

УДК 004.9

DOI <https://doi.org/10.36994/2788-5518-2025-01-09-20>

СИНТЕЗ ПЕРЦЕПТРОННОЇ СТРУКТУРИ ДЛЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ЕЛЕМЕНТІВ ТЕКСТУ

Топалов А. М., к.т.н., доц., Національний університет кораблебудування ім. адмірала Макарова, Миколаїв, Україна. <https://orcid.org/0000-0002-0959-4357>, andrii.topalov@nuos.edu.ua

Герасін О. С., к.т.н., доц., Національний університет кораблебудування ім. адмірала Макарова, Миколаїв, Україна. <https://orcid.org/0000-0001-5107-9677>, oleksandr.gerasin@nuos.edu.ua

Мануляк І. З., к.т.н., доц., Івано-Франківський національний технічний університет нафти і газу, Івано-Франківськ, Україна. <https://orcid.org/0000-0002-0072-1532>, manulyak-iryna@ukr.net

Стрілецький Ю. Й., д.т.н. проф., Івано-Франківський національний технічний університет нафти і газу, Івано-Франківськ, Україна. <https://orcid.org/0000-0002-0105-8306>, momental@ukr.net

Анотація. У статті розглядається проблема синтезу нейронної мережі на основі перцептронів кореляційного типу для задачі автоматизованого розпізнавання символів латинського алфавіту. Розроблена структура мережі орієнтована на використання вбудованих засобів програмного середовища Matlab, зокрема його спеціалізованих функцій і бібліотек машинного навчання. Як навчальні дані для формування вхідних і вихідних параметрів нейронної мережі застосовано стандартний генератор `prng`, який створює бінарні растрові представлення латинських літер розміром 5×7 пікселів. Результатом є матриця вхідних даних X і цільових значень T , що формують основу для навчання мережі.

Основна увага приділена синтезу двошарової нейронної мережі прямого поширення сигналу. Перший шар побудовано з 25 прихованих модифікованих перцептронів, що використовують сигмоїдні функції активації, другий шар є лінійним. Це забезпечує підвищену гнучкість порівняно з класичними перцептронами з пороговими функціями активації. Окремо розглядається питання стабілізації випадкової ініціалізації вагових коефіцієнтів для забезпечення відтворюваності результатів навчання. Навчання першої нейронної мережі виконано на еталонних зображеннях літер.

Також детально описано експериментальні дослідження стійкості нейронної мережі до зашумлених вхідних даних. Навчання другої мережі проводилося з урахуванням додаткового шуму у вхідних образах. Аналіз результатів показав, що модель, навчена на зашумлених даних, демонструє менший відсоток помилок розпізнавання при різних рівнях шуму, що підтверджено графічними й числовими результатами. Подано алгоритм обчислення середньоквадратичної помилки (MSE) для оцінювання якості навчання й результати числового моделювання роботи нейромереж, навчених за різної кількості навчальних прикладів.

Результати дослідження можуть бути використані для розробки адаптивних систем розпізнавання тексту, здатних працювати в умовах зовнішніх перешкод, а також як приклад побудови практичних нейронних мереж у середовищі Matlab для навчальних цілей.

Ключові слова: перцептрон, імовірнісні оцінки, цифрові компоненти, нейронне навчання, логічні функції.

SYNTHESIS OF A PERCEPTRONIC STRUCTURE FOR THE PROBLEM OF RECOGNITION OF TEXT ELEMENTS

Peter Topalov, Ph.D., Ass. Prof., Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine. <https://orcid.org/0000-0002-0959-4357>, andrii.topalov@nuos.edu.ua

Oleksandr Gerasin, Ph.D., Ass. Prof., Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine. <https://orcid.org/0000-0001-5107-9677>, oleksandr.gerasin@nuos.edu.ua

Iryna Manulyak, Ph.D., Ass. Prof., Ivano-Frankivsk National Technical University of Oil and Gas, Ivano-Frankivsk, Ukraine. <https://orcid.org/0000-0002-0072-1532>, manulyak-iryna@ukr.net

Yurii Striletskyi, Doctor of Engineering Science Prof. Ivano-Frankivsk National Technical University of Oil and Gas, Ivano-Frankivsk, Ukraine. <https://orcid.org/0000-0002-0105-8306>, momental@ukr.net

Abstract. The article considers the problem of synthesizing a neural network based on correlation-type perceptrons for the task of automated recognition of Latin alphabet characters. The developed network structure is oriented towards the use of built-in tools of the Matlab software environment, in particular its specialized functions and machine learning libraries. As training data for forming the input and output

parameters of the neural network, the standard prprob generator was used, which creates binary raster representations of Latin letters with a size of 5×7 pixels. The result is a matrix of input data X and target values T , which form the basis for training the network.

The main attention is paid to the synthesis of a two-layer neural network of direct signal propagation. The first layer is built from 25 hidden modified perceptrons using sigmoid activation functions, the second layer is linear. This provides increased flexibility compared to classical perceptrons with threshold activation functions. The method of stabilizing random initialization of weight coefficients to ensure reproducibility of training results is considered separately. The first neural network was trained on reference letter images.

