

Міністерство освіти і науки України
Департамент освіти і науки Полтавської обласної державної адміністрації
Полтавське територіальне відділення МАН України

Відділення комп'ютерних наук

Секція: інформаційні системи, бази
даних та системи штучного інтелекту

СИСТЕМА РОЗПІЗНАВАННЯ МАТЕМАТИЧНИХ ВИРАЗІВ

Роботу виконав:

Головко Євгеній Олегович,
учень 11 класу

Полтавського обласного наукового
ліцею-інтернату II-III ступенів при
Кременчуцькому педагогічному коледжі
імені А.С. Макаренка

Науковий керівник:

Лисенко Тетяна Іванівна,
викладач інформатики

Кременчуцького педагогічного
коледжу імені А.С. Макаренка,
Заслужений учитель України

Полтава-2020

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Теоретичні основи розробки нейронних мереж	5
1.1. Біологічні основи нейронних мереж	5
1.2. Штучний нейрон та штучна нейронна мережа	6
1.3. Архітектура нейронних мереж	7
1.4. Процес навчання мережі	9
1.5. Сфери застосування нейронних мереж	10
РОЗДІЛ 2. Комп'ютерна реалізація нейронної мережі	12
2.1. Модель нейронної мережі для розпізнавання математичних виразів	12
2.2. Комп'ютерна реалізація моделі нейронної мережі	13
2.3. Опис інтерфейсу програми	16
2.4. Налаштування програми <i>NeuroMath</i>	18
ВИСНОВКИ	21
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	23

ВСТУП

Протягом тисячоліть людство намагається розгадати таємницю роботи людського мозку та створити пристрої, здатні мислити. Ідея проектування інтелектуальних обчислювальних пристроїв за подобою біологічних систем призвела до створення теорії нейронних мереж, що стала одним з найбільш потужних підходів до розробки штучного інтелекту. Особливістю розробок у цій галузі є пошук алгоритмів, які моделюють людську здатність приймати рішення на основі попереднього досвіду та аналізу поточної ситуації.

Розвиток засобів роботизації, поширення пристроїв сенсорного введення даних, потреби криптографічного захисту потребують розробки систем для розпізнавання графічних образів, що зберігаються на твердих носіях. Уміння розпізнавати букви, цифри та інші символи, рукописні або надруковані, ми сприймаємо як елементарну дію людського мозку. Але в галузі комп'ютерних наук – це **актуальна науково-практична проблема**, що входить до категорії задач штучного інтелекту.

Наразі існує багато підходів до вирішення проблеми розпізнавання символів, у тому числі – застосування штучних нейронних мереж. Вивчення штучних нейронних мереж розпочалось на початку 40-х років ХХ ст. з праць У. Маккалока та У. Піттса, які сформулювали поняття «нейронна мережа». Дослідженням штучних нейронних систем також займалися такі провідні вчені як Н. Вінер, Д. Хебб, Ф. Розенблатт, Т. Кохонен, Дж. Андерсон та інші.

Наразі нейронні мережі покладені в основу ряду технологій, таких як системи автоматичного розпізнавання автомобільних номерів або зчитування поштових індексів, написаних від руки. Але залишається багато галузей діяльності, у яких застосування нейронних мереж могло б позбавити людину від нетворчої діяльності та забезпечити безпомилковість у прийнятті рішень. Такими є, наприклад, питання перевірки правильності виконання математичних розрахунків у науковій, інженерній, освітній, побутовій та інших галузях. Це пояснює **актуальність даного дослідження**.

Об'єктом дослідження є штучні нейронні мережі.

Предмет дослідження – штучна нейронна мережа як засіб розпізнавання математичних виразів, отриманих шляхом рукописного вводу.

Мета дослідження – змодельовати, розробити, провести попереднє навчання штучної нейронної мережі, що виконуватиме розпізнавання математичних виразів і на основі результатів розпізнавання – обчислення їх значень.

Для досягнення мети були поставлені такі **завдання**:

1. Ознайомитись з поняттям штучної нейронної мережі, її складовими.
2. Проаналізувати архітектуру штучних нейронних мереж та методи їх навчання.
3. Розробити математичну модель та комп'ютерну реалізацію нейронної мережі для розпізнавання математичних виразів, отриманих шляхом рукописного вводу.

Для розв'язання поставлених завдань використано теоретичні **методи дослідження** (аналіз, синтез, порівняння), емпіричні (моделювання, тестування), методи кількісного й якісного аналізу даних, комп'ютерний експеримент.

Наукова новизна дослідження полягає у тому, що набуло подальшого розвитку питання про комп'ютерну реалізацію штучних нейронних мереж, призначених для розпізнавання даних, отриманих шляхом рукописного введення. Результати роботи сприятимуть подальшому розвитку теорії штучного інтелекту та штучних нейронних мереж.

Робота має **прикладне значення**. Матеріали та результати дослідження можуть бути використані для моделювання та навчання штучних нейронних мереж, що орієнтовані на розпізнавання даних, отриманих шляхом рукописного введення. Створений програмний засіб може бути використаний для пояснення сутності поняття нейронної мережі учням класів інформаційного профілю або студентам ІТ-спеціальностей.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ НЕЙРОННИХ МЕРЕЖ

1.1. Біологічні основи нейронних мереж

Розвиток штучних нейронних мереж, як одного з напрямків штучного інтелекту, надихається біологією живих істот, принципами побудови та функціонуванням мозку. Базовим елементом мозку є специфічні клітини, відомі як **нейрони**, здатні запам'ятовувати та застосовувати попередній досвід до кожної дії [2]. У мозку людини нараховується майже 100 млрд нейронів, і людина здатна мислити. У мозку мухи дрозофіли – 100 тис. нейронів, і вона здатна літати, харчуватися, уникати небезпеки, знаходити їжу та виконувати інші дії. Мозок круглого черв'яка нематоди містить усього 302 нейрони, і цього достатньо для забезпечення його життєдіяльності.

Біологічний нейрон має зв'язки з тисячами інших нейронів. Він отримує сигнал, що поступає від інших нейронів, певним чином перетворює його та передає наступним нейронам.

На рисунку 1.1 зображена будова біологічного нейрона.

