Machine Learning II
2024/2025 (CC3043), DCC/FCUP

Project 2024/2025
Submission deadline: November $29^{\text{th}}, 2024$

This work will need to be submitted using the Moodle platform. It will be developed during the practical classes, but it is expected that the students will complement this work using extra-class hours.

1 Objective

The objective of this work is to develop deep learning classifiers for urban sound data.

2 Dataset

The data that you will be using belongs to the urbansound8k dataset, which contains 8732 labeled sound excerpts of duration less than or equal to 4 seconds. The sounds excerpts are labelled according to the following 10 classes:

- air conditioner
- car horn
- children playing
- dog bark
- drilling
- engine idling
- gun shot
- jackhammer
- siren
- street music

The objective of this work will be that of building classifiers able to determine to which of the previous 10 classes a given, unseen, sound excerpt belongs to.
A detailed description of the dataset as well as instructions about how to download the dataset can be found at this link: `https://urbansounddataset.weebly.com/urbansound8k.html`
and in the following paper:
`http://www.justinsalamon.com/uploads/4/3/9/4/4394963/salamon_urbansound_acmmm14.pdf`

3 Implementation

In order to complete this work, you will need to implement **two out of the three** following classifiers, at your choice:

- A classifier based on a multilayer perceptron (MLP)
- A classifier based on a convolutional neural network (CNN)
- A classifier based on a recurrent neural network (RNN)

For each of the chosen classifier, the implementation will need to consider the following steps:

1. Data pre-processing and preparation
2. Model architecture definition
3. Training strategies
4. Performance evaluation.

3.1 Data pre-processing and preparation

In this step, you will need to form the input of your neural networks, by processing original raw sound data. Take into account that raw sound files will possibly have different durations and different sampling rates. You will then need to uniformize and normalize the network input and decide if you want to give as input of the network the raw data or some features extracted from it. You can use functions from libraries (e.g., `librosa`) to perform signal processing and feature extraction.
The class labels that will be used to train the models and evaluate their performance can be obtained from the file `UrbanSound8K.csv` provided with the dataset.

3.2 Model architecture definition

In the case of the MLP, this steps consists in the definition of the number of layers, number of neurons for each layer, and in the choice of the activation functions to adopt at each layer.
In the case of CNNs, a first design choice is related to using 1-dimensional or 2-dimensional inputs. 1D inputs can be obtained by applying the CNN directly on portions (windows) of the original sound signal (after downsampling and normalization). 2D representations of sounds, on the other hand, are based on time-frequency analysis of sounds, as the Mel-frequency cepstral coefficients (MFCCs) described in `http://www.justinsalamon.com/uploads/4/3/9/4/4394963/salamon_urbansound_acmmm14.pdf`. After defining if you are using a 1D or 2D CNN, model definition involves the definition of the model architecture to adopt (number layers, kinds of layers, filter dimensions, number of feature maps, etc.)
In the case of RNN, the design of the model architecture encompasses the kind of units used in the neural network and kind of connections (unidirectional vs. bidirectional). Possible choice in this sense are vanilla RNNs, gated recurrent units (GRUs), long-short term memory (LSTM) networks, etc.

3.3 Training strategy

In this part, you will need to define the details for the training strategies adopted to train the models. These details include:

- Optimizer

- Learning hyperparameters (e.g., learning rate, mini-batch size, number of epochs, etc.)
- Regularization techniques to adopt (e.g., early stopping, weight regularization, dropout, data augmentation, etc.)
- Possibility of using transfer learning
- etc.

3.4 Performance evaluation

The performance of the considered classifiers will be evaluated using the urbansound8k dataset taking care of appropriately separating train, validation, and test data. In order to do that, you will need to use the 10-fold cross-validation scheme described at `https://urbansounddataset.weebly.com/urbansound8k.html`. More specifically, use a 10-fold cross-validation strategy where, in each iteration, 1 fold is used for test, 1 fold is used for validation, and 8 folds are used for training.
Then, classification performance will be quantified using a confusion matrix (cumulative over the 10 folds) and reporting the average classification accuracy and its standard deviation over the 10 experiments included in the 10-fold cross-validation scheme.

4 Bonus

The bonus part of the work will be evaluated up to 1 point (over the 8 points reserved for the work). The maximum grade for the project is, nevertheless, 8.
The objective of the bonus task is to evaluate the robustness of the proposed classifier against *adversarial examples*. In order to do that, you can implement the approach dubbed DeepFool, which attempt to find the minimal input perturbation leading to a classification error. This approach is described in this paper: `https://openaccess.thecvf.com/content_cvpr_2016/html/Moosavi-Dezfooli_DeepFool_A_Simple_CVPR_2016_paper.html`
Further information regarding the DeepFool approach and the evaluation of robustness against adversarial perturbation can be found here: `https://medium.com/machine-intelligence-and-deep-learning-lab/a-review-of-deepfool-a-simple-and-accurate-method-to-fool-deep-neural-networks-b016fb ~:text=DeepFool%20finds%20the%20minimal%20perturbations,so%20the%20classification%20label%20changes.`
Alternatively, you can obtain bonus evaluation by implementing network architectures that were not discussed in detail in the classes as, for example, attention modules, transformers, diffusion models, etc.

5 Submission of the solution

Your project must be delivered in Moodle by November 29, 2024, at 23:59:59;
You are required to submit the following material:

- Final code solution, as a notebook. The notebook needs to contain the following information:
  - Brief description of the deep learning solutions considered for the problem (MLP, CNN, and/or RNN).
  - Implementation description: Motivate your choices in terms of pre-processing and data preparation step, model design, and learning strategy adopted for the considered classifiers.

    - Results: Classification performance obtained by the different classifiers implemented.
    - Discussion and conclusions: Comments on the performance obtained and final remarks.
- Filled auto-evaluation file regarding the contribution of each member of the group.

Further information about the project submission and presentation:

- The code provided as the solution will need to allow to train the considered models and reproduce the results that you reported. **Please do not include dataset files.** You can assume I have local access to the dataset.
- The work must be done by groups of **3 people**. Groups formed by less than 3 people must be justified and approved before starting working.
- Delays in the submission will incur in a grade penalization and eventually in not accepting the work.
- All works must be presented on December 5th and 12th. All group members must be present during the demonstration. **If a member of the group is not present to the work presentation, he will receive a zero grade for this work, thus implying failing to pass.**
- Each member of the group must comment on their contribution to the work, and must know exactly what the other members of the group have done. **Failing to describe in details what your solution is doing and why will determine a penalization in the overall evaluation of the project.**