**Machine Learning Project**

**Spoken and written numbers**

In this challenge the task is to learn to recognize whether an image of a hand- written 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 (see attached file named datasets.zip)**

The dataset is available 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

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

**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.