

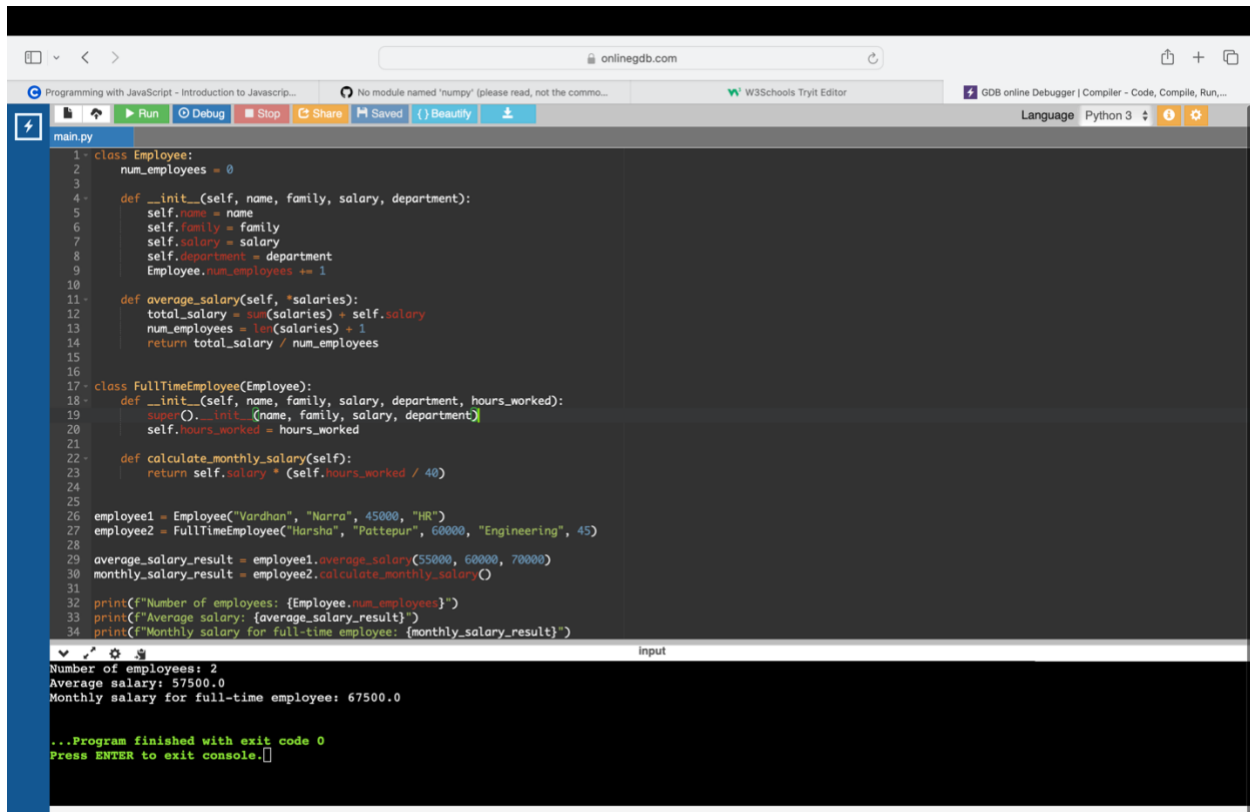
Neural Networks & Deep Learning Assignment – 3

Sai Vardhan Narra - 700756163

- GitHub Link: https://github.com/saivardhan-dev/Neural-Networks_Assignment-3_700756163-.git
- Video Link: https://drive.google.com/file/d/1WDMp0a-pow5nkOM2BfBK7uGPP3Tu3yp3/view?usp=drive_link

Question1: 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.



```
1 class Employee:
2     num_employees = 0
3
4     def __init__(self, name, family, salary, department):
5         self.name = name
6         self.family = family
7         self.salary = salary
8         self.department = department
9         Employee.num_employees += 1
10
11     def average_salary(self, *salaries):
12         total_salary = sum(salaries) + self.salary
13         num_employees = len(salaries) + 1
14         return total_salary / num_employees
15
16 class FullTimeEmployee(Employee):
17     def __init__(self, name, family, salary, department, hours_worked):
18         super().__init__(name, family, salary, department)
19         self.hours_worked = hours_worked
20
21     def calculate_monthly_salary(self):
22         return self.salary * (self.hours_worked / 40)
23
24 employee1 = Employee("Vardhan", "Narra", 45000, "HR")
25 employee2 = FullTimeEmployee("Harsha", "Pattepur", 60000, "Engineering", 45)
26
27 average_salary_result = employee1.average_salary(55000, 60000, 70000)
28 monthly_salary_result = employee2.calculate_monthly_salary()
29
30 print(f"Number of employees: {Employee.num_employees}")
31 print(f"Average salary: {average_salary_result}")
32 print(f"Monthly salary for full-time employee: {monthly_salary_result}")
```

Number of employees: 2
Average salary: 57500.0
Monthly salary for full-time employee: 67500.0

