Sound event classification using ontology-based neural networks

Abelino Jiménez*
ECE Department
Carnegie Mellon University
abelinoj@andrew.cmu.edu

Benjamin Elizalde*
ECE Department
Carnegie Mellon University
bmartin1@andrew.cmu.edu

Bhiksha Raj
LTI
Carnegie Mellon University
bhiksha@cs.cmu.edu

Abstract

State of the art sound event classification relies in neural networks to learn the associations between class labels and audio recordings within a dataset. These datasets typically define an ontology to create a structure that relates these sound classes with more abstract super classes. Hence, the ontology serves as a source of domain knowledge representation of sounds. However, the ontology information is rarely considered, and specially under explored to model neural network architectures. We propose two ontology-based neural network architectures for sound event classification. We defined a framework to design simple network architectures that preserve an ontological structure. The networks are trained and evaluated using two of the most common sound event classification datasets. Results show an improvement in classification performance demonstrating the benefits of including the ontological information.

1 Introduction

Humans can identify a large number of sounds in their environments e.g., a *baby crying*, a *wailing ambulance siren*, *microwave bell*. These sounds can be related to more abstract categories that aid interpretation e.g., *humans*, *emergency vehicles*, *home*. These relations and structures can be represented by ontologies [1], which are defined for most of the available datasets for sound event classification (SEC). However, sound event classification rarely exploits this additional available information. Moreover, although neural networks are the state of the art for SEC [2–4], they are rarely designed considering such ontologies.

An ontology is a formal representation of domain knowledge through categories and relationships that can provide structure to the training data and the neural network architecture. The most common type of ontologies are based on abstraction hierarchies defined by linguistics, where a super category represents its subcategories. Generally, the taxonomies are defined by either nouns or verbs e.g., *animal* contains *dog* and *cat*, *dog* contains *dog barking* and *dog howling*. Examples of datasets are ESC-50 [5], UrbanSounds [6], DCASE [7], AudioSet [8]. Another taxonomy can be defined by interactions between objects and materials, actions and descriptors e.g., contains *Scraping*, which contains *Scraping Rapidly* and *Scraping a Board* [9–11]. Another example of this type is given by physical properties, such as frequency and time patterns [12–14]. There are multiple benefits of considering hierarchical relations in sound event classifiers. They can allow the classifier to back-off to more general categories when encountering ambiguity among subcategories. They can disambiguate classes that are acoustically similar, but not semantically. They can be used to penalize classification differently, where miss classifying sounds from different super classes is worse than within the same super class. Lastly, they can be used as domain knowledge to model neural networks.

*Authors contributed equally.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, CA.

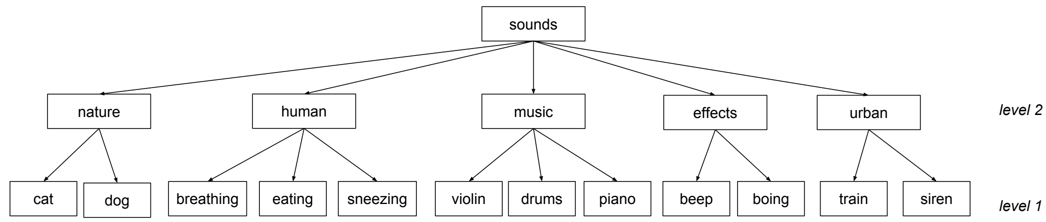

Figure 1: Ontology of sound events in the MSoS dataset, level 1 has a total of 97 classes distributed across 5 super classes in level 2.

In fact, ontological information has been evaluated in computer vision [15] and music [16], but has rarely been used for sound event classification.

