

## РОЗДІЛ 2

# ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ

УДК 004.032.26

А.В. Кожевников<sup>1</sup>, І.М. Удовик<sup>1</sup>

<sup>1</sup>Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

### СТВОРЕННЯ ВІДКРИТОЇ НЕЙРОННОЇ МЕРЕЖІ БІНАРНОГО КЛАСИФІКАТОРА ЗАСОБАМИ СИСТЕМИ MATHCAD

**Анотація.** В роботі наведена математична модель нейронної мережі бінарного класифікатора. Отримані аналітичні співвідношень, які пов'язують коефіцієнти лінійної дискримінантної функції бінарного класифікатора з вагами синапсів навченої нейронної мережі класифікатора. Вперше за допомогою засобів системи MathCAD створена програмна реалізація нейронної мережі бінарного класифікатора. Досліджені залежності ваг синапсів і нейронної мережі і її похибки від кількості циклів навчання.

**Ключові слова:** нейронна мережа, персептрон, синапс, бінарний класифікатор.

**Вступ.** В теперішній час штучні нейронні мережі набувають все більшої популярності і сфера їх використання також розширюється. Ось тільки невеликий перелік галузей де практично застосовуються нейронні мережі:

- Введення і обробка інформації. Розпізнавання текстів, що подані у вигляді растрових графічних даних, розпізнавання голосових команд, голосове введення тексту.
- Безпека. Розпізнавання обличчя і різних біометричних даних, аналіз трафіку в мережі, виявлення підробок.
- Інтернет. Таргетингова реклама, капча, блокування спаму, асоціативний пошук інформації.
- Зв'язок. Маршрутизація пакетів, стиснення відеоінформації, збільшення швидкості кодування і декодування інформації.

При підготовці фахівців з ІТ-спеціальностей нейронні мережі стали невід'ємною частиною таких дисциплін, як: “Системи штучного інтелекту”, “Інтелектуальний аналіз даних”, “Методи та інформаційні технології обробки великих даних (Big Data)”.

Наразі створені численні засоби для проектування, моделювання, розробки та візуалізації нейронних мереж. Перш за все слід відзначити Neural Network та Deep Learning MATLAB Toolboxes [1, 2], зростаючою альтернативою яким виступають програмні засоби, що мають Pythonic інтерфейс, такі як [3]: Scikit Learn, NeuroLab, Ffnet, Scikit Neural Network Librarys. Також засоби роботи з нейронними мережами реалізовані на деяких

аналітичних платформах, наприклад WEKA [4], Deductor Academic [5] та спеціалізованих програмних засобах: ST Neural Networks, BrainMaker, NuroNet.

Вказані продукти виступають потужними засобами реалізації практичних задач з використанням нейронних мереж, але, переважним чином, розглядають і використовують мережу як “чорну скриньку”. Топологія та характеристики мережі визначаються у першому випадку за допомогою заздалегідь визначених параметрів функцій відповідних бібліотек, а у другому – інструментів програмних засобів. Відповідно обмеженими є і можливості дослідження еволюції характеристик мережі в процесі її навчання.

В той же час, при першому знайомстві студентів з нейронними мережами бажано мати програмні реалізації хоча б найпростіших мереж, які мають відкриті коди до того ж прозоро пов’язаних з математичними моделями навчання та функціонування. Рішення таких завдань навіть з використанням спеціалізованих мов MATLAB чи Python не кажучи вже про популярні C та Java є трудомістким і не забезпечує належної наочності створеного програмного коду. Найкращим програмним засобом, на наш погляд, в цьому випадку є власна система комп’ютерної алгебри MathCAD [6], або її подальший розвиток MathCAD Prime [7]. Основна відмінність цих систем від аналогічних полягає у системі графічних шаблонів для введення програмних речень. Шаблони відтворюють символіку, що використовується у математиці, що робить програмний код компактним та наочним. MathCAD використовує власну скриптову мову. Для організації найпростіших циклів використовуються ранжовані змінні, більш складні алгоритми реалізуються у рамках програмних блоків.

Найпростішим завданням, яке може бути вирішене за допомогою нейронної мережі є бінарна класифікація лінійно роздільних об’єктів, які характеризуються двома ознаками.

**Мета дослідження.** Полягає у створенні за допомогою засобів системи MathCAD програмної реалізації нейронної мережі бінарного класифікатора, що має відкритий код, який прозоро пов’язаний з математичною моделлю мережі і дозволяє гнучко задавати параметри мережі і наочно відтворювати процеси формування навчальної та тестової вибірок, навчання та тестування мережі.

