INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Acoustic Scene Classification with Mismatched Devices Using CliqueNets and Mixup Data Augmentation

Truc Nguyen , Franz Pernkopf

Graz University of Technology,
Signal Processing and Speech Communication Lab.,
Inffeldgasse 16c, A-8010 Graz, Austria/Europe

t.k.nguyen@tugraz.at, pernkopf@tugraz.at

Abstract

Deep learning (DL) is key for the recent boost of acoustic scene classification (ASC) performance. Especially, convolutional neural networks (CNNs) are widely adopted with affirmed success. However, models are large and cumbersome, i.e. they have many layers, parallel branches or large ensemble of individual models. In this paper, we propose a resource-efficient model using CliqueNets for feature learning and a mixture-of-experts (MoEs) layer. CliqueNets are a recurrent feedback structure enabling feature refinement by the alternate propagation between constructed loop layers. In addition, we use mixup data augmentation to construct adversarial training examples. It is used for balancing the dataset of DCASE 2018 task 1B over the recordings of the mismatched devices A, B and C. This prevents over-fitting on the dataset of Device A, caused by the gap of data amount between the different recording devices. Experimental results show that the proposed model achieves 64.7% average classification accuracy for Device C and B, and 70.0% for Device A with less than one million of parameters.

Index Terms: Acoustic scene classification, mixture-of-experts layer, CliqueNets, mixup data augmentation

1. Introduction

The aim of acoustic scene classification (ASC) is to recognize the sounds of environments called acoustic scenes such as street, shopping areas or public transportation i.e. metro, bus or tram. In real environments, sound events composing an acoustic scene are varying and can have different degrees of overlap. Therefore, the acoustic scenes are unstructured and often unpredictable resulting in more challenges for ASC compared to speech and music signal processing. ASC has been embedded in many real-world applications such as mobile robot navigation systems, smart home monitoring systems or as a modern trend in smart portable devices in which there are limitations about resources and power consumption.

In conventional ASC systems, feature extraction and classification are two key stages contributing almost equally to its effectiveness. However, with the advent of DL, this gradually shifted to an approach composed of feature learning and classification e.g. CNNs. Log-mel energies have been the most popular features applied in ASC. Their success is attributed to a reasonably good representation of the spectral properties of the signal and a reasonably high inter-class variability allowing for class discrimination [1]. Furthermore, the features can be used as basis for higher level features such as harmonic percussive source separation (HPSS) [2] or feature learning of CNN models such as VGGs [2], [3], Xception [4], or DenseNets [5]. In addition, techniques such as transfer learning, mixup data augmentation, Generative Adversarial Networks (GANs), and attention techniques as well as ensemble techniques [6], [2], [3], [7] leverage the performance of ASC. However, some of these models have many parameters. Therefore, we are interested for a robust ASC model with good performance but with low number of parameters.

In this paper, we use a combination of CliqueNets as feature learning layers and a mixture-of-experts layer for classification of the DCASE 2018 data of task 1B [8]. For this task we have focused on mismatched recording devices. Log-mel energies are fed to the CliqueNets, which is a recently extended CNN architecture [9], where the feature refinement and multi-scale feature properties of the clique block allow the model to maximize the information flow among layers. Additionally, second-order information is included by using a convolutional layer with filter size 1x1 between the clique blocks. This enables the CliqueNets to learn relevant information. Furthermore, the multi-scale feature decomposition of the CliqueNets avoids the rapid increase of parameters in case the model becomes deeper. In order to enhance the robustness of the ASC model, we use a dense mixture-of-experts layer replacing the fully connected layer introduced in our previous work [10]. Many experts correspond to a variety of extracted features based on the outputs the CliqueNets feature learning layers. In addition, mixup data augmentation is used as a dataset balancing technique to enlarge the training set recorded by low quality devices i.e. Device B and Device C of the DCASE 2018 data of task 1B. In particular, the amount of data is equalized for each device. Mixup is also reused on the entire enlarged training dataset to leverage the ASC performance. Our contributions are summarized as follows:

1. We propose an efficient and robust model for ASC in terms of number of parameters and classification performance using CliqueNets combined with a mixture-of-experts layer.
2. We tackle the over-fitting problem for the DCASE 2018 task 1B caused by the training data imbalance between the recording devices using mixup data augmentation.

