Distilled Binary Neural Network for Monaural Speech Separation

Xiuyi Chen[1,2,3], Guangcan Liu[1,2,3], Jing Shi[1,2,3], Jiaming Xu[1,2*], Bo Xu[1,2,3,4]
[1]Institute of Automation, Chinese Academy of Sciences (CASIA). Beijing, China
[2]Research Center for Brain-inspired Intelligence, CASIA
[3]University of Chinese Academy of Sciences
[4]Center for Excellence in Brain Science and Intelligence Technology, CAS. China
{chenxiuyi2017, liuguangcan2016, shijing2014, jiaming.xu, xubo}@ia.ac.cn

***Abstract*—Monaural speech separation, aiming at solving the cocktail party problem, has many important application scenarios, most of which ask for the real-time response, high energy efficiency and efficient storage. However, the state-of-the-art Deep Neural Network based separation models usually require huge memory and computation for the 32-bit floating point multiply accumulations, hence most of them cannot meet those requirements. Recently, there are many methods proposed to solve the problem, and binary neural networks have drawn many attentions for they compress and speed up its counterparts at the cost of some performance. Hence, in this paper, we binarize Deep Neural Network based separation models, aiming to deploy them on embedded devices for real-time applications. Furthermore, we improve the separation performance by integrating knowledge distillation into the training phase of binary neural network based models, which is referred as Distilled Binary Neural Network (DBNN). To the best of our knowledge, DBNN is the first attempt to integrate two types of model compression. In the experiments, we demonstrate the effectiveness of our proposed method, which successfully binarizes the Deep Neural Network based separation models with a comparable performance.**

***Index Terms*—Monaural Speech Separation, Binary Neural Network, Knowledge Distillation**

I. Introduction

Speech separation is to separate the target speech from background interference, which aims at solving the cocktail party problem described by Colin Cherry [1], and it characterizes the human attentional ability as humans are adept at listening to one voice in the midst of other conversations and noise [2]. Monaural speech separation with only one channel signal available is more difficult and fundamental to separate the clean target speech from a mixed noisy speech.

Monaural speech separation has many important application scenarios, because there are many hearing-impaired people in the real world and also some annoying auditory scenes where even normal human feel impossible to concentrate for long. Thus, it is significant to figure out how do humans segregate speech sounds and build a machine to do the task [2]. Recently, supervised speech separation has substantially advanced the state-of-the-art performance by leveraging large training data and increasing computing resources, and Deep Neural Network (DNN) based separation methods are the most practical ones [3]. DNNs have been successfully applied to speech enhancement [4], speaker-dependent speech separation [5], [6], target-dependent speech separation [7] and speaker-independent multi-talker speech separation [8], [9]. However, many state-of-the-art DNN-based models usually require huge memory and computation, it's hard to deploy those models on real-time embedded platforms with limited computational capacity, such as smart phones, hearing aids and cochlear implants. Hence, there is still a huge gap between the most existing speech separation models and their practical applications for the cocktail party problem, including hearing prosthesis, mobile telecommunication and frontend for Automatic Speech Recognition (ASR).

The gap between the DNN-based models and their deployment is a general issue beyond speech separation, so there are already substantial research efforts invested in speeding up DNNs at run-time, which can be roughly categorized into three types: network pruning [10], [11], network quantization [12]–[15] and knowledge distillation [16], [17]. In this paper, we mainly attend to binarize the DNN-based separation models with a comparable performance to bridge the gap, not only because DNN-based model are themselves computationally faster than Recurrent Neural Network (RNN) based models, but because Binary Neural Networks (BNNs), with weights and activations both quantized to ± 1, are the extremely low-precision networks. Although BNN is not a new thing, there are some significant performance degradation (in accuracy) and those efforts rarely focus on speech separation tasks. We attribute this degradation to BNNs' impaired learning ability as BNNs drastically reduce the weights and activations representation precision. Hence, we expect to improve the performance by leveraging knowledge distilled from their counterparts (such as DNNs) during training phase as they have the same network structures and DNNs learn better.

