SEQUENCE TO SEQUENCE AUTOENCODERS FOR UNSUPERVISED REPRESENTATION LEARNING FROM AUDIO

Shahin Amiriparian[1,2,3], *Michael Freitag*[1], *Nicholas Cummins*[1,2], *Björn Schuller*[2,4]

[1] Chair of Complex & Intelligent Systems, Universität Passau, Germany
[2] Chair of Embedded Intelligence for Health Care & Wellbeing, Augsburg University, Germany
[3] Machine Intelligence & Signal Processing Group, Technische Universität München, Germany
[4] GLAM – Group on Language, Audio & Music, Imperial College London, London, UK
shahin.amiriparian@tum.de

ABSTRACT

This paper describes our contribution to the Acoustic Scene Classification task of the IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2017). We propose a system for this task using a recurrent sequence to sequence autoencoder for unsupervised representation learning from raw audio files. First, we extract mel-spectrograms from the raw audio files. Second, we train a recurrent sequence to sequence autoencoder on these spectrograms, that are considered as time-dependent frequency vectors. Then, we extract, from a fully connected layer between the decoder and encoder units, the learnt representations of spectrograms as the feature vectors for the corresponding audio instances. Finally, we train a multilayer perceptron neural network on these feature vectors to predict the class labels. In comparison to the baseline, the accuracy is increased from 74.8 % to 88.0 % on the development set, and from 61.0 % to 67.5 % on the test set.

Index Terms— deep feature learning, sequence to sequence learning, recurrent autoencoders, audio processing acoustic scene classification

1. INTRODUCTION

Machine learning algorithms for audio processing typically operate on expert-designed feature sets extracted from the raw audio signals. Arguably among the most widely used features are Mel-band energies and features derived from them, such as *Mel Frequency Cepstral Coefficients* (MFCCs). Both feature spaces are widely used in acoustic scene classification [1–3], with the former being employed in the DCASE 2017 Challenge baseline system [4], and the later being the low level feature space used by the winners of the DCASE 2016 acoustic scene classification challenge [5].

It takes considerable effort and human intervention to manually engineer such features for a specific purpose, which then may not perform well on unrelated tasks. Further, many feature spaces such as MFCCs are non-task specific and are equally adept in a range of audio and speech-based classification tasks [6–8]. For these reasons, among others, unsupervised representation learning has recently gained considerable popularity as a highly effective substitute for using conventional feature sets [9, 10]. Representation learning with deep neural networks (DNNs), in particular, has been shown to be superior to feature engineering for a wide variety of tasks, including speech recognition [6, 11] and music transcription [11, 12]. However, the advantages of deep representation learning are yet to be fully established for acoustic scene classification.

Audio sequences are typically varying length signals; this presents a drawback for deep representation learning using architectures such as convolutional neural networks (CNNs) which typically require inputs of fixed dimensionality. Furthermore, typical DNN architectures used for representation learning, such as stacked autoencoders or Restricted Boltzmann Machines, do not explicitly account for the inherent sequential nature of acoustic data [9]. For the learning of fixed-length representations of variable-length sequential data, sequence to sequence learning with recurrent neural networks (RNNs) has been proposed in machine translation [13,14]. Moreover, RNNs have been successfully used in a range of audio-based classification tasks such as novelty detection [15], scene classification [16], and speech recognition [17].

In this paper, we extend the RNN encoder-decoder model proposed by Cho et. al. [13] to develop a recurrent sequence to sequence autoencoder for deep unsupervised representation learning suitable for use with acoustic data. Sequence to sequence autoencoders have been employed for unsupervised pretraining of RNNs, with promising results on text classification and image recognition tasks [18], as well as machine translation [19]. Variational sequence to sequence autoencoders have been used to learn representations of sentences, and to generate new sentences from the latent space [20]. Furthermore, de-noising recurrent autoencoders have been used for reverberated speech recognition [21], moreover, in this approach, variable-length representations of audio are learnt. Despite this success in a range of applications, to the best of the authors' knowledge, there is no previous work on extracting the learnt representations from sequence to sequence autoencoders for further processing, such as audio classification.

The remainder of this paper is organised as follows: In Section 2, we describe our proposed recurrent sequence to sequence autoencoder approach in detail. Subsequently, we outline our experimental evaluation and results thereof for the DCASE 2017 Acoustic Scene Classification challenge in Section 3. Finally, concluding remarks and our future work plans are given in Section 4.

2. APPROACH

A high-level overview of our system is given in Figure 1. First, mel-spectrograms are extracted from the raw audio files (cf. Figure 1a). Subsequently, a recurrent sequence to sequence autoencoder is trained on these spectra (cf. Figure 1b), which are viewed as time-dependent sequences of frequency vectors. The learnt representations of the spectrograms are then extracted for use as feature

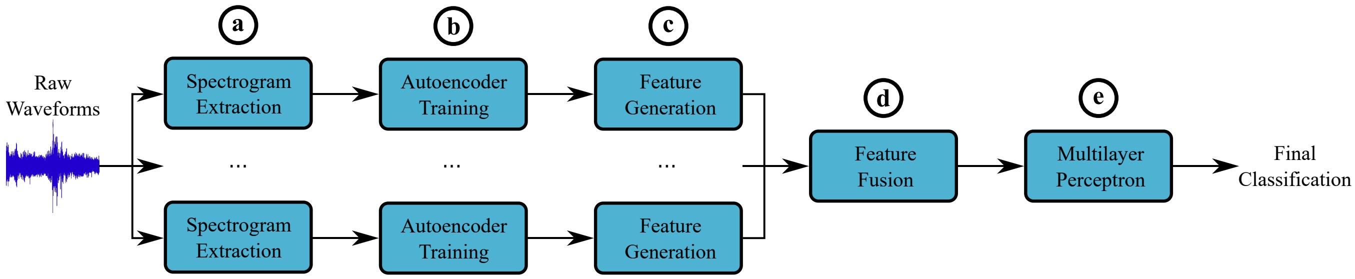


Figure 1: Illustration of the proposed representation learning and classification approach. Except for the final classification, the approach is entirely unsupervised. A detailed account of the procedure is given in Section 2.

vectors for the corresponding instances (cf. Figure 1c). This step is repeated for the different spectral representations made possible by the stereo recordings provided in the challenge dataset, with the resulting set of learnt representation – per audio instance – being concatenated together (cf. Figure 1d). Finally, we train a multilayer perceptron (MLP) (cf. Figure 1e) on the fused feature vectors to predict the labels of instances.

2.1. Spectrogram Extraction

First, the power spectra of audio samples are extracted using periodic Hann windows with width w and overlap $0.5\,w$. From these, we then compute a given number N_m of log-scaled Mel frequency bands. Finally, the mel-spectra are normalised to have values in $[-1;1]$, since the outputs of the recurrent sequence to sequence autoencoder are constrained to this interval.

The challenge corpus contains audio samples which have been recorded in stereo [4]. In such data sets, there may be instances in which important information related to the class label has been captured in only one of the two channels. Following the winners of the DCASE 2016 acoustic scene classification challenge [5], we thus extract mel-spectrograms from each individual channel, as well as from the mean and difference of the two channels.

We extract separate sets of mel-spectrograms for different parameter combinations, each containing one mel-spectrogram per audio sample. As illustrated in Figure 1, representations are learnt independently on different sets of mel-spectrograms, and we investigate feature-level fusion of these representations.

2.2. Recurrent Sequence to Sequence Autoencoders

We use recurrent sequence to sequence autoencoders to learn representations of the extracted mel-spectra in an unsupervised manner [13,14]. An illustration of the structure of these autoencoders is shown in Figure 2. Mel-spectra are viewed as a time-dependant sequence of frequency vectors in $[-1;1]^{N_m}$, each of which describes the amplitudes of the N_m Mel frequency bands within one audio frame. This sequence is fed to a multilayered *encoder* RNN, which updates its hidden state in each time step based on the input frequency vector. Therefore, the final hidden state of the encoder RNN contains information about the whole input sequence. This final hidden state is transformed using a fully-connected layer, and another multilayered *decoder* RNN is used to reconstruct the original input sequence from the transformed representation.

