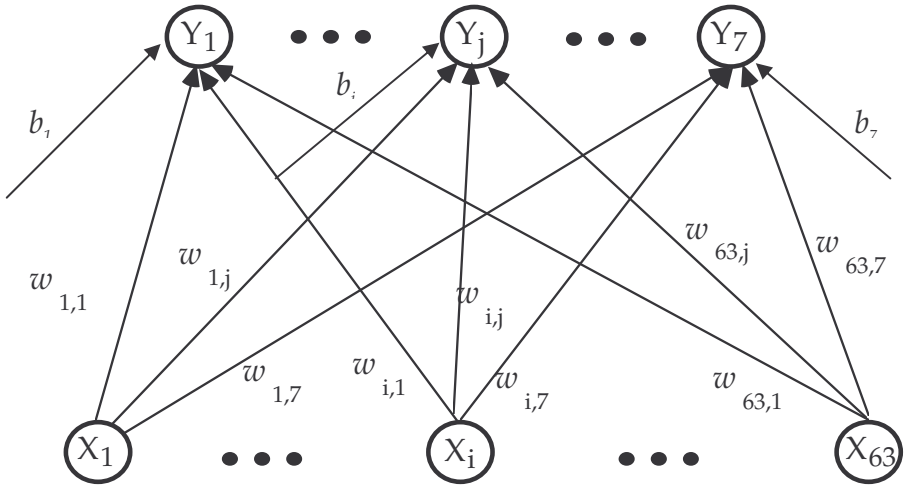
Course 3BA2 - ML Assignment ('05)

Single Layer Perceptron for OCR

(marked: Due on Friday 18th March at 5.00pm**)**

You may work and this project in pairs and submissions should be emailed to Padraig.Cunningham@cs.tcd.ie . A submission should contain the code and an account (1-2 pages of the performance of the code on the data).

Write a C or Java program that will implement a single layer perceptron to learn the patterns described on the next page. There are 7 patterns to be recognised and three different examples of each pattern. It will be easier to get a perceptron to learn these patterns if they are encoded as bipolar (i.e.) output pattern for B is (-1,1,-1,-1,-1,-1,-1). Test the network on the three noisy patterns shown on page 3. The network and algorithm could be as follows:



Step 0. Initialise weights and biases.
(small random values)

Step 1. While stopping condition is false , do Steps 1-6.

Step 2. For each bipolar training pair **s : t**, do Steps 3-5.

Step 3. Set activation of each input unit, $i = 1, \ldots n$:

$$x_i = s_i$$

Step 4. Compute activation of each output unit, $j = 1, \ldots, m$:

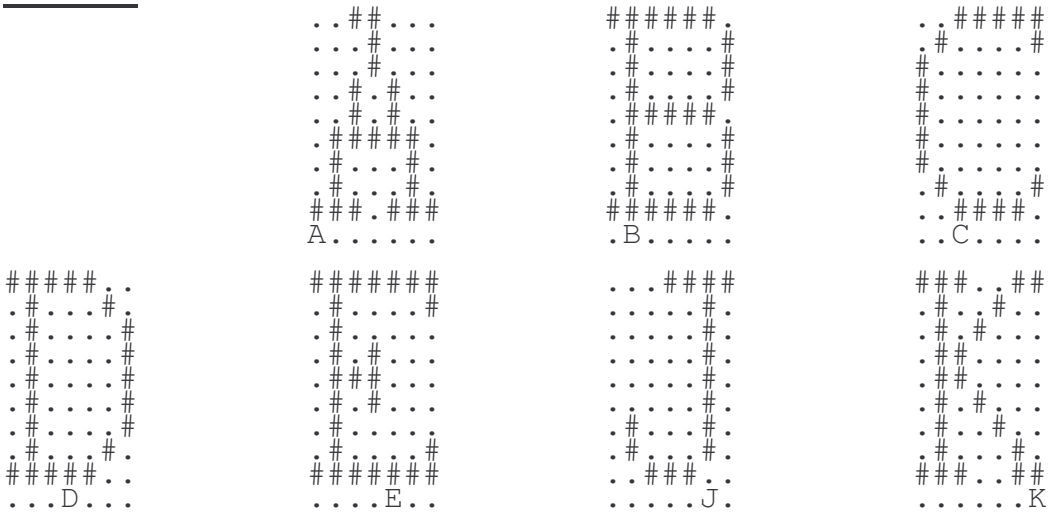
$$y\_in_j = b_j + \sum_i x_i w_{ij}$$

$$y_j = \begin{cases} 1 & \text{if } y\_in_j > \theta \\ 0 & \text{if } -\theta \le y\_in_j \le \theta \\ -1 & \text{if } y\_in_j < -\theta \end{cases}$$

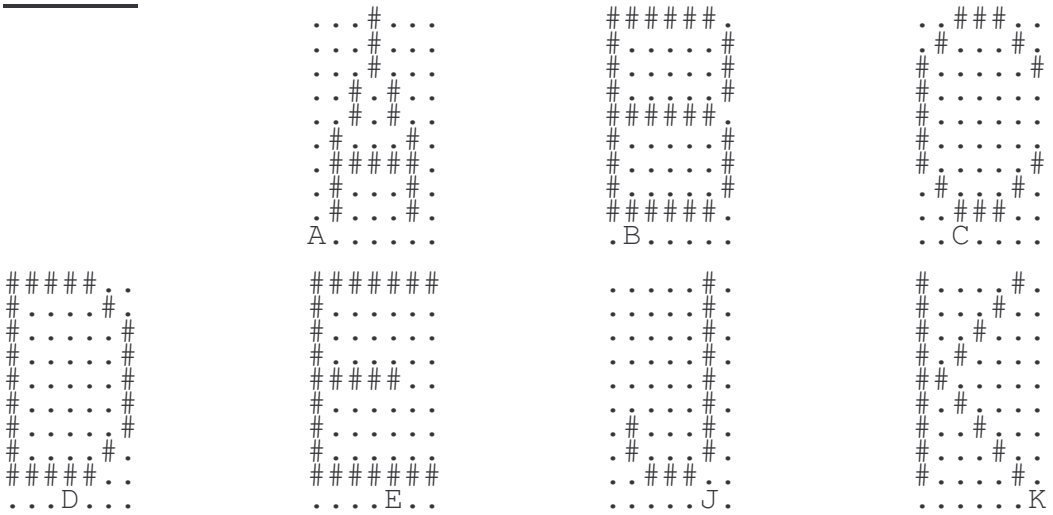
Step 5. Update biases and weights, $j = 1, \ldots, m$;
$i = 1, \ldots, n$:
If $t_j \neq y_j$, then
$b_j(\text{new}) = b_j(\text{old}) + t_j$;
$w_{ij}(\text{new}) = w_{ij}(\text{old}) + t_j x_i$.
Else biases and weights remain unchanged.

Step 6. Test for stopping condition:
If no weight changes occurred in Step 2, stop; otherwise continue.

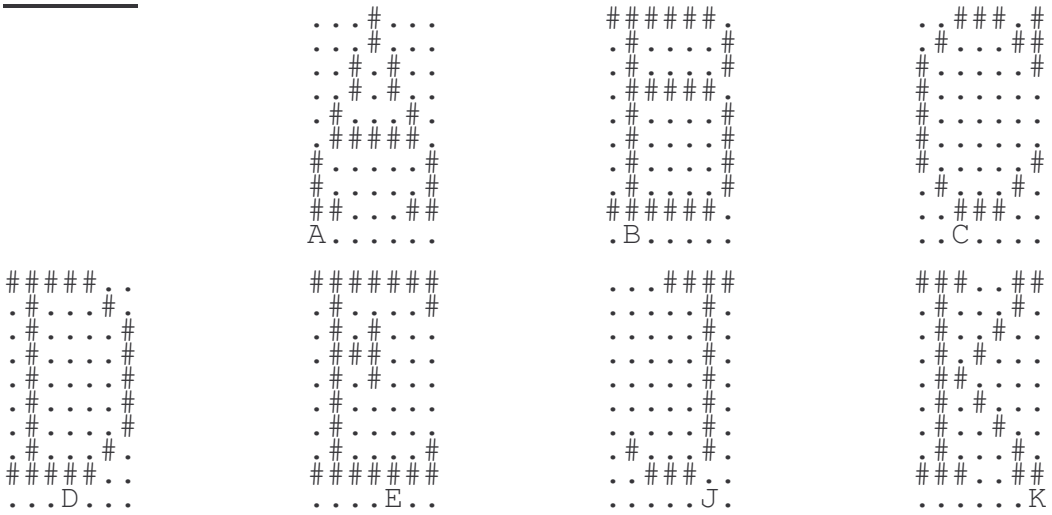
Font 1



Font 2



Font 3



Test Patterns