APPSA - Práctica 2: Reconocimiento de Locutor en VoxCeleb

Alicia Lozano Díez

Abril, 2022

Objetivo

Explorar y evaluar un sistema de Reconocimiento de Locutor basado en embeddings (x-vectors) sobre la tarea de VoxCeleb.

Materiales

- Presentación de la práctica - Moodle
- Guión de la práctica - Moodle
- Información sobre VoxCeleb Challenge - https://www.robots.ox.ac.uk/~vgg/data/voxceleb/competition2020.html
- Código de la práctica - Moodle y One Drive:
    - https://dauam-my.sharepoint.com/:u:/g/personal/alicia_lozano_uam_es/EeymgCUQ2EhPrAvEWf6SQm8By5753NkZHmvmPMslib0A3A?download=1
- Conjunto de datos de VoxCeleb 1 test para evaluación
    - https://dauam-my.sharepoint.com/:u:/g/personal/alicia_lozano_uam_es/ER9-4DcM62lLtyqMkcU0ZAwBGHMFaSrPZgzuLq0bAxLB-Q?download=1
- Subconjunto de datos de Voxceleb 2 dev para entrenamiento
    - https://dauam-my.sharepoint.com/:u:/g/personal/alicia_lozano_uam_es/EQkvCJ901zxApjlVbbSeyooBfa7it0iQP2Eie1MIsy3V-Q?download=1

1. Preparación del entorno

1.1. Preparación del entorno software

Primero vamos a descargar el código de la práctica.

Para ello, necesitaremos descargar de Moodle el script **code_download_onedrive.sh**, y ejecutar las siguientes líneas de código, que nos permitirán subir el archivo a Google Colab desde el disco local:

```
from google.colab import files
uploaded = files.upload()
```

```
Elegir archivos code_downl…nedrive.sh
• code_download_onedrive.sh(text/x-sh) - 398 bytes, last modified: 7/4/2022 - 100% done
Saving code_download_onedrive.sh to code_download_onedrive.sh
```

Podemos comprobar la ruta donde estamos y listar los ficheros contenidos en el directorio actual:

```
!pwd
!ls
```

```
/content
code_download_onedrive.sh  sample_data
```

Le damos permisos de ejecución al script, y descargamos el código de la práctica:

```
!chmod 755 code_download_onedrive.sh
!./code_download_onedrive.sh
```

```
--2022-04-22 14:37:04--  https://dauam-my.sharepoint.com/:u:/g/personal/alicia_lozanc
Resolving dauam-my.sharepoint.com (dauam-my.sharepoint.com)... 40.108.241.27
Connecting to dauam-my.sharepoint.com (dauam-my.sharepoint.com)|40.108.241.27|:443..
HTTP request sent, awaiting response... 302 Found
Location: /personal/alicia_lozano_uam_es/Documents/APPSA/PR2/APPSA_PR2.tar.gz?ga=1 [f
--2022-04-22 14:37:04--  https://dauam-my.sharepoint.com/personal/alicia_lozano_uam_e
Reusing existing connection to dauam-my.sharepoint.com:443.
HTTP request sent, awaiting response... 200 OK
Length: 47310444 (45M) [application/x-gzip]
Saving to: ‘EeymgCUQ2EhPrAvEWf6SQm8By5753NkZHmvmPMslib0A3A?download=1’

EeymgCUQ2EhPrAvEWf6 100%[===================>]  45.12M  61.8MB/s    in 0.7s

2022-04-22 14:37:05 (61.8 MB/s) - ‘EeymgCUQ2EhPrAvEWf6SQm8By5753NkZHmvmPMslib0A3A?dow
```

Nos movemos al directorio APPSA_PR2. Podemos comprobar los requisitos software que necesitaremos en el fichero **requirements.txt**. La mayoría ya están integrados en Google Colab.

```
%cd APPSA_PR2
```

```
/content/APPSA_PR2
```

```
!cat requirements.txt
```

```
torch>=1.5.0
torchaudio>=0.5.0
numpy
scipy
scikit-learn
tqdm
pyyaml
```

Comprobamos que podemos importar todos los módulos mencionados como requisitos para usar este toolbox.

```
import torch
import torchaudio
import numpy
import scipy
import sklearn
import tqdm
import yaml
```

Podemos comprobar las versiones de los módulos de la siguiente manera:

```
print(torch.__version__)
print(torchaudio.__version__)
```

```
1.10.0+cu111
0.10.0+cu111
```

1.2. Descarga y descripción de los datos

Ahora vamos a descargar los conjuntos de datos preparados para la práctica desde sus enlaces de One Drive.

