COMPACT CONVOLUTIONAL RECURRENT NEURAL NETWORKS VIA BINARIZATION FOR SPEECH EMOTION RECOGNITION

Huan Zhao[1⋆] *Yufeng Xiao*[1] *Jing Han*[2] *Zixing Zhang*[1,3]

[1]College of Computer Science and Electronic Engineering, Hunan University, China
[2] Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, Germany
[3]Group on Language, Audio & Music, Imperial College London, UK

ABSTRACT

Despite the great advances, most of the recently developed automatic speech recognition systems focus on working in a server-client manner, and thus often require a high computational cost, such as the storage size and memory accesses. This, however, does not satisfy the increasing demand for a succinct model that can run smoothly in embedded devices like smartphones. To this end, in this paper we propose a neural network compression method, in the way of quantizing the weights of the neural networks from the original full-precised values into binary values that then can be stored and processed with only one bit per value. In doing this, the traditional neural network-based large-size speech emotion recognition models can be greatly compressed into smaller ones, which demand lower computational cost. To evaluate the feasibility of the proposed approach, we take a state-of-the-art speech emotion recognition model, i. e., convolutional recurrent neural networks, as an example, and conduct experiments on two widely used emotional databases. We find that the proposed binary neural networks are able to yield a remarkable model compression rate but at limited expense of model performance.

Index Terms— binary neural network, compact convolutional recurrent neural network, speech emotion recognition, green computing

1. INTRODUCTION

Automatic Speech Emotion Recognition (SER) has become one of active research topics over the past decades in both academic and industrial communities, due to its widespread applications in, such as natural and friendly human–machine communication [1–3]. Thanks to the tremendous success of deep learning in image and speech processing [4, 5], nowadays it has been frequently employed for SER as well, and continually reported to achieve the most state-of-the-art performance [6–12]. For example, Wöllmer et al. [13] took temporal dynamics of speech signal into account and extracted the long-dependent context utterance-level features using Long Short-Term Memory Recurrent Neural Networks (LSTM-RNNs). Han et al. [6] utilized Deep Neural Networks (DNNs) to estimate the segment-level emotion state probability distribution. This distribution is then used to compute an super-segment-level feature which are fed into an Extreme Learning Machine (ELM) for a final emotion state prediction. Mao et al. [7] used Convolutional Neural Networks (CNNs) to learn the salient affective features automatically which are robust and discriminative to emotion recognition. Chen et al. [10] utilized Convolutional Recurrent Neural Networks (CRNNs) with an attention model to learn discriminative features for SER.

All these developments have greatly advanced SER, which further facilitated its applications in real-life scenarios. However, it has to be noticed that almost all these systems were mainly designed in a server-client (central) manner [14]. That is, the models have been innovated to achieve better recognition performance, but without any consideration on the consumption of computational resource. Once models are trained, they are moved to the server (or cloud) side, where the powerful computational resource is often available. The data collected from the client side are then delivered to the server side through Internet for SER. Despite the efficiency of this framework, it suffers from many issues in realistic applications, including: i) privacy protection. The speech data collected from users are considered to be extremely private, as it may not only contain the emotion information that one needs, but also other highly sensitive information. To cope with this issue, some research has been done. For example, Zhang et al. [14] have intended to transfer vector quantitized statistic features rather than raw signals to the server. Albeit the irreversibility of these features, they still contain additional user information like the user identification. ii) limited network bandwidth. It is a common case where no Internet accessibility or limited network bandwidth is available, which constraints the application of SER systems.

All these issues highlight the necessity to shift the systems from such a central framework into a distributed framework, where the SER models can be run at individual devices, such as smartphones and intelligent speakers. By doing this, there is no need to upload the data and access the Internet, and thus the user private information can be well-protected. Note that, these individual devices, however, often lack disk storage, memory size, and battery power. All these analyses motivate us to compress the current SER models into smaller ones.

