

Radosław MATUSIK

UNIwersytet Łódzki, KATEDRA ANALIZY MATEMATYCZNEJ I TEORII STEROWANIA

Metody uczenia sieci neuronowej Hopfielda

mgr Radosław MATUSIK

Absolwent Uniwersytetu Łódzkiego. Od 2006 roku bierze udział w projekcie programistycznym na macierzystej uczelni. W pracy naukowej zajmuje się badaniem stabilności sieci neuronowych oraz konstrukcją funkcji typu Lapunowa.

e-mail: radmat@math.uni.lodz.pl

**Streszczenie**

W artykule przedstawione zostały od strony teoretycznej i porównane od strony praktycznej różne metody uczenia sieci neuronowej Hopfielda. Oprócz znanej i powszechnie stosowanej reguły Hebba, przedstawione zostały modyfikacje tej metody. W celu porównania reguł uczenia sieci Hopfielda napisana została specjalna aplikacja, w której zaimplementowane zostały przedstawione w artykule metody. Regułą najlepiej rozpoznającą zapamiętane wzorce okazała się metoda pseudoinwersji.

Słowa kluczowe: sieci neuronowe, Hopfield, reguły uczenia

Learning methods of Hopfield neural network**Abstract**

Hopfield neural network may have many applications, such as approximation, compression, association, steering or patterns recognition. When neural network is used to association, its role is association memory. This task consists on original patterns recognition even when on Hopfield neural network entrance distorted patterns were provided. In this paper various learning methods of Hopfield neural network were presented from theoretical point of view as well as compared from practical point of view. Besides well known and generally used Hebb rule there were presented its modifications. To compare the learning methods of Hopfield neural network the application implementing the described in the paper methods was written. In the second chapter Hopfield neural network model, Hopfield neural network definition and general schema of neural network were presented. Also activation function was described, which was used to test Hopfield neural network. In the third chapter various Hopfield network learning rules were presented, such as an original Hebb method, its modifications, the Oja rule and pseudoinversion rule. In the fourth chapter the testing process and its results are presented. The main task of this neural network is patterns recognition. Hopfield neural network stored 10 patterns. Every of stored patterns has 35 neurons. Then on the neural network entrance distorted patterns were provided. The tests proved that the pseudoinversion rule recognizes the patterns in the best way.

Keywords: neural networks, Hopfield, learning methods

1. Wstęp

W ciągu ostatnich dekad obserwuje się gwałtowny rozwój badań nad sztucznymi sieciami neuronowymi, których zadaniem jest jak najwierniejsze odwzorowanie zachowań zachodzących w mózgu człowieka. Sieci neuronowe mogą spełniać określone funkcje, które można podzielić na kilka grup: rozpoznawania i klasyfikacji wzorców, aproksymacji, kompresji, asocjacji, predykcji, czy też identyfikacji i sterowania. W niniejszym artykule sieć neuronowa Hopfielda wykorzystana została do zadań asocjacji. W tym przypadku sieć neuronowa pełni rolę pamięci skojarzeniowej. Zadanie takiej sieci polega na odtworzeniu oryginalnego wzorca mimo, że na wejściu podany został wzorec zaszumiony (zniekształcony). Takie zachowanie sieci związane jest z jedną z ich najważniejszych cech, czyli zdolnością do uczenia się i generalizowania nabytej wiedzy. Celowym jest przyjrzenie się różnym metodom uczenia sieci neuronowej, gdyż strategie te mają wpływ na jakość rozpoznawanych wzorców.

W niniejszym artykule zaprezentowane zostały od strony teoretycznej i porównane od strony praktycznej różne metody uczenia sieci neuronowej Hopfielda. Uczenie sieci może przebiegać z nauczycielem (gdzie oprócz sygnałów wejściowych znane są również pożądane sygnały wyjściowe) lub bez nauczyciela (gdzie na etapie adaptacji neuronu nie jesteśmy w stanie przewidzieć sygnału wyjściowego). W artykule przedstawione zostały wybrane strategie trenowania sieci neuronowej Hopfielda bez nauczyciela.

Wszystkie zaprezentowane w artykule metody uczenia sieci neuronowej są bezgradientowe. Związane jest to z zastosowaniem nieciągłej funkcji aktywacji, gdyż w tym przypadku nie możemy wykorzystać informacji o wartości pochodnej. Znamy jedynie aktualną wartość sygnału wyjściowego neuronu.

