APPLYING TRIPLET LOSS TO SIAMESE-STYLE NETWORKS FOR AUDIO SIMILARITY RANKING

Brian Margolis, Madhav Ghei, Bryan Pardo

Northwestern University
Electrical Engineering and Computer Science
2133 Sheridan Rd
Evanston, IL 60208, USA
{brianmargolis, madhavghei2018}@u.northwestern.edu, pardo@northwestern.edu

ABSTRACT

Query by vocal imitation (QBV) systems let users search a library of general non-speech audio files using a vocal imitation of the desired sound as the query. The best existing system for QBV uses a similarity measure between vocal imitations and general audio files that is learned by a two-tower semi-Siamese deep neural network architecture. This approach typically uses pairwise training examples and error measurement. In this work, we show that this pairwise error signal does not correlate well with improved search rankings and instead describe how triplet loss can be used to train a two-tower network designed to work with pairwise loss, resulting in better correlation with search rankings. This approach can be used to train any two-tower architecture using triplet loss. Empirical results on a dataset of vocal imitations and general audio files show that low triplet loss is much better correlated with improved search ranking than low pairwise loss.

Index Terms— vocal imitation, information retrieval, convolutional Siamese-style networks, triplet loss

1. INTRODUCTION

Finding ways to easily access relevant audio content is a task that has increased in importance as multimedia collections proliferate and grow. For example, the widely-used *Freesound*[1] website contains hundreds of thousands of individual sound recordings from many categories of sound. Such repositories typically let users search their collections of recordings using text-based search. This allows search through any tags, descriptions and file names, but does not support search on the content of audio files. This is not true just for online repositories. Sound designers rely on commercially deployed sound library management tools, such as Soundly [1], to index their sound file collections. These systems also search on text-based metadata and not the audio content.

Indexing a collection of audio files using text-based descriptors imposes certain natural limitations on how search may be done. Such descriptions often do not provide the necessary detail to evaluate this sound in comparison to others with a similar label. This forces the user to listen to all sounds sharing a label, which can be prohibitively time-consuming. Relying on text descriptions also means that every file must be assigned text labels before one can search for it. Further, the important fine grain characteristics that differentiate between audio files of the same general category may not have widely agreed upon text descriptors, making it difficult to create tags that support fine grained search.

When words fail, vocal imitations can help to bridge the gap left by text descriptors. Since imitation allows for description of sounds in ways that text cannot [2], using a vocal imitation as the query has the potential to yield useful results when text search fails [3]. Query by vocal imitation (QBV) systems allow search in a collection of sound files using a vocal imitation of the desired sound as the search key.

The current state-of-the-art in QBV [4] measures the similarity of the imitation to each sound in the database using a similarity measure output by a two tower Siamese-style neural network. The network takes a vocal imitation and a sound file from the collection as input and outputs a similarity value in the range [0,1]. These similarity numbers are used to rank audio files in the collection. In training, networks are trained on labeled pairs, where 0 indicates a vocal imitation was paired with the incorrect sound and 1 indicates the imitation was paired with the target of the imitation.

In this work, we show that the error signal provided by this pairwise training does not necessarily correlate well with improving the ranking of the target file. We show that a loss function (triplet loss) that explicitly compares the similarity of the imitation query to two different sounds in the collection is much better correlated with the rank of the target. Finally, we show how to adapt triplet loss training to work in a two-tower architecture (see Section 3). This approach can be used to train any two-tower architecture using triplet loss.

2. RELATED WORK

There are a number of audio search approaches that are related to query by vocal imitation of general sounds. Audio fingerprinting services (e.g. the song-finding service *Shazam*[2]) require the query be a portion of the exact audio file sought. One cannot vocally imitate the desired sound and find a match using audio fingerprinting. There are services that make speech recordings searchable as text (e.g. the Microsoft Speech API [3]), but these are not designed to meaningfully encode general sounds or vocal imitations. Query by humming systems focus specifically on melody (e.g. Tunebot [5])

This work supported by USA National Science Foundation Award 1617497.

[1] https://freesound.org/browse/

[2] https://www.shazam.com

[3] https://docs.microsoft.com/en-us/azure/cognitive-services/speech/home

and rhythm (e.g. Query by Beatboxing [6]) and are not suited for general query by vocal imitation of general sounds.

Synthassist [7] is a search tool for music synthesizer sounds. It compares vocal imitations to a library of synthesizer sounds by creating temporal vectors of standard audio features (e.g. mel-frequency cepstral coefficients) and using an edit-distance to compare the query to audio files in the database. However, in recent years, better retrieval accuracy has been achieved by deep learning models which learn the relevant feature sets during training. The state-of-the-art in query by vocal imitation was improved when convolutional auto encoders were applied to the problem by multiple research groups [8, 9, 10, **?**].

Zhang and Duan further improved upon the QBV accuracy of CAEs with IMINET [11], a two-tower feed-forward convolutional network. In their work, each of the two inputs (an audio file from a database and an imitation to be compared) is encoded by one of the two towers and the output of both towers are input to a fully-connected network that produces a similarity measure between the two audio files. Their most recent work, and the current state of the art in QBV, is TL-IMINET [4], which they call a Siamese-style architecture since the tower that takes the vocal imitation as input has a different architecture than the tower accepting sounds from the collection (as opposed to fully Siamese nets, where the towers share weights and architecture). In all of the recent work, pairwise loss was used in training.

In the domain of image search, Wang et al. [12] proposed a triplet loss method for learning feature models of images where training examples consist of a query and two rank-ordered database elements. This allowed learning fine grained distinctions between images of the same class. They showed this approach outperformed existing models that used hand-crafted features. However they did not compare the triplet loss training approach to pairwise training. They applied their work to a three-tower deep net model but did not show how it could be applied to a two-tower model. Also, their queries (photos) were drawn from the same data as their search results (photos from the same collection as the queries).

Our work combines and builds on ideas from Wang et al.[12] and Zang et al.[4] We illustrate how to adapt an existing two-tower architecture to be trained using triplet loss, instead of forcing the use of an altogether new architecture. We apply this approach to train a system to do pairwise comparison between very different classes of sound objects: vocal imitations and reference audio recordings. An analogous task in the image domain would be to search a set of photos using hand-drawn images as the queries. We then compare the effectiveness of triplet loss to pairwise loss in training a network to solve a ranking problem.

3. METHODS

We assume a set of sound recordings R where r is a recording in the set. The search key is a vocal imitation v of some file in the set, known as the target, t. Given a similarity measure $s(v, r)$ that returns a similarity value in the range [0,1] for any recording, we can provide an ordering of the files in R, based on similarity. This ordering is used as a ranking returned by a search engine. The more consistently the target is returned as a highly ranked recording, the better the search engine. The question then becomes how to create a similarity measure that will consistently rank that target highly.

3.1. TL-IMINET

TL-IMINET [4] is the most successful system, to-date, for QBV. In that work they learn the similarity measure using a neural network with two convolutional towers: one tower for a vocal imitation and the other for a recording from the data set. These towers both learn embeddings that are fed into a fully connected network. A trained network takes a vocal imitation and a reference audio file as input and outputs a value in the range [0,1], where 1 indicates a perfect match.

Our goal is not to design a superior network architecture, but to develop a superior training method. Therefore, we use the exact architecture and audio encoding used in the TL-IMINET paper. We provide an overview of the network structure and audio encoding below. More detail can be found in [4].

The input to the vocal imitation tower is 4 seconds of audio (vocal imitations in the data set are typically less than 4 seconds long). This is encoded as a 39-band mel-spaced spectrogram with 8.33 ms for both the window size and hop size. The resulting input spectrogram has 39 frequency bins and 482 time bins. The vocal imitation tower has three convolutional layers. For the first two convolutional layers, each layer has 48 filters with ReLU activations. Both layers are followed by a 6×6 pooling layer. The third convolutional layer also has 48 filters and a receptive field of 6×6. It is followed by a 1×2 pooling layer with 1 in frequency and 2 in time.

The general audio tower is passed reference recordings truncated to 3 seconds and converted into a mel spectrogram with 23 ms window size and 23 ms hop size. This leads to an input dimensionality of 128 mel-frequency bands by 128 time steps. This is input to a tower with 3 convolutional layers. The first convolutional layer has 24 filters with a receptive field of 5×5, and followed by a ReLU activation function. This is fed into a 2×4 (both shape and stride) max-pooling layer with 2 in frequency and 4 in time. The second and third convolutional layers each have 48 filters with a 5×5 receptive field. The second convolutional layer is followed by a 2×4 pooling layer. Unlike the vocal imitation tower, no pooling layer follows the third convolutional layer.

The embeddings output from each convolutional tower are concatenated and passed into 2 fully connected layers, which calculate a distance between each vector.