In this paper, we proposed a *binarization* approach to cope with the raised problem. Specifically, rather than using the full-precised values (normally 32-bit) to present the neural network weights, we prefer to using their binarized values, i. e., either $+1$ or -1. In doing this, the model can be stored with less disk storage, and can be processed in less computational complexity. To the best of our knowledge, this is the first time to investigate the binary neural network in the context of SER. Particularly, we selected the most recently developed CRNN model [10] for our evaluation. Overall, the main contributions of this paper include i) proposing a binarization approach to compress the most developed SER model; 2) evaluating the feasibility and effectiveness of the proposed model on two popularly used emotion databases.

This work was supported by the National Science Foundation of China under Grant 61772188 and National Key R&D Program of China Grant 2018YFC0831800.

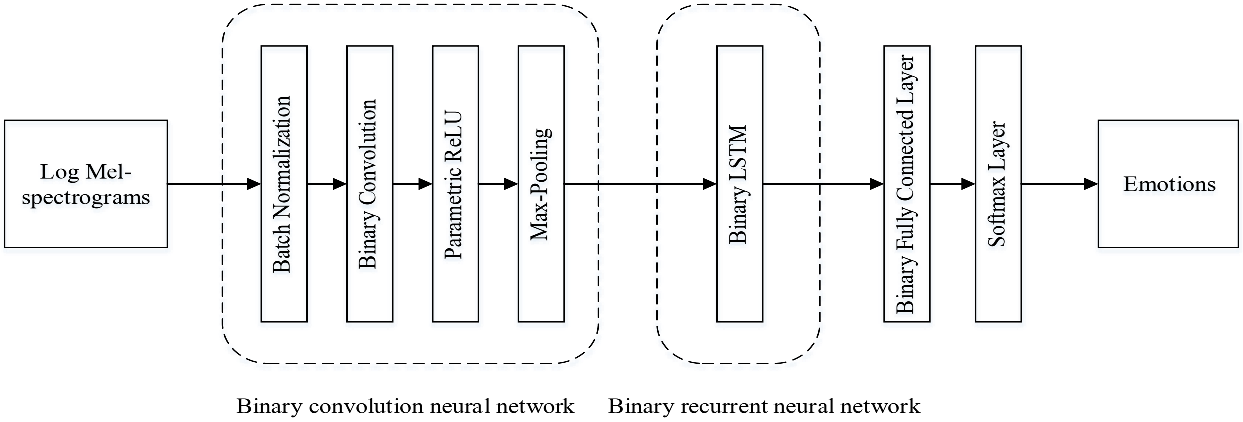

Fig. 1. The framework of the proposed compact convolutional recurrent neural network via binarization for speech emotion recognition, which consists of a binary CNN, a binary LSTM-RNN, and a binary fully-connected network.

2. RELATED WORK

Although neural network-based models have received great success in machine learning, the high requirement of computational cost severely limits their capability to be run in low-power devices. To address this problem, several methods have been proposed to reduce the complexity of neural networks over the past few years [15, 16], which can generally be grouped into binarization-based and pruning-based approaches.

Binarization-based approaches attempt to replace the real-valued parameters with binary values. The Binary Neural Networks (BNNs) were firstly proposed in [16], where Courbariaux et al. applied deterministic and stochastic sign functions to obtain binary weights in training. The authors further proposed to constraint the weights and activations to $+1$ and -1, leading to a dramatic storage reduction. Similar work was also done in [17], where Rastegari et al. binarized filters and inputs of convolution layers. Unlikely, *pruning-based* approaches believe that not all weights are useful for inference processing [15, 18]. Thus, removing the redundant and non-informative weights from the training network, the network size will then be reduced. Typical work can be found in [15, 18]. In more details, Han et al. [15] removed the weights according to a threshold value; whereas Chen et al. [18] implemented a hash function to randomly group the weights.

Despite the importance of model compression as mentioned in Section 1, in the context of speech processing, merely a handful of research studies were reported. For instance, in [19], Xiang et al. investigated the binarized DNNs for speech recognition. To date, there is no related studies in the SER literature, to our knowledge.

