Sound event classification using ontology-based neural networks

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 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 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 accuracy demonstrating the benefits of the ontology.

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 already 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 as domain knowledge to model neural networks. 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 superclass to its subclasses. 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 subclasses. 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 find more discriminative features, which are 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 \(\{(x_1, y_1), \ldots, (x_n, y_n)\}\), where \(x_i \in \mathcal{X}\) is an audio
representation, which is associated to a set of labels given by the ontology \(y_i \in C_1 \times C_2 \times \ldots \times C_k\).
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 an example, consider the illustration of an ontology in Figure 1. In this case \(k = 2\), \(C_1 = \{\text{cat, dog, breathing, eating, sneezing, violin, drums, piano, beep, boing, train, siren}\}\) and
\(C_2 = \{\text{nature, human, music, effects, urban}\}\). As the figure shows, every element in \(C_1\) is related
to one element in \(C_2\); e.g., \text{cat} belongs to \text{nature}, or \text{drums} belongs to \text{music}.

Furthermore, for a given representation \(x \in \mathcal{X}\), 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
straightforward to see that, assuming \(p(y_2|y_1, x) = p(y_2|y_1)\), the following is satisfied:

\[
p(y_2|x) = \sum_{y_1} p(y_2, y_1|x) = \sum_{y_1} p(y_2|y_1, x) \cdot p(y_1|x) = \sum_{y_1 \in \text{children}(y_2)} p(y_1|x)
\]  

Therefore, if we want to estimate \(p(y_2|x)\) using a model, we just need to compute the estimation of
\(p(y_1|x)\) and sum the values corresponding to the children of \(y_2\). This case is valid for inference time,

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.
however, it is not clear that using the representation and label \((x, y_1)\) 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 \(z\), 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 \(x\) and generates a vector \(z\). This vector is used to generate two outputs, \(p(y_1|x)\) a probability vector for \(C_1\) and \(p(y_2|x)\) a probability vector for \(C_2\). First, the vector \(z\) is passed to a softmax layer of the size of \(C_1\). Second, the same vector \(z\) is multiplied by the **ontological layer** \(M\) and generates a layer of size \(C_2\). Once the FFN is trained, it can be used to predict any class \(C_1\) and \(C_2\) for any input \(x\).

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

\[
p(y_2|x) = M \cdot p(y_1|x) \tag{4}
\]

Note that the **ontological layer** \(M\) 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.

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 \(p(y_1|x)\) and \(\mathcal{L}_2\) corresponding to \(p(y_2|x)\). Formally,

\[
\mathcal{L} = \lambda \mathcal{L}_1 + (1 - \lambda) \mathcal{L}_2 \tag{5}
\]

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.
Figure 3: Architecture of the SNN that is used to learn ontological embeddings. 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.

### 2.3 Ontology-based embeddings

In this section, we describe how we learned the ontology-based embeddings. The embeddings are computed using a Siamese neural network (SNN), shown in Fig. 3, consisting of twin networks that have the same base architecture (Net) with shared weights. The weights are learned simultaneously at every parameter update. Each base network utilizes an input vector of audio features $x$. Then, for the inputs $x_1$ and $x_2$, we can obtain the outputs $p(y_1|x_1)$, $p(y_1|x_2)$, $p(y_2|x_1)$ and $p(y_2|x_2)$. In addition, we consider as output the similarity metric between the hidden vectors $z_1$ and $z_2$, as illustrated in Fig. 3.

For training, first, we need to associate a loss function to every output. The innovation of this model is the loss for the similarity metric. Our attempt is that the similarity metric describes somehow the ontology; the difference of the embeddings $z_1$ and $z_1$ should indicate how different $x_1$ and $x_2$ are with respect to the ontology. In a 2-level ontology there are 3 possible distances, for this work we chose 0 or 1 or 10 depending on whether the inputs are from the same subclass, or different subclass, but same super class, or different super class. Hence, the model attempts to approximate the distance within the ontology using the distance between embeddings.

In order to train the full model, we need to provide pairs of audio examples and apply a gradient-based method to minimize the loss function $L$. We propose to use a linear combination between four categorical cross-entropy functions: $L_1^1$ and $L_2^1$ the categorical cross-entropy corresponding to $p(y_1|x_1)$ and $p(y_1|x_2)$ respectively, and $L_2^1$ and $L_2^2$ corresponding to $p(y_2|x_1)$ and $p(y_2|x_2)$, and finally the similarity metric $D_w$ given by Euclidean Distance. Formally,

$$ L = \lambda_1(L_1^1 + L_2^1) + \lambda_2(L_1^2 + L_2^2) + \lambda_3 D_w $$

### 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 improvement in classification at different levels of the hierarchy.
3.1 Datasets and Ontologies

Making Sense of Sounds Challenge - 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 database, the ESC-50 dataset and the Cambridge-MIT 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. All files are exactly 5 seconds long, but may feature periods of silence.