3.2. Pairwise loss

TL-IMINET is a Siamese-style network and uses a pairwise loss function. This is the typical loss function used for Siamese and Siamese-style networks. The training pairs consist of a vocal example and a recording from the data set. If the recording in the pair is the target, then the label is 1. If the recording is not the target, the label is 0. The loss function used is binary cross entropy and the goal is for the similarity measure to output 1 for the target of a vocal query and 0 for all other recordings.

While the error signal described above has proven useful in many cases, it may not always correlate strongly with the desired behavior of the system when the goal is to rank order a set of items by similarity to a single query example. For ranking problems, it is not important that the output of the similarity measure be either 1 or 0, for any particular pair. In fact, this goal may even be counterproductive for ranking problems, as we now show. Consider a case with 100 recordings in the data set. Assume the similarity function returned a value in the range [.9, 1] for all recordings and the target is the only recording to get a similarity of 1. The target would be ranked first, which is perfect performance. The error

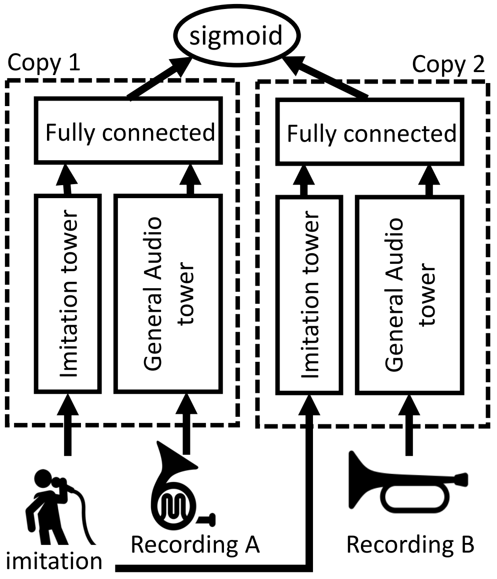

Figure 1: Triplet loss training configuration of the network used to measure similarity between vocal imitations and reference audio files. Two copies of a two-tower Siamese-style network (TL-IMINET) are used. Weights are tied between the two copies. A training example consists of a vocal imitation, the target, and a distractor. The desired output is 1 if the target is recording A and 0 if the target is recording B.

signal, however, would show large amounts of error for 99 out of 100 recordings, since all non-target recordings would be ranked .9 or higher, when 0 is the correct label. Consider another case: the similarity function returns values only in the range $[0, .01]$ and the target is the only file to receive a 0. Here, the error would be very low, since 99 out of 100 recordings were very closed to the training label output of 0, even though the target would be ranked last.

3.3. Triplet loss: an error signal more suited to ranking

In the previous section, we illustrated how pairwise loss may not provide the ideal error signal for ranking problems using two-tower networks. How, then, can the error signal be tied more closely to the desired ranking behavior? In this work, we adopt the idea of *triplet loss*, which has been used successfully to train a three-tower convolutional network to perform fine-grained similarity measurements in the domain of still images [12]. In this paradigm, a training example consists of a triplet (v, a, b). If recording a should be ranked closer to the vocal imitation v, the label is 1. If b should be ranked closer than a, the label is 0. This allows the use of binary cross entropy, but explicitly takes ranking between pairs into account.

We apply this loss function to a Siamese-style network architecture. In our case, that network architecture is TL-IMINET; however, the same approach can be applied to any Siamese or Siamese-style network. Two copies of the Siamese-style architecture are used. Weights are tied between the two copies. The vocal imitation is input to both copies. Recording A is passed to one copy, and recording B is passed to the other copy. The output of both copies is passed to a single sigmoid node that outputs a value in the range 0 to 1. This is illustrated in Figure 1.

The two input weights of the sigmoid are fixed to be +50 and -50, with a bias of 0. This yields the function $\sigma(50a - 50b)$. With this configuration, the output tends to 1 if the output from recording A is higher than B and tends towards 0 if B is higher than A. This configuration allows for triplet-loss training. [4]

Once trained, either copy of the TL-IMINET architecture can be extracted and used in testing to estimate similarity between a vocal imitation query and a recording in the database. Recordings can be ranked by similarity as is done using the two tower TL-IMINET.

By training in this fashion we can apply a triplet loss approach to train a two-tower network designed for pairwise loss. This also allows for a truly meaningful comparison between triplet and pairwise loss, since both training approaches modify the exact same number of weights, and the testing architecture is identical for both approaches.