The article also describes in detail experimental studies of the stability of the neural network to noisy input data. The training of the second network was carried out taking into account additional noise in the input images. Analysis of the results showed that the model trained on noisy data demonstrates a lower percentage of recognition errors at different noise levels, which is confirmed by graphical and numerical results. An algorithm for calculating the mean square error (MSE) for assessing the quality of training is presented and the results of numerical simulation of neural networks trained on different numbers of training examples are presented.

The results of the study can be used to develop adaptive text recognition systems capable of operating in conditions of external interference, as well as as an example of building practical neural networks in the Matlab environment for training purposes.

Key words: perceptron; probabilistic estimates; digital components; neural learning; logical functions.

Вступ

Однією з основних проблем, що виникає при розв'язанні задач, пов'язаних із розпізнаванням рукописного тексту, є складність формалізації алгоритмічного підходу. Традиційні методи обробки інформації вимагають чіткого визначення правил, за якими система має ідентифікувати кожен символ. Проте в умовах варіативності рукописного письма – зміни нахилу, товщини ліній, індивідуальних особливостей написання – створення універсального алгоритму стикається з численними винятками й унікальними випадками, які складно передбачити на етапі проектування.

З огляду на ці труднощі, значного поширення набув альтернативний підхід на основі штучних перцептронних мереж, які дають змогу вирішувати проблему розпізнавання рукописних символів із використанням навчальних прикладів. Цей підхід ґрунтується на збиранні великого обсягу рукописних символів, які формують навчальну вибірку. На основі цієї вибірки створюється модель, здатна навчатися, виявляючи приховані закономірності, притаманні цим символам.

Нейронна мережа, сприймаючи навчальні приклади, формує власну систему правил для розпізнавання, що не задається явно, а виникає внаслідок узагальнення досвіду. Таким чином, що більше прикладів буде представлено під час навчання, то вища ймовірність того, що модель досягне високої точності в розпізнаванні нових, раніше не бачених символів. Перцептрон, як базова модель штучного нейрона, відіграє фундаментальну роль у структурі та функціонуванні нейронних мереж, що використовуються для задач розпізнавання рукописного тексту. Цей елемент, запропонований Френком Розенблаттом, є першою спробою математичного моделювання біологічного нейрона й поклав початок розвитку сучасних нейромережевих технологій. У контексті розпізнавання рукописних символів, перцептрон виступає як елементарна одиниця обчислення, що дає змогу класифікувати вхідні дані шляхом обчислення лінійної комбінації вхідних сигналів із наступним застосуванням функції активації. Саме завдяки здатності до навчання – зміни вагових коефіцієнтів на основі помилок – перцептрони дають змогу моделі адаптуватися до різноманітних зразків написання символів. Більше того, у багатoshарових нейронних мережах, де прості перцептрони об'єднуються в складні архітектури, забезпечується можливість апроксимації складних функцій і розпізнавання високорівневих патернів. Завдяки цій властивості перцептрони стали невід'ємною частиною алгоритмів глибокого навчання, що демонструють високу ефективність у задачах класифікації, зокрема у сфері оптичного розпізнавання символів (далі – OCR).

Упродовж останніх років число дослідників у галузі розпізнавання рукописного тексту нейронними мережами значно зросло. Це зумовлено широким застосуванням OCR у цифровізації документів, автоматизації вводу даних та освітніх проєктах. Особливо активні дослідження ведуться в Україні й за кордоном: учені орієнтуються на оптимізацію моделей на базі перцептронів, CNN, RNN, Transformer-архітектур. Велику увагу приділяють адаптації підстроювання під особливості мови й рукописні зразки різних культур і мов. Автори [1] запропонували архітектуру на базі лише 1D-CNN шарів, яка дає змогу досягти продуктивності, порівнянної з RNN/Transformer-моделлями. Уведено нову стратегію аугментації «Tiling and Corruption (TACO)» та модуль Squeeze and Excitation, що значно підвищують точність. Модель показала найкращі результати на базі IAM при роботі з обмеженим набором тренувальних даних. У праці [2] запропоновано підхід «Text Perceptron», що є скінчено-регулярною моделлю для розпізнавання тексту нестандартної форми. Завдяки Shape Transform Module забезпечується єдина модель для детекції й розпізнавання зі склесним оптимізаційним фронтом. Досягнуто високої точності на ICDAR і SCUT CTW1500. У роботі [3] модель Self ONN із самоорганізо-