**Основний зміст роботи.** Бінарний класифікатор лінійно роздільних об’єктів, які характеризуються двома ознаками може бути побудований на основі перцептрона, що складається з одного нейрона. Нейрон має вхід зсуву, два інформаційних входи і один вихід (рис.1).

Дані представляють собою крапки числової площини з координатами  $(x_1, x_2)$ , які рівномірно розподілені в квадраті  $x_1, x_2 \in (0,1)$ . Навчальна вибірка окрім матриці вказаних вхідних даних сигналів містить вектор вихідних сигналів, які формуються за правилом

$$Y_k = \begin{cases} 0 & \text{при } X_{k2} < aX_{k1} + b \\ 1 & \text{при } X_{k2} \geq aX_{k1} + b \end{cases}, \quad a, b \in (0,1), \quad (1)$$

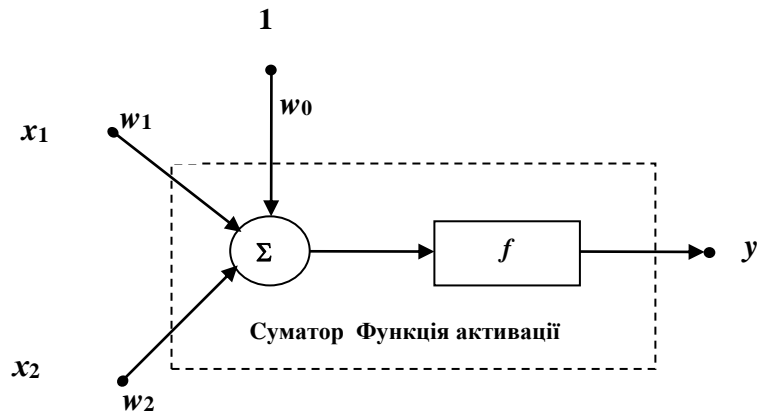


Рис. 1. Схема персептрона бінарного класифікатора лінійно роздільних об'єктів, які характеризуються двома ознаками

Тобто, якщо крапка знаходиться нижче прямої  $x_2 = ax_1 + b$ , вона відноситься до класу 0, інакше – до класу 1. Тут і далі прийняті такі позначення  $\{w_0^n\}$ ,  $\{w_i^n\}_{i=1, \dots, imax}$  – ваги синапсів на  $n$ -му кроці навчання,  $imax=2$  – кількість входів персептрону без входу зсуву,  $\{X_{ki}\}_{k=1, \dots, kmax}$  – навчальна вибірка входних сигналів,  $kmax$  – об'єм вибірки, який може бути довільним,  $\{Y_k\}_{k=1, \dots, kmax}$  – навчальна вибірка вихідних сигналів. При навчанні мережі здійснюється її тренування на кожній варіанті навчальної вибірки впродовж  $nmax$  циклів. Фактичний вихідний сигнал персептрону на  $n$ -му циклі навчання при подачі на вхід входних сигналів  $k$ -ї варіанти навчальної вибірки при цьому складає

$$y_k^n = f\left(w_0 + \sum_{i=1}^{imax} w_i^n X_{ki}\right). \quad (2)$$

Функція активації  $f$  при цьому є пороговою, а вихідний сигнал – бінарним.

Похибка мережі на  $n$ -му циклі навчання визначається як сума за всіма варіантами навчальної вибірки модулів різниць між очікуваним та фактичним значенням вихідного сигналу мережі

$$\varepsilon^n = \sum_{k=1}^{kmax} |Y_k - y_k^n|. \quad (4)$$

При цьому умовою припинення навчання може бути як досягнення необхідного значення похибки, так і просте обмеження кількості ітерацій певною величиною  $nmax$ . Правила налаштування ваг синапсів залежить від вида мережи. Для персептрона може бути використане співвідношення [8]

$$\begin{cases} w_0^{n+1} = w_0^n + \sum_{k=1}^{kmax} \eta (Y_k - y_k^n) \\ w_i^{n+1} = w_i^n + \sum_{k=1}^{kmax} \eta (Y_k - y_k^n) X_{ki} \text{ при } 1 \leq i \leq imax \end{cases}, \quad (5)$$

де  $\eta \in (0,1)$  – швидкість навчання. У якості початкових значень ваг синапсів  $w_i^0$  беруться випадкові величини, які рівномірно розподілені на сегменті  $(-0.1,0.1)$ . Наведене вище правило налаштування ваг синапсів носить назву дельта-правила і є узагальненням правил Хебба. Останні були сформульовані для персептрона, який працює з бінарними вхідними і вихідними сигналами і мають вигляд:

- якщо сигнал персептрона невірний, і дорівнює нулю, то необхідно збільшити ваги тих входів, на які була подана одиниця;
- якщо сигнал персептрона невірний і дорівнює одиниці, то необхідно зменшити вагу тих входів, на які була подана одиниця.