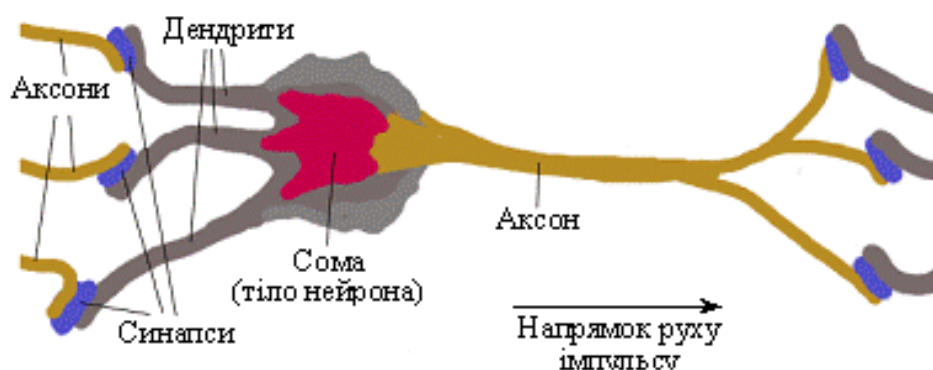


Рис. 1.1. Будова біологічного нейрона

Аксони – вихідні елементи нейронів, передають сигнали від одного нейрона іншому. **Дендрити** приймають сигнали від попередніх нейронів у точках з'єднання – **синапсах**, та передають до тіла нейрона **сома**. Там сигнали додаються, причому одні входи намагаються активізувати нейрон, інші –

запобігти його активності. Коли сумарна дія в тілі нейрона перевищує деякий поріг, нейрон активізується та передає сигнал іншим нейронам через аксон.

Структуру, утворену з поєднаних між собою нейронів мозку, називають біологічною нейронною мережею.

За рахунок передавання імпульсів у біологічній нейронній мережі мозок опрацьовує дані, отримані від зовнішнього середовища, навчається, виконує операції з розпізнавання образів, робить передбачення та ін.

Для реалізації засобами комп'ютерної техніки операцій, подібних до функціонування мозку, використовують штучні нейронні мережі.

1.2. Штучний нейрон та штучна нейронна мережа

Штучні нейронні мережі – математичні моделі, їх програмні чи апаратні реалізації, побудовані за принципом організації та функціонування біологічних нейронних мереж – мереж нервових клітин живого організму.

З інженерної точки зору штучні нейронні мережі – це паралельно розподілена система обробки інформації, утворена тісно зв'язаними простими обчислювальними вузлами (однотипними або різними), що має властивість накопичувати експериментальні знання, узагальнювати їх і робити доступними для користувача у формі, зручній для інтерпретації й прийняття рішень [4].

Трьома основними поняттями в теорії штучних нейронних мереж є нейрон, архітектура мережі та поняття навчання [5].

Базовий модуль штучних нейронних мереж – **штучний нейрон** моделює основні функції природного нейрона (рис. 1.2).

Аналогами сигналів, що поступають з аксонів біологічного нейрона, у штучному нейроні є вхідні сигнали мережі – x_1, x_2, \dots, x_n . Сигнали поступають через синапси, кожний з яких характеризується своєю вагою w_1, w_2, \dots, w_n відповідно [1].

Суматор Σ додає вхідні сигнали за формулою:

$$NET = x_1w_1 + x_2w_2 + \dots + x_nw_n$$

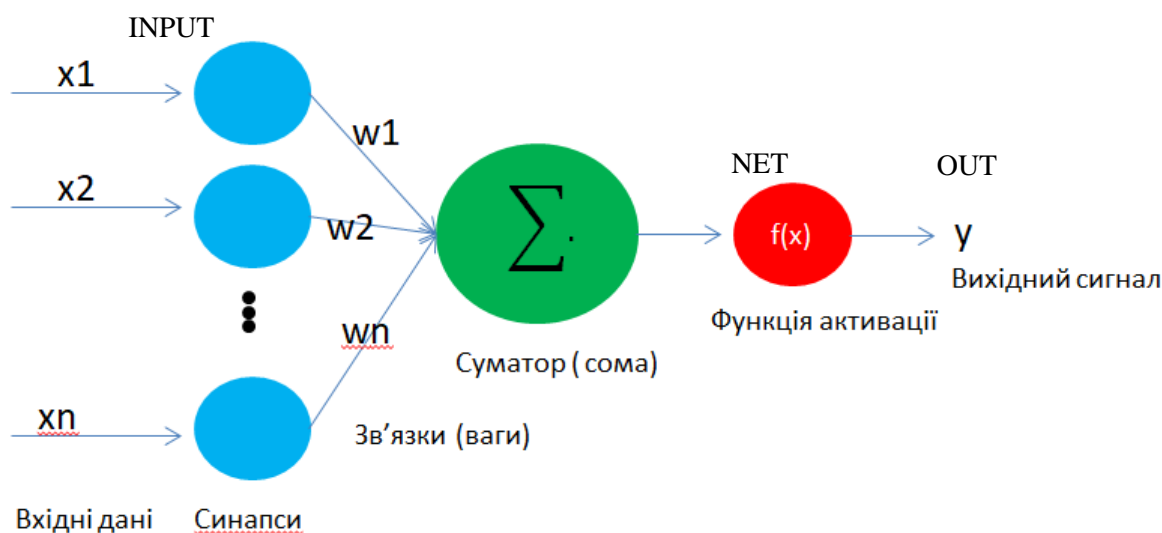


Рис. 1.2. Модель штучного нейрона

Функція активації f описує правило утворення вихідного сигналу нейрона OUT при надходженні нових сигналів x . В якості функції активації можуть бути використані лінійна функція, синус, гіперболічний тангенс та інші, але найчастіше використовують функцію, яка має назву сигмоїда або логічна функція:

$$y = \frac{1}{1 + e^{-NET}}$$

Якщо комбінований сигнал недостатньо сильний, то сигмоїда подавляє його, інакше підсилює – «збуджує нейрон» та передає оновлений сигнал до наступного шару нейронів.

Група штучних нейронів, що взаємопов'язані між собою та складають певні прошарки, утворюють штучну нейронну мережу.

