Department of Information Technology – University of the Punjab
Programming for AI – MPhil/PhD (AI) F22
Lab-06

Max Time: 2.5 hours **Date: 25-1-2023**

Topics: Problem solving, operators, basic constructs, selection, repetition

Instructions:

- Please provide your own solutions and **DO NOT COPY** the code from your colleagues or the web.
- You can discuss your problems only with the teachers.
- All tasks carry equal points.

Task 1 [20]

1. Create a 2D NumPy array with shape (3,3) filled with random integers between 0 and 9.
2. Use indexing to extract the element in the second row and third column of the array you created in question 1.
3. Use slicing to extract the subarray from the first to the second row and second to the third column of the array you created in question 1.
4. Create two 1D arrays of shape (3,) and use the dot product function to find the dot product of two 1D arrays with shape (3,).
5. Use the reshape function to reshape the array you created in question 1 to shape (9,1).
6. Use the sum function along the column axis to find the sum of all elements in the array you created in question 1.
7. Using the array you created in question 1, take the transpose of it and perform element wise multiplication of the array with its transpose.
8. Given two arrays A and B of shape (m,n) and (p,q) respectively, write a function using NumPy to concatenate these arrays in the following ways:
   a. Vertically (stacking the arrays on top of each other)
   b. Horizontally (stacking the arrays next to each other)
9. Given two arrays, A and B, with dimensions (n, m) and (p, q) respectively, write a function in Python using NumPy that concatenates the arrays along the third axis (depth) in a way that the resulting array has dimensions (n, m, p+q).

10. Given the array [6, 10, 2, 5, 11, 3, 1]. Use the numpy.where function to find the indices of elements in an array that meet a certain condition, i.e. Indices of the elements that are greater than 5.

Task 2 [14]

(Handing a csv file)

1. Use the loadtxt function to load a CSV file called "data.csv" and assign it to a variable.
2. Find the mean of all the columns in the loaded CSV data using numpy's mean function.
3. Find the maximum value of a specific column in the loaded CSV data using numpy's max function.

4. Find the standard deviation of all the columns in the loaded CSV data using numpy's std function.
5. Sort the loaded CSV data in ascending order using numpy's sort function
6. Save the loaded CSV data to a new CSV file using numpy's savetxt function.
7. Find the correlation between two specific columns in the loaded CSV data using numpy's corrcoef function.

Task 3 [6]

(Linear Algebra)

Given the matrix:

A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

1. Use the linalg.eig function to find the eigenvalues and eigenvectors of a given square matrix.
2. Use the linalg.inv function to find the inverse of a given square matrix.
3. Use the linalg.det function to find the determinant of a given square matrix.

Note: The following task is **mandatory for PhD students.** However, MPhil students can perform this task as a bonus task.

Task 4 [20]
(dot product, sigmoid)

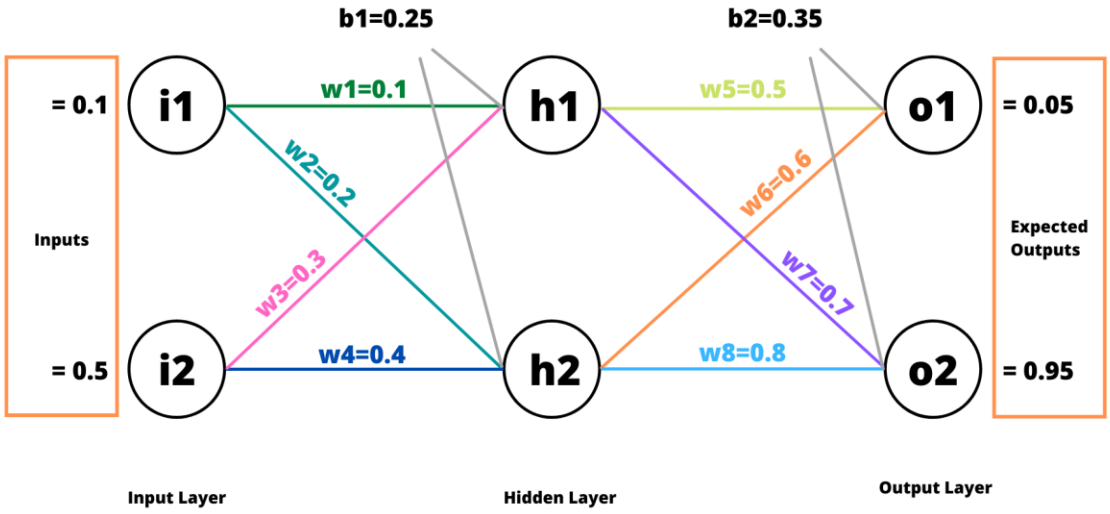
Consider the following graph with 6 nodes and 8 weighted edges. The nodes i1, and i2 are input nodes and array input contains data for these nodes. Nodes h1 and h2 are intermediate nodes and each of them receives inputs and weights from i1, i2 and corresponding connected edges.
The node **h1** computes and outputs: h1 = 1/(1+exp(-(i1*w1+i2*w3+b1))
The node **h2** computes and outputs: h2 = 1/(1+exp(-(i1*w2+i2*w4+b1))

The node **o1** computes and outputs: o1 = 1/(1+exp(-(h1*w5+h2*w6+b2))
The node o2 computes and outputs: o2 = 1/(1+exp(-(h1*w7+h2*w8+b2))

Write a program that produces the values of o1 and o2 for each of the following inputs:

i1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
i2	0.5	0.4	0.3	0.2	0.1	0.0	-0.1	-0.2	-0.3

Avoid loops as much as possible,


b1=0.25
b2=0.35
= 0.1
i1
w1=0.1
h1
w5=0.5
o1
= 0.05
Inputs
w2=0.2
w6=0.6
Expected Outputs
w3=0.3
w7=0.7
= 0.5
i2
w4=0.4
h2
w8=0.8
o2
= 0.95
Input Layer
Hidden Layer
Output Layer