3. BINARY CONVOLUTIONAL RECURRENT NEURAL NETWORKS

In this work, we propose a novel Binary Convolutional Recurrent Neural Network (BCRNN) model for speech emotion recognition. As shown in Fig. 1, the model consists of a Binary Convolution Neural Network(BCNN), cascaded by a Binary Recurrent Neural Network (BRNN) and a Binary Fully Connected (BFC) layer. In this section, we detail the binarization function, which is the core operation of the proposed model. Then, we describe how to apply the function to construct BCNN, BRNN, and BFC, respectively. Finally, we discuss how we back propagate through these binarized neural networks.

3.1. Deterministic Binarization Function

Following previous work in [20], we employ the deterministic binarization function to constraint real-valued variables to either $+1$ or -1. Given a real-valued variable x, the function can be expressed as:

$$b = \mathrm{sign}(x) = \begin{cases} +1 & \text{if} \quad x \geq 0, \\ -1 & \text{otherwise}, \end{cases} \tag{1}$$

where b is the corresponding binarized variable. In other words, b is simply determined by the signs of x. As a consequence, it is very efficient to be implemented in practice.

Moreover, given an n-dimensional vector $\mathbf{X} \in \mathbb{R}^n$ and its corresponding binary vector $\mathbf{B} \in \{+1, -1\}^n$, a scaling factor α is introduced to deal with the massive loss between $\mathbf{X}$ and $\mathbf{B}$. Mathematically, L2 loss function is minimized to obtain an optimal $\alpha^\star$, the process of which can be formulated as follows:

$$\begin{aligned} \alpha^\star &= \arg\min_\alpha L(\alpha) = \arg\min_\alpha ||\mathbf{X} - \alpha\mathbf{B}||^2 \\ &= \arg\min_\alpha \mathbf{X}^\mathrm{T}\mathbf{X} - 2\alpha\mathbf{X}^\mathrm{T}\mathbf{B} + \alpha^2\mathbf{B}^\mathrm{T}\mathbf{B}. \end{aligned} \tag{2}$$

Thus, the optimal $\alpha^\star$ can be derived by setting the derivative of $L(\alpha)$ with respect to α to be zero:

$$\alpha^\star = \frac{\mathbf{X}^\mathrm{T}\mathbf{B}}{\mathbf{B}^\mathrm{T}\mathbf{B}}. \tag{3}$$

Note that, $\mathbf{B}^\mathrm{T}\mathbf{B}$ equals to n, i. e., the size of $\mathbf{X}$. Thus, Eq. (3) can be reformulated to:

$$\alpha^\star = \frac{\mathbf{X}^\mathrm{T}\,\mathrm{sign}(\mathbf{X})}{n} = \frac{\sum |X_i|}{n}. \tag{4}$$

Therefore, the optimal $\alpha^\star$ is the average over the absolute value of all real-valued elements X_i in $\mathbf{X}$.

3.2. Architecture of BCRNN

The proposed BCRNN is mainly made up of three components, i. e., BCNN to extract high-level representations from log Melsepctrograms (log-Mels), BRNN to obtain contextual information, and BFC to produce final emotion predictions.

3.2.1. *Binary Convolutional Neural Network*

A standard CNN structure consists of a batch normalization layer, a convolutional layer with activation function, and a pooling layer. In this work, we adopt BCNN instead, by conducting binary convolution in the convolutional layer.

The binary convolution layer consists of a set of filters, similar as in a typical convolutional layer. In particular, let us denote the weights in one filter and inputs of the binary convolution layer as $\mathbf{W} \in \mathrm{R}^{c \times w \times h}$ and $\mathbf{I} \in \mathrm{R}^{c \times w_{in} \times h_{in}}$, respectively, with $w_{in} \gg w, h_{in} \gg h$. When $\mathbf{W}$ is convolved across $\mathbf{I}$, the dot product $\mathbf{I_s}^{\mathrm{T}}\mathbf{W}$ is computed for each sub-tensor of $\mathbf{I}$, namely, $\mathbf{I_s}$, which has the same size of $\mathbf{W}$. In BCNN, $\mathbf{I_s}$ and $\mathbf{W}$ are binarized with $\mathbf{H} = \mathrm{sign}(\mathbf{I_s})$ and $\mathbf{B} = \mathrm{sign}(\mathbf{W})$, respectively. After that, two scaling factors α and β are further introduced, the optimization of which can be expressed as:

$$\alpha^\star, \beta^\star = \underset{\alpha,\beta}{\arg\min} ||\mathbf{I_s}^{\mathrm{T}}\mathbf{W} - \alpha\beta\mathbf{H}^{\mathrm{T}}\mathbf{B}||, \tag{5}$$

where $\alpha^\star\mathbf{H}$ and $\beta^\star\mathbf{B}$ would be the approximate estimations of $\mathbf{I_s}$ and $\mathbf{W}$, accordingly. Then we simplify the product of α and β in Eq. (5) with $\gamma = \alpha\beta$, and optimize γ according to the knowledge from Eq. (4) and a hypothesize that $\mathbf{I}$ and $\mathbf{W}$ are independent:

$$\gamma^\star = \alpha^\star\beta^\star = \frac{\sum |I_i||W_i|}{n} \approx \frac{\sum |I_i|}{n}\frac{\sum |W_i|}{n}, \tag{6}$$

where $n = c \times w \times h$, with I_i and W_i being elements in $\mathbf{I_s}$ and $\mathbf{W}$, respectively.

Considering that there are overlaps between sub-tensors, the scaling factor α is shared cross channels. As a consequence, we average the absolute values of the elements of input $\mathbf{I}$ across channels and obtain a matrix $\mathbf{A} = \frac{\sum |\mathbf{I}_{:,:,i}|}{c}$. Then, the matrix $\mathbf{A}$ convolves with a filter $\mathbf{k} \in \mathbb{R}^{w \times h}$ (where $\mathbf{k}_{ij} = \frac{1}{w \times h}$) to obtain a scaling factor matrix $\mathbf{K}$. The factor matrix $\mathbf{K}$ contains all possible factor β for all I_s. Therefore, the convolution between $\mathbf{W}$ and $\mathbf{I}$ can be approximated by the binary convolution operation:

$$\mathbf{I} * \mathbf{W} = (\mathrm{sign}(\mathbf{I}) * \mathrm{sign}(\mathbf{W})) * \beta\mathbf{K}. \tag{7}$$

Furthermore, we apply Parametric ReLU activation function so that the scaling factor for weights β can be estimated automatically by the activation function. In other words, β can be deemed as a parameter for the network to figure out itself.

3.2.2. *Binary Recurrent Neural Network*

After a sequence of high-level representations are extracted by BCNN, we feed the representations into a bi-directional RNN with binary-LSTM cell. The structure of binary-LSTM cells is simlar with traditional LSTM cells, but the weights and inputs of layers are constrained to binary values, i. e., $+1$ or -1. Assuming $\mathbf{x}_t$ is the input at the timestep t and $\mathbf{h}_{t-1}$ is the hidden state of the previous timestep $t-1$, the mathematical expression of LSTM structure can be expressed as:

$$\begin{aligned}
\mathbf{d}_t &= [\mathbf{x}_t, \mathbf{h}_{t-1}] \\
\mathbf{I}_t, \mathbf{F}_t, \mathbf{O}_t, \mathbf{G}_t &= \mathbf{W}\mathbf{d}_t \\
\{\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t\} &= \sigma(\{\mathbf{I}_t, \mathbf{F}_t, \mathbf{O}_t\}) \\
\mathbf{g}_t &= \tanh(\mathbf{G}_t) \\
\mathbf{c}_t &= \mathbf{f}_t \cdot \mathbf{c}_{t-1} + \mathbf{i}_t \cdot \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t \cdot \tanh(\mathbf{c}_t),
\end{aligned} \tag{8}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, and $\mathbf{o}_t$ denote the state the input gate, forget gate, and output gate, respectively, which helps to protect and control the cell state $\mathbf{c}_t$. Meanwhile, $\mathbf{c}_t$ is updated based on the old cell state $\mathbf{c}_{t-1}$ as well as the three gates.

