SIEĆ HOPFIELDA

Spis treści

1. Wstęp

W roku 1982 John Hopfield zaprezentował w swojej pracy sieć ze sprzężeniem zwrotnym, która okazała się punktem zwrotnym w badaniach nad sieciami neuronowymi i przyczyniła się do intensywnych prac badawczych w tej dziedzinie. Stała się ona podstawowym przedstawicielem sieci rekurencyjnych, w których istnieje sprzężenie zwrotne między warstwami, a ze względu na pełnioną funkcję sieć Hopfielda często nazywana jest pamięcią autoasocjacyjną.

Początkowa działalność naukowa Johna Hopfielda nie była związana z badaniami sztucznych sieci neuronowych. Jednak możliwość przetwarzania przez system nerwowy ogromnej ilości informacji, zaintrygowała tego fizyka z Działu Biofizyki Laboratoriów Bella na tyle, że poświęcił się on intensywnym badaniom nad jego sztucznymi odpowiednikami. Wynikiem pracy było zaprezentowanie sieci ze sprzężeniem zwrotnym, którą można było wykorzystać do odtwarzania obrazów z ich fragmentów oraz do rozwiązywania zadań optymalizacyjnych.

Sieci Hopfielda są tak ważne głównie dlatego, że taka sieć może być traktowana jako neuronowy model pewnych interesujących systemów fizycznych np. szkieł spinowych, nad badaniem których pracował m.in. Hopfield Procesy zachodzące w tych sieciach są zawsze stabilne, można więc bezpiecznie stosować je do rozwiązywania różnych zadań

2. Sieć Hopfielda jako pamięć autoskojarzeniowa

Sieć Hopfielda często nazywana jest pamięcią autoasocjacyjna lub autoskojarzeniową, której zasada działania związana jest z jedną z podstawowych funkcji mózgu. Koncentrując się potrafimy zrozumieć mocno niewyraźną mowę lub odczytać prawie nieczytelne pismo. Wiele teleturniejów opartych jest na zasadzie odgadywania przez uczestników tytułu znanego utworu muzycznego na podstawie jedynie kilku jego dźwięków, rozwiązywaniu krzyżówek lub całego hasła znając tylko kilka jego liter.

Jak można opisać pojęcie pamięci autoskojarzeniowej w języku teorii systemów? Podstawowym zadaniem pamięci asocjacyjnej jest zapamiętanie zbioru wzorców w taki sposób aby system podczas prezentacji nowego wzorca mógł wygenerować odpowiedz, która będzie odpowiadać jednemu z zapamiętanych wcześniej wzorców. Korzystając z badań zawartych w pracy [1] załóżymy, że dany jest zbiór M różnych wzorców $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)},\ldots, \boldsymbol{x}^{(M}\} \subset R^N$. Pamięcią autoskojarzeniową (autoasocjacyjną) związanym z tym zbiorem nazywamy układ realizujący odwzorowanie $F : R^N \rightarrow R^N$ takie, że $F(\boldsymbol{x}^{(m)}) = \boldsymbol{x}^{(m)}$, $m = 1,\ldots,M$, oraz $F(\boldsymbol{x}) = \boldsymbol{x}^{(l)}$, jeżeli $\boldsymbol{x}^{(l)}$ jest najbardziej „podobny” do $\boldsymbol{x}$ spośród wszystkich wzorców. Stopień podobieństwa (a właściwie różności) dwóch wektorów $\mathbf{x}$ i $\boldsymbol{y}$ najczęściej

wyznaczana jest przy użyciu miary Hamminga, która jest łączną sumą pozycji, na których różną się dwa ciągi bitowe. W przypadku wielkości binarnych (0,1) odległość Hamminga dwóch wektorów $\boldsymbol{x} = [x_1, x_2,\ldots, x_n]^T$ i $\boldsymbol{y} = [y_1, y_2,\ldots, y_n]^T$ definiuje się w postaci [2]