The encoder RNN consists of N_l layers, each containing N_u Gated Recurrent Units (GRUs) [13]. During our initial system design phase we conducted experiments using Long Short-Term Memory cells instead of GRUs. However, we observed that this

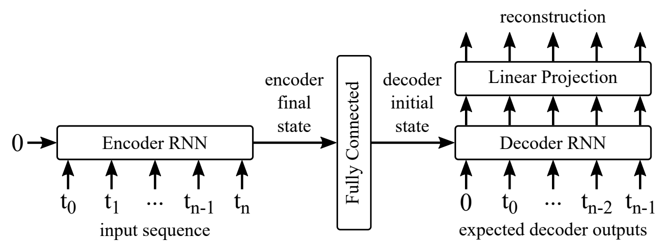


Figure 2: An overview of the implemented recurrent autoencoder.

did not lead to improvements in system performance. The hidden states of the encoder GRUs are initialised to zero for each input sequence, and their final hidden states in each layer are concatenated into a one-dimensional vector. This vector can be viewed as a fixed-length representation of a variable-length input sequence, with dimensionality $N_l \cdot N_u$, if the encoder RNN is unidirectional, and dimensionality $2 \cdot N_l \cdot N_u$ if it is bidirectional.

The representation vector is then passed through a fully connected layer with hyperbolic tangent activation. The output dimensionality of this layer is chosen in such a way that it can be used to initialise the hidden states of the decoder RNN.

The decoder RNN contains the same number of layers and units as the encoder RNN. Its task is the frame-by-frame reconstruction of the input mel-spectrogram, based on the representation which was used to initialise the hidden states of the decoder RNN. At the first time step, a zero input is fed to the decoder RNN. During subsequent time steps t, the expected decoder output at time $t-1$ is fed as input to the decoder RNN [14]. Stronger representations could potentially be obtained by using the actual decoder output instead of the expected output, since this reduces the amount of information available to the decoder. However, during initial experiments we observed that our approach greatly accelerates model convergence with negligible effects on representation quality.

The outputs of the decoder RNN are passed through a single linear projection layer with hyperbolic tangent activation at each time step in order to map the decoder RNN output dimensionality to the target dimensionality N_m. The weights of this output projection are shared across time steps. In order to introduce greater short-term dependencies between the encoder and the decoder, our decoder RNN reconstructs the reversed input sequence [14,22].

Autoencoder training is performed using the root mean square error (RMSE) between the decoder output and the target sequence as the objective function. Dropout is applied to the inputs and outputs of the recurrent layers, but not to the hidden states. Once training is complete, the activations of the fully connected layer are extracted as the learnt representations of spectrograms.

2.3. Multilayer Perceptron Classifier

A multilayer perceptron, similar to the one used in the challenge baseline system [4], is employed for classification. Our MLP contains two hidden fully-connected layers with rectified linear activation, and a softmax output layer. The hidden layers contain 150 units each, and the output layer contains one unit for each class label. Training is performed using cross entropy between the ground truth and the network output as the objective function, with dropout applied to all layers except the output layer. A range of different classifiers were tested during our initial experimentation. However, we observed that more sophisticated classification paradigms did not aid our overall system performance.

3. EXPERIMENTAL SETTINGS AND RESULTS

3.1. Database

The DCASE 2017 acoustic scene classification challengeis carried out on the TUT Acoustic Scenes 2017 data set [4]. This data set contains binaural audio samples of 15 acoustic scenes recorded at distinct geographic locations. For each location, between 3 and 5 minutes of audio were initially recorded and then split into 10 second segments. The development set for the challenge contains 4 680 instances, with 312 instances per class, and the evaluation set contains 1 620 instances with unknown labels.

A four-fold cross-validation setup is provided by the challenge organisers for the development set. In each fold, roughly 75 % of the samples are used as the training split, and the remaining samples are used as the evaluation split. Samples from the same original recording are always included in the same split. For further detail on the challenge data and the cross fold validation set-up, the interested reader is referred to [4].

3.2. Common Experimental Settings

We have implemented the representation learning approach outlined above as part of the AUDEEP toolkit[1] for deep representation learning from audio. AUDEEP is implemented in Python, and relies on TENSORFLOW[2] for the core sequence to sequence autoencoder and MLP implementations.

