SVD-BASED CHANNEL PRUNING FOR CONVOLUTIONAL NEURAL NETWORK IN ACOUSTIC SCENE CLASSIFICATION MODEL

Jun Wang[1], *Shengchen Li*[1], *Wenwu Wang*[2]

[1] Beijing University of Posts and Telecommunications, Beijing, P. R. China
[2] Centre for Vision, Speech and Signal Processing, University of Surrey, UK
{wangjun19930314, shengchen.li}@bupt.edu.cn, {w.wang}@surrey.ac.uk

ABSTRACT

Convolutional Neural Network (CNN) offers promising performance in Acoustic Scene Classification (ASC) tasks. The CNN model, however, often involves a large number of parameters, and thus requires large storage space for the implementation of the model. In this paper, we propose a new method for model pruning based on singular value decomposition (SVD). More specifically, the number of parameters is reduced by a low-rank decomposition method, where a matrix is decomposed into products of three small matrices. As a result, the original convolutional layer is decomposed into three smaller convolutional layers resulting in an overall reduction in the number of parameters involved in the model. The proposed method is evaluated on the dataset of ASC task in D-CASE2018. The results illustrate that the proposed approach dramatically reduces the CNN layers size by more than 90 % with relatively 1 % performance loss and the activity of parameters in convolutional layers increases with performance loss of the compressed model.

Index Terms— Convolutional Neural Network, singular value decomposition, compression

1. INTRODUCTION

The goal of acoustic scene classification (ASC) is to assign a test recording into one of the predefined classes that characterizes the environment in which it was recorded. The problem arises in a wide range of applications such as audio surveillance and forensics. Because of the capability of learning hierarchical features from low-level audio features, neural networks based deep learning methods have shown great potential and significant improvement in ASC [1]. Neural networks require a huge number of parameters to produce state-of-the-art results, although many parameters seem to be redundant.

With a more complex model, CNN achieves little performance gain with expensive cost of computational power. This problem becomes serious when resources are limited. Studies have shown that most of the weight parameters in deep neural networks are of very small values, and the smaller weights have negligible impact on the output of the network layer [2]. As a result, the deep learning algorithm can be compressed.

Recently, there are works focusing on CNN compression. For example, quantization [3] and structured simplification [4] are commonly used to compress CNN. Quantization [5] reduces the computational complexity related to floating point operation. Sparse connections of network [6] eliminate the connection between neurons which is an effective way to reduce the model size. While the speed-up of quantization should be implemented on the specific multiplier, the implementation of sparseness becomes difficult on the hardware architecture, because of the usage of computational and storage resources.

Compared to quantization and sparseness, low rank decomposition does not depend on a specific multiplier nor introduces an extra memory index. Kim et al [7] proposed low rank factorization method to decompose a convolutional layer into several efficient parts and SVD pruning is a method for low rank matrix decomposition. The use of SVD-based pruning in this paper is inspired by work of Denton [8] and Xue et al [9] who proposed truncated SVD to accelerate Multilayer Perceptron (MLP). The idea of channel pruning is inspired by He et al [10] who proposed to factorize a convolutional layer into combination of 3×3 and 1×1 convolutional block to remove redundant channels on feature maps. In addition, some training-based approaches are proposed to simplify networks. As a training-based approach, Wen et al [11] proposed channel-based structured sparsity learning is restructure convolutional layers of LeNet and AlexNet with high compression ratio. Training-based pruning methods are more time-consuming, and the effectiveness of pruning very deep networks on large datasets is not guaranteed [12]. So this paper compresses model firstly and then fine tunes the compressed model to restore the original performance.

In this paper, we study the problem of SVD-based channel pruning for convolutional neural network to reduce the redundant parameters in deep CNNs with little performance loss in the scenarios of acoustic scene classification.

Previous works [8, 9, 10] set the number of singular values artificially, and they can not adjust the number of singular

978-1-5386-9214-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMEW.2019.00073

values according to different layers automatically. In this paper, we find the optimal number of singular values for each convolution layer under the given reconstruction error.