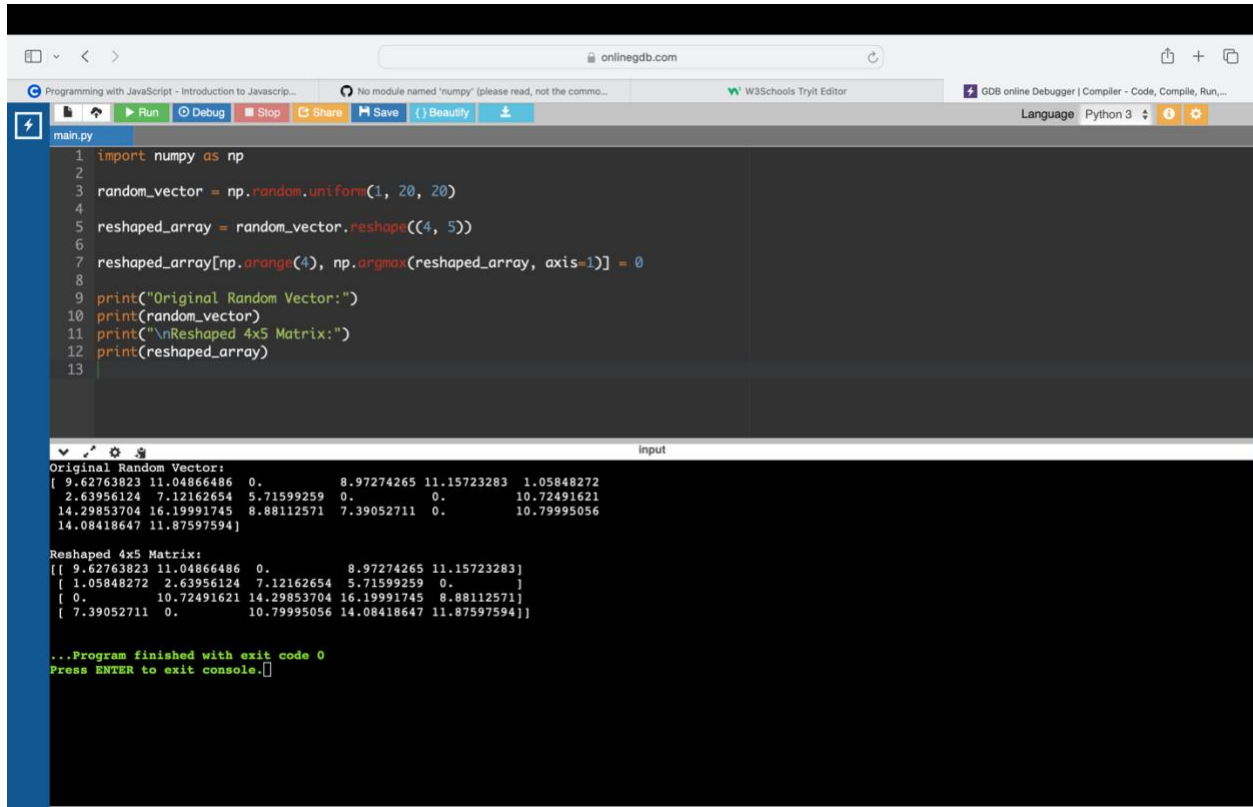
...Program finished with exit code 0
Press ENTER to exit console.

Question-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)

Neural Networks & Deep Learning Assignment – 3

Sai Vardhan Narra - 700756163



The screenshot shows an online Python IDE interface. The top bar includes a file explorer, a toolbar with icons for Run, Debug, Stop, Share, Save, and Beautify, and a language dropdown set to Python 3. The main editor area contains a Python script named `main.py` with the following code:

```
1 import numpy as np
2
3 random_vector = np.random.uniform(1, 20, 20)
4
5 reshaped_array = random_vector.reshape((4, 5))
6
7 reshaped_array[np.arange(4), np.argmax(reshaped_array, axis=1)] = 0
8
9 print("Original Random Vector:")
10 print(random_vector)
11 print("\nReshaped 4x5 Matrix:")
12 print(reshaped_array)
13
```

Below the editor, the output console displays the results of the script execution:

```
Original Random Vector:
[ 9.62763823 11.04866486  0.          8.97274265 11.15723283  1.05848272
 2.63956124  7.12162654  5.71599259  0.          0.          10.72491621
14.29853704 16.19991745  8.88112571  7.39052711  0.          10.79995056
14.08418647 11.87597594]

Reshaped 4x5 Matrix:
[[ 9.62763823 11.04866486  0.          8.97274265 11.15723283]
 [ 1.05848272  2.63956124  7.12162654  5.71599259  0.          ]
 [ 0.          10.72491621 14.29853704 16.19991745  8.88112571]
 [ 7.39052711  0.          10.79995056 14.08418647 11.87597594]]

...Program finished with exit code 0
Press ENTER to exit console.
```