Listen to Interpret: Post-hoc Interpretability for Audio Networks with NMF

Jayneel Parekh

(Co-authors: Sanjeel Parekh, Pavlo Mozharovskyi, Florence d'Alché-Buc, Gaël Richard)

LTCI, Télécom Paris, IP Paris

June 15, 2022

Introduction

- **Interpretability**: To make decision process of an AI system human-understandable.

Introduction

- **Interpretability**: To make decision process of an AI system human-understandable.
- Most developed methods for interpretability applicable for image or tabular data. Do not transfer well to other modalities. For *e.g.* saliency maps over spectrograms hard to interpret for end-user.
- Require audio-specific understandability features in our interpreter. Imagine a classifier detecting an “alarm” sound event. An ideal interpreter would be able to:

Introduction

- **Interpretability**: To make decision process of an AI system human-understandable.
- Most developed methods for interpretability applicable for image or tabular data. Do not transfer well to other modalities. For *e.g.* saliency maps over spectrograms hard to interpret for end-user.
- Require audio-specific understandability features in our interpreter. Imagine a classifier detecting an “alarm” sound event. An ideal interpreter would be able to:
  - Localize the alarm event amid a host of other background events
  - Provide it as listenable audio to an end-user

Introduction

- **Interpretability**: To make decision process of an AI system human-understandable.
- Most developed methods for interpretability applicable for image or tabular data. Do not transfer well to other modalities. For *e.g.* saliency maps over spectrograms hard to interpret for end-user.
- Require audio-specific understandability features in our interpreter. Imagine a classifier detecting an "alarm" sound event. An ideal interpreter would be able to:
  - Localize the alarm event amid a host of other background events
  - Provide it as listenable audio to an end-user
- However, note that interpretation is NOT the same as classical audio source separation or denoising tasks!

Problem formulation

- **Post-hoc interpretation** problem for **audio processing networks**
  - We are provided with a fixed model f whose decisions we wish to interpret
  - f is a deep neural network that processes audio signals

Problem formulation

- **Post-hoc interpretation** problem for **audio processing networks**
  - We are provided with a fixed model f whose decisions we wish to interpret
  - f is a deep neural network that processes audio signals
- We operate in a supervised classification setting (both *multi-class* or *multi-label* classification possible)

Problem formulation

- **Post-hoc interpretation** problem for **audio processing networks**
  - We are provided with a fixed model f whose decisions we wish to interpret
  - f is a deep neural network that processes audio signals
- We operate in a supervised classification setting (both *multi-class* or *multi-label* classification possible)
- Working under the FLINT framework, i.e., propose to learn an interpreter module / interpreter $\mathcal{I}$ (relying on hidden layers of f) by minimizing a loss function $\mathcal{L}$ s.t. we can satisfy requirements for interpretability

$$\arg\min_{V_{\mathcal{I}}} \mathcal{L}(f, \mathcal{I}, \mathcal{S})$$

System Motivation

The functions of the ideal interpreter can be described as:

1. Interpretations through high-level audio objects constituting a scene
2. Ability to identify parts of input relevant to decision.
3. Extract the identified parts as listenable audio.

System Motivation

The functions of the ideal interpreter can be described as:

1. Interpretations through high-level audio objects constituting a scene
2. Ability to identify parts of input relevant to decision.
3. Extract the identified parts as listenable audio.

Is it possible to process intermediate layers of audio network and extract representation which can serve the above functions?

System Motivation

The functions of the ideal interpreter can be described as:

1. Interpretations through high-level audio objects constituting a scene
2. Ability to identify parts of input relevant to decision.
3. Extract the identified parts as listenable audio.

Is it possible to process intermediate layers of audio network and extract representation which can serve the above functions?

Design of representations as in NMF an attractive option!

System Motivation

The functions of the ideal interpreter can be described as:

1. Interpretations through high-level audio objects constituting a scene
2. Ability to identify parts of input relevant to decision.
3. Extract the identified parts as listenable audio.

Is it possible to process intermediate layers of audio network and extract representation which can serve the above functions?

Design of representations as in NMF an attractive option!

1. Decompose input audio in spectral patterns + time activations (via a loss function).
2. Encourage approximation of classifier decision from extracted representation (via a loss function).
3. Take advantage of soft-masking and inverse STFT operations.

