УДК 004.89

Грицик В.В., ФедорівТ.С.

ДОСЛІДЖЕННЯ ШЛЯХІВ ПОКРАЩЕННЯ НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЗАДАЧ РОЗПІЗНАВАННЯ ОБРАЗІВ

У роботі досліджено можливості оптимізації штучних нейронних мереж, що навчаються за правилом Хебба для задач розпізнавання.

***Ключові слова**: нейронна мережа,правило Хебба, розпізнавання образів.*

В работе исследовано возможности оптимизации искуственных нейронных сетей, которые обучаюся по правилу Хебба для задач распознавания.

***Ключевые слова:** нейронная сеть, правило Хебба, распознавание образов.*

There has been researched the neuronet (Hebb rule) optimization for pattern recognition tasks.

***Key words:** neuronet, Hebb rule, pattern recognition.*

1. ВСТУП

У комп'ютерних технологіях (штучний інтелект, автоматизовані системи управління, проектування автономних систем реагування тощо) задача розпізнавання образів є дуже актуальною. Це зумовлено як складністю опису об'єкту розпізнавання, його виділення з фону на дискретному полі уваги, так і потребою застосування (аерокосмічна галузь, системи безпеки, автоматизовані системи розпізнавання для прийняття рішень… У системах розпізнавання навіть виділено окрему галузь – розпізнавання біомедичних образів [3]. Про актуальність тільки одного цього напрямку говорить той факт, що над цими задачами працюють такі відомі компанії, як Цейс, Олімпус, ... [4]).

Для розпізнавання візуальних образів одним з найефективніших підходів при побудові алгоритмів розпізнавання та класифікації є застосування штучний нейронних мереж.

2. БАЗОВІ ПРИНЦИПИ ФУНКЦІОНУВАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Базовою нейромережею є одношарова нейромережа [1]. Будова базової одношарової нейромережі зображена на рис. 1

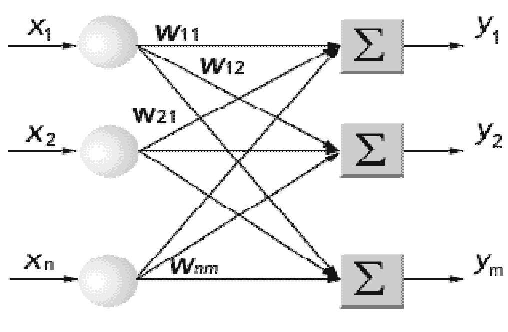


Рис. 1. Будова базової нейромережі

Найпростіша мережа складається з групи нейронів, що створюють прошарок, як показано в правій частині рис. 1 [2]. У лівій частині зображення знаходиться група вхідних вузлів.

Навчання базової нейромережі

Перцептрон навчають, подаючи множину образів по одному на його вхід і налаштовуючи ваги доти, поки для всіх образів не буде досягнутий необхідний вихід [2].

Одним з базових методів навчання нейромережі є навчання за правилом Д. Хебба [2], згідно з яким приріст значення ваг w_{ij} здійснюється пропорційно до його відношення вхідного і вихідного сигналу, як показано у формулі [1]:

$$\Delta w_{ij} = \eta \cdot x_i \cdot y_j, \tag{1}$$

де η – коефіцієнт навчання, який змінюється від 0 до 1 включно, і – номер вхідного синапса, j – номер вихідного синапса, x_i значення і-ого входу, y_j – значення j-ого виходу. Отже, щоб знайти значення вагового коефіцієнта в даний момент часу, потрібно до значення вагового коефіцієнта в попередній момент часу додати приріст значення ваг [1]:

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}, \tag{2}$$

де *t* – номер кроку навчання. Як видно із формули, значення ваг постійно збільшується, тому правило Хебба характеризується тим, що вагові коефіцієнти можуть набувати доволі великих значень. Одним зі способів стабілізації є врахування для уточнення ваги останнього значення вагового коефіцієнта на коефіцієнт забування $\lambda \in (0,1)$ [1]:

$$W_{ij}(t+1) = W_{ij}(t)(1-\lambda) + \Delta W_{ij} \tag{3}$$

Приклад навчання ваг за правилом Хебба: на рис. 2 зображений перцептронний нейрон з трьома входами; проведемо навчання даного нейрона за правилом Хебба. Отже, в даному перцептроні $i \in (1,3), j = 1$.