Para descargarlos y descomprimirlos en el directorio que esperan los scripts de entrenamiento, podemos utilizar el script **data_download_onedrive.sh** (previamente descargarlo de Moodle).

```
from google.colab import files
uploaded = files.upload()
```

```
Elegir archivos data_downl…nedrive.sh
• data_download_onedrive.sh(text/x-sh) - 812 bytes, last modified: 7/4/2022 - 100% done
Saving data_download_onedrive.sh to data_download_onedrive.sh
```

```
!chmod 755 data_download_onedrive.sh
!./data_download_onedrive.sh
```

```
--2022-04-22 14:39:20--  https://dauam-my.sharepoint.com/:u:/g/personal/alicia_lozanc
Resolving dauam-my.sharepoint.com (dauam-my.sharepoint.com)... 13.107.136.9, 13.107.1
Connecting to dauam-my.sharepoint.com (dauam-my.sharepoint.com)|13.107.136.9|:443...
HTTP request sent, awaiting response... 302 Found
Location: /personal/alicia_lozano_uam_es/Documents/APPSA/PR2/wav_voxceleb1_test.tar.g
--2022-04-22 14:39:20--  https://dauam-my.sharepoint.com/personal/alicia_lozano_uam_e
Reusing existing connection to dauam-my.sharepoint.com:443.
HTTP request sent, awaiting response... 200 OK
Length: 1039131372 (991M) [application/x-gzip]
Saving to: 'ER9-4DcM62lLtyqMkcU0ZAwBGHMFaSrPZgzuLq0bAxLB-Q?download=1'

ER9-4DcM62lLtyqMkcU 100%[===================>] 990.99M  79.9MB/s    in 12s

2022-04-22 14:39:33 (79.7 MB/s) - 'ER9-4DcM62lLtyqMkcU0ZAwBGHMFaSrPZgzuLq0bAxLB-Q?dow

--2022-04-22 14:39:51--  https://dauam-my.sharepoint.com/:u:/g/personal/alicia_lozanc
Resolving dauam-my.sharepoint.com (dauam-my.sharepoint.com)... 13.107.136.9, 13.107.1
Connecting to dauam-my.sharepoint.com (dauam-my.sharepoint.com)|13.107.136.9|:443...
HTTP request sent, awaiting response... 302 Found
Location: /personal/alicia_lozano_uam_es/Documents/APPSA/PR2/wav_voxceleb2_dev_subset
--2022-04-22 14:39:51--  https://dauam-my.sharepoint.com/personal/alicia_lozano_uam_e
Reusing existing connection to dauam-my.sharepoint.com:443.
HTTP request sent, awaiting response... 200 OK
Length: 2318204601 (2.2G) [application/x-gzip]
Saving to: 'EQkvCJ901zxApjlVbbSeyooBfa7it0iQP2Eie1MIsy3V-Q?download=1'

EQkvCJ901zxApjlVbbS 100%[===================>]   2.16G  90.5MB/s    in 25s

2022-04-22 14:40:17 (88.3 MB/s) - 'EQkvCJ901zxApjlVbbSeyooBfa7it0iQP2Eie1MIsy3V-Q?dow
```

```
!ls data
```

```
voxceleb1  voxceleb1.tar.gz  voxceleb2  voxceleb2.tar.gz
```

```
!ls data/voxceleb1/*/* | head
```

```
data/voxceleb1/id10270/5r0dWxy17C8:
00001.wav
00002.wav
00003.wav
00004.wav
00005.wav
00006.wav
00007.wav
00008.wav
00009.wav
```

Como vemos, el script de descarga de datos los ha preparado en la carpeta **data**. Dentro de la misma tenemos:

- *voxceleb1*: ficheros de audio WAV correspondientes al conjunto de evaluación (verificación).