What is NMF?

Non-negative Matrix Factorization – popular for *unsupervised decomposition of audio signals*.

What is NMF?

Non-negative Matrix Factorization – popular for *unsupervised decomposition of audio signals*. Given positive time–frequency representation $\mathbf{X} \in \mathbb{R}_+^{F \times T}$ (F frequency bins & T time frames), NMF decomposes it as,

$$\mathbf{X} \approx \mathbf{W}\mathbf{H}, \mathbf{W} \geq 0, \mathbf{H} \geq 0$$

What is NMF?

Non-negative Matrix Factorization – popular for *unsupervised decomposition of audio signals*. Given positive time–frequency representation $\mathbf{X} \in \mathbb{R}_+^{F \times T}$ (F frequency bins & T time frames), NMF decomposes it as,

$$\mathbf{X} \approx \mathbf{WH}, \mathbf{W} \geq 0, \mathbf{H} \geq 0$$

- $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K] \in \mathbb{R}_+^{F \times K}$ is interpreted as the spectral pattern or dictionary matrix containing K components

What is NMF?

Non-negative Matrix Factorization – popular for *unsupervised decomposition of audio signals*. Given positive time–frequency representation $\mathbf{X} \in \mathbb{R}_+^{F \times T}$ (F frequency bins & T time frames), NMF decomposes it as,

$$\mathbf{X} \approx \mathbf{W}\mathbf{H}, \mathbf{W} \geq 0, \mathbf{H} \geq 0$$

- $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K] \in \mathbb{R}_+^{F \times K}$ is interpreted as the spectral pattern or dictionary matrix containing K components
- $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K]^\intercal \in \mathbb{R}_+^{K \times T}$ a matrix containing the corresponding time activations

What is NMF?

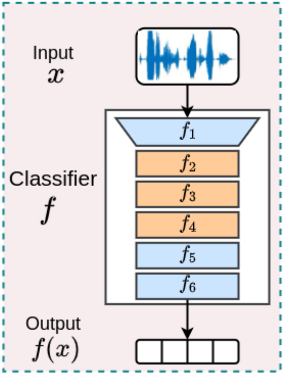
Non-negative Matrix Factorization – popular for *unsupervised decomposition of audio signals*. Given positive time–frequency representation $\mathbf{X} \in \mathbb{R}_+^{F \times T}$ (F frequency bins & T time frames), NMF decomposes it as,

$$\mathbf{X} \approx \mathbf{WH}, \mathbf{W} \geq 0, \mathbf{H} \geq 0$$

- $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K] \in \mathbb{R}_+^{F \times K}$ is interpreted as the spectral pattern or dictionary matrix containing K components
- $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K]^\intercal \in \mathbb{R}_+^{K \times T}$ a matrix containing the corresponding time activations

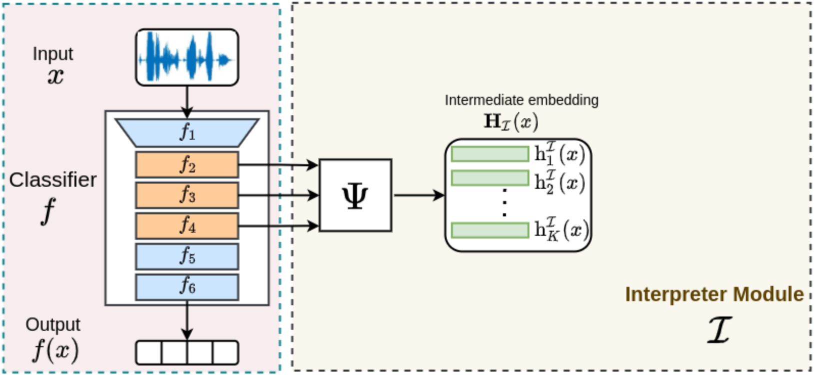
Variants of NMF-algorithm can also be used for dictionary learning on a dataset, by estimating $\mathbf{W}$ on a training dataset matrix $\mathbf{X}_{\text{train}}$.