The rest of paper is organized as follows. Section 2 presents the related work using deep feature learning. Section 3 introduces the proposed ASC system, including audio processing, mixup data augmentation for the dataset balancing, CliqueNet and mixture-of-experts layer. In Section 4, we provide experiments and evaluate the performance of the proposed approach. Section 5 concludes the paper.

http://dx.doi.org/10.21437/Interspeech.2019-3002

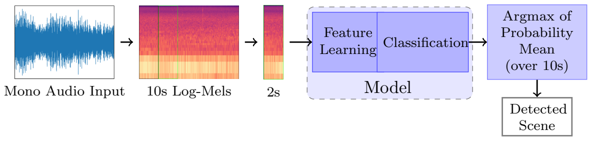

Figure 1: *Proposed System.*

2. Related work

Nowadays, the exploration of network architectures has been an interesting part of deep learning research. The increasing number of layers in modern networks amplifies the differences between architectures and motivates the exploration of different connectivity patterns and revisiting of old research ideas [11]. For widening the network, GoogLeNet introduces the Inception modules fusing the features of different feature maps to construct a multi-scale representation [12]. Wide residual networks [13] and FractalNet [14] are architectures which deepen and widen the networks to enhance performance of the models. However, deepening the network is a more attractive and efficient strategy. VGGs [15] have been a popular CNN architecture in this respect. ResNets with skip connection use widely adopted strategies to ease training and achieve a better performance by dropping a subset of layers in order to remove parameter redundancy. Recently, DenseNets [11] have been introduced which reuse the feature maps of previous layers. DenseNets concatenate feature maps instead of the identity mapping of residual block in order to reinforce learning of new features. Both ResNets and DenseNets use recurrent information propagation and achieve good performances. CliqueNets are considered as a different version of the recurrence structure. However, they use an iterative mechanism in each step of the propagation, extending the limitation of information flow between neighboring layers; all layers in a block participate in the recurrent loop so that the filters are communicated sufficiently and the blocks play both roles of information carrier and refiner [9].

For ASC, VGGs are widely used for feature learning because the models are large and deep enough to extract discriminative features and avoids the over-fitting problem[2],[3]. Although there are several successful researches incorporating recurrent connections into CNNs for ASC i.e. InceptionNets [4], or DenseNets [5], the models are built with a huge number of parameters. Therefore, we focus on using CliqueNets for the ASC task in order to benefit from maximizing the information flow and achieving an optimal feature refinement. Furthermore, we adjust the CliqueNet architecture to adapt to the properties of the ASC database.

3. Proposed architecture

The proposed system is illustrated in Fig.1. The system consists of three stages. First, the mono audio signals are converted to time-frequency representation and then split into 2s segments. These features are fed to a CliqueNet of 2 blocks for feature learning and a dense MoEs layer replacing the traditional fully connected dense layer for classification. Finally, the probability outputs of the CNN are averaged over 5 continuous 2s segments. Subsequently, the argmax operation is performed to obtain the final label.

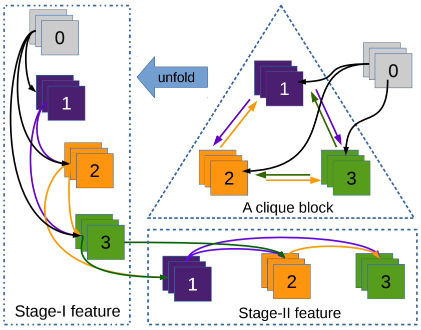

Figure 2: *An illustration of a clique block with 3 layers and unfolded propagation between the layers.*

3.1. Audio pre-processing

The DCASE 2018 data set of task 1B is recorded with different recording devices at different cities. The sampling rate is 44.1 kHz. The audio segments are 10 s in length. We extract 128 bin mel energies of the provided audio (mono) to obtain the spectral characteristics of the data. The window function of the short-time Fourier transform (STFT) is a Hann window and the window size is selected as 40ms with 20ms hop size. We split the log-mel energies into 2 s samples (128 bins x 100 frames per sample). All features are converted into logarithmic scale and normalized to zero mean and unit variance.

3.2. Mixup data augmentation

Mixup [16] is a simple and effective data augmentation method used in most of the best systems in the recent DCASE challenge [3], [17] and [10]. Mixup constructs virtual training examples by a convex combination of two randomly selected training data samples (x_i, y_i) and (x_j, y_j), i.e.

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)x_j, \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j, \end{aligned} \tag{1}$$

where x_i and x_j are 2s samples (log-mel spectrogram) and y_i and y_j are one-hot encoded class labels i.e. output vectors. $\lambda \in [0,1]$ is acquired by sampling from a beta distribution *Beta(*α*,* α*)* with α being a hyper parameter.

We perform mixup to generate adversarial samples for the low quality recording devices i.e. for Device B and Device C; we enlarge each training set by a factor of 5 (= enlargement factor). This reduces the gap in number of samples between the training dataset of Device A, B and C and enhances the generalization ability of the model. It helps to tackle the challenge, caused by the gap in data amount and quality of the different recording devices of DCASE 2019 task 1B. Empirically, we choose the same factor for Device B and Device C so that the number of training data equals Device A while an α of 0.2 is used for mixup on the balanced training set.

3.3. CliqueNets

CliqueNets are an extension of the classical CNN architecture with alternately updated cliques in order to improve the information flow in the networks. Usually, training is easier and parameters are utilized more efficiently [9]. Two layers inside a clique block are bidirectionally connected. This is different to a dense block in the DenseNet architecture [11] where each layer

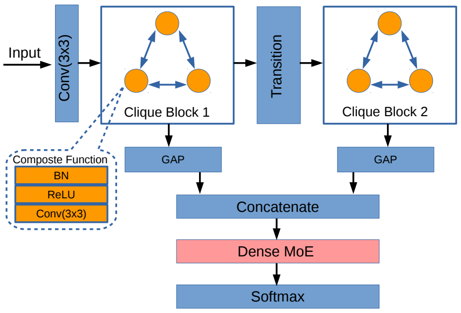

Figure 3: *A CNN model incorporating two CliqueNet blocks and a dense MoEs layer.*

has of all previous layers as inputs. The information flow among clique layers is maximized. To do so, the clique block performs two propagation stages. At Stage-I, all layers of a clique block are initialized, each updated layer is then concatenated to update the next layer by uni-directional connections. In stage-II, each layer receives feedback information from the later updated layers. This recurrent feedback architecture plays the role of feature refinement. Each layer is a composition of three consecutive operations: batch normalization (BN), followed by a rectified linear unit (ReLU) and a 3x3 convolution (Conv(3x3)). Figure 2 shows an illustration of a clique block of 3 layers and unfolding the propagation into two stages.

The proposed CliqueNet model starts with a Conv(3x3) layer using 64 filters and stride of 2, followed by 2 clique blocks updated with both propagation stages. The outputs of the first block are fed to a transition block including a convolution composition i.e. BN-ReLU- Conv(1x1) and a 2x2 max pooling layer for reducing the number of parameters to be transferred to the second block. The outputs of each block are concatenated with its inputs before the global average pooling (GAP) layer. The GAP layer allows to reduce the number of outputs of each block. These block outputs are concatenated before feeding the data to the dense MoEs layer. Finally, a softmax layer is used to process the MoEs outputs. Figure 3 shows the CNN model incorporating of two CliqueNet blocks and a dense MoEs layer.

Since DenseNets are combinations of clique blocks with only Stage-I processing, we test on several DenseNet architectures. Empirically, we observe that the very deep DenseNet structures cause over-fitting for the ASC task. Therefore, we build low-complexity CliqueNets using both propagation stages by shrinking the original CliqueNets [9] i.e. with small number of blocks and layers per block i.e. 2, 3 and 4. Beside that, the number of neurons per layer (growth rate) are kept the same in all clique blocks. We test with growth rates of 12, 24, 36 and 64.

