Aprendizaje Profundo para Procesamiento de Señales de Audio

Práctica 1 – Detección de eventos de audio en DCASE

En esta práctica se ofrece una introducción a la tarea de detección y clasificación de eventos acústicos, cuyo objetivo es encontrar los límites temporales de determinados eventos (por ejemplo: "voz hablada", "perro") a lo largo de señales de audio.

Los eventos acústicos son aquellas señales de audio que pueden asociarse directamente a acciones que ocurren en el entorno, como por ejemplo el sonido de una persona que habla o un teléfono sonando. Según la acción con la que estén relacionados, los eventos de audio pertenecen a distintas categorías ("Speech", "Telephone ringing"). El conjunto de categorías a detectar será, en general, diferente para cada aplicación. En esta práctica se utilizará un conjunto de diez categorías que describen eventos encontrados habitualmente en entornos domésticos.

La práctica consistirá en **dos sesiones de laboratorio**, durante las cuales **cada estudiante escribirá una memoria** contestando a las preguntas formuladas en este enunciado. **La memoria será entregada tras la última sesión y antes del comienzo de la siguiente práctica.**

1. Preparación del entorno.

a) Entorno software.

El software necesario puede encontrarse en el fichero `appsa_pr1_software.zip`, disponible en Moodle o siguiendo las instrucciones del profesor. Extraiga los contenidos en una carpeta.

El script de bash contiene las instrucciones para crear el entorno de Anaconda que será utilizado en esta práctica. Una vez creado dicho entorno, ha de ejecutarse en la terminal el commando `conda activate dcase2019` para activarlo. Con el entorno creado y activado, el programa Spyder puede ejecutarse mediante el comando `spyder`.

b) Descripción de la base de datos.

Los datos utilizados en la práctica pueden ser encontrados en el fichero `appsa_pr1_dataset.zip`, repartidos en las siguientes carpetas:

- *audio:* Ficheros WAV del set de validación.
- *features:* Mel-espectrogramas de los ficheros de audio
- *metadata:* Ficheros TSV con las anotaciones 'ground truth'.

La base de datos se divide en varios subconjuntos:

- *Train/unlabeled* (entrenamiento no etiquetado).

- *Train/weak* (entrenamiento con etiquetado débil).
- *Train/synthetic* (entrenamiento con datos sintéticos).
- *Validation* (validación).

2. Representación de los segmentos de audio y las anotaciones.

a) Formas de onda y características de audio

Elija un fichero de audio cualquiera en la carpeta `dataset/audio/validation/`. Escúchelo utilizando auriculares. Represente su forma de onda utilizando el módulo de Python incluido en el material, `appsa_pr1.py`.

- ¿Cuál es la frecuencia de muestreo (f_s)?
- ¿Cuál es la duración del fichero de audio en segundos?
- Incluya la figura obtenida en su memoria.

Represente el mel-espectrograma del fichero de audio usando `appsa_pr1.py`.

- ¿Cuántas ventanas temporales (frames) contiene dicha representación?
- ¿Cuántos componentes frecuenciales (filtros mel) se representan en el mel-espectrograma obtenido?
- Incluya la figura obtenida en su memoria.
- Compare el tamaño de la forma de onda con el del mel-espectrograma, en términos del número de muestras presentes en cada representación.

b) Anotaciones de eventos

Cargue las anotaciones (etiquetas) de eventos del set de validación (`dataset/metadata/validation/validation.tsv`) en Spyder o utilizando un editor de textos y busque las etiquetas correspondientes al fichero de audio previamente representado.

- ¿Qué categorías de eventos están presentes en el segmento de audio? ¿En qué instante temporal empieza (onset) y termina (offset) cada una de ellas?

Represente las etiquetas en Python utilizando el material provisto.

- ¿Existe solapamiento de eventos durante el segmento de audio?
- Incluya la figura obtenida en su memoria.
- Compare la forma de onda y el mel-espectrograma con las anotaciones de eventos, y trate de asociar los eventos de audio anotados con distintas partes de dichas representaciones.

3. Detección de eventos acústicos con un modelo pre-entrenado.

a) Resultados y métricas

Abra la terminal y ejecute el script de Python `TestModel.py` para generar predicciones del set de validación con el modelo pre-entrenado.

```
python TestModel.py --model_path=pretrained_model.p
```

Se mostrarán los resultados en la línea de comandos, organizados en las siguientes categorías:

- **Event based metrics (onset-offset):**
  - Overall metrics (micro-average)
  - **Class-wise average metrics (macro-average)**
  - **Class-wise metrics**
- Segment based metrics:
  - Overall metrics (micro-average)
  - Class-wise average metrics (macro-average)
  - Class-wise metrics
- Weak F_1-score per class
- Weak F_1-score macro averaged

Durante la práctica solamente se tendrán en cuenta las **secciones resaltadas**, que se corresponden con métricas basadas en eventos (event-based).

El F_1-score es la métrica principal para evaluación de sistemas de detección de eventos acústicos. Se obtiene como función del número de aciertos positivos (True Positive, TP), falsos negativos (False Negative, FN) y falsos positivos (False Positive, FP) en cada categoría:

$$\boldsymbol{F_1 = \frac{2 \times TP}{2 \times TP + FP + FN}}$$

La puntuación F_1 se suele expresar como un porcentaje, de modo que el máximo posible sería 100% (en un caso ideal en el que todos los eventos se detectan de forma correcta sin falsos positivos) y el mínimo sería 0% (cuando ningún evento se detecta de forma correcta).

El rendimiento global del sistema se calcula como el promedio (macro-average) de los F_1 de cada categoría.

- Incluya en su memoria los F_1 basados en eventos que se obtienen en cada categoría y el F_1 promedio (macro-average) obtenido por el modelo pre-entrenado.

b) Predicciones

De manera adicional, el script ejecutado en el apartado anterior guardará las predicciones de eventos generadas por el modelo en un fichero TSV con el nombre `validation2019_predictions.tsv`.

- Utilice el módulo `appsa_pr1.py` para representar las predicciones del modelo pre-entrenado correspondientes al segmento de audio elegido en los apartados previos, e incluya la figura obtenida en su memoria.

- Comente el ajuste de las predicciones generadas al etiquetado 'ground truth'.

- Repita el proceso de los dos puntos anteriores para dos segmentos de audio adicionales en los que aparezcan categorías de eventos diferentes.