Ontology-based network architectures have showed improvement in performance along with other benefits. Authors in [17] proposed an ontology-based deep restricted Boltzmann machine for textual topic classification. The architecture replicates the tree-like structure adding intermediate layers to model the transformation from a super class to its sub classes. Authors showed improved performance and reduced overfitting in training data. Another example used a perceptron for each node of the hierarchy, which classified whether an image corresponded to such class or not [18]. Authors showed an improvement in performance due to the ability of class disambiguation by comparing predictions of classes and sub classes. Motivated by these approaches and by the flexibility to adapt structures in a deep learning model we propose our ontology-based networks detailed in the following section.

2 Methods

In this section we present a framework to deal with ontological information using deep learning architectures. First, we describe a set of assumptions we consider along this paper. In particular, we describe the type of ontologies we work and some of their implications. Later, we present a Feed-forward model that includes the discussed constraints, defining our proposed ontological layer. Second, in order to preserve an embedding space consistent with the ontological structure, we extended the learning model to compute ontology-based embeddings using Siamese Neural Networks.

2.1 Framework and Assumptions

The framework is defined to make use of the ontology structure and to model the neural network architectures. It should be noted that we considered ontologies with two levels, which are the most common in sound event datasets. Nevertheless, the presented framework can be easily generalized to more levels.

In our framework, we considered the training data $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$, where $\mathbf{x}_i \in \mathcal{X}$ is an audio representation, which is associated to a set of labels given by the ontology $\mathbf{y}_i \in C_1 \times C_2 \times ... \times C_k$. In this case, C_i is the set of possible classes at i-level. Assuming a hierarchical relation, we can consider that each possible class in C_i is mapped to one element in C_{i+1}. The higher the value of i, the higher the level in the ontology.

For example, consider the illustration of an ontology in Figure 1. In this case $k = 2$, $C_1 = \{$*cat* , *dog*, *breathing* , *eating* , *sneezing* , *violin* , *drums* , *piano* , *beep* , *boing* , *train* , *siren*$\}$ and $C_2 = \{$*nature* , *human* , *music* , *effects* , *urban*$\}$. As the figure shows, every element in C_1 is related to one element in C_2; e.g., *cat* belongs to *nature*, or *drums* belongs to *music*.

Furthermore, for a given representation $\mathbf{x} \in \mathcal{X}$, if we know the corresponding label y_1 in C_1, we can infer its label in C_2. This intuition can be formalized using a probabilistic formulation, where it is

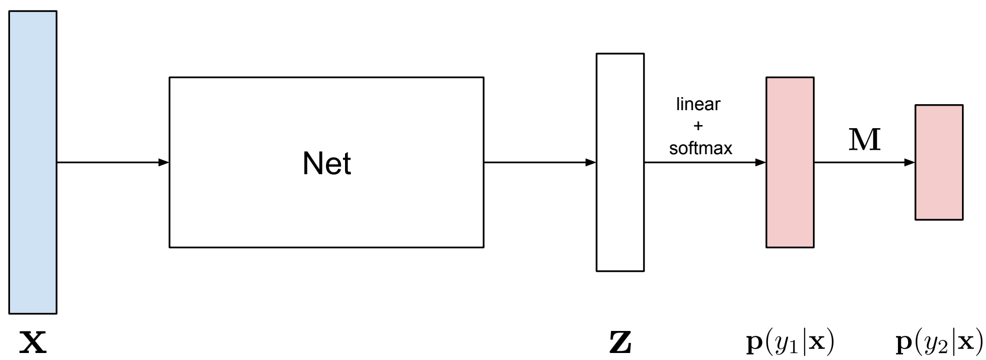

Figure 2: Architecture of the Feed-forward Network with Ontological Layer. The blue column represents the acoustic feature vector, the red columns are the output probabilities for both levels.

straightforward to see that, assuming $p(y_2|y_1, \mathbf{x}) = p(y_2|y_1)$, the following is satisfied:

$$p(y_2|\mathbf{x}) = \sum_{y_1} p(y_2, y_1|\mathbf{x}) \tag{1}$$

$$= \sum_{y_1} p(y_2|y_1, \mathbf{x}) \cdot p(y_1|\mathbf{x}) \tag{2}$$

$$= \sum_{y_1 \in \text{children}(y_2)} p(y_1|\mathbf{x}) \tag{3}$$

Therefore, if we want to estimate $p(y_2|\mathbf{x})$ using a model, we just need to compute the estimation of $p(y_1|\mathbf{x})$ and sum the values corresponding to the children of y_2. This case is valid for inference time, however, it is not clear that using the representation and label $(\mathbf{x}, y_1)$ should be enough to train the model. If at training time we can make use of knowledge to relate the different classes in y_1, it should improve the performance of the model, specially at making predictions for classes y_2.