ваними операційними нейронами замінює CNN, використовуючи нелінійну параметризацію нейронів. Використання deformable convolutions забезпечило кращу адаптацію до різноманітності рукописних зразків. Результати на IAM і HADARA80P показали зменшення CER і WER порівняно з класичними CNN-архітектурами. У статті [4] модель DAN об'єднує сегментацію та розпізнавання в єдиний трансформер-базований конвеєр без явних сегментаційних міток. Для покрокового виведення символів зі структурою XML використовується FCN-енкодер і стек Transformer-декодерів. Досягнуто CER = 3,43 % на READ 2016, що є конкурентоспроможним результатом. У праці [5] представлено рекурсивні нейромережні моделі для зменшення розміру моделі без втрати точності. Використано повторне застосування ваг, масштабування ядерних функцій і повторювані шари, що дає змогу зменшити обсяг пам'яті. Показано, що такі моделі зберігають або підвищують точність на HTR-бенчмарках порівняно зі стандартними CNN/RNN. Український дослідник у роботі [6] порівняв щільні нейронні мережі (DNN) і CNN у розпізнаванні рукописних цифр. Виявлено, що DNN втрачають продуктивність зі збільшенням розміру вхідного зображення, тоді як CNN забезпечують сталість і точність. Робота підтверджує важливість використання перцептронів у шарі класифікації та архітектурних рішеннях для інтеграції локальних ознак.

Таким чином, значення перцептронів у нейронних мережах полягає не лише в їхній історичній ролі, а й у практичній ефективності як базового елемента, що забезпечує здатність системи до узагальнення, адаптації та точного розпізнавання складних візуальних структур.

Мета дослідження полягає в розробленні системи розпізнавання символів за допомогою штучної нейронної мережі, реалізованої у вигляді програмного засобу. Передбачається створення програмного засобу, здатного приймати на вхід літери, здійснювати їх обробку за допомогою навченої моделі та повертати показники точності розпізнавання.

Особливу увагу в рамках дослідження планується приділити динаміці навчального процесу, а саме забезпеченню стабільного зростання точності мережі на кожному етапі навчання.

Особливості функціонування перцептрона та вимоги до системи розпізнавання елементів тексту

Перцептрон є однією з фундаментальних моделей штучних нейронних мереж, яка має суттєве аналітичне значення в теорії машинного навчання й обчислювального інтелекту [7]. Його робота ґрунтується на обробці кількох вхідних сигналів, які, як правило, подаються у вигляді двійкових значень. Кожен вхідний сигнал позначається як x_i , де i змінюється від 1 до n , що відповідає кількості входів нейронного елемента.

Основним функціональним елементом класичного перцептрона є так званий пороговий елемент, який здійснює лінійне зважування вхідних сигналів і приймає рішення про активацію виходу на основі порівняння результату цієї операції з фіксованим або адаптивним порогом. Формально вихідний сигнал у цього елемента може набувати лише двох дискретних значень: 0 або 1, що відображає принцип бінарної класифікації [8].

Математичний опис роботи порогового елемента має вигляд:

$$y = \begin{cases} 0, & \text{якщо } \sum_{i=1}^n w_i x_i \geq w_0, \\ 1, & \text{якщо } \sum_{i=1}^n w_i x_i < w_0, \end{cases}$$

де w_i – вагові коефіцієнти, що відповідають значущості кожного входу, а w_0 – величина порога спрацювання елемента.

Першу технічну модель зорового аналізатора, яку розробив Ф. Розенблатт, автор назвав перцептроном (від англійського слова *perception* – сприйняття). Структура первинної моделі Розенблатта включала рецепторний шар S, що складався з 400 фотоелементів, які формували прямокутне рецепторне поле розміром 20×20 елементів. Кожен фотоелемент функціонував як елементарний сенсорний детектор, що перетворював оптичне зображення на електричні сигнали. Інформація з рецепторного шару передавалася на вхід елементів наступного рівня – порогових нейронів перетворюючого шару (елементів A). Усього в моделі реалізовано 512 таких елементів. Кожен нейрон шару A мав по 10 входів, що випадковим чином підключалися до різних фотоелементів рецепторного поля. Половина із цих входів мала збуджуючий характер із коефіцієнтом посилення +1, інша половина – гальмівний характер із коефіцієнтом −1. Порогове значення для активації нейрона встановлювалося рівним нулю, що відповідало бінарній моделі роботи елемента (активація або неактивація). Вихідні сигнали нейронів перетворюючого шару A подавалися на вхід реагуючого нейрона – елемента R, який виконував роль інтегративного вузла, здійснюючи остаточне рішення про розпізнавання вхідної інформації [8–11].

Розглянемо математичну модель функціонування класичного перцептрона, яка стала фундаментом для подальшого розвитку штучних нейронних мереж:

1. Формування вхідного сигналу. На першому етапі в рецепторному полі формується сигнал, який відповідає зовнішньому подразнику. Цей сигнал подається у вигляді векторної величини x . Кож-

не рецепторне закінчення (рецептор) передає досить простий сигнал – або генерує імпульс, або не генерує його. З математичного погляду це означає, що вектор x є бінарним, тобто його компоненти приймають значення лише з множини 0 або 1. Таким чином, рецепторне поле здійснює кодування вхідної інформації у вигляді бінарного вектора.