Although there are many effective ways to compress the CNN model, there is no literature analysis on the parameters activity which represents the ratio of weights importance. In this paper, an experiment is presented to analyze why SVD-based channel pruning can effectively compress CNN in terms of parameters activity. On this basis, this paper shows that the SVD pruning aims to reduce performance loss by improving the parameter activity. In SVD-based pruning, a convolutional channel is factorized into several small effective convolutional layers with low-rank approximation and fewer parameters than the original channel. After compression, the compressed layers are fine-tuned until the prediction performance is restored with the standard back-propagation method. The restructured layer has similar input and output as the original layer, with a couple of extra smaller layers. After SVD restructuring, the total parameter size of the convolutional layer is reduced with certain loss in accuracy depending on the extent to which the model is compressed. Then the restructured model can be fine-tuned with the standard back-propagation method to recover the performance.

Main contributions in this paper are: (1) a SVD-based channel pruning methods is presented to reduce the redundant parameters in deep CNNs; (2) the proposed method is applied to perform ASC. The results of our experiments show that when more than 10 % of the original structure parameters in the convolutional layer are retained, the performance drop of the compressed model is only about 1 %.

2. PROPOSED PRUNING METHOD

In this section, we introduce the baseline architecture which will be compressed and then propose a SVD-based channel pruning algorithm for convolutional layers.

Since we only prune the convolutional neural network, the number of channels is represented as the number of feature maps in the convolutional neural network in this paper. Channel-based pruning reduces the number of feature maps in the convolutional layer by SVD to reduce the amount of parameters in the convolutional neural network. A convolution layer has an input channel and an output channel. Therefore, the SVD-based pruning method proposed in this paper uses a matrix decomposition method to compress the input channel and output channel of the original convolution layer respectively.

Firstly, the weight of CNN is transformed into two weight matrices according to input channel and output channel, and then the two weight matrices are pruned by the SVD method respectively. The binary search algorithm is used to find the optimal number of singular values with a pre-defined reconstruction error, and then the two weight matrices after being pruned are used to reconstruct new convolutional layers. Finally, an original convolutional layer is decomposed into three small convolutional layers, and the number of channels of new convolutional layers is smaller than the number of channels of original convolutional layer.

For the SVD decomposition, we assume that the original matrix $W \in R^{m \times n}$, k singular values are retained, so W can be decomposed as formula (1).

$$W_{m\times n} \approx \widehat{U}_{m\times k}\widehat{\Sigma}_{k\times k}\widehat{V}_{k\times n} \tag{1}$$

where $\widehat{U}_{m\times k}$ is represented as "left" matrix, $\widehat{\Sigma}_{k\times k}$ is called the "central" matrix and $\widehat{V}_{k\times n}$ is represented as "right" matrix.

2.1. Baseline Architecture

In this paper, the Convolution Recurrent Neural Networks (CRNNs) baseline system [13] for ASC will be introduced and only CNN structure will be compressed in the CRNN baseline system. In other words, networks other than the CNN structure will not be processed. Figure 1 shows the baseline system used in this paper, , which was presented in [13].

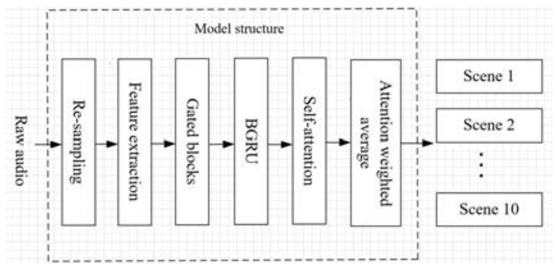

Fig. 1. The diagram of the baseline model

In the CNN structure, a "convolution block" is composed of two consecutive convolution layers and one maximum pooling layer with a small receptive field (3×3), followed by a max-pooling layer. A learnable gated activation function using GLU (Gated Linear Unit) [14] in the convolutional layer is used to replace the traditional nonlinear activation function. Recurrent neural network are following the convolutional blocks. After recurrent neural networks, a self-attention mechanism [15] is used to process all features in the output of previous layer. Finally, the weighted pooling is used to get the score for each category.

In the CRNN network structure of [13], there is a large number of convolutional layers, and most of the operations in the network structure are convolutional operations. Compressing convolutional layers can improve the efficiency of the network. To address this problem, we only prune the channel of CNN layers of the CRNN baseline system.

2.2. SVD-based pruning in channel

Here, a SVD-based model pruning method for the CNN channels is presented.

2.2.1. *Pruning*

The first line in Figure 2 represents the calculation of the original convolutional layer, and the second row represents the calculation method of the convolutional layer after decomposition. In Figure 2, $X \in R^{W_1 \times H_1 \times i}$ represents the input feature map in the convolutional layer, where i represents the number of input feature maps, W_1 and H_1 represent the width and height of the input feature map respectively; $Y \in R^{W_2 \times H_2 \times o}$, where o represents the number of output feature maps, and W_2 and H_2 represent the width and height of the output feature map, respectively.