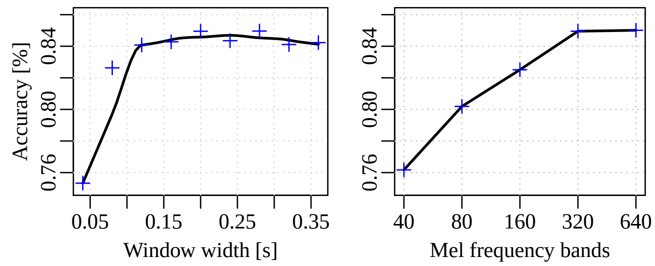
Both the autoencoders and MLPs are trained using the Adam optimiser with a fixed learning rate of 0.001 [23]. Autoencoders are trained for 50 epochs in batches of 64 samples, and we apply 20 % dropout to the outputs of each recurrent layer. Furthermore, we clip gradients with absolute value above 2 [14]. The MLPs used for classification are trained for 400 epochs without batching or gradient clipping, and 40 % dropout is applied to the hidden layers. Features are standardised to have zero mean and unit variance during MLP training, and the corresponding coefficients are used to transform the validation data.

3.3. Hyperparameter Optimisation

Our proposed approach contains a large number of adjustable hyperparameters, which prohibits an exhaustive exploration of the parameter space. Instead, we select suitable values for the hyperparameters in stages, using the results of our preliminary experiments to bootstrap the process. During these experiments, we observed

[1] https://github.com/auDeep/auDeep

[2] https://www.tensorflow.org/



(a) (b)

Figure 3: Classification accuracy on the development set for different FFT window widths (a), and different numbers of Mel frequency bands (b). A detailed account of the experiments leading to these results is given in Section 3.3.

that very similar parameter choices lead to comparable performance on spectrograms extracted from different combinations of the audio channels (mean, difference, left and right). We therefore performed hyperparameter optimisation on the mean-spectrograms only, and used the resulting parameters for the other spectrogram types.

In the first stage, we selected a suitable autoencoder configuration, i. e. the optimal number of recurrent layers N_l, the number of units per layer N_u, and either unidirectional or bidirectional encoder and decoder RNNs. In this stage, autoencoders are trained on mel-spectrograms extracted with window width $w = 0.16$ seconds, window overlap $0.5\,w = 0.08$ seconds, and $N_m = 320$ Mel frequency bands, without amplitude clipping. These choices proved to be reasonable during our preliminary evaluation. We exhaustively evaluated $N_l \in \{1, 2, 3\}$, $N_u \in \{16, 32, 64, 128, 256, 512\}$ and all combinations of uni- or bidirectional encoder and decoder RNNs. The highest classification accuracy was achieved when using $N_l = 2$ layers with $N_u = 256$ units, a unidirectional encoder RNN, and a bidirectional decoder RNN.

Our second development stage served to optimise the window width w used for spectrogram extraction. We use the autoencoder configuration determined in the previous stage, and once again set $N_m = 320$. The window width w is evaluated between 0.04 and 0.36 seconds in steps of 0.04 seconds. For each value of w, the window overlap is chosen to be $0.5\,w$. As shown in Figure 3a, classification accuracy quickly rises above 84 % for $w > 0.10$ seconds, and peaks at 85.0 % for $w = 0.20$ seconds and $w = 0.28$ seconds. For larger values of w, classification accuracy decreases again. As a larger window width may blur some of the short-term dynamics of the audio signals, we choose $w = 0.20$ seconds. Correspondingly, the window overlap is chosen to be $0.5\,w = 0.10$ seconds.

In the final optimisation stage, we evaluated different numbers of Mel frequency bands $N_m \in \{40, 80, 160, 320, 640\}$, the results of which are shown in Figure 3b. Classification accuracy rises with larger values of N_m until it reaches 85.0 % for $N_m = 320$. Increasing N_m beyond 320 does not improve performance further, so we choose $N_m = 320$ to minimise the amount of data the system has to process.

3.4. Fusion Experiments

Given the supplied stereo audio tracks [4], we extract separate sets of spectrograms from the mean and difference of channels, and from the left and right channels individually (cf. Section 2.1). On each set of spectrograms, an autoencoder is trained, and the learnt

Table 1: Comparison of the classification accuracies of the different variants of our proposed system with the challenge baseline. We extract four different feature sets of spectrograms from the mean (M) and difference (D) of channels, and from the left (L) and right (R) channels separately. We obtain the highest accuracy after fusing the features generated from all channels.

System	Features	Accuracy [%] Devel.	Eval.
Baseline	200 (per frame)	74.8	61.0
Proposed: Individual Feature Sets			
Mean (M)	1 024	85.0	–
Left (L)	1 024	84.6	–
Right (R)	1 024	83.8	–
Difference (D)	1 024	82.0	–
Proposed: Fused Feature Sets			
Mean, Left	2 048	86.2	–
Mean, Left, Right	3 072	86.9	–
All (M + L + R + D)	4 096	**88.0**	67.5

representations are extracted as features for the instances. This results in four feature sets herein identified by the spectrogram type from which they have been extracted (i. e. ‘mean’, ‘difference’, ‘left’, and ‘right’). On the development set, the ‘mean’ feature set achieves the highest classification accuracy with 85.0 %, followed by ‘left’ with 84.6 %, ‘right’ with 83.8 %, and ‘difference’ with 82.0 % (cf. Table 1). In order to determine if the different spectral representations contain complementary information, we perform feature-level fusion. We perform a weighted fusion, in which the weights are proportional to performance of the individual systems. Fusing the ‘mean’ and ‘left’ feature sets improves classification accuracy to 86.2 % on the development set. Adding the ‘right’ feature set further increases classification accuracy to 86.9 %, and fusing all four feature sets results in 88.0 % accuracy (cf. Table 1). The latter constitutes our best result on the development set, with an improvement of 13.2 % over the baseline [4]. A confusion matrix for this result is given in Figure 4.

Besides fusion between different channels, we also investigated fusion between different window sizes w and numbers of Mel frequency bands N_m. We also trialled fusion with various conventional acoustic feature sets which we extracted from the raw audio samples using the openSMILE toolkit [24]. However, we did not identify a combination of these options which resulted in increased performance on the development set.

3.5. Challenge Submission and Evaluation Set Results

For our submission to the DCASE 2017 Acoustic Scene Classification Challenge, we select the four feature sets with the best performance on the development partition, i. e. the ‘mean’ feature set and all three fused feature sets. We extract spectrograms from audio samples in the evaluation set using the same parameters that we used for the development set. Subsequently, we extract the four individual feature sets described above with the respective autoencoder that we trained on the development set. Finally, fusion of these feature sets is performed as detailed above.

For prediction on the evaluation set, the MLP classifier is trained using the entire development set as training data. As shown in Table 1, our approach achieved classification accuracies of 00.0 %, 00.0 %, 00.0 % and 00.0 % on the evaluation set.

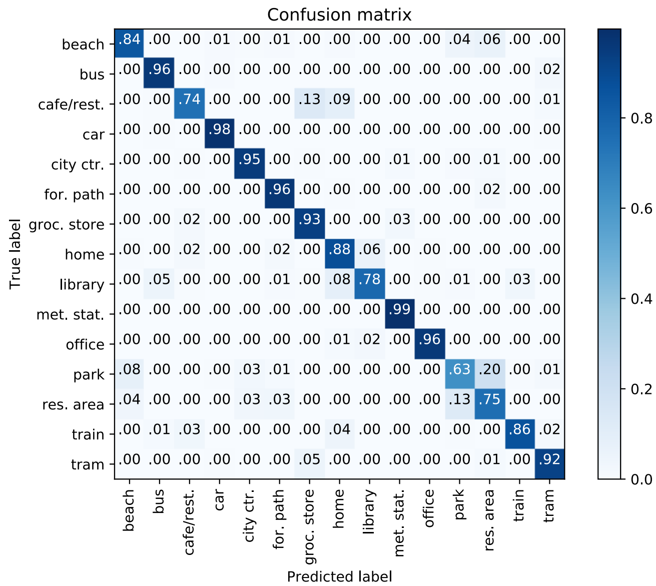


Figure 4: Confusion Matrix of our strongest performing system on the development partition of the TUT Acoustic Scenes 2017 data set which achieved a classification accuracy of 88.0 %.

4. CONCLUSIONS

Despite representation learning with deep neural networks (DNNs) has shown superior performance of hand-crafted feature sets in a variety of machine learning recognition and classification tasks, such approaches have not been widely explored with the domain of acoustic scene classification. In this regard, our entry to the 2017 DCASE 2017 Acoustic Scene Classification challenge has demonstrated the feasibility of using a recurrent sequence to sequence autoencoder for the unsupervised feature representation. A major advantage of our approach is that it is able to learn a fixed length representation from variable length audio signals while taking account of their time-dependent nature. A fused combination of features learnt from our system was able to achieve an accuracy of 88.0 % on the challenge development data, an improvement of 17.65 percentage points over the official baseline.

