943 році. Пізніше вони досліджували мережеві парадигми для розпізнавання зображень, що піддаються зсувам і поворотам [1].

Основні завдання, які ставляться перед нейронними мережами, відносяться до завдань розпізнавання образів. Вони полягають в тому, щоб класифікувати вхідний образ, тобто віднести його до якого-небудь відомого мережі класу. Спочатку мережі даються еталонні образи – такі образи, належність яких до певного класу відома. Потім на вхід мережі подається деякий невідомий образ, і мережа намагається за певним алгоритмом співвіднести його з будь-яким еталонним образом. Можна сказати, що нейронні мережі проводять кластеризацию образів. Так як кластерний аналіз застосовується дослідниками ринку цінних паперів, то нейронні мережі можуть бути використані і для прогнозування вартості акцій, що є актуальним завданням, при цьому суворо нерозв'язною на даний момент.

Так само нейронні мережі можуть застосовуватися в криміналістиці (аналіз відбитків пальців) або ж для полегшення роботи правоохоронних органів у затриманні злочинців, тому що зараз вже створені такі програми, які розпізнають обличчя [2].

Все вищенаведене свідчить про те, що проблеми дослідження можливостей нейронних мереж і їх розвитку є актуальною на даний момент часу.

Розпізнавання образів (об'єктів, сигналів, ситуацій, явищ або процесів) – завдання ідентифікації об'єкта або визначення будь-яких його властивостей по його зображенню (оптичне розпізнавання) або аудіозапису (акустичне розпізнавання) та іншим характеристикам.

Базою для вирішення завдань віднесення об'єктів до того чи іншого класу послужили результати класичної теорії статистичних рішень. В її рамках будувалися алгоритми, що забезпечують на основі експериментальних вимірювань параметрів, що характеризують цей об'єкт, а також деяких апріорних даних, що описують класи, визначення конкретного класу, до якого може бути віднесений об'єкт розпізнавання.

Приклади завдань розпізнавання образів:

- розпізнавання букв;
- розпізнавання штрих-кодів;
- розпізнавання автомобільних номерів;
- розпізнавання осіб та інших біометричних даних;

- розпізнавання мови [3].

Досить часто початківці і вже досвідчені водії стикаються з низкою проблем під час керування транспортним засобом. Водієві необхідно встигати стежити за розміткою на дорозі і бачити всі дорожні знаки, щоб уникнути аварійних і непередбачуваних ситуацій.

Розроблено програмний застосунок TrafficSignRecognition, який працює на основі нейронної мережі з використанням мов програмування C++, та C# для розпізнавання дорожніх знаків. Під час роботи над застосунком досліджено можливості бібліотеки та систему комп'ютерного зору OpenCV.

Авторами запропоновано схему апаратної реалізації нейрона на основі ПЛІС – матричних структур, що забезпечують паралельну обробку інформації і володіють високим ступенем інтеграції, масштабованістю, розвиненими засобами проектування, реалізації виконання алгоритмів навчання.

Запропоновані технічні рішення можуть бути використані для оптимізації та автоматизації процесів в різних галузях науки і техніки.

Література

1. Тимощук П. В. Штучні нейронні мережі / П. В. Тимощук. Львів: «Львівська політехніка», 2011. 444 с.

2. І. А. Терейковський, Д. А. Бушуєв, Л. О. Терейковська. Штучні нейронні мережі: базові положення. Навч.посібник. Київ: КПІ ім. Ігоря Сікорського, 2022. 124 с.

3. Karpathy A. et al. Convolutional Neural Networks for Visual Recognition. URL: http://www.cs231n.stanford.edu (дата звернення 05.03.2024)

Маслак Роман
студент 4 курсу ІННІ ім. Ю.М. Потебні
Наук. кер.: викладач Туришев К.О.

ЗАРЯДНИЙ ПРИСТРІЙ ДЛЯ ЦЕХОВОГО ЕЛЕКТРОТРАНСПОРТУ

Перевезення вантажів на відносно невеликі відстані в межах промислових підприємств, у тому числі всередині цехів та на складах здійснюється за допомогою електрокарів. Електрокар є компактним електромобілем з відкритою платформою і призначений для транспортування одиночних вантажів по дорогах з твердим покриттям. Привід електрокара здійснюється від двигуна постійного струму. Джерелом електропостачання є лужна або кислотна акумуляторна батарея (АКБ) ємністю від 165 А/год до 280 А/год [1, 2].

Напруга акумуляторних батарей електрокарів, залежно від модифікації, становить 40 В або 80 В (дві акумуляторні батареї, включені послідовно).

Для ефективної експлуатації електрокарів необхідний пристрій заряду АКБ, що забезпечує можливість завдання величини напруги та струму заряду.

Структурну схему зарядного пристрою для цехового електротранспорту (ЗП) наведено на рисунку 1.