Alicja Kapiszka, Informatyka N1, gr. 30C

**Sieci neuronowe – sprawozdanie nr 5**

**Wstęp:**

Przedmiotem laboratorium była sieć Hopfielda. Jest to sieć autoasocjacyjna, która jest stosowana w celu porównania danych wejściowych do wzorców, które pamięta sieć. Zwraca ona na wyjściu wzorzec, do którego wejście jest najbardziej podobne. Sieci Hopfielda są sieciami rekurencyjnymi, czyli takimi, w których wyjścia neuronów są połączone z wejściami innych neuronów (sprzężenia zwrotne).

**Zadanie:**

Zadaniem do wykonania było samodzielne opracowanie skryptu, w którym powinna zostać stworzona macierz z wzorcami 10 znaków, każdy z nich zbudowany na matrycy min. 4x4 (u mnie było to 5x5) wypełnionej wartościami 1 (odpowiadającej pikselowi czarnemu) i -1 (pikselowi białemu). Następnie należało utworzyć sieć Hopfielda pamiętającą te wzorce i sprawdzić jej działanie dla znaków pamiętanych oraz zniekształconych.

Kod programu:

% utworzenie wzorcow znakow (litery A-G, cyfry 1-3) na matrycach 5x5

% -1 - piksel bialy, 1 - czarny

A = [-1 -1 1 -1 -1; -1 1 -1 1 -1; -1 1 1 1 -1; -1 1 -1 1 -1; -1 1 -1 1 -1];

B = [-1 1 1 1 -1; -1 1 -1 1 -1; -1 1 1 -1 -1; -1 1 -1 1 -1; -1 1 1 1 -1];

C = [-1 -1 1 1 -1; -1 1 -1 -1 -1; -1 1 -1 -1 -1; -1 1 -1 -1 -1; -1 -1 1 1 -1];

D = [-1 1 1 -1 -1; -1 1 -1 1 -1; -1 1 -1 1 -1; -1 1 -1 1 -1; -1 1 1 -1 -1];

E = [-1 1 1 1 -1; -1 1 -1 -1 -1; -1 1 1 -1 -1; -1 1 -1 -1 -1; -1 1 1 1 -1];

F = [-1 1 1 1 -1; -1 1 -1 -1 -1; -1 1 1 -1 -1; -1 1 -1 -1 -1; -1 1 -1 -1 -1];

G = [-1 -1 1 1 -1; -1 1 -1 -1 -1; -1 1 -1 1 1; -1 1 -1 -1 1; -1 1 1 1 -1];

m1 = [-1 -1 1 -1 -1; -1 1 1 -1 -1; -1 -1 1 -1 -1; -1 -1 1 -1 -1; -1 1 1 1 -1];

m2 = [-1 1 1 1 -1; -1 -1 -1 1 -1; -1 1 1 1 -1; -1 1 -1 -1 -1; -1 1 1 1 -1];

m3 = [-1 1 1 1 -1; -1 -1 -1 1 -1; -1 1 1 1 -1; -1 -1 -1 1 -1; -1 1 1 1 -1];

% przeksztalcenie matryc na wektory kolumnowe

vA = reshape(A, 25, 1);

vB = reshape(B, 25, 1);

vC = reshape(C, 25, 1);

vD = reshape(D, 25, 1);

vE = reshape(E, 25, 1);

vF = reshape(F, 25, 1);

vG = reshape(G, 25, 1);

v1 = reshape(m1, 25, 1);

v2 = reshape(m2, 25, 1);

v3 = reshape(m3, 25, 1);

% utworzenie macierzy z wzorcami (1 kolumna - 1 wzorzec)

M = [vA vB vC vD vE vF vG v1 v2 v3];

net = newhop(M); % utworzenie sieci Hopfielda

w = net.LW{1,1} % wagi sieci

b = net.b{1,1} % wagi wejsc progowych

% symulacja dzialania sieci dla wzorcow z macierzy M:

[y,pf,af] = sim(net, 10, {}, M)

y'