We first binarize the DNN-based separation models to compress and speed up those models at the cost of performance. Then, we exploit knowledge distillation to help the training of binary neural network and finally find our DBNN guarantees a better performance. Our contributions are three-fold:

(1) We successfully binarize the DNN-based models to separate the mixed speech, which is a regression problem, harder than classification problems. (2) Two types of model compression, BNN and knowledge distillation, are integrated

* Corresponding author

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

at the training phase for further improving the performance and stability on the BNN with a limited size. (3) Experimental results show that we can binarize the DNN-based separation models with a little loss in performance and our DBNN performs better than BNN.

The organization of this paper is as follows: Section II discusses the background of our work. Section III introduces the proposed methods, including training process of BNNs, and the integration of BNNs and knowledge distillation. Section IV presents the experimental setting and the effectiveness of our proposed approaches by showing the experimental results. Finally, conclusions are given in the last Section.

II. RELATED WORK

A. DNN-based Speech Separation

Speech separation is to separate the clean target speech from a noisy mixed speech, among which the target speech is masked by other speech or noise. For decades, researchers have been committed to solving the problem, and put forward many effective methods.

Before deep learning, Nonnegative Matrix Factorization (NMF) [18] and Computational Auditory Scene Analysis (CASA) [19] are the most popular techniques. NMF factorizes time-frequency spectral representations by decomposing speech signal into sets of bases and weight matrices. CASA is based on perceptual principles of auditory scene analysis and aims to estimate a time-frequency mask that isolates the signal components belonging to different speakers. As a result of Shallow Learning, NMF and CASA only achieved limited success in single-channel speech separation.

At present, deep learning has made great breakthroughs, especially in the field of image and speech recognition. Hence, with the booming of deep learning, DNNs have been successfully applied to the speech separation problem. In the first, DNNs is trained to learn a mapping from noisy features to a time-frequency representation of the target of interest and Wang et al. [20] compared separation results by using different training targets, including the Ideal Binary Mask (IBM), the Ideal Ratio Mask (IRM) and so on. Then, a discriminative training objective [5] is proposed, which takes into account the similarity between the prediction and other sources when minimizing the squared error between the output of neural network and the target reference, and Wang et al. [6] developed this idea, aiming to preserve the mutual difference between two source signals during training. A novel maximum likelihood approach is used in DNN-based speech separation with a reasonable assumption that the prediction error vector of DNN follows the Gaussian distribution [21]. Deep Clustering (DC) [8] method and Deep Attractor Network (DANet) [22] employed a clustering algorithm in the embedding space to generate a partition of the time-frequency units. Permutation Invariant Training (PIT) method [9] solved the permutation problem by pooling over all possible permutation for N mixed sources ($N!$ permutations), and minimized the source reconstruction error no matter how labels are ordered. Xu et al. [23] proposed a unified Auditory Selection framework with Attention and Memory (ASAM) to solve the cocktail party problem, using top-down and bottom-up attention and memory mechanism. Furthermore, Wang's lab [24] has already built a program, which uses deep neural networks to solve the cocktail party problem through an advanced hearing aid.

Among the proposed methods, there are roughly three type: speaker-dependent speech separation [5], [6], target-dependent speech separation [7] and speaker-independent multi-talker speech separation [8], [9], [23]. Although, DNN-based methods have substantially advanced the state-of-the-art performance by leveraging large training data and increasing computing resources [3], there are some constraints when such advanced program or models are applied in practice. The biggest challenge is the deployment on resource-constrained embedded devices. When deploying such program or separation models, we have to consider the response speed, power constraints, energy budgets and memory overhead. Hence, Enea et al. [25] presented a study of the impact of reduced precision on deep regression RNNs, however, their weight precision lower than 4 bits did not guarantee a good performance and RNNs are themselves computationally complex.

B. Model Compression

As mentioned in the introduction, there are roughly three types of model compression, and here we mainly introduce BNN and knowledge distillation, respectively.

