Problem E15.9

An outstar network with is given with its architecture. We are simply to train and simulate a variety of scenarios on it.

First we will initialize the network:

```
% unconditional stimuli
p_0 = cell(5, 1);
p_0{1} = [-1; -1];
p_0{2} = [1; -1];
p_0{3} = [-1; 1];
p_0{4} = [1; 1];
% this is p_00
p_0{5} = [0; 0];
% conditional stimuli
p = cell(3, 1);
p{1} = [1; 0; 0];
p{2} = [0; 1; 0];
p{3} = [0; 0; 1];
% weights and biases
W0 = eye(2,2);
W = [0 0 0; 0 0 0];
b = [-0.5; -0.5];
% learning rate
alpha = 0.6;
```

i.

We will train the network on the sequence presented in this part of the problem. First we define the training sequences:

```
training_sequence = cell(6,1);
% 1
training_sequence{1}{1} = 4;
training_sequence{1}{2} = 1;
% 2
training_sequence{2}{1} = 3;
training_sequence{2}{2} = 2;
% 3
training_sequence{3}{1} = 1;
training_sequence{3}{2} = 3;
% 4
training_sequence{4}{1} = 3;
training_sequence{4}{2} = 2;
% 5
training_sequence{5}{1} = 2;
training_sequence{5}{2} = 3;
% 6
training_sequence{6}{1} = 4;
training_sequence{6}{2} = 1;
```

And finally the training loop:

Recall that the outstar rule was as follows (vector form):

$$W_j(q) = W_j(q-1) + \alpha(a(q) - W_j(q-1))p_j(q)$$

```
for i=1:6
    a = hardlims(W0*p_0{training_sequence{i}{1}} + W*p{training_sequence{i}{2}} + b);
    W(:,training_sequence{i}{2}) = W(:,training_sequence{i}{2}) + ...
        alpha*(a - W(:,training_sequence{i}{2}))*p{training_sequence{i}{2}}(training_sequence{i
end
```

ii.

Final weights:

```
W
```

```
W = 2×3
    0.8400   -0.8400   -0.8400
    0.8400    0.8400   -0.8400
```

iii.

We will run the desired simulations:

```
% president does not push a button
p_test_president = p1;
p_0_test_president = p_0{5};
hardlims(W0*p_0_test_president + W*p_test_president + b)
```

```
ans = 2×1
     1
     1
```

```
% vice-president does not push a button
p_test_vp = p2;
p_0_test_vp = p_0{5};
hardlims(W0*p_0_test_vp + W*p_test_vp + b)
```

```
ans = 2×1
    -1
     1
```

```
% chairman does not push a button
p_test_chairman = p3;
p_0_test_chairman = p_0{5};
hardlims(W0*p_0_test_chairman + W*p_test_chairman + b)
```

```
ans = 2×1
    -1
    -1
```

iv.

As we can see from the result above, the predictions were as follows:

- 4th floor was predicted for the President.
- 3rd floor was predicted for the Vice-President.
- 1st floor was predicted for the Chairman.
-

v.

If:

- President pushes '3' repeatedly and,
- Vice-President pushes '2' repeatedly and,
- Chairman pushes '4' repeatedly.

We expect each column of the final weights to represent floors '3', '2' and '4' respectively because the network eventually learns the exact patterns it needs to recall. Concretely, we expect the final weights to be:

$$\begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix}$$