In this paper, SVD pruning is applied to the channel (the number of feature maps) of the convolutional layers. However, the convolutional layer contains input channels and output channels, so the input channel and output channel need to be trimmed separately. The weight of a convolutional layer is seen as a four-dimensional tensor, and we assume W_{whio} as the weight of convolutional layers with the dimension of $R^{w \times h \times i \times o}$, where w and h are the size of kernels, i is the number of input channels and o is the number of output channels. For the pruning of input channel, W_{whio} is reshaped into W_{iN} with the dimension of $R^{i \times N}$ (where $N = w \times h \times o$, "×" represents multiplication) along the i axis. Then W_i is applied to the SVD pruning, and we assume that n singular values are preserved. To be consistent with the input channel, the "left" matrix $\widehat{U}_{in}$ should be preserved thus $\widehat{U}_{in}$ has a dimension of $R^{i \times n}$ in the SVD pruning. Similarly, for the pruning of the output channel, W is reshaped into W_{Mo} with the dimension of $R^{M \times o}$ (where $M = w \times h \times i$) along the o axis. Then W_o is applied to the SVD pruning and m singular values are preserved. To be consistent with the output channel, the "right" matrix $\widehat{V}_{mo}$ should be preserved with the dimension of $R^{m \times o}$.

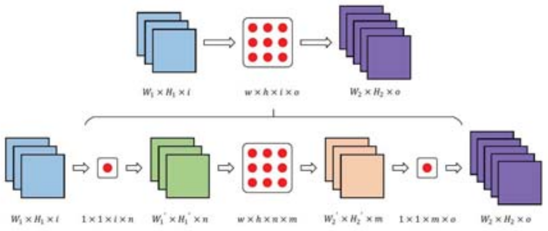
$W_1 \times H_1 \times i$
$w \times h \times i \times o$
$W_2 \times H_2 \times o$
$W_1 \times H_1 \times i$
$1 \times 1 \times i \times n$
$W_1' \times H_1' \times n$
$w \times h \times n \times m$
$W_2' \times H_2' \times m$
$1 \times 1 \times m \times o$
$W_2 \times H_2 \times o$

Fig. 2. Weight matrix decompositiong of a convolutional layer

Through the matrices W_{iN} and W_{Mo}, we transform the original matrix W_{whio} into "central" matrix C_{whnm} with the dimension $R^{w \times h \times n \times m}$, we formulate the process as formula (2).

$$C_{whnm} = W_{whio} \times \widehat{U}_{in} \times \widehat{V}_{mo} \tag{2}$$

After the SVD decomposition, the calculation method of the new convolutional layer is shown in the second line in Figure 2, the original convolution kernel is divided into three convolution kernels. The three convolution kernels are $C_{whnm} \in R^{w \times h \times n \times m}$, $C_L \in R^{1 \times 1 \times i \times n}$ which is from $\widehat{U}_{in}$ and $C_R \in R^{1 \times 1 \times n \times m}$ which is from $\widehat{V}_{mo}$. The input feature map X is subjected to convolution operation of the convolution kernel C_L to obtain a new feature map $X_1 \in R^{w_1 \times H_1 \times n}$, X_1 is subjected to a convolution operation of the convolution kernel C_{whnm} to obtain a feature map $Y_1 \in R^{W_2 \times H_2 \times m}$, and Y_1 is subjected to a convolution kernel C_R convolution operation to obtain a final output characteristic map $Y \in R^{W_2 \times H_2 \times o}$.

2.2.2. *Reconstruction*

After the SVD decomposition, one convolutional layer is decomposed into three small convolutional layers.

As shown in Figure 2, the first small convolutional layer has i input channels and n output channels with the kernel size of 1×1; the second small convolutional layer has n input channels and m output channels with kernel size of $h \times w$; the third small convolution layer has m input channels and o output channels, and the kernel size of 1×1.