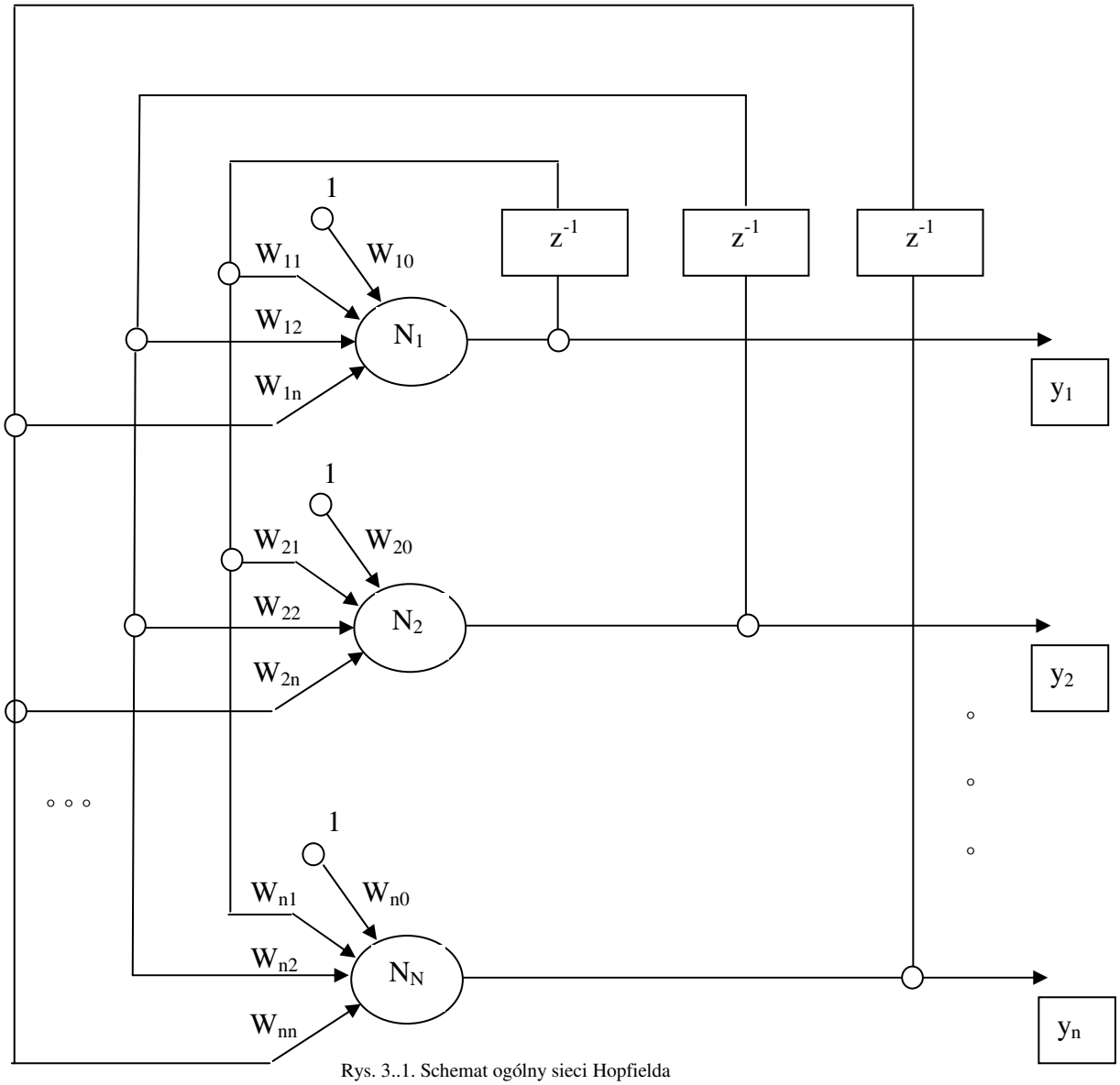
$$d_H(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{n}\left[x_i(1-y_i)+(1-x_i)y_i\right] \tag{2.1}$$

Przykład:

$$\left.\begin{array}{l} x = (1,1,0,0,1,1) \\ y = (1,0,1,0,0,0) \end{array}\right\} \Rightarrow d_H(x,y) = 4 \tag{2.2}$$

3. Struktura sieci Hopfielda

Sieć Hopfielda można przedstawiać na wiele sposobów - zwykle jednak przyjmuje się układ w postaci bezpośredniego sprzężenia wyjścia z wejściem (Rys. 3.1)[2].