BNN, the neural network with binary weights and activations at runtime, is actually an extreme of Quantized Neural Network (QNN) [14]. Although already existed, binary neural network is recently first proposed by Matthieu et al. [12], who proposed two different binarization functions and applied approximate back-propagation to train the binary neural networks from scratch; then Mohammad et al. [13] extended this idea, proposed new quantization function and first applied BNNs to large-scale classification problem, and Tang et al. [15] attempted to train a compact binary neural network with high accuracy through a careful analysis of previous work on BinaryNets, gave some effective training strategies for BinaryNets and proposed a new kind of regularization term which pilots the weights to 1 or -1 other than to 0 as in a L_2 regularizationis suitable for BinaryNets.

As a very promising method for model compression, BNNs have been demonstrated to compute and store efficiently, and achieve nearly the state-of-the-art results on small scale datasets such as MNIST, CIFAR-10 and SVHN datasets [12] with a larger structure or suffer some loss of accuracy on large scale datasets such as ImageNet [13], [15]. Furthermore, Esser et al. [26] exploited the methodology of BNN to run inference tasks on neuromorphic hardware, approaching the state-of-the-art classification accuracy on eight standard datasets while preserving the hardware's underlying energy-efficiency. Hence, in this paper, we mainly attend to how to train binary neural networks with good performance on speech separation, and do not focus much on the computation complexity and memory consumption, because BNNs have been demonstrated to compute and store efficiently and im-

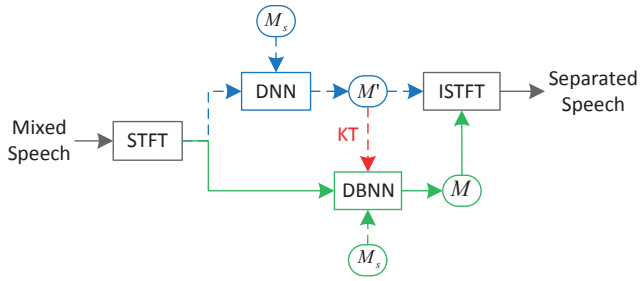

Figure 1. DBNN for monaural speech separation. We first train the DNN-based model along the blue lines and then train the DBNN-based model along the green and red lines. The dotted lines are used at the training phase and KT stands for knowledge transfer, which means we transfer the knowledge distilled from the pre-trained DNN to the corresponding DBNN at the training phase. M_s is the training target and M' and M are the masks predicted by DNN and DBNN, respectively. STFT is used to extract features from the time-domain waves (mixed speech) and ISTFT is used to generate time-domain waves (separated speech) from time-frequency-domain features. Note that we actually train a BNN only along the green lines.

plemented on both general-purpose and specialized computer hardware [12], [26], [27]. And Galloway et al. [28] showed that for binarized neural networks, difficulty in training leads to difficulty when attacking. Moreover, BNNs can achieve a better performance by extending to QNNs at the cost of computation and storage [14].

Knowledge distillation, also known as model compression [29], can be regarded as a teacher-student training strategy, where a very cumbersome teacher model extracts structure from the large, highly redundant datasets and then transfer the distilled knowledge to the student model [16]. The teacher model could be an ensemble of models or a very large model and the knowledge distilled from the teacher model can have various forms. Bucila et al. [29] proposed to train a neural network to mimic the output of a cumbersome ensemble by training it on the unlabeled data labeled by the ensemble. Hinton et al. [16] used the soft targets predicted by the teacher model to train a small network for deployment. Romero et al. [17] used both the final output and intermediate hidden layer values of the teacher network to teach information to the student network. Yim et al. [30] transferred the distilled knowledge as the flow between two layers and showed the gifts from knowledge distillation.

And the network pruning usually compress models by pruning less important weights and refine the models. Although, there are already three type of model compression, there are still something to improve. To the best of our knowledge, there is no work of integrating two of those methods before. Through a careful analysis of those three compression methods, we find knowledge distillation is compatible with BNN or network pruning.