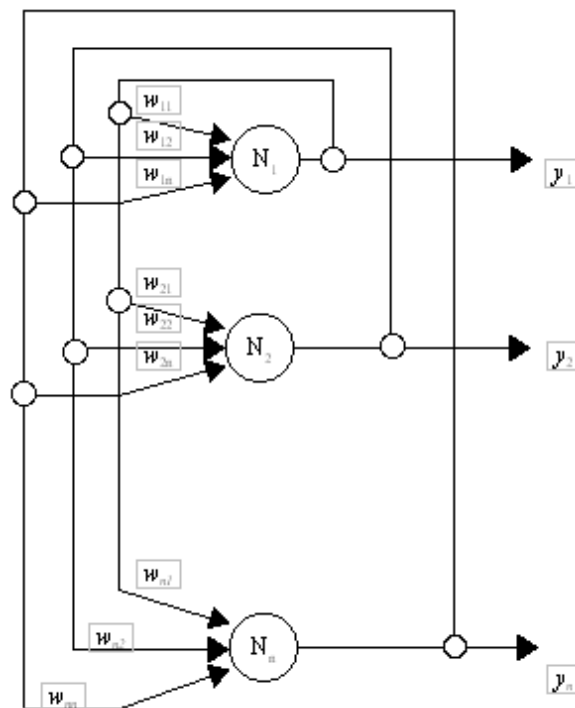
2. Sieć neuronowa Hopfielda

Sieć Hopfielda jest przykładem sieci rekurencyjnej. Taka architektura sieci charakteryzuje się sprzężeniem zwrotnym między warstwami sieci. Zależności dynamiczne występują na każdym etapie działania, a zmiana stanu jednego neuronu przenosi się na całą sieć.

Sieć neuronowa Hopfielda składa się z n neuronów, przy czym w danej chwili czasu t aktywny jest tylko jeden. Każdy neuron wybierany jest z jednakowym prawdopodobieństwem, a zmiany stanu neuronu następują w dyskretnych chwilach czasu. Zatem stan i -tego neuronu może być opisany następująco:

$$y_i(t+1) = f\left(\sum_{j=0}^n (w_{ij} y_j(t) + b_i)\right), \quad (1)$$

przy warunku początkowym $y_j(0)=x_j$, przy czym: x_j oznacza wektor wejściowy, $y_i(t+1)$ oznacza stan i -tego neuronu w chwili $t+1$, w_{ij} oznacza macierz wag, b_i oznacza bias, f oznacza funkcję aktywacji.



Rys. 1. Schemat ogólny sieci neuronowej Hopfielda
Fig. 1. General schema of Hopfield neural network

Jako funkcję aktywacji przyjmujemy funkcję sygnum zdefiniowaną następująco:

$$f(y_j(t)) = \begin{cases} 1, & \text{gdy } \sum_{i=0}^n w_{ij} y_j(t) + b_i > 0 \\ 0, & \text{gdy } \sum_{i=0}^n w_{ij} y_j(t) + b_i \leq 0 \end{cases}$$

W dalszej części rozważań będziemy zakładać, że bias wynosi zero (nie jest uwzględniony na rysunku 1).

W działaniu sieci Hopfielda wyróżnia się dwa tryby: uczenia i odtwarzania. W pierwszym z nich na podstawie dostarczonych do sieci wzorców uczących x_j dobierane są wagi w_{ij} . Proces uczenia sieci kształtuje obszary przyciągania (atrakcji) poszczególnych punktów równowagi, które odpowiadają danym uczącym. Po ustaleniu wartości wag następuje przejście do drugiego trybu, w którym następuje proces przebiegający według zależności (1), kończący się w określonym minimum lokalnym, w którym $y(t+1)=y(t)$.

W przypadku schematu sieci Hopfielda zakłada się symetryczność macierzy wag (co w zapisie macierzowym oznacza, że $W=W^T$) oraz brak sprzężenia zwrotnego neuronu z samym sobą (co oznacza, że $w_{ii}=0$).

Na rodzaj i ilość atraktorów wpływa dobór wag, czyli proces uczenia sieci Hopfielda. W dalszej części artykułu przyjrzymy się różnym metodom uczenia sieci neuronowej Hopfielda.

3. Metody uczenia sieci neuronowej Hopfielda

Przyjrzymy się teraz różnym algorytmom uczenia sieci neuronowej Hopfielda.

3.1 Metoda Hebba

D. Hebb, który badał zachowanie komórek nerwowych zauważył, że połączenie między dwoma komórkami jest wzmacniane, jeśli obie komórki są w stanie pobudzenia w tym samym czasie. Jeśli j -ta komórka o sygnale wyjściowym y_j jest powiązana z i -tą o sygnale wyjściowym y_i przez wagę w_{ij} to wówczas na stan powiązań tych komórek wpływają wartości sygnałów wyjściowych y_i oraz y_j .

