

```

close all
clear
clc
%% rule 4
pattern1=zeros(17,17);
pattern2=zeros(17,17);
pattern3=zeros(17,17);
pattern4=zeros(17,17);
%pattern 1
for i = 1:17
    pattern1(i,i) = 1;
    pattern1(i,17+1-i) = 1;
end
%pattern2
for i = 1:(17);
    pattern2(i,8) = 1;           % Top left to center
    pattern2(8,i) = 1;           % Double the thickness
end

% pattern3
for j = 1:17

    pattern3(j,5) = 1;
    pattern3(j,12) = 1;

    pattern3(8,5:12) = 1;
end

% pattern 4
for i = 2:17-1
    pattern4(i,2) = 1;           % Top horizontal line
    pattern4(i,17-1) = 1;       % Bottom horizontal line
end
for j = 2:17-1
    pattern4(2,j) = 1;           % Left vertical line
    pattern4(17-1,j) = 1;       % Right vertical line
end
% Combine all patterns into a single array
all_patterns = zeros(17,17,4);
all_patterns(:,:,1) = pattern1;
all_patterns(:,:,2) = pattern2;
all_patterns(:,:,3) = pattern3;
all_patterns(:,:,4) = pattern4;
f1=figure
% Plot the patterns
for i = 1:4
    subplot(2,2,i);
    imshow(all_patterns(:,:,i));
    title(sprintf('Pattern %d', i));
end

```

```

saveas(f1, sprintf('1.png'));
dt = 0.001;           % time step for simulation
tau = 0.010;          % time constant for cells
tmax = 1;             % maximum time to wait
time = 0:dt:tmax;      %time vector
N_t = length(time);
N_unit=289;
r_max=50;
I_threshold=10;
del_I=1;
tau_r=10e-3;
pattern=zeros(17,17);
W=ones(N_unit,N_unit)*(-0.3/N_unit);
figure
for trial = 1:400+4
    r_i = zeros(N_t,N_unit);
    pattern_no = randi(4);
    input_rand = all_patterns(:, :, pattern_no);
    prob = find(rand(N_unit,1) < 0.1 );
    input_rand(prob) = 1-input_rand(prob);
    for i = 2:N_t
        if ( i < N_t/ 2 )

            I_i = input_rand(:)'*50 + r_i(i-1,:)*W;
        else
            I_i= r_i(i-1,:)*W;
        end

        r_i(i,:) = r_i(i-1,:) + dt/tau_r*((r_max./(1+exp(-(I_i-
I_threshold)/del_I)))-r_i(i-1,:));
    end
    pattern(:) = r_i(end,:);

    rate_t = 25;
    epsilonp = 0.1/N_unit;
    epsilononn = 0.25*epsilonp;

    dW = epsilonp*(double(r_i'>rate_t))*(double(r_i>rate_t)) -
epsilononn*(double(r_i'<rate_t))*(double(r_i>rate_t));
    W = W+dW*dt;
    W = min(W,8/N_unit);
    W = max(W,-8/N_unit);
    if ( mod(trial,100) < 5)
        figure(pattern_no)
        subplot(2,1,1)
        imagesc(input_rand)           % Input to network
        subplot(2,1,2)
        imagesc(pattern)             % Response at end of trial
        drawnow
        caxis([0 r_max])
    end
end

for trial = 1:4
    rate = zeros(N_t,N_unit);