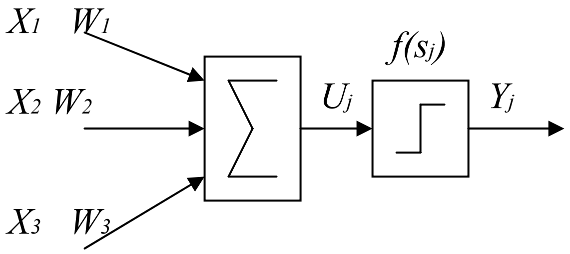


Рис. 2. Будова однонейронного перцептрона

Припустимо, що початкові вагові коефіцієнти $W^T{}_{(0)} = [0 \quad 0 \quad 0]$, для спрощення виберемо коефіцієнт навчання $\eta = 1$. На першому кроці навчання подамо на вхід навчальну вибірку $X^T{}_{(0)} = [1 \quad 0 \quad 1]$, а на вихід 1 $Y_i = 1$; внаслідок першого кроку навчання встановились такі вагові коефіцієнти $\Delta W^T{}_1 = W^T_{(1)} = [1 \quad 0 \quad 1]$. На другому кроці навчання подамо на вхід навчальну вибірку $X^T{}_{(1)} = [0 \quad 0 \quad 1]$, а на вихід – 1; матриця приросту вагових коефіцієнтів матиме вигляд $\Delta W^T{}_2 = [0 \quad 0 \quad 1]$; додавши приріст вагових коефіцієнтів до існуючих вагових значень, отримаємо – $W^T_{(2)} = [1 \quad 0 \quad 2]$. Як видно із результатів, вагові коефіцієнти можуть набувати доволі великих значень, найбільше можливе значення вагового коефіцієнта дорівнює початкове значення плюс найбільше можливе значення приросту, помножене на кількість кроків навчання.

$$W_{\max ij}(t) = W_{ij}(t_0) + \Delta W_{\max ij} \cdot t = W_{ij}(t_0) + (\eta \cdot x_{\max i} \cdot y_{\max j}) \cdot t. \tag{4}$$

З формули видно, що для нашого прикладу $W_{\max ij}(2) = 0 + (1{*}1{*}1){*}2 = 2$. Для того, щоб перевести вагові коефіцієнти синапсів до можливих значень (0,1)? потрібно кожний ваговий коефіцієнт поділити на $W_{\max ij}$, отже, для нашого випадку $W^T_{(2)} = [0{,}5 \quad 0 \quad 1]$. Від попередніх вагових значень можна легко перейти до можливих значень (-1, 1) – потрібно відняти від кожного вагового коефіцієнта 0,5, а потім домножити різницю на 2, отже, $W_{(2)}^{T} = [0 \quad -1 \quad 1]$.

Використання методу Хебба для навчання нейронних мереж привело до великих успіхів, але нарівні з цим показало обмеженість методу; деякі образи просто не можуть використовуватися для навчання цим методом [2, 11, 12].У результаті з'явилася велика кількість розширень і нововведень, більшість з яких значною мірою заснована на роботі Хебба [2]. Серед розширень є метод сигнального навчання Хебба та метод диференційного навчання Хебба.

3. ПОСТАНОВКА ТА РОЗВ'ЯЗОК ЗАДАЧІ

Нехай на бінарному полі уваги полі уваги потрібно розпізнати і класифікувати об'єкти.

Для виконання завдання використано базову модель: одношарову нейромережу. Адже на найпростішій мережі демонстрація алгоритмів буде найбільш наочною. Особливістю даного перцептрона є те, що нейрони не мають функції активації, а виходи нейромережі – це результати сумування нейрона; дана властивість введена заради того, щоб краще дослідити результати роботи перцептрона.

Перцептрон складається [1] з шару вхідних нейронів та шару вихідних нейронів. У шарі вхідних нейронів знаходиться n (200) нейронів, кожному з яких відповідає значення конкретного пікселя у вхідному зображенні. В прихованому шарі знаходиться m (2) нейронів, яка залежить від кількості класів, на які розділяє нейрона мережа (перцептрон класифікує до m + 1 класу) (рис. 1).

Методи навчання ваг

Як було сказано, методи навчання можна вибрати в самій програмі, а також оцінити ефективність цих методів багатьма різними параметрами.

На спосіб надання вагових значень впливає той факт, що ваги виводяться на екран в доступній для користувача формі. А, отже, набувають цілих значень. Це не відповідає загальноприйнятій практиці надавати значень від мінус одиниці до плюс одиниці. Принципового значення ця особливість не несе, якщо поділити значення ваг, отримане подібним методом, на найбільше число, яке може отримати даний синапс, то отримаємо значення ваг в проміжку від мінус одиниці до плюс одиниці [5-10, 15].

1. Базовий метод на основі навчання за правилом Хебба (з коефіцієнтом запам'ятовування $\eta = 1$, з областю значень вхідних даних $X_i \in (-1,1)$ полягає у пропорційному додаванні до вже існуючих ваг значень, які подаються на вхід (1). Користувач разом із вхідними даними подає значення класу, до якого належить даний рисунок.

Значення ваг при $t = 0$ зазвичай дорівнює нулю, хоча можна задати значення наперед визначені, наприклад базовий клас чи класи.

Отже, опис алгоритму наступний.

Усім вагам присвоюємо початкові значення (в даному випадку нулі).

Вибираємо наступний навчальний приклад і посилаємо на кожен відповідний вхід.

Множимо вхідне значення синапса на вихідне виходу нейрона (яке задано учителем) і додаємо до вже існуючої ваги синапса.

Якщо залишилися навчальні приклади, то переходимо до кроку 2, інакше завершуємо навчання. Ваги при використанні першого способу навчання виглядають наступним чином.

