audias
Audio, Data Intelligence and Speech

UAM Universidad Autónoma de Madrid

Práctica 1: Detección de eventos de audio en DCASE

Aprendizaje Profundo para Procesamiento de Señales de Audio

Diego de Benito (diego.benito@uam.es)
AUDIAS Research Group
Universidad Autónoma de Madrid
http://audias.ii.uam.es

DCASE

“Detection and Classification of Acoustic Scenes and Events”

Evaluación competitiva y workshop a escala internacional

- Task 1: Acoustic Scene Classification
- Task 2: Unsupervised Detection of Anomalous Sounds
- Task 3: Sound Event Localization and Detection
- …

http://dcase.community/challenge2020

DCASE

"Detection and Classification of Acoustic Scenes and Events"

Evaluación competitiva y workshop a escala internacional

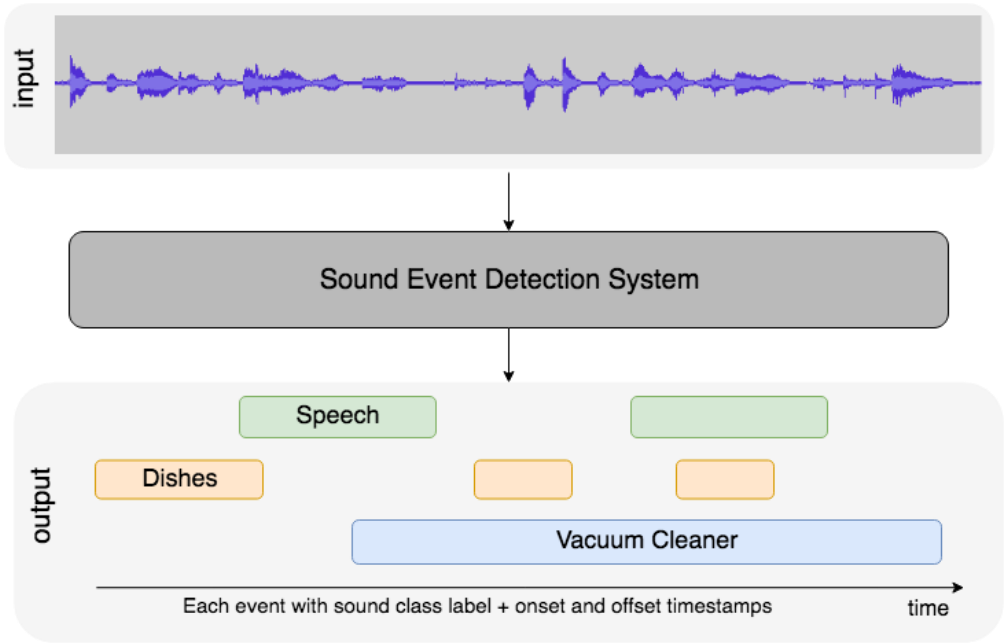
- …
- **Task 4: Sound Event Detection in Domestic Environments**
- Task 5: Urban Sound Tagging with Spatiotemporal Context
- Task 6: Automated Audio Captioning
- *… Diferentes tareas cada año*

http://dcase.community/challenge2020

Detección de eventos de audio

- Determinar la localización temporal de los eventos en una grabación de audio
  - p.ej.: “Speech” desde 3,4 segundos hasta 7,2 segundos

- Set cerrado de categorías
  - P. ej.: “Speech”, “Dog”, “Vacuum cleaner”…



Source: DCASE

Detección de eventos de audio

- Dataset y etiquetado
  - Etiquetado fuerte (strong labels): categoría + marcas de tiempo
  - Etiquetado débil (weak labels): solo categoría (sin marcas de tiempo)
  - Datos sin etiquetar (unlabeled data)
- Input features (Representación de los audios)
  - P. ej.: Forma de onda, espectrograma, melgrama
- Sistema de detección de eventos de audio
  - P. ej.: Redes convolucionales (CNNs)
  - Aprendizaje semi-supervisado

Ejemplo en YouTube

"Y0bO6DhC2tlU_110.000_120.000.wav"

↓

http://youtube.com/watch?v=0bO6DhC2tlU&t=110

validation.tsv → Y0bO6DhC2tlU_110.000_120.000.wav

onset	offset	event
0.000	10.000	Frying
0.671	1.116	Dishes
2.114	3.159	Dishes
3.557	5.048	Dishes
6.075	6.574	Dishes
7.476	7.797	Dishes
7.690	8.563	Speech
8.420	10.000	Dishes

Índice

Dos sesiones: 24 y 31 de marzo, 16h – 17h

- Preparación del entorno
  - Software / Dataset
- Representaciones de audio y etiquetas
  - Formas de onda / Melgramas / Etiquetado de eventos
- Métricas y resultados de detección de eventos acústicos

