Deep Embeddings for Rare Audio Event Detection with Imbalanced Data
Vipul Arora, Ming Sun and Chao Wang

WHAT IF THE DATA IS IMBALANCED OVER DIFFERENT CLASSES?

SUPERVISED LEARNING OF CLASSIFIERS

INPUT FEATURES \( x_i \) \rightarrow CLASSIFIER \rightarrow OUTPUT CLASS \( y_i \)

TRAINING DATA

WHAT IF THE DATA IS IMBALANCED OVER DIFFERENT CLASSES?

EMBEDDING LEARNING

LEARN THE EMBEDDING SUCH THAT
\( S_1 > S_2 > S_3 \)
\( \langle . \rangle \) is cosine similarity

INPUT FEATURES \( x_i \) \rightarrow DEEP NETWORK \rightarrow \mathcal{E}(x_i) \rightarrow CLASSIFIER \rightarrow OUTPUT CLASS \( y_i \)

\( W \) = Embedding Network

\( h_i = W^T \mathcal{E}(x_i) \)

If \( W^T \) and \( \mathcal{E}(x_i) \) are L2 normalized, then
\[ v_i[k_1] = \frac{e^{aS_1}}{e^{aS_1} + e^{aS_2 - \beta} + e^{aS_3 - \beta}} \]

The loss function is
(derived from weighted categorical cross entropy loss)
\[ \mathcal{L} = -\sum_i \log v_i[k_1] \]

CLASSIFIER TRAINING

INPUT FEATURES \( x_i \) \rightarrow DEEP NETWORK \rightarrow \mathcal{E}(x_i) \rightarrow OUTPUT CLASS \( y_i \)

Train last layer, and fine tune end-to-end

AUDIO EVENT DETECTION

Dataset:
- 10s audio samples from AudioSet
- Weakly labeled with 1 event or background
- Training (70%), validation (20%), testing (20%)

Input Features from audio:
- Frame length of 25ms and hop size of 10ms
- 64 dimensional log mel filter bank energies
- Mean and variance normalization

Embedding Networks:
1. LSTM model
   - Single LSTM layer with 128 nodes
2. CNN model
   - First layer: 32 7x7 conv filters, ReLU activation
   - Batch normalization
   - 5x4 max pooling, 30% dropout
   - Second layer: 64 7x7 conv filters, ReLU activation
   - Batch normalization
   - 100x4 max pooling, 30% dropout

Trained with Adam, batch size 64 with 8 parallel GPUs

Baseline:
Class-weighted loss function, same network architecture

LSTM MODEL WITH 6:1:2:16 DATA RATIO
CNN MODEL WITH 2:2:1:26 DATA RATIO

EER Baseline Proposed1 Proposed2
Dog 21.5 19.1 18.3
Baby 22.4 19.3 15.5
Gun 17.0 17.1 13.6
Overall 20.3 18.5 15.8

(Proposed2 does final end-to-end tuning of Embedding+Classifier, Proposed1 does not)