Unsupervised Feature Learning for Audio Analysis

Matthias Meyer, Jan Beutel, Lothar Thiele
Computer Engineering and Networks Laboratory
ETH Zurich
Zurich, Switzerland
`{matthias.meyer, beutel, thiele}@tik.ee.ethz.ch`

Abstract

Identifying acoustic events from a continuously streaming audio source is of interest for many applications including environmental monitoring for basic research. In this scenario neither different event classes are known nor what distinguishes one class from another. Therefore, an unsupervised feature learning method for exploration of audio data is presented in this paper. It incorporates the two following novel contributions: First, an audio frame predictor based on a Convolutional LSTM autoencoder is demonstrated, which is used for unsupervised feature extraction. Second, a training method for autoencoders is presented, which leads to distinct features by amplifying event similarities. In comparison to standard approaches, the features extracted from the audio frame predictor trained with the novel approach show 13 % better results when used with a classifier and 36 % better results when used for clustering.

1 Introduction

The advance of low-power mobile sensors leads to new applications for real-time environmental monitoring, which include acoustic event detection (AED) systems using wireless sensor networks, e.g. for vehicle classification (Duarte & Hu (2004)). Acoustic events can be effectively classified by convolutional neural networks (CNN) as shown by Espi et al. (2015), if a large labeled training dataset is available (Hershey et al. (2016)). However, for some application scenarios, e.g. in micro-seismic acoustic emission monitoring (Girard et al. (2012)), the advantages of mentioned CNNs are diminished due to the lack of ground truth. In this scenario distinguishing between relevant events in an audio stream and non-relevant events is non-trivial, since the domain-specific categories of these acoustic events are not known a priori. Unsupervised feature learning provides a mean to gather useful information from such unexplored datasets, which has been shown for video (Srivastava et al. (2015)) and audio (Honglak Lee (2009)) analysis. Recently, convolutional long short-term memory (ConvLSTM) layers have proven to be effective as feature extractors for time series data (Xingjian et al. (2015)) and for unsupervised video analysis (Finn et al. (2016)). Except for speech recognition (Zhang et al. (2016)), ConvLSTM layers have not been used for audio analysis, despite their advantages for time series data.

As a consequence, in this paper a novel audio frame predictor (AFP) using ConvLSTM layers is presented, which is used to learn representative features of acoustic events. Second a novel approach to train the AFP in regard to generating and extracting distinct features is presented which diversifies the features based on inter-sample similarities.

2 Approach

An autoencoder encodes an arbitrary input signal into a compressed intermediate representation (code) from which it tries to recreate the original signal (Hinton & Salakhutdinov (2006)). In this section it is assumed that an autoencoder using convolutional layers, as depicted in Figure 1, can be trained such that each channel of the code contains information about a specific feature of acoustic events. Then, also the mean of this channel would define an acoustic feature and per-channel average
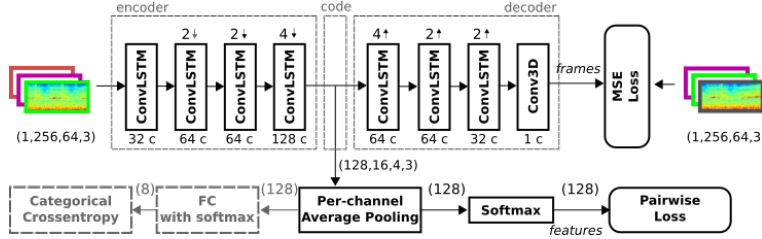
Figure 1: Model of the audio frame predictor, which is trained to predict the next frame of the mel-spectrogram input. Multiple ConvLSTM layers are used which differ by number of channels (e.g. 32 c) and down-/upsampling (e.g. $2 \downarrow$). Per-channel average pooling reduces the code to a lower dimensional feature vector on which a pairwise loss is applied. For evaluation a simple classifier, consisting of a fully connected (FC) layer, classifies events based on the extracted features.

pooling can be used to reduce the K-channel code to a K-dimensional vector (feature vector). A softmax function transforms this vector into a categorical distribution (from here on called feature distribution)

In this work, similarities between two audio samples are characterized by the statistical distance between their feature distributions, a concept similar to Hsu & Kira (2015). Since training of a neural network is typically done in batches of size N, a pairwise loss function $L(n,m)$ for the *features* output can be defined as:

$$L(n,m) = \begin{cases} KL(B^{(n)} \| B^{(m)}), & \text{if } f_{n,m} < \text{threshold} \\ \max(0, margin - KL(B^{(n)} \| B^{(m)})), & \text{else} \end{cases} \tag{1}$$

$$f_{n,m} = \frac{KL(B^{(n)} \| B^{(m)})}{\frac{1}{N^2} \sum_{o=0}^{N-1} \sum_{p=0}^{N-1} KL(B^{(o)} \| B^{(p)})} \tag{2}$$

where $B^{(n)}$ defines the feature distribution of the n-th element of the batch. KL is the Kullback-Leibler divergence and the two parameters *threshold* and *margin* can be adjusted based on the actual data. $f_{n,m}$ is a measure for the similarity of two elements of the batch. Therefore thresholding $f_{n,m}$ means that the loss $L(n,m)$ penalizes events based on similarity, for similar events the feature distribution gets more confined while dissimilar events are shifted away from each other. Thus, no (weak) labels are required in contrast to Hsu & Kira (2015). The overall loss per batch is $\sum_{n=0}^{N-1} \sum_{m=0}^{N-1} L(n,m)$, which can be backpropagated through the network and affects the weights of the encoder. The total loss for the encoder weights is the sum of the autoencoder loss and the pairwise loss. The autoencoder enforces a meaningful representation while $L(n,m)$ amplifies similarities and dissimilarities. Thus, this method optimizes for inter-sample similarities in comparison to the method from Xie et al. (2016), which optimizes for fixed cluster centers.

For the actual implementation an autoencoder is used as an AFP, which is illustrated in Figure 1. Data frames of length $T_f = 2.56$ s are extracted from the audio waveform with a hopsize of $T_f/4$ and transformed into a mel-spectrogram. The AFP considers three consecutive frames and predicts the following frame. Each ConvLSTM layer uses a 3x3 filter kernel and is followed by a ReLu activation and batch normalization. The last convolutional layer (3D convolution) applies a 3x3x1 kernel with a linear activation function. Keras (Chollet (2015)) and Tensorflow (Martn Abadi et al. (2015)) are used for the implementation.

The training with batchsize $N = 16$ is done in two steps. First, only the AFP is trained by minimizing the mean squared error (MSE) between the *frames* output of the AFP and the true next frame, initializing the weights of the whole system. After training the AFP for 6 epochs the system is trained for another 3 epochs by minimizing the above mentioned pairwise loss at the *features* output and the MSE at the *frames* output, simultaneously. In the next section a system trained with this two-step procedure (PairLoss) will be compared against a system trained only by minimizing the MSE at the *frames* output for 9 epochs (w/o PairLoss).

2

3 EVALUATION

An acoustic event dataset (AED) from Takahashi et al. (2016) was used for evaluation. In Figure 2 eight categories are listed which were chosen to have an approximate uniformly distributed number of samples. The dataset has been partitioned into training (75%) and test set (25%). The number of code channels has been set to $K = 128$ to maintain an accurate prediction and to provide enough features to explain the data.
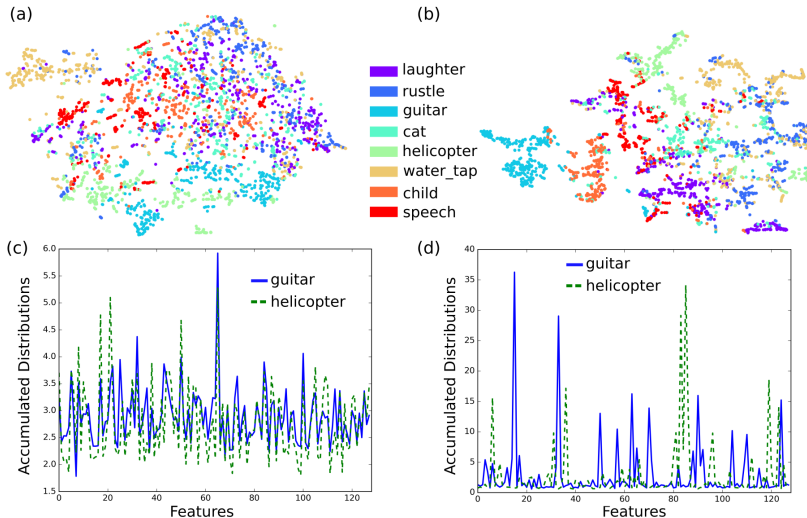

