% Main Script

% Set prototypes

orange\_prototype = [1; -1; -1];

apple\_prototype = [1; 1; -1];

prototypes = [orange\_prototype, apple\_prototype];

% Validate book example

test\_fruit = [-1; -1; -1]; % Create test cases

hammingFruitClassifier = hammingNetwork(prototypes);

result = hammingFruitClassifier.classify(test\_fruit, 0.5);

classifyFruit(result)

% Compute exercice 1

% 1) Crear a W1 con S=7 y R=9, un solo valor de epsilon,

% proponga un valor de p desconocido y obtenga su resultado de

% clasificación.

fruit1\_prototype = [1; 1; -1; -1; 1; -1; 1];

fruit2\_prototype = [-1; 1; 1; -1; -1; 1; 1];

fruit3\_prototype = [1; -1; 1; -1; 1; -1; 1];

fruit4\_prototype = [1; 1; 1; -1; -1; -1; 1];

fruit5\_prototype = [-1; -1; -1; 1; 1; -1; 1];

fruit6\_prototype = [1; -1; -1; -1; 1; -1; -1];

fruit7\_prototype = [1; 1; -1; 1; -1; 1; -1];

fruit8\_prototype = [-1; 1; -1; 1; 1; 1; -1];

fruit9\_prototype = [1; -1; 1; 1; -1; -1; 1];

prototypes\_1 = [fruit1\_prototype, fruit2\_prototype, fruit3\_prototype, fruit4\_prototype, fruit5\_prototype, fruit6\_prototype, fruit7\_prototype, fruit8\_prototype, fruit9\_prototype];

test\_fruit\_1 = [-1; 1; -1; -1; 1; 1; -1]; % Create test cases

hammingFruitClassifier\_1 = hammingNetwork(prototypes\_1);

result\_1 = hammingFruitClassifier\_1.classify(test\_fruit\_1, 0);

classify\_e1(result\_1)

% Compute excercie 2

% 2) Usar el W1 anterior y repetir el experimento para 3 valores

% de epsilon diferentes al anterior y el mismo valor de p del ejercicio anterior.

result\_2\_1 = hammingFruitClassifier\_1.classify(test\_fruit\_1, 0.1);

result\_2\_2 = hammingFruitClassifier\_1.classify(test\_fruit\_1, 0.01);

result\_2\_3 = hammingFruitClassifier\_1.classify(test\_fruit\_1, -0.1);

classify\_e1(result\_2\_1)

classify\_e1(result\_2\_2)

classify\_e1(result\_2\_3)

% Compute ecercise 3

% 3) Crear un W1 con cualquier S, R y epsilon de tal manera que a2 se

% quede oscilando use un máximo de iteraciones para detener el programa, itmax=100.

test1\_prototype = [1; -1; -1; 1; 1];

test2\_prototype = [1; 1; -1; 1; -1];

test3\_prototype = [-1; -1; -1; 1; -1];

prototypes\_3 = [test1\_prototype, test2\_prototype, test3\_prototype];

% Validate book example

test\_p\_e3 = [-1; -1; -1; 1; 1]; % Create test cases

hammingFruitClassifier\_3 = hammingNetwork(prototypes\_3);

result\_3 = hammingFruitClassifier\_3.classify(test\_p\_e3, 0);

classify\_e1(result\_3)

% Compute ecercise 4

% 4) Crear un W1 con cualquier S, R. Dado un vector de entrada p, muestre que para

% un valor de epsilon caiga en una clase y para otro valor de epsilon cambie de clase,

% es decir que el primer valor de epsilon de un resultado de clasificación y el segundo

% valor de epsilon de un resultado diferente.

prototype\_1 = [1; -1; 1];

prototype\_2 = [1; 1; -1];

prototype\_3 = [-1; 1; 1];

prototypes\_4 = [prototype\_1, prototype\_2, prototype\_3];

test\_vector = [1; -1; -1];

hammingFruitClassifier\_4 = hammingNetwork(prototypes\_4);

result\_4\_1 = hammingFruitClassifier\_4.classify(test\_vector, 0.3);

result\_4\_2 = hammingFruitClassifier\_4.classify(test\_vector, -0.1);

classify\_e1(result\_4\_1)

classify\_e1(result\_4\_2)

Después de 10 intentos (véase el anexo), no se pudo lograr cambiar las predicciones con el valor de épsilon, mi consideración es que no hay forma de lograr esto debido a que solo puede haber un valor con distancia de Hamming que arroje un resultado, en cambio, se pudo lograr lo solicitado en el ejercicio 3, ya que en caso de que esta distancia sea igual en dos prototipos el algoritmo estará oscilando indefinidamente.

# ANEXO

Los valores de epsilon sobre los cuales se intentó lograr la predicción distinta del ejercicio 4 son:

1. e1 = -0.9, e2 = -0.8
2. e1 = -0.7, e2 = -0.6
3. e1 = -0.5, e2 = -0.4
4. e1 = -0.3, e2 = -0.2
5. e1 = -0.1, e2 = 0
6. e1 = -0.85, e2 = -0.75
7. e1 = -0.65, e2 = -0.55
8. e1 = -0.45, e2 = -0.35
9. e1 = -0.25, e2 = -0.15
10. e1 = -0.2, e2 = -0.6

% Make interpretations over results rom the book

function classifyFruit(prediction)

switch(find(prediction))

case 1

disp('The fruit is an Orange')

case 2

disp('The fruit is an Apple')

otherwise

disp('Could not classify')

end

end

% Make interpretations over results from Exercices

function classify\_e1(prediction)

if isempty(prediction)

disp('Could not classify')

else

switch(find(prediction))

case 1

disp('The fruit 1')

case 2

disp('The fruit 2')

case 3

disp('The fruit 3')

case 4

disp('The fruit 4')

case 5

disp('The fruit 5')

case 6

disp('The fruit 6')

case 7

disp('The fruit 7')

case 8

disp('The fruit 8')

case 9

disp('The fruit 9')

end

end

end