After SVD-based channel pruning, the original matrix W is reconstructed by three smaller matrices as $\widehat{W}$ which has the same dimension as W, as follows.

$$\widehat{W}_{whio} = C_{whnm} \times \widehat{U}_{in} \times \widehat{V}_{mo} \tag{3}$$

In order to determine the optimal number of singular values, we define the reconstruction error, represented as ERR, in terms of the relative error range of the original convolutional layers and the convolutional layers after reconstruction. When n and m singular values of the input and output channels are retained, the reconstruction error is formulated as

$$ERR = F(n, m) = \frac{\|\widehat{W} - W\|}{\|W\|} \tag{4}$$

where $\|\cdot\|$ denotes the Frobenius norm of a matrix. Within the tolerance of the error, a binary search algorithm is used to find the smallest n and m.

Before reconstruction, the number of parameters in a convolution layer is presented as ϕ:

$$\phi = h \times w \times i \times o \tag{5}$$

The number of parameters in three small convolution layers is presented as ϕ' after reconstruction:

$$\phi' = i \times n + h \times w \times n \times m + m \times o \tag{6}$$

when ϕ is less than ϕ', it shows that the SVD method is effective and achieves the goal of reducing parameters.

Table 1. Binary search algorithm. In the algorithm description, $F(\cdot)$ returnes the relative error when giving the number of channels according to formula 4.

```
Algorithm : Binary search
Input
$N$: the number of input channels;
$M$: the number of output channels; $ERR$ : relative error
Output
$n$: reserved the number of channels;
$m$: reserved the number of channels
Initialize: $min\_in \leftarrow 1$, $max\_in \leftarrow N$
$min\_out \leftarrow 1$, $max\_out \leftarrow M$
Functions
1 repeat
2 if $F(N-1, M-1) > ERR$: return $None$
3 while($min\_in <= max\_in$ or $min\_out <= max\_out$):
4 $mid\_in = \frac{max\_in + min\_in}{2}$
5 $mid\_out = \frac{max\_out + min\_out}{2}$
6 if F $(mid\_in, mid\_out) < ERR$ and
$F(mid\_in, mid\_out) > ERR$ :
7 return $mid\_in$, $mid\_out$
8 elseif $F(mid\_in, mid\_out) > ERR$ :
9 $min_in = mid_in+1$
10 $min\_out = mid\_out + 1$
11 else :
12 $max\_in = mid\_in - 1$ ,$max\_out = mid\_out - 1$
13 End
```

2.2.3. *Binary search*

After the SVD decomposition, the number of singular values needs to be set appropriately, in previous work, such as [9], this is achieved artificially rather than automatically according to different convolution layers. In our method, the optimal number of singular values for each convolution layer is found in terms of the allowed reconstruction error. A brutal force method could be used to find the optimal number, which is nevertheless computationally expensive. Instead, we use a binary search method to find the optimal number of singular values.

According to section 2.2.1 and section 2.2.2, the numbers of input channel and output channel of the original convolutional layer are i and o. After the SVD decomposition, the input channel retains n singular values, and the output channel retains m singular values. Therefore, the role of the binary search algorithm is to find the optimal n and m.

The iterative process based on the binary search algorithm shown in Table 1. The inputs of this algorithm are i and o which are the number of channels in the original convolutional layer, and the given reconstruction error ERR. The outputs of the algorithm are n and m which are the optimal numbers of the retained singular values .

2.2.4. *Fine-tuning*

Fine tuning is to retrain the model based on the already compressed model and the initial parameters of the model are the parameters retained after compression. Compared with the original model, the compressed model adds several convolution layers. The parameters can be updated using the conventional backpropagation algorithm. Thus, we can fine-tune the compressed model to restore original performance.

3. EXPERIMENT AND RESULTS

3.1. Experiment setup and performance

In this section, we evaluate the proposed method on the subtask 1 of DCASE2018 ASC task. The original recordings were split into segments with 10 seconds that are provided in individual files. The dataset of this task consists of 10 kinds of acoustic scenes which are, all having distinct recording locations.

Mel-frequency Cepstral Coefficients (MFCCs) have been used inclusively in acoustic sound classification [16]. In recent works of sound event detection [16], MFCCs are shown to be sensitive to background noise hence they are not the best choice. In speech recognition, Mel-filter bank (MBK) features have already been demonstrated to be better than MFCCs in the deep neural network [17]. In this paper, we take log-Mel filter banks as features.