III. PROPOSED METHODS

In this paper, we mainly show how to binarize DNN for speech separation and find the knowledge distillation can help the training of BNN, which integrates two types of model compression methods and demonstrates our DBNN performs better than BNN. Thus, following a common setting, we use the STFT magnitude spectra of mixed speech as input feature X, and the Ideal Ratio Mask (IRM) of clean speeches as the basic target M_s. To separate the mixed speech, we first exploit the already trained DNN, BNN or DBNN to estimate the mask M of the target speech, formulated as $M = \mathrm{NN}(X)$, where $\mathrm{NN}(\cdot)$ represents the mapping function of DNN, BNN or DBNN. We then estimate the clean speech s as $\mathrm{ISTFT}(M \circ X \circ e^{i\theta})$, where $\circ$ is the element-wise product of two operands, θ is the phase information of the mixed speech. Fig. 1 shows the training process of DBNN. First, we normally train a well-performing DNN-based separation model using IRM as target. Then, we train the corresponding DBNN by exploiting M' predicted by the DNN-based model as the form of knowledge, together with IRM. Finally, our DBNN performs inference (along the solid line) as a well trained BNN. As Fig. 1 shows, our DBNN differs from BNN mainly on the training, that is we can leverage additional information distilled from the pre-trained DNN teacher. Hence, our DBNN does not introduce additional computation at runtime, compared with BNN.

A. Binarizing DNN for Speech Separation

Though it is easy to build a BNN-based model by replacing the float32 weights in DNN with binary weigths and using a binarization function as the activation function, there are some tough challenges training a well-performance BNN. First, the binarization function is non-differentiable, making back-propagation impossible. Second, the binary weights cannot be updated directly, which easily leads to learning failure. Finally, BNN-based models have limited accuracy on large scale datasets or complex tasks, which we will attempt to alleviate in Section III-B. Thus, we here mainly focus on how to binarize DNN-based models for speech separation with a good performance and follow the common setting, using the magnitude of the mixed spectrum as input feature and the IRM as training target. To address the first two problems mentioned above, we follow the previous work [12] and adopt two key strategies. The first idea is to associate a real weight w_r with each binary weight w_b during training, and the other is to replace the non-differentiable binarization function with a continuous one during back-propagation.

In this paper, we integrate two key strategies by proposing a new binarization function $\mathrm{binarize}(\cdot)$, which has the form as Fig. 2 shows. We use the binarization function to binarize the weights and use it as activation function to binarize the activations.

Inspired by the previouse work [12], [16], as Alg. 1 demonstrates, we put forward a new method of training BNN (here that is DBNN) by leveraging additional information distilled from the pre-trained DNN teacher along the red line as shown in the Fig. 1. We associate a real weight w_r with each binary weight w_b via equation $w_b = \mathrm{binarize}(w_b)$ during the whole procedure for training. During forward propagation at training-time and running-time, we use w_b for matrix computation, which dramatically improves the energy efficiency and compu-

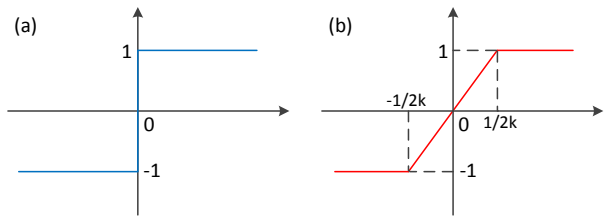

Figure 2. (a) The Binarization function; (b) An approximated shape of the function, which is differentiable.

tation speed. While during backward propagation, we have to use w_r for the weights updating because w_b cannot accumulate the gradient updates over multiple mini-batches. Furthermore, to reconcile the contradictory of updating the real weight and propagating the binary weight, we integrate the task-related loss term and a variant of the L_2 regularization proposed in [15] with a controlling parameter ℓ to match the two weights (the binary weight and corresponding weigh) by driving the real weights to near $+1$ and -1 for binarization.