```

```

pattern_no = trial;

% Now set the chosen pattern to be the current trial's input pattern
input_rand = all_patterns(:, :, pattern_no);
prob = find(rand(N_unit, 1) < 0.1);
input_rand(prob) = 1 - input_rand(prob)

f44 = figure(44)

subplot('Position', [0.03 + 0.25 * (trial - 1) 0.52 0.2 0.35])
imagesc(input_rand); % View input patterns
set(gca, 'XTick', [])
set(gca, 'YTick', [])
title(strcat(['Final Input ', ' ', num2str(pattern_no)]))
colormap(gray)
for i = 2:N_t
    if (i < N_t / 2)

        I_i = input_rand(:)' * 50 + r_i(i-1, :) * W;
    else
        I_i = r_i(i-1, :) * W;
    end

    r_i(i, :) = r_i(i-1, :) + dt / tau_r * ((r_max ./ (1 + exp(-(I_i -
I_threshold) / del_I))) - r_i(i-1, :));
end
pattern(:) = r_i(end, :);

% Finally plot all data on one figure

subplot('Position', [0.03 + 0.25 * (trial - 1) 0.04 0.2 0.35])
imagesc(pattern);
set(gca, 'XTick', [])
set(gca, 'YTick', [])
title(strcat(['Final Response ', ' ', num2str(pattern_no)]))
colormap(gray)
end
saveas(f44, sprintf('44.png'));
%% rule 2
%clf
pattern1 = zeros(17, 17);
pattern2 = zeros(17, 17);
pattern3 = zeros(17, 17);
pattern4 = zeros(17, 17);
%pattern 1
for i = 1:17
    pattern1(i, i) = 1;
    pattern1(i, 17 + 1 - i) = 1;
end
%pattern2
for i = 1:(17);
    pattern2(i, 8) = 1; % Top left to center
    pattern2(8, i) = 1; % Double the thickness

end

```

```

% pattern3
for j = 1:17

    pattern3(j,5) = 1;
    pattern3(j,12) = 1;

    pattern3(8,5:12) = 1;

end

% pattern 4
for i = 2:17-1
    pattern4(i,2) = 1;           % Top horizontal line
    pattern4(i,17-1) = 1;       % Bottom horizontal line
end
for j = 2:17-1
    pattern4(2,j) = 1;           % Left vertical line
    pattern4(17-1,j) = 1;       % Right vertical line
end
% Combine all patterns into a single array
all_patterns = zeros(17,17,4);
all_patterns(:, :, 1) = pattern1;
all_patterns(:, :, 2) = pattern2;
all_patterns(:, :, 3) = pattern3;
all_patterns(:, :, 4) = pattern4;
f11=figure
% Plot the patterns
for i = 1:4
    subplot(2,2,i);
    imshow(all_patterns(:, :, i));
    title(sprintf('Pattern %d', i));
end

dt = 0.001;           % time step for simulation
tau = 0.010;          % time constant for cells
tmax = 1;             % maximum time to wait
time = 0:dt:tmax;     %time vector
N_t = length(time);
N_unit=289;
r_max=50;
I_threshold=10;
del_I=1;
tau_r=10e-3;
pattern=zeros(17,17);
W=ones(N_unit,N_unit)*(-0.3/N_unit);
figure
for trial = 1:400+4
    r_i = zeros(N_t,N_unit);
    pattern_no = randi(4);
    input_rand = all_patterns(:, :, pattern_no);
    prob = find(rand(N_unit,1) < 0.1 );

```

```

input_rand(prob) = 1-input_rand(prob);
for i = 2:N_t
    if ( i <N_t/ 2 )

        I_i = input_rand(:)'*50 + r_i(i-1,:)*W;
    else
        I_i= r_i(i-1,:)*W;
    end

    r_i(i,:) = r_i(i-1,:) + dt/tau_r*((r_max./(1+exp(-(I_i-
I_threshold)/del_I)))-r_i(i-1,:));
end
pattern(:) = r_i(end,:);

rate_t = 25;
epsilonp = 0.1/N_unit;
epsilononn = 0.15*epsilonp;

dW = epsilonp*(double(r_i'>rate_t))*(double(r_i>rate_t)) -2*
epsilononn*(double(r_i'<rate_t))*(double(r_i>rate_t));
W = W+dW*dt;
W = min(W,8/N_unit);
W = max(W,-8/N_unit);
if ( mod(trial,100) < 5)
    figure(pattern_no)
    subplot(2,1,1)
    imagesc(input_rand)                % Input to network
    subplot(2,1,2)
    imagesc(pattern)                  % Response at end of trial
    drawnow
    caxis([0 r_max])
end
end

for trial = 1:4
    rate = zeros(N_t,N_unit);
    pattern_no = trial;

    % Now set the chosen pattern to be the current trial's input pattern
    input_rand = all_patterns(:, :,pattern_no);
    prob = find(rand(N_unit,1) < 0.1 );
    input_rand(prob) = 1-input_rand(prob)

    f22=figure(22)

    subplot('Position',[0.03+0.25*(trial-1) 0.52 0.2 0.35])
    imagesc(input_rand);                % View input patterns
    set(gca,'XTick',[])
    set(gca,'YTick',[])
    title(strcat(['Final Input ', ' ', num2str(pattern_no)]))
    colormap(gray)
    for i = 2:N_t
        if ( i <N_t/ 2 )