Evaluación

- Se elaborará una memoria respondiendo a las preguntas del enunciado
  - Word / LaTex

- Entrega:
  - Documento **PDF** a través de **Moodle**
  - Desde el 31 de marzo hasta el **7 de abril**

Desarrollo de la práctica

Primera sesión (24 de marzo)

Preparación del entorno

Material disponible en Moodle

- Dataset
- Código

Instalación del entorno conda:

Ubuntu:

```
conda env create -f environment.yml
conda activate dcase2019
chmod 755 install_pip_requirements.sh
./install_pip_requirements.sh
```

Windows:

```
conda env create -f environment_windows.yml
conda activate dcase2019win
conda install spyder
pip install torch==1.7.1+cpu torchvision==0.8.2+cpu torchaudio==0.7.2 -f https://download.pytorch.org/whl/torch_stable.html
```

Dataset

- DESED (Domestic Environment Sound Event Detection)
  - Segmentos de audio de Google AudioSet
    - Extraídos de vídeos de YouTube
    - Conjuntos ‘Train/weak’, ‘Train/unlabeled’, ‘Validation’
  - Segmentos de audio sintéticos
    - Conjunto ‘Train/synthetic’

- 10 categorías de eventos
  - Speech, Dog, Cat, Alarm/bell/ringing, Dishes, Frying, Blender, Running water, Vacuum cleaner, Electric shaver/toothbrush

Código

- Python + Spyder IDE
- Basado en el sistema baseline de DCASE 2019 Task 4 (PyTorch)

- Funciones auxiliares: appsa_pr1.py
- Script de ejemplo: appsa_pr1_figures.py

Enlaces y bibliografía (1ª sesión)

- DCASE 2019 Task 4: http://dcase.community/challenge2019/task-sound-event-detection-in-domestic-environments
- DCASE 2020 Task 4: http://dcase.community/challenge2020/task-sound-event-detection-and-separation-in-domestic-environments
- DESED Dataset: https://project.inria.fr/desed/
- Google AudioSet: https://research.google.com/audioset/

Desarrollo de la práctica

Segunda sesión (31 de marzo)

Detección de eventos con un modelo pre-entrenado

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

Resultados:

- Event-based metrics (onset-offset)
  - Overall metrics (micro-average)
  - Class-wise average metrics (macro-average)
  - Class-wise metrics
- Segment-based metrics
  - …
- Weak F1-scores

Resultados y métricas

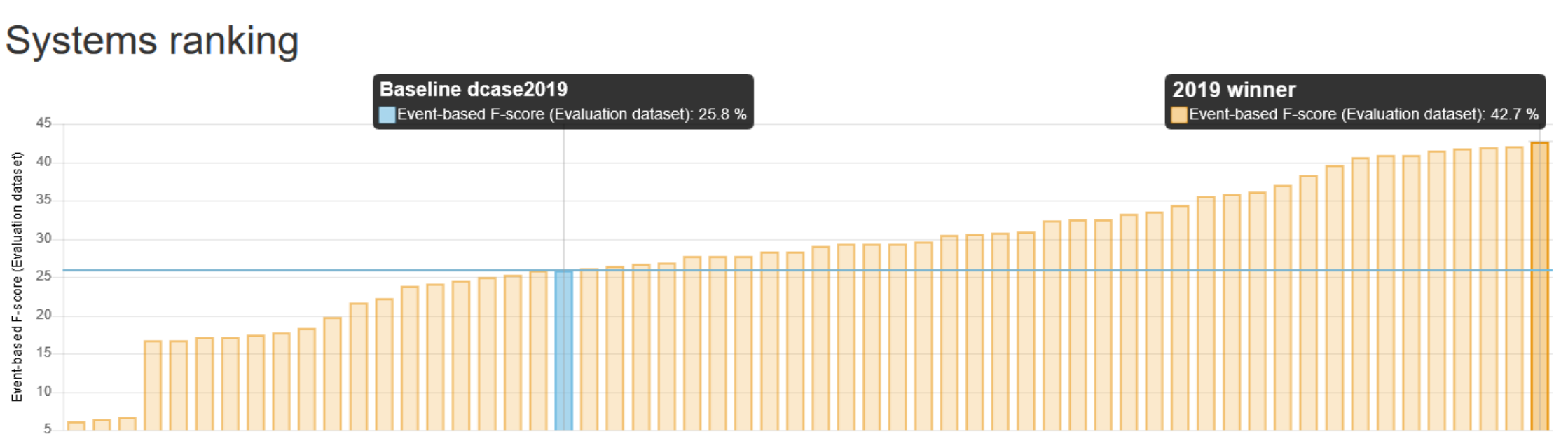
F1-score: función del número de aciertos positivos (True Positive, TP), falsos negativos (False Negative, FN) y falsos positivos (False Positive, FP):

$$F_1(\%) = 100 \times \frac{2 \times TP}{2 \times TP + FP + FN}$$

Rendimiento global del sistema: promedio de los F1-scores de cada categoría

Resultados del sistema baseline

- Resultados de la evaluación DCASE 2019 Task 4



Enlaces y bibliografía (2ª sesión)