2. Передача й перетворення сигналів другим шаром нейронів. Сформований бінарний вектор x передається до другого шару нейронів, де відбувається його трансформація у вихідний вектор y , який також є бінарним. Цей процес описується функціональним співвідношенням $y = f(x)$. Характер перетворення визначається двома основними принципами:

а) **пороговий характер обчислень:** трансформація вхідних сигналів здійснюється за допомогою порогових елементів, кожен із яких активується, коли сума вхідних сигналів, помножених на відповідні вагові коефіцієнти, перевищує заданий поріг;

б) **випадкове з'єднання входів:** зв'язки між рецепторами першого шару й пороговими елементами другого шару формуються випадковим чином, що імітує випадкову організацію нейронних зв'язків у біологічних мережах.

3. Класифікація на основі активації вихідних нейронів. Модель передбачає, що перцептрон відносить вхідний вектор до певного поняття j , якщо активується лише j -й реагуючий нейрон, а всі інші нейрони залишаються неактивними. Інакше кажучи, вихідний бінарний вектор $y = (y_1, y_2, \dots, y_m)$ задовольняє таку систему нерівностей:

$$\sum_{i=1}^m w_{i,j} y_i \geq 0,$$

$$\sum_{i=1}^m w_{i,k} y_i < 0, \text{ для всіх } k \neq j.$$

У цих нерівностях $w_{i,k}$ – коефіцієнти посилення k -го реагуючого нейрона.

4. Процес формування понять у структурі перцептрона, запропонованого Ф. Розенблаттом, ґрунтується на коректному налаштуванні вагових коефіцієнтів, що відповідають кожному елементові шару реагуючих нейронів (позначимо цей шар як R). Основною метою є встановлення таких значень ваг, які забезпечують правильне реагування моделі на вхідні сигнали.

Нехай до цього моменту існують деякі ваги елементів R і $w_{1,j}, w_{2,j}, \dots, w_{n,j}$ – ваги j -того елемента. У момент часу t для класифікації на вхід перцептрона надходить сигнал, що описується вектором $x(t)$. Вектор $x(t)$ може або відповідати поняттю j або не відповідати йому. Розглянемо обидва ці випадки.

Випадок перший. У цьому випадку очікуваною реакцією j -го нейрона є активація, тобто правильна відповідь системи полягає в тому, що нейрон збуджується у відповідь на поданий сигнал $x(t)$. Математично це виражається у вигляді такої нерівності:

$$\sum_{i=1}^m w_{i,j} y_i \geq 0.$$

Якщо ваги елемента j забезпечують правильну реакцію на вектор $x(t)$, то вони не змінюються. Якщо ж ваги не забезпечують правильної реакції елемента j , тобто вони такі, що

$$\sum_{i=1}^m w_{i,j} y_i < 0,$$

то ваги елемента j змінюються за правилом:

$$w_{i,j}(t) = w_{i,j}(t-1) + y_i.$$

Випадок другий. У ситуації, коли вхідний вектор $x(t)$ не належить до поняття j , правильна реакція перцептрона полягає у відсутності активації відповідного нейронного елемента. Це означає, що елемент j має залишатися неактивним у відповідь на цей стимул. Математично така умова виражається у вигляді такої нерівності:

$$\sum_{i=1}^m w_{i,j} y_i \geq 0,$$

Якщо поточні значення ваг забезпечують правильну реакцію елемента j , тобто коли наведена нерівність виконується, ваги залишаються незмінними. Однак якщо вага елемента j не забезпечує правильної реакції, тобто виконується протилежна умова:

$$\sum_{i=1}^m w_{i,j} y_i \geq 0,$$

то ваги змінюються відповідно до правила:

$$w_{i,j}(t) = w_{i,j}(t-1) - y_i.$$

При здійсненні навчання аналогічно змінюються ваги всіх R-елементів перцептрона. Таким чином, коригування вагових коефіцієнтів усіх R-елементів забезпечує узгоджену адаптацію всієї мережі, що сприяє підвищенню точності класифікації.

Вигляд перцептрона та зібраної з перцептронів структури показаний на рис. 1. Дані зображення перцептрона й перцептронної структури вважаються загальними для першочергового етапу проектування систем розпізнавання [12].

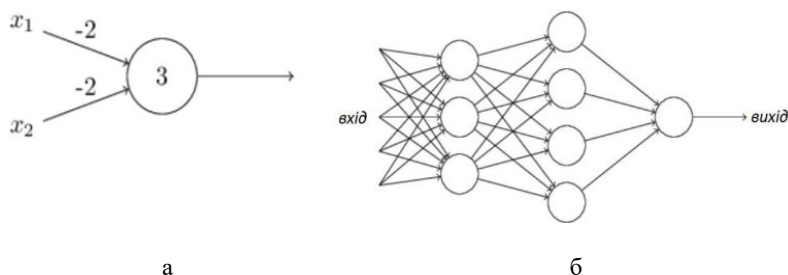


Рис. 1. Перцептрон і його структура: а) окремий перцептрон з ваговими коефіцієнтами та загальним зміщенням; б) математична модель мережі перцептронів

Аналізуючи схему, представлену на рисунку 1, б, можемо спостерігати наявність декількох вертикальних груп елементів, які утворюють структуровані ряди, або шари перцептронів [13]. Така багаторівнева структура є характерною рисою багатшарових нейронних мереж, де кожен наступний шар здійснює обробку інформації, яка є результатом роботи попереднього шару.