1.3. Архітектура нейронних мереж

З точки зору архітектури штучна нейронна мережа може розглядатися як орієнтований граф зі зваженими зв'язками, в якому штучні нейрони є вузлами. За архітектурою зв'язків штучні нейронні мережі можуть бути згруповані в два класи: мережі прямого поширення, в яких графи не мають петель, і рекурентні

мережі, або мережі зі зворотними зв'язками [7], які представлені графами з петлями.

Ми розглядатимемо лише мережі прямого поширення, які називаються багатошаровим **перцептроном**. У них нейрони розташовані шарами та мають односпрямовані зв'язки між шарами.

Багатошарові нейронні мережі прямого поширення містять як мінімум три типи прошарків: вхідний, прихований та вихідний [2]. Шар вхідних нейронів отримує вхідні дані. Вихідний прошарок пересилає інформацію безпосередньо в зовнішнє середовище. Між цими двома шарами може бути кілька прихованих шарів, що містять багато пов'язаних нейронів. Кількість нейронів у кожному шарі може бути різною. Входи та виходи кожного з прихованих нейронів з'єднані з іншими нейронами.

Кожен нейрон прихованого шару отримує сигнали від всіх нейронів попереднього шару. Після виконання операцій над сигналами, нейрон передає свій вихід до всіх нейронів наступного шару, забезпечуючи передачу сигналу вперед на вихід (рис. 1.3).

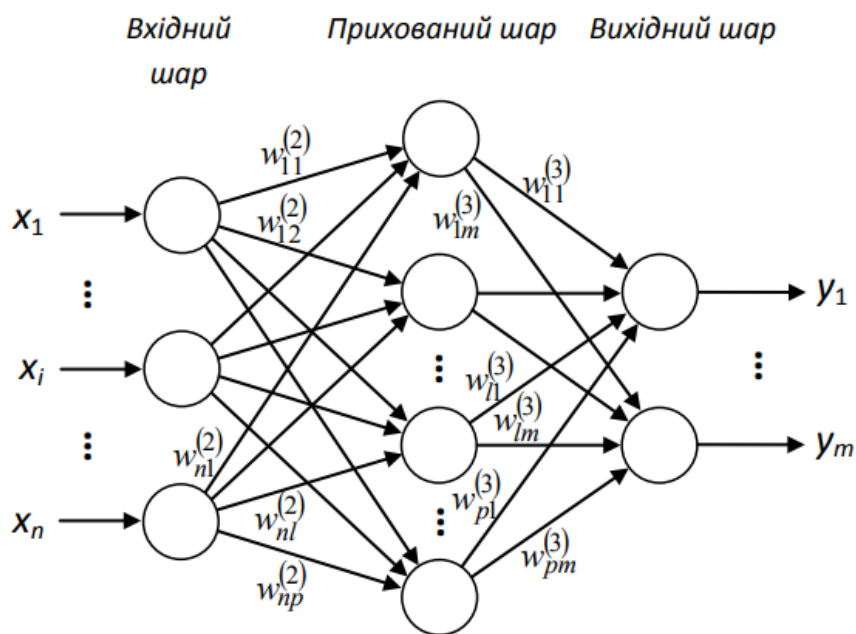


Рис. 1.3. Схематична архітектура тришарового перцептрону

Модель перцептрону може бути описана математично. Ваги є елементами матриці W , що має m рядків та n стовпців, де m – кількість нейронів у прошарку, а n – кількість входів. На вхід мережі подається вектор X вхідних сигналів. Обчислення вихідного вектору Y , компонентами якого є виходи m нейронів, зводиться до матричного множення $Y = W \cdot X$, де X та Y – вектори-стовпці.

$$W = \begin{pmatrix} w_{11} & w_{12} \dots & w_{1n} \\ w_{21} & w_{22} \dots & w_{2n} \\ \dots & \dots & \dots \\ w_{m1} & w_{m2} \dots & w_{mn} \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad Y = W \cdot X = \begin{pmatrix} w_{11}x_1 + w_{12}x_2 + \dots + w_{1n}x_n \\ w_{21}x_1 + w_{22}x_2 + \dots + w_{2n}x_n \\ \dots \\ w_{m1}x_1 + w_{m2}x_2 + \dots + w_{mn}x_n \end{pmatrix}.$$

Якщо функцією активації є сигмоїда $y = \frac{1}{1 + e^{-N}}$, то вона звужує діапазон зміни сигналу Y таким чином, що значення y лежать у межах від 0 до 1.

1.4. Процес навчання мережі

Переваги нейронних мереж порівняно з іншими системами штучного інтелекту в тому, що вони мають змогу навчатися – змінювати власні параметри таким чином, щоб після завершення процесу навчання отримувати на виході очікувані (правильні) результати при будь-яких наборах коректних вхідних даних.

Для процесу навчання необхідно мати модель зовнішнього середовища, в якому функціонує нейронна мережа.

При навчанні мережі кожна вхідна множина сигналів розглядається як вектор. Навчання здійснюється шляхом послідовного пред'явлення вхідних векторів з одночасним налаштуванням ваги відповідно до певної процедури. У процесі навчання ваги мережі поступово стають такими, щоб кожний вхідний вектор відповідно виробляв вихідний.

Навчальні алгоритми можуть бути класифіковані як алгоритми навчання з учителем і без учителя.

Навчання з учителем передбачає, що для кожного вхідного вектору існує цільовий вектор, що представляє собою необхідний вихід. Разом вони називаються навчальною парою. Зазвичай мережа навчається на деякій значній кількості таких навчальних пар. На вхід подається вхідний вектор, обчислюється вихід мережі та порівнюється з відповідним цільовим вектором. Помилка, отримана на виході, повертається в мережу, і ваги змінюються відповідно до алгоритму, який прагне мінімізувати помилку.

Для коригування ваги у таких нейронних мережах застосовується метод зворотного поширення похибки – різниці між цільовим значенням навчальної пари та фактичним вихідним значенням. Похибка на вузлах вихідного шару визначається як проста різниця між бажаним та фактичним вихідним значенням. На прихованих прошарках похибка розподіляється пропорційно вазі кожного зв'язку з подальшим об'єднанням відповідних окремих частин похибки на кожному внутрішньому вузлі.

