1.1 Introduction to MATLAB

This section of your assignment/practical activities does not have to be reported.

MATLAB is a widely adopted package which integrates computation, data analysis, visualization, and programming in one environment. It is available on many PCs and Unix/Linux workstations around the campus. You can also purchase a MATLAB Student Version, in particular, from http://www.mathworks.com/academia/student_version/

We will be using MATLAB as a computational platform in our practicals/assignments. In particular, we will be using a collection of specialised functions useful for our subject known as the Neural Network Toolbox.

MATLAB can work as an on-line interpreter, accept scripts and dedicated C-programs and allows you to build and execute a block-diagram of your computational algorithms (SIMULINK). AN elegant GUI editor is also included in MATLAB graphics.

Practical activities:

- Set up correctly your X-Windows environment and invoke MATLAB by typing in /usr/local/bin/matlab
- At the MATLAB prompt >> execute first intro and demo commands.
- After that at the MATLAB prompt >> type in the following commands and study the results. Make sure that you understand them.

```matlab
diary you.d % where you = <your_user_name>
echo on
% Part 1
help colon
a = 1:10
help +
ah’
b = a*a’
c = a.*a
d = a’*a
rank(d)
e = d([1:3, 6,8], :)
size(e)
f = [e sum(e)’ ; sum(e) sum(diag(d))]
whos
t = 5*pi*(0:100)’/100 ;
z = exp(-t/10).*exp(j*t) ;
```
help j
size(z)
plot(z), grid on, title('my first plot')

% Part 2
x = 5*pi*(-40:40)/40 ;
size(x)
find(x==0)
x(find(x == 0)) = 1e-3 ;
x(41)
y = sin(x)./x ;
plot(x, y, '-r', x, y, '*b'), grid on, xlabel('angle x')
x1 = -2:2
x2 = (-3:3)'
x1([1 1 1], :) 
x2(:, [1 1 1 1])
help meshgrid
wns = ones(1, max(size(x))) ;
x1 = x ;  % a row vector
x2 = x' ;  % a column vector
r = sqrt(x1(wns, :).^2 + x2(:, wns).^2) ;
X3 = sin(r)./r ;
size(X3)
surf(x1, x2, X3), grid on, title('3D surface')

• The result of the surf command, which is a 3-D surface, is presented in Figure 1

• Please continue execution of selected MATLAB commands:

colormap(hsv)
help view
help print
image(1+120*(X3-min(min(X3))))
help colormap
help hsv
colormap(hot(160))
colormap(flag(160))
load clown
whos
image(X),
colormap(map)
help drawnow
diary off

• After that all results are stored in the diary file you.d
Copy you.d to you.m and edit the script file you.m keeping the command lines and removing all result lines.
• Execute the script `you.m` entering `you` at the MATLAB prompt. Debug the script if necessary.

• Prepare and execute your own short MATLAB script for plotting an interesting 3D surface.

1.2 Basic properties of typical activation functions

• Derive analytical expressions (pen-and-paper method) for derivatives $\frac{d\varphi}{dv}$ of the following activation functions:

  **Unipolar sigmoidal:**
  \[
  \varphi(v) = \frac{1}{1 + e^{-\beta v}}
  \]

  **Hyperbolic tangent:**
  \[
  \varphi(v) = \tanh(\beta v)
  \]

  Compare it with the sigmoidal function.

  **Gaussian:**
  \[
  \varphi(v) = \exp\left(-\frac{v^2}{2\sigma^2}\right)
  \]
Rational:

\[ \varphi(v) = \frac{v}{\beta + |v|} \]

- Plot the derivatives using MATLAB.

1.3 Perceptron

1. Study the perceptron learning law as demonstrated in the following MATLAB M-files:

   $CSE5301/Mtib/perc.m$

   and

   $NNET/nndemos/demop1.m, demop4.m, demop5.m, demop6.m$

   where

   $CSE5301 = \text{http://www.csse.monash.edu.au/~app/CSE5301}$

   $NNET = \text{MATLAB/toolbox/nnet}$

2. Write a MATLAB M-file, similar to perc.m and the other demo files, which implements the perceptron learning law:

   - use 4-D augmented input patterns \( (p = 4) \)
   - plot the projections of input patterns and separation planes in all three 2-D projection planes, \( x_1 - x_2, x_2 - x_3, x_3 - x_1 \). Use the subplot command. Plot also in the above figure the paths of the tip of the normalised weight vector during training.
   - Use the modified learning rule as described in the lecture notes.

   Clearly indicate sections of code that you wrote yourself.

3. Implement a non-trivial logic function of your choice of three variables: \( y = f(x_1, x_2, x_3) \) using a single perceptron. Give a boolean expression describing your function. Derive the weight vector using two methods:

   - by working out the equation of the decision plane,
   - by training using the perceptron learning law.

   Remember that a logic function of three variables may be visualised by a 3-D cube, each vertex being labeled with the value of function. You must have a logic function which values are linearly separable. This is equivalent to bisecting the cube with a plane.

Your submission should include:

- Brief comments regarding the demo files,
- Relevant derivations, equations, scripts and plots with suitable comments.