LAB 7: Training with Convolutional Neural Network

Due date: Nov 29, 11:59 am

This is a competitive group lab work. In your original group solve the following assignment and submit one document per group. Do not forget to include names of your members. This is a competitive group assignment. There will be two winning groups. Each winning group will receive a modest holiday gift from myside. You can also solve this assignment individually if you do not want to be in a group.

Learning Objective: Students will learn how to work with a CNN trained using 2D data and compare its performance with a balanced version of the CNN.

Tasks: You will train 2 CNN models. One will be without balancing and the other will be with balancing technique. You can use any balancing technique that gives the best performance for this dataset. You will then compare the classification performance between the two models i.e., with and without balancing approach. A benchmark has been given. Your performance should be closer or higher to the benchmark which is 72% based on a weighted CNN model. So your model performance should be 70% or higher.

Helper code has been given for reading .mat files.

Description: The dataset is a time-series obtained SpectraQuest's Gearbox Fault Diagnostics data.

This original dataset is transformed to a 2D dataset containing the data and its labels. These are saved in Matlab's .mat format

The dataset "2D_img.mat" contains 994 different 2D time-series. The length of each time-series is 100. Thus, there are 994 examples of 2D data each of dimension 100*100.

The second dataset contains the labels. Label 0 denotes the healthy category and 1 denotes the broken category. There are 489 examples of 2D time-series belonging to healthy category and 505 examples of 2D time-series belonging to broken category. There is a slight imbalance in this dataset upon careful inspection. Consider each 2D time series to be an image. Thus, you will feed each of these 2D images into a CNN. You will train the CNN using 5 fold Cross-Validation (CV) on the entire 994 examples (no need for nested CV)

You have to check how many layers, filter size work best to give a training accuracy of 70% or higher after a balancing approach. As a benchmark, compare with my results: I got approx. 72% by applying the weighted balancing approach. The configuration for my weighted CNN were:

- Two convolutional layers
- The first convolutional layer has 12 filters of size (3 × 3) which are applied at a stride of 1 pixel.
- It is followed by the rectified linear unit (ReLU) activation function and a max pooling layer of size 2 x 2 with stride 2.
- The second convolutional layer has twice the number of filters of same size with a stride 1.

- It is again followed by a ReLU and a max pooling layer of same size and stride as before.
- The last layer is a fully connected layer which uses the features extracted from the previous layers for classifying. This layer computes the class scores and the Weighted Cross-Entropy loss function.
- The learning is performed using the adam optimizer with a learning rate of $3e-4$ and a mini-batch size of 128. So at a time we train on 128 images in each epoch and the total number of epochs are 25

  Training Set: Total CV Acccuracy = 72.2324%
  ~~~~~~~~~~~~Training Performance Stats obtained from Confusion Matrix: ~~~~~~~~~~~~~~
  Out of 489 actual healthy samples, correctly predicted healthy (TN)352
  Out of 505 actual abnormal samples, correctly predicted broken(TP)366
  FP (CNN ppredicted broken, but there were not broken) 137
  TNR(when its not broken, how often does it predict not broken) 71.9836
  TPR(when its actually broken, how often does it predict broken) 72.4752

  Deliverables: In a folder you should have the data files and the python code. Upload the zipped folder. The instructor should be able to run the python code without having to manipulate any path file etc.
  ~~~~~~~~~~~~