In BRNN, we focus on binarizing weights and inputs of layers. Therefore, Eq. (1) can be applied to generate the binarized $\mathbf{W}$ and $\mathbf{d}_t$ accordingly, which can further be represented as $\mathbf{W}^b$ and ${\mathbf{d}_t}^b$, respectively. Then, similarly as in the BCNN model, scaling factors α and β are introduced to approximate the term $\mathbf{W}\mathbf{d}_t$ in Eq. (8) by $\alpha\mathbf{W}^b\beta{\mathbf{d}_t}^b$. We can use the optimal solutions from Eq. (5) to deal with this approximation.

Additionally, apart from BCNN and BRNN, we further apply a BFC layer to take the place of a traditional fully connected layer. Again, both the inputs and weights of the layer are shifted from real values to binary values, while the related optimal scaling factors can be estimated as in Eq. (5).

3.3. Propagating Gradients

In backward propagation, the gradients are computed with respect to the estimated binary weights, to update the parameters. Note that, during the learning phase, both the real-valued weights and the estimated binary weights are demanded. While the real-valued weights are used similarly as in the conditional back-propagation process, the gradient for $sign$ function is problematic as the derivative of it is zero almost everywhere. Following previous work in [20], we compute it using the straight-through estimator approach [20]. In specific, given a $sign$ function $sign(r)$, the gradient can be estimated by $\frac{\partial sign}{\partial r} = r1_{|r|\leq 1}$. Then, the estimator g_r of the gradient $\frac{\partial C}{\partial q}$ can be obtained by:

$$g_r = g_q 1_{|r|\leq 1}, \tag{9}$$

where C is the loss function, and the gradient is canceled when r is too large. For a more in-depth description of the straight-through estimator approach, the reader is referred to [20].

4. EXPERIMENTS AND RESULTS

To evaluate the feasibility and effectiveness of the proposed BCRNN framework for SER, we conducted extensive experiments on two widely used databases in the affective computing community, i. e., IEMOCAP and Emo-DB.

4.1. Databases and Acoustic Features

The IEMOCAP database contains approximately 12 hours of recordings from five pairs of experienced actors [21]. The recordings were then segmented into utterances and further annotated into both categorical and dimensional emotions. In this work, we only considered the emotion classification task with five categories, i. e., happiness, anger, sadness, frustration and neutral, since all other categories appear very sparsely in the dataset. Besides, for our experiments, only the improvised utterances were considered, resulting in 2 837 sample in total in order to keep in line with the setting in [10]. The other database we evaluated is Emo-DB [22], which consists of 535 utterances that collected by ten professional actors, covering seven emotions (i. e., neutral, fear, joy, angry, sadness, disgust and boredom).

LogMel filterbanks are used as inputs of the proposed model. Then, the salient representations for each emotion are automatically learnt from them. To compute log-Mels, speech signals were first split into frames with Hamming windows of $25ms$ and a step size of $10ms$. Then, log-Mels were computed with 40 filterbanks.

Table 1. Performance comparison in term of Unweighted Average Recall (UAR [%]) between the proposed BCRNN with the baseline system and other state-of-the-art systems on the IEMOCAP and Emo-DB databases.

Approach	IEMOCAP	Emo-DB
DNN-ELM [6]	51.2	71.6
3-D ACRNN [10]	64.2	81.5
Full-precision CRNN	62.4	80.1
BCRNN	61.9	79.7

4.2. Experimental setups

For our experiments, both IEMOCAP and Emo-DB were split into training, development, and test sets with a speaker independent strategy. In addition, we took an online standardization over the databases before feeding them into the neural networks, in order to reduce the influence of speaker variation. To augment the data, we split all the utterances into sub-segments with a fixed length 3 s. Zero padding was then applied if the sub-segments are less than 3 s. In the training phase, each sub-segment was considered independently, with the same label information with its corresponding utterance. Nevertheless, in evaluation phase, a max pooling was utilized over the posterior probabilities of the sub-segment predictions, to come up with an utterance-level prediction.

As to the network structure, the convolution layer of BCNN has 128 feature maps and the filter size is 5×3. Max-pooling is performed with the size 2×2. Meanwhile, the stacked BRNN contains 128 binary-LSTM cells. As a result, a sequence of 256-dimensional features are generated per utterance. For the binary fully connected layer, we set the number of hidden units to be 64, Finally, a softmax layer is attached for a final emotion state prediction. In the network training process, we utilized the cross-entropy loss as the objective function, which was minimized by Adam optimizer with the learning rate 10^{-5}. The mini-batch size was set to be 40.