In the following sections we take our proposed framework and use it to design ontology-based neural network architectures.

2.2 Feed-forward Network with Ontological Layer

In this section, we describe how we use our proposed framework to design the architecture. Also, we introduce the *ontological layer*, which makes use of the ontology structure.

The Feed-forward Network (FFN) with Ontological Layer consists of a base network (Net), an intermediate vector $\mathbf{z}$, and two outputs, one for each ontology level. The base network weights are learned at every parameter update. The base network utilizes an input vector of audio features $\mathbf{x}$ and generates a vector $\mathbf{z}$. This vector is used to generate two outputs, $\mathbf{p}(y_1|\mathbf{x})$ a probability vector for C_1 and $\mathbf{p}(y_2|\mathbf{x})$ a probability vector for C_2. First, the vector $\mathbf{z}$ is passed to a softmax layer of the size of C_1. Then, this output is multiplied by the *ontological layer* $\mathbf{M}$ and generates a layer of size of C_2. Once the FFN is trained, it can be used to predict any class in C_1 and C_2 for any input $\mathbf{x}$.

The *ontological layer* reflects the relation between super classes and sub classes given by the ontology. To describe how we used this layer, we refer to Equation 3, where $p(y_2|\mathbf{x})$ is the sum of all the values of $p(y_1|\mathbf{x})$ corresponding to the children of y_2. If we consider this equation as a directed graph where $\mathbf{M}$ is the $|C_2| \times |C_1|$ incidence matrix, then, it is clear that Equation 3 can be rewritten as,

$$\mathbf{p}(y_2|\mathbf{x}) = \mathbf{M} \cdot \mathbf{p}(y_1|\mathbf{x}) \tag{4}$$

Note that the *ontological layer* $\mathbf{M}$ defines the weights of a standard layer connection. Although we do not consider that these weights are trainable, they are part of our training data.

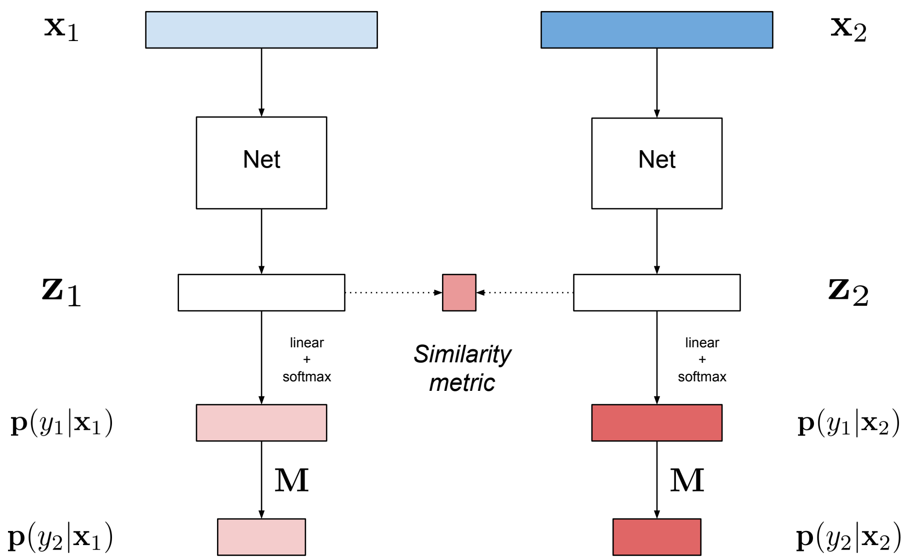

Figure 3: Architecture of the Siamese Neural Network with the Feed-forward Network with Ontological Layer. The blue rows represent the acoustic feature vectors, the white rows represent the ontological embeddings and the red rows are the output probabilities for both levels. Note that the SNN is trained with three types of pairs depending on whether the inputs are from the same subclass, or different subclass, but same super class, or different super class.

In order to train this model, we simply propose to apply gradient-based method to minimize the loss function $\mathcal{L}$, which is a convex combination between two categorical cross-entropy functions; $\mathcal{L}_1$ the categorical cross entropy corresponding to $\mathbf{p}(y_1|\mathbf{x})$ and $\mathcal{L}_2$ corresponding to $\mathbf{p}(y_2|\mathbf{x})$. Formally,

$$\mathcal{L} \quad = \quad \lambda \mathcal{L}_1 + (1-\lambda)\, \mathcal{L}_2 \tag{5}$$

Hence, we consider $\lambda \in [0,1]$ as a hyper parameter to be tuned. Note that, when $\lambda = 1$, we are reducing the problem to train a standard classifier just using the information from the first level of the ontology.

2.3 Ontology-based embeddings for Feed-forward Model with Ontological layer

In this section, we describe how we learned the ontology-based embeddings.

Our goal is to create embeddings that preserved the ontological structure. We used a Siamese neural network (SNN), which enforces samples of the same class to be closer, while separating samples of different classes. If two samples belong to different subclasses, but they belong to the same super class, they are closer than two samples that belong to different super classes. The architecture of the SNN with the Feed-forward Network with Ontological Layer is shown in Fig. 3. The blue rows represent the acoustic feature vectors of two different samples; they can be from the same subclass, different subclass but same super class, or different super class. Then, the twin networks have the same base architecture (Net) with shared weights. The weights are learned simultaneously at every parameter update. The white rows represent the ontological embeddings used to compute a Similarity metric (Euclidean Distance), where the distance of the embeddings $\mathbf{z}_1$ and $\mathbf{z}_2$ should indicate how different $\mathbf{x}_1$ and $\mathbf{x}_2$ are with respect to the ontology. For this work, we imposed that the distance between $\mathbf{z}_1$ and $\mathbf{z}_2$ is close to 0 if the samples are from the same subclass, close to 5 if they are from different sub classes, but the same super class, and close to 10 if they are from different super classes. Finally, the red rows are the output probabilities for both levels, $\mathbf{p}(y_1|\mathbf{x}_1)$, $\mathbf{p}(y_1|\mathbf{x}_2)$, $\mathbf{p}(y_2|\mathbf{x}_1)$ and $\mathbf{p}(y_2|\mathbf{x}_2)$.

To train the Feed-forward Model with Ontological layer using Ontology-based embeddings, we provided the three types of pairs of audio examples and applied a gradient-based method to minimize

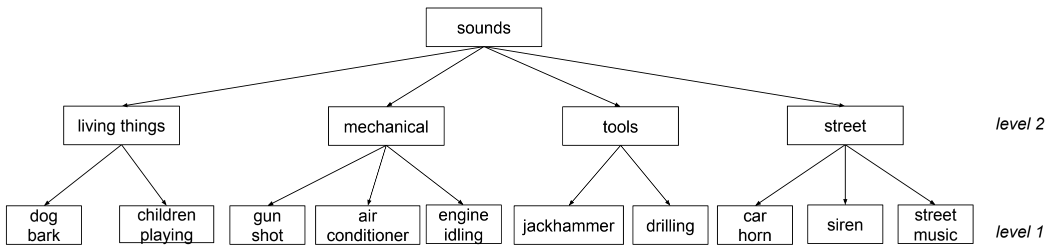

Figure 4: Ontology of sound events in the US8K dataset, level 1 has a total of 10 classes distributed across 4 super classes in level 2.

the loss function $\mathcal{L}$. The loss is a linear combination between four categorical cross-entropy functions: $\mathcal{L}_1^1$ and $\mathcal{L}_1^2$ are the categorical cross-entropy losses corresponding to level 1 $\mathbf{p}(y_1|\mathbf{x}_1)$ and $\mathbf{p}(y_1|\mathbf{x}_2)$ respectively, $\mathcal{L}_2^1$ and $\mathcal{L}_2^2$ are the categorical cross-entropy losses corresponding to level 2 $\mathbf{p}(y_2|\mathbf{x}_1)$ and $\mathbf{p}(y_2|\mathbf{x}_2)$, and the metric D_w given by square of the difference between the Euclidean Distance between embeddings $\mathbf{z}_1$ and $\mathbf{z}_2$ and the target value given by the ontology (0, 5 or 10). Formally,

$$\mathcal{L} \quad = \quad \lambda_1(\mathcal{L}_1^1 + \mathcal{L}_1^2) + \lambda_2(\mathcal{L}_2^1 + \mathcal{L}_2^2) + \lambda_3 D_w \qquad (6)$$

3 Experimental Results

In this section, we evaluate the sound event classification performance of the ontological-based neural network architectures. We present the datasets and its ontologies, the baseline and proposed architectures, and the classification performance at different levels of the hierarchy.

3.1 Datasets and Ontologies

Making Sense of Sounds Challenge [2] **- MSoS:** The dataset is designed for a challenge which objective is to classify the most abstract classes or highest level in its taxonomy. The ontology, illustrated in Fig. 1 has two levels, the lowest level 1, has 97 classes and the highest level 2, has 5 classes. The audio files were taken from Freesound data base, the ESC-50 dataset and the Cambridge-MT Multitrack Download Library. The development dataset consists of 1500 audio files divided into the five categories, each containing 300 files. The number of different sound types within each category is not balanced. The evaluation dataset consists of 500 audio files, 100 files per category. All files have an identical format: single-channel 44.1 kHz, 16-bit *.wav* files. We randomly partitioned the set in 80% for training and tuning parameters and 10% for testing. All files are exactly 5 seconds long, but may feature periods of silence. The official blind evaluation set of the challenge consisted on 500 files distributed among the 5 classes.

Urban Sounds - US8K: The dataset is designed to evaluate classification of urban sounds, which are organized using a taxonomy with more nodes than the annotated number of classes. Due to this reason, we adjusted the taxonomy to avoid redundant levels with only one annotated child. The resulting ontology is illustrated in Fig. 4, with two levels, the lowest level 1, has 10 classes and the highest level 2, has 4 classes. The audio files were taken from Freesound data base and corresponded to real field recordings. All files have an identical format: single-channel 44.1 kHz, 16-bit *.wav* files. The dataset contains 8,732 audio files divided into 10 stratified subsets. We used 9 folds to train and tune parameters and one fold for testing.

3.2 Audio Features

We used state-of-the-art Walnet features [2] to represent audio recordings. For each audio, we computed a 128-dimensional logmel-spectrogram vector and transformed it *via* a convolutional neural network (CNN) that was trained separately on the balanced set of AudioSet. The network

[2]http://cvssp.org/projects/making_sense_of_sounds/site/challenge/

Dataset	Model	Accuracy in Level 1	Accuracy in Level 2
MSoS	Baseline (Challenge)	-	0.810
	Baseline	0.686	0.853
	FF + Ontology	0.740	0.913
	Ontology Embeddings	0.736	0.886
US8K	Baseline	0.790	0.861
	FF + Ontology	0.825	0.863
	Ontology Embeddings	0.818	0.856

Table 1: EDIT THIS: Both proposed methods outperformed the baseline, which does not use any ontology information.

comprised 8 convolutional layers, resulting in an output feature vector of dimensionality 527. To this, we concatenated intermediate outputs from the 8^{th} layer of the CNN with dimensionality of 1024.

