Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Industriales

Máster en Automática y Robótica

Applied Artificial Intelligence

Assignment 5.1: 2D Classification with MLP

Josep María Barberá Civera (17048)

April 9, 2024

ASSIGNMENT 5.1

2D Classification with MLP

For solving the classification problem defined in following picture (Fig. 1) we use three neurons in the hidden layer, each of them defines a straight line in the 2D input space.

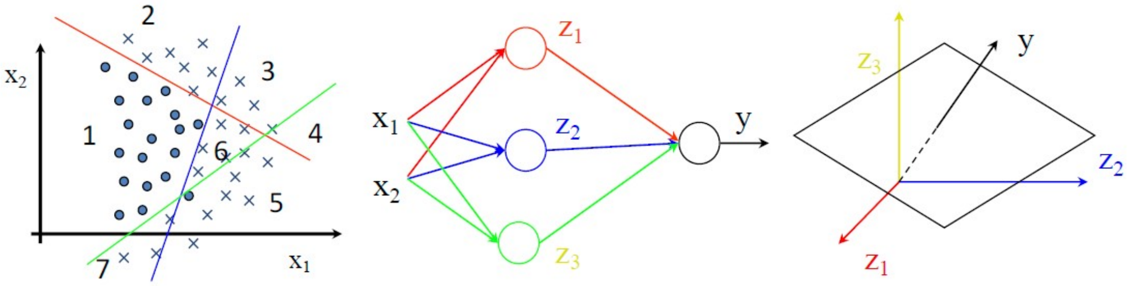

Figure 1: *Classification problem, neural network structure and Z space*

1.1 Exercise 1

Write at both sides of each straight line the output value of the associated neuron for inputs laying at each side (i.e. zero or one):

Figure 2 shows the output value of each neuron in different colours. The output value chosen is 1 when classifying `circles` and 0 when classifying `crosses`.

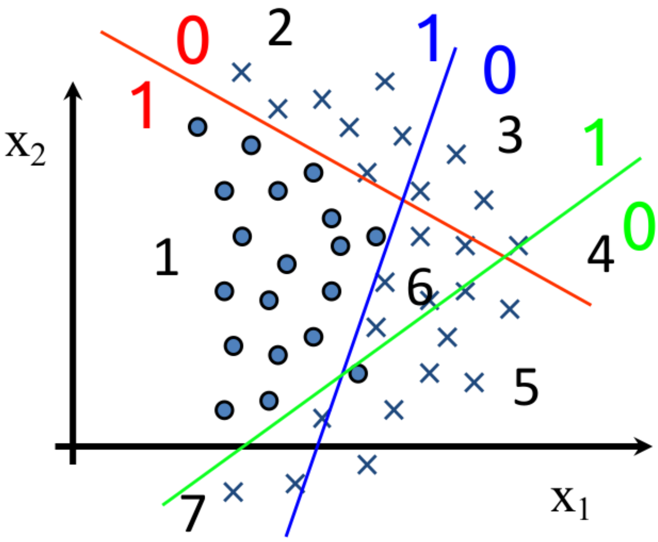

Figure 2: *Classification problem with the straight lines corresponding to each neuron and the output value associated to each of them.*

1.2 Exercise 2

For each numerated zone (from 1 to 7) write the output of the three neurons:

Figure 3 shows the resulting classification zones of the three neurons. On the right, the values of each neuron have been assigned to each region.