Приклад навчання перцептрона елементарними зображеннями. Для вхідної послідовності візьмемо чотири зображення класу одиниця, подані у виді числових значень пікселів, і проведемо навчання. Як видно з рис. 3, вхідні зображення мають різні шрифти, різні розміри. Результатом навчання будуть конкретні значення вагових коефіцієнтів. Щоб показати результати навчання, кожний ваговий коефіцієнт поставимо на місце тих вхідних значень, які впливають на даний ваговий коефіцієнт (рис. 4). В результаті можна просто прослідкувати, до яких змін вагових коефіцієнтів будуть приводити різні навчальні приклади.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	255	255	0	0	0
0	0	255	255	255	255	255	0	0	0
255	255	255	0	255	255	255	0	0	0
255	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	255	255	255	255	255	255	255	255	255
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

а) еталон № 1

0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	255	255	255	0	0	0	0
0	0	255	255	255	255	0	0	0	0
0	0	255	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

б) еталон № 2

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	255	255	255	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	255	255	255	255	255	255	255	255	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

в) еталон № 3

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	255	255	255	255	255	0	0	0
0	255	255	255	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	255	255	255	255	255	255	255	255	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

г) еталон № 4

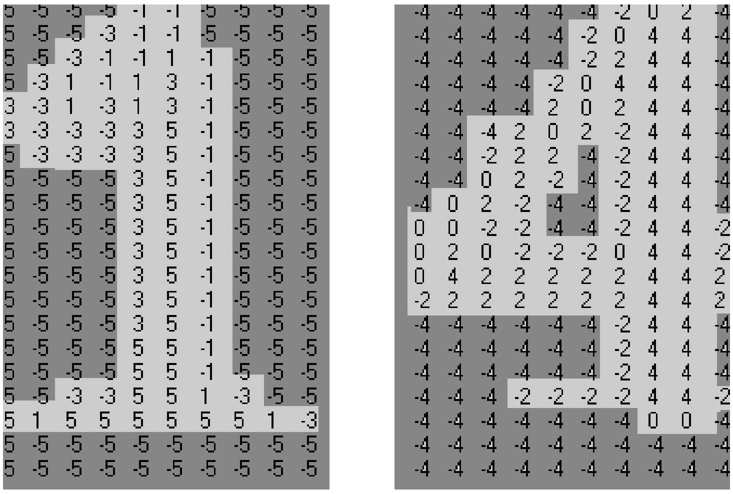
Рис. 3. Вхідні зображення класу одиниця, подані у виді числових значень пікселів

-4	-4	-4	-4	-2	-2	-4	-4	-4	-4
-4	-4	-4	-4	-2	-2	-4	-4	-4	-4
-4	-4	-4	-2	-2	0	0	-4	-4	-4
-4	-4	0	0	2	2	0	-4	-4	-4
-2	-2	2	-2	2	2	0	-4	-4	-4
-2	-2	-2	-2	4	4	0	-4	-4	-4
-4	-2	-2	-2	4	4	0	-4	-4	-4
-4	-4	-4	-4	4	4	0	-4	-4	-4
-4	-4	-4	-4	4	4	0	-4	-4	-4
-4	-4	-4	-4	4	4	0	-4	-4	-4
-4	-4	-4	-4	4	4	0	-4	-4	-4
-4	-4	-4	-4	4	4	0	-4	-4	-4
-4	-4	-4	-4	4	4	0	-4	-4	-4
-4	-4	-4	-4	4	4	0	-4	-4	-4
-4	-4	-4	-4	4	4	0	-4	-4	-4
-4	-4	-4	-4	4	4	0	-4	-4	-4
-4	-4	-2	-2	4	4	2	-2	-4	-4
-4	2	4	4	4	4	4	4	2	-2
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4

Рис. 4. Значення вагових коефіцієнтів після навчання методом Хебба

Всі ті частини зображення, які ніколи не використовувалися, обведені темнішим кольором. На світлому фоні видно ділянки, які хоча б раз використовувались при малюванні навчальних зображень. Як видно, значення вагових коефіцієнтів не співпадає з жодним навчальним прикладом, проте схоже на будь-яке з чотирьох навчальних. Очевидно, що в центрі світлої зони значення ваг найбільші, тому що вони використовувались у більшій кількості випадків (у даному випадку у всіх навчальних прикладах). Відповідно на краях зони використання синапсів, де синапси навчалися меншу кількість раз, вагові коефіцієнти менші. Кількість епох у даному випадку відповідає найбільшому значенню вагового коефіцієнта, хоча це не обов’язково [13, 14].

Покажемо вагові коефіцієнти при навчанні нейромережі іншими навчальними зображеннями одиниці і четвірки. Для класу одиниці число навчальних прикладів – п’ять, для класу четвірки – чотири. Як було вище описано, вагові коефіцієнти при методі Хебба можуть набувати доволі малих значень.



а) б)

Рис. 5. Вагові коефіцієнти класу одиниця (а) і четвірка (б) після навчання методом Хебба

Наступні методи беруть за основу метод навчання за правилом Хебба, або іншим методом, і переробляють уже існуючі вагові значення. Тому загальна формула, за якою будуть встановлювати синаптичні значення дані алгоритми, буде така:

$$W_{ij}(meth) = W_{ij}(heb) + \Delta W_{ij}\,, \qquad (5)$$

де $W_{ij}(heb)$ – значення вагового коефіцієнта між і-им вхідним та j-им вихідним нейроном після навчання методом за правилом Хебба; ΔW_{ij} – зміщення вагових коефіцієнтів після одного з наступних методів; $W_{ij}(meth)$ – значення вагового коефіцієнта після навчання одним з наступних методів.