3.3 Base Network Architecture (Net)

The architecture of the base network (Net) considered in this experiment, shown in Fig. 2, is a feed-forward multi-layer perceptron network. It consists of 4 layers: the input layer of dimensionality 1024, which takes audio feature vectors, 2 dense layers of dimensionality 512 and 256, respectively, and the output layer of dimensionality 128, which is the dimensionality of the vector $\mathbf{z}$. The dense layers utilize Batch Normalization, a dropout rate of 0.5 and the ReLU activation function; $\max(0, x)$, where x is input to the function. We tuned the parameters in the Net box as well as the parameters that transformed $\mathbf{z}$ into $\mathbf{p}(y_1|\mathbf{x})$.

3.4 Performance of Baseline Models

We considered baseline models in both level 1 and 2 for the different data sets. In this case, the baseline models did not consider any ontological information, hence the models consist of the Base Network Architecture with the addition of an output layer that was either for level 1 or level 2.

Note that for level 1 this is equivalent to training the Feed-forward model with Ontological Layer using $\lambda = 1$. Indeed, with $\lambda = 1$ the loss function associated to level 2 is not considered. For level 2, the baseline model is different from the Feed-forward model with $\lambda = 0$, because in the baseline model there is no layer corresponding to the prediction of y_1. Table 1 shows the results of baseline models for both, MSoS and US8K data set in level 1 and level 2.

The baseline performance of the development set in the MSoS challenge was reported to be 0.81 for level 2 and no baseline was provided for level 1.

3.5 Performance of Feed-forward Model with Ontological layer

To validate the architecture presented in Section 2.2 and analyze the utility of the ontological layer, we trained models taking different values of λ. Figure 5 shows the effect of λ in both data sets. In general, we observe that considering values different from 0 and 1 helps to increase the performance. Note that the classification in both level is affected by the ontological layer.

In the case of MSoS data set, the best performance was obtained using $\lambda = 0.8$, getting 0.74 and 0.913 of accuracy in level 1 and 2 respectively. Thus, using the ontological structure we can get an absolute improvement of 5.4% and 6% respect baseline models.

Running the same experiment on the US8K data set, we observe a smaller improvement. The best performance was obtained using $\lambda = 0.7$, being the accuracy of 0.82 and 0.86 for level 1 and 2 respectively. This means an improvement of 2.5% and 0.2% only, respect baseline models.

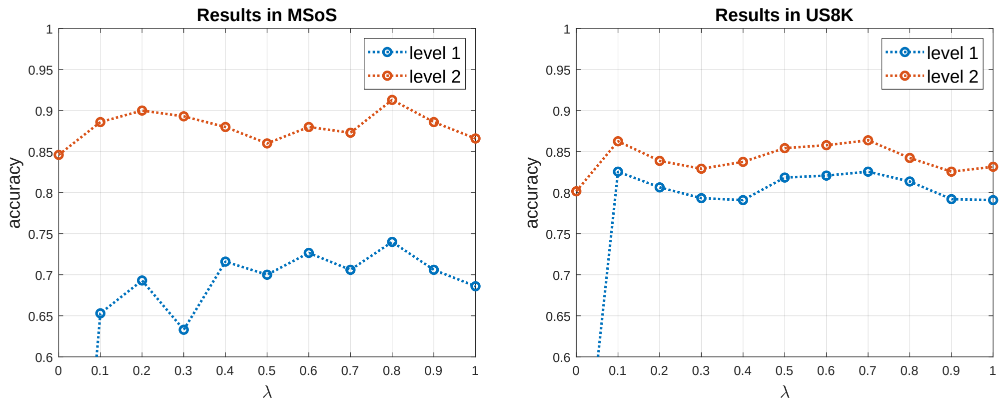

Figure 5: Results in different level prediction of Feed-forward Network with Ontological Layer using different values of λ in training phase. (Left) Results in MSoS data set. Best result is achieved using $\lambda = 0.8$ (Right) Results in US8K data set. Best result is achieved using $\lambda = 0.7$

