Francisco Javier Sáez Maldonado
franciscojavier.saez@estudiante.uam.es

# Speaker Recognition

Deep Learning for Audio Signals

2022-04-21

## Contents

**1 Examining the code** **1**

## Introduction

Speaker recognition is the task of identifying a person from his/her voice. In this lab report, we will examine a speaker recognition system that will use *x-vectors* embeddings to classify the speakers. The system has been developed to solve the VoxCeleb challenge.

The system *extracts* the fixed-size embeddings trying to include in these representations as much discriminative information as possible for the speaker recognition task. A few architectures of neural networks can be used with this purpose, as well as different loss functions.

The code that I used to test the model, which was provided by Alicia Lozano, can be found in this link.

## Examining the code

In this section, we will examine the provided code in order to find the most relevant parts of it. Let us begin by examining the `trainSpeakerNet.py` file, since it is the main script. The code is properly commented so identifying the different parts is quite simple.

**Questions.**

- *Which line of the code trains the model?*

  We found `line 169`:

  ```
  loss, traineer = s.train_network(loader=trainLoader);
  ```

  which trains the code. The variable `s` was previously initialized as `s = SpeakerNet(**vars(args));`, creating an instance of the SpeakerNet defined model class. This class has the called `train_network` function, which performs (vaguely explained) a classic forward pass with loss computation and returns the average loss and the `train eer`.

- *Which command loads a pre-trained model?*

  Having a look from lines $112-121$, we find that in lines $116$ and $120$, the function `s.loadParameters(args)` is called. This function loads the pretrained model from a specific file. As we can see in the code, a distinction is made in our case, being able to load the model from the `save_path` or from the argument `args.initial_model`.

- *Which line evaluates the performance of the neural network?*

  A few lines below the training line, in line `176` we find:

  ```
  sc, lab, _ = s.evaluateFromList(args.test_list, print_interval=100, test_path=args.test_path,
  ```

  which evaluates the model in the `test_list`. In this function, all the features for each of the test file audios (*.wav*) files are extracted and then the scores are computed using `pairwise_distance`.

- *Which variable contains the scores of the trials list? In which file are those scores saved?*

  In line $177$ we observe that the scores are thresholded and stored in the variable `result`. Then, they are saved in the `scorefile` file. This is a variable that, in line $150$ opened a file called `result_save_path/scores.txt`, where `result_save_path` came as an argument from the execution of the script.

- *Which variable controls the number of epochs(iterations) that the model is trained?*

  We find in line $198$ and $201$ that the variable `it` is modified and checks the `max_epoch` indicated as an argument.