2. Складнішим є метод, який, крім навчання синапса вхідним сигналом порівнює значення ваг, які відповідають даному вхідному сигналу в інших класах. Формула вагового зміщення для даного метода наступна:

$$\Delta W_{ij} = \eta \sum_{k=1}^{N} \left(\frac{W_{j\max} + W_{ij}}{W_{j\max}}\right) - \left(\frac{W_{k\max} + W_{ik}}{W_{k\max}}\right), \qquad (6)$$

де N – кількість нейронів вихідного шару, η – коефіцієнт навчання. $W_{j\max}$ найбільше можливе значення ваг для j-ого вихідного нейрона.

Отже, опис алгоритму № 2 наступний.

1. Всім вагам присвоюємо початкові значення.
2. Початково навчаємо мережу першим методом.
3. Обираємо перший вхідний нейрон *i.*
4. Обираємо перший вихідний нейрон j вагове зміщення для даного вихідного нейрона прирівнюємо до нуля.
5. Обираємо перший вихідний нейрон *k*.

6. Від найбільшого можливого значення ваг *ij* віднімаємо поточне значення ваг синапса *ij*, різницю ділимо на найбільш можливе значення ваг *ij*, аналогічні дії виконуємо із вагами k_j, різницю часток додаємо до вагового зміщення (інший варіант алгоритму – додаємо різницю часток, тільки якщо вона більша нуля).
7. Якщо ще є хоча б один наступний після *k*-ого вихідний нейрон, то *k* присвоюємо значення $k+1$ і переходимо до кроку 6, якщо ні – то вагове зміщення помножаємо на η.
8. Якщо ще є хоча б один наступний після *j*-ого вихідний нейрон, то *j* присвоюємо значення $j+1$ і переходимо до кроку 5.
9. Якщо ще є хоча б один наступний після *i*-ого вхідний нейрон, то *i* присвоюємо значення $i+1$ і переходимо до кроку 4.

Метод складається з двох етапів. На першому етапі встановлюються початкові значення ваг синапсів (наприклад, вищеописаним пропорційним методом). На основі даних першого етапу формуються кінцеві значення ваг.

Як правило, класи кардинально не відрізняються один від одного, а отже, є ділянки, ідентичні в декількох класах. При функціонуванні системи виникала проблема – різні класи подавали високі значення виходу, хоча вхідна послідовність належала до якогось одного класу. Така ситуація виникла внаслідок схожості частин зображення. Суть цього методу полягає в тому, щоб надати тим частинам зображення, які не повторюються в інших класах більших вагових значень, і, як наслідок, ці частини будуть більше впливати на кінцевий результат, а отже, кінцевий результат буде точнішим [11].

Для наочності опишемо даний алгоритм у випадку, коли мережа працює з двома класами і трьома вхідними вузлами та $\eta = 4$.

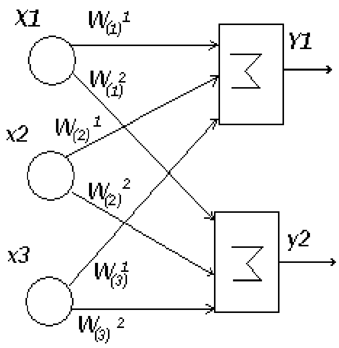
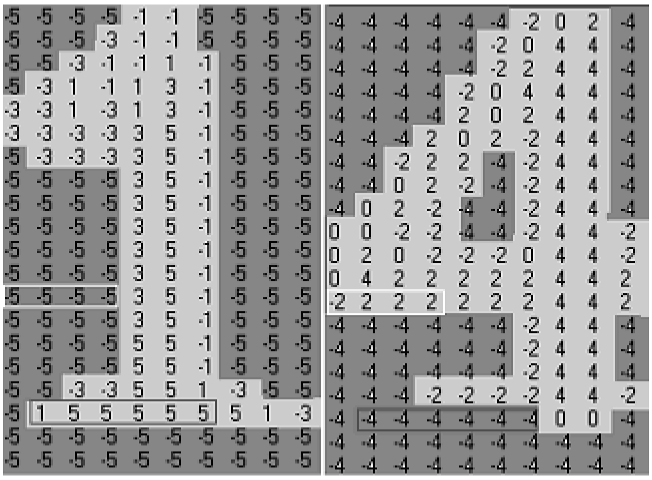