```

```

        I_i = input_rand(:)'*50 + r_i(i-1,:)*W;
    else
        I_i= r_i(i-1,:)*W;
    end

    r_i(i,:) = r_i(i-1,:) + dt/tau_r*((r_max./(1+exp(-(I_i-
I_threshold)/del_I)))-r_i(i-1,:));
    end
    pattern(:) = r_i(end,:);

    % Finally plot all data on one figure
    figure(22)
    subplot('Position',[0.03+0.25*(trial-1) 0.04 0.2 0.35])
    imagesc(pattern);
    set(gca,'XTick',[])
    set(gca,'YTick',[])
    title(strcat(['Final Response ', ' ', num2str(pattern_no)]))
    colormap(gray)
end
saveas(f22, sprintf('22.png'));

%% rule 3
pattern1=zeros(17,17);
pattern2=zeros(17,17);
pattern3=zeros(17,17);
pattern4=zeros(17,17);
%pattern 1
for i = 1:17
    pattern1(i,i) = 1;
    pattern1(i,17+1-i) = 1;
end
%pattern2
for i = 1:(17);
    pattern2(i,8) = 1;
    pattern2(8,i) = 1;
    % Top left to center
    % Double the thickness

end

% pattern3
for j = 1:17

    pattern3(j,5) = 1;
    pattern3(j,12) = 1;

    pattern3(8,5:12) = 1;

end

% pattern 4
for i = 2:17-1
    pattern4(i,2) = 1;

```

```

    pattern4(i,17-1) = 1;
end
for j = 2:17-1
    pattern4(2,j) = 1;
    pattern4(17-1,j) = 1;
end
% Combine all patterns into a single array
all_patterns = zeros(17,17,4);
all_patterns(:,:,1) = pattern1;
all_patterns(:,:,2) = pattern2;
all_patterns(:,:,3) = pattern3;
all_patterns(:,:,4) = pattern4;
f1=figure
% Plot the patterns
for i = 1:4
    subplot(2,2,i);
    imshow(all_patterns(:,:,i));
    title(sprintf('Pattern %d', i));
end
saveas(f1, sprintf('1.png'));
dt = 0.001;           % time step for simulation
tau = 0.010;          % time constant for cells
tmax = 1;             % maximum time to wait
time = 0:dt:tmax;      %time vector
N_t = length(time);
N_unit=289;
r_max=50;
I_threshold=10;
del_I=1;
tau_r=10e-3;
pattern=zeros(17,17);
W=ones(N_unit,N_unit)*(-0.3/N_unit);
figure
for trial = 1:400+4
    r_i = zeros(N_t,N_unit);
    pattern_no = randi(4);
    input_rand = all_patterns(:,:,pattern_no);
    prob = find(rand(N_unit,1) < 0.1 );
    input_rand(prob) = 1-input_rand(prob);
    for i = 2:N_t
        if ( i < N_t/ 2 )

            I_i = input_rand(:)'*50 + r_i(i-1,:)*W;
        else
            I_i= r_i(i-1,:)*W;
        end

        r_i(i,:) = r_i(i-1,:) + dt/tau_r*((r_max./(1+exp(-(I_i-
I_threshold)/del_I)))-r_i(i-1,:));
    end
    pattern(:) = r_i(end,:);

    rate_t = 25;
    epsilonp = 0.1/N_unit;
    epsilononn = 0.4*epsilonp;