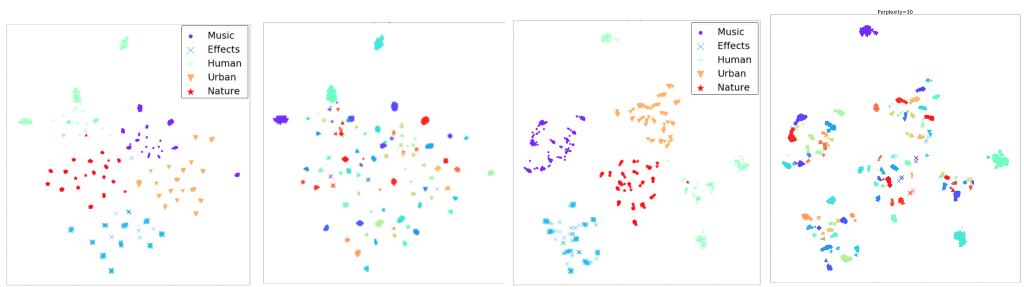

Figure 6: The MSoS t-SNE plots of the samples in classes from level 2 (1st and 3rd) and level 1 (2nd and 4th). The first two boxes are from the base network vectors and the second two boxes are the ontology-based embeddings. We observe in 1st and 3rd, the groups of classes in level 2 and in 2nd and 4th the same level 2 groups, but using the level 1 class samples. The ontology-based embeddings results in tighter and better defined clusters.

3.6 Performance of Ontology-based Embeddings with Feed-forward Model with Ontological layer

We tested the architecture described in Section 2.3 to evaluate the performance of the ontology-based embeddings for sound event classification. Additionally, we include t-SNE plots, to illustrate how the embeddings cluster at different levels.

We processed the Walnet audio features and chose different super and sub class pairs to train the Siamese neural network to produce the ontology-based embeddings. The embeddings are passed to the architecture of the base network (Net), which is the same as the one used in the previous section. We trained the SNN for 50 epochs using the Adam algorithm. We also tuned the hyper-parameters of the SNN to achieve good performance with the input features that are described in the next section. We also tried different number of pairs for the input training data, from 100 to 1,000,000 pairs and found that 100,000 yielded the best performance. For the loss function we used the values derived in the previous experiment. We used the value of 0.8 for the lambda of the classifiers of level 2 and 0.2 for the classifiers in level 1, and 0.2 for the similarity metric. Modifying the lambdas in the loss function affected the overall performance.

The results in Table 1 show that the accuracy performance of MSoS and US8K were respectively as follows, in level 1 0.736 and 0.818, and in level 2 0.886 and 0.856. Based on these results we made the following conclusions. The performance of this architecture is better than the baseline, but slightly under performed the method without the embeddings. Nevertheless, the ontology-based embeddings have the benefit of better grouping as illustrated in Figure6. We took the MSoS data and created the t-SNE plots (perplexity=30) of the classes in level 2 and level 1. We observed that the FF + Ontology vectors and the ontology-based embeddings provided clustered groups of level 2 classes. However, the ontology-based embeddings have tighter and better defined clusters.

In the case of the US8K data set performance was limited. We think this was because the number of sub classes was similar to the number of super classes. We had 10 sub classes for 4 classes unlike the MSoS data set, where we had 97 sub classes and 5 classes. It seems when the ratio between the number of sub classes and the number of classes is not large, the contribution of the ontology is negligible.

Both approaches were used to compete in the Making Sense of Sounds Challenge. The baseline for the blind evaluation set was 0.80 accuracy for level 2. The Feed-forward Network with Ontological Layer achieved 0.88 while using the ontological-embeddings achieved 0.89. Again, both architectures outperformed significantly the baseline.