4. EXPERIMENTAL DESIGN

We have argued that using triplet loss will result in a error signal that is more correlated with the goal (ranking the right answer highly) than happens with the pairwise error signal used in previous vocal imitation search work. We tested this hypothesis by measuring the correlation between improving on the loss function and improving search results as training progresses. To ensure a controlled experiment we used a TL-IMINET architecture, trained using pairwise training and compare the results to an identical TL-IMINET with the same initialization weights, trained using triplet loss. We repeated the comparison on a variety of data splits and with a variety of initializations and measured the statistical difference between the two training approaches. We now describe the experiment in detail.

4.1. Data Set

For this work, we used the Vocal Imitation Set [13], a collection of crowd-sourced vocal imitations of a set of 302 classes of sound. The classes were drawn from Google's AudioSet ontology [14]. Each sound class has an average of 10 clean, single-sound recordings taken from FreeSound (e.g. 10 police siren recordings). A single one of the 10 recordings in each class was used as a reference recording (the *target*) for the vocal imitations. Given a reference recording as the target, Amazon Mechanical Turk workers were asked to record a vocal imitation of the target. Recorded imitations were evaluated by expert listeners and only those recordings that were of sufficiently high quality were used. This resulted in 5,601 high quality imitations of 302 sound classes, or roughly 19 imitations per sound class. For more detail on this data set, please see [13].

4.2. A single trial

A recording of a vocal imitation of a sound is the *query*. The *target* is the single audio file that the query is an imitation of. A *distractor* is a file in the collection that is not the search goal.

In a single trial we randomly select 10 sound classes from the vetted Vocal Imitation data set of 302 sound classes. Each sound class contains an average of 19 imitations and 10 reference recordings. This results in a set of roughly 100 reference files and 190

[4] Note the value of 50 is simply selected to be sufficiently large to saturate the sigmoid activation function, and is not a magic number.

imitations. Training examples for the pairwise loss function require one imitation and one reference (either the target file, or some other file), resulting roughly 19,000 unique training pairs, of which 190 are positive pairs and 18810 are negative. Each training example for the triplet loss function requires one imitation and two reference recordings (the target + distractor), also resulting in roughly 1,881,000 unique training triplets. This data is split into validation (30%) and training (70%) data.

A coarse grained comparison in triplet loss is one where the distractor is drawn from a different sound class than the target. In the case of triplet loss, there are many more coarse grained examples than fine grained examples. Balancing coarse (across class) and fine (within class) distinctions is desirable. Therefore, on each epoch, we train using all the fine grained examples and an equally sized, randomly selected subset of the coarse grained examples. In the case of pairwise lose, there are many more negative examples than positive examples. Similarly, on each epoch, we train using all the positive examples and an equally sized, randomly selected subset of the negative examples.

For each trial, we randomly initialized a TL-IMINET network. We trained the network twice from the same initialization weights, once with triplet loss and once with pairwise loss. We used the ADAM optimization function [15]. See Section 3 for details of the loss functions. Each network was trained for 300 epochs. At each epoch, we use the trained network as a similarity measure to rank the target for each of the vocal imitations among the 100 reference files. The mean rank of the target, as well as the loss function, is recorded at each epoch for both the training and validation data.

The code used to run these trials can be found at our Github repository[5].

5. RESULTS

We ran 28 trials and measured Pearson's correlation coefficient between the ranking results and loss curves for both loss function over the 300 training epochs in each trial. See Figure 2 for loss and rank curves for a representative trial and their correlations. It is clear that triplet loss correlates much better with ranking results than pairwise loss does. This trial was chosen because its correlation between the loss function and ranking results was close to the mean correlation over all the trials, for both pairwise and triplet loss.

We performed a Wilcoxon signed-rank test on the 28 trials, comparing the Pearson correlation of triplet loss to ranking results with the Pearson correlation of pairwise loss to ranking results. The resulting p-value of 5.3×10^{-6}, indicates the improvement in correlation between rank and loss gained from switching to a triplet loss function is statistically significant. Figure 3 shows the distribution of the correlation.

6. CONCLUSIONS

We have shown how, with a simple modification to training, any two tower Siamese-style network can be adapted to learn using triplet loss. We have shown that, for QBV on a dataset of audio files, triplet loss correlates much more closely with ranking results than pairwise loss does. This higher correlation means that the network learns to perform a task closely related to the end-goal of search. This approach to training is promising in that it can be easily applied to any existing Siamese-style network.

[5] https://git.io/fNMfe.

