Machine Learning for Engineering Applications
Homework #5
Fall 2023

Due: **11/1/2023 by 11:59 PM – submit your zip file to CANVAS**

Concrete Dataset

□ Dataset info and download: https://data.mendeley.com/datasets/5y9wdsg2zt/2

The CNN Network

□ Develop your own *CNN* model to classify all classes.
□ Provide the training and test confusion matrices.
□ Provide the test accuracy, precision, recall, and F1-scores to a text file.
□ Provide the Loss and Accuracy curves for training and validation (you can use a single plot for these four curves)
□ Expected results: High 90's for training, validation, and testing **without overfitting/underfitting.**

Submit:

□ Python file(s) (.py)
□ Confusion matrix image (.png or .jpg)
□ Curve image(s) (.png or .jpg)
□ The text file with all the metrics
□ The SLURM/.OUT files used for the LEAP cluster execution

Rubric per Part and Overall:

□ Header in the code (5pts)
□ Comments in the code (10pts)
□ Running code without errors (20pts)
□ Executes requirements & produces required output information (65pts)

Template:

```
#*************************************************************************************
# Damian Valles
# ML – HW#1
# Filename:  hw1-perceptron.py
# Due:  Sept. 6, 2023
#
# Objective:
# To demonstrate header and comment expectations for assignments for the semester.
#**************************************************************************************
#Importing all required libraries
from sklearn import datasets
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Perceptron

#Downloading the dataset, creating dataframe
iris = datasets.load_iris()

#Separating data & target information
X = iris.data[:, [2, 3]]
y = iris.target

#Data split for training and testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1, stratify=y)

#Scaling training data
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)

#Creating perceptron with hyperparameters
ppn = Perceptron(max_iter=40, eta0=0.01, shuffle=True)

#This is training the model
ppn.fit(X_train_std, y_train)

#Scaling test data
sc.fit(X_test)
X_test_std = sc.transform(X_test)

#Testing the model data
y_pred = ppn.predict(X_test_std)
```