

EEE 6608: Machine Learning and Pattern Recognition

Midterm assignment

Due date: Feb 12, 2022 (the portal will close at 10am)

Midterm assignment instructions

1. Use the Happy dataset uploaded in the Files section. The dataset is for binary classification: it is an image dataset, where the classification task is to label the faces as happy or unhappy. Each sample is an $64 \times 64 \times 3$ image. The dataset has 600 train samples and 150 test sample. You may divide the train set into train and validation for experimental purposes. You can use sklearn library for splitting the data into these partitions.
2. Write the Logistic Regression code by yourself from scratch (do not use any built-in classification function).
3. Write the fully connected neural network code using Tensorflow or PyTorch from scratch (do not use higher level APIs like keras).
4. Once you are confident about your models (by evaluating the performance on validation partition of the dataset), test the performance for **both** models:
 - a. Plot the loss with iterations and show that the loss is non-increasing with iterations
 - b. Plot the test partition accuracy with learning rate
 - c. Plot test partition accuracy with varying training set size.
5. Remember that you need to vary the training sample size (4c) but use the whole test partition for generating these plots.
6. You can use the following code to get the train and test features and labels from the dataset files.
7. You will have to show a presentation on how you solved this assignment with **no more than 5 slides**.
8. You will have to submit a report on this assignment that will contain your **code, result plots and a brief discussion**. You can add other sections in the report if you deem necessary.

```
import numpy as np
import h5py
def load_dataset():
    train_dataset = h5py.File('datasets/train_happy.h5', "r")
    test_dataset = h5py.File('datasets/test_happy.h5', "r")

    train_set_x_orig = np.array(train_dataset["train_set_x"][:])
    train_set_y_orig = np.array(train_dataset["train_set_y"][:])

    test_set_x_orig = np.array(test_dataset["test_set_x"][:])
    test_set_y_orig = np.array(test_dataset["test_set_y"][:])

    classes = np.array(test_dataset["list_classes"][:])

    train_set_y_orig = train_set_y_orig.reshape((1, train_set_y_orig.shape[0]))
    test_set_y_orig = test_set_y_orig.reshape((1, test_set_y_orig.shape[0]))

    return train_set_x_orig, train_set_y_orig, test_set_x_orig, test_set_y_orig, classes
```