Hopfield network

An example of the work with Hopfield network:

```
m = [1 -1 1; -1 1 1];
m = m';
```
matrix with two 3-dimensional patterns
each pattern fills one column of the matrix

```
net = newhop(m);
w = net.LW{1,1}
b = net.b{1,1}
```
create the network and calculate its weights
display weights of the network
display thresholds

```
[y,pf,af] = sim(net, 2, {}, m)
```
simulation of the network work for remembered patterns on the input (the second parameter of `sim` function means the number of patterns, variables `pf` and `af` are negligible)

```
y'
```
display the result

```
a = {[0.9; 0.1; 0.8]};
[y,pf,af] = sim(net, {1 20}, {}, a);
```
test the input different from remembered pattern
simulation of the network work for the pattern `a` on the input (the second parameter of `sim` function means the number of iterations – here 20)

```
y1 = cell2mat(y)
```
display the result

```
plot3(m(1,:), m(2,:), m(3,:), '*');
hold on
plot3(y1(1,:), y1(2,:), y1(3,:));
hold off
```
plot points corresponding to patterns
plot the trajectory of state changing from the input pattern to the most similar remembered pattern

To do:

1. Create 10 character patterns (letters, numbers or other characters). Each pattern shold be created as matrix of the size minimum 4 x 4 pixels. Each matrix should be filled with values 1 (black pixel) or -1 (white pixel).
2. Convert characters to vectors and create one matrix with patterns in its columns.
3. Create Hopfield network remembering patterns.
4. Test the network work for remembered patterns and for deformed patterns.

Useful commands:

`imagesc(A)` – to display a character
`V = reshape(A,n,m)` – to covert a character to a vector and vice versa.

The report should include

1. The set of created character patterns.
2. MATLAB code.
3. Results of the network work for remembered patterns on the input.
4. Results of the network work for deformed patterns on the input.