Перший шар перцептронів відповідає за первинний рівень прийняття рішень, здійснюючи обчислення на основі зваженого підсумовування вхідних сигналів. Кожен нейрон цього шару аналізує сирі дані, що надходять безпосередньо із зовнішнього середовища або рецепторних елементів. Другий шар перцептронів приймає на вхід результати обробки, здійсненої першим шаром, і виконує більш складну обробку, комбінуючи й переосмислюючи ці проміжні результати. Таким чином, другий шар здатен виділяти складніші закономірності та зв'язки, ніж перший. Аналогічно на третьому рівні (третьому шарі) відбувається ще глибше узагальнення інформації, що дає змогу нейронній мережі формувати висновки ще вищого рівня абстракції. Цей підхід демонструє ієрархічну природу обробки даних у багатшарових нейронних мережах, де кожен наступний шар виконує дедалі складніші функції аналізу. Окремо варто зазначити, що в класичному вигляді перцептрон, як модель окремого нейрона, має лише один вихідний сигнал, що відповідає його бінарному рішенням – активації або відсутності активації. Однак на схемі, представленій на рисунку 1, для наочності показано, що цей вихід одночасно може бути підключений до кількох нейронів наступного шару. Таким чином, формується уявлення про множинне використання одного вихідного сигналу: один вихідний сигнал перцептрона може виступати як кілька вхідних сигналів для різних нейронів наступного шару. Це не означає наявності декількох фізичних виходів у перцептрона, а лише відображає можливість розгалуження інформаційного потоку у структурі мережі.

Отже, наведена модель демонструє принцип пошарової обробки інформації, де кожен шар виконує функцію узагальнення й ускладнення прийнятих рішень, а вихідні сигнали кожного нейрона можуть одночасно впливати на декілька наступних елементів мережі, формуючи таким чином складну багаторівневу систему взаємозв'язків.

Виконання досліджень

Розглянемо задачу розпізнавання будь-якого із символів латинського алфавіту за допомогою перцептронної нейронної мережі, яка створюється та навчається за допомогою вбудованих функцій Matlab [14]. Для вирішення цієї задачі доцільно розробити відповідну методику, яка містить такі кроки:

1. Генерування масиву вхідних даних (букв латинського алфавіту). Для цього необхідно запустити вбудований скрипт Matlab:

```
[X, T] = prprob;
```

Скрипт prprob визначає матрицю X з 26 колонками, по одній для кожної букви алфавіту. Кожна колонка має 35 значень, які можуть дорівнювати або 1 або 0. Кожен стовпець із 35 значень являє собою растрове зображення букви 5x7. Матриця T – це одинична матриця розміром 26x26, яка відображає 26 вхідних векторів у 26 класів.

Необхідно зазначити, що для роботи команди prprob необхідно встановлювати tollbox deep learn або повністю задавати її як функцію в коді програми перед викликом.

2. Вибір і побудова зображення еталонної літери для розпізнавання (N – номер літери):

```
plotchar (X (:, N));
```

У результаті виконання цієї команди отримується зображення еталонної літери у вигляді бітовій карти.
 3. Створення першої двохарової перцептронної нейронної мережі прямого зв'язку з 25 прихованими нейронами для класифікації букв латинського алфавіту:
`setdemorandstream(pi);`
`net1 = feedforwardnet(25);`
`view(net1)`

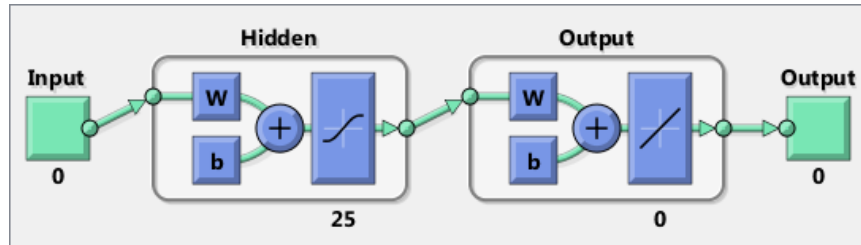


Рис. 2. Ініційована нейронна мережа

Оскільки нейронна мережа ініціюється випадковими початковими вагами, результати після навчання будуть незначно відрізнятися кожного разу після запуску скрипту. Щоб позбутися цієї випадковості, випадкова стала за допомогою функції *setdemorandstream* встановлюється константою. У результаті виконання скрипту отримується зображення нейронної мережі прямого поширення, що наведено на рис. 2. Ця мережа містить 2 шари, причому модифіковані перцептрони першого шару містять сигмоїдні функції активації, а другого – лінійні. Такий підхід дає змогу розширити межі застосування нейронних мереж на базі класичних перцептронів з пороговими функціями активації.