```

```

        dW = epsilonp*(double(r_i'>rate_t))*(double(r_i>rate_t)) -
epsilon_n*(double(r_i'<rate_t))*(double(r_i>rate_t));
        W = W+dW*dt;
        W = min(W,8/N_unit);
        W = max(W,-8/N_unit);
        if ( mod(trial,100) < 5)
            figure(pattern_no)
            subplot(2,1,1)
            imagesc(input_rand)           % Input to network
            subplot(2,1,2)
            imagesc(pattern)           % Response at end of trial
            drawnow
            caxis([0 r_max])
        end
    end

for trial = 1:4
    rate = zeros(N_t,N_unit);
    pattern_no = trial;

    % Now set the chosen pattern to be the current trial's input pattern
    input_rand = all_patterns(:, :, pattern_no);
    prob = find(rand(N_unit,1) < 0.1 );
    input_rand(prob) = 1-input_rand(prob)

    f33=figure(33)

    subplot('Position',[0.03+0.25*(trial-1) 0.52 0.2 0.35])
    imagesc(input_rand);           % View input patterns
    set(gca,'XTick',[])
    set(gca,'YTick',[])
    title(strcat(['Final Input ', ' ', num2str(pattern_no)]))
    colormap(gray)
    for i = 2:N_t
        if ( i < N_t/ 2 )

            I_i = input_rand(:)'*50 + r_i(i-1,:)*W;
        else
            I_i= r_i(i-1,:)*W;
        end

        r_i(i,:) = r_i(i-1,:) + dt/tau_r*((r_max./(1+exp(-(I_i-
I_threshold)/del_I)))-r_i(i-1,:));
    end
    pattern(:) = r_i(end,:);

    % Finally plot all data on one figure

    subplot('Position',[0.03+0.25*(trial-1) 0.04 0.2 0.35])
    imagesc(pattern);
    set(gca,'XTick',[])
    set(gca,'YTick',[])
    title(strcat(['Final Response ', ' ', num2str(pattern_no)]))
    colormap(gray)

```



```

end
saveas(f33, sprintf('33.png'));

%% rule 1
pattern1=zeros(17,17);
pattern2=zeros(17,17);
pattern3=zeros(17,17);
pattern4=zeros(17,17);
%pattern 1
for i = 1:17
    pattern1(i,i) = 1;
    pattern1(i,17+1-i) = 1;
end
%pattern2
for i = 1:(17);
    pattern2(i,8) = 1;
    pattern2(8,i) = 1;
    % Top left to center
    % Double the thickness

end

% pattern3
for j = 1:17

    pattern3(j,5) = 1;
    pattern3(j,12) = 1;

    pattern3(8,5:12) = 1;

end

% pattern 4
for i = 2:17-1
    pattern4(i,2) = 1;
    pattern4(i,17-1) = 1;
    % Top horizontal line
    % Bottom horizontal line
end
for j = 2:17-1
    pattern4(2,j) = 1;
    pattern4(17-1,j) = 1;
    % Left vertical line
    % Right vertical line
end
% Combine all patterns into a single array
all_patterns = zeros(17,17,4);
all_patterns(:,:,1) = pattern1;
all_patterns(:,:,2) = pattern2;
all_patterns(:,:,3) = pattern3;
all_patterns(:,:,4) = pattern4;
f1=figure
% Plot the patterns
for i = 1:4
    subplot(2,2,i);
    imshow(all_patterns(:,:,i));
    title(sprintf('Pattern %d', i));
end