System Overview



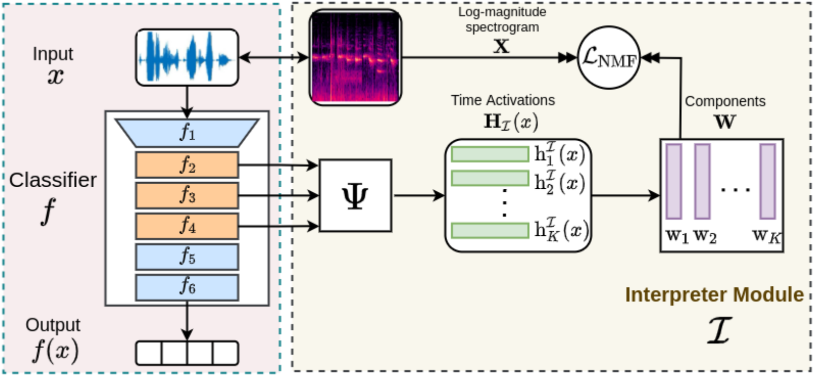
- f is the audio-processing deep network we wish to interpret.

System Overview



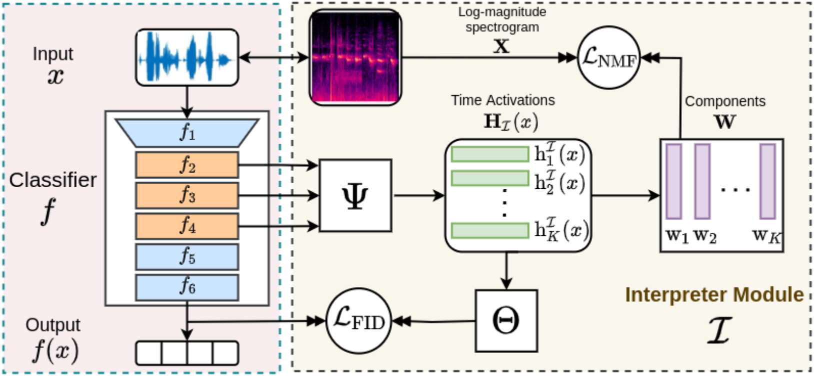
- $\Psi(f_{\mathcal{I}}(x)) \in \mathbb{R}_+^{K \times T}$ produces an intermediate encoding of the interpreter. For simplicity, denote it as $\mathbf{H}_{\mathcal{I}}(x) = \Psi \circ f_{\mathcal{I}}(x)$

System Overview



- Intermediate encoding used with dictionary **W** (learnt apriori, fixed) to reconstruct **X**. $\mathbf{H}_{\mathcal{I}}(x)$ can then be seen as time activations.

System Overview



- The interpreter computes the output $\Theta \circ \mathbf{H}_{\mathcal{I}}(x)$ and aims to mimic output of classifier $f(x)$. Shapes $\mathbf{H}_{\mathcal{I}}(x)$ to interpret classifier output.

Design of Interpreter network

Design of Ψ.

- Downsample on the frequency axis, upsample on time axis

Design of Interpreter network

Design of Ψ.

- Downsample on the frequency axis, upsample on time axis
- Axis for # of channels should transform to axis of number of components K.

Design of Interpreter network

Design of Ψ.

- Downsample on the frequency axis, upsample on time axis
- Axis for # of channels should transform to axis of number of components K.

Design of Θ.

- First pools activations $\mathbf{H}_{\mathcal{I}}(x)$ across time. Attention–based pooling (Ilse et al., 2018), $\mathbf{z} = \mathbf{H}_{\mathcal{I}}(x)\mathbf{a}$, where $\mathbf{a} \in \mathbb{R}^T$, $\mathbf{z} \in \mathbb{R}^K$.

Design of Interpreter network

Design of Ψ.

- Downsample on the frequency axis, upsample on time axis
- Axis for # of channels should transform to axis of number of components K.

Design of Θ.

- First pools activations $\mathbf{H}_{\mathcal{I}}(x)$ across time. Attention–based pooling (Ilse et al., 2018), $\mathbf{z} = \mathbf{H}_{\mathcal{I}}(x)\mathbf{a}$, where $\mathbf{a} \in \mathbb{R}^T$, $\mathbf{z} \in \mathbb{R}^K$.
- Operate on $\mathbf{z}$ with a linear layer to generate the output.