Рис. 6. Функціональна схема алгоритму № 2

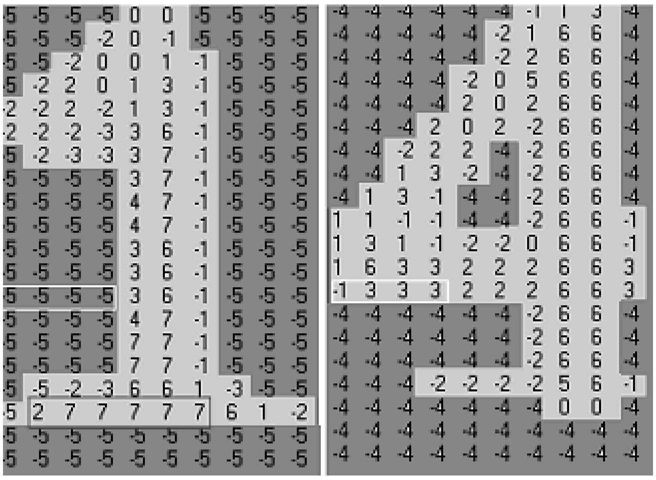
Нехай матриця вагових коефіцієнтів після першого методу навчання матиме вигляд $W^{m1} = \begin{bmatrix} 2 & 7 & 0 \\ 1 & 2 & 1 \end{bmatrix}$, кількість кроків попереднього навчання для одного класу – 10 і 5 відповідно, отже, і одного нейрона (ваговим коефіцієнтам *n*-ого нейрона відповідають вагові коефіцієнти n-ого рядка матриці) однакова, тоді матриця $\Delta W = \begin{bmatrix} 0 & 1.2 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}$, якщо додаємо різницю > 0, або $\Delta W = \begin{bmatrix} 0 & 1.2 & -0.8 \\ 0 & -1.2 & 0.8 \end{bmatrix}$, якщо ми додаємо різницю завжди. Значення вагових коефіцієнтів після навчання методом $W^{m2} = \begin{bmatrix} 2 & 8.2 & 0 \\ 1 & 2 & 1.8 \end{bmatrix}$ або $W^{m2} = \begin{bmatrix} 2 & 8.2 & -0.8 \\ 1 & 0.8 & 0.8 \end{bmatrix}$ відповідно.

Для прикладу результати навчання перцептрона елементарними зображеннями з $\eta = 1$ показано на рис. 7 відповідно класу одиниця і четвірка за правилом Хебба і поперечним способами, а також подібні зображення другого класу.

Опишемо роботу даного методу для прикладу навчання перцептрона елементарними зображеннями, проте для наочності використаємо алгоритм з додаванням різниці часток, більших за нуль (2).



а) б)



в) г)

Рис. 7. Вагові коефіцієнти після навчання методом Хебба (а – класу одиниця, б – класу четвірка), а також після 2-го методу на його основі (в – класу одиниця; г – класу четвірка)

На рис. 7 (а, в) видно відмінність значення ваг при тренуванні класу одиниця першим (а) та другим методам (в) та аналогічно – класу четвірку (відповідно б і г). Різницю ваг ми бачимо на тих частинах зображення (які обведені білим кольором), які малюються в одиниці і не задаються в четвірці.

Порівнявши нижні та верхні зображення, ми бачимо – розміщення світлих зон використання не змінилося, змінилися вагові коефіцієнти в них. Також змінилося максимальне значення синапса: в першому методі це було 5 для класу одиниці та 4 для четвірки, в другому алгоритмі – це, відповідно, 7 і 6. Зміна найбільшої ваги спричинена додаванням до індивідуальних частин зображення константи двійки. Збільшилося максимальне значення синапса, проте кількість епох не змінилася.

3. Метод № 3 збільшує ваги тим синапсам, які частіше використовуються. Формула вагового зміщення для даного метода наступна:

$$\Delta W_{ij} = \begin{cases} C, W_{ij} > \min W \\ 0, W_{ij} \le \min W \end{cases}, \qquad (7)$$

де $\min W$ – переломне значення синаптичних ваг, C – стала, яку додаємо до ваг, що прив'язана до кількості кроків навчання. Якщо значення синаптичних ваг перевищує $\min W$, то дана вага нами оцінюється як «корисна», і ми її збільшуємо. Якщо ні, то ми не змінюємо ваги. В досліді, розглянутому нижче, С – стала додавання = 5, а $\min W$ вибрано найменше з можливих, а саме – 0 для прикладу навчання ваг за правилом Хебба, число, яке дорівнює кількості кроків навчання із знаком мінус, для прикладу навчання перцептрона елементарними зображеннями.

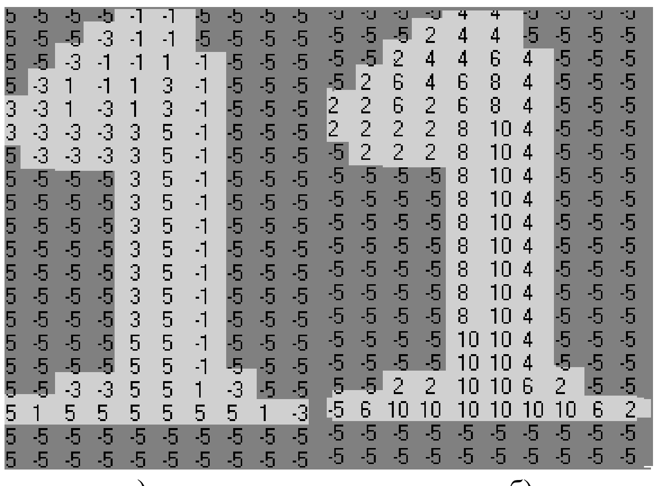
Отже, опис алгоритму наступний.

1. Всім вагам присвоюємо початкові значення (в даному випадку нулі).
2. Початково навчаємо мережу першим методом.
3. Обираємо перший вхід.
4. Якщо значення ваг менше рівне $\min W$, то переходимо до кроку 5, інакше додаємо до значення ваг константу.
5. Якщо ще є входи, то вибираємо наступний, якщо ні, то завершуємо навчання.

