ASSIGNMENT 2 BACKPROPAGATION NETWORK

CISC/CMPE 452/COGS400/CISC 874

15 marks

Assignment 2: The Original Dataset

- You will use the handwritten character database for this assignment published by Lecun et al available at the following URL yann.lecun.com/exdb/mnist/ (accessible on OnQ as MNIST handwritten digit database, Yann LeCun, Corinna Cortes an.pdf)
- Four files are available on the website (accessible on OnQ as given below):

  train-images-idx3-ubyte.gz: training set images (9912422 bytes)
  train-labels-idx1-ubyte.gz: training set labels (28881 bytes)
  t10k-images-idx3-ubyte.gz: test set images (1648877 bytes)
  t10k-labels-idx1-ubyte.gz: test set labels (4542 bytes)

- Information about the data is provided on the URL or the pdf file.
- **MNIST dataset can also be loaded directly in Python using packages such as Keras and scikit-learn.**

Assignment 2: Loading the Data in Python

- Using Keras package:

```
from tensorflow.keras.datasets import mnist
# Load data
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

- Using scikit-learn package:

```
from sklearn.datasets import load_digits, fetch_openml
from sklearn.model_selection import train_test_split

# Load data from https://www.openml.org/d/554
X, y = fetch_openml('mnist_784', version=1, return_X_y=True)

# Split data into train and test splits
X_train, X_test, y_train, y_test = train_test_split(
    X, y, train_size=60000, test_size=10000, shuffle=False)
```

- Both will load same data but input features (i.e. pixel values for an image) will have different dimensions
  - *i.e. Multi-dimensional array of shape 28x28 Vs. One-dimensional array of shape 784*
- For more details, please check:
  - https://www.tensorflow.org/api_docs/python/tf/keras/datasets/mnist/load_data
  - https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_openml.html

Assignment 2 – Your Task

- Implement a Backpropagation training algorithm in a multilayer perception to learn to classify handwritten characters using the MNIST training dataset and the corresponding label dataset.
- The input should be the training data and the output should be the corresponding digit class. In other words, you must decide the following design criteria:
  - *Initial weights and learning rate*
  - *Training iterations and terminating criteria*
  - *Number of layers and nodes*
  - *Momentum*
- You are expected to implement the Backpropagation algorithm to solve this problem. Use sigmoidal output function at every layer and node. **You cannot use any GUI based tool or built-in libraries to create/train/test the ANN.**
- Note down the training time and the process of selecting optimal values of the model parameters and the final parameter values used. You must write about these in the final report text file to be submitted with the code.

General Instructions for Code and Submission (for all assignments)

- You can use any programming language (preferred Python, C, C++, Java – No GUI tool or built-in libraries should be used to build/train/test the ANN)
- Make one zip file named as Asgn2_studentID which should include
  - *A text file containing the ANN design, accuracy in precision, recall, and confusion matrix*
  - *Program code with comments in the code to explain what each of your program files and functions are for.*
- Upload zip file to the OnQ site.
- Submit the following (see the mark distribution)
  - *Program code with comments, any associated libraries needed to execute your code (and/or executables and instructions for execution) and the output labels generated by your code for the test data (i.e., test data with generated labels).*
  - *An output text file similar to the test data set with the predicted output labels instead of the given class labels.*
  - *A report text/WORD file describing the structure of the ANN (diagram or description), parameter values used and accuracy of the model in terms of precision and recall and a confusion matrix.*

Mark Distribution – 15 Marks

- 5 marks - for fully functional executable code with comments in the code
- 1 mark - for the output text file with predicted class labels for the test data
- 9 marks - for the report text file containing
  - *ANN structure (1 mark) – number of nodes and layers*
  - *Parameter values such as node output function, the initial and final weight vectors, learning rate, parameter for momentum, training iterations and terminating condition, and other value choices made (2 marks) and why you made the above choices*
  - *Results (4 marks) – Overall good accuracy (1 mark) in terms of precision-recall (for test data only) and confusion matrix. Show confusion matrix for the trained model for the training dataset (1.5 marks) and also for the test data (1.5 marks) .*
  - *Critical discussion (2 marks) - are results good/bad, why, what were the challenges other than training time.*