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 database 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. All files are of up to 4 seconds in duration.

3.2 Audio Features

We used state-of-the-art Walnut features 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 comprised 8 convolutional layers, resulting in an output feature vector of dimensionality 527. To this, we concatenated intermediate outputs from the 8th 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 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 transform \( z \) into \( p(y_1 | x) \).

3.4 Performance of Baseline Models

We considered baseline models both in level 1 and 2 in different data sets. In this case, the baseline models do not consider any ontological structure, so the models take the Base Network Architecture adding an output layer to either level 1 or 2.
Figure 5: Results in different level prediction of Feed-forward Network with Ontological Layer using different values of $\lambda$ 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 $\lambda = 0.7$

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Accuracy in Level 1</th>
<th>Accuracy in Level 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSoS</td>
<td>Baseline (Challenge)</td>
<td>-</td>
<td>0.810</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td>0.686</td>
<td>0.853</td>
</tr>
<tr>
<td></td>
<td>FF + Ontology</td>
<td>0.740</td>
<td>0.913</td>
</tr>
<tr>
<td></td>
<td>Ontology Embeddings</td>
<td>0.736</td>
<td>0.886</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td>0.790</td>
<td>0.861</td>
</tr>
<tr>
<td>US8K</td>
<td>FF + Ontology</td>
<td>0.825</td>
<td>0.863</td>
</tr>
<tr>
<td></td>
<td>Ontology Embeddings</td>
<td>0.813</td>
<td>0.861</td>
</tr>
</tbody>
</table>

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

Note that for level 1 this is equivalent to train 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 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 $\lambda$. Figure 5 shows the effect of $\lambda$ 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 74.0% and 91.3% 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 82.5% and 86.3% for level 1 and 2 respectively. This means an improvement of 2.5% and 0.2% only, respect baseline models.
3.6 Performance of Ontology-based embeddings

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 the super classes clustered with our embeddings.

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 architecture of the base network (Net) 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 considered the values computed 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.

The results in Table 1 show that the accuracy performance of MSoS and US8K were respectively as follows, in level 1 73.6% and 81.3%, and in level 2 88.6% and 86.1%. Based on these results we make the following conclusions. The performance of this architecture is better than the baseline, but slightly under performs the original method of FF+Ontology. Nevertheless, the ontology-based embeddings have the added benefit of better grouping as illustrated in the t-SNE plots of the level 2 classes of the MSoS ontology in Figure 6. We also noticed that modifying the lambdas in the loss function affected the overall performance.

4 Conclusions and Future work

In this paper we proposed a framework to learn classification task using hierarchical ontologies. We have shown that it is possible to add ontological information to a deep learning model in a simple manner, without adding more learnable parameters, but considering a structure between prediction of different levels in the ontology in the learning phase. This seems to be fundamental for Sound event classification, where within a class we can find a lot of diverse sounds. Hence, the proposed models help to deal with the class ambiguity including extra knowledge. Our experimental results shows that using the ontological structure it is possible to obtain an improvement of an absolute 5% to 6% percent approximately in different levels of prediction in the MSoS data set.

However, the improvement is smaller in the case of the US8K data set. We think this is because the number of subclasses is comparable with the number of classes; we have 10 subclasses and 4 classes, unlike the case of the MSoS data set, where we have 97 subclasses and 5 classes. It seems when the ratio between the number of subclasses and the number of classes is not large, the contribution of the ontology is minor.
Currently, we have many possible directions to continue this work. One is the study of non-hierarchical ontologies. To do so, one alternative is to generalize the ontological layers, adding some degree of freedom to the matrix $M$ but keeping the probabilistic interpretation. Other alternative is using distances on the graph that represents the ontology, similarly to the approach taken in the Ontology-based embedding model, and use those distances as a measure of the relation between classes.

Other questions are about the way we select the parameters $\lambda$ in these models. Is there condition depending on the number of classes and subclasses that determines the optimal value of $\lambda$?

Finally, we think necessary to study the extension to multiclass detection, especially for audio, since multiple sounds can appear in the same audio recording. How can we model this type of data into an ontological framework? These and other questions are part of future work.

References