Якщо б використати даний метод зі сталою додавання 5, і $\min W = 0$ до прикладу навчання ваг за правилом Хебба, описаним вище, то отримаємо значення синапсів $W_{(3)}^T = [6 \quad 0 \quad 7]$. Найбільше можливе значення для даного методу дорівнює найбільшому значенню (4) додати сталу додавання. Отже, якщо перевести $W_{(3)}^T$ до значень синапсів (0,1), то отримаємо $W_{(3)}^T = [0.85 \quad 0 \quad 1]$; порівнявши $W_{(2)}^T$ переведеними до значень синапсів (0,1), робимо висновок, що метод працює, як своєрідний фільтр.

Дивлячись на поле відображення ваг після першого методу у прикладі навчання перцептрона елементарними зображеннями, можна помітити, що ті частини, які не задіяні, і ті, які задіяні, проте рідко, за числовими значеннями вагових коефіцієнтів схожі. А нам було б доцільніше збільшити різницю між задіяними і незадіяними вагами. Отже, метод перевіряє, чи ваги хоча б один раз задіювались ($\min W =$ кількості раз, коли був задіяний даний клас зі знаком мінус), і якщо так, тоді додаємо до них певне значення (константу, що прив'язана до кількості ітерацій навчання). У даному випадку, щоб перевірити, чи ваги задіяні, потрібно значення ваг розділити на кількість кроків.

На рисунку 8 видно значення ваг класу одиниця, навчених першим та третім методом.



а) б)

Рис. 8. Порівняння вагових коефіцієнтів після навчання а) – методом Хебба та б) – 3-м методом

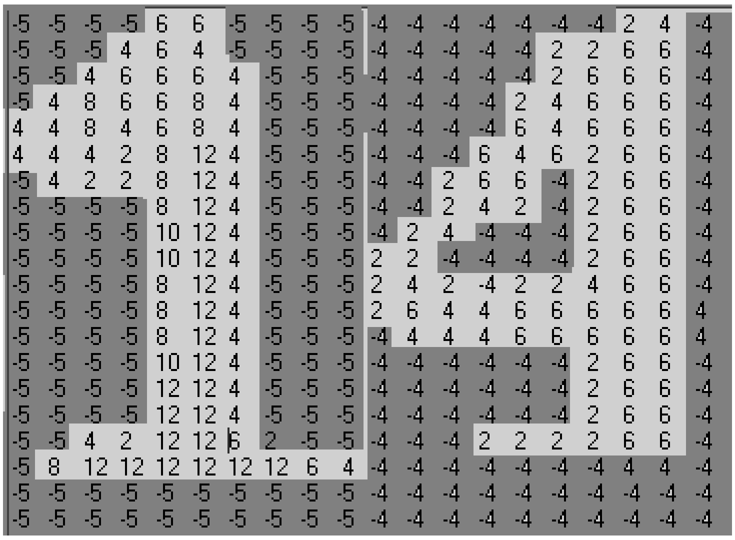
Як видно з рис. 8, форма світлої зони не змінилася. Пропорції між вагами світлої зони також не змінилися, адже зображення ваг 3-м методом відрізняється від зображення першого методу тим, що у значення ваг світлої зони 3-го методу на константу збільшені. На відміну від другого методу, збільшені значення всіх ваг світлої зони, незалежно від інших класів.

4. Найбільших значень виходи набувають при об'єднаному методі навчанні. Спочатку встановлюються значення ваг пропорційним способом, потім збільшуються значення тих ваг, які хоч раз задіяні, і врешті порівнюємо два класи за схожими частинами зображення.

Отже, опис алгоритму наступний.

1. Всім вагам присвоюємо початкові значення (в даному випадку нулі).
2. Обираємо перший вхід.
3. Виконуємо навчання 3-м способом
4. Якщо ще є входи, то вибираємо наступний і переходимо до кроку 3.
5. Виконуємо навчання 2-м способом.
6. Завершуємо навчання.

Початково навчаємо мережу пропорційним методом.



а) б)

Рис. 9. Навчання 4-м методом класу: а – одиниці, б – четвірки

Як видно із рис. 9, найбільше відрізняються частини зображення, які є індивідуальними, а також збільшені зображення світлої зони використання.

5. Можна використати другий метод навчання, якщо потрібно не навчити два класи, знаходячи відмінності між ними, а навчити один клас, якщо є помилкові зображення, схожі на нього. Нам потрібно ввести помилкове зображення як окремий клас і здійснити навчання 2-м методом. Помилкове зображення як клас нас не цікавить, тому ми його не використовуватимемо при функціонуванні, або можемо перевчити на корисне.

У прикладі навчання перцептрона елементарними зображеннями постала проблема, що в першому методі на одне з невірних зображень (10-а) видавалося високе вихідне значення (10-б, вихід 1-го нейрона). Був проведений дослід, у якому нейромережу навчили додатковому класу. Хибне зображення було введено як додатковий клас, і задіяно 5-м методом навчання. Результатом досліду було те, що нейрон, який 1-м методом на невірне зображення виводив високі результати, даним способом на невірні вхідні дані показав нижчі результати. Що ми і намагалися зробити, проводячи дослід. На рис. 10 б) показано, що 2-й нейрон, навчений 5-м методом, має меншу різницю між 1-м методом, навченим методом Хебба, якщо подано правильне зображення (клас 1), і більшу різницю, якщо невірне зображення (клас 2).

Використовуючи даний метод, можна використати наступний прийом. Зображення, яке не належить до даного класу, але внаслідок випадкової схожості сприймається перцептроном як даний клас, можна створити новий клас фіктивний (адже він при функціонуванні не задіюється і може бути знищений), який використовується тільки при навчанні, далі ми проводимо другий етап даного методу. Як наслідок, посилюється різниця між виходами нейрона, коли подається правильне і помилкове зображення.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	255
0	0	0	0	255	255	255	255	255	255
0	0	0	0	255	255	255	255	255	255
0	0	0	0	255	255	0	0	0	255
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

а) помилкове зображення,
схоже на клас одиниці

Клас	Значення виходу 1ий нейрон	---2ий нейрон	різниця
1	131	130	1
1	103	103	0
1	108	105	3
1	134	133	1
1	98	97	1
2	90	83	-7

б) порівняння вихідних даних нейрона 1,
який навчений методом Хеббаі методом 5

-5	-5	-5	-5	-1	-1	-5	-5	-5	-5
-5	-5	-5	-3	-1	-1	-5	-5	-5	-5
-5	-5	-3	-1	-1	1	-1	-5	-5	-5
-5	-3	1	-1	1	3	-1	-5	-5	-5
-3	-3	1	-3	1	3	-1	-5	-5	-5
-3	-3	-3	-3	3	5	-1	-5	-5	-5
-5	-3	-3	-3	3	5	-1	-5	-5	-5
-5	-5	-5	-5	3	5	-1	-5	-5	-5
-5	-5	-5	-5	3	5	-1	-5	-5	-5
-5	-5	-5	-5	3	5	-1	-5	-5	-5
-5	-5	-5	-5	3	5	-1	-5	-5	-5
-5	-5	-5	-5	3	5	-1	-5	-5	-5
-5	-5	-5	-5	3	5	-1	-5	-5	-5
-5	-5	-5	-5	3	5	-1	-5	-5	-5
-5	-5	-5	-5	5	5	-1	-5	-5	-5
-5	-5	-5	-5	5	5	-1	-5	-5	-5
-5	-5	-3	-3	5	5	1	-3	-5	-5
-5	1	5	5	5	5	5	5	1	-3
-5	-5	-5	-5	-5	-5	-5	-5	-5	-5

в) методом Хебба

-5	-5	-5	-5	0	0	-5	-5	-5	-5
-5	-5	-5	-2	-1	-1	-5	-5	-5	-5
-5	-5	-2	0	-1	1	0	-5	-5	-5
-5	-2	2	0	1	3	0	-5	-5	-5
-2	-2	2	-2	1	3	0	-5	-5	-5
-2	-2	-2	-2	3	5	0	-5	-5	-5
-5	-2	-2	-2	3	5	0	-5	-5	-5
-5	-5	-5	-5	3	5	0	-5	-5	-5
-5	-5	-5	-5	3	5	0	-5	-5	-5
-5	-5	-5	-5	3	5	-1	-5	-5	-5
-5	-5	-5	-5	3	5	-1	-5	-5	-5
-5	-5	-5	-5	3	5	0	-5	-5	-5
-5	-5	-5	-5	3	5	0	-5	-5	-5
-5	-5	-5	-5	3	5	0	-5	-5	-5
-5	-5	-5	-5	5	5	0	-5	-5	-5
-5	-5	-5	-5	5	5	0	-5	-5	-5
-5	-5	-2	-2	5	5	2	-2	-5	-5
-5	2	7	7	7	7	7	7	2	-2
-5	-5	-5	-5	-5	-5	-5	-5	-5	-5

г) 5-ім методом

**Рис. 10. Порівняння вагових коефіцієнтів після навчання
Методом Хебба та методом № 5**

Зображення звичайним способом навчання зліва, з даним способом навчання справа. В даному випадку збільшилась точність, вихідне значення невірного зображення зменшилося.

ВИСНОВОК

У більшості випадків мережа працює коректно, як і очікувалось, кожний наступний метод показує кращі результати за попередній. Це зумовлено тим, що кожний наступний метод враховує недоліки попереднього і використовує його як базовий. До переваг базового методу належить його швидкодія. Проте ця перевага у швидкодії залежить від багатьох факторів, таких як кількість входів, кількість нейронів прихованого шару, кількість кроків навчання, структури навчальних послідовностей. Наступні методи встановлюють ваги на основі правила навчання ваг способом Хебба, тобто включають цей метод у собі. Отже, чим довше триває перший метод навчання, тим меншою є його перевага щодо складніших методів, і чим довше тривають другі етапи навчання, тим перевага пропорційного метода збільшується. Збільшення часу навчання першим методом відбувається внаслідок збільшення кількості навчальних прикладів і, як наслідок, кроків навчання.

$$K_{(hebb)} = N \cdot t \,, \tag{8}$$

де $K_{(hebb)}$ – кількість операцій надання ваг методом Хебба, t – кількість кроків навчання. N – кількість вхідних нейронів.

