Learning challenge

Last updated: 2019-05-07

For this assignment you will participate in groups of 3 students in a learning challenge. You will need to submit your predictions, as well as a report describing your solutions and the code which you used to generate the predictions.

This assignment is worth 40% of your course grade. The assignment grade will be based on the quality of your work as judged by the instructor based on your report and code (50%), as well as your score and ranking in the competition (50%). At the very minimum you should aim to do better than baseline solution on Codalab submitted by account `gchrupala`.

Your report should be a single-page PDF document, and should include the following:

- Brief description of your computational learning experiments, including:
  - feature engineering
  - learning algorithm(s) tried and parameter tuning
  - discussion of the performance of your solution
- Detailed specification of the work done by group members
- The name of the account under which you submitted your results to the competition on Codalab (see below)

Your code should be a plain Python script which can be run to generate your solution file. You do not need to include the training data. Put the report in **PDF format** and the code in a single zip file named with your group number, e.g. `Group7.zip`, and submit it to the BlackBoard assignment.

In addition you will need to submit your solution file to the competition server. The competition is hosted on https://competitions.codalab.org. One member of the team will need to get a codalab account, and will be responsible for submitting your solution. Indicate the name of this account in your report. See section **Submission to Codalab** for additional details.

IN SUMMARY: submission consists of two parts

1. Zip file with your report and your code (BLACKBOARD)
2. Submission of your solution (CODALAB)

Group work

Your report needs to contain a detailed description of who did what, so make sure to keep track of this information. Note: it is **not acceptable** to just say *All members worked together and contributed equally.*

If there are any problems with collaboration, such as serious disagreements, a group member not contributing, or a group dissolving, make sure inform the

course coordinator as soon as possible via email.

Code reuse rules

Remember this assignment is group work. You are **not allowed** to collaborate or share code with students outside your group. **Submissions will be checked for plagiarism.**

If you are found breaking the above rules you will be reported to the Board of Examiners for fraud.

You are, however, allowed to use code examples provided by the instructor during the course, or as part of the competition. You can also use any Python libraries of your choice as long as they are open source and publically available.

Dataset

Spoken and written numbers

In this challenge the task is to learn to recognize whether an image of a handwritten digit and a recording of a spoken digit refer to the same or different number.

- False: the image and the recording refer to different numbers
- True: the image and the recording refer to the same number

Each image is given as 784-dimensional vector, which represents 28x28 pixel grayscale image. Pixel intensities range from 0 (black) to 255 (white).

Figure 1: Handwritten digit 7

Each sound recording of a spoken name of a digit (e.g. "zero", "one" etc, pronuounced in Arabic) is given as an array of pre-extracted audio features, so called Mel-Frequency Cepstral Coefficients (MFCC). These features encode the characteristics of a 10 milisecond frame of speech. Each recording is of variable length, and thus each example is given as an array of shape (N, 13), where N is the number of frames in the recording, and 13 the number of MFCC features.

Data files

The dataset is available for download on BlackBoard, in numpy array format:

- `written_train.npy`: array with 45,000 rows and 784 columns
- `written_test.npy`: array with 15,000 rows and 784 columns
- `spoken_train.npy`: array with 45,000 rows. Each row is an object of shape (N, 13)
- `spoken_test.npy`: array with 15,000 rows. Each row is an object of shape (N, 13)
- `match_train.npy`: array with 45,000 boolean values (False or True)

The value at index j in the array from `match_train.npy` specifies whether the image at row j from `written_train.npy` and the audio at row j from `spoken_train.npy` refer to the same number or not.

You can load the files using the function `numpy.load` (you may need to specify `allow_pickle=True`).

Evaluation metric

The evaluation metric for this task is error rate accuracy (the proportion of incorrect predictions).

Submission format

You need to create an array of 15000 boolean values, specifying whether the images and sounds from the test data mare matched or not. Save this array in the file names `result.npy`, and then compress it as a `.zip` file and upload to the competition website. Make sure there are no internal folders in the zip file.

Method

There are three important restrictions on the method used:

- the method should be fully automatic, that is, by re-running your code it should be possible to re-create your solution file;
- the method shouldn't use any external training data;
- every software component used should be open-source and possible to install locally. This means that you cannot, for example, access a web service to carry out any data processing.

Submission to Codalab

The competition is hosted on Codalab at: competitions.codalab.org/competitions/22825?secret_key=eff5bd83-f701-4b97-bd4a-99129534b854

You can submit your results in the **Participate** link. After uploading your file, make sure to *submit to leaderboard*.

Over the course of the competition you can make 7 submissions. The results from all the participating teams will be displayed in the **Results** tab.