4 Conclusions and Future work

In this paper we proposed a framework to design neural networks for sound event classification using hierarchical ontologies. We have shown two methods to add such structure into deep learning models in a simple manner without adding more learnable parameters. We used a Feed-forward Network with an ontological layer to relate predictions of different levels in the hierarchy. Additionally, we proposed a Siamese neural Network to compute ontology-based embeddings to preserve the ontology in an embedding space. The embeddings plots showed clusters of super classes containing different sub classes. Our results in the datasets and MSoS challenge improved over the baselines. We expect that our results pave the path to further explore ontologies and other relations, which is fundamental for sound event classification due to wide acoustic diversity and limited lexicalized terms to describe sounds.

References

[1] Balakrishnan Chandrasekaran, John R Josephson, and V Richard Benjamins, "What are ontologies, and why do we need them?," *IEEE Intelligent Systems and their applications*, vol. 14, no. 1, pp. 20–26, 1999.

[2] Ankit Shah, Anurag Kumar, Alexander G Hauptmann, and Bhiksha Raj, "A closer look at weak label learning for audio events," *arXiv preprint arXiv:1804.09288*, 2018.

[3] Khaled Koutini, Hamid Eghbal-zadeh, and Gerhard Widmer, "Iterative knowledge distillation in r-cnns for weakly-labeled semi-supervised sound event detection," Tech. Rep., DCASE2018 Challenge, September 2018.

[4] Qiuqiang Kong, Iqbal Turab, Xu Yong, Wenwu Wang, and Mark D. Plumbley, "DCASE 2018 challenge baseline with convolutional neural networks," Tech. Rep., DCASE2018 Challenge, September 2018.

[5] Karol J. Piczak, "ESC: dataset for environmental sound classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15, Brisbane, Australia, October 26 - 30*, 2015.

[6] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*, Orlando, FL, USA, 2014, pp. 1041–1044.

[7] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017.

[8] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, New Orleans, LA, 2017, IEEE, pp. 776–780.

[9] Guillaume Lemaitre and Laurie M Heller, "Evidence for a basic level in a taxonomy of everyday action sounds," *Experimental brain research*, vol. 226, no. 2, pp. 253–264, 2013.

[10] R Murray Schafer, *The soundscape: Our sonic environment and the tuning of the world*, Simon and Schuster, 1993.

[11] Sebastian Säger, Benjamin Elizalde, Damian Borth, Christian Schulze, Bhiksha Raj, and Ian Lane, "Audiopairbank: towards a large-scale tag-pair-based audio content analysis," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, no. 1, pp. 12, 2018.

[12] Susanne Burger, Qin Jin, Peter F. Schulam, and Florian Metze, " Noisemes: Manual Annotation of Environmental Noise in Audio Streams ," Tech. Rep., 2012.

[13] Tomohiro Nakatani and Hiroshi G Okuno, "Sound ontology for computational auditory scence analysis," in *AAAI/IAAI*, 1998, pp. 1004–1010.

[14] Oliver Bones, Trevor J Cox, and William J Davies, "Sound categories: category formation and evidence-based taxonomies," *Frontiers in psychology*, vol. 9, 2018.

[15] Ora Lassila, "Towards the semantic web," in *Towards the Semantic Web and Web Services Conference, Helsinki, October*, 2002, pp. 21–22.

[16] Yves Raimond, Samer A Abdallah, Mark B Sandler, and Frederick Giasson, "The music ontology.," in *ISMIR*. Citeseer, 2007, vol. 2007, p. 8th.

[17] Hao Wang, Dejing Dou, and Daniel Lowd, "Ontology-based deep restricted boltzmann machine," in *International Conference on Database and Expert Systems Applications*. Springer, 2016, pp. 431–445.

[18] Casey Breen, Latifur Khan, and Arunkumar Ponnusamy, "Image classification using neural networks and ontologies," in *Database and Expert Systems Applications, 2002. Proceedings. 13th International Workshop on*. IEEE, 2002, pp. 98–102.