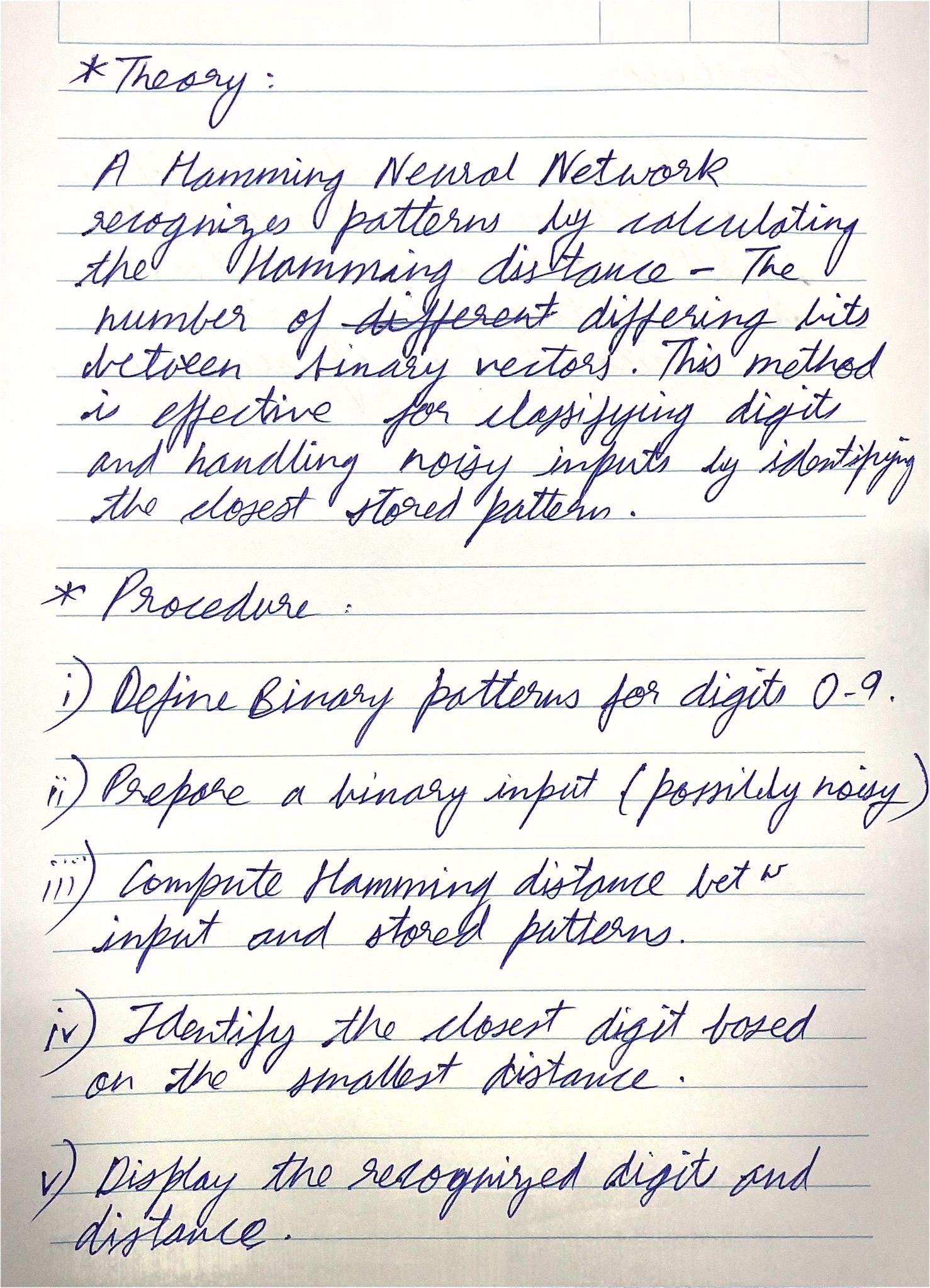
Assignment 7

**Experiment 13:**

**Aim:** Implement a Program for Pattern storage of 10 digits with Hamming Neural Network.

**Objective:** Students will be able to understand practical aspects of Hamming Neural Network model



**Code:**

% Hamming Neural Model

% Binary representations of digits (simplified example)

patterns = [

0 0 0 0 0; % Digit 0 (binary pattern)

0 0 0 0 1; % Digit 1 (binary pattern)

0 0 0 1 0; % Digit 2 (binary pattern)

0 0 0 1 1; % Digit 3 (binary pattern)

0 0 1 0 0; % Digit 4 (binary pattern)

0 0 1 0 1; % Digit 5 (binary pattern)

0 0 1 1 0; % Digit 6 (binary pattern)

0 0 1 1 1; % Digit 7 (binary pattern)

0 1 0 0 0; % Digit 8 (binary pattern)

0 1 0 0 1; % Digit 9 (binary pattern)

];

% Input pattern (example: input for which you want to find the closest match)

input\_pattern = [0 0 0 1 0]; % Input could be a noisy version of digit 2

% Initialize variables

min\_hamming\_distance = inf; % Start with a large number

closest\_digit = -1;

% Compute Hamming distance for each stored pattern

for digit = 1:10

% Calculate Hamming distance (number of differing bits)

hamming\_distance = sum(input\_pattern ~= patterns(digit, :));

% If the current pattern is closer, update the closest digit

if hamming\_distance < min\_hamming\_distance

min\_hamming\_distance = hamming\_distance;

closest\_digit = digit - 1; % Because digit index is 1-based

end

End

% Output the recognized digit

disp("Recognized Digit:")

disp(closest\_digit)

disp("Hamming Distance:")

disp(min\_hamming\_distance)

**Output:**

