Neural Networks & Deep Learning - ICP-3

CS 5720 (CRN 23216)

Student ID: 700745451

Student Name: Kamala Ramesh

1. The purpose of this program is to create a class Employee, create a data member to count the number of Employees, create a function to calculate the average salary of the employees. Create a child class for Fulltime Employee inheriting the properties from Employee Class and to call the member function through the instances created.

In [131]:

```
#create a class Employee
class Employee:
    count = 0
    totalSalary = 0
    def __init__(self, name, family, salary, department):
        self.name = name
        self.family = family
        self.salary = salary
        self.department = department
        Employee.count += 1 #to track the total count of employees
        Employee.totalSalary += self.salary #to add the salary which is to be used to calculate average income

    #function to calcualte the average income
    def averageSalary(self):
        average = Employee.totalSalary/Employee.count
        return average

    #funciton to display Employee details
    def displayEmpDetials(self):
        print("\nDetails of Employee,",self.name,":")
        print("\tNumber of Family Members:",self.family)
        print("\tSalary per month:", self.salary)
        print("\tDepartment:",self.department)

#create a child class inheriting properties from Employee class
class FulltimeEmployee(Employee):
    pass

#create objects of Employee Class and FulltimeEmployee class
e1 = Employee("Jackson Mcguire", 3, 3000, "Biology")
e2 = Employee("Mandy Williams", 4, 2500, "Economis")
ft1 = FulltimeEmployee("Kathy Coffman", 2, 2700, "Geography")
ft2 = FulltimeEmployee("Crystal Jonas", 3, 2900, "Mathematics")
```

```
#call the member funcitons using class instances
print("Total Number of Employees:",e1.count)
print("Average Salary of the Employees:",e1.averageSalary())
e1.displayEmpDetials()
e2.displayEmpDetials()
ft1.displayEmpDetials()
ft2.displayEmpDetials()
```

```
Total Number of Employees: 4
Average Salary of the Employees: 2775.0

Details of Employee, Jackson Mcguire :
        Number of Family Members: 3
        Salary per month: 3000
        Department: Biology

Details of Employee, Mandy Williams :
        Number of Family Members: 4
        Salary per month: 2500
        Department: Economis

Details of Employee, Kathy Coffman :
        Number of Family Members: 2
        Salary per month: 2700
        Department: Geography

Details of Employee, Crystal Jonas :
        Number of Family Members: 3
        Salary per month: 2900
        Department: Mathematics
```

2. This program is to create a vector of size 20 with random floating point numbers between 1 to 20, reshape it into an array of size 4 X 5 and then replace the maximum values in each row by zero.

In [116]:

```
import numpy as np

randVector = np.random.uniform(1.0,20.0,20) #random vector of size 20 with float values from 1.0 to 20.0
print("Randon vector of size 20:",randVector)

reshapedArr = randVector.reshape(4,5) #reshape into array of dimesion 4 X 5
print("\nAfter reshaping into Array of dimension 4 X 5 \n", reshapedArr)
print("\nMaximum Values in each row:\n", reshapedArr.max(axis=1).reshape(-1,1)) #print the max values from each row

#replace max values in each row by zero
modifiedArr = np.where(reshapedArr == np.amax(reshapedArr, axis=1).reshape(-1, 1), 0, x)
print("\nModified Array:\n",modifiedArr)
```

```
Randon vector of size 20: [11.97397962  4.45196162 14.42001024  2.14108671  7.95124073  8.838523
  6.34744566 16.83257405 14.00057275  6.68882835 10.91661392 15.65931403
  2.99796634 16.93739802  1.61722511 15.38433898 11.72980989  9.0893499
 14.10845714 15.50902342]

After reshaping into Array of dimension 4 X 5
 [[11.97397962  4.45196162 14.42001024  2.14108671  7.95124073]
 [ 8.838523    6.34744566 16.83257405 14.00057275  6.68882835]
 [10.91661392 15.65931403  2.99796634 16.93739802  1.61722511]
 [15.38433898 11.72980989  9.0893499  14.10845714 15.50902342]]

Maximum Values in each row:
 [[14.42001024]
 [16.83257405]
 [16.93739802]
 [15.50902342]]

Modified Array:
 [[19.26092178 11.9448386   0.          8.83602259  6.55465903]
 [16.0560136   1.27791689  0.         11.36633742 17.88928545]
 [16.3231785  17.88752262  9.04336703  0.          1.70819003]
 [16.19101628  9.23782085  7.95472002  9.75235438  0.        ]]
```