3.4. Mixture of Experts Layers

MoEs are a long-existing modeling approach [18]. It consists of a set of modules referred to as expert networks which are suitable to model various characteristics of the input space. A gating network selects the suitable expert for each input. Recently, MoEs have been used as a part of a deep model that can be any specific layer in a network [19], [20].

Similar to our previous work [10], we use a dense MoEs layer [21] replacing a fully connected layer. It is similar to an attention mechanism and successful in exploiting the diversity of extracted features from different experts. The MoE layer usually improves the performance of the proposed model. The dense MoEs layer is defined as

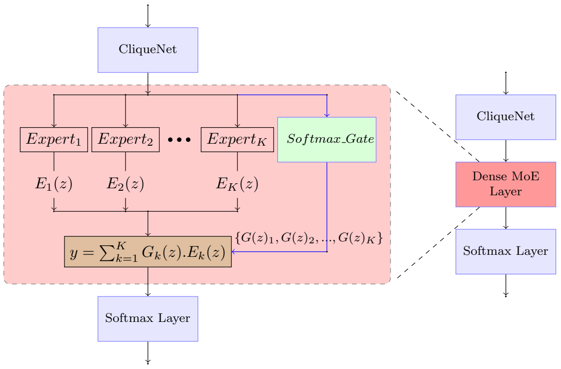

Figure 4: *A Mixture of Experts layer embedded within a CNN model.*

$$\mathbf{y} = \sum_{k=1}^{K} \underbrace{g(\mathbf{V}_k^T \cdot \mathbf{z} + \mathbf{b}_k)}_{G(\mathbf{z})_k} \cdot \underbrace{f(\mathbf{W}_k \cdot \mathbf{z} + \mathbf{c}_k)}_{E(\mathbf{z})_k}, \qquad (2)$$

where K denotes the number of experts, $g(\cdot)$ is the softmax gating, $f(\cdot)$ are the experts using ReLU activation with a linear operation for the dense MoEs layer. $\mathbf{V}_k$, $\mathbf{b}_k$ and $\mathbf{W}_k$, $\mathbf{c}_k$ denote the weights and the bias of the gating function and the expert k, respectively, while $\mathbf{z}$ is the input vector of the gating and expert function.

In this work, we use the dense MoEs layer as a fully connected layer between the CliqueNet model and the output (softmax) layer (see Figure 3). We test the model with 1 expert i.e. a normal fully connected layer and expert numbers of $K \in \{2, 5, 10, 15\}$. Figure 4 shows the structure of a MoEs layer embedded in our ASC system where the dense MoEs layer is illustrated as a combination of experts and their corresponding weights from a MoE softmax gate.

4. Experiments

4.1. Data

The audio dataset for the ASC task 1B [8] is the TUT Urban Acoustic Scene 2018 Mobile data recorded in six European cities. It consists of 10 scenes. The development set is comprised of the task 1A data set recorded by using the same binaural microphone at a sampling rate of 48kHz. The recordings are resampled and averaged into a single channel. A small amount of data is recorded by other devices. The original recordings were split into 10-second segments that are provided in individual files.

The training subset is composed of 6122 segments from device A, 540 segments from device B, and 540 segments from device C. The test subset contains 2518 segments from device A, 180 segments from device B, and 180 segments from device C. Since the evaluation data set has been provided without ground truth, we use a training subset and a test subset of the development set for training and evaluating the models, respectively.

4.2. Setup

The validation set for our model accounts for approximately 30% of the original training data and there are no segments from the same location and city in both training and validation data sets. The rest of the training dataset is used to perform the mixup data set balancing with enlarging factor of 5 for the data of both Device B and Device C. Training the network is carried out by optimizing the categorical cross-entropy using the Adam optimizer at a learning rate of 0.001. We use the Glorot uniform initializer for the network weights. The number of epochs and batch size was 350 and 128, respectively. Data is shuffled between the epochs. Model performance is evaluated on the validation set after each epoch and the selected model is the best performing one on the validation set.

