Specialized Embedding Approximation for Edge Intelligence: A Case Study in Urban Sound Classification

Sangeeta Srivastava
Ohio State University

Dhrubojyoti Roy
Ohio State University

Mark Cartwright
New York University

Juan Pablo Bello
New York University

Anish Arora
Ohio State University

June 11, 2021

Acoustic Event Detection on Edge

- AED use large audio embedding models for generalizability
- Edge devices use low-compute and low-memory SoC for energy efficiency
  - Cortex-M7 has 1 MB RAM and 2 MB Flash

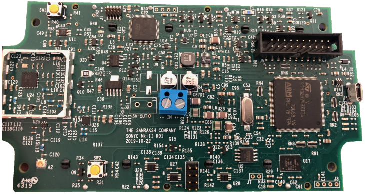


Figure 1: ARM Cortex-M7 based edge device deployed in New York.

- Generalizable audio embedding models too big for edge devices

Knowledge Distillation

- **Traditional setup**: Student trained with the same data as the teacher
- Student tasked to preserve both intra-domain and cross-domain generalizability learned by teacher

Limitation

Traditional setup leads to sub-optimal compression when cross-domain generalizability not necessary

Goal

Simplify the student embedding model for edge devices by specializing for a target domain

Domain Specialized Distillation

- **Requirement:** Preserve intra-domain generalizability
- Approximate teacher's embedding space relevant to target domain
  - Sacrifice cross-domain generalizability

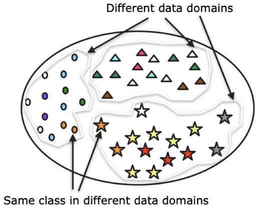


Figure 2: Teacher's Embedding Space: Cross- and Intra-domain generalizable

Figure 3: Student's Embedding Space: Intra-domain generalizable

- **How?** Leverage data related to the target domain for training student embedding

Specialized Embedding Approximation (SEA)

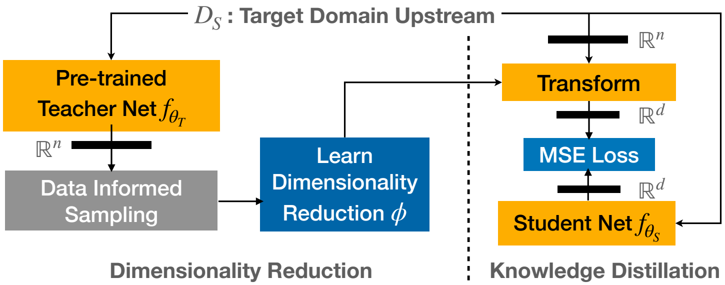


Figure 4: SEA pipeline to train a student produce $\mathbb{R}^d$ embedding from a teacher with $\mathbb{R}^n$ output where $d < n$. D_S is the training data for the student network.

Urban Sound Classification

Figure 5: Acoustic unit deployed in New York

- Sounds of New York City (SONYC) aims at continuous monitoring, analysing, and mitigating urban noise pollution
- Embedding model: L^3-Net[1]
- L^3-Net audio requires 18 MB and 12 MB of static and dynamic memory respectively

[1] Arandjelovic, Relja and Zisserman, Andrew. "Look, Listen and Learn". IEEE ICCV. 2017.

SONYC Data

- Upstream
  - Unlabeled audio recordings collected by a subset of 15 sensors (with diversity in deployment location)
  - Audio + Sound Pressure Level (SPL) data
- Downstream: SONYC-UST[2]
  - Multi-label dataset consisting of 3068 annotated 10-second audio recordings
  - Imbalanced dataset with 8 classes
  - Evaluation metric: Micro-AUPRC

[2] https://doi.org/10.5281/zenodo.2590742

SEA Students on SONYC-UST

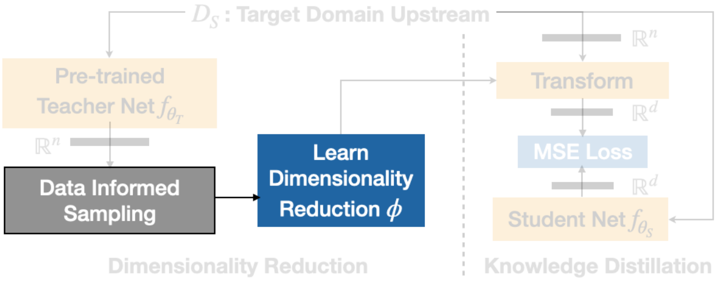
- Student Nets
  - Reduced input representation (8 kHz sampling, 64 mel filters instead of L^3's 48 kHz and 256 mels)

Model	Filter reduction in conv. blocks		Emb. Dim.	Model Size (MB)	Act. Mem. (MB)	Micro-AUPRC
	1, 2, 3	4				
L3-Audio	N/A		512	18.80	12.79	0.810
Student 0	N/A		512	18.80	0.82	0.823
Student 1	50%	50%	256	4.70	0.41	0.793
Student 2	50%	75%	128	2.34	0.41	0.797
Student 3	50%	87.5%	64	1.60	0.41	0.783

Table 1: SEA improves baseline and produces a much smaller Student 2 with comparable performance

- Train Efficiency
  - 10x lesser train data
  - converges 5x (10x) faster with a learning rate of 10^{-5} (10^{-4})

Dimensionality Reduction with Informed Sampling



- Learning ϕ is memory intensive for large SONYC upstream
- Upstream sampling with as much structural information in the target manifold as possible
- Subsets with one or more properties:
    - Random
    - Relevant
    - Diverse

Effect of Sampling on SONYC-UST

- **Relevance**: More informative data points
  - Higher relative loudness (SPL) → potential noise source
- **Diversity**: Diverse set to capture most of the global structure information

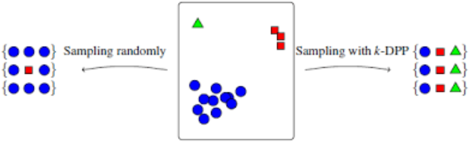


Figure 6: Zhang, Cheng, et al. "DPP for mini-batch diversification."

Sampling type	Micro-AUPRC
Diversity + Relevance	**0.783**
Only Diversity	0.781
Random	0.782
Only Relevance	0.779

Table 2: Student 3 on SONYC-UST when trained with PCA reduced embeddings with different sampling techniques. SONYC SEA Students used PCA with Diversity + Relevance.

edgel3 Python Package

- Reference L^3 audio models for edge
- *pip install edgel3*

```
import edgel3
import soundfile as sf

audio, sr = sf.read('/path/to/file.wav')

# Get embedding out of SEA Student 2 (UST data domain)
emb, ts = edgel3.get_embedding(audio, sr, model_type='sea', emb_dim=128)

# Get embedding out of 95.45% sparse fine-tuned L3
emb, ts = edgel3.get_embedding(audio, sr, model_type='sparse',
                retrain_type='ft', sparsity=95.45)

# Get embedding out of 81.0% sparse knowledge distilled L3
emb, ts = edgel3.get_embedding(audio, sr, model_type='sparse',
                retrain_type='kd', sparsity=81.0)
```

Conclusion

- More compression and train efficiency in knowledge distillation when student restricted to target domain
- Which model do we use for SONYC?
  - 8-bit quantized Student 2
  - **0.585 MB of static and 0.1025 MB of dynamic memory**
- Audio embedding models for the edge made available in *edgel3* package
- Source code for SEA pipeline available at Github

SCAN ME