Dla jednego wzorca wagi dobierane są według następującego wzoru: $w_{ij} = \frac{1}{n} \xi_i \xi_j$, gdzie n jest liczbą neuronów, natomiast ξ_i

oraz ξ_j stanowią odpowiednio i -tą i j -tą współrzędną wektora uczącego ξ . Natomiast przy prezentacji wielu wzorców uczących $\xi^{(p)}$ dla $k=1,2,\dots,p$ wagi dobierane są według wzoru:

$$w_{ij} = \frac{1}{n} \sum_{\mu=1}^p \xi_i^{\mu} \xi_j^{\mu}, \quad (2)$$

przy czym p stanowi liczbę wszystkich zapamiętanych wzorców.

Taki tryb uczenia sieci neuronowej sprawia, że wagi przyjmują wartości wynikające z uśrednienia wielu próbek uczących. Warto zwrócić uwagę na fakt, że wzór (2) daje dodatnią zmianę wag nawet w przypadku, gdy żaden neuron nie jest w stanie pobudzenia, tj. gdy $\xi_i^{\mu} \xi_j^{\mu} = -1$. Prawdopodobnie to zjawisko jest nieuzasadnione z fizjologicznego punktu widzenia. Co więcej, równanie (2) może nawet spowodować zmianę rodzaju danego połączenia z pobudzającego na hamujące i odwrotnie w przypadku dodawania nowych wzorców, co także wydaje się nie mieć uzasadnienia w rzeczywistych synapsach.

3.2 Modyfikacja metody Hebba

Zaprezentowany oryginalny wzór Hebba posiada istotną wadę. W trybie uczenia wartości wag rosną w sposób nieograniczony, co może prowadzić do bardzo dużych wartości. Z tego względu stosuje się pewną modyfikację reguły Hebba. Modyfikacja ta polega na zwiększaniu wartości synapsy, gdy wyjście jest aktywne oraz na jej osłabieniu, w przypadku, gdy wyjście jest nieaktywne. Za-

tem formuła uczenia sieci Hopfielda może być zapisana w iteracyjnej formie:

$$w_{ij}^{\mu+1} = w_{ij}^{\mu} + \eta \xi_i (2\xi_j - 1),$$

gdzie ξ_i , ξ_j stanowią tak jak poprzednio odpowiednio i -tą i j -tą współrzędną wektora uczącego, natomiast η jest współczynnikiem uczenia. Współczynnik ten dobierany jest najczęściej z przedziału $[0,7,0,9]$.

W kolejnej modyfikacji wzoru Hebba wartości synapsy są wzmacniane niezależnie od tego, czy wyjście jest aktywne, czy nieaktywne. Wówczas wagi dobierane są zgodnie z iteracyjnym wzorem:

$$w_{ij}^{\mu+1} = w_{ij}^{\mu} + \eta (2\xi_i - 1)(2\xi_j - 1),$$

przy czym ξ_i , ξ_j oraz η określone są jak poprzednio.

3.3 Metoda Oji

Aby zapobiec rozbieżności metody Hebba, można ograniczyć wzrost wektora wag. Oja zaproponował podejście polegające na pewnej modyfikacji reguły Hebba. Otóż wykazał on, że wektorom wag można nadać stałą długość równą 1, czyli

$$|w|=1,$$

gdzie w oznacza wektor wag.

Reguła Oji polega na dodaniu do oryginalnej metody Hebba wyrazu zanikania wagi, który jest proporcjonalny do V^2 , przy czym:

$$V = \sum_{i=1}^n w_{ij} \xi_j.$$

Wówczas wagi zgodnie z regułą Oji dobierane są w kolejnych krokach według następującego wzoru iteracyjnego:

$$w_{ij}^{m+1} = w_{ij}^m + \eta V (\xi_i - V w_{ij}^m).$$

3.4 Metoda pseudoinwersji

Punktem wyjścia w metodzie pseudoinwersji jest założenie, że każdy wzorec podany na wejście, generuje na wyjściu samego siebie. Takie założenie prowadzi oczywiście do stanu ustalonego. W zapisie macierzowym oznacza to, że:

$$WX=X,$$

przy czym W jest macierzą wag o wymiarach $n \times m$ natomiast X macierzą prostokątną o wymiarach $n \times p$ gdzie p oznacza kolejne wektory uczące, czyli $X=[x^{(1)}, x^{(1)}, \dots, x^{(p)}]$. Rozwiązaniem takiego układu równań jest macierz

$$W = XX^+$$

gdzie "+" oznacza pseudoinwersję.