4. Навчання створеної нейронної мережі за допомогою функції *train*:

```
net1.divideFcn = "";
net1 = train(net1,X,T,nnMATLAB);
```

Функція *train* ділить вхідні дані на три масиви – для навчання, валідації (перевірки) і тестування (випробування) нейронної мережі. Навчальний набір використовується для оновлення вагових коефіцієнтів мережі. Валідаційний набір даних потрібен для зупинки мережі, перш ніж відбудеться перенасичення навчальними даними. Набір для тестування виступає як повністю незалежна міра того, наскільки добре можна очікувати виконання поставленої задачі мережею на нових вхідних сигналах. Навчання припиняється, коли мережа вже не може покращити свої результати за рахунок навчальних або валідаційних наборів даних. Властивість *divideFcn* об'єкта нейронної мережі визначає, яким чином дані автоматично поділятимуться на ці три набори (підмножини) перед навчанням. Для першої нейронної мережі вся вибірка використовується для навчання. Структура навченої нейронної мережі наведена на рисунку 3.

Навчання першої нейронної мережі тривало 11 ітерацій (4 с) методом Левенберга-Марквардта за критерієм мінімуму середньоквадратичної помилки ($mse_{min} = 1,2 \cdot 10^{-16}$).

Далі доцільно дослідити, як мережа буде розпізнавати не лише ідеально сформовані літери, а і їх зашумлені версії. Тому необхідно створити й навчити другу мережу на зашумлених даних і порівняти її здатність до узагальнення з першою мережею.

5. Створення масиву вхідних даних з додатковим забрудненням сигналів:

```
numNoise = 30;
Xn = min(max(repmat(X,1,numNoise)+randn(35,26*numNoise)*0.2,0),1);
Tn = repmat(T,1,numNoise);
```

Цей скрипт створює набори даних із 30 копій кожної букви з масиву X, що додатково забруднені шумом. Причому забрудненість шумом виражається в тому, що значення відповідних клітин масиву представлення частини літери (5x7) обмежені на проміжку між 0 і 1. Також визначено відповідні цілі Tn.

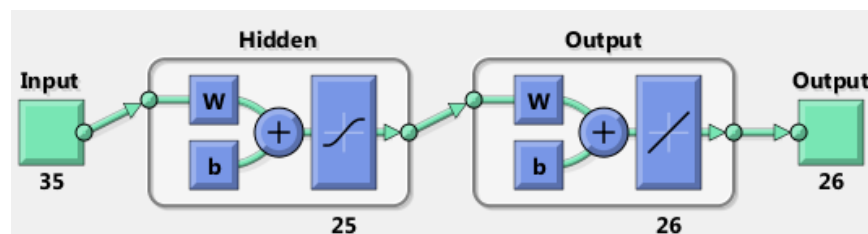


Рис. 3. Структура навченої перцептронної нейронної мережі для класифікації букв латинського алфавіту

6. Побудова зображення еталонної букви, забрудненої шумом:

```
figure
plotchar(Xn(:,N));
```

Порівняння еталонного зображення й забрудненого шумом зображення для літери «А» наведено на рис. 4 (результати виконання кроків 2 та 6 методики).

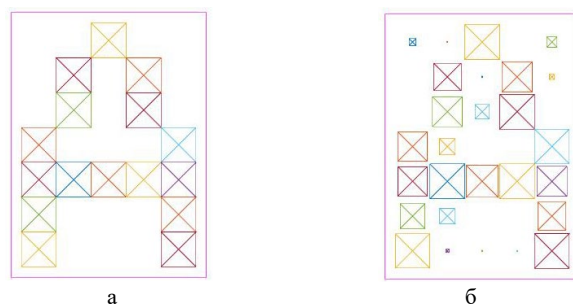


Рис. 4. Візуалізація літер у вигляді бітових карт:
а – еталонне зображення; б – забруднене шумом зображення

7. Створення й навчання нейронної мережі для роботи з забрудненими даними:

```
net2 = feedforwardnet(25);
net2 = train(net2,Xn,Tn,nnMATLAB);
```

Навчання другої нейронної мережі тривало 28 ітерацій (101 с) методом Левенберга-Марквардта за критерієм мінімуму середньоквадратичної помилки ($mse_{min} = 1,345 \cdot 10^{-3}$).