Після достатнього числа повторень цих кроків різниця між дійсними виходами та цільовими виходами повинна зменшитись до прийнятної величини. При цьому кажуть, що мережа навчилася і може використовуватися для реалізації основного завдання.

При навчанні без учителя навчальна множина складається лише з вхідних векторів. Навчальний алгоритм підлаштовує ваги мережі так, щоб пред'явлення досить близьких вхідних векторів давало однакові виходи. Процес навчання виділяє статистичні властивості навчальної множини і групує подібні вектори в класи. Таким чином, до завершення навчання неможливо передбачити, який вихід буде отриманий з даного набору вхідних векторів.

1.5. Сфери застосування нейронних мереж

Наразі нейронні мережі знаходять широке застосування у різних галузях науки та людської діяльності (табл. 1.1).

Таблиця 1.1

Основні функції штучних нейронних мереж

Функція	Зміст	Застосування
Класифікація образів	Визначення належності вхідного образу, поданого вектором ознак, до одного або декількох попередньо визначених класів	Розпізнавання літер, мови
Кластеризація / категоризація	Віднесення схожих образів до одного кластеру	Для здобування знання, стиснення даних та дослідження їх властивостей
Апроксимація функції	Знаходження невідомої функції F , спотвореної шумом	Для задач моделювання
Прогноз / передбачення	Передбачення значення в наступний період часу	Для прийняття рішень в бізнесі, науці та техніці
Оптимізація	Знаходження такого рішення, яке задовольняє системі обмежень і максимізує або мінімізує цільову функцію	Для розв'язання задач оптимізації в статистиці, економіці та ін.
Асоціативна пам'ять	Знаходження необхідного документа по заданому змісту	Мультимедійні інформаційні бази даних
Управління	Розрахунок такого вхідного впливу, при якому система діє за бажаною траєкторією, заданою еталонною моделлю	Керівництво приладами

Таким чином, для розробки нейронної мережі, призначеної для розпізнавання математичних виразів, отриманих шляхом рукописного введення,

потрібно окреслити архітектуру мережі, вибрати функцію активізації та метод навчання мережі, підготувати набір навчальних пар та реалізувати виконання основної функції мережі.

РОЗДІЛ 2

КОМП'ЮТЕРНА РЕАЛІЗАЦІЯ НЕЙРОННОЇ МЕРЕЖІ

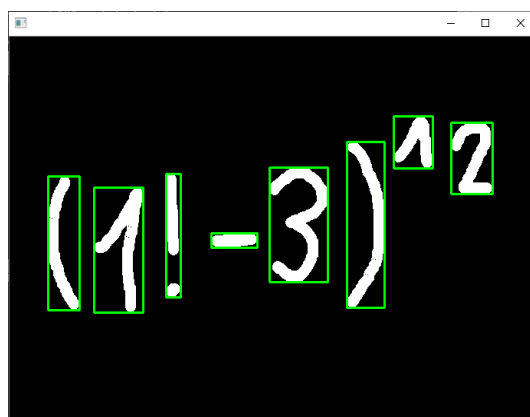
2.1. Модель нейронної мережі для розпізнавання математичних виразів

Виходячи з поставленого завдання – створити інструмент для моделювання та навчання нейронної мережі для розпізнавання математичних виразів – визначимо основні параметри штучної нейронної мережі.

Об'єкт розпізнавання – математичні вирази, отримані шляхом рукописного введення.

Для реалізації вибрано штучну нейронну мережу типу багатошаровий перцептрон. Система складається з 3-х нейронних мереж: перша – для розпізнавання цифр, друга – для символів математичних операцій та дужок, третя – для латинських букв, з яких складаються математичні функції та константи.

Для отримання вхідного вектору розіб'ємо зображення математичного виразу на окремі символи (рис. 2.1), масштабуємо їх до рівних розмірів 28*28 пікселів, з яких буде



утворено вектор кольорів пікселів. У результаті маємо вектори для кожного символу. Це потребує 784-ох нейронів на вхідному шарі кожної мережі. Кількість нейронів обумовлена форматом даних, які будуть використані для навчання мережі розпізнавання цифр.

Вихідний прошарок першої мережі містить 11 нейронів, 10 з яких відповідають цифрам від 0 до 9; 17 нейронів у другій мережі, 16 з яких відповідають символам математичних операцій; 27 нейронів у третій мережі, 26 з яких відповідають буквам латиниці. Кожна мережа має один додатковий нейрон, який активується, якщо введений символ не відповідає її задачі. Це дає

змогу в майбутньому розширювати функціонал системи шляхом додавання нових нейронних мереж.

Вихідні вектори нейронних мереж містять набори дійсних значень які відповідають області визначення функції активації $E(\text{sigmoid}) \in (0; 1)$. Кожне з них виражає долю ймовірності, що уведений символ дійсно є відповідною цифрою, символом математичної операції або буквою. За результат обирається той символ, для якого значення відповідної долі буде найбільшим у векторі.

У кожній нейронній мережі створеної системи використано по одному прихованому прошарку: у мережі з цифрами – на 100 нейронів, у мережі з розпізнавання символів математичних операцій – на 300 нейронів, у мережі з буквами – на 200 нейронів.

Мережа з цифрами містить найменшу кількість вихідних нейронів, тому їй надана відносно невелика кількість прихованих нейронів. Найкращі результати спостерігаються для двох прихованих шарів із 100 і 50 прихованими нейронами, але приріст не дуже значний (6%), на відміну від затрачених ресурсів, тому доцільно залишити один прихованих прошарок.

Мережа з математичними символами містить середню кількість вихідних нейронів, порівняно із двома іншими, а також для неї найменша кількість тренувальних прикладів, тому їй надано 300 прихованих нейронів на одному шарі. З такою кількістю нейронів вона навчається найшвидше.

Мережа для розпізнавання букв має найбільшу кількість вихідних нейронів. Їй надано 200 нейронів на одному прихованому шарі, оскільки букв, які можуть використовуватися в найменуванні функцій, лише 13 (s, i, n, c, o, t, h, a, l, g, e, x, p).

