

Introduction to ML

Exercise 5

Due: 28.1.2020 22:00

Yosi Shrem and Joseph Keshet

January 13, 2020

Guidelines

1. You are allowed to work in pairs.
2. For pairs, only **ONE** student should submit.
3. In order to submit your solution please submit the following files:
 - (a) **details.txt** - A text file with your full name (in the first line) and ID (in the second line), see the attached example file **details.txt**.
 - (b) **ex_5.py** - The file that contains your main function (attach ANY additional files needed for your code to run).
 - (c) **ex_5_report.pdf** - A pdf file in which you describe your model and parameters.
 - (d) **test_y** - your model's predictions on the given test set (see instructions below).

Follow the instructions and submit all files needed for you code to run.

Good Luck!

Ex5

You will train a model to classify a speech command using speech data. The provided dataset contains 30 different categories of commands. Your task is to train a classifier that classifies this data. You can use all models / techniques we have talk about during the course, e.g. DNN / CNN / RNN / Dropout / BatchNormalization / Data Augmentation / Optimization Methods / Etc.

Data. Each speech utterance is ~ 1 sec long. You are provided with a python loader called `gcommand_loader.py`. This data loader will load your data, create batches, randomly shuffle the data, etc. (An example of using this loader is provided in the `data_loader.py` file). Suggestion: save a small portion of the dataset and use it for debugging. When you are done, load the entire dataset to train your model.

Instructions

1. Your goal is to train a multi-class classifier based on all what we have learned during the course. Your model should reach the best performance you can get on the validation set.
2. You will receive the data already split to train, validation, and test sets. Each category will be in a different folder.
3. You are provided with a file named: `gcommand_loader.py` to read the data and extract features from it. Additionally this file will create batches and shuffle the data for you.
4. You should train and validate your model. Finally, you should output you model's predictions on the examples in test folder to a file named `test_y` where the *i*th row should contain
<file name>, <prediction of your mode to the *i*th example>.
Example:
`0.wav,happy`
`1.wav,bed`
...
`6834.wav,three`
5. Describe your model's architecture and explain how you chose it and all hyper-parameters in `ex_5_report.pdf`.

6. Submit **ALL** source code files along with your predictions file `test_y`. Note that you name it exactly as specified. Your grade will be based on your performance on the test set (accuracy of 90%+ will be graded as 100).