Дійсно, якщо вихідний сигнал бінарного класифікатора невірний, і дорівнює нулю  $y_k^n = 0$ ,  $Y_k - y_k^n = 1$  і, з урахуванням невід’ємності величини  $X_{ki}$ , застосування дельта-правила веде до збільшення ваг всіх синапсів, що викликає виправлення значення вихідного сигналу на наступному циклі навчання,  $y_k^{n+1} = 1$ , або принаймні збільшення сигналу на виході суматора, який є аргументом функції активації. Навпаки, якщо вихідний сигнал бінарного класифікатора невірний, і дорівнює одиниці  $y_k^n = 1$ ,  $Y_k - y_k^n = -1$ , застосування дельта-правила веде до зменшення ваг всіх синапсів, що викликає виправлення значення вихідного сигналу на наступному циклі навчання,  $y_k^{n+1} = 0$ , або принаймні збільшення сигналу на виході суматора.

При тестуванні мережі вихідний сигнал навченої мережі, який виникає при поданні на її входи сигналу  $\{x_i\}_{i=1, \dots, imax}$  становить

$$y^{nmax} = f\left(w_0 + \sum_{i=1}^{imax} w_i^{nmax} x_i\right). \quad (6)$$

Об’єм тестової вибірки  $nmax$  може бути довільним. Сенс елементів тестової вибірки  $\{X_{tmi}\}_{m=1, \dots, nmax}$  та  $\{Y_{tm}\}_{m=1, \dots, nmax}$  такий же як і у навчальній.

Зауважимо, що, оскільки умовою віднесення крапки, яка класифікується, до класу 1 є  $x_2 \geq ax_1 + b$ , а умовою того, що вихідний сигнал мережі дорівнюватиме 1 –  $w_0^{nmax} + w_1^{nmax} x_1 + w_2^{nmax} x_2 \geq 0$ , для синапсів навченої мережі повинні виконуватися умови

$$a \approx -\frac{w_1^{nmax}}{w_2^{nmax}}, \quad b \approx -\frac{w_0^{nmax}}{w_2^{nmax}}. \quad (7)$$

Програмна реалізація нейронної мережі бінарного класифікатора засобами системи MathCAD включає підготовчий фрагмент у якому задаються параметри мережі, формуються навчальна та тестова вибірки, визначаються початкові ваги синапсів, а також програмні блоки навчання та тестування мережі. Вхідними даними підготовчого блока відповідно виступають  $a, b, \eta, k_{max}, m_{max}, n_{max}$ . На його виході формуються значення  $\{X_{ki}\}_{k=1, \dots, k_{max}, i=1, \dots, i_{max}}$ ,

$\{Y_k\}_{k=1, \dots, k_{max}}, \{X_{tmi}\}_{m=1, \dots, m_{max}, i=1, \dots, i_{max}}, \{Y_{tm}\}_{m=1, \dots, m_{max}}, \{w_0^0\}, \{w_i^0\}_{i=1, \dots, i_{max}}$  На виході

програмного блока навчання отримуються значення  $\{w_0^{n_{max}}\}, \{w_i^{n_{max}}\}_{i=1, \dots, i_{max}}$  та  $\varepsilon^{n_{max}}$ , а на виході програмного блока тестування – значення  $\{y_m^{n_{max}}\}_{m=1, \dots, m_{max}}$ , які відповідають елементам тестової вибірки.

Мережа швидко навчається, для отримання нульової похибки потрібно порядку десяти циклів навчання, при цьому ваги синапсів збігаються до значень, які задовольняють співвідношенням (7), що свідчить про правильність функціонування мережі рис. 2.