```

```

saveas(f1, sprintf('1.png'));
dt = 0.001;           % time step for simulation
tau = 0.010;          % time constant for cells
tmax = 1;             % maximum time to wait
time = 0:dt:tmax;      %time vector
N_t = length(time);
N_unit=289;
r_max=50;
I_threshold=10;
del_I=1;
tau_r=10e-3;
pattern=zeros(17,17);
W=ones(N_unit,N_unit)*(-0.3/N_unit);
figure
for trial = 1:400+4
    r_i = zeros(N_t,N_unit);
    pattern_no = randi(4);
    input_rand = all_patterns(:, :, pattern_no);
    prob = find(rand(N_unit,1) < 0.1 );
    input_rand(prob) = 1-input_rand(prob);
    for i = 2:N_t
        if ( i < N_t/ 2 )

            I_i = input_rand(:)'*50 + r_i(i-1,:)*W;
        else
            I_i= r_i(i-1,:)*W;
        end

        r_i(i,:) = r_i(i-1,:) + dt/tau_r*((r_max./(1+exp(-(I_i-
I_threshold)/del_I)))-r_i(i-1,:));
    end
    pattern(:) = r_i(end,:);

    rate_t = 25;
    epsilonp = 0.1/N_unit;
    epsilonon = 0.0;

    dW = epsilonp*(double(r_i'>rate_t))*(double(r_i>rate_t)) ;
    W = W+dW*dt;
    W = min(W,8/N_unit);
    W = max(W,-8/N_unit);
    W = W - ones(N_unit,1)*mean(W)-0.3/N_unit;
    if ( mod(trial,100) < 5)
        figure(pattern_no)
        subplot(2,1,1)
        imagesc(input_rand)           % Input to network
        subplot(2,1,2)
        imagesc(pattern)             % Response at end of trial
        drawnow
        caxis([0 r_max])
    end
end

for trial = 1:4
    rate = zeros(N_t,N_unit);