Наслідком того, що методи 2 і 3 за основу беруть метод Хебба, кількість їхніх операцій надання ваг дорівнює

$$K_{2,3} = K_{(hebb)} + N \cdot K = N \cdot t + N \cdot K \, , \qquad (9)$$

де $K_{2,3}$ – кількість операцій надання ваг 2-м і 3-м методами, K – кількість нейронів вихідного шару.

Одна операція надання ваг синапсам у 3-му методі дорівнює K-операцій віднімання поточного синапса (між вхідним нейроном та вихідним) від наступного синапса (між вхідним нейроном та наступним вихідним), а точніше K-1 операції адже недоцільно віднімати значення синапса від себе, а також K (K-1) операцій додавання різниці до значення поточного синапса.

Одна операція надання ваг синапсам у 3-му методі дорівнює операції порівняння (W_{ij} і $\min W$), а також операції додавання до ваг сталої (кількість операцій додавання залежно від значень синаптичних ваг після навчання методом Хебба може змінюватись від 0 до N).

Найкращі результати (рис. 12), як і очікувалось, у методі № 4. Також четвертий метод є універсальним, адже він використовує всі три методи. Отже, формула 4-го метода:

$$K_{2,3} = K_{(hebb)} + 2 \cdot N \cdot K \, , \qquad (10)$$

де одна операція надання ваг дорівнює подібним операціям 2-го і 3-го методу.

	Значення виходу	
Клас	1ий нейрон	---2ий нейрон
1	107	81
0	-105	-79
1	131	52
1	103	69
1	108	66
1	134	46
1	98	63
2	72	85
2	81	82
2	82	109
2	61	119
2	45	115
2	46	93
2	39	95
2	81	83

а) метод Хебба

	Значення виходу	
Клас	1ий нейрон	---2ий нейрон
2	93	97
0	-91	-95
1	138	62
1	99	81
1	102	78
1	141	58
1	94	74
2	63	90
2	75	89
2	69	115
2	50	121
2	35	112
2	35	92
2	30	91
2	71	89

б) метод № 2

	Значення виходу	
Клас	1ий нейрон	---2ий нейрон
0	29	-30
0	-27	32
1	147	12
1	99	25
1	78	0
1	148	0
1	84	13
2	54	65
1	69	56
2	36	63
2	43	115
2	29	131
2	24	97
2	21	117
2	43	45

в) метод № 3

	Значення виходу	
Клас	1ий нейрон	---2ий нейрон
0	15	2
0	-13	0
1	154	35
1	95	50
1	72	23
1	155	25
1	80	37
2	45	75
2	63	74
2	23	86
2	32	134
2	19	137
2	13	99
2	12	112
2	33	58

г) метод № 4

Рис. 12. Порівняльна характеристика класифікації за розглянутими методами

Експериментально доведено, що результат класифікації за методом № 4 можна покращити шляхом навчання алгоритму помилок, що зустрічаються найчастіше. Такий підхід відображений у методі № 5.

ЛІТЕРАТУРА

1. Станіслав Осовський. Нейронні мережі для обробки інформації.
2. Ф.Уосермен. Нейрокомп'ютерна техніка: Теорія і практика.
3. http://mi.eng.cam.ac.uk/milab.html
4. В.В. Грицик, М.А. Влах. Технічні та програмні засоби розпізнавання та аналізу зображень складних біологічних об'єктів // Інформаційні технології і системи. – Львів. – 2005.– Том. 8. – № 1. – P. 17-28.
5. Горбань А.Н., Россией Д.А. Нейронные сети на персональном компьютере. – Новосибирск: Наука. Сибирская издательская фирма РАН, 1996. – 276 с.
6. Haykin S. Neural Networks. A Comprehensive Foundation. – Upper Saddle River, N.J.: Prentice Hall, Inc., 1999. – 842 p.
7. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применение // Харьков: ТЕЛЕТЕХ, 2004. – 372 с.
8. Медведев В.С., Потемкин В.Г. Нейронные сети. MATLAB 6 / Под общ. ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002. – 496 с.
9. Назаров А.В., Лоскутов А.И. Нейросетевые алгоритмы прогнозирования и оптимизации систем. – СПб.: Наука и Техника, 2003. – 384 с.: ил.
10. Нейронные сети. STATISTICA Neural Networks // Пер. с англ. – М.: Горячая линия – Телеком, 2001. – 182 с.
11. Руденко О.Г., Бодянский Е.В. Основы теории искусственных нейронных сетей. – Харьков: ТЕЛЕТЕХ, 2002. – 317 с.
12. Rojas R. Neural Networks. A Systematic Introduction. – Berlin: Springer-Verlag, 1996. – 502 p.
13. Pham D. T., Liu X. Neural Networks for Identification, Prediction and Control. – London: Springer–Verlag, 1995. – 238 p.
14. Naftaly U., Intrator N., Horn D. Optimal ensemble averaging of neural networks // Network: Comput. Neural Syst. – 1997. – 8. – P. 283-296.
15. Giacinto G., Roli F. Design of effective neural network ensembles for image classification purposes // Image and Vision Computing, Elsevier Science. – 2001. – Vol. 19. – P. 699-707.

Рецензенти: д.т.н., проф. Коваленко І.І.,
д.т.н., проф. Кондратенко Ю.П.

Стаття надійшла до редакції 11.09.09