Rys. 3..1. Schemat ogólny sieci Hopfielda

Wszystkie sygnały wyjściowe traktowane są jako wejścia $x_i(k) = y_i(k-1)$ i wszystkie wejścia przenoszą sygnały zwrotne do wszystkich neuronów. W układzie nie występuje sprzężenie zwrotne neuronu z jego własnym wejściem ($w_{ij} = 0$ gdy $i = j$). Istnieje jednak możliwość wpływania przez sygnał wyjściowy z danego neuronu na swoją wartość w przyszłości dzięki sprzężeniu zwrotnemu dodatkowych neuronów pośredniczących. Działanie dodatkowych neuronów wpływa bardzo stabilizująco na taki sygnał.

Macierz wag układu jest symetryczna ($w_{ij} = w_{ji}$), co oznacza, że współczynnik określający połączenie neuronu *i*-te*go* z *j*-tym ma taką samą wartość co *j*-ty z *i*-tym.

4. Zasada działania sieci Hopfielda

Sieć składa się z *N*-neuronów, z których w danej chwili czasu *k* aktywny jest tylko jeden element przetwarzający *p*, który wybiera się losowo (każdy neuron z jednakowym prawdopodobieństwem), a zmiany stanu sieci następują tylko w dyskretnych chwilach czasu. Hopfield opisał zasadę działania elementu przetwarzającego zależnościami [1]:

$$\varphi_p(k) = \sum_{j=1}^{N} \omega_{pj} x_j(k) - \theta_p \tag{4.1}$$

$$x_p(k+1) = \begin{cases} 1 & gdy \quad \varphi(k) > 0, \\ x_p(k) & gdy \quad \varphi(k) = 0, \\ 0 & gdy \quad \varphi(k) < 0, \end{cases} \tag{4.2}$$

gdzie $x_j(k)$ oznacza wyjście *j*-tego elementu przetwarzającego w chwili k; w_{pj} – wagę połączenia między wyjściem *j*-tego i wejściem *p*-tego elementu; składnik θ_p oznacza wartość progową każdego neuronu, a jego działanie opisano równaniem [1]:

$$x_p(k+1) = \begin{cases} 1 & gdy \quad \sum_{j=1}^{N} w_{pj} x_j(k) > \theta_p, \\ x_p(k) & gdy \quad \sum_{j=1}^{N} w_{pj} x_j(k) = \theta_p, \\ 0 & gdy \quad \sum_{j=1}^{N} w_{pj} x_j(k) < \theta_p, \end{cases} \tag{4.3}$$

W działaniu sieci Hopfielda można wyróżnić dwa tryby pracy:

- tryb uczenia
- tryb odtwarzania

5. Tryb uczenia sieci Hopfielda

Proces uczenia sieci polega na takim doborze wag połączeń sieciowych, aby prezentowany wzorzec stał się jednym atraktorów (stanów stabilnych).

Hopfield zaproponował sposób modyfikacji wag korzystając z reguły Hebba, w której dla M różnych wzorców $x^{(1)},\ldots, x^{(M)} \in \mathrm{R}^N$ wartości wag w_{ij} należy zwiększyć o 1, gdy *i*-ta i *j*-ta składowa danego wzorca są sobie równe ($x_i^{(m)} = x_j^{(m)}$). Natomiast gdy składowe są sobie różne, wartości wag w_{ij} należy pomniejszyć o 1 [1]:

$$w_{ij} = \begin{cases} \sum_{m=1}^{M} (2x_i^{(m)} - 1)\,(2x_j^{(m)} - 1) & gdy\ i \neq j \\ 0 & gdy\ i = j \end{cases} \tag{5.1}$$

Taki tryb uczenia powoduje, że wartościami wag są średnie wszystkich wzorców uczących, czyli informacja o danej próbce zawarta jest we wszystkich wagach w_{ij}. Jest to podstawowa zaleta sieci Hopfielda, która jest czynnikiem wpływającym na odporność sieci na uszkodzenia. Zmiana wag połączeń lub uszkodzenie któregoś z neuronów mają nieznaczny wpływ na działanie sieci. Dlatego sieć Hopfielda znalazła szerokie zastosowanie w systemach, w których jego uszkodzenie oznaczałoby poważne konsekwencje. Wykorzystywana jest m.in. w systemach sterowania statków kosmicznych, elektrowniach atomowych itp.

Czym są atraktory? Są to stany stabilne, odpowiadające minimum funkcji energii, która jest bardzo ważnym czynnikiem sieci Hopfielda. Korzystając z twierdzeń zawartych w pracy [1] możemy stwierdzić, że każdy stan sieci określony przez zbiór aktualnych wartości wyjść jej neuronów, określa pewną wartość tej funkcji, którą najczęściej definiuje się w postaci [1]:

$$E(x) = -\frac{1}{2} x^T W x + \theta^T x = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} x_i x_j + \sum_{i=1}^{N} \theta_i x_i \tag{5.2}$$

gdzie

$$x(k) = [x_1(k)\ \ x_2(k)\ \ \ldots\ \ x_N(k)]^T \tag{5.3}$$

$$\theta = [\theta_1\ \ \theta_2\ \ \ldots\ \ \theta_N]^T \tag{5.4}$$

$$W = \begin{bmatrix} w_{11} & w_{22} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \dots & \dots & \dots & \dots \\ w_{N1} & w_{N2} & \dots & w_{NN} \end{bmatrix} \tag{5.5}$$

Zbiór wartości funkcji energii jest ograniczony, gdyż [1]:

$$\left| E(x) \right| \leq \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left| w_{ij} \right| + \sum_{i=1}^{N} \left| \theta_i \right| \tag{5.6}$$

Działanie sieci polega na generowaniu zmian sygnałów wyjściowych wszystkich jej elementów przetwarzających, aby wartość funkcji energetycznej malała. Funkcja energetyczna jest ciągła, w wyniku czego zmierza do najbliższego minimum lokalnego, a po jego osiągnięciu procesy zachodzące w sieci zanikają, ponieważ dalsze zmiany wartości wag nie spowodują uzyskania lepszego rozwiązania.

Istotną wadą sieci Hopfielda jest możliwość powstawania fałszywych atraktorów. Jeśli wzorce są ortogonalne tzn. $x^{(k)}x^{(l)} = 0$ dla $l \neq k$, to w roli atraktorów występują tylko wzorce uczące. W rzeczywistości taka sytuacja występuje bardzo rzadko, ponieważ wzorce przeważnie Sącze sobą w jakiś sposób skorelowane. Powoduje to powstawanie tzw. fałszywych atraktorów, które nie odpowiadają żadnemu ze wzorców.

Znaczne lepsze efekty można uzyskać zamieniając podczas uczenia regułę Hebba na metodę pseudoinwersji. Głównym założeniem tej metody jest uzyskanie na wyjściu sieci wzorca podanego na jej wejście przy odpowiednio dobranych wagach. Korzystając z twierdzeń zawartych w pracy [2] możemy przedstawić powyższe założenia w zapisie macierzowym [2]:

$$WX = X \tag{5.7}$$

gdzie W jest macierzą wagową o wymiarach $N \times N$, a X jest macierzą prostokątną o wymiarach $N \times p$, złożoną p kolejnych wektorów uczących $x^{(i)}$, *to jest* $X = [x^{(1)}, x^{(2)}, \dots x^{(p)}]$
Rozwiązaniem takiego układu liniowego jest [2]:

$$W = XX^{+} \tag{5.8}$$

gdzie znak + oznacza pseudoinwersji. Jeśli wektory uczące są liniowo niezależne może być uproszczona i przedstawiona w postaci [2]:

$$W = X(X^T X)^{-1} X^T \tag{5.9}$$

Pseudoinwersji macierzy o wymiarach $N \times p$ została zastąpiona zwykłą inwersją macierzy kwadratowej $\boldsymbol{X^T X}$ o wymiarach $p \times p$. Zależność (5.9) nosi nazwę *metody rzutowania.*

6. Tryb odtworzeniowy sieci Hopfielda

Proces odtwarzania rozpoczyna się w chwili początkowej $k = 0$ od wprowadzenia do neuronów sygnałów wejściowych, które są stanem początkowym sieci. Jest to proces iteracyjny, a stan całej sieci zmienia się w sposób zapewniający obniżenie wartości funkcji energii. Długość tego procesu zależy od wielkości sieci i ukształtowania minimów lokalnych, a po jego zakończeniu sieć osiąga stan stabilny, w którym zachodzi [1]:

$$x_i(k+1) = x_i(k), \ \forall\, i \ \ \{i = 1,..., N\} \tag{6.1}$$

Po ustaleniu się odpowiedzi, która jest wyznaczeniem minimum funkcji energii, proces odtwarzania jest przerywany, a stan sieci przekazywany jest na jej wyjście.

Nie zawsze jednak na wyjściu sieci uzyskuje się właściwą odpowiedz, która odpowiada jednemu z wcześniej zapamiętanych wzorców. Jednym z powodów mogą być fałszywe atraktory czyli pośrednie minima lokalne, w których proces może utknąć. Powoduje to uzyskanie odpowiedzi na wyjściu sieci, która nie odpowiada żadnemu stanowi neuronów biorącemu udział w procesie uczenia.

Drugim powodem niepoprawnego odtwarzania wzorców na wyjściu sieci może być możliwość mieszania różnych składowych zapamiętanych wzorców. Wynikiem jest uzyskanie wyższego poziomu energetycznego niż stan pożądany.

Możliwa jest także zmiana polaryzacji stanów elementów przetwarzających, co powoduje generowanie fałszywych wyników na wyjściu sieci.

Badania sieci Hopfielda przeprowadzone w *Instytucie Elektrotechniki Teoretycznej i Miernictwa Elektrycznego Politechniki Warszawskiej* przy użyciu programu *Hfnet* wykazały, że skuteczność odtwarzania wzorców na wyjściu sieci jest zależne od wykorzystanej metody uczenia sieci. Bardzo słabe efekty dała metoda Hebba, natomiast metoda rzutowania prawie bezbłędnie odtwarzała wzorce.

7. Pojemność sieci

Bardzo ważnym parametrem pamięci asocjacyjnej jest jej pojemność, która jest maksymalną liczbą wzorców zapamiętanych i odtworzonych z maksymalnie określonym błędem. Dla błędu wynoszącego 1% i przy wykorzystaniu w procesie uczenia reguły Hebba, maksymalna pojemność pamięci stanowi 13,8% liczby neuronów tworzących pamięć asocjacyjną. Pojemność ta może ulec zmniejszeniu w wyniku niewłaściwego stanu początkowego sieci, który w wyniku sprzężeń zwrotnych powoduje dalsze powielanie błędu co jest równoznaczne ze zmniejszaniem pojemności pamięci.

Zastosowanie metod pseudoinwersji zwiększa pojemność maksymalną sieci Hopfielda do N-1, gdzie N jest liczbą neuronów tworzących pamięć asocjacyjną.

8. Zastosowanie sieci Hopfielda

Sieci Hopfielda znajdują liczne zastosowania przy rozwiązywaniu zadań optymalizacji i przy generacji określonych sekwencji sygnałów, następujących po sobie w pewnej kolejności. Pozwala to za pomocą takich sieci tworzyć i wysyłać do różnych obiektów sygnały sterujące. Układy sterujące ruchami maszyn kroczących zawsze zawierają sieć posiadającą sprzężenia zwrotne, najczęściej sieć Hopfielda.

Dzięki autoasocjacji wytrenowana sieć Hopfielda może automatycznie korygować błędne albo uzupełniać niekompletne dane. Sieć może odtworzyć dokładną sylwetkę zbliżającego się obiektu w sytuacji, gdy kamera zarejestrowała obraz niekompletny lub zaszumiany. Ma to olbrzymie znaczenie w działaniach militarnych, gdzie na podstawie takiego niekompletnego obrazu system musi rozstrzygnąć, czy obiekt należy do wroga czy jest jednostką należącą do naszych struktur. Sieć pracująca jako pamięć autoasocjacyjna może być także wykorzystywana w systemach bazodanowych do uzupełniania niekompletnych zapytań, dzięki czemu wyszukanie będzie wykonywane poprawnie nawet w przypadku nieprecyzyjnego zapytania. Wysoka skuteczność sieci Hopfielda wynika z faktu, że sieć odtwarza sygnały wzorcowe ze swoich zasobów pamięciowych, a zniekształcony sygnał podany na wejście ma ją tylko ukierunkować przy wyborze właściwego obrazu spośród wszystkich analizowanych.

9. Przykłady zastosowania sieci Hopfielda w programie Matlab

9.1 Projektowanie sieci Hopfielda przy użyciu funkcji *newhop*

Chcemy zaprojektować sieć Hopfielda składającą się z trzech neuronów do zapamiętania dwóch wzorców:

```
T = [-1 -1 1; 1 -1 1]'
T =
     -1     1
     -1    -1
      1     1
```

Projektujemy (uczymy) sieć:

```
net = newhop(T);
```

Możemy odczytać wartość współczynników wagowych:

```
W= net.LW{1,1}
```

```
W =
    1.1618         0         0
         0    0.2231         0
         0         0    0.2231
```

Możemy odczytać wartości biasów:

```
b = net.b{1,1}
b =
         0
   -0.8546
    0.8546
```

Możemy sprawdzić działanie sieci poprzez jej symulacje:

```
Ai = T;
[Y,Pf] = sim(net,2,[],Ai);
```

gdzie

`net` – sieć

`2` – wejścia sieci

`[]` – wartości początkowe wejść (domyślnie 0)

`Ai` – punkt startowy sieci

`Y` – wyjścia sieci

`Pf` – stan wejść

Otrzymamy odpowiedź

```
Y =
    -1     1
    -1    -1
     1     1
```

co będzie oznaczało poprawnie wykonany projekt.

Możemy teraz podać na wejście sieci wzorzec, który nie był wykorzystany podczas uczenia sieci:

```
Ai = {[-0.9; -0.8; 0.7]}
```

Wzorzec ten jest bardzo zbliżony do pierwszego wzorca użytego do projektowania sieci, więc na jej wyjściu spodziewamy się go uzyskać.

```
[Y,Pf,Af] = sim(net,{1 5},{},Ai);
```

gdzie 5 jest liczbą kroków.

Otrzymamy odpowiedź

```
Y{1}

Y =
    -1
    -1
     1
```

Tak więc sieć działa zgodnie z założeniami działania sieci Hopfielda.

9.2 Graficzna interpretacja działania sieci Hopfielda

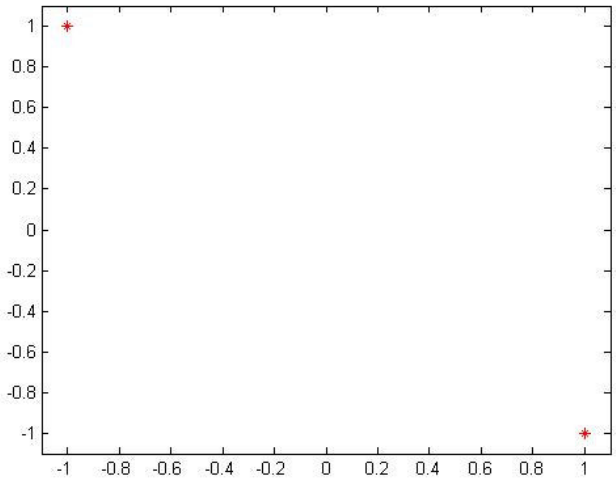
Zaprojektujmy sieć Hopfielda składającą się z dwóch neuronów do zapamiętania dwóch wzorców:

```
T = [1 -1;-1 1]'

T =
     1    -1
    -1     1
```

Postać graficzna:

```
plot(T(1,:),T(2,:),'r*')
```



Rys. 9.2.1. Układ współrzędnych z dwoma punktami w przestrzeni.

Projektujemy (uczymy) sieć:

```
net = newhop(T);
```

Sprawdzamy działanie sieci poprzez jej symulacje:

```
[Y,Pf] = sim(net,2,[],T);
```

Otrzymamy odpowiedź

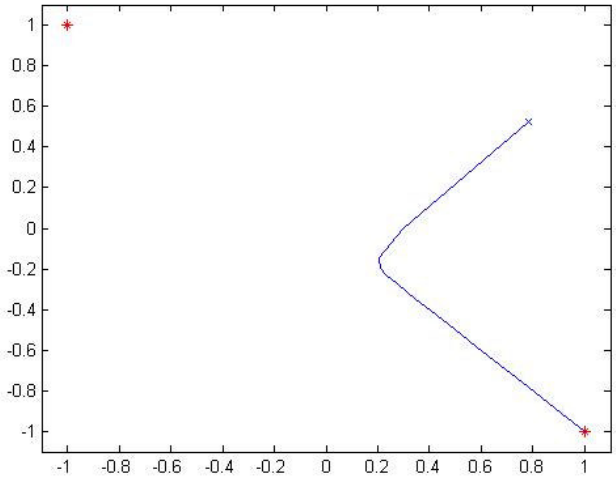
```
Y =
     1    -1
    -1     1
```

Wylosujmy punkt startowy i zasymulujmy działanie sieci w 20 krokach:

```
a = {rands(2,1)};
[y,Pf] = sim(net,{1 20},{},a);
```

Postać graficzna:

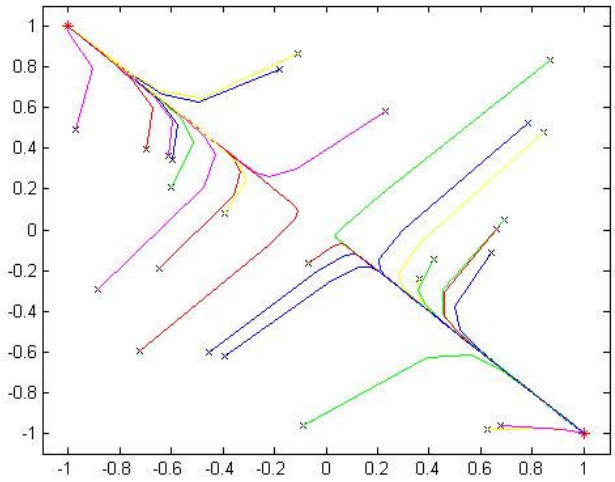
```
record = [cell2mat(a) cell2mat(y)];
start = cell2mat(a);
hold on
plot(start(1,1),start(2,1),'bx',record(1,:),record(2,:))
```



Rys. 9.2.2. Tryb odtwarzania sieci Hopfielda dla jednego wzorca.

Powtórzmy działanie dla 25 innych losowo wybranych punktów:

```
color = 'rgbmy';
for i=1:25
   a = {rands(2,1)};
   [y,Pf] = sim(net,{1 20},{},a);
   record=[cell2mat(a) cell2mat(y)];
   start=cell2mat(a);
   plot(start(1,1),start(2,1),'kx',record(1,:),
   record(2,:),color(rem(i,5)+1))
end
```

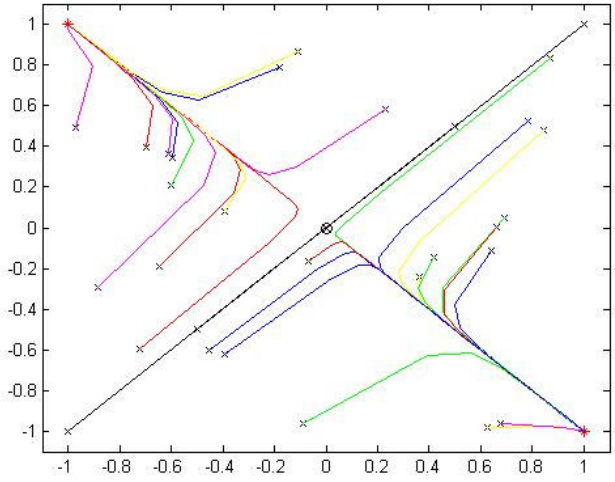


Rys. 9.2.3. Tryb odtwarzania sieci Hopfielda dla 25 wzorców.

Doskonale widać tu graficzną definicję pamięci autoasocjacyjnej: układ realizujący odwzorowanie $F : R^N \rightarrow R^N$ takie, że $F(\boldsymbol{x}^{(m)}) = \boldsymbol{x}^{(m)}$, $m = 1,\ldots,M$, oraz $F(\boldsymbol{x}) = \boldsymbol{x}^{(l)}$, jeżeli $\boldsymbol{x}^{(l)}$ jest najbardziej „podobny” do $\boldsymbol{x}$ spośród wszystkich wzorców.

Możemy wyznaczyć także „granicę przyciągania” każdego z punktów:

```
for i=1:5
   a = {P(:,i)};
   [y,Pf] = sim(net,{1 50},{},a);
   record=[cell2mat(a) cell2mat(y)];
   start = cell2mat(a);
   plot(start(1,1),start(2,1),'kx',record(1,:),
   record(2,:),'k')
   drawnow
end
```



Rys. 9.2.4. „Granica przyciągania” każdego z punktów.

10. Program do symulacji sieci Hopfielda

W oparciu o dyskretna siec Hopfielda zaprojektowany zostanie system rozpoznawania znaków alfanumerycznych.

System będzie składał się z trzech części:

- wczytywanie plików ze znakami
- uczenie sieci
- testowanie sieci

Skrypt wczytujący znaki alfanumeryczne:

```
clear,
U=[];
for i=1:26,
    nazwa=strcat(int2str(i),'.bmp');
    a=imread(nazwa,'bmp');
    a=double(a);
    index=find(a==0);
    a(index)=-1;
    a=a(:);
    U=[U,a];
end;
```

gdzie nazwy plików przyjmują nazwy *1.bmp, 2.bmp, …, 26.bmp*

Do uczenia wykorzystane nie zostały opisane funkcje opisane wyżej, jednak działające w analogiczny sposób:

```
[W,B]=solvehop(U);
odp=simuhop(U,W,B);
blad=mse(odp-U)
```

gdzie

`solvehop()` – funkcja projektująca sieć Hopfielda

`simuhop()` – funkcja symulująca sieć Hopfielda

`W` – macierz współczynników wagowych

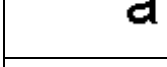
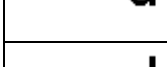
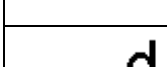
`B` – wektor biasów

`mse()` – funkcja obliczająca błąd średniokwadratowy

Ostatnim skryptem jest skrypt testujący działanie sieci i zapisująca wynik do pliku *odp.bmp*:

```
nazwa='1.bmp';
a=imread(nazwa,'bmp');
a=double(a);
index=find(a==0);
a(index)=-1;
a=a(:);
od=simuhop(a,W,B);
imwrite(reshape(od,40,30),'odp.bmp','bmp');
```

System bardzo dobrze rozpoznawaje znaki wykorzystane do uczenia sieci oraz znaki zaszumione. Niestety przesunięcie lub skalowanie badanego obrazu powodowało błędne działanie układu.

Znak użyty do nauki	Znak testujący	Odpowiedź sieci
		
		
		
		
		

10.1 Znaki wykorzystane do badania sieci

10. Literatura

1. Korbicz J., Obuchowicz A., Uciński D. *„Sztuczne sieci neuronowe. Podstawy i zastosowania.”* Akademicka Oficyna Wydawnicza PLJ, Warszawa 1994

2. Osowski S. *„Sieci neuronowe do przetwarzania informacji.”* Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2000

3. http://aneksy.pwn.pl/efw/?id=584

4. http://www.mathworks.com/access/helpdesk/help/pdf_doc/nnet/nnet.pdf