Spring 2024: CS5720
Neural Networks & Deep Learning - ICP-1

Name: Katam Vamsi Krishna

Student Id: 700756973

GitHub link: https://github.com/kvamsi7/mscs/blob/mscs_nn/CS5720-Neural%20Network%20and%20Deep%20Learning/Assignments/ICP-2/ICP-2.ipynb

1. Write a program that takes two strings from the user: first_name, last_name. Pass these variables to fullname function that should return the (full name).

*** For example:**

▪ First_name = "your first name", last_name = "your last name"

▪ Full_name = "your full name"

- **Code snippet:**

```
In [1]: def get_name(first_name:str,last_name:str):
            return first_name+" "+ last_name

        first_name = input("Enter first name: ")
        last_name = input("Enter last name: ")

        full_name = get_name(first_name,last_name)

        print("Full_name =",full_name)

        Enter first name: Vamsi Krishna
        Enter last name: Katam
        Full_name = Vamsi Krishna Katam
```

*** Write function named "string_alternative" that returns every other char in the full_name string.**

Str = "Good evening"

Output: Go vnn

- **Code snippet:**

```
In [4]: def string_alternative(string:str):
            return string[::2]

        str_ = "Good evening"
        string_alternative(str_)

Out[4]: 'Go vnn'
```

Note: You need to create a function named "string_alternative" for this program and call it from

main function.

2. Write a python program to find the wordcount in a file (input.txt) for each line and then print the output.

* Finally store the output in output.txt file.

Example:

Input: a file includes two lines:

Python Course

Deep Learning Course

Output:

Python Course

Deep Learning Course

Word_Count:

Python: 1

Course: 2

Deep: 1

Learning: 1

- **Code snippet:**

In [16]:
```
def get_word_count(file_name):
    output_file = "output.txt"
    # open the output file in write mode
    with open(output_file,'w+') as opt_file_obj:
        # dictionary to store the frequencies of each
        word_freq  = {}
        # open the file in read mode
        with open(file_name) as inp_file_obj:
            # iterate over each line from the list of read lines
            for line in inp_file_obj.readlines():
                # write each line to output file
                opt_file_obj.write(line)
                # iterate over each word after splitting each line with seperator as " "(space).
                for word in line.split(" "):
                    # stripping each word to avoid any trailing characters
                    # word = word.capitalize().strip()  # to keep word in same format
                    word = word.strip()
                    # updating the frequency of each word
                    word_freq[word] = word_freq.get(word,0)+1
        opt_file_obj.write("\nWord_Count:\n")
        # iterating over each set of key and value in the frequency dict
        for key,value in word_freq.items():
            # writing each word and its frequency in output file
            opt_file_obj.write(f"{key} : {value} \n")

# run the function to get word frequency
get_word_count("input.txt")


# To read the data of output.txt
with open("output.txt") as opt:
    for line in opt.readlines():
        print(line)
```

```
Python Course

Deep Learning Course

Word_Count:

Python : 1

Course : 2

Deep : 1

Learning : 1
```

3. Write a program, which reads heights (inches.) of customers into a list and convert these heights to centimetres in a separate list using:

1) Nested Interactive loop.

2) List comprehensions

Example: L1: [150,155, 145, 148]

Output: [68.03, 70.3, 65.77, 67.13]

- **Code snippet:**

In [6]:
```
# method 1
# using nested interactive loop
def convert_heights_1(heights_in:list):
    height_cm = []
    for entry in heights_in:
        height_cm.append(entry * 2.54)
    return height_cm

# method 2
# using list comprehensions
def convert_heights_2(heights_in:list):
    height_cm = [entry * 2.54 for entry in heights_in]
    return height_cm

L1 = [150,155, 145, 148]
print("Converted using interactive loop: ",convert_heights_1(L1))
print("Converted using list comprehensions: ",convert_heights_2(L1))
```

```
Converted using interactive loop:  [381.0, 393.7, 368.3, 375.92]
Converted using list comprehensions:  [381.0, 393.7, 368.3, 375.92]
```