- Annamaria Mesaros, Toni Heittola, Tuomas Virtanen. "*Metrics for Polyphonic Sound Event Detection.*" Applied Sciences, 2016. https://doi.org/10.3390/app6060162

- Nicolas Turpault, Romain Serizel, Ankit Parag Shah, Justin Salamon. "*Sound event detection in domestic environments with weakly labeled data and soundscape synthesis.*" Workshop on Detection and Classification of Acoustic Scenes and Events, Oct 2019, New York City, United States. https://hal.inria.fr/hal-02160855v2

- Antti Tarvainen, Harri Valpola. *"Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results."* Advances in neural information processing systems, 2017. https://arxiv.org/abs/1703.01780

Material adicional

Entrenamiento de un sistema de detección de eventos acústicos

Training a SED system

Two options to run training:

a) `python main.py`

- Output is printed to the command line

b) `nohup Python main.py > nohup.out &`

- Output is saved to text file nohup.out
- `tail -f nohup.out`

Tune parameters in `config.py`

Training a SED system

Observe the code of the following scripts:

- `main.py`
  - Try to locate the parts where:
    - The dataset is loaded
    - The model is trained
- `config.py`
  - Locate the part where the structure of the neural network is defined

Training a SED system: Mean Teacher

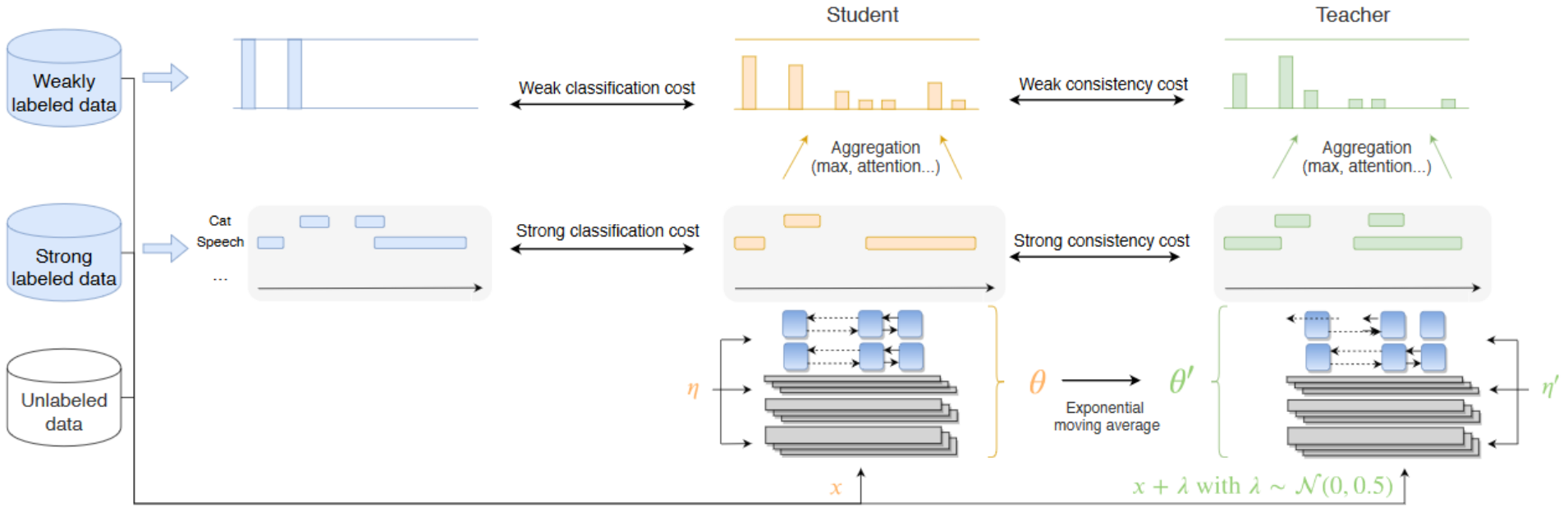


Figure 1: Mean-teacher model. η and η' represent noise applied to the different models (in this case dropout).

N. Turpault et al. "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis". Workshop on Detection and Classification of Acoustic Scenes and Events, Oct 2019, New York