ECE 457 B: COMPUTATIONAL INTELLIGENCE

ASSIGNMENT #4

Due Date: March 21, 2023

NOTE: To help with the adequate marking of your assignments, please follow these rules carefully:

1. All work should be carried out **individually.** Submission of the solutions is mandatory and the mark has to be at least 40% to be counted.
2. The first page of the assignment must have the name and ID of the student
3. The items required must be numbered, labelled, and in numerical order.
4. The solutions can be handwritten (in very neat way) or typewritten (best option). All graphs should be computer generated. You can use existing libraries. Please include the code for each problem and mention which library you have used.
5. All pages must be numbered sequentially
6. Show your steps and state any additional assumption you make.
7. Presentation of your results and the organization of your copy count for **10% of the overall mark. Completion level of the assignment count for 5%.**

Problem 1 (Support Vector Machines)

We need to construct an SVM classifier to classify data from Wine Dataset; Our main target is to try to classify Wine's Quality based on the inputs. Please download the dataset from this link (The dataset is originally from UCI: https://archive.ics.uci.edu/ml/datasets/wine+quality)

a. Load Wine dataset and discover its parameters, you may need to normalize data if it improves your results.
Then use SVM in Sklearn library to classify Wine quality, vary the hyperparameters as follows: Use Kernels of RBF, Linear, and Poly. Use the regularization parameter "C" as [1,10,50,100] for RBF and Poly, and [1,10,20,30] for Linear Kernel.
Add results to a **table**.

b. Explain your findings of the results from Part a.

c. Modify your best kernel model to find a **better performing model** by changing hyperparameters "C and Gamma" by trying at least 10 different combinations and record results in a **table**. Comment briefly on your findings.

Problem 2 (DNN/CNN)

Using your preferred deep learning library, train a small convolutional neural network (CNN) to classify images from the CIFAR10 dataset. Note that most libraries have utility functions to download and load this dataset (TensorFlow, PyTorch).

Using the API for loading the dataset will readily divide it into training and testing sets. Randomly sample 20% of the training set and use that as your new training set for the purposes of this problem. Use the test set from the original dataset for validation. Normalize your training and testing sets using Min-Max normalization.

1- Build a multi-layer perceptron with the following layers:
   - Dense layer with 512 units and a sigmoid activation function
   - Dense layer with 512 units and a sigmoid activation function
   - Dense layer (output layer) with 10 units (representing 10 classes in the dataset) and a suitable activation function for the classification task

2- Build a Convolutional neural network with the following architecture:
   - 2D Convolutional layer with 64 filters (size of 3x3) and ReLU activation function
   - 2D Convolutional layer with 64 filters (size of 3x3) and ReLU activation function
   - Flatten layer (to pass to the Fully Connected layers)
   - Fully connected (Dense) layer with 512 units and a sigmoid activation function
   - Fully connected layer with 512 units and a sigmoid activation function
   - Dense layer (output layer) with 10 units and a suitable activation function for the classification task

3- Build a Convolutional Neural network with the following architecture:
   - 2D Convolutional layer with 64 filters (size of 3x3) and ReLU activation function
   - 2x2 Maxpooling layer

- 2D Convolutional layer with 64 filters (size of 3x3) and ReLU activation function
- 2x2 Maxpooling layer
- Flatten layer (to pass to the Fully Connected layers)
- Fully connected (Dense) layer with 512 units and a sigmoid activation function
- Dropout layer with 0.2 dropout rate
- Fully connected layer with 512 units and a sigmoid activation function
- Dropout layer with 0.2 dropout rate
- Dense layer (output layer) with 10 units and a suitable activation function for the classification task

Use a batch size of 32, utilize Adam as the optimizer and choose an appropriate loss function while monitoring the accuracy in both networks. Train each network for 5 epochs.

a) Report the training and testing accuracy for all three networks and comment on the performance of the MLP vs CNNs.
b) Plot the training and validation curves for the two CNNs and comment on the output. How does the training time compare for each of the CNNs? How does the different architectures influence these results? What do you expect the accuracies to be if the networks were trained for more epochs?