The binarize$(\cdot)$ function, whose derivative is zero almost everywhere, makes it apparently incompatible with back-propagation. Thus, the basic idea is to replace it with a continuous one during back-propagation. As shown in Fig. 2, we here propose a new approximation as follows:

$$x_b = \text{hardtanhk}(x) = \begin{cases} +1 & \text{if } x > 1/2k\,; \\ -1 & \text{if } x < -1/2k\,; \\ 2kx & \text{otherwise}\,. \end{cases} \tag{1}$$

Our process is similar to the method in [12], but our process is more powerful since the parameter k can be learned by SGD, which softly encourage the real value toward binary value by approximating the binarization function with k growing. The approximated function has the same idea with Bounded rectifiers [31], but with the k as a hyper-parameter for simplicity in this paper. In this way, we can propagate the error backward and the derivative of the activation function can be formed as follows:

$$f'(x) = \text{mask}(x,k) = \frac{\partial \text{hardtanhk}(x)}{\partial x} = \begin{cases} 2k & \text{if } |x| \le 1/2k\,; \\ 0 & \text{otherwise}\,. \end{cases} \tag{2}$$

We also use the clip$(\cdot)$ function to constrain the w_r between -1 and 1 as [12]. We adopt strict binary neural network, that is all of the weights and activation except the output layer is binary. We exploit the hard sigmoid function Hsigmoid$(\cdot)$ as the nonlinear function in the output layer, which can be formed as follows:

$$\text{Hsigmoid}(x) = \max(0, min(1, \frac{x+1}{2}))\,. \tag{3}$$

Hsigmoid$(\cdot)$ function can be regarded as a fast approximation of the sigmoid function to get the IRM.

Algorithm 1 Training a L layers DBNN

Require: a minibatch of inputs and targets (X, M_s), previous weightsW_r, previous BatchNorm parameters θ, regularization controlling parameter ℓ, previous learning rate lr, learning rate decay factor d_c.
Also require soft targets M' predicted by the pre-trained teacher model and ensemble weight λ when training DBNN.
Ensure: updated weights W_r, updated BatchNorm parameters θ and updated learning rate lr.

```
{1. Computing the gradients:}
{1.1. Forward propagation:}
A^0 ← X
for l = 1 → L do
    W_b^l ← Binarize(W_r^l)
    S^l ← A^{l-1} W_b^l
    A^l ← BatchNorm(S^l, θ^l)
    if l < L then
        A_b^l ← Binarize(A^l)
    else
        M ← Hsigmoid(A^L)        // get predicted mask
    end if
end for
{1.2. Backward propagation:}
if training DBNN then
    L = Ensemble(M, M_s, M', λ)        // get loss
else
    L = MSE(M, M_s)
end if
g_{A^L} ← ∂L/∂M ∂M/∂A^L        // knowing M, A^L and L
for l = L → 1 do
    if l < L then
        g_{A^l} ← g_{A_b^l} ∘ mask(A^l, k)
    end if
    (g_{S^l}, g_{θ^l}) ← BackBatchNorm(g_{A^l}, S^l, θ^l)
    g_{A_b^{l-1}} ← g_{S^l} W_b^l
    g_{W_b^l} ← g_{S^l} A_b^{l-1}^T
    gnorm_{W_b^l} ← -2ℓ W_r^l        // for regularization
end for
{2. Accumulating the parameters gradients:}
for l = 1 → L do
    θ^l ← Update(θ^l, lr, g_{θ^l})
    W_r^l ← Clip(Update(W_r^l, lr, g_{W_b^l}, gnorm_{W_b^l}), -1, 1)
    lr ← d_c * lr
end for
```

B. Training Model with Knowledge Distillation

After binarizing the DNN-based separation models, we enjoy two remarkable benefits. The first is that BNN drastically reduces memory size and accesses due to its binary weights and activations only using one-bit in the memory, and the other is that BNN can drastically improve computation and energy efficiency by replacing the 32-bit floating point multiply-accumulations with 1-bit XNOR-count operations [12].

However, we find it hard to train BNN with a comparable

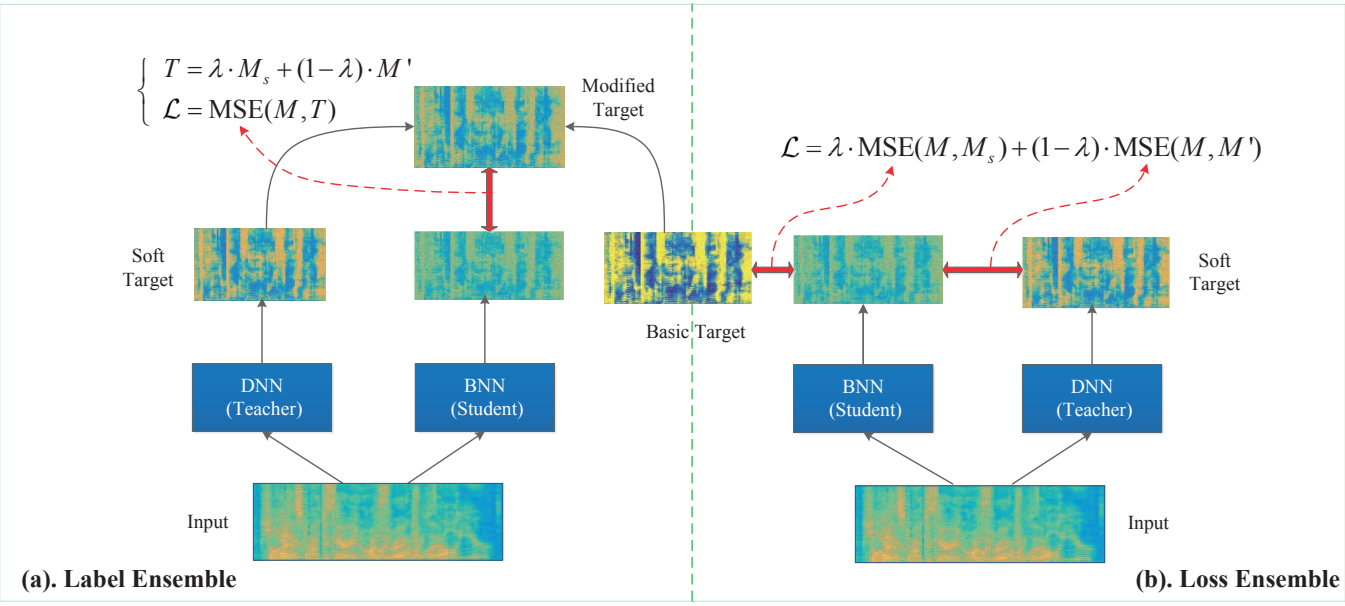

Figure 3. Knowledge distillation for our DBNN. Two ensemble methods to integrate the basic (original) target (that is the ideal ratio mask of target speech) and knowledge (in the form of soft target) distilled from the teacher model, label ensemble and loss ensemble, respectively.

separation performance, especially in a limited size, which we believe is due to BNN's impaired learning ability. Hence, in this paper, we mainly attend to how to train binary neural networks with good performance on speech separation, and pay less attention to the computation complexity and memory consumption, because BNNs have been demonstrated to compute and store efficiently and implemented on both general-purpose and specialized computer hardware. Our key idea comes from the intuition that BNN can be regarded as a "small and simple" network for BNN drastically reduces the redundancy of its counterpart. Thus, integrating knowledge distillation into the training phase of BNN would improve the performance. Knowledge distillation aims at a "small and good" student model for deployment by training the student model with knowledge distilled from a cumbersome teacher model. Thus, knowledge distillation is naturally compatible with BNN. Some other model compression methods, such as network pruning [10], deep compression [11], HashedNet [32], actually conflict with BNN for they all reduce the redundancy of the original network, one by pruning the parameters of the network while the other by reducing the weight and activation representation precision.

Hinton et al. [16] has demonstrated that it could guarantee a better result of classification tasks by integrating the original hard target and the soft target predicted by the teacher model. However, speech separation is a regression task rather than classification tasks, the ensemble methods in classification tasks maybe not fit for speech separation task. Hence, we here explore two ensemble methods, label ensemble and loss ensemble, respectively, as Fig. 3 shows. We first train a good DNN-based separation model by minimizing the Mean Square Error (MSE) between the estimated mask M' and the basic target M_s. When training DBