A Temporal Approach To Anomalous Sound Detection

Ashlae Blum
GEM Internship Final Presentation
09 August 2021
Introduction to Unsupervised Anomalous Sound Detection

Unsupervised Deep Learning
• Neural network feeds data through a series of layers (DL)
• Algorithm trains on unlabeled data (UL)

Anomalous Sound Detection (ASD)
• Determines whether a sound is ‘normal’ or ‘anomalous’

Predictive Maintenance (PM)
• Uses anomaly detection to predict abnormal behavior
• Event classification
  - Single incidents
• Scene classification
  - Changes in quality of sound
Pros & Cons of Unsupervised Anomalous Sound Detection

**Advantages**

- Unsupervised models require less data preparation
- Can deploy on edge devices
  - Hard to reach places
  - Intolerable environmental conditions

**Disadvantages**

- Accuracy tradeoff
- Low-power applications
  - Complexity
- Real-time processing
  - Latency
Superconducting Magnet Quench

- High-energy superconductors require cryogenic temp
- Coils become supercharged up to 14.5 TeV
- As energy increases instability does too

**QUENCH!!!**

Physical imperfections and complex dynamics cause energy saturation

- Same phenomenon as seismic activity

The quench is audible

- Acoustic sensors

Magnets trained by gradually raising the current

>> Goal: edge deployment using TinyML
MIR Approach Using DCASE2020 Task2 Challenge

DCASE2020
- International ASD challenge
- Created a benchmark for testing in 2020
- Vanilla autoencoder

Methods
- Librosa package for Python
- MIR approach to explore low-level features
  - Spectral
    - MFCC, Chroma, Spectrogram, STFT, Tonnetz
  - Temporal
    - Onset detection, Beat tracking, Tempo, PLP
Neural Network Architecture

Trains on MFCC spectral data
7000 audio files, each approx 10s long
Sliding window of 5 frames, 128-mel bins, hop length of 512, 100 epochs

Image taken from Jules Muhizi’s Presentation: “Sparse and Low Bit Precision Networks For Hardware Deployment”
Neural Network Architecture

Autoencoder uses 9-layer dense network

4x128, 1x8, 4x128 node architecture with 270k parameters

ReLU and BatchNorm after each hidden layer

Adam optimizer with mean-squared-error loss

```python
max_fpr : 0.1 # false positive rate

feature:
    n_mels: 128
    frames : 5
    n_fft: 1024
    hop_length: 512
    power: 2.0

fit:
    compile:
        optimizer : adam
        loss : mean_squared_error
        epochs : 100
        batch_size : 512
        shuffle : True
    validation_split : 0.1
    verbose : 1
```
Anomaly Score Methods & Results

Training is done with the Toy Car data from ToyADMOS dataset

Autoencoder determines Anomaly Scores for audio files
Anomaly Score Methods & Results

Receiver Operator Characteristic (ROC) Curve

- Performance based on threshold

Area Under Curve (AUC) and Partial AUC (pAUC)

- Measure accuracy
- AUC=0.5 means the classifier can’t distinguish, AUC=1.0 means it’s perfect

**Average AUC: 0.778**
Spectral Features Methods

Waveform
- Plot of the audio signal

Short Time Fourier Transform (STFT)
- Fourier transform to determine local phase and frequency

Mel Frequency Cepstral Coefficients (MFCC)
- Amplitude of frequencies in a signal, non-linearly scaled to human hearing

Spectrogram
- Power of signal at various frequencies
Temporal Features Methods

Onset strength

Determines a thresholded of spectral energy in a spectrogram, 1D array represents increasing energy at each frame.

Onset detection

Selects peak positions from onset strength curve
Temporal Features Methods

Tempo

Average global tempo (bpm) is estimated using beat tracking and onset correlation.

Predominant Local Pulse (PLP)

Correlates local maxima of onset peaks local tempo estimates:

1. get the onset envelope
2. get the fourier tempogram
3. pin to the feasible tempo trange
4. discard everything below the peak and normalize
5. invert the Fourier tempogram to get pulse
6. retain only the positive part of the pulse cycle
7. return the normalized pulse
The main idea for this research is to compare the tempo data across all audio files, as well as the plp across all audio files to see if correlations emerge.

We define an approach to analyzing temporal information:

- Tempo
- Predominant Local Pulse

Construct a dataset similar to MFCC datasets

Plot Onset, Tempo, and PLP information
Conclusions

!! Size of the temporal data is NON-UNIFORM !!

• Different audio files have different numbers of datapoints
  - Column lengths of data result in different sizes (lengths)
Conclusions

Need to use a different type of model to feed the data.

- Long Short Term Memory (LSTM) network
  - Can accept variably sized data
Proposal for Future Work

Develop LSTM to train the temporal audio data

Extend the R&D to other audio features
  • Tempogram, periodogram, tonnetz

Explore other methods of analysis and visualization

Correlate the occurrence of spectral and temporal features

Make this approach portable for implementation with TinyML edge devices
Thanks !

Much thanks and appreciation for support from:

Ryan Rivera
Nhan Trang
Ben Woods

Charlie Orozco
Arden Warner

Judy Nunez

Fermilab
National GEM Consortium