- *voxceleb2*: ficheros de audio WAV correspondientes al conjunto de entrenamiento. Es un subconjunto del conjunto original VoxCeleb2 dev.

Además, tenemos las listas de ficheros que utilizaremos para entrenamiento y evaluación (test) en el directorio **lists**. Dichas listas tienen la siguiente forma:

- *train_list.txt*: SPKID WAVFILE
- *veri_test.txt*: KEY(0 - non target, 1 - target) SPKMODEL_FILE TEST_FILE

Podemos ver unas cuantas líneas de cada fichero:

```
!head lists/train_list.txt
```

```
id00012 id00012/21Uxsk56VDQ/00001.wav
id00012 id00012/Z-G8-wqpxwU/00101.wav
id00015 id00015/8JaRU2CyG8A/00037.wav
id00015 id00015/GUkQMD_WvOw/00153.wav
id00015 id00015/MRDbSqDD6ZI/00237.wav
id00015 id00015/fd_mtL88o1k/00337.wav
id00015 id00015/r7bSdlWy014/00437.wav
id00016 id00016/DxXD9zkX0ZI/00037.wav
id00016 id00016/oziSBLYisZk/00137.wav
id00018 id00018/FBwCS2s6NIw/00058.wav
```

```
!head lists/veri_test.txt
```

```
1 id10270/x6uYqmx31kE/00001.wav id10270/8jEAjG6SegY/00008.wav
0 id10270/x6uYqmx31kE/00001.wav id10300/ize_eiCFEg0/00003.wav
1 id10270/x6uYqmx31kE/00001.wav id10270/GWXujl-xAVM/00017.wav
0 id10270/x6uYqmx31kE/00001.wav id10273/0OCW1HUxZyg/00001.wav
1 id10270/x6uYqmx31kE/00001.wav id10270/8jEAjG6SegY/00022.wav
0 id10270/x6uYqmx31kE/00001.wav id10284/Uzxv7Axh3Z8/00001.wav
1 id10270/x6uYqmx31kE/00001.wav id10270/GWXujl-xAVM/00033.wav
0 id10270/x6uYqmx31kE/00001.wav id10284/7yx9A0yzLYk/00029.wav
1 id10270/x6uYqmx31kE/00002.wav id10270/5r0dWxy17C8/00026.wav
0 id10270/x6uYqmx31kE/00002.wav id10285/m-uILToQ9ss/00009.wav
```

1.3. Descripción del sistema

El sistema utilizado en esta práctica consiste de un extractor de embeddings (x-vectors), cuyo objetivo es extraer una representación del fichero de audio de longitud fija que contenga información distriminativa para la tarea de reconocimiento de locutor. Este extractor de x-vectors puede diseñarse con diferentes arquitecturas de redes neuronales y se puede entrenar con diferentes funciones de coste, así como diferentes parámetros de configuración.

Los scripts de entrenamiento del extractor de x-vectors principales son:

- *trainSpeakerNet.py*: es el script principal para entrenar o evaluar (un modelo previamente entrenado) el extractor de x-vectors. Recibe diferentes valores para sus argumentos con los que se determina el comportamiento del script.

- *models/**: en esta carpeta se encuentra la definición de diferentes arquitecturas que se pueden emplear como extractores de x-vectors.
- *loss/**: este directorio contiene la definición de diferentes funciones de coste que se pueden utilizar como funciones objetivo durante el entrenamiento del modelo.

Además, para realizar un análisis del rendimiento de los diferentes modelos, hemos incluido el siguiente script:

- *plot_score_histograms.py*: este script genera los histogramas de puntuaciones target y non-target de un modelo previamente evaluado.

2. Identificación de líneas claves de código

Ya que como se ha mencionado previamente, el script **trainSpeakerNet.py** es el principal, vamos explorar el código de dicho script para entender el funcionamiento del sistema.

PREGUNTAS: Incluye las respuestas a estas preguntas en la memoria de la práctica:

- ¿Qué línea de código se corresponde con el entrenamiento del modelo?
- ¿Qué comando carga un modelo previamente entrenado?
- ¿Qué código evalúa el rendimiento de la red neuronal?
- ¿Qué variable contiene las puntuaciones (scores) de la lista de trials? ¿En qué fichero se guardarán dichas puntuaciones?
- ¿Qué argumento controla el número de épocas (iteraciones) que se entrena el modelo?
- ¿Qué parámetro determina la función de coste?
- ¿Para qué se usa el parámetro *test_interval*?
- ¿Qué argumento necesitarías modificar para cambiar el tipo (arquitectura) del modelo que quieres entrenar?
- ¿Para qué se utiliza el parámetro *nClasses*? ¿Cómo obtendrías este valor a partir de tu conjunto de datos? Escribe el comando que utilizarías (por ejemplo, en Linux).

3. Evaluación de modelos preentrenados

En el directorio **pretrained_models/** podemos encontrar dos modelos previamente entrenados con la base de datos VoxCeleb2 dev (partición de desarrollo).

Para evaluar cada uno de estos modelos previamente entrenados, podemos utilizar los siguientes comandos:

```
!python ./trainSpeakerNet.py --eval --model ResNetSE34L --log_input True \
         --trainfunc angleproto --save_path exps/test_lite --eval_frames 400 \
         --test_list lists/veri_test.txt \
         --initial_model pretrained_models/baseline_lite_ap.model
```

```
      Embedding size is 512, encoder SAP.
      Initialised AngleProto
      Initialised Adam optimizer
      Initialised step LR scheduler
      Model has 1437080 parameters
      Model pretrained_models/baseline_lite_ap.model loaded!
      Reading 4700 of 4715: 3.04 Hz, embedding size 512
      Computing 37700 of 37720: 2270.38 Hz

      EER 2.3489
```

```
!python ./trainSpeakerNet.py --eval --model ResNetSE34V2 --log_input True \
  --encoder_type ASP --n_mels 64 --trainfunc softmaxproto \
  --save_path exps/test_v2 --eval_frames 400 --test_list lists/veri_test.txt \
  --initial_model pretrained_models/baseline_v2_ap.model
```

```
      Embedding size is 512, encoder ASP.
      Initialised Softmax Loss
      Initialised AngleProto
      Initialised SoftmaxPrototypical Loss
      Initialised Adam optimizer
      Initialised step LR scheduler
      Model has 11103416 parameters
      Model pretrained_models/baseline_v2_ap.model loaded!
      Reading 4700 of 4715: 4.29 Hz, embedding size 512
      Computing 37700 of 37720: 2187.15 Hz

      EER 1.7020
```

PREGUNTAS: Incluye en la memoria de la práctica las respuestas a las siguientes preguntas:

- ¿Cuál es el rendimiento de cada uno de los modelos?
- ¿Cuál es mejor? ¿Por qué?
- ¿Qué es el EER? ¿Cómo podemos interpretarlo?
- ¿Cuáles son las diferencias entre ambos modelos?

