Biologically Inspired Computing

# EECS 6180

# Homework 5

# Hopfield Networks

# Edris Amin

Mar. 22, 2012

**Max. Pts 20**

**Use 1, -1 for black white**

1. **Design a Hopfield Neural Network to recognize the following two patterns.**
2. **Describe the network architecture. How many neurons will it require? 5pts**

This network will require as many neurons as there are pixels [16 neurons for 16 pixels]. The neurons will each have weight values representing the connections to all other neurons this will require each of the 16 neurons having 16 weights for a total number of 162 weights = 256 weights.

1. **Calculate the weight matrix. 5pts**

The weight matrix W is calculated using the following formula:

Where M is the total number of states (images) needed to be memorized and I is the identity matrix.

For calculating the weight matrix both patterns were made into a single row matrix so:

P1 = Patter 1 = [1 1 1 1 1 -1 -1 1 1 -1 -1 1 1 1 1 1]

P2 = Pattern 2 = [1 -1 -1 1 -1 1 1 -1 -1 1 1 -1 1 -1 -1 1]

%in matlab

P1 = [1 1 1 1 1 -1 -1 1 1 -1 -1 1 1 1 1 1];

P1t = P1';

P2 = [1 -1 -1 1 -1 1 1 -1 -1 1 1 -1 1 -1 -1 1];

P2t = P2';

Y1 = P1t\*P1;

Y2 = P2t\*P2;

W = Y1 + Y2 - 2\*eye(16)=

0 0 0 2 0 0 0 0 0 0 0 0 2 0 0 2

0 0 2 0 2 -2 -2 2 2 -2 -2 2 0 2 2 0

0 2 0 0 2 -2 -2 2 2 -2 -2 2 0 2 2 0

2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2

0 2 2 0 0 -2 -2 2 2 -2 -2 2 0 2 2 0

0 -2 -2 0 -2 0 2 -2 -2 2 2 -2 0 -2 -2 0

0 -2 -2 0 -2 2 0 -2 -2 2 2 -2 0 -2 -2 0

0 2 2 0 2 -2 -2 0 2 -2 -2 2 0 2 2 0

0 2 2 0 2 -2 -2 2 0 -2 -2 2 0 2 2 0

0 -2 -2 0 -2 2 2 -2 -2 0 2 -2 0 -2 -2 0

0 -2 -2 0 -2 2 2 -2 -2 2 0 -2 0 -2 -2 0

0 2 2 0 2 -2 -2 2 2 -2 -2 0 0 2 2 0

2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 2

0 2 2 0 2 -2 -2 2 2 -2 -2 2 0 0 2 0

0 2 2 0 2 -2 -2 2 2 -2 -2 2 0 2 0 0

2 0 0 2 0 0 0 0 0 0 0 0 2 0 0 0

1. **Test if these two patterns are classified correctly with that weight matrix. 5pts**

Testing Pattern1 with W:

%test P1

testP1 = W\*P1t;

%sign function

for i = 1:length(testP1)

if testP1(i) > 0

testP1(i) = 1;

elseif testP1(i) < 0

testP1(i) = -1;

else

testP1(i) = 0;

end

end

testP1’ =

1 1 1 1 1 -1 -1 1 1 -1 -1 1 1 1 1 1

P1 =

1 1 1 1 1 -1 -1 1 1 -1 -1 1 1 1 1 1

W works for Pattern 1 recognition

Testing Pattern2 with W:

%test second original image

testP2 = W\*P2t;

%sign function

for i = 1:length(testP2)

if testP2(i) > 0

testP2(i) = 1;

elseif testP2(i) < 0

testP2(i) = -1;

else

testP2(i) = 0;

end

end

testP2' =

1 -1 -1 1 -1 1 1 -1 -1 1 1 -1 1 -1 -1 1

P2=

1 -1 -1 1 -1 1 1 -1 -1 1 1 -1 1 -1 -1 1

W works for Pattern 2 recognition

|  |
| --- |
| figure9 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Fig. 2

1. **If a new pattern is given as shown in Fig. 2, to which category it will be classified? 5pts**

Testing figure 2 for recognition to Pattern 2 or Patter 1

For this test figure 2 was made into a row matrix in MATLAB:

fig2 = [1 -1 -1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 -1 -1 1];

fig2mat = vec2mat(fig2,4)

fig2 = fig2';

testfig2 = W\*fig2;

test2mat = vec2mat(testfig2, 4)

fig2mat =

1 -1 -1 -1

-1 1 -1 -1

-1 -1 1 -1

-1 -1 -1 1

test2mat =

-2 -14 -14 2

-14 14 18 -14

-14 18 14 -14

2 -14 -14 -2

% for iterations until conferging to a pattern

%check sign of pixels

for i = 1:length(testfig2)

if testfig2(i) > 0

testfig2(i) = 1;

elseif testfig2(i) < 0

testfig2(i) = -1;

else

testfig2(i) = 0;

end

end

resultmat = vec2mat(testfig2,4)

testfig2 = W\*testfig2;

Iteration 1:resultmat =

-1 -1 -1 1

-1 1 1 -1

-1 1 1 -1

1 -1 -1 -1

Iteration 2: resultmat =

1 -1 -1 -1

-1 1 1 -1

-1 1 1 -1

-1 -1 -1 1

Iteration 3: resultmat =

-1 -1 -1 1

-1 1 1 -1

-1 1 1 -1

1 -1 -1 -1

Iteration 4: resultmat =

1 -1 -1 -1

-1 1 1 -1

-1 1 1 -1

-1 -1 -1 1

Iteration 5: resultmat =

-1 -1 -1 1

-1 1 1 -1

-1 1 1 -1

1 -1 -1 -1

Iteration 6: resultmat =

1 -1 -1 -1

-1 1 1 -1

-1 1 1 -1

-1 -1 -1 1

Iteration 7: resultmat =

-1 -1 -1 1

-1 1 1 -1

-1 1 1 -1

1 -1 -1 -1

Iteration 8: resultmat =

1 -1 -1 -1

-1 1 1 -1

-1 1 1 -1

-1 -1 -1 1

After at 8 iterations of the sign check it seems that the network would converge to choosing Patter2 for figure 2. This will NOT be an exact match.