ICP-3 Neural Networks & Deep Learning

Nirmala Yarlagadda

700733102

1.Create a class Employee and then do the following • Create a data member to count the number of Employees

- Create a constructor to initialize name, family, salary, department
- Create a function to average salary
- Create a Fulltime Employee class and it should inherit the properties of Employee class
- Create the instances of Fulltime Employee class and Employee class and call their member functions

```
## Creating a constructor to initialize name, family, salary, department

    def __init__(self,name, family, salary, department):
        self.name = name
        self.family = family
        self.salary = salary
        self.department = department
        Employee.noOfEmployees += 1
        Employee.empSalary = Employee.empSalary+self.salary

## Creating a function to average salary
    def avgSalary(self):
        avgSalary = Employee.empSalary/Employee.noOfEmployees
        return(print(  "Average Salary of the Employees:",  avgSalary))

    # This method prints the Employee Details
    def displayEmpDetails(self):
        print("\n Employee Name:",self.name, "\n Family:", self.family, "\n Salary:", self.salary, "\n Department:", self.departm

## Creating a Fulltime Employee class and it should inherit the properties of Employee class
class FulltimeEmployee(Employee):
    def __init__(self,name, family, salary, department):
        Employee.__init__(self,name, family, salary, department)


a = Employee('Maya', 'Ramula', 25000, 'ME')
b = Employee('Dia', 'vanya', 55000, 'BE')
c = FulltimeEmployee('raha', 'kapoor', 200000, 'MCA')

a.displayEmpDetails()
b.displayEmpDetails()
c.displayEmpDetails()

## Display total employees
print("Total No. of Employees:", Employee.noOfEmployees)
## Calling AvgSalary method of EmployeeClass
a.avgSalary()
```

```
 Employee Name: Maya
 Family: Ramula
 Salary: 25000
 Department: ME


 Employee Name: Dia
 Family: vanya
 Salary: 55000
 Department: BE


 Employee Name: raha
 Family: kapoor
 Salary: 200000
 Department: MCA

Total No. of Employees: 3
Average Salary of the Employees: 93333.33333333333
```

2. NumPy

Using NumPy create random vector of size 20 having only float in the range 1-20.

Then reshape the array to 4 by 5

Then replace the max in each row by 0 (axis=1)

(you can NOT implement it via for loop)

```
33]: import numpy as Np

     ## Creating a random vector of size 20 having only float in the range 1-20.
     vector = Np.random.uniform(1, 20, 20)

     ## Reshape the array to a 4 by 5
     array = Np.reshape(vector, (4, 5))

     ## Replace the max in each row by 0
     array[Np.arange(4), array.argmax(axis=1)] = 0

     print(array)
```

```
[[ 3.48508764 13.4228754   0.          9.43347362 12.3210569 ]
 [ 6.47979115 14.76461031 13.06331745  8.38591067  0.        ]
 [14.33017386  9.26024599 16.74059017 14.69504615  0.        ]
 [ 6.83773433  9.19787107  8.16021381  1.68595204  0.        ]]
```

GitHub link: https://github.com/niryarjessy22/ICP-3-Neural-Networks-Deep-Learning.git

Video link: https://drive.google.com/file/d/12vMXejBudGf3lqan16K3vhDcjDY2NAw9/view?usp=share_link