4. Entrenamiento de un modelo

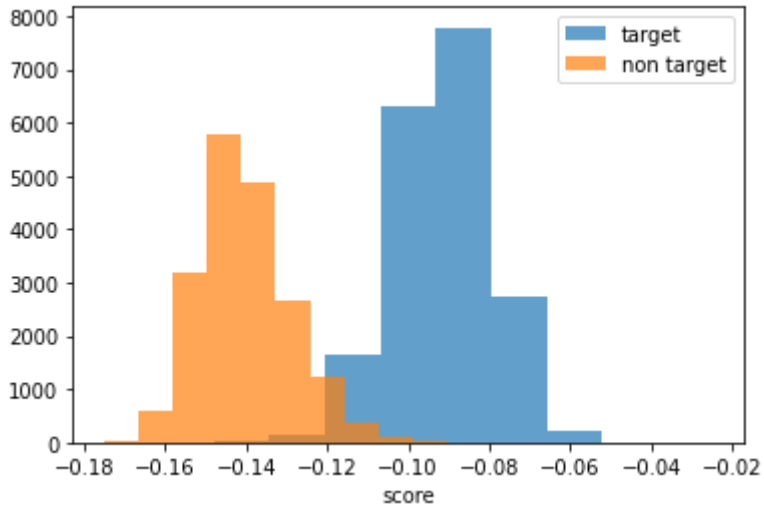
Utilizando el script **trainSpeakerNet.py**, entrena dos extractores de embeddings (x-vectors) a partir del subconjunto dado en **data/voxceleb2**. Ten en cuenta que el entrenamiento puede llevar bastante tiempo dependiendo del hardware disponible. Puedes parar el entrenamiento fijando un número máximo de iteraciones (épocas).

4.1. Entrenamiento de un modelo con una configuración fija

Entrena el siguiente modelo: ResNetSE34L, SAP encoder, tamaño de batch de 200, función de coste amsoftmax (con escalado 30 y margen 0.3). También necesitarás el número de locutores que contiene el subconjunto de entrenamiento dado para especificar el número de clases, así como las rutas a los conjuntos y listas de entrenamiento y evaluación.

Una vez que el entrenamiento ha finalizado, analiza los resultados sobre el conjunto de evaluación **data/voxceleb1**, representando los histogramas de scores correspondientes (**plot_score_histograms.py**).

```
# plot_score_histograms.py
import numpy as np
import matplotlib.pyplot as plt

scores_file = 'exps/test_lite/result/scores_test.txt' # Modify to point at your scores fil
keys_file = 'lists/veri_test.txt'

all_scores = np.loadtxt(scores_file, usecols=[0])
all_trials_scores = np.loadtxt(scores_file, usecols=[1,2], dtype='str')

all_keys = np.loadtxt(keys_file, usecols=[0])
all_trials_keys = np.loadtxt(keys_file, usecols=[1,2], dtype='str')


target_scores = all_scores[all_keys==1]
nontarget_scores = all_scores[all_keys==0]


plt.hist(target_scores, alpha=0.7)
plt.hist(nontarget_scores, alpha=0.7)

plt.xlabel('score')
plt.legend(('target','non target'))
plt.show()
```



PREGUNTAS: Responde a las siguientes preguntas:

- ¿Cuántas épocas has usado?
- ¿Cuál es el rendimiento del modelo?
- ¿Cómo es este rendimiento en comparación con el analizado en el apartado anterior de los modelos preentrenados? ¿Por qué piensas que ocurre esto?

- Incluye los histogramas de tu sistema en la memoria de la práctica. ¿Qué umbral elegirías (valor aproximado) para clasificar entre target y non-target trials? ¿Por qué?

4.2. OPCIONAL: Entrenamiento de un modelo a tu elección

Ahora, selecciona un modelo de aquellos definidos en **models/** así como una función de coste de las disponibles en la carpeta **loss/**. Los argumentos que necesitarás para entrenar dicho modelo dependerán de la configuración elegida.

PREGUNTAS: Contesta las siguientes preguntas en la memoria de la práctica:

- ¿Qué comando (modelo, parámetros, etc.) has utilizado?
- ¿Cuál es el rendimiento del sistema?
- ¿Cómo es el rendimiento de este sistema con respecto al resto de modelos de la práctica? ¿Por qué?

Por último, entrena este mismo modelo pero en una configuración con menos parámetros (modelo más pequeño) y evalúalo.

PREGUNTAS: Incluye las respuestas a estas preguntas en tu informe de la práctica:

- ¿Cómo es el rendimiento de este modelo pequeño con respecto al grande? ¿Por qué?
- Dibuja el histograma de scores para ambos modelos (pequeño y grande).

✓ 0 s completado a las 17:26