% dzialanie sieci dla wzorca a (znieksztalcone znaki):

a = [1 1 1 1 -1; -1 1 -1 -1 -1; -1 1 1 -1 -1; -1 1 -1 -1 -1; -1 1 1 -1 -1];

a2 = reshape(a, 25, 1);

[y,pf,af] = sim(net, {1 20}, {}, {a2}); % ilosc iteracji - 20

y1 = cell2mat(y)

a3 = reshape(y1(:,end),5,5);

subplot(1,2,1)

imagesc(a)

subplot(1,2,2)

imagesc(a3)

net = newhop(M); % utworzenie sieci Hopfielda

w = net.LW{1,1} % wagi sieci

b = net.b{1,1} % wagi wejsc progowych

% symulacja dzialania sieci dla wzorcow z macierzy M:

[y,pf,af] = sim(net, 10, {}, M)

y'

% dzialanie sieci dla wzorca a (znieksztalcone znaki):

a = [1 1 1 1 -1; -1 1 -1 -1 -1; -1 1 1 -1 -1; -1 1 -1 -1 -1; -1 1 1 -1 -1];

a2 = reshape(a, 25, 1);

[y,pf,af] = sim(net, {1 20}, {}, {a2}); % ilosc iteracji - 20

y1 = cell2mat(y)

a3 = reshape(y1(:,end),5,5);

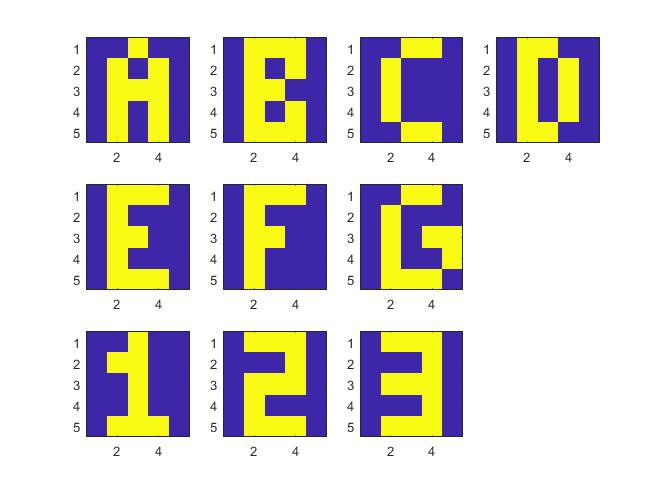
subplot(1,2,1)

imagesc(a)

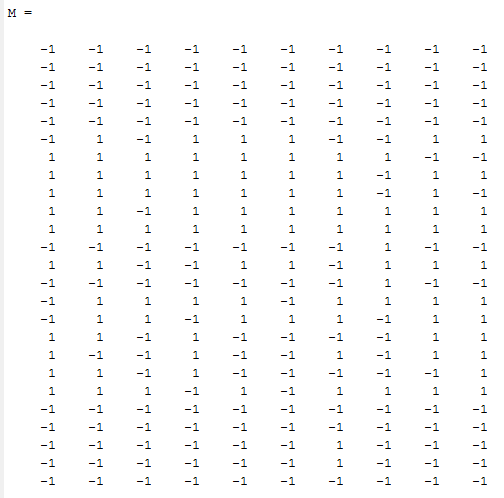
subplot(1,2,2)

imagesc(a3)

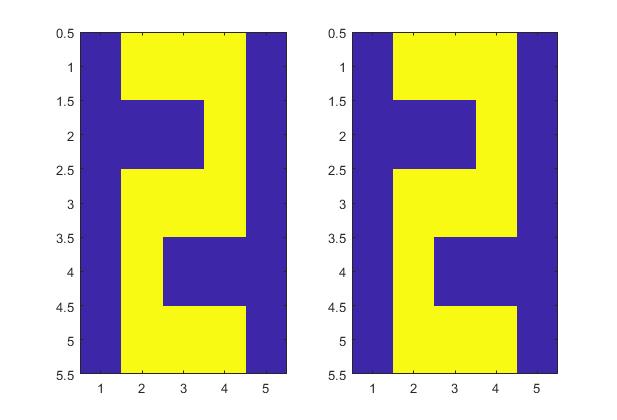
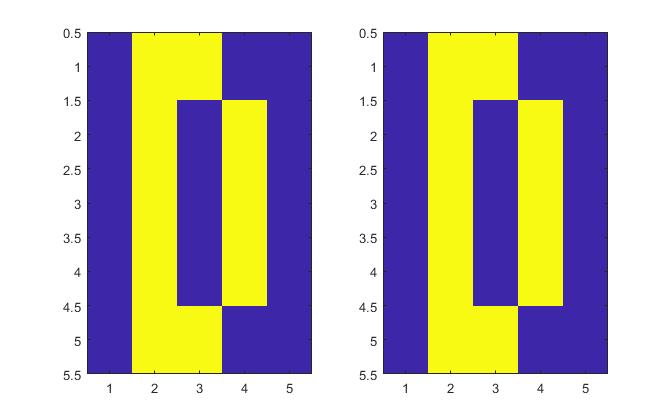
Znaki pamiętane przez sieć (litery A-G i cyfry 1-3):



Macierz wzorców:



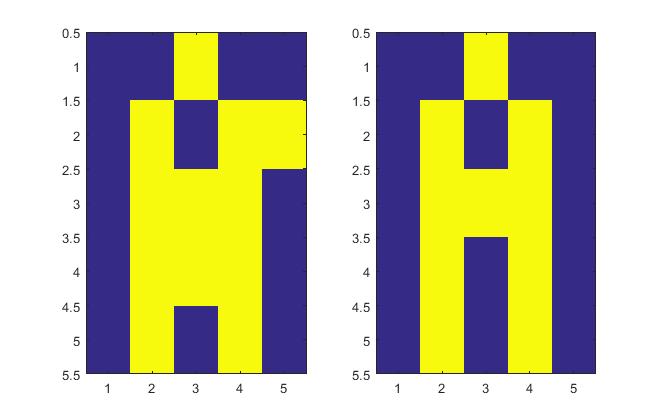
Przykład działania sieci dla pamiętanych znaków:



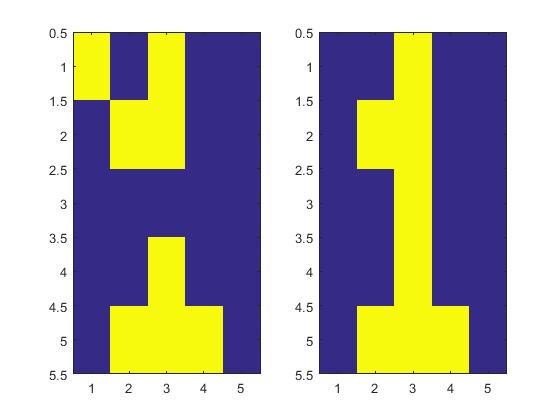
Przykład działania sieci dla zniekształconych znaków:

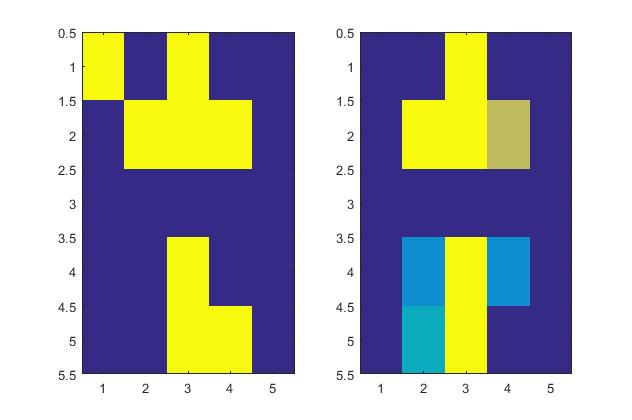
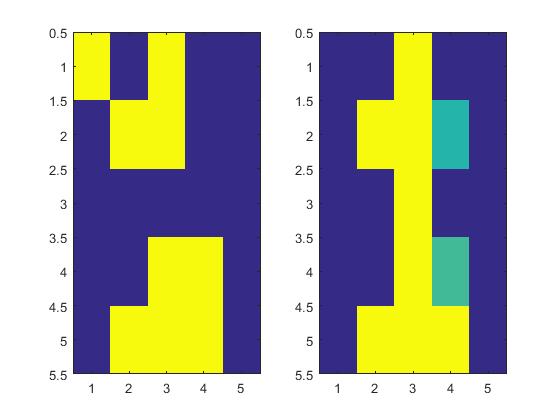
Na wejściu sieci podawano zniekształcone wzorce – ze zmienionymi kilkoma wartościami pikseli. W zależności od wielkości zmian, sieć rozpoznawała wzorzec lub nie. Jeśli sieć nie rozpoznała wzorca, podawała na wyjście tzw. fałszywe wzorce. Piksele o błędnych kolorach w fałszywych wzorcach odpowiadają nieprawidłowym wartościom ułamkowym (wzorce te mogą być podobne do właściwego wzorca, jeśli leżą blisko niego).

* Prawidłowe rozpoznanie wzorca litery A:

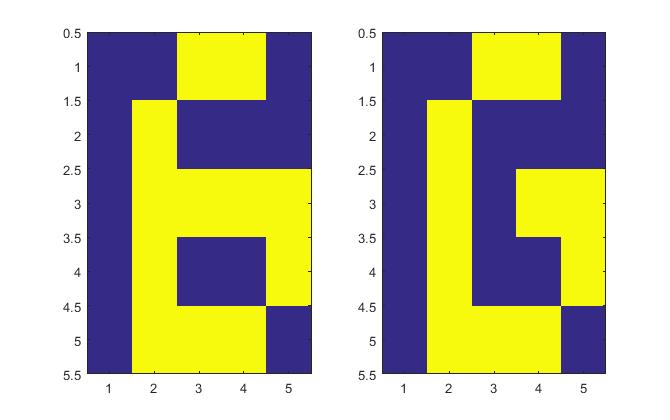
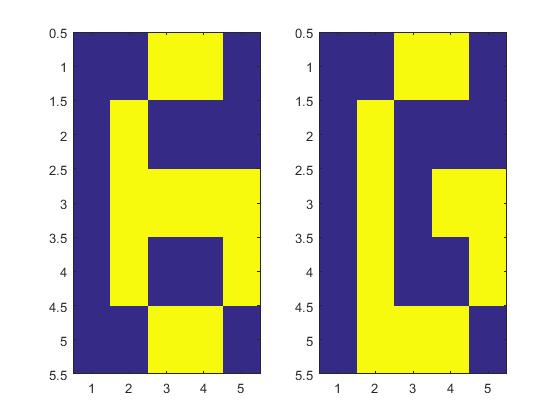


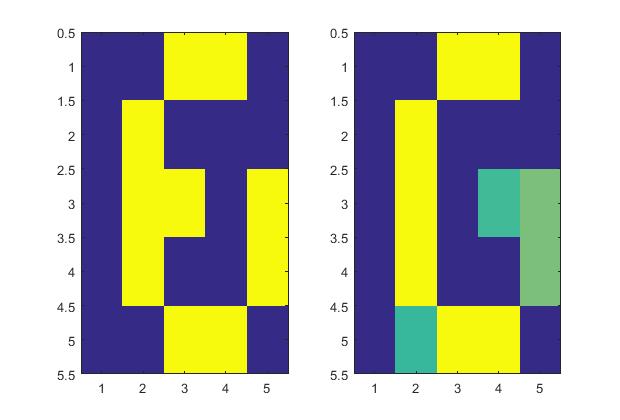
* Prawidłowe rozpoznanie wzorca cyfry 1 (wyżej) oraz podanie fałszywych wzorców, gdy zniekształcenie wejścia było zbyt duże (niżej):





* Prawidłowe (wyżej) i nieprawidłowe (niżej) rozpoznanie wzorca litery G:



* Nieprawidłowe rozpoznanie wzorca D i F:

