**Practical Assignment**

**Objective: - Predict Next Sequence**

**To start with deep learning, the very basic project that you can build is to predict the next digit in a sequence.**

**Dataset: -Create a sequence like a list of odd numbers and then build a model and train it to predict the next digit in the sequence.**

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Task: -A simple neural network with 2 layers would be sufficient to build the model.**

**Assignment Submission: - Only submit the Google Colab/Github link.(Make the Link Public).**

**import numpy as np**

***## Odd Numbers***

**odd\_numbers = np.arange(1,1000, 2)**

**odd\_numbers**

***## Every third item is set as label and preceeding two are used as input***

**trainset = list()**

**labelset = list()**

**for idx in range(len(odd\_numbers) - 2):**

**trainset.append(odd\_numbers[idx:idx+2])**

**labelset.append(odd\_numbers[idx+2])**

**trainset = np.array(trainset)**

**labelset = np.array(labelset)**

**print(trainset[:10], labelset[:10])**

**input\_size = 2**

**output\_size = 1**

**weights = 0.002 \* np.random.random((input\_size, output\_size)) - 0.001**

**num\_iterations = 100000**

**lr = 0.000001**

**for itera in range(num\_iterations):**

**error = 0**

**for idx in range(len(trainset)):**

**label = labelset[idx]**

**layer\_0 = trainset[idx]**

**output = layer\_0.dot(weights)**

**error = error + (output - label) \*\* 2**

**delta\_output = output - label**

**delta\_weights = layer\_0.reshape(input\_size,1).dot(delta\_output.reshape(1,output\_size))**

**weights = weights - lr \* delta\_weights**

**if itera % 9999 == 0:**

**print(error/len(trainset))**

**def predict\_next\_odd(input\_sequence):**

**layer\_0 = input\_sequence**

**output = layer\_0.dot(weights)**

**print("Predicted: {}".format(output))**

**predict\_next\_odd(np.array([99, 101]))**

**predict\_next\_odd(np.array([-901, -903]))**

***## Try them on even numbers***

**predict\_next\_odd(np.array([902, 904]))**

**weights**

***## Try them on even numbers***

**predict\_next\_odd(np.array([9999991, 9999993]))**

**predict\_next\_odd(np.array([501, 503]))**