Таким чином, у першій мережі ваги зв'язків між нейронами вхідного та прихованого прошарків виражені матрицею W :

$$W_{784 \times 100} = (w_{11} \cdots w_{1,100} \vdots \vdots w_{784,1} \cdots w_{784,100}),$$

а між нейронами прихованого та вихідного прошарків – матрицею W' :

$$W'_{100 \times 11} = (w_{11} \cdots w_{1,11} \vdots \vdots w_{100,1} \cdots w_{100,11}).$$

Для інших мереж матриці ваги зв'язків аналогічні, відрізняються лише кількістю стовпців у першій та кількістю рядків у другій.

Функцією активації у всіх нейронних мережах є сигмоїда:

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}.$$

Для мереж вибрано алгоритми навчання з учителем, корекція ваги зв'язків відбувається за алгоритмом зворотного поширення похибки.

2.2. Комп'ютерна реалізація моделі нейронної мережі

Програма **NeuroMath** розроблена мовою програмування Python. Графічний інтерфейс побудований з використанням бібліотеки Kivy. Опрацювання зображень виконують функції бібліотеки OpenCV. Для реалізації нейронної мережі використано функцію сигмоїда з бібліотеки scipy пакета scipy та функції матричних операцій з бібліотеки numpy.

Для забезпечення функціонування нейронної мережі розроблено клас `NeuralNetwork`, основними функціями якого є:

- `init(lr: float, nodes: list<int>)` – ініціалізація нейронної мережі,
- `train(inputsList: list<float>, targetsList: list<float>)` – навчання мережі,
- `query(inputsList: list<float>)` – запит до мережі.
- `saveAs(fileName: string, scores=None: float)` – збереження мережі
- `loadFrom(fileName: string)` – завантаження збереженої мережі

Наведемо фрагмент коду, що відповідає за реалізацію моделі штучної нейронної мережі на основі створеного класу та збереження її параметрів (рис. 2.2).

```

1  import numpy as np          # імпортуємо numpy, скоротивши назву до np
2  import neural_network as NN # імпортуємо neural_network, скоротивши назву до NN
3
4  inputNodes      = 28*27      # кількість вхідних нейронів
5  hiddenNodes     = 100       # кількість нейронів прихованого шару
6  outputNodes     = 11        # кількість вихідних нейронів
7  learning_rate   = .1        # коефіцієнт навчання
8  epochs          = 10        # кількість epoch
9
10 n = NN.NeuralNetwork()      # створюємо екземпляр класу NeuralNetwork
11
12 # ініціалізуємо мережу методом з сигнатурою void init(float, list<int>)
13 # перший параметр - коефіцієнт навчання, змінна дійсного типу
14 # другий параметр - список цілих чисел з шарами мережі від вхідного до вихідного
15 n.init(learning_rate, [inputNodes, hiddenNodes, outputNodes])
16
17 # якщо у вас вже є мережа і ви не хочете створювати нову, то слід використати
18 # NeuralNetwork().loadFrom(String), єдиний аргумент - шлях до файлу мережі
19
20 n.saveAs("TestNetwork.json") # зберігаємо мережу в \TestNetwork.json
21

```

Рис. 2.2. Фрагмент коду для реалізації моделі нейронної мережі

Після виконання наведеного коду маємо нейронну мережу, яку можна застосовувати. Ваги усіх її зв'язків мають попереднє стале значення 0.1, тому спочатку її потрібно навчити.

Для навчання мережі розпізнавання цифр використана відкрита бази даних рукописних цифр MNIST (<http://yann.lecun.com/exdb/mnist/>, <https://pjreddie.com/projects/mnist-in-csv/>). Зображення цифр у базі оцифровано та подано у файлах формату csv. Першим символом йде маркер – очікуваний результат на виході мережі, наступні 784 символів – значення кольорів пікселів зображення, числа від 0 (білий) до 255 (чорний). Для навчання мережі розпізнавання літер використані зображення з подібної бази даних EMNIST (<https://www.nist.gov/itl/products-and-services/emnist-dataset>), для символів – власні зображення. Усього для навчання трьох мереж використано понад 184800 зображень.

Наведемо код функції навчання нейронної мережі для розпізнавання цифр (рис. 2.3).

Рис. 2.3. Код функції навчання нейронної мережі для розпізнавання цифр

```

20     def train(self, inputsList, targetsList):
21         # перетворюємо списки у двовимірний масив, транспонуємо його
22         inputs = np.array(inputsList, ndmin=2).T
23         targets = np.array(targetsList, ndmin=2).T
24
25         # зберігаємо результати для кожного шару
26         layerOutputs = [inputs]      # вихідні данні на шарі (після активації)
27
28         # зберігаємо результат для останньої дії
29         finalInputs = None           # вхідні дані на шарі (перед активацією)
30
31
32         for i in range(len(self.nodes)-1): # для кожного шару
33             # обчислюємо значення нейронів на шарі в два етапи
34             # 1) обчислюємо вхідні значення нейрону
35             # 2) обчислюємо вихідні значення нейрону
36             tempLayerInputs = np.dot(self.weights[i], layerOutputs[i])
37             tempLayerOutputs = self.activationFunction(tempLayerInputs)
38
39             # зберігаємо результати
40             finalInputs = tempLayerInputs
41             layerOutputs.append(tempLayerOutputs)
42
43         # значення похибки для кожного шару
44         outputErrors = [targets-finalInputs]
45
46         for i in range(len(self.weights)-1, 0, -1):
47             # в зворотньому напрямі для кожного шару підраховуємо похибку
48             outputErrors.insert(0,
49                                 np.dot(self.weights[i].T, outputErrors[0]))
50
51         # корегуємо значення зв'язків для кожного шару, маючи значення похибки на кожному шарі
52         # і попередні значення зв'язків
53         for i in range(len(self.weights)-1, -1, -1):
54             self.weights[i] += self.lr * np.dot((outputErrors[i] * layerOutputs[i+1] *
55             (1. - layerOutputs[i+1])), np.transpose(layerOutputs[i]))
56

```

Після завершення навчання дані про параметри кожної мережі – вага зв'язків, кількість шарів, нейронів, відсоток правильних відповідей за результатами тестування, містяться у відповідному файлі: NNNumbers.json для розпізнавання цифр, NNOperators.json – для символів, NNLetters.json – для букв, та за потреби можуть бути змінені.

