University of Glasgow | School of Computing Science

CSC1103 Laboratory/Tutorial 4-5: Control structure and Functions

1. In probability and statistics, the normal (gaussian) probability distribution is the most common distribution which found its place in Central Limit Theorem where the distribution of a population will tend to normal distribution when the size approaches infinity regardless the original distribution of the population. Furthermore, all measurement noise in nature are gaussian distributed. The normal probability density function for a random variable x is given by

$$f(x) = \frac{1}{\sqrt[2]{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \qquad -\infty < x < \infty$$

where μ and the σ are the mean value and the standard deviation of the distribution. Design the algorithm and pseudocode. Hence write a C program to generate a table of values for the above gaussian probability function where μ and the σ are equal to 1 and 2 respectively for value of x ranges from -20 to 20 in the increments of 0.5

2. The exponential function e^x is one of the functions frequently used in neural network as part of the sigmoid activation function. It can be represented by the infinite series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \frac{x^n}{n!}$$

The n^{th} term in the series can be obtained by multiplying the $(n-1)^{th}$ term by (x/n) given rise to

$$term_n = (term_{n-1})\left(\frac{x}{n}\right)$$

Design the algorithm and hence pseudocode to calculate the e^x using the above series and hence write a C program. Your C program should read the value of x and print the value of the sum and the increment to the sum for each term. Terminate the computation when the absolute value of the last term is less than 1×10^{-6}. Run your program for several values of x and compare with the C library function **exp().**

3. Using a series expansion, the value of $(x+y)^n$ can be obtained as follows

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n\times(n-1)\times(n-2)\cdots(n-k+1)}{1\times 2\cdots k}$$

By using function and recursive function methodology, design the algorithm and pseudocode and hence write a C program. The C program should read in the three variables x, y and n and return the value of the evaluation.

4. Theory of Classification is a one of the important fields in machine learning and data analytics applications such as image classification in fake news detection. For a simple binary classification such as logical AND operation of two binary variables where its truth table is given as below
where

$$y = x_1 . x_2$$

Case j	x_1	x_2	AND operation (y_j)
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

By using functions and iterative control loop structure, design the algorithm and pseudocode to implement above AND operation classification using linear formulae, $\hat{y} = \sum_{i=1}^{2} w_i\, x_i + b$ where w_i is the respective weight to be trained for the input x_i and b is the bias. Hence write a C program for the pseudocode to show the weight will be trained according to the truth table above with zero error. Hence, show the classification works by allowing the user to key in the different value of x_1 and x_2 and produce the output y.
Hint:

error in output estimation in each case j, $\mathcal{E}_j = \hat{y}_j - y_j = \left(\sum_{i=1}^{2} w_i\, x_{ji} + b\right) - y_j$
Total error in output estimation, $\mathcal{E} = \sum_{j=1}^{4} \mathcal{E}_j = \sum_{j=1}^{4} \hat{y}_j - y_j$

The weights and bias are keep updating with new value to achieve total error $\mathcal{E}$ less than error threshold tolerance which in this case is zero tolerance where $\mathcal{E}_{threshold} =$

0. This mean that $\mathcal{E} = \mathcal{E}_{threshold} = 0$. For each update or iteration t,new weight and bias for next iteration are updated as

$$w_1{}^{t+1} = w_1{}^{t} - \alpha \sum_{j=1}^{4} \mathcal{E}_j x_{j1}$$

$$w_2{}^{t+1} = w_2{}^{t} - \alpha \sum_{j=1}^{4} \mathcal{E}_j x_{j2}$$

$$b^{t+1} = b^{t} - \alpha\, \mathcal{E}$$

where α is the training speed. Set $\alpha = 0.5$ and initial weight w_1 , w_2 and b = 0