4.3. Performance Testset

Table 1: *Accuracy (in %) and number of model parameters;* ***Den_x_k_t*** *denotes the DenseNet with* ***x*** *blocks, growth rate of* ***k*** *and a total layer number* ***t*** *of the entire model,* ***Cli_x_k_l*** *denotes the CliqueNet with* ***x*** *blocks, growth rate of* ***k*** *and a total layer number* ***l*** *of* ***x*** *clique blocks with both propagation stages.* ***_muBC*** *denotes mixup for dataset balancing of data from Device B and Device C.* ***_nE*** *denotes* ***n*** *number of experts of a MoEs layer, _0E is the case without using the MoEs layer.* ***_S1*** *denotes the CliqueNet with only Stage-I processing.*

Accuracy	Dev.A	Dev.B	Dev.C	Ave.BC	Parameters
Baseline [8]	58.9 (±0.8)	45.1 ±3.6	46.2 ±4.2	45.6 ±3.6	-
Best model of DCASE2018 task 1B [7]	68.4	63.3	63.9	63.6	12M
ICME 2019 model [10]	**68.7**	**65.6**	**66.7**	**66.1**	**2,166,633**
Den_3_12_40_muBC	64.2	52.2	48.9	50.6	3,422,058
Den_4_12_121_muBC	65.0	51.7	51.7	51.7	7,041,482
Cli_2_36_6_S1_0E_muBC	65.1	56.7	56.7	56.7	122,874
Cli_2_36_6_0E_muBC	68.9	58.3	62.2	60.3	267,154
Cli_2_36_6_2E_muBC	66.8	55.6	56.7	56.1	346,396
Cli_2_36_6_5E_muBC	67.6	58.9	55.6	57.2	583,279
Cli_2_36_6_10E_muBC	**70.0**	**66.7**	**62.8**	**64.7**	**978,084**
Cli_2_36_6_15E_muBC	66.1	56.1	57.8	56.9	1,372,889
Cli_2_36_6_0E	66.8	58.3	57.8	58.1	267,154
Cli_2_36_6_10E	69.5	60.0	57.8	58.9	978,084

We can see from Table 1 that the accuracy on Device A data of all models is always higher than that of Device B and C because of the imbalance of the data amount of the recording devices. However, with balancing the training set, we can see that the accuracy of Device A is on par, but there are significant performance improvements for Device B and Device C. The best CliqueNet structure (Cli_2_36_6_10E_muBC) improves accuracy by 6.7% and 5.0% (absolute) for Device B and Device C, respectively compared to the same structure without balancing the training set. The improvement is similar for the other structures. Furthermore, the optimal number of experts of the MoEs layer is 10 experts for Cli_2_36_6. This structure significantly outperforms the structures of different number of experts.

The deep DenseNet structures do not outperform the CliqueNets in terms of both the accuracy and the number of parameters. The CliqueNet with only Stage-I processing (Cli_2_36_6_S1_0E_muBC), which is similar to the DenseNet structure of 2 blocks, outperforms the deeper DenseNets. However, it is not able to perform better than the CliqueNets with both propagation stages.

Table 2: *Class-wise average accuracy of Device B and C of the Cli_2_36_6_10E_muBC system on the test set compared to the baseline system and the best model of DCASE 2018 task 1B [7].*

Scene labels	Baseline [8]	Best model DCASE 2018 task 1B [7]	ICME 2019 [10]	Proposed
Airport	72.5	58.3	47.2	69.4
Bus	78.3	80.6	77.8	80.6
Metro	20.6	41.7	30.6	33.3
Metro station	32.8	61.1	75.0	63.9
Park	59.2	91.7	91.7	97.2
Public square	24.7	55.6	47.2	38.9
Shopping mall	61.1	75.0	83.3	55.6
Street_pedestrian	20.8	50.0	63.9	50.0
Street_traffic	66.4	83.3	83.3	86.1
Tram	19.7	38.9	61.1	72.2
Average	45.6 ± 3.6	63.6	**66.1**	**64.7**

The best CliqueNet model outperforms the baseline system and the best model of DCASE 2018 task 1B in terms of both accuracy and the model complexity. Although it can not outperform our previous model based on the modified VGGs structure of 4 double convolutional blocks [10] in terms of average accuracy on Device B and C (64.7% versus 66.1%), it is stronger in terms of performance for Device A and Device B. Furthermore, the complexity of the model is significantly reduced i.e. around 1M (million) versus 2M parameters. The multi-scale feature processing and the shrinking of the number of parameters of the proposed CliqueNet structure is responsible for the differences.