8. Аналіз роботи обох синтезованих нейронних мереж (першої та другої) для 30 наборів зашумлених даних за допомогою такого скрипту:

```
noiseLevels = 0:.05:1;
numLevels = length(noiseLevels);
percError1 = zeros(1,numLevels);
percError2 = zeros(1,numLevels);
for i = 1:numLevels
    Xtest = min(max repmat(X,1,numNoise)+randn(35,26*numNoise)*noiseLevels(i),0,1);
    Y1 = net1(Xtest);
    percError1(i) = sum(sum(abs(Tn-compet(Y1))))/(26*numNoise*2);
    Y2 = net2(Xtest);
    percError2(i) = sum(sum(abs(Tn-compet(Y2))))/(26*numNoise*2);
end
figure
plot(noiseLevels,percError1*100,'--',noiseLevels,percError2*100);
title('Percentage of Recognition Errors');
xlabel('Noise Level');
ylabel('Errors');
legend('Network 1','Network 2','Location','NorthWest');
```

Результат порівняння роботи двох нейронних мереж для різних рівнів шуму наведений на рисунку 5. Виходячи із цього зображення (рис. 5), перша мережа, навчена без шуму, має більші значення помилок через шум, ніж друга мережа, яка була навчена з шумом. Зроблено висновок, що при вирішенні завдання розпізнавання елементів тексту в умовах присутності шуму доцільно проводити навчання нейронної мережі одразу на основі зашумлених даних.

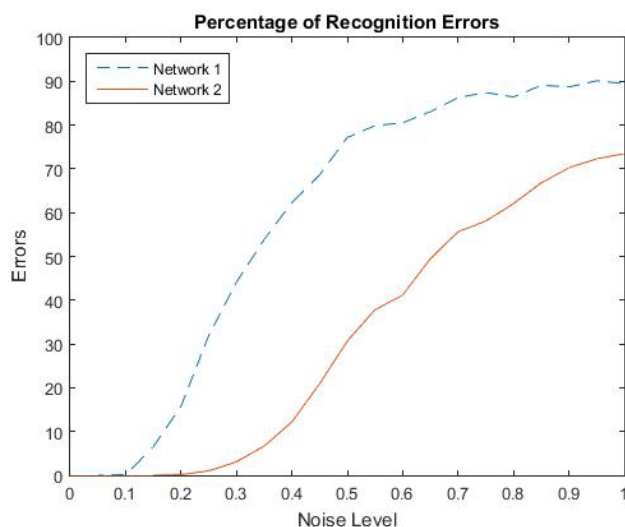


Рис. 5. Порівняння помилок розпізнавання символів першою та другою перцептронними нейронними мережами

9. Визначення кількості наборів зашумлених даних, необхідних для навчання перцептронної нейромережі.

Для визначення необхідної кількості копій кожної літери необхідно виконати кроки 5–8 навчання другої мережі на різній кількості наборів вхідних зашумлених даних. Фрагмент результатів моделювання перцептронних нейромереж, навчених на різній кількості наборів вхідних даних, наведені в таблиці 1, де MSE_k – показник середньоквадратичної помилки, розрахований за виразом:

$$MSE_k = (1/\text{length}(\text{percError2}))(\text{percError2} - MSE_0)^2;$$

де MSE_0 – еталонне значення помилки, MSE_0 = 0.

Таблиця 1
Результати моделювання роботи нейромереж

Кількість наборів зашумлених вхідних даних, k	Кількість ітерацій навчання	Час навчання, с	Помилка валідації, 10 ⁻³	MSE_k
10	15	22	9,024	2,57
20	25	61	2,749	1,97
30	28	101	1,345	2,08
40	22	106	2,538	1,71
50	32	180	0,962	2,36

Виходячи з отриманих результатів (таблиця 1), для вирішення задачі розпізнавання елементів тексту більш придатною є перцептронна нейронна мережа, для навчання якої використовувалося по 40 копій зашумлених літер. Перевагою цього набору також є невелика кількість ітерацій і прийнятний час навчання порівняно з іншими наборами.

Висновок

У роботі наведено методику синтезу перцептронної структури для задачі розпізнавання елементів тексту, написаного латиницею. Розглянуто механізми генерації вхідних навчальних даних для еталонних і зашумлених літер у векторному форматі, який придатний для подальшої обробки нейронною мережею, показано спосіб їх візуалізації. Отримано двошарову структуру нейронної мережі прямого поширення з модифікованими перцептронами, у яких, замість порогових функцій активації, використані сигмоїдальні (перший, прихований, шар) і лінійні (другий, вихідний шар). Далі виконано навчання двох таких структур перцептронних мереж: а) на еталонних даних і б) на даних, які містять шум. Реалізовано процес тестування навчених мереж на даних із різним рівнем шуму, обчислено відсоток помилок розпізнавання для кожної мережі й побудовано графіки залежності помилки від рівня шуму. Аналіз ефективності роботи двох мереж показав вищу пристосованість (менший відсоток помилок) до розпізнавання зашумлених даних перцептронною мережею, яка навчена на зашумлених даних. Крім того, визначено необхідну кількість зашумлених копій кожної літери для навчання перцептронної нейронної мережі, щоб мінімізувати середньоквадратичну помилку.