Jeśli natomiast wektory uczące są liniowo niezależne, pseudoinwersję macierzy o wymiarach $n \times p$ można zastąpić zwykłą inversją macierzy o wymiarach $p \times p$. Wówczas wagi dobierane są według wzoru:

$$W = X(X^T X)^{-1} X^T.$$

Powyższy wzór można zapisać w postaci iteracyjnej, która nie wymaga kłopotliwego wyznaczania macierzy odwrotnej. W tym przypadku wagi dobierane są w kolejnych krokach zgodnie z zależnością:

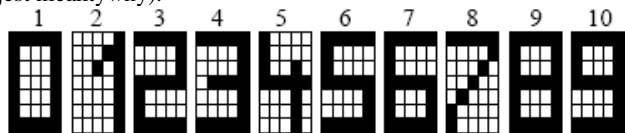
$$W^{(i+1)} = W^{(i)} + \frac{(W^{(i)} x^{(i)} - x^{(i)})(W^{(i)} x^{(i)} - x^{(i)})^T}{[x^{(i)}]^T x^{(i)} - [x^{(i)}]^T W^{(i)} x^{(i)}},$$

przy warunku początkowym $W^{(0)}=0$. Zależność ta wymaga jednokrotnej prezentacji wszystkich p wzorców uczących. W wyniku tych operacji macierz wag przyjmuje wartość ustaloną $W=W^{(p)}$.

4. Przebieg testów

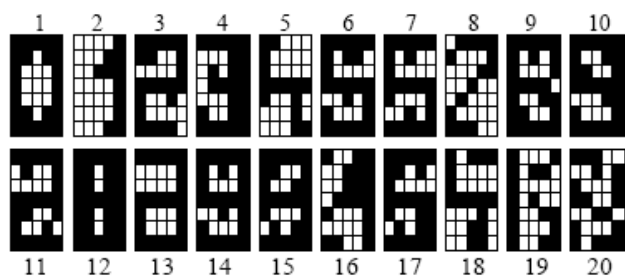
W celu praktycznego porównania zaprezentowanych w artykule metod uczenia sieci neuronowej Hopfielda, napisana została aplikacja, w której zaimplementowane zostały opisane metody.

Zadanie, jakie zostało postawione przed siecią Hopfielda, polegało na zapamiętaniu 10 cyfr (0, 1, 2, 3, 4, 5, 6, 7, 8 oraz 9), a następnie, po dostarczeniu do sieci zaburzonych wzorców, próba odtworzenia wzorców zapamiętanych. Cyfry zapamiętane zostały w układzie pikselowym, o wymiarze 5×7 . Liczba neuronów wynosi zatem 35. Czarnemu pikselowi odpowiada 1 (czyli gdy neuron jest w stanie wzbudzonym), natomiast białemu 0 (gdy neuron jest nieaktywny).



Rys. 2. Zapamiętane wzorce
Fig. 2. Stored patterns

W celu przetestowania opisanych metod uczenia sieci neuronowej Hopfielda dobrane zostały wzorce testujące o różnym stopniu zaszumienia.



Rys. 3. Zaszumione wzorce użyte w testowaniu sieci Hopfielda
Fig. 3. Distorted patterns used to test Hopfield neural network

Tab. 1. Porównanie działania metod uczenia sieci Hopfielda.

Wzorce ponumerowane kolejno zgodnie z rys. 3

Tab. 1. Comparison learning methods of Hopfield neural network.
Patterns are numbered according to fig. 3

Zaburzony wzorec podany na wejście sieci	Numer rozpoznanego wzorca metodą:			
	Hebba	modyfikacjami Hebba	Oji	pseudoinwersji
1	×	×	×	1
2	×	×	×	2
3	×	×	×	3
4	4	4	4	4
5	×	×	×	×
6	6	6	6	6
7	6	6	6	6
8	×	×	×	×
9	×	×	×	9
10	10	10	10	10
11	×	×	×	3
12	10	10	10	9
13	4	4	4	×
14	10	10	10	10
15	10	10	10	10
16	×	×	×	×
17	6	6	6	6
18	×	×	×	×
19	×	×	×	×
20	×	×	×	×

Znak × w tabeli oznacza, że żaden z zapamiętanych wzorców nie został poprawnie rozpoznany, co oznacza, że dana metoda utknęła w jednym z minimów lokalnych.

5. Wnioski

Na podstawie przeprowadzonych testów widać, że oryginalna metoda Hebba oraz jej modyfikacje (opisane w rozdziale 3 w punkcie b) dają te same wyniki, tj. umożliwiają rozpoznanie tych samych wzorców po zaprezentowaniu takich samych danych zaburzonych. Spowodowane jest to niewielkimi modyfikacjami danych reguł, które tylko nieznacznie różnią się odpowiednio wzmacnianiem, bądź też osłabianiem sygnałów pochodzących z danej synapsy. Podobnie zachowuje się metoda Oji, która bazuje na oryginalnej metodzie Hebba. Metody te pozwoliły na rozpoznanie jedynie 9 na 20 dostarczonych do sieci zaburzonych danych wejściowych. Skuteczniejszą okazała się metoda pseudoinwersji, która umożliwiła rozpoznanie 13 z 20 zaprezentowanych zaburzonych wzorców. Warto zwrócić uwagę na fakt, że metoda pseudoinwersji nie odnajdywała żadnego ze wzorców, podczas gdy skuteczne okazały się reguły Hebba i jej modyfikacje, np. dla wzorca o numerze 13. Zastosowanie do uczenia sieci neuronowej Hopfielda reguły pseudoinwersji spowodowało rozpoznanie innego wzorca, niż w przypadku uczenia sieci metodą Hebba. Sytuacja ta ma miejsce dla wzorca numer 12. Zastosowanie w uczeniu reguły Hebba spowodowało rozpoznanie przez sieć cyfry 9, natomiast w przypadku uczenia sieci metodą pseudoinwersji, spowodowało rozpoznanie cyfry 8. Reguła Hebba i jej modyfikacje pozwoliła na rozpoznanie tylko 3 wzorców (cyfry 9 – cztery razy, cyfry 5 – trzy razy i cyfry 3 – dwa razy), natomiast reguła pseudoinwersji pozwoliła na rozpoznanie (cyfry 5 – trzy razy, cyfry 9 – trzy razy, cyfry 2 – dwa razy, cyfry 8 – dwa razy, cyfry 0 – jeden raz, cyfry 1 – jeden raz oraz cyfry 3 – jeden raz). W sumie najczęściej rozpoznawanym wzorcem była cyfra 9 (siedmiokrotnie) oraz 5 (sześciokrotnie). Przy tak wybranych zaburzonych wzorcach żadna z reguł uczenia nie pozwoliła na rozpoznanie cyfr 4, 6 oraz 7.

Wszystkie z zaprezentowanych w artykule metod uczenia sieci neuronowej Hopfielda poprawnie rozpoznają zapamiętane wzorce w przypadku mniej zaburzonych danych wejściowych. Dane zniekształcone w dużym stopniu (18-20) powodują, że także skuteczniejsza metoda pseudoinwersji kończy działanie w jednym z minimów lokalnych nie rozpoznając żadnego z zapamiętanych wzorców.

6. Literatura

- [1] R. Tadeusiewicz: Sieci neuronowe. Akademicka Oficyna Wydawnicza RM, Warszawa 1993.
- [2] S. Osowski: Sieci neuronowe do przetwarzania informacji. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2000.
- [3] S. Osowski: Sieci neuronowe w ujęciu algorytmicznym. Wydawnictwa Naukowo-Techniczne, Warszawa 1996.
- [4] J. Mańdziuk: Sieci neuronowe typu Hopfielda. Teoria i przykłady zastosowań. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2000.
- [5] J. Hertz, A. Krogh, R. G. Palmer: Wstęp do teorii obliczeń neuronowych. Wydanie drugie. Wydawnictwa Naukowo-Techniczne, Warszawa 1993, 1995
- [6] J. Żurada, M. Barski, W. Jędruch: Sztuczne sieci neuronowe. Wydawnictwo Naukowe PWN, Warszawa 1996.
- [7] J. Korbicz, A. Obuchowicz, D. Uciński: Sztuczne sieci neuronowe. Podstawy i zastosowania, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1994.
- [8] W. Kwiatkowski: Metody automatycznego rozpoznawania wzorców. IAI R WAT, Warszawa 2001.
- [9] R. A. Kosiński: Sztuczne sieci neuronowe. Dynamika nieliniowa i chaos. Wydanie trzecie uaktualnione. Wydawnictwa Naukowo-Techniczne, Warszawa 2002, 2007
- [10] M. M. Gupta, L. Jin, N. Homma: Static and Dynamic Neural Networks. From Fundamentals to Advanced Theory. Wiley-Interscience, New Jersey 2003