Data Science Lab
Lab #7

Politecnico di Torino
November 20-21, 2019

Intro

The main objective of this laboratory is to put into practice what you have learned on classification techniques. You will mainly work on audio signals. In particular, you will try to build a classification model that is able to identify which digit was uttered analyzing the content of short audio samples from different speakers.

Important note. For what concerns this laboratory, you are encouraged to upload your results to the competition we launched on Kaggle, even if the submission will not count on your final exam mark. If you do not have it yet, you will need to create an account. It is **mandatory** to follow the subscription instructions reported in the Kaggle guide from the course website, otherwise your score will be excluded from the competition. Reference Section 3 to read more about the competition.

1 Preliminary steps

1.1 Datasets

1.1.1 Free Spoken Digit Dataset

The dataset for this laboratory has been inspired by the Free Spoken Digit Dataset.

It is composed of 2,000 recordings by 4 speakers of numbers from 0 to 9 with English pronunciation. Thus, each digit has a total of 50 recordings per speaker. Each recording is a mono `wav` file. The sampling rate is 8 kHz. The recordings are trimmed so that they have near minimal silence at the beginnings and ends.

The data has been distributed uniformly in two separate collections:

- Development (dev): a collection composed of 1500 recordings **with** the ground-truth labels. This collection of data has to be used during the development of the classification model. Each file in this portion of the dataset is a recording named with the following format `<Id>_<Label>.wav`.
- Evaluation (eval): a collection composed of 500 recordings **without** the labels. This collection of data has to be used to produce the submission file containing the labels predicted for each evaluation recording, exploiting the previously built model. Each file in this portion of the dataset is a recording named with the following format `<Id>.wav`.

So far, you should be used to work, developing your models, with training, validation and test sets. In this case, the Development data must be used to tune your hyper-parameters while you should consider the Evaluation portion as the actual test set.

1.1.2 Dataset tree hierarchy

The dataset archive is organized as follows:

- `dev`: the folder that contains the labeled recordings.

- eval: the folder that contains the unlabeled recordings. Use this data to produce the submission file containing the predicted labels.

- sample_eval_sumbission.csv: a sample submission file.

You can find the dataset on the competition we launched on Kaggle. Head to Section 3 to know how to register on Kaggle and download the dataset. For the sake of simplicity, you can also download the dataset at:

https://github.com/dbdmg/data-science-lab/raw/master/datasets/free-spoken-digit.zip

2 Exercises

In this laboratory you have a single classification task to carry out.

2.1 Free Spoken Digit classification

In this exercise you will build a complete data analytics pipeline to pre-process your audio signals and build a classification model able to distinguish between the classes available in the dataset. More specifically, you will load, analyze and prepare the Free Spoken Digit dataset to train and validate a classification model. Finally, you will be able to upload your classification results and participate to the lab competition.

1. Load the dataset from the root folder. Here the Python's os module comes to your help. You can use the `os.listdir` function to list files in a directory. Furthermore, you can use the `wavefile` module from SciPy to read a file in wav format. You can read more about it on the official documentation.

2. Focus now on the data preparation step. You should have noticed that `wavfile` gives you an array of floating point values, plus the sampling rate. Before continuing, take you your time to answer these questions:

    - what do these numbers represent?
    - were the audios recorded under the same conditions? (e.g. recording volume, noise, etc.)
    - do the arrays have an equal length? How different lengths could impact on your pre-processing solution? If it was needed, could you figure a way out to align them to the same number of sample?

    Now, in order to train your model, you are required to design and build a vector representation. This mainly involves extracting several features out of your initial representation. Bear in mind that, since you are dealing with audio signals, you can work either on the time domain or the frequency one. In the former case, you might opt, for example, to split the signal into chunks and characterize them by means of statistical measures (e.g. mean, standard deviation). In the latter case, you can base your features on the frequencies contained in the signal. This involves reshaping the signal using a transformation function (e.g. Fourier transform) and work on its spectrum of frequencies (e.g. spectogram). Data preparation on frequencies can be hard to carry out. To know more about it, please refer to Camastra and Vinciarelli 2015 and Oppenheim and Schafer 2014, and Presti and Neri 1992.

    Identify a set of possible feature candidates and transform your data using them.

3. Once you have your vector representation, choose one classification algorithm of those you know. Then, perform the classic training-validation pipeline on the Development dataset to identify the best set of hyper-parameters for your model. As you can read in section 3.3, we will evaluate your results on the Mean F1 score. Hence, it is a reasonable option trying to optimize it on the Development dataset [1].

    > **Info:** the Mean F1 score, also known as *macro average*, calculates the F1 for each label, and computes their unweighted mean. This does not take label imbalance into account.

4. Assign a classification label (i.e. the spoken digit) to each recording in the Evaluation dataset.

5. Submit your results to the Kaggle competition. Head to section 3 to know more about it.

6. Compile your final report and upload it to the "Portale della Didattica" as described in section 3.2.

[1] Actually, since your task does not present class imbalance, optimizing the classification accuracy would have been equally correct.

3 Submitting you work

For this laboratory, you have to upload two files to two different web sites. The first file contains the classification results, the second file contains a report on the experiments you carried out. The following sections provide further details on that.

Deadline. You can submit your work until **Monday November 25, 11:59PM**. Late submissions will not be evaluated.

3.1 Submit your classification results

In order to get you results evaluated, you have to upload a result file on our Kaggle competition. You have to register an account and change your username to accomplish the submission correctly. Please reference the Kaggle guide from the course website to go through the submission procedure.
You can find the competition at https://www.kaggle.com/t/eb3c50d1381a43efbfb6ff565710ff47

The submission file has to be a `.csv` file formatted as follow:

```
Id,Predicted
0,0
1,0
2,0
3,1
4,1
...
```

As you can see, it must contain an header line and a row for each recording in the Evaluation collection. Each row must have two fields:

- the `Id` of the recording, as an integer number
- the `Predicted` label, as an integer number

You can find a sample submission file on the competition web site.

3.2 Upload your report

You are required to upload a single PDF file. Please respect the following requirements:

- state clearly which pre-processing step characterized your final solution;
- describe which classification algorithm you used;
- describe which validation strategy you adopted and which are the best hyper-parameters you found on the Development set.

If you have developed your solution on a Jupyter notebook, you can export it as a PDF and use it for the submission. However the file must be sufficiently commented.
The file must be uploaded to the "Portale della Didattica", under the Homework section of the course. Please use as description: `report_lab_7`.

3.3 Evaluation

Your classification results will be evaluated via the Mean F1 score. Your report will be evaluated via the old, always-working human reading.

References

[1] Francesco Camastra and Alessandro Vinciarelli. *Machine learning for audio, image and video analysis: theory and applications*. Springer, 2015.

[2] Alan V Oppenheim and Ronald W Schafer. *Discrete-time signal processing*. Pearson Education, 2014.

[3] Letizia Lo Presti and Fabio Neri. *L'analisi dei segnali*. CLUT, 1992.