[Project Code : EONN]

Classifying Near Earth Objects using Artificial Neural Networks

Project Duration: 25-Feb-2024 ~~ 16-Mar-2024
Submission Information: (via) CSE-Moodle

Objective:

There are an infinite number of objects in outer space. Some of them are closer than we think. Even though we might think that a distance of 70,000 km can not potentially harm us, but at an astronomical scale, this is a very small distance and can disrupt many natural phenomena. These objects/asteroids can thus prove to be harmful. Hence, it is wise to know what is surrounding us and what can harm us amongst those. Thus, this dataset compiles the list of NASA certified asteroids that are classified as the Near Earth Objects.

In this assignment, you will implement some simple multi-layer perceptrons and neural networks. In particular, you shall be doing the following tasks.

Tasks to be done:

1. Starter code provided does this using python, can be used: Randomly divide the dataset into 80% training set and the rest as test set. Choose the important features from the dataset by modifying relevant parts of the starter code. Choose a mini-batch size to divide the dataset into batches.
2. Build the ANN model. These operations have been demonstrated in the starter code using Pytorch.
    a. Build the MLP classifiers by identifying the number of input and output nodes required for the problem, and specifying the number of hidden layers as:
        i. 0 hidden layers
        ii. 1 hidden layer with 16 nodes
        iii. 1 hidden layer with 32 nodes
    b. Use Sigmoid or ReLU activation function for the input and hidden layers. Use ReLU activation for the output layer.
    c. Define the forward and backward operations for your network. They are required for inference and weight updation of your model.
    d. Define the training function to train the model using a forward and a backward pass. Define the prediction function for obtaining the outputs from the network.
    e. Compare the implementation of your model compared to that using the Pytorch library, on the same dataset (code snippet provided).
3. Hyper-parameter tuning.
    a. For each of the architectures, vary the learning rates in the order of 0.1, 0.01, 0.001, 0.0001, 0.00001. Plot graph for the results with respect to accuracy and loss. (Learning rate vs accuracy/loss for each model).
    b. Report test set accuracy for all the learning rates in a tabular form and identify the best model.
4. Classification Report
    a. Create a classification report for comparing the performance of your algorithm, for your best performing algorithm in terms of accuracy, with that of the Pytorch algorithm.

b. You need to calculate precision, recall, f1-score and accuracy of the model. You can use the sklearn code provided in the code snippet for this.

5. You can use any number of training epochs. Any additional analysis or findings from the dataset is well appreciated.

Note: *The program can be written in C / C++ / Java / Python programming language from scratch. No machine learning /data science /statistics package / library should be used for model creation.*

Relevant information:

Source: https://cneos.jpl.nasa.gov/ca/

Dataset Filename: `neo.csv`

Data Description:

Number of Instances: approx. 90,000

Number of Attributes: 9 + binary class = 10

Submission Details: (to be submitted in CSE-Moodle, **by one representative of the group**)

1. ZIPPED folder containing code (with comments) and the dataset files
2. Report (in pdf format)

Submission Guidelines:

1. You may use one of the following languages: C/C++/Java/Python.
2. Your Programs should run on a Linux Environment.
3. You are **not** allowed to use any library apart from these (Also explore all these libraries if doing in Python, or equivalent of these):

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import operator
from math import log
from collections import Counter
from statistics import mean
```

   Your program should be standalone and should **not** use any *special purpose* library for Machine Learning. Numpy and Pandas may be used. And, you can use libraries for other purposes, such as generation and formatting of data.

4. You should submit the program file and README file and not the output/input file.
5. You should name your file as <GroupNo_ProjectCode.extension> (e.g., Group1_WONN.pdf or Group1_WONN.zip).
6. The submitted program file *should* have the following header comments:

```
# Group Number
# Roll Numbers : Names of members (listed line wise)
# Project Number
# Project Title
```

7. Submit through CSE-MOODLE only.

Link to course page: https://moodlecse.iitkgp.ac.in/moodle/course/view.php?id=561

You should not use any code available on the Web. Submissions found to be

plagiarized or having used ML libraries (except for parts where specifically allowed) will be awarded zero marks.

For any questions about the assignment, contact the following TA:
Sumanta Dey (Email: sumanta.dey@iitkgp.ac.in)