Design of Interpreter network

Design of Ψ.

- Downsample on the frequency axis, upsample on time axis
- Axis for # of channels should transform to axis of number of components K.

Design of Θ.

- First pools activations $\mathbf{H}_{\mathcal{I}}(x)$ across time. Attention–based pooling (Ilse et al., 2018), $\mathbf{z} = \mathbf{H}_{\mathcal{I}}(x)\mathbf{a}$, where $\mathbf{a} \in \mathbb{R}^T$, $\mathbf{z} \in \mathbb{R}^K$.
- Operate on $\mathbf{z}$ with a linear layer to generate the output.

Fidelity loss: To encourage $\Theta \circ \mathbf{H}_{\mathcal{I}}(x)$ to approximate $f(x)$

$$\mathcal{L}_{\mathsf{FID}}(x, V_\Psi, V_\Theta) = -f(x)^\intercal \log(\Theta(\mathbf{H}_{\mathcal{I}}(x))) \tag{1}$$

For multi-label classification,

$$\begin{aligned}\mathcal{L}_{\mathsf{FID}}(x, V_\Psi, V_\Theta) = -\sum & f(x) \odot \log(\Theta(\mathbf{H}_{\mathcal{I}}(x))) \\ &+ (1 - f(x)) \odot \log(1 - \Theta(\mathbf{H}_{\mathcal{I}}(x))).\end{aligned} \tag{2}$$

NMF dictionary decoder

Additionally constrain $\mathbf{H}_{\mathcal{I}}(x)$, such that, when fed to a decoder it is able to reconstruct the input audio.

This decoder is a pre-learnt NMF dictionary, $\mathbf{W}$, learnt via SparseNMF (Le Roux et al., 2015).

NMF dictionary decoder

Additionally constrain $\mathbf{H}_{\mathcal{I}}(x)$, such that, when fed to a decoder it is able to reconstruct the input audio.

This decoder is a pre-learnt NMF dictionary, $\mathbf{W}$, learnt via SparseNMF (Le Roux et al., 2015).

Formally, through $\mathcal{L}_{\text{NMF}}$ we require $\mathbf{H}_{\mathcal{I}}(x)$ to approximate log-magnitude spectrogram of input audio as $\mathbf{X} \approx \mathbf{W}\mathbf{H}_{\mathcal{I}}(x)$:

$$\mathcal{L}_{\text{NMF}}(x, V_{\Psi}) = \|\mathbf{X} - \mathbf{W}\mathbf{H}_{\mathcal{I}}(x)\|_2^2. \tag{3}$$

NMF dictionary decoder

Additionally constrain $\mathbf{H}_{\mathcal{I}}(x)$, such that, when fed to a decoder it is able to reconstruct the input audio.

This decoder is a pre-learnt NMF dictionary, $\mathbf{W}$, learnt via SparseNMF (Le Roux et al., 2015).

Formally, through $\mathcal{L}_{\text{NMF}}$ we require $\mathbf{H}_{\mathcal{I}}(x)$ to approximate log-magnitude spectrogram of input audio as $\mathbf{X} \approx \mathbf{W}\mathbf{H}_{\mathcal{I}}(x)$:

$$\mathcal{L}_{\text{NMF}}(x, V_{\Psi}) = \|\mathbf{X} - \mathbf{W}\mathbf{H}_{\mathcal{I}}(x)\|_2^2. \tag{3}$$

The reconstruction loss allows us to consider $\mathbf{H}_{\mathcal{I}}(x)$ as a time activation matrix for $\mathbf{W}$.

Training

Training loss. Additionally ℓ_1 regularization on $\mathbf{H}_{\mathcal{I}}(x)$ is imposed to encourage sparsity of activations. The complete training loss function:

$$\mathcal{L}(V_\Psi, V_\Theta) = \sum_{x \in \mathcal{S}} \mathcal{L}_{\mathsf{FID}}(x, V_\Psi, V_\Theta) + \alpha \mathcal{L}_{\mathsf{NMF}}(x, V_\Psi) + \beta ||\mathbf{H}_{\mathcal{I}}(x)||_1 \quad (4)$$

$\alpha, \beta \geq 0$ are loss hyperparameters.

- Parameters of $\mathcal{I}$ constituted in the functions Ψ, Θ and dictionary $\mathbf{W}$
- $\mathbf{W}$ is pre-learnt and fixed, thus $\mathcal{L}$ is optimized w.r.t V_Ψ, V_Θ.

Interpretation Algorithm

- **Step 1:** Estimate "importance" of components $r_{k,c,x} = \frac{(\mathbf{z}_k \theta^{w}_{c,k})}{\max_l |\mathbf{z}_l \theta^{w}_{c,l}|}$ using pooled activations z, weights of linear layer in Θ,

Interpretation Algorithm

- **Step 1:** Estimate "importance" of components $r_{k,c,x} = \frac{(\mathbf{z}_k \theta^w_{c,k})}{\max_l |\mathbf{z}_l \theta^w_{c,l}|}$ using pooled activations z, weights of linear layer in Θ, and select set of important components $L_{c,x} = \{k : r_{k,c,x} > \tau\}$.

Interpretation Algorithm

- **Step 1:** Estimate "importance" of components $r_{k,c,x} = \frac{(\mathbf{z}_k \theta^w_{c,k})}{\max_l |\mathbf{z}_l \theta^w_{c,l}|}$ using pooled activations z, weights of linear layer in Θ, and select set of important components $L_{c,x} = \{k : r_{k,c,x} > \tau\}$.
- **Step 2:** Extract parts of input signal captured by each relevant component and invert them back to time-domain

Interpretation Algorithm

- **Step 1:** Estimate "importance" of components $r_{k,c,x} = \frac{(\mathbf{z}_k \theta^w_{c,k})}{\max_l |\mathbf{z}_l \theta^w_{c,l}|}$ using pooled activations z, weights of linear layer in Θ, and select set of important components $L_{c,x} = \{k : r_{k,c,x} > \tau\}$.
- **Step 2:** Extract parts of input signal captured by each relevant component and invert them back to time-domain

Algorithm 2 Audio interpretation generation

```
Input: log-magnitude spectrogram X, input phase P_x
    components W = {w_1, ..., w_K}, time activations
    H_I(x) = [h_1^I(x), ..., h_K^I(x)]^T, set of selected compo-
    nents L_{c,x} = {k_1, ..., k_B}.
for all k ∈ L_{c,x} do
    X_k ← (w_k h_k^I(x)^T / Σ_{l=1}^K w_l h_l^I(x)^T) ⊙ X      {// Soft masking}
    x_k = INV(X_k, P_x)                                          {// Inverse STFT}
end for
X_int ← Σ_{k ∈ L_{c,x}} X_k
x_int = INV(X_int, P_x)
Output: {x_{k_1}, ..., x_{k_B}}, x_int
```

Experiments: Overview

Datasets

- *Multi-class classification*: Dataset for Environmental Sound Classification – **ESC-50**. 50 classes, 2000 samples (5 seconds).
- *Multi-label classification*: Sounds of New York City – Urban Sound Tagging – **SONYC-UST**. 8 classes, 14000+ samples (10 seconds). Real-world audio with high background noise, weak sources makes it very challenging.

Experiments: Overview

Datasets

- *Multi-class classification*: Dataset for Environmental Sound Classification – **ESC-50**. 50 classes, 2000 samples (5 seconds).
- *Multi-label classification*: Sounds of New York City – Urban Sound Tagging – **SONYC-UST**. 8 classes, 14000+ samples (10 seconds). Real-world audio with high background noise, weak sources makes it very challenging.

Network interpreted

- VGG-styled network pre-trained on AudioSet.
- Fine-tuned on each task before being interpreted

Experiments: Overview

Datasets

- *Multi-class classification*: Dataset for Environmental Sound Classification – **ESC-50**. 50 classes, 2000 samples (5 seconds).
- *Multi-label classification*: Sounds of New York City – Urban Sound Tagging – **SONYC-UST**. 8 classes, 14000+ samples (10 seconds). Real-world audio with high background noise, weak sources makes it very challenging.

Network interpreted

- VGG-styled network pre-trained on AudioSet.
- Fine-tuned on each task before being interpreted

Evaluation

- **Fidelity**: How well the interpreter approximates the classifier
- **Faithfulness**: Are the features captured by the interpreter *truly* important to the classifier's decision?
- **Subjective Evaluation**: Understandability of interpretations.

Evaluated Systems for Fidelity

- L2I + Θ_{ATT}: proposed Listen to Interpret (L2I) system, with attention based pooling in Θ
- L2I + Θ_{MAX}: proposed L2I system, with max pooling in Θ
- Baselines: *post-hoc* methods that approximate the classifier with a single surrogate model: FLINT & VIBI.
- The baseline methods are themselves not usable for listenable interpretations, only to quantify fidelity.

ESC-50 Fidelity

Dictionary size K: 100

top-k Fidelity for multi-class: Fraction of samples where the class predicted by f is among the top-k classes predicted by the interpreter.

	Fidelity (in %)		
System	top-1	top-3	top-5
L2I + Θ_{ATT}	65.7 $\pm$ 2.8	81.8 $\pm$ 2.2	88.2 $\pm$ 1.7
L2I + Θ_{MAX}	73.3 $\pm$ 2.3	87.8 $\pm$ 1.8	92.7 $\pm$ 1.2
FLINT	73.5 $\pm$ 2.3	89.1 $\pm$ 0.4	93.4 $\pm$ 0.9
VIBI	27.7 $\pm$ 2.3	45.4 $\pm$ 2.2	53.0 $\pm$ 1.8

Table: Top-k fidelity results on ESC-50 (5 fold mean, std)

SONYC-UST: Fidelity

Dictionary size K: 80

To compute *fidelity* on multi-label classification tasks, use Area Under Precision-Recall Curve (AUPRC) based metrics between the classifier output $f(x)$ and interpreter output $\Theta(\mathbf{H}_{\mathcal{I}}(x))$.

	Fidelity		
System	macro-AUPRC	micro-AUPRC	max-F1
L2I + Θ_{ATT}	0.900	0.914	0.847
L2I + Θ_{MAX}	0.864	0.912	0.840
FLINT	0.807	0.898	0.811
VIBI	0.608	0.575	0.549

Table: Fidelity results on SONYC-UST

Faithfulness evaluation

- One prior proposed way of computing faithfulness **simulate feature removal from the input** $\longrightarrow$ Observe change in classifier output.

Faithfulness evaluation

- One prior proposed way of computing faithfulness **simulate feature removal from the input** $\longrightarrow$ Observe change in classifier output.
- Very hard to simulate removal for "concept" based methods. However, our decomposition structure allows this possibility!

Faithfulness evaluation

- One prior proposed way of computing faithfulness **simulate feature removal from the input** $\longrightarrow$ Observe change in classifier output.
- Very hard to simulate removal for "concept" based methods. However, our decomposition structure allows this possibility!
- Given sample x with predicted class c, remove the set of relevant components $L_{c,x} = \{k : r_{k,c,x} > \tau\}$ by creating a new signal $x_2 = \mathrm{INV}(\mathbf{X}_2, \mathbf{P}_x)$, where $\mathbf{X}_2 = \mathbf{X} - \sum_{l \in L_{c,x}} \mathbf{X}_l$. Faithfulness for x:

$$\mathrm{FF}_x = f(x)_c - f(x_2)_c \tag{5}$$

Faithfulness evaluation

- One prior proposed way of computing faithfulness **simulate feature removal from the input** $\longrightarrow$ Observe change in classifier output.
- Very hard to simulate removal for "concept" based methods. However, our decomposition structure allows this possibility!
- Given sample x with predicted class c, remove the set of relevant components $L_{c,x} = \{k : r_{k,c,x} > \tau\}$ by creating a new signal $x_2 = \mathrm{INV}(\mathbf{X}_2, \mathbf{P}_x)$, where $\mathbf{X}_2 = \mathbf{X} - \sum_{l \in L_{c,x}} \mathbf{X}_l$. Faithfulness for x:

$$\mathrm{FF}_x = f(x)_c - f(x_2)_c \tag{5}$$

- Not perfect, can lead to unpredictable changes in classifier's output, samples with poor fidelity have $-$ve FF_x, thus we report median over testing data.

Faithfulness evaluation

- One prior proposed way of computing faithfulness **simulate feature removal from the input** $\longrightarrow$ Observe change in classifier output.
- Very hard to simulate removal for "concept" based methods. However, our decomposition structure allows this possibility!
- Given sample x with predicted class c, remove the set of relevant components $L_{c,x} = \{k : r_{k,c,x} > \tau\}$ by creating a new signal $x_2 = \mathrm{INV}(\mathbf{X}_2, \mathbf{P}_x)$, where $\mathbf{X}_2 = \mathbf{X} - \sum_{l \in L_{c,x}} \mathbf{X}_l$. Faithfulness for x:

$$\mathrm{FF}_x = f(x)_c - f(x_2)_c \tag{5}$$

- Not perfect, can lead to unpredictable changes in classifier's output, samples with poor fidelity have $-$ve FF_x, thus we report median over testing data.
- Compare it against *Random Baseline*: Randomly select same # of components to remove from the remaining components.

ESC-50 Faithfulness

System	Threshold τ	$\mathrm{FF}_{\mathrm{median}}$
L2I + Θ_{ATT}	$\tau = 0.9$	0.21
	$\tau = 0.7$	0.42
	$\tau = 0.5$	0.89
	$\tau = 0.3$	1.29
Random Baseline	$\tau = 0.3$	0.00

Table: Faithfulness results (absolute drop in logit value) on ESC-50.

SONYC-UST: Faithfulness

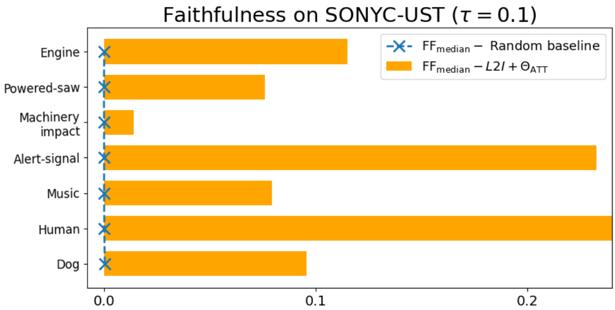


Figure: Faithfulness (absolute drop in probability value) results for SONYC-UST arranged class-wise for threshold, $\tau = 0.1$

Subjective Evaluation

- User study (15 participants) to evaluate quality & understandability of L2I interpretations on SONYC-UST. Compared against SLIME.

Subjective Evaluation

- User study (15 participants) to evaluate quality & understandability of L2I interpretations on SONYC-UST. Compared against SLIME.
- Participants provided with following information for 10 samples:
  - Input audio
  - Predicted class of classifier
  - Interpretation audio from L2I and SLIME

Subjective Evaluation

- User study (15 participants) to evaluate quality & understandability of L2I interpretations on SONYC-UST. Compared against SLIME.
- Participants provided with following information for 10 samples:
  - Input audio
  - Predicted class of classifier
  - Interpretation audio from L2I and SLIME
- Rate both interpretations (scale 0-100) for the following question: "*How well does the interpretation correspond to the part of input audio associated with the given class?*"

Subjective Evaluation

- User study (15 participants) to evaluate quality & understandability of L2I interpretations on SONYC-UST. Compared against SLIME.
- Participants provided with following information for 10 samples:
  - Input audio
  - Predicted class of classifier
  - Interpretation audio from L2I and SLIME
- Rate both interpretations (scale 0-100) for the following question: "*How well does the interpretation correspond to the part of input audio associated with the given class?*"

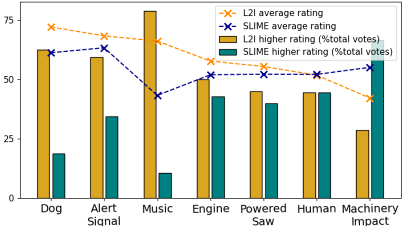


Figure: Class-wise average scores for L2I, SLIME and fraction of votes in favour of each system

Qualitative results

https://listen2interpret.000webhostapp.com/

Conclusions

- In summary, presented a post-hoc interpretability system for networks that process audio
- Using high-level audio objects for listenable interpretations
- Novel usage of NMF to link with deep neural network representations, specially for interpretations
- Real-world multi-label dataset tackled, first of its kind faithfulness evaluation

The End

THANK YOU!

Paper available on arxiv (arXiv:2202.11479)