To measure the performance of the proposed model, we utilized the widely used metric Unweighted Average Recall (UAR), i. e., the sum of classwise recall divided by the number of classes, for emotion recognition.

4.3. Results and Discussions

To evaluate the performance of the proposed BCRNN, we selected the following three state-of-the-art models for performance comparison. i) DNN-ELM [6]. This model uses DNN to extract representation and EML for emotion classification. ii) 3-D ACRNN [10]. In this model, 3-D log-Mels and attention mechanism are used. iii) Full-precision CRNN [10]. This is our baseline model, with traditional full-precised weights and inputs. For all these controlled experiments, we retained the default network structures and training hyper-parameters in the previous work [6, 10].

Table 1 shows the obtained recognition results in terms of UAR for different SER models on both IEMOCAP and Emo-DB databases; whereas Table 2 compares the size of each investigated models. From Table 1, we can observe that the model 3-D ACRNN performs the best among the baseline systems and other state-of-the-art systems. This mainly attributes to i) the end-to-end framework design, which aims to directly learn salient representation from Mel-spectrogram; ii) its attention mechanism can filter some redundant representations; iii) the model complexity. As can be seen from Table 2, the 3-D ACRNN is characterized with the largest model size of 323.46 MB, which is almost three times of the full-precision CRNN model and 26 times of the DNN-ELM model. For our selected baseline (i. e., full-precision CRNN), one can observe that it is slightly inferior to the 3-D ACRNN but with approximate 1/3 the model size of 3-D ACRNN.

Table 2. Model size comparison between the proposed Binary Convolutional Recurrent Neural Network (BCRNN) with its original full-precised system and other state-of-the-art systems.

Approaches	Model size (MB)
DNN-ELM [6]	12.33
3-D ACRNN [10]	323.46
Full-precision CRNN	105.48
BCRNN	4.34

When we compared the proposed BCRNN with the baseline model, one can observe that it performs very competitive to the baseline model on both IEMOCAP (i. e., 61.9 % vs. 62.4 %) and Emo-DB (i. e., 79.7 % vs. 80.1 %) databases. However, it considerably compresses the original full-precision CRNN, from 105.48 MB to 4.34 MB, leading to an approximate model compression rate of 26. When we further compared BCRNN with other two models, it can be seen that it outperforms the DNN-ELM model with a large performance gain (i. e., 10.7 % and 8.1 % for IEMOCAP and Emo-DB corpora, respectively), even though the model is also small. It can also be seen that it performs a slightly worse than 3-D ACRNN model, but with a remarkable model size compression.

Besides, for the binary convolution, our model needs to execute the number of cN_fN_i binary operations and N_i add operation. Note that common CPU is able to process 64 binary operations per CPU clock time. Therefore, our model is much faster than full-precise models. Mathematically, the speedup is calculated by $S = \frac{cN_fN_i}{\frac{1}{64}cN_fN_i+N_i} = \frac{64}{1+\frac{64}{cN_f}}$. It means the speedup rate is relating to the number of channels and the size of filters. In our setting (i. e., the channel: 128; filter size: 5×3), we obtained a speedup rate with 61.9 %.

Overall, all these results promote the possibility to apply most recently developed deep learning models to the devices with limited storage and computational resource for SER.

5. CONCLUSION

To facilitate the application of Speech Emotion Recognition (SER) to embedded devices, in this paper we proposed a Binary Convolutional Recurrent Neural Network (BCRNN). In the proposed model, the weights and inputs of the layers are constraint to binary values that are $+1$ or -1. Firstly, log-Mels are extracted from the raw speech signals. Then, BCRNN takes log-Mels as inputs to generate higher-level discriminative representations for emotion classification. IEMOCAP and Emo-DB corpora are used to evaluate the performance of the model in term of unweighted average recall. Results indicate that our proposed model can yield comparable results compared with state-of-the-art methods but with a high model size compression rate. The complex convolution operations are largely accelerated by simple binary operations. Therefore, it increases the possibility to integrate SER systems on embedded devices where computational resources are limited.

Encouraged by the obtained results, our future work will focus on advanced approaches that can lead to better accuracy and higher compression rate for SER.

6. REFERENCES

[1] Zhihong Zeng, Maja Pantic, Glenn I. Roisman, and Thomas S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, Mar. 2009.

[2] Zixing Zhang, Nicholas Cummins, and Björn Schuller, "Advanced data exploitation for speech analysis – An overview," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 107–129, July 2017.

[3] J. Han, Z. Zhang, N. Cummins, and B. Schuller, "Adversarial training in affective computing and sentiment analysis: Recent advances and perspectives," *IEEE Computational Intelligence Magazine*, 2018, 13 pages.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436, May 2015.

[5] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, Canada*. IEEE, 2013, pp. 6645–6649.

[6] Kun Han, Dong Yu, and Ivan Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in *Proc. INTERSPEECH*, Singapore, 2014.

[7] Qirong Mao, Ming Dong, Zhengwei Huang, and Yongzhao Zhan, "Learning salient features for speech emotion recognition using convolutional neural networks," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2203–2213, Dec. 2014.

[8] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou, "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China*. IEEE, 2016, pp. 5200–5204.

[9] Jing Han, Zixing Zhang, Fabien Ringeval, and Björn Schuller, "Prediction-based learning for continuous emotion recognition in speech," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017, pp. 5005–5009.

[10] Mingyi Chen, Xuanji He, Jing Yang, and Han Zhang, "3-D convolutional recurrent neural networks with attention model for speech emotion recognition," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1440–1444, Oct. 2018.

[11] Zixing Zhang, Jing Han, Xinzhou Xu, Jun Deng, Fabien Ringeval, and Björn Schuller, "Leveraging unlabelled data for emotion recognition with enhanced collaborative semi-supervised learning," *IEEE Access*, vol. 6, no. 1, pp. 22196–22209, Dec. 2018.

[12] Zixing Zhang, Jing Han, and Björn Schuller, "Dynamic difficulty awareness training for continuous emotion prediction," *IEEE Transactions on Multimedia*, vol. PP, Sep. 2018, 13 pages.

[13] Martin Wöllmer, Florian Eyben, Stephan Reiter, Björn Schuller, Cate Cox, Ellen Douglas-Cowie, and Roddy Cowie, "Abandoning emotion classes – Towards continuous emotion recognition with modelling of long-range dependencies," in *Proc. INTERSPEECH*, Brisbane, Australia, 2008, pp. 597–600.

[14] Zixing Zhang, Eduardo Coutinho, Jun Deng, and Björn Schuller, "Distributing Recognition in Computational Paralinguistics," *IEEE Transactions on Affective Computing*, vol. 5, no. 4, pp. 406–417, Oct. 2014.

[15] Song Han, Huizi Mao, and William J Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[16] Matthieu Courbariaux, Yoshua Bengio, and Jean Pierre David, "Binaryconnect: training deep neural networks with binary weights during propagations," in *Proc. International Conference on Neural Information Processing Systems (NIPS), Montreal, Canada*, 2015, pp. 3123–3131.

[17] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Proc. European Conference on Computer Vision (ECCV), Amsterdam, Netherlands*. Springer, 2016, pp. 525–542.

[18] Wenlin Chen, Stephen Tyree, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen, "Compressing neural networks with the hashing trick," in *Proc. International Conference on International Conference on Machine Learning (ICML), Lille, France*, 2015, pp. 2285–2294.

[19] Xu Xiang, Yanmin Qian, and Kai Yu, "Binary deep neural networks for speech recognition," in *Proc. INTERSPEECH, Stockholm, Sweden*, 2017, pp. 533–537.

[20] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[21] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Journal of Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, Dec. 2008.

[22] Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F Sendlmeier, and Benjamin Weiss, "A database of German emotional speech," in *Proc. INTERSPEECH*, Lisbon, Portugal, 2005, pp. 1517–1520.