Література

1. Chaudhary K., Bali R. Easter 2.0: Improving convolutional models for handwritten text recognition. *Computer Vision and Pattern Recognition*. 2022. 12 p.
2. Qiao L., Tang S., Cheng Z., Xu Y., Niu Y., Pu S., Wu F. Text Perceptron: Towards End-to-End Arbitrary-Shaped Text Spotting. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020. P. 11899–11907.
3. Hassen Mohammed H., Malik J., Al-Madeed S., Kiranyaz S. 2D Self-Organized ONN Model For Handwritten Text Recognition. *Computer Vision and Pattern Recognition*. 2022. 12 p.
4. Coquenat D., Chatelain C., Paquet T. DAN: A Segmentation-free Document Attention Network for Handwritten Document Recognition. *Computer Vision and Pattern Recognition*. 2022. 13 p.
5. Mas-Candela E., Calvo-Zaragoza J. Exploring recursive neural networks for compact handwritten text recognition models. *International Journal on Document Analysis and Recognition*. 2024. № 27(3). P. 213–223.
6. Матвійчук А. М. Аналіз архітектур нейронних мереж для розпізнавання рукописних цифр. *Телекомунікаційні та інформаційні технології*. 2023. № 4. С. 118–127. DOI: 10.31673/2412-4338.2023.041827.
7. Гороховатський О. В., Тесленко О. В. Розпізнавання зображень літер із використанням проєкцій та багатозарового перцептронну. *Системи обробки інформації*. 2017. Вип. 2. С. 163–167.
8. Jeng J.-H., Hsieh J.-G. Pathways to Machine Learning and Soft Computing. EHGBooks, 2018. 397 p.
9. Khan G. M. Evolution of Artificial Neural Development: In Search of Learning Genes. Springer, New York, 2018. 142 p.
10. McKeon R. Neural Networks for Electronics Hobbyists. Apress, 2018. 139 p.
11. Руденко О. Г., Бодяньський Є. В. Штучні нейронні мережі Н навчальний посібник для студентів вищих навчальних закладів. К іїв : СМІТ, 2006. 404 с.
12. Chapter 1: Using neural nets to recognize handwritten digits. Electronic resource. <http://neuralnetworksanddeeplearning.com/chap1.html>.
13. Chollet F. Deep Learning with Python. 2018. 384 p.
14. Коробко О. В. Комп'ютеризовані системи штучного інтелекту : методичні вказівки до лабораторних робіт. Миколаїв : НУК, 2014. 76 с.

References

1. Chaudhary K., Bali R. Easter2.0: Improving convolutional models for handwritten text recognition. *Computer Vision and Pattern Recognition*, 2022. 12 p.
2. Qiao L., Tang S., Cheng Z., Xu Y., Niu Y., Pu S., Wu F. Text Perceptron: Towards End-to-End Arbitrary-Shaped Text Spotting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 11899–11907
3. Hassen Mohammed H., Malik J., Al-Madeed S., Kiranyaz S. 2D Self-Organized ONN Model For Handwritten Text Recognition. *Computer Vision and Pattern Recognition*, 2022. 12 p.
4. Coquenat D., Chatelain C., Paquet T. DAN: A Segmentation-free Document Attention Network for Handwritten Document Recognition. *Computer Vision and Pattern Recognition*, 2022. 13 p.
5. Mas-Candela E., Calvo-Zaragoza J. Exploring recursive neural networks for compact handwritten text recognition models. *International Journal on Document Analysis and Recognition*, 2024. 27(3): 213–223.
6. Matviichuk A. M. Analiz arkhitektury neironnykh merezh dlia rozpoznavannia rukopysnykh tsyfr. *Telecommunications and Information Technologies*, 2023. № 4, 118–127. DOI: 10.31673/2412-4338.2023.041827 [in Ukrainian].
7. Gorokhovatskyi O. V., Teslenko O. V. Recognition of letter images using projections and a multilayer perceptron. *Information Processing Systems*, 2017. Issue 2. 163–167 [in Ukrainian].
8. Jeng J.-H., Hsieh J.-G. Pathways to Machine Learning and Soft Computing. EHGBooks, 2018. 397 p.
9. Khan G. M. Evolution of Artificial Neural Development: In Search of Learning Genes. Springer, New York, 2018. 142 p.
10. McKeon R. Neural Networks for Electronics Hobbyists. Apress, 2018. 139 p.
11. Rudenko O. G., Bodiansky E. V. Artificial neural networks Textbook for students of higher educational institutions K.: SMIT, 2006, 404 p. [in Ukrainian].
12. Chapter 1: Using neural nets to recognize handwritten digits. Electronic resource. <http://neuralnetworksanddeeplearning.com/chap1.html>.
13. Chollet F. Deep Learning with Python. 2018. 384 p.
14. Korobko O. V. Kompiuteryzovani systemy shtuchnoho intelektu: Metodychni vkazivky do laboratornykh robot. Mykolaiv: NUK. 2014. 76 p. [in Ukrainian].

Дата першого надходження рукопису до видання: 16.05.2025
Дата прийнятого до друку рукопису після рецензування: 18.06.2025
Дата публікації: 25.07.2025