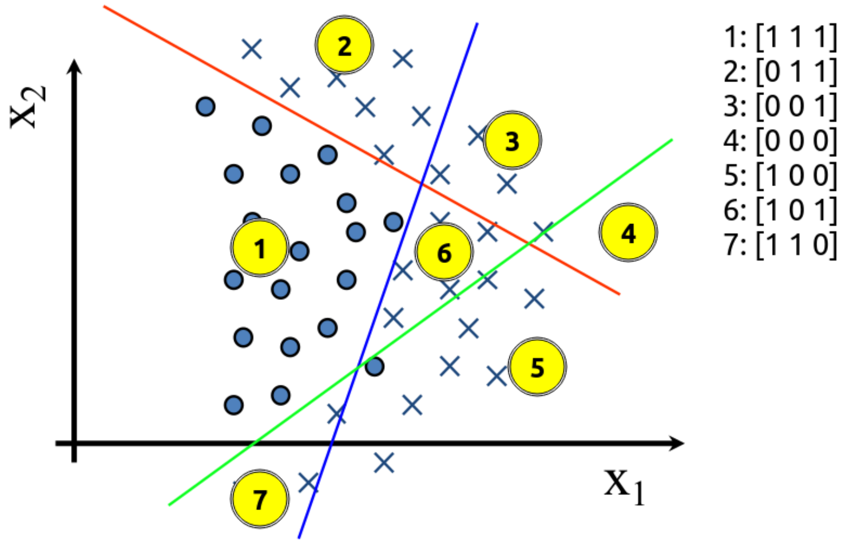

Figure 3: *Classification zones and values of each neuron.*

1.3 Exercise 3

Plot in `the Z space` all possible three values calculated on previous section, writing the number of the associated zone. Also write the desired final classification output for each zone (i.e. zero or one) in brackets beside the zone number:

Figure 4 shows the 7 corners of the unit cube resulting from assigning the three possible values of the neurons for each zone. Next to each vertex are included:

$$\text{(the coordinates)}, \quad \text{[associated zone : desired final classification]}$$

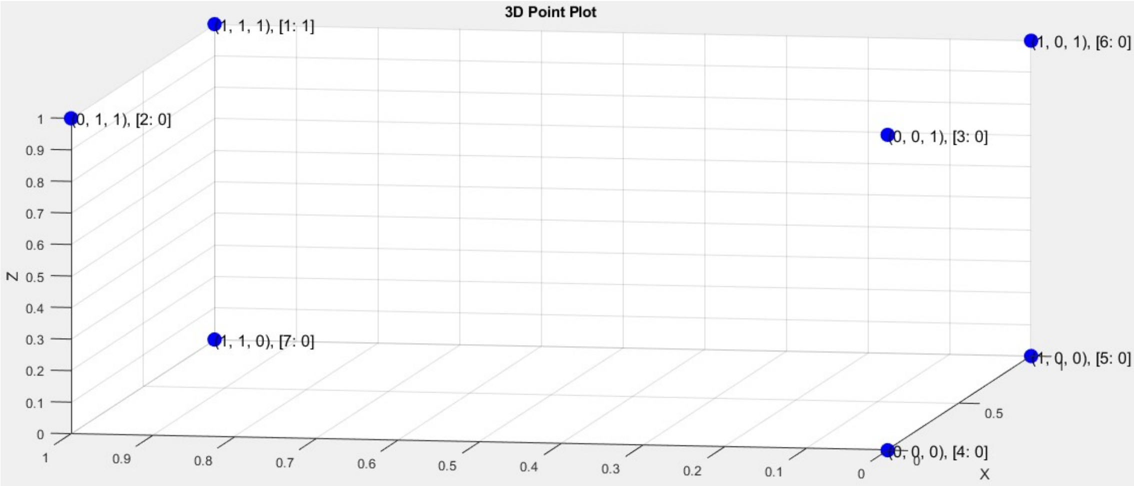

Figure 4: *The seven vertices of the unit cube emerge from assigning the three potential values of neurons to each zone.*

1.4 Exercise 4

Plot in `the Z space` the plane that should define the output neuron to solve the classification problem (i.e. providing the desired one or zero for points in each zone):

The classification plane shall be perpendicular to the vector joining the origin with the correct classification zone, in this case the point (1, 1, 1). The boundary points are unknown as we do not have the affine functions of each neuron, so an approximate point (1, 1, 0.4), has been taken in order to obtain the equation of a possible plane. Figure 5 shows the points associated with each zone, (1, 1, 1) being the only one with an output value of `1`, and the plane that allows classification.

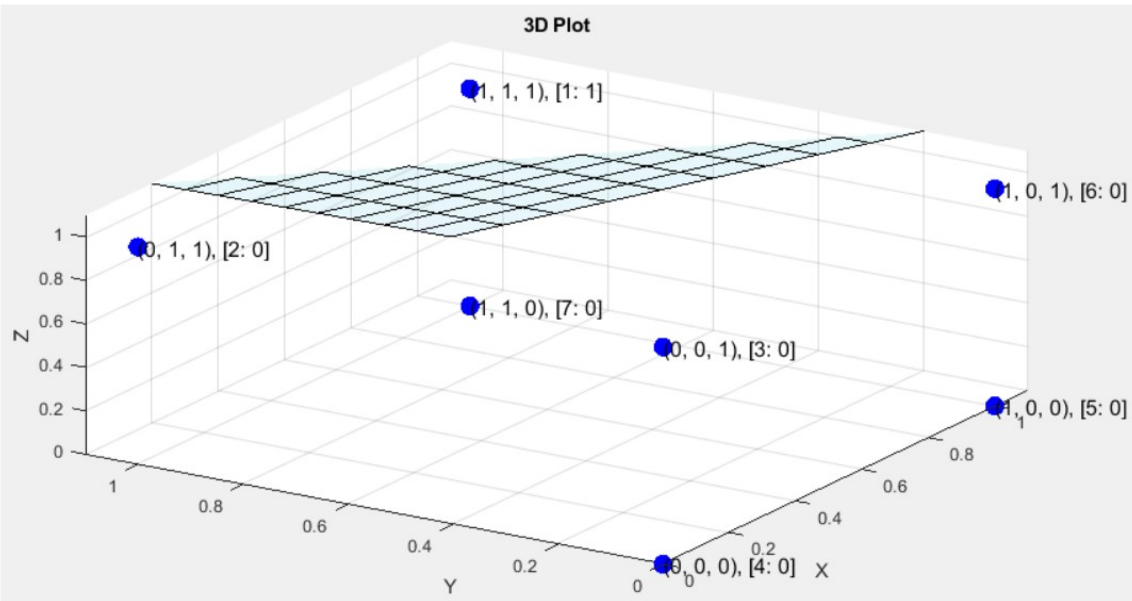

Figure 5: *Points corresponding to each zone, where (1, 1, 1) uniquely represents a NN output value of 1, along with the classification plane.*

The equation of the plane is as follows:

$$x + y + z = 2.4$$

1.5 Relevant Code

The code prepared for this assignment is shown in the next pages in a wider style.

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                           Master in Robotics
%                    Applied Artificial Intelligence
%
% Assinment 5.1:  Classification Error
% Student: Josep Barbera Civera
% ID: 17048
% Date: 09/04/2024
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 3. Plot in the Z space all possible three values calculated on
%   previous section, writing the number of the associated zone.
%   Also write the desired final classification output for each
%   zone (i.e. zero or one) in brackets beside the zone number.
% 4. Plot in the Z space the plane that should define the output
%   neuron to solve the classification problem (i.e. providing
%   the desired one or zero for points in each zone)

%% Define 3D points from NN outputs
x1 = [1, 0, 0, 0, 1, 1, 1];
y1 = [1, 1, 0, 0, 0, 0, 1];
z1 = [1, 1, 1, 0, 0, 1, 0];
zone = [1, 2, 3, 4, 5 ,6 ,7];
class = [1, 0, 0, 0, 0, 0, 0];

%% Plot the points
figure
plot3(x1, y1, z1, 'bo', 'MarkerSize', 10, 'MarkerFaceColor', 'b');
xlabel('X');
ylabel('Y');
zlabel('Z');
title('3D Plot');

grid on;

for i = 1:numel(x1)
    text(x1(i), y1(i), z1(i), sprintf('(%d, %d, %d), [%d: %d]', ...
         x1(i), y1(i), z1(i), zone(i), class(i)), 'FontSize', 12);
end

%% Define the point on the plane and the normal vector
point = [1, 1, 0.4];
normal_vector = [1, 1, 1];
% Normalize the normal vector
normal_vector = normal_vector / norm(normal_vector);

% Find the constant D
D = -dot(normal_vector, point);

% Define a grid of points
[x, y] = meshgrid(-5:0.1:5, -5:0.1:5);
```

```

% Evaluate the plane equation for each point in the grid
z = (-normal_vector(1) * x - normal_vector(2) * y - D) / normal_vector
    (3);

% Plot the plane
hold on;
surf(x, y, z, 'FaceAlpha', 0.1);

% Set the limits for the axes
xlim([0, 1.1]);
ylim([0, 1.1]);
zlim([0, 1.1]);
```