# RECOGNISING DIGITS

## Table of Contents



# Initial setup, loads patterns

```
clear, clc
x1=[ [ -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, 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],[ -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, -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],[ -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, -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],[ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1] ];

x2=[ [ -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, 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],[ -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, 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],[ -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, 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],[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1] ];

x3=[ [ 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, 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],[ -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, 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],[ 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, -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],[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1] ];

x4=[ [ -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, -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],[ -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, 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],[ -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,
 -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],[ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1] ];
```

```
x5=[ [ -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, -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],[ -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, 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],[ -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, -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],[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1] ];

stored_patterns = [ x1', x2', x3', x4', x5' ];


fed_pattern_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, -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, 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, 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, -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, -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, -1], [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1]]';

fed_pattern_2 = [[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, -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],
 [-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, 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], [-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, -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], [-1, -1, 1, 1, 1, 1, 1, 1,
 -1, -1]]';

fed_pattern_3 = [[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, -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],
 [-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, 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], [-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, -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], [-1, -1, -1, -1, -1, -1,
 -1, 1, 1, -1]]';
```

# Calculate and classify steady states of patterns

```
clc
weight_matrix = generate_weight_matrix_zero_diag(stored_patterns);

steady_state_pattern_1 =
 update_pattern_until_steady_state(weight_matrix, fed_pattern_1);
```

```
col_index_pattern_1 = classify_pattern(steady_state_pattern_1,
 stored_patterns);
disp(strcat("The steady state of pattern 1 corresponds to digit index
 ", int2str(col_index_pattern_1), "."))

steady_state_pattern_2 =
 update_pattern_until_steady_state(weight_matrix, fed_pattern_2);
col_index_pattern_2 = classify_pattern(steady_state_pattern_2,
 stored_patterns);
disp(strcat("The steady state of pattern 2 corresponds to digit index
 ", int2str(col_index_pattern_2), "."))

steady_state_pattern_3 =
 update_pattern_until_steady_state(weight_matrix, fed_pattern_3);
col_index_pattern_3 = classify_pattern(steady_state_pattern_3,
 stored_patterns);
disp(strcat("The steady state of pattern 3 corresponds to digit index
 ", int2str(col_index_pattern_3), "."))
```

```
The steady state of pattern 1 corresponds to digit index 2.
The steady state of pattern 2 corresponds to digit index 4.
The steady state of pattern 3 corresponds to digit index 5.
```

# Functions

```
disp('')
function col_index = classify_pattern(pattern, stored_patterns)
    for i = 1:size(stored_patterns, 1)
        if isequal(pattern, stored_patterns(:,i))
            col_index = i;
            break
        elseif isequal(-pattern, stored_patterns(:,i))
            col_index = -i;
            break
        end
    end
end

function steady_state_pattern =
 update_pattern_until_steady_state(weight_matrix, fed_pattern)
    old_pattern = fed_pattern;
    new_pattern = update_state_asynchronously(weight_matrix,
 old_pattern);
    while ~isequal(old_pattern, new_pattern)
       old_pattern = new_pattern;
       new_pattern = update_state_asynchronously(weight_matrix,
 old_pattern);
    end
    steady_state_pattern = new_pattern;
end

function state = update_state_asynchronously(weight_matrix, state)
    for neuron_number = 1:size(state, 1)
```

```
        state(neuron_number) =
 signum(weight_matrix(neuron_number,:)*state);
    end
end

function weight_matrix =
 generate_weight_matrix_zero_diag(stored_patterns)
    weight_matrix = stored_patterns*stored_patterns'/
size(stored_patterns,1);
    weight_matrix = weight_matrix - diag(diag(weight_matrix));
end

function sign = signum(x)
    sign = 2*(x >= 0) - 1;
end
```

*Published with MATLAB® R2020a*