In future work, we will be testing our system over a wide range of different acoustic classification tasks. We also want to collect further data from social multimedia using our purpose built software [25] to train the autoencoder with more real life audio recordings. Finally we plan to investigate the potential of Generative Adversarial Networks for acoustic based deep representation learning.

5. ACKNOWLEDGEMENTS

This research has received funding from the European Unions’s Seventh Framework under grant agreement No. 338164 (ERC StG iHEARu) and the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 115902. This Joint Undertaking receives support from the European Union’s Horizon 2020 research and innovation programme and EFPIA.

6. REFERENCES

[1] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference (EUSIPCO 2016)*. Budapest, Hungary: IEEE, Aug 2016, pp. 1128–1132.

[2] E. Marchi, D. Tonelli, X. Xu, F. Ringeval, J. Deng, S. Squartini, and B. Schuller, "Pairwise Decomposition with Deep Neural Networks and Multiscale Kernel Subspace Learning for Acoustic Scene Classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 IEEE AASP Challenge Workshop (DCASE 2016), satellite to EUSIPCO 2016*. Budapest, Hungary: IEEE, Sep 2016, pp. 65–69.

[3] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. Plumbley, "Detection and Classification of Acoustic Scenes and Events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[4] 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)*, Nov 2017, submitted.

[5] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: A hybrid approach using binaural i-vectors and deep convolutional neural networks," *Detection and Classification of Acoustic Scenes and Events 2016 IEEE AASP Challenge (DCASE 2016)*, Sep 2016, Technical Report.

[6] G. Hinton, L. Deng, D. Yu, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[7] B. Schuller and A. Batliner, *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. Chichester, United Kingdom: Wiley.

[8] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[9] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[10] M. Schmitt and B. Schuller, "openXBOW — Introducing the Passau open-source crossmodal bag-of-words toolkit," *Journal of Machine Learning Research*, vol. 18, 2017, 5 pages.

[11] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1096–1104.

[12] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*. Edinburgh, Scotland: Omnipress, June 2012, pp. 1881–1888.

[13] K. Cho, B. van Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: ACL, Oct 2014, pp. 1724–1734.

[14] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112.

[15] E. Marchi, F. Vesperini, S. Squartini, and B. Schuller, "Deep Recurrent Neural Network-based Autoencoders for Acoustic Novelty Detection," *Computational Intelligence and Neuroscience*, vol. 2017, 2017, 14 pages.

[16] M. Zöhrer and F. Pernkopf, "Gated Recurrent Networks applied to Acoustic Scene Classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 IEEE AASP Challenge Workshop (DCASE 2016), satellite to EUSIPCO 2016*. Budapest, Hungary: IEEE, Sep 2016, pp. 115–119.

[17] F. Weninger, J. Bergmann, and B. Schuller, "Introducing CURRENNT: the Munich Open-Source CUDA RecurREnt Neural Network Toolkit," *Journal of Machine Learning Research*, vol. 16, pp. 547–551, 2015.

[18] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 3079–3087.

[19] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *arXiv preprint arXiv:1511.06114*, 2015, 10 pages.

[20] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, 2015, 12 pages.

[21] F. Weninger, S. Watanabe, Y. Tachioka, and B. Schuller, "Deep recurrent de-noising auto-encoder and blind dereverberation for reverberated speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, May 2014, pp. 4623–4627.

[22] Y.-A. Chung, C.-C. Wu, C.-H. Shen, and H.-Y. Lee, "Unsupervised learning of audio segment representations using sequence-to-sequence recurrent neural networks," in *INTERSPEECH*. ISCA, 2016, pp. 765–769.

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

[24] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM International Conference on Multimedia*. ACM, Nov 2013, pp. 835–838.

[25] S. Amiriparian, S. Pugachevskiy, N. Cummins, S. Hantke, J. Pohjalainen, G. Keren, and B. Schuller, "CAST a database: Rapid targeted large-scale big data acquisition via small-world modelling of social media platforms," in *Proc. ACII 2017*. San Antionio, TX: IEEE, October 2017, 6 pages.