```

```

pattern_no = trial;

% Now set the chosen pattern to be the current trial's input pattern
input_rand = all_patterns(:, :, pattern_no);
prob = find(rand(N_unit, 1) < 0.1);
input_rand(prob) = 1 - input_rand(prob)

f11 = figure(11)

subplot('Position', [0.03 + 0.25 * (trial - 1) 0.52 0.2 0.35])
imagesc(input_rand); % View input patterns
set(gca, 'XTick', [])
set(gca, 'YTick', [])
title(strcat(['Final Input ', ' ', num2str(pattern_no)]))
colormap(gray)
for i = 2:N_t
    if (i < N_t / 2)

        I_i = input_rand(:)' * 50 + r_i(i-1, :) * W;
    else
        I_i = r_i(i-1, :) * W;
    end

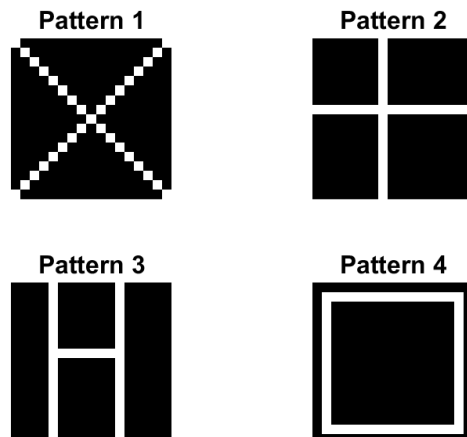
    r_i(i, :) = r_i(i-1, :) + dt / tau_r * ((r_max ./ (1 + exp(-(I_i -
I_threshold) / del_I))) - r_i(i-1, :));
end
pattern(:) = r_i(end, :);

% Finally plot all data on one figure

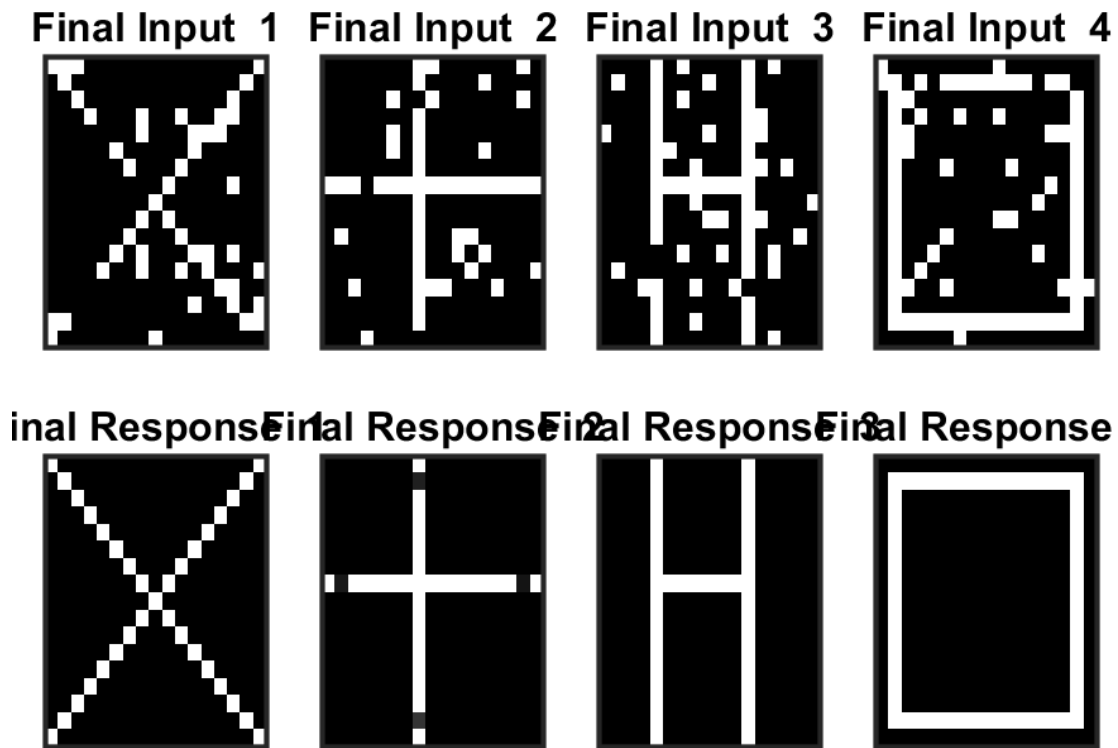
subplot('Position', [0.03 + 0.25 * (trial - 1) 0.04 0.2 0.35])
imagesc(pattern);
set(gca, 'XTick', [])
set(gca, 'YTick', [])
title(strcat(['Final Response ', ' ', num2str(pattern_no)]))
colormap(gray)
end
saveas(f11, sprintf('11.png'));

```

The patterns I used are as follows:

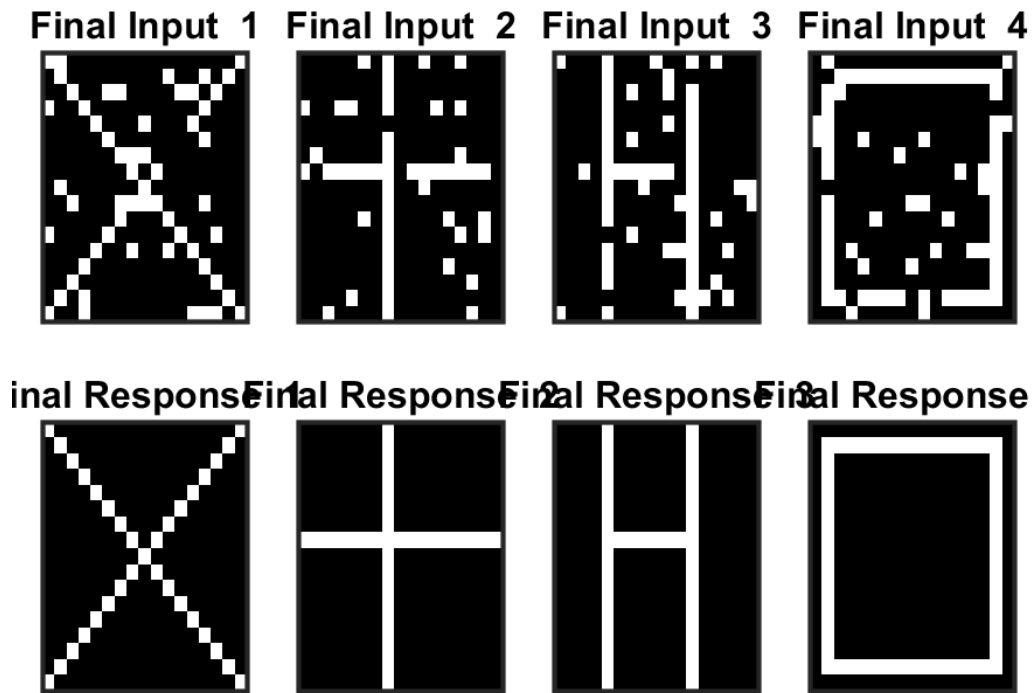


For rule 1 the final result is:



Frequent successful recognition of these patterns likely triggered LTP. Repeated pre-synaptic firing corresponding to these patterns, along with post-synaptic neurons firing above their thresholds, would have led to increased calcium influx and LTP, strengthen the connections for these specific patterns. The presented figure with a cross with empty spots suggests that some neurons representing the complete cross might not fire at all (because there's no corresponding lit pixel to stimulate them) or not provide sufficient excitatory input. This lack of co-activation with connected neurons doesn't trigger potentiation (since it relies on co-firing), and the existing connections might remain unchanged. However, the rule doesn't explicitly incorporate LTD in the classical sense.

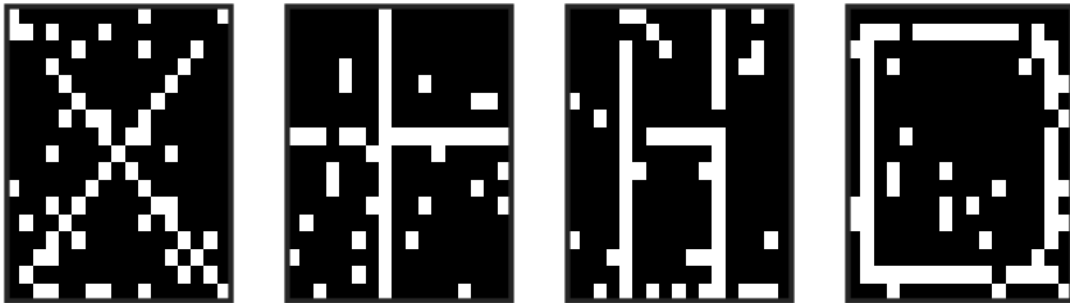
For rule 2 the final result is:



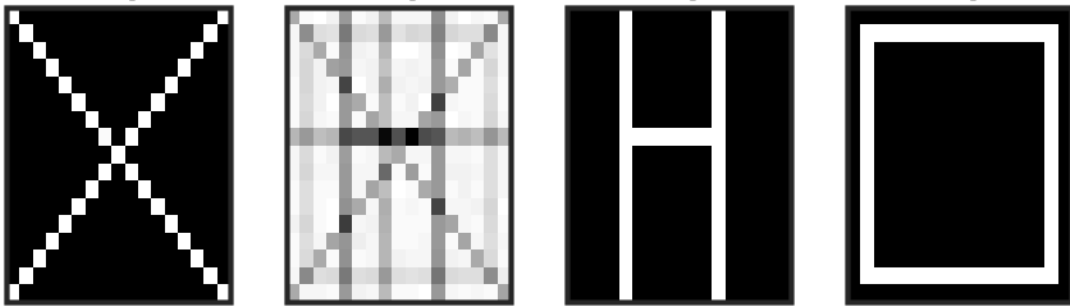
This indicates well-balanced LTP and LTD across all patterns. Encounters with each pattern likely triggered LTP for its specific recognition pathway, while encountering non-matching patterns didn't trigger enough stimulation for LTP and might have even induced slight LTD, keeping irrelevant pathways weak. This balance between strengthening the correct connections (LTP) and weakening irrelevant ones (LTD) allows for accurate recognition of all patterns.