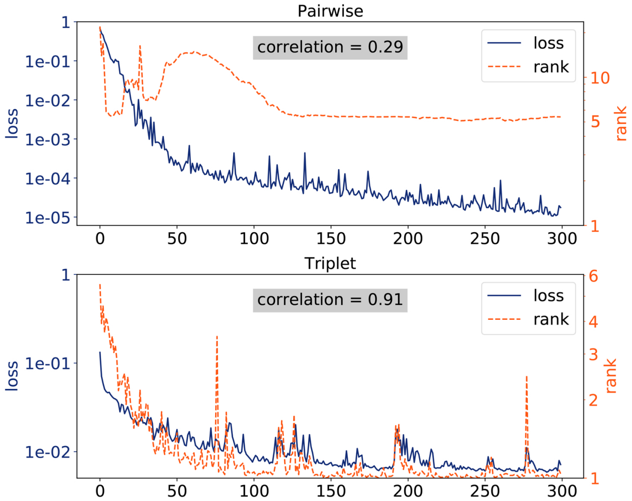

Figure 2: One representative trial. Training loss as a function of training epoch is shown in blue. We measured mean rank across 133 queries in a 100-file search on the training set. This curve is shown in orange. Lower ranking is better. The upper panel shows traditional pairwise loss. The lower shows triplet loss. Lower loss is better. Correlation is the Pearson correlation between the loss and the target rank.

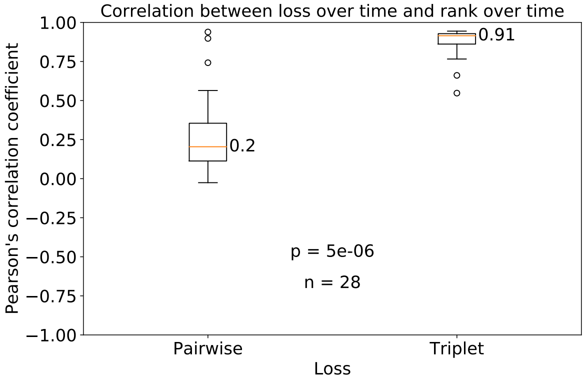

Figure 3: Distribution of the correlation between the search rank of the target file and the the loss function used in training. The value for each trial is the Pearson correlation coefficient between the loss function and the rank of the target. Higher correlation is better. Numbers next to boxes are median values.

7. REFERENCES

[1] "The complete sfx platform." [Online]. Available: https://www.getsoundly.com/

[2] G. Lemaitre and D. Rocchesso, "On the effectiveness of vocal imitations and verbal descriptions of sounds," *The journal of the Acoustical Society of America*, vol. 135, no. 2, pp. 862–873, 2014.

[3] G. Lemaitre, O. Houix, F. Voisin, N. Misdariis, and P. Susini,

"Vocal imitations of non-vocal sounds," *PloS one*, vol. 11, no. 12, p. e0168167, 2016.

[4] Y. Zhang and Z. Duan, "Visualization and interpretation of siamese style convolutional neural networks for sound search by vocal imitation (to be presented)," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[5] A. Huq, M. Cartwright, and B. Pardo, "Crowdsourcing a real-world on-line query by humming system," in *Proceedings of the Sixth Sound and Music Computing Conference (SMC 2010)*, 2010.

[6] A. Kapur, M. Benning, and G. Tzanetakis, "Query-by-beat-boxing: Music retrieval for the dj," in *Proceedings of the International Conference on Music Information Retrieval*, 2004, pp. 170–177.

[7] M. Cartwright and B. Pardo, "Synthassist: Querying an audio synthesizer by vocal imitation." in *NIME*. Citeseer, 2014, pp. 363–366.

[8] Y. Zhang and Z. Duan, "Imisound: An unsupervised system for sound query by vocal imitation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2269–2273.

[9] ——, "Retrieving sounds by vocal imitation recognition," in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*. IEEE, 2015, pp. 1–6.

[10] A. Mehrabi, K. Choi, S. Dixon, and M. Sandler, "Similarity measures for vocal-based drum sample retrieval using deep convolutional auto-encoders," *arXiv preprint arXiv:1802.05178*, 2018.

[11] Y. Zhang and Z. Duan, "Iminet: Convolutional semi-siamese networks for sound search by vocal imitation," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2017 IEEE Workshop on*. IEEE, 2017, pp. 304–308.

[12] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[13] B. Kim, M. Ghei, B. Pardo, and Z. Duan, "Vocal imitation set: a dataset of vocally imitated sound events using the audioset ontology," in *Proceedings of the 2018 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE2018)*, 2018.

[14] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 776–780.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.