MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG483 Introduction to Computer Vision – Assignment

Date handed out: **11 May 2023**
Date submission due: **25 May 2023 23:00**

Image Classification

Objectives: The purpose of this assignment is to familiarize yourselves with the fundamental deep learning solutions to computer vision problems and the framework for classification problems. The assignment aims to give insights into deep learning-based computer vision research and its evaluation methods.

Description: In this assignment, you are required to implement an image classification system based on deep learning methods and to evaluate it with the provided dataset. The text continues with detailed explanations of the methods and requirements.

Image Classification: The main purpose of image classification is to determine the class label of a query image. The classification is done by evaluating the semantic contents of the query image. However, there is difficulty in revealing the semantics of images due to the semantic gap. To overcome this difficulty, images are described as feature vectors which are higher-level representations than collections of numbers. With these feature vectors, image classification can be formulated as a learning problem to match an image representation with the class label. In this assignment, you are required to construct a fully connected feed-forward neural network with a nonlinearity function (ReLU) between layers and train it with an optimizer (Batch (Vanilla) Gradient Descent) using the extracted feature vectors. While training the fully connected feed-forward neural network you are required to use the SoftMax (Cross-Entropy Loss) function to minimize the difference between the actual class label and the estimated one.

Design Requirements: You are required to implement the aforementioned image classification system step by step;

1) (10 points) Write a function ***imageLoaderAndFeatureExtractor*** to initially load the database (given in the Assignment 2) and then form the class labels (building, forest, and glacier). Please note that a class label of images will be simply the name of an image (this is already done in Assignment 2). Subsequently, resize all the images to the same size (you should decide the size). Then, the features will be vectorized pixels of your image. For example, if you resize all images to 32x32x3, then your feature vector for each image will be 1x3072. The function should return the features and class labels.

2) (5 points) Write a function ***setFormer*** to form the training, validation, and testing sets. (This is already done in Assignment2)
   Training and testing sets: are already given to you.
   Validation set: you should form this by taking the first 25% images of each class of training set.

3) (20 points) Design two different fully connected feed-forward neural networks with 2 and 3 hidden layers.

   - How many neurons to use in each hidden layer? (find out by yourself)
   - Use a nonlinearity function (ReLU) between the layers.
   - Use SoftMax (Cross-Entropy Loss) function to minimize the difference between the actual class label and the estimated one.
   - Initialize weights (see lecture notes and decide what to use)
   - Regularization (see lecture notes and decide what to use)

4) Train two different designed fully connected feed-forward neural networks (with 2 and 3 hidden layers) on the validation set.

   - (5 points) Train the network with an optimizer (Batch (Vanilla) Gradient Descent)
   - (5 points) Regularization hyperparameter (find out by yourself)
   - (5 points) Learning Rate hyperparameter (find out by yourself)
   - (5 points) Number of epochs (find out by yourself)

   How to monitor and choose the right hyperparameters, you need to check the following:
   - (10 points) Plot and visualize the loss curve (see lecture notes 11, pages 44 and 45).
   - (10 points) Plot and visualize the accuracy curve (see lecture notes 11, page 46).

   Important hints about the implementation are as follows:

   - Saving model at intermediate epochs. To get a good fit for data, the number of epochs that the network will be trained should be determined. This can be done with the help of loss history plots that show the loss computed using training and validation sets for each epoch. After examining the plot, one can decide on the number of epochs. In order not to retrain the network, you can save model and optimizer parameters at some epochs (i.e. at every 5 epochs).
   - Setting a seed for random number generators. This allows the experiments to be repeatable since whenever it is run, it starts from the same random parameters.

5) (10 points) After training (deciding/modifying all hyper-parameters), you should evaluate these configurations (both with 2 hidden layers and 3 hidden layers fully connected feed-forward neural networks) with the testing set and print their accuracy.

6) (5 points) Good test accuracy.

7) Along with the implementation, you are required to explain the rationale behind your choices and the results of the experiments. It should include at least the following items;

   - (5 points) Discussion on the effects of the number of layers.
   - (5 points) Discussion on the effects of the number of neurons in the hidden layers.

Programming Requirements:

In this assignment, you can use MATLAB or Python for the implementation. Your file name should be your name_surname_id. You can use built-in functions (no restrictions). You must use comments and explain what your code is doing step by step.

Submission Requirements:

- You can only submit one file with all functions defined inside it and saved features. Otherwise -5 pts.
- I will put your code inside the "Database" folder which is given to you and run the code. Hence, you must implement your code to work properly accordingly. Otherwise -5 pts.
- Uncompiled codes will be graded as zero.
- Late submissions will not be accepted and graded as zero.
- Any other implementation rather than fully connected feed-forward neural networks (such as K-NN, SVM, CNN, etc.) will be graded as zero.