2.3. Обчислення значень виразів

Окремою задачею системи є обчислення значення виразу, отриманого в результаті розпізнавання. При використанні вбудованої в Python функції *eval(string)* не буде отримано правильного результату, якщо вираз записаний не за правилами мови Python, тобто містить символи ^ для піднесення до степені, ! для обчислення факторіалу числа тощо.

У системі використано алгоритм обчислення значення виразу за польською нотацією з попереднім опрацюванням виразу для дотримання пріоритету операцій.

Опрацювання виразу здійснюється з використанням бінарного дерева. В нелистових вузлах розміщуються оператори або функції, у листових – числа і константи. Для цього складові виразу розподіляються на групи, для яких задано пріоритети: числа, константи (π , e), математичні оператори, спеціальні символи (дужки, модуль), математичні функції, спеціальні функції (факторіал).

На основі розподілу даних на категорії будується бінарне дерево. Вузли операторів мають двох нащадків, константи та числа – листва, функції – кількість нащадків залежить від аргументів, спеціальні функції та спеціальні символи мають лише одного нащадка.

Розглянемо дерево, яке утвориться за наведеними правилами для математичного виразу:

$$16 + 2 * (5 - | - 2|)^2 + (\sin \sin (\pi/2) + \log_2(4))!$$

На рис. 2.4 кольорами позначено: зелений – оператори, синій – числа і константи, червоний – функції, фіолетовий – спеціальні символи, помаранчевий – спеціальні операції. Оскільки дужки не впливають на числові дані, а лише визначають порядок виконання операцій, вони відсутні у дереві як окремий вузол.

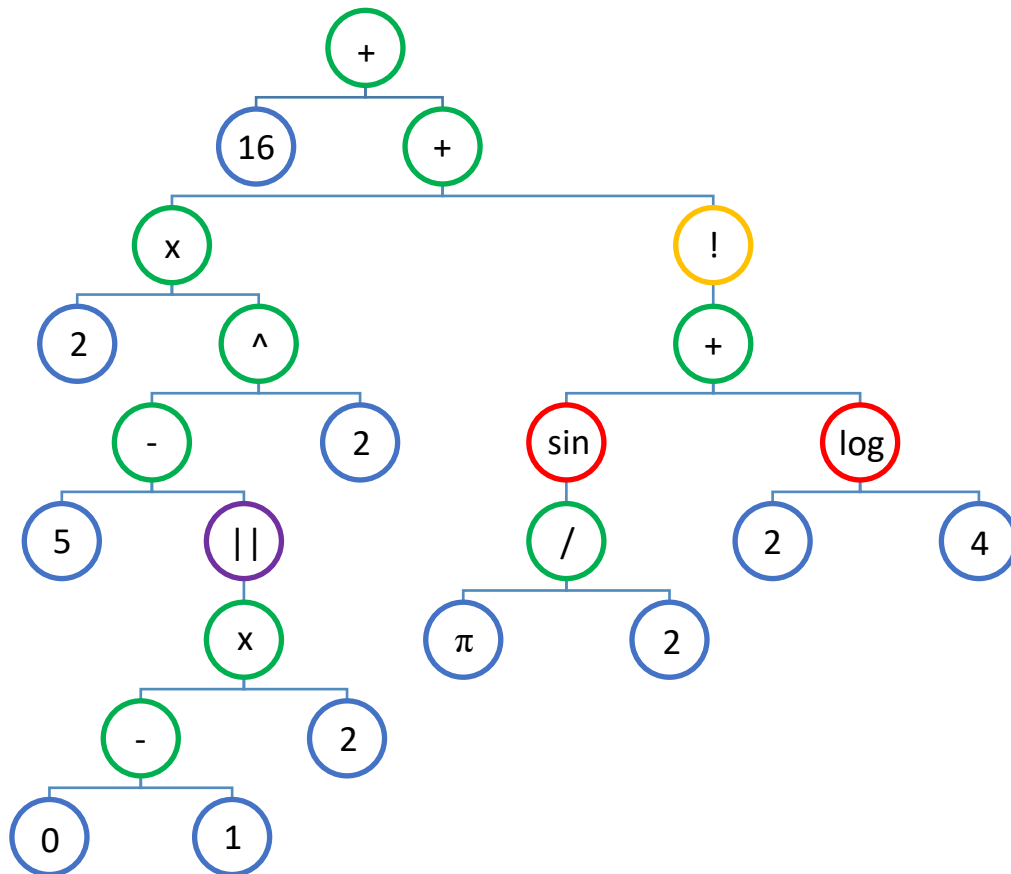


Рис. 2.4. Бінарне дерево математичного виразу

2.4. Опис інтерфейсу програми

Для роботи програми NeuroMath потрібно:

- Комп'ютер під керуванням ОС Windows 7 або вище / Linux / MacOS X.
- Встановлений на комп'ютері інтерпретатор мови Python версії не нижче 3.6.5.

Для встановлення програми потрібно виконати інсталяцію, використовуючи файл NeuroMathInstaller.exe, що розміщено у хмарному сховищі з URL-адресою <https://cutt.ly/IriVqEE>. У процесі інсталяції відбудеться завантаження з сайтів розробників додаткових бібліотек для роботи програми. При цьому потрібно з'єднання з Інтернетом. Назви бібліотек, що завантажуються, відображатимуться в окремому вікні. У результаті на вибраному носії в папці NeuroMath буде розміщено комплект файлів нейронної

мережі, що вже пройшла навчання. Якщо завантаження бібліотек не розпочнеться автоматично, то потрібно запустити на виконання файл `install.py`.

Для запуску програми на виконання призначений файл `MainApp.py`. Після запуску відкривається головне вікно програми (рис. 2.5).

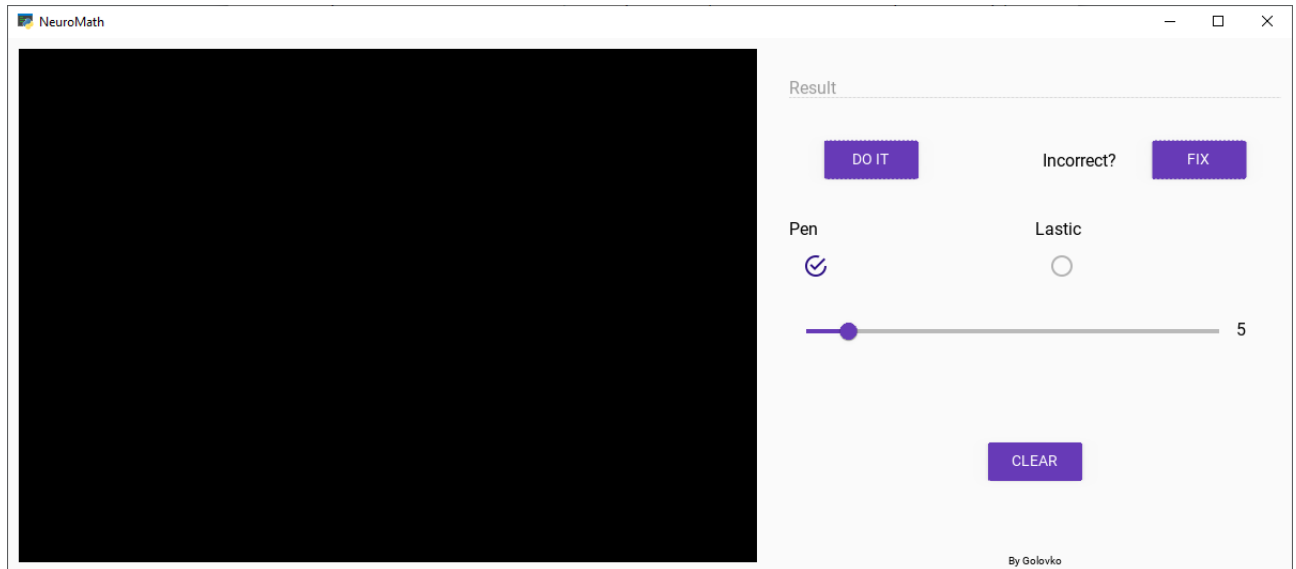


Рис. 2.5. Головне вікно програми **NeuroMath**

Чорне поле в лівій частині вікна призначене для запису математичного виразу. Товщину ліній можна регулювати, використовуючи повзунок у правій частині вікна. Установлення позначки прапорця *Lastic* змінює пензель на гумку для редагування тексту. Для продовження запису слід установити позначку прапорця *Pen*.

Математичний вираз може містити такі символи:

- цифри від 0 до 9,
- крапка для розділення цілої та дробової частини дійсних чисел,
- знаки математичних операцій $+$, $-$, \times , $/$,
- піднесення до степені $^$,
- **квадратний корінь** $\sqrt{}$,
- обчислення остачі від ділення $\%$,
- модуль (абсолютне значення) $||$,
- круглі дужки $(,)$,
- знаки $<$, $>$ для запису нерівності,

- знак = для запису рівняння,
- знак ! для обчислення факторіала числа,
- константи e , π ,
- математичні функції \log , \lg , \ln , \exp , \sin , \cos , \tan , \arcsin , \arccos , \arctan , \sinh , \cosh , \tanh , asinh , acosh , atanh .

Після введення виразу та вибору кнопки *DO IT* починається процес розпізнавання. Для попереднього контролю правильності в окремому вікні буде відображено уведений вираз, у якому символи обведені рамками (рис. 2.1). Після закриття цього вікна відповідь нейронної мережі буде розміщено в полі *Result*. Відповідь містить вираз у тому вигляді, в якому його розпізнала система, та значення виразу, якщо система може його обчислити. Для числових виразів значенням буде дійсне число, для рівнянь та нерівностей – логічне значення *is True* або *is False*.

На рис. 2.6 наведено вигляд вікна програми після виконання розпізнавання введеного математичного виразу – обчислення об’єму кулі з радіусом 10.

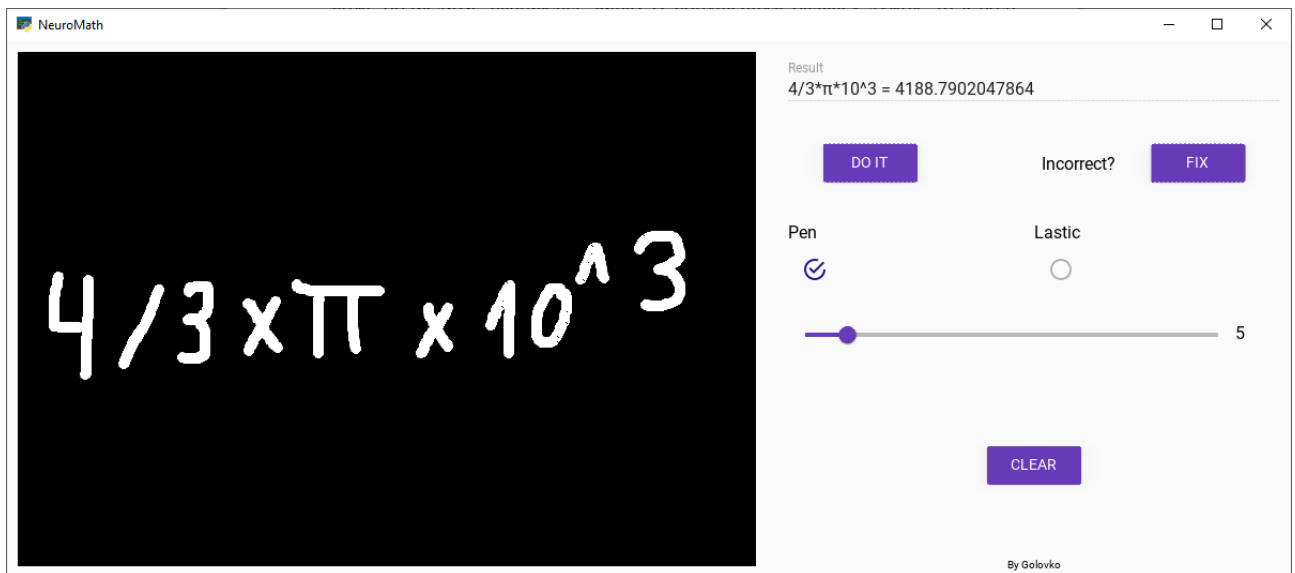


Рис. 2.6. Вікно програми після розпізнавання виразу

Якщо в ході розпізнавання отримано вираз, значення якого система не може обчислити, наприклад, вираз із порушенням балансу дужок, то в полі *Result* буде міститися лише вираз. Його можна відредагувати та вибрати кнопку *FIX* для

продовження навчання нейронної мережі та повторного обчислення значення виразу.

