Neural Networks & Deep Learning: ICP2
Name: Lalitha Sowjanya Kamuju
ID: 700747213

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.

```python
class Employee:
    # Class variable to count the number of Employees
    employee_count = 0

    def __init__(self, name, family, salary, department):
        # Instance variables
        self.name = name
        self.family = family
        self.salary = salary
        self.department = department
        # Increment the employee count when a new instance is created
        Employee.employee_count += 1

    def average_salary(self, *salaries):
        # Calculate and return the average salary
        return sum(salaries) / len(salaries)

class FulltimeEmployee(Employee):
    # Inheriting properties from the Employee class

    def __init__(self, name, family, salary, department, hours_worked):
        # Calling the constructor of the parent class
        super().__init__(name, family, salary, department)
        self.hours_worked = hours_worked

# Creating instances of Employee class
employee1 = Employee("Chid", "Family1", 50000, "Team Lead")
employee2 = Employee("Sowjanya", "Family2", 60000, "Manager")

# Creating instances of FulltimeEmployee class
fulltime_employee = FulltimeEmployee("Priya", "Family3", 70000, "CEO", 40)

# Calling member functions
average_salary = employee1.average_salary(employee1.salary, employee2.salary)
print(f"Average Salary of Employees: ${average_salary}")

print(f"Total Number of Employees: {Employee.employee_count}")

# Accessing properties of FulltimeEmployee class
print(f"{fulltime_employee.name} works in the {fulltime_employee.department} department and earns ${fulltime_employee.salary} per year.")
```

```
Average Salary of Employees: $55000.0
Total Number of Employees: 3
Priya works in the CEO department and earns $70000 per year.
```

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)

```python
import numpy as np

# Create a random vector of size 20 with float values in the range 1-20
random_vector = np.random.uniform(1, 20, 20)

# Reshape the array to a 4x5 matrix
reshaped_array = random_vector.reshape((4, 5))

# Replace the max value in each row with 0 along axis=1
reshaped_array[np.arange(len(reshaped_array)), reshaped_array.argmax(axis=1)] = 0

print("Random Vector:")
print(random_vector)
print("\nReshaped Array (4x5):")
print(reshaped_array)
```

```
Random Vector:
[ 6.76319763  4.11724778  0.          6.65285184  5.95431245  4.34313063
 15.5891919   5.29642248  0.         12.58197722  4.80076915  0.
  1.42451033  7.57321011  7.07966432 12.54780825 12.6708516   0.
 13.02029146  6.24914325]

Reshaped Array (4x5):
[[ 6.76319763  4.11724778  0.          6.65285184  5.95431245]
 [ 4.34313063 15.5891919   5.29642248  0.         12.58197722]
 [ 4.80076915  0.          1.42451033  7.57321011  7.07966432]
 [12.54780825 12.6708516   0.         13.02029146  6.24914325]]
```

GitHub Link : https://github.com/sowjanya-kamuju/Assignment3
Video Link : https://vimeo.com/906227961/bec83af3d4?share=copy