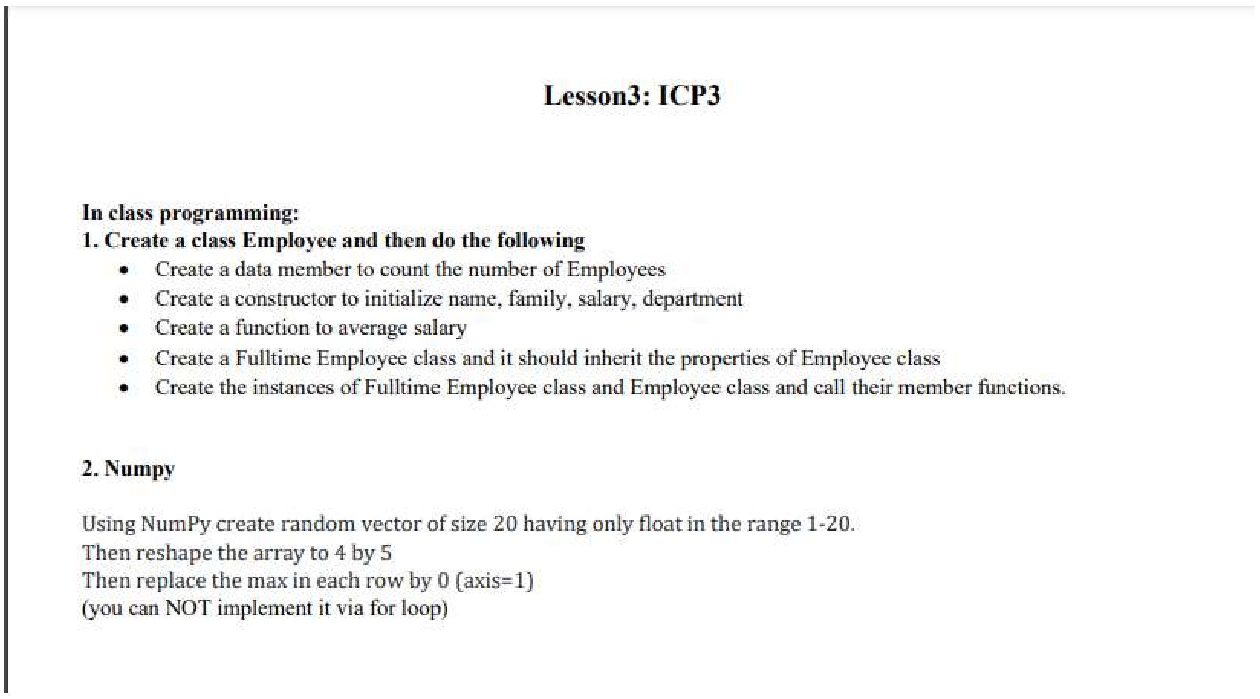
Neural Networks Assignment-3

700739769

Anjani Priya Marthati



Lesson3: ICP3

In class programming:

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.

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)

In [1]:

```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.sa
```

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

In [2]:

```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:
[ 8.99563196  0.          8.51584982  6.45581748  1.55444401 14.92572632
 16.88310569 13.33885848  0.         13.78157237  9.09861701  0.
  5.44702344 10.13114513  7.99440781 16.58719263  6.50495448  5.61822807
  0.         15.02907225]

Reshaped Array (4x5):
[[ 8.99563196  0.          8.51584982  6.45581748  1.55444401]
 [14.92572632 16.88310569 13.33885848  0.         13.78157237]
 [ 9.09861701  0.          5.44702344 10.13114513  7.99440781]
 [16.58719263  6.50495448  5.61822807  0.         15.02907225]]
```

Video Link:

https://vimeo.com/906228790/aab799bcb8?share=copy

Github Link:

https://github.com/Priyamarthati/Assignment3