При виборі кнопки *CLEAR* очищується поле введення математичного виразу та поле *Result*, процес розпізнавання можна виконати повторно.

Таким чином, створена система дає змогу обчислювати значення математичних виразів, отриманих шляхом рукописного введення. Модель нейронної мережі може бути легко розширена для розпізнавання інших рукописних символів.

Збільшення точності розпізнавання можна досягти за рахунок змінення структури нейронної мережі: збільшення кількості нейронів у вхідному шарі або збільшення кількості нейронів у прихованих шарах та загального числа прихованих шарів. У кожному випадку потрібне повторне навчання нейронної мережі для налаштування ваги зв'язків між нейронами.

ВИСНОВКИ

1. На виконання першого завдання було розглянуто структуру та принципи функціонування біологічного нейрона, поняття штучного нейрона та штучної нейронної мережі.

Штучні нейронні мережі – математичні моделі, їх програмні чи апаратні реалізації, побудовані за принципом організації та функціонування біологічних нейронних мереж – мереж нервових клітин живого організму.

Основними поняттями в теорії штучних нейронних мереж є нейрон, архітектура мережі та поняття навчання. Штучний нейрон моделює основні функції природного нейрона. Складовими елементами штучного нейрону є вхідні сигнали, суматор та функція активації. Група штучних нейронів, що взаємопов'язані між собою та складають певні прошарки, утворюють штучну нейронну мережу.

2. На виконання другого завдання було проаналізовано поняття архітектури нейронної мережі та методи навчання.

Штучна нейронна мережа може розглядатися як орієнтований граф зі зваженими зв'язками, в якому штучні нейрони є вузлами. За архітектурою зв'язків штучні нейронні мережі поділяються на мережі прямого поширення, в яких графи не мають петель, і мережі зі зворотними зв'язками.

Мережу прямого поширення називаються багат шаровим перцептроном. У ній нейрони розташовані шарами та мають односпрямовані зв'язки між шарами. Багат шарові перцептрони містять три типи прошарків: вхідний, прихований та вихідний. Шар вхідних нейронів отримує вхідні дані. Вихідний пересилає інформацію в зовнішнє середовище. Між цими шарами може бути кілька прихованих шарів. Кожен нейрон прихованого шару отримує сигнали від усіх нейронів попереднього шару. Після виконання операцій над сигналами, нейрон передає свій вихід до всіх нейронів наступного шару.

При навчанні мережі кожна вхідна множина сигналів розглядається як вектор. Навчання здійснюється шляхом послідовного пред'явлення вхідних

векторів з одночасним налаштуванням ваги зв'язків. У процесі навчання ваги зв'язків мережі поступово стають такими, щоб кожний вхідний вектор правильно виробляв вихідний.

Для коригування ваги застосовується метод зворотного поширення похибки. Похибка на вузлах вихідного шару визначається як різниця між бажаним та фактичним вихідним значенням. На прихованих прошарках похибка розподіляється пропорційно вазі кожного зв'язку з подальшим об'єднанням відповідних окремих частин похибки на кожному внутрішньому вузлі.

Таким чином, для розробки нейронної мережі потрібно окреслити її архітектуру, вибрати функцію активізації та метод навчання, підготувати набір навчальних пар та реалізувати виконання основної функції мережі.

3. На виконання третього завдання розроблена математична модель та комп'ютерна реалізація системи нейронних мереж для розпізнавання математичних виразів, отриманих шляхом рукописного вводу.

Змодельована нейронна мережа є багатошаровим перцептроном, має 784 нейрони у вхідному шарі, один прихований шар, використовує сигмоїду як функцію активації, процес навчання з учителем відбувається за алгоритмом зворотного поширення похибки.

Програма NeuroMath розроблена мовою Python. У структурі системи розпізнавання три нейронні мережі, що пройшли навчання на більш ніж 180000 вхідних прикладах. Модель може бути розширена для розпізнання інших рукописних символів. Обчислення значення розпізнаного виразу здійснюється за алгоритмом польської нотації з розкладанням даних у бінарне дерево.

Програма NeuroMath використовується для пояснення сутності нейронних мереж на уроках інформатики в класах профілю інформаційних технологій ліцею «Політ» при Кременчуцькому педагогічному коледжі імені А.С. Макаренка та для обчислення значень математичних виразів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бодянский Е.В. Искусственные нейронные сети: архитектуры, обучение, применения : монографія / Е.В. Бодянский, О.Г. Руденко. – Харьков : Телетех, 2004. – 369 с.
2. Лукін В. Є. Аналіз використання технології штучних нейронних мереж в якості нового підходу до обробки сигналів / В.Є. Лукін // Телекомунікаційні та інформаційні технології. – 2014 – №3 – с. 81-88.
3. Месюра В.І., Ваховська Л.М. Основи проектування систем штучного інтелекту: Навчальний посібник. – Вінниця: ВДТУ, 2000. – 96 с.
4. Руденко, О. Г. Штучні нейронні мережі : навч. посібник / О.Г. Руденко, Є. В. Бодяньський. – Харків : ТОВ «Компанія СМІТ», 2006. – 404 с.
5. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский / Пер. с польск. И. Д. Рудинского. – Москва: Горячая линия – Телеком, 2006. – 452 с.
6. Тарик Рашид. Создаем н
7. ейронную сеть. : Пер. с англ. – СПб. : ООО “Альфа-книга”, 2017. – 272 с.
8. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / Ф. Уоссермен. – М.: Мир, 1992. –184 с.
9. Филиппенко О.И. Биологические, искусственные и нейроавтоматные сети – сравнительный анализ. Ч. 2. Искусственные нейронные сети / О.И. Филиппенко, И.Г. Филиппенко // Восточно-Европейский журнал передовых технологий. – 2005. – № 3/2(15). – С. 87-93.
10. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – Москва : Издательский дом «Вильямс», 2006. – 1104 с.