For rule 3 the final result is:

Final Input 1 Final Input 2 Final Input 3 Final Input 4



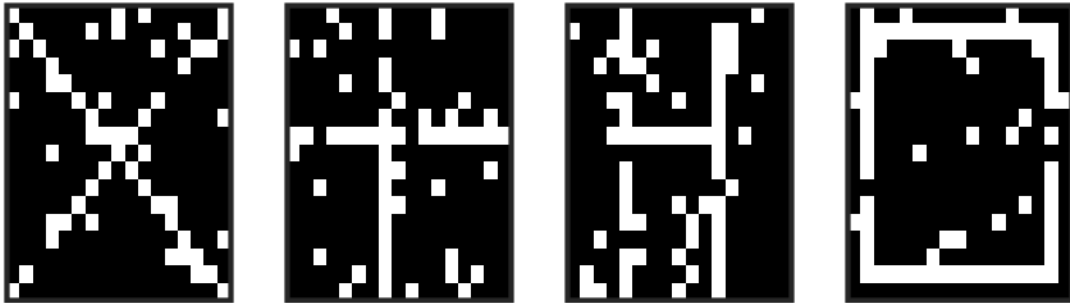
Final Response 1 Final Response 2 Final Response 3 Final Response



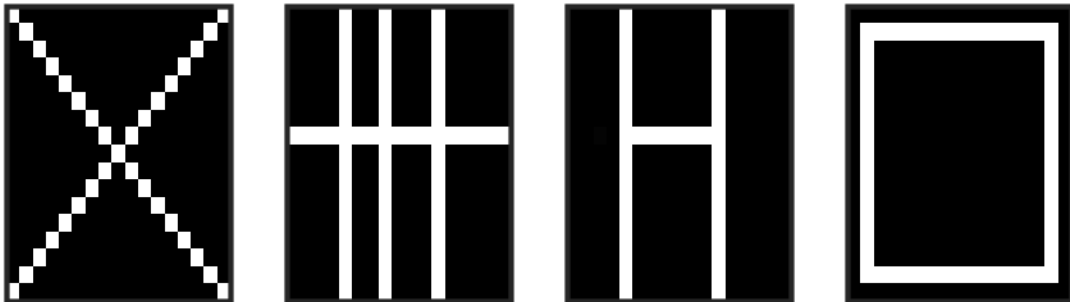
Correct recognition of X, H, and Square suggests strong synaptic connections between the input patterns and the correct output neurons for X, H, and Square. Frequent exposure to these patterns with successful recognition likely triggered LTP, strengthening these connections. Misrecognition of Cross (confusion with all patterns) suggests a more significant LTD effect on the complete cross recognition pathway. I think repeated encounters with incomplete crosses have triggered excessive LTD. The presynaptic neurons stimulated by the incomplete pattern might not reach their firing threshold, or the incomplete pattern do not provide enough overall excitatory input for the post-synaptic neurons to reach their threshold. This could lead to a significant weakening of the connections for a full cross, causing confusion with other patterns that have some overlapping features with the incomplete cross.

For rule 4 the final result is:

Final Input 1 Final Input 2 Final Input 3 Final Input 4



Final Response 1 Final Response 2 Final Response 3 Final Response



Correct recognition of X, H, and Square suggests strong synaptic connections between the input patterns and the correct output neurons for X, H, and Square. Frequent exposure to these patterns with successful recognition likely triggered LTP, strengthening these connections. The misrecognition of cross (Confused with H) is because the incomplete cross have some shared features with the H pattern (H horizontal line). These shared features could unintentionally stimulate the pre-synaptic neurons in the H recognition pathway, causing them to reach their firing threshold. Additionally, the corrupted input pattern containing these features provide enough excitatory input for the post-synaptic neurons in the H pathway to reach their threshold. This coordinated firing in the H pathway could trigger LTP, strengthening the connections for recognizing H-like patterns.