Finally, Table 2 shows the class-wise accuracy of the models. By comparing performance of the individual scenes, we can see that the proposed model outperforms the best model of the DCASE 2018 challenge for almost all scenes. Furthermore, it is better on 6 scenes compared to [10]. The simplest scene of our proposed model is always park reaching a accuracy at 97.2%. While metro is the most difficult scene for our model, but the performance for this scene is still a bit better than that of [10].

5. Conclusion

We present a novel and efficient ASC system based on feature learning of the CliqueNet structure which consists of 2 clique blocks with a growth rate of 36 and 3 composition layers per block, and a MoEs layer using 10 experts. In addition, we leverage the performance by using mixup data augmentation for balancing the training set of Device B and Device C as well as for data augmentation of the balanced training set. Balancing the training set and the mixture-of-experts layer play an important role in dealing with the problem of mismatched devices of the DCASE 2018 task 1B. Finally, the proposed model performs on par with a MoEs model while the number of model parameters have been reduced by a factor of 2. Hence, our model has computational benefits while requiring less memory for the parameters.

6. Acknowledgements

This research was supported by the Vietnamese - Austrian Government scholarship and by the Austrian Science Fund (FWF) under the project number I2706-N31. We acknowledge NVIDIA for providing GPU computing resources.

7. References

[1] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 2, pp. 379–393, 2018.

[2] Y. Han and J. Park, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," in *Proceedings of DCASE 2017 Workshop*, November 2017, pp. 46–50.

[3] Y. Sakashita and M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," DCASE2018 Challenge, Tech. Rep., September 2018.

[4] Y. Liping, C. Xinxing, and T. Lianjie, "Acoustic scene classification using multi-scale features," in *Proceedings of DCASE 2018 Workshop*, November 2018, pp. 29–33.

[5] D. Wang, L. Zhang, K. Xu, and Y. Wang, "Acoustic scene classification based on dense convolutional networks incorporating multi-channel features," in *Journal of Physics: Conference Series*, vol. 1169, no. 1. IOP Publishing, 2019, p. 012037.

[6] S. Mun, S. Park, D. K. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane," in *Proceedings of DCASE 2017 Workshop*, November 2017, pp. 93–102.

[7] T. Nguyen and F. Pernkopf, "Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters," in *Proceedings of DCASE 2018 Workshop*, November 2018, pp. 34–38.

[8] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of DCASE 2018 Workshop*, November 2018, pp. 9–13.

[9] Y. Yang, Z. Zhong, T. Shen, and Z. Lin, "Convolutional neural networks with alternately updated clique," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2413–2422.

[10] T. Nguyen and F. Pernkopf, "Acoustic scene classification using deep mixture of parallel convolutional neural networks," in *Proceedings of the ICME*, 2019, accepted.

[11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[13] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[14] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648*, 2016.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proceedings of the ICLR*, 2018.

[17] M. Dorfer, B. Lehner, H. Eghbal-zadeh, H. Christop, P. Fabian, and W. Gerhard, "Acoustic scene classification with fully convolutional neural networks and I-vectors," DCASE2018 Challenge, Tech. Rep., September 2018.

[18] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[19] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *Proceedings of the ICLR*, 2017.

[20] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank rnn language model," in *Proceedings of the ICLR*, 2018.

[21] O. Emin. (2018) The softmax bottleneck is a special case of a more general phenomenon. [Online]. Available: https://severelytheoretical.wordpress.com/2018/06/08/the-softmax-bottleneck-is-a-special-case-of-a-more-general-phenomenon/