For this task, each 10-second chunk has 320 frames by 128 mel-frequency channels. As shown in Fig. 1, four gated convolutional neural network blocks with 128 feature maps and a common convolutional neural with 256 feature maps are adopted. Each convolutional network has 3×3 receptive fields. After the convolutional layer, 3 dimensions matrix was flattened to 2 dimensions matrix data into the RNN module. The Adam optimizer [18] was used for gradient optimization to improve learning convergence. Mixup is used as a method of data augmentation [19] in this paper.

3.2. Results

The scoring of acoustic scene classification will be based on classification accuracy: the number of correctly classified segments among the total number of segments. The average of the class-wise accuracy represented as “ACC” is calculated as accuracy because each segment is considered an independent test sample. In addition, in order to study the efficiency of SVD decomposition, we compared the number of parameters of the original model to the model after pruning, and then “RPA” represents the ratio of reserved parameters to original parameters in the whole convolution layers. In order to compare the performance of the original model with the model after pruning, we task "RAC" as the ratio of fine-turning accuracy to original accuracy.

In order to analyze the reason that SVD-based channel pruning can effectively compress CNN, this paper studies the weight activity of the convolutional layer, including the original model, and the compressed model. If the absolute value of a weight is larger than a threshold σ, the weight is said to be active or important. The ratio of weights activity A is defined as ratio between the number of active weights and the total number of parameters. WA indicates the ratio between the sum of the absolute value in weight parameters greater than the defined threshold to the sum of the total parameters after compression. Referring to previous work [20], the activity ratio A of weights (ω) in convolution layers is defined as follows:

$$A = \frac{count(|\omega| > \sigma)}{N} \tag{7}$$

WA is defined as follows:

$$WA = \frac{sum(|\omega| > \sigma)}{sum(\omega)} \tag{8}$$

The SVD pruning is applied on the whole convolutional layers of the acoustic model. The ERR is set as 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. Table 2 summarizes the experimental results, including the comparison of the performance between the compressed model and the original model, the comparison of the parameters of the convolutional layers, and the comparison of the weight activity.

Table 2. Results of SVD restructuring on the ASC task. "ERR" represents the reconstruction error.

$Model$	ACC	RPA	RAC
$Original$	70.81%	100%	100%
$SVD(ERR = 0.2)$	70.18%	81.46%	99.10%
$SVD(ERR = 0.3)$	70.17%	90.13%	99.09%
$SVD(ERR = 0.4)$	70.15%	57.42%	99.06%
$SVD(ERR = 0.5)$	70.04%	36.34%	99.48%
$SVD(ERR = 0.6)$	70.12%	20.73%	99.02%
$SVD(ERR = 0.7)$	70.47%	10.76%	99.84%
$SVD(ERR = 0.8)$	68.02%	5.58%	96.06%
$SVD(ERR = 0.9)$	66.69%	2.28%	94.18%

From Table 2 we can see that when we apply SVD pruning on CNN layers and ERR is set less than 0.7, only $1/10$ parameters of the CNN model are retained and the performance decreases less than 1 % relatively. When ERR is set larger than 0.7, less than $1/10$ parameters of the original model are retained and the performance decreases about 4 % relatively. If we compress the model more aggressively, accuracy will drop further after the SVD decomposition.

Table 3. Optimized results of the circular arch for different slenderness and opening angles

Model	$\sigma = 0.01$		$\sigma = 0.05$	
	A	WA	A	WA
$Original$	88.54%	98.97%	47.26%	77.48%
$SVD(ERR = 0.2)$	90.61%	99.40%	56.42%	86.71%
$SVD(ERR = 0.3)$	91.10%	99.45%	58.25%	87.65%
$SVD(ERR = 0.4)$	92.50%	99.60%	64.00%	90.77%
$SVD(ERR = 0.5)$	93.39%	99.69%	67.77%	92.56%
$SVD(ERR = 0.6)$	94.39%	99.77%	72.46%	94.54%
$SVD(ERR = 0.7)$	94.38%	99.79%	72.67%	94.90%
$SVD(ERR = 0.6)$	94.62%	99.81%	73.72%	95.49%
$SVD(ERR = 0.7)$	95.11%	99.84%	75.94%	96.26%

Table 3 shows the activity ratio of weights A and WA with different reconstruction errors when the thresholds are set as 0.01 and 0.05 respectively. When the reconstruction error increases, the activity ratio of weights in CNN layers shows an increasing trend, but the performance of the model gradually decreases. When the activity is large, there are many important parameters in the weight, so the pruning is able to maintain performance. As can be seen from Table 2, the SVD-based channel pruning method reduces model size and maintains performance by increasing the weight activity of the convolution layer.