Figure 2: t-SNE embedding (Maaten & Hinton (2008)) of the feature distributions for a system trained without PairLoss (a) and with PairLoss (b). Accumulated feature distributions for all training samples of the category *guitar* and *helicopter* for a system trained without PairLoss (c) and with PairLoss (d). Interactive version at: http://people.ee.ethz.ch/matthmey/

Table 1: Test accuracies for classifier and clustering algorithm. Note: The reference classifier uses supervised training for all weights, the others only for the fully connected layer

	Classifier		Clustering	
Reference	w/o PairLoss	w/ PairLoss	w/o PairLoss	w/ PairLoss
(90.85 %)	69.59 %	**78.94 %**	38.32 %	**52.23 %**

First, it is shown that the feature vectors contain useful information by learning a mapping from the 128-dimensonal feature vectors to the 8 categories. A simple classifier, consisting of only one fully connected layer, as depicted in Figure 1, is trained by minimizing the cross-entropy loss between the classifier's output and the ground-truth labels. The weights of the encoder are fixed. The PairLoss system outperforms the one without PairLoss on the test set, which can be attributed to the different feature distributions, as illustrated in Figure 2 (c) and (d). The plot displays the accumulated feature distributions for two event categories. In contrast to standard training, it becomes apparent that for PairLoss training two different event categories show distinct patterns for their feature distributions, which means that only a few channels contain event information. It is also shown that using a system trained with PairLoss shows better clustering accuracies than a system trained without, as presented in Table 1. Although the system can potentially be expanded to perform clustering itself, for this evaluation only k-means clustering (8 clusters) is applied to the t-SNE embedding of the feature distributions. The best cluster assignment is chosen using the Hungarian algorithm (Kuhn (1955)).

In this work a novel approach to train autoencoders has been used in a novel unsupervised feature extraction system for audio data. It has been shown experimentally that training with the new method generates distinct features which increases classification accuracy by 13 % and clustering accuracy by 36 %.

REFERENCES

Franois Chollet. *Keras*. GitHub, 2015. URL `https://github.com/fchollet/keras`.

Marco F. Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004.

Miquel Espi, Masakiyo Fujimoto, Keisuke Kinoshita, and Tomohiro Nakatani. Exploiting spectro-temporal locality in deep learning based acoustic event detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):26, December 2015.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised Learning for Physical Interaction through Video Prediction. *Advances in Neural Information Processing Systems 29*, pp. 64–72, 2016.

L. Girard, J. Beutel, S. Gruber, J. Hunziker, R. Lim, and S. Weber. A custom acoustic emission monitoring system for harsh environments: application to freezing-induced damage in alpine rock walls. *Geoscientific Instrumentation, Methods and Data Systems*, 1(2):155–167, 2012.

Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. CNN Architectures for Large-Scale Audio Classification. *arXiv: 1609.09430*, September 2016.

G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006.

Peter T. Pham Honglak Lee. Unsupervised feature learning for audio classification using convolutional deep belief networks. *NIPS*, pp. 1096–1104, 2009.

Yen-Chang Hsu and Zsolt Kira. Neural network-based clustering using pairwise constraints. *arXiv: 1511.06321*, November 2015.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, March 1955.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Martn Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vigas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. URL `http://tensorflow.org/`.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs. In *ICML*, pp. 843–852, 2015.

Naoya Takahashi, Michael Gygli, Beat Pfister, and Luc Van Gool. Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Recognition. In *Proc. Interspeech 2016*, San Fransisco, 2016.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning (ICML)*, 2016.

S. H. I. Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pp. 802–810, 2015.

Yu Zhang, William Chan, and Navdeep Jaitly. Very Deep Convolutional Networks for End-to-End Speech Recognition. *arXiv: 1610.03022*, October 2016.