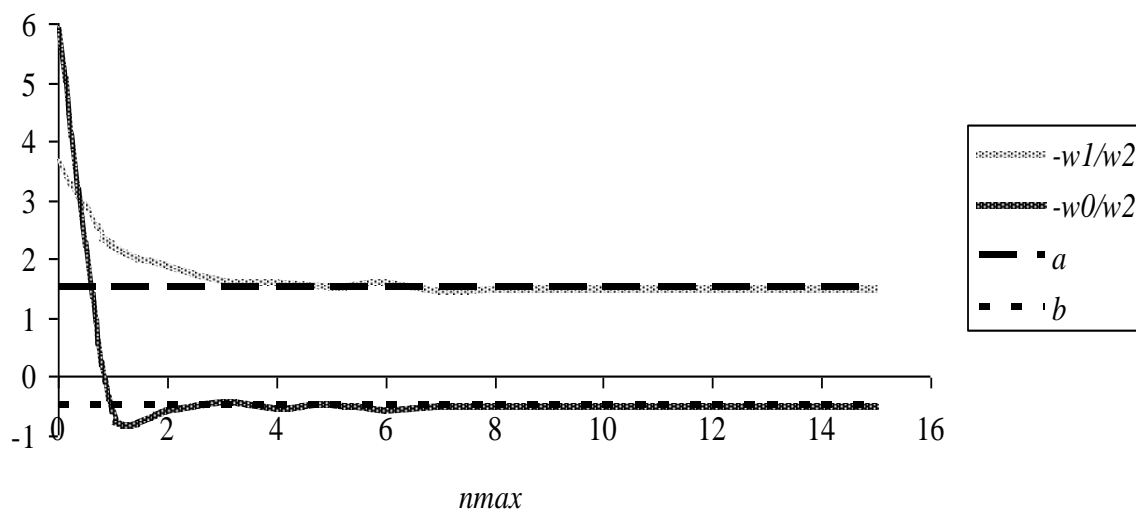


Рис. 2. Графіки залежностей ваг синапсів мережі від кількості циклів навчання

**Наукова новизна** полягає у отриманні аналітичних співвідношень, які пов'язують коефіцієнти лінійної дискримінантної функції бінарного класифікатора з вагами синапсів навченої нейронної мережі класифікатора.

**Висновки.** В роботі наведена математична модель нейронної мережі бінарного класифікатора. Отримані аналітичні співвідношень, які пов'язують коефіцієнти лінійної дискримінантної функції бінарного класифікатора з вагами синапсів навченої нейронної мережі класифікатора. Вперше за допомогою засобів системи MathCAD створена програмна реалізація нейронної

мережі бінарного класифікатора. Програма має відкритий код, який прозоро пов'язаний з математичною моделлю мережі і дозволяє гнучко задавати параметри мережі і наочно відтворювати процеси формування навчальної та тестової вибірок, навчання та тестування мережі. Досліджені залежності ваг синапсів і нейронної мережі і її похибки від кількості циклів навчання. Результати роботи використані при викладанні дисциплін “Інтелектуальний аналіз даних”, “Методи та інформаційні технології обробки великих даних (Big Data)” і сприяють поліпшенню зацікавленості студентів дисциплінами і опанування їми матеріалу, який присвячений нейронним мережам.

### ПЕРЕЛІК ПОСИЛАНЬ

1. Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth. Neural Network Toolbox For Use with MATLAB. User's Guide Version 4 [Електрон. ресурс]. Режим доступу: [http://cda.psych.uiuc.edu/matlab\\_pdf/nnet.pdf](http://cda.psych.uiuc.edu/matlab_pdf/nnet.pdf) (дата звернення: 09.12.2021).

2. Mark Hudson Beale, Howard B. Demuth. Deep Learning Toolbox For Use with MATLAB. User's Guide R2020a [Електрон. ресурс]. Режим доступу: <https://dokumen.pub/matlab-deep-learning-toolbox-users-guide-r2020anbsped.html> (дата звернення: 09.12.2021).

3. Andreas C. Müller, Sarah Guido. Introduction to Machine Learning with Python [Текст] / O'Reilly Media, Inc., 2016, 393 с.

4. Eibe Frank, Mark A. Hall, and Ian H. Witten. The WEKA Workbench. Online Appendix for “Data Mining: Practical Machine Learning Tools and Techniques” Morgan Kaufmann, 4-th ed. [Електрон. ресурс]. Режим доступу: [https://www.cs.waikato.ac.nz/ml/weka/Witten\\_et\\_al\\_2016\\_appendix.pdf](https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf) (дата звернення: 09.12.2021).

5. Мороз Б.І. Лабораторний практикум з курсу: “Аналіз даних та процесів”. [Електрон. ресурс]. Режим доступу: <https://do.nmu.org.ua/course/view.php?id=3214> (дата звернення: 09.12.2021).

6. User's Guide Mathcad 14.0. Parametric Technology Corporation. [Електрон. ресурс]. Режим доступу: <http://www2.peq.coppe.ufrj.br/Pessoal/Professores/Arge/Nivelamento/Mathcad/2-Apostilas/Mathcad%20Users%20Guide.pdf> (дата звернення: 09.12.2021).

7. Brent Maxfield,. Essential PTC Mathcad Prime\_ 3.0 A Guide for New and Current Users [Текст] / Elsevier Inc., 2014, 563 с.

8. Stanislaw Osowski. Sieci Neuronowe do Przetwarzania Informacji [Текст] / Warszawa: Oficyna Wydawnicza Politechniki Warszawskiej, 2000, 344 с.