4. CONCLUSION

In this paper, we have presented a channel pruning method based on SVD. The accuracy can be maintained compared to the original model when we reduce a modest number of parameters. If we reduced a large number of parameters heavily, the compressed model can be fine-tuned to restore the accuracy with little performance loss. From the results presented, most of the parameters are inactive despite the presence of a lot of parameters in the convolutional layers of the original model. This method can effectively factorize a convolutional layer into several small convolutional layers and reduce the redundant parameters in the channels to increase parameters activity. In addition, we found that as the number of layers in the convolutional layer increases, the activity of the weight is lower, so the strength of the pruning becomes smaller.

5. ACKNOWLEDGEMENT

Dr Wenwu Wang was supported by EPSRC grant EP/N014111/1 "Making Sense of Sounds".

6. REFERENCES

[1] Eduardo Fonseca, Rong Gong, Dmitry Bogdanov, Olga Slizovskaia, Emilia Gomez, and Xavier Serra, "Acoustic scene classification by ensembling gradient boosting machine and convolutional neural networks," *Detection*

and Classification of Acoustic Scenes and Events 2017, Nov. 2017.

[2] Dong Yu, Frank Seide, Gang Li, and Li Deng, "Exploiting sparseness in deep neural networks for large vocabulary speech recognition," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4409–4412, Aug. 2012.

[3] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi, "XNOR-net: Imagenet classification using binary convolutional neural networks," *2016 IEEE European Conference on Computer Vision (ECCV2016)*, pp. 1–17, Oct. 2016.

[4] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman, "Speeding up convolutional neural networks with low rank expansions," 2014.

[5] Rainer Lienhart and Malcolm Slaney, "PLSA on large scale image databases," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2007, IEEE.

[6] Song Han, Huizi Mao, and William J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *ICLR*, vol. 56, no. 4, pp. 1–14, May 2016.

[7] Yong-Deok Kim, Taelim Choi, Lu Yang, Dongjun Shin, Eunhyeok Park, and Sungjoo Yoo, "Compression of deep convolutional neural networks for fast and low power mobile applications," *Computer Science*, vol. 71, no. 2, pp. 576–584, 2015.

[8] Emily Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus Dept, "Exploiting linear structure within convolutional networks for efficient evaluation," *Computer Vision and Pattern Recognition(CVPR2014)*, , no. 1, pp. 1–9, Apr. 2014.

[9] Jian Xue, Jinyu Li, and Yifan Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," *INTERSPEECH2013*, pp. 2365–2369, Aug. 2013.

[10] Yihui He, Xiangyu Zhang, and Jian Sun, "Channel pruning for accelerating very deep neural networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, vol. 1, pp. 1398–1406, Oct. 2017.

[11] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li, "Learning structured sparsity in deep neural networks," *IEEE 2016 Conference and Workshop on Neural Information Processing Systems (NIPS2016)*, Aug. 2016.

[12] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung, "Structured pruning of deep convolutional neural networks," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 13, no. 3, pp. 1–18, feb 2017.

[13] Yong Xu, Qiuqiang Kong, Wenwu Wang, and Mark D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2018)*, pp. 121–125, Oct. 2018.

[14] Rui Cai, Lie Lu, Alan Hanjalic, Hong-Jiang Zhang, and Lian-Hong Cai, "A flexible framework for key audio effects detection and auditory context inference," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 3, May 2006.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is All you need," *31st Conference on Neural Information Processing Systems (NIPS 2017)*, June 2017.

[16] Courtenay V. Cotton and Daniel P. W. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 69–72, Oct. 2011.

[17] Michael L. Seltzer, Dong Yu, and YongqiangWang, "An investigation of deep neural networks for noise robust speech recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7398–7402, Oct. 2013.

[18] Asmelash Teka Hadgu, Aastha Nigam, and Ernesto Diaz-Aviles, "Large-scale learning with adagrad on spark," *Proceedings of 2015 IEEE International Conference on Big Data (Big Data*, vol. 2, pp. 2828–2830, Dec. 2015.

[19] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," *2018 International Conference on Learning Representations (ICLR 2018)*, pp. 1–13, 2018.

[20] Jun Wang and Shengchen Li, "Comparing the influence of depth and width of deep neural network based on fixed number of parameters for audio event detection," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2681–2685, Apr. 2018.