

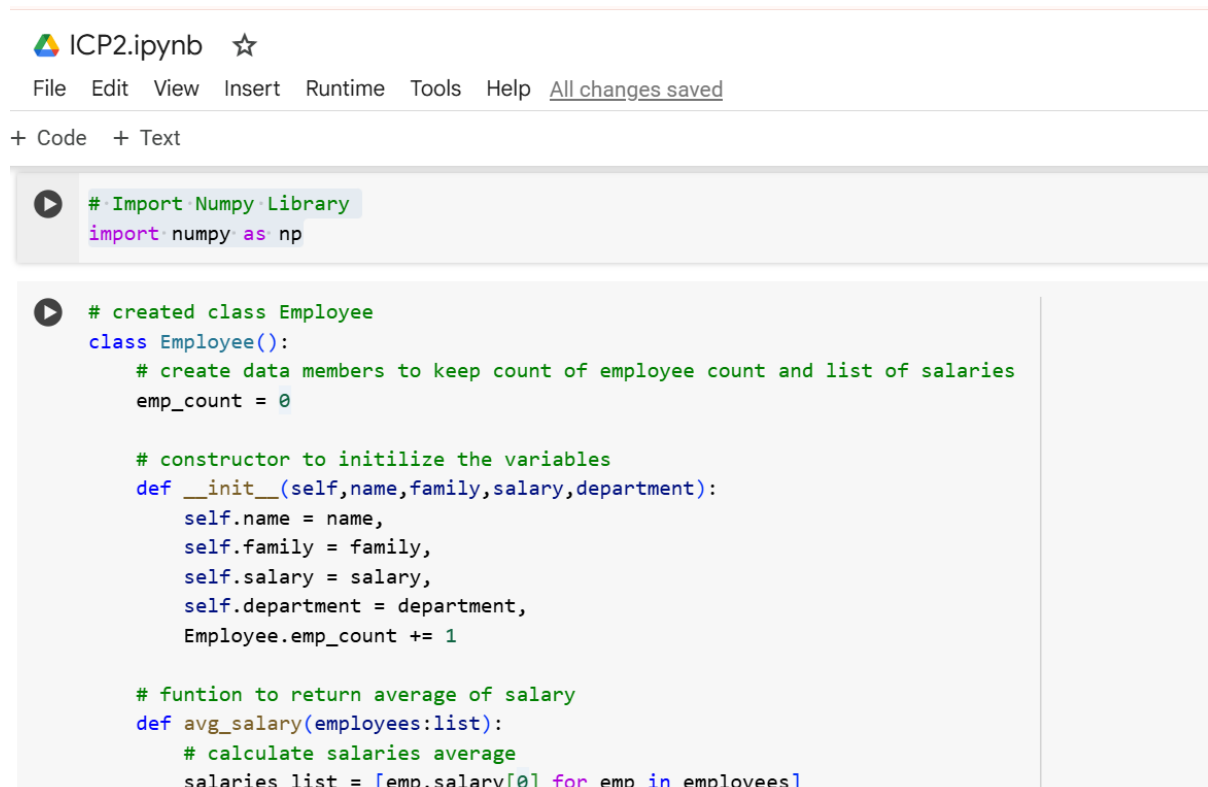
Neural Networks and Deep Learning - ICP-2

Name: Usha Kiran Reddy Patlolla

Student Id: 700764998

1Q: 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.

Description: This code defines an Employee class and a subclass called FullTimeEmployee to model employees in a company. The Employee class has a class variable emp_count that keeps track of the total number of employees. Its constructor initializes the employee's name, family, salary, and department, and increments the employee count each time a new employee object is created. The class also contains a method avg_salary that calculates the average salary from a list of employees by summing their salaries and dividing by the total number of employees. The FullTimeEmployee class inherits from the Employee class, and its constructor simply calls the parent constructor to initialize the necessary attributes. The program creates a list of employees, including both Employee and FullTimeEmployee objects, and calculates the average salary using the avg_salary method from both the Employee and FullTimeEmployee classes, printing the results.



```
ICP2.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

# Import Numpy Library
import numpy as np

# created class Employee
class Employee():
    # create data members to keep count of employee count and list of salaries
    emp_count = 0

    # constructor to initialize the variables
    def __init__(self,name,family,salary,department):
        self.name = name,
        self.family = family,
        self.salary = salary,
        self.department = department,
        Employee.emp_count += 1

    # funtion to return average of salary
    def avg_salary(employees:list):
        # calculate salaries average
        salaries_list = [emp.salary[0] for emp in employees]
```

```
    # avg_sal = np.mean(salaries_list)
    return avg_sal

# created FullTimeEmployee class and inherit the properties from Employee class
class FullTimeEmployee(Employee):
    def __init__(self, name, family, salary, department):
        # calling parent class constructor
        super().__init__(name, family, salary, department)

employees = []
employees.append(Employee("Naveen", "Indluru", 40000, "analytics"))
employees.append(Employee("Anvesh", "Kalikiri", 74000, "CTO"))

employees.append(FullTimeEmployee("Naveen Reddy", "Indluru", 50000, "data science"))
employees.append(FullTimeEmployee("Jagadeesh", "Kethu", 84000, "data analytics"))

print("using employee class", Employee.avg_salary(employees))
print("using fulltime employee class", FullTimeEmployee.avg_salary(employees))
```

using employee class 62000.0
using fulltime employee class 62000.0

2Q: 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).

Description: This code generates a random vector of size 20 with values between 1 and 20, and then reshapes it into a 4x5 array. The task is to replace the maximum element in each row with the value 1. First, the array before the replacement is printed. Then, the code calculates the maximum value of each row using `np.max` with `axis=1`, reshapes it, and replaces those maximum values in the original array with 1. Finally, the array after the replacement is printed.

```
ICP2.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved
Code + Text RAM Disk ✓

[4] arr = np.random.uniform(1,20,20) # created a random vector of size 20 in the range of 1 - 20.
    arr = arr.reshape(4,5) # reshaped the vector to array of (4,5) shape


# Task : replace the max in each row by 0 (axis = 1)

# print array before replacement
print("Array before :\n\n",arr,end="\n\n\n")

# get max elements of each row and reshape it
each_row_max = np.max(arr,axis = 1).reshape(-1,1)

arr[arr == each_row_max] = 1

# print array after replacement
print("Array after replace the max element of each row with 1: \n\n", arr)
```

 Array before :

```
[[ 7.33128224  5.56995393  9.0312551   7.45693199  8.21054117]
 [ 5.97029035 14.19808821 16.83025735  7.03973153 15.09887243]
 [ 4.63396206 15.30745409  6.34197445 13.009619   2.73947355]
 [12.35704378 18.6223202   4.73015523 14.45032106  4.39796606]]
```

Array after replace the max element of each row with 1:

```
[[ 7.33128224  5.56995393  1.          7.45693199  8.21054117]
 [ 5.97029035 14.19808821  1.          7.03973153 15.09887243]
 [ 4.63396206  1.          6.34197445 13.009619   2.73947355]
 [12.35704378  1.          4.73015523 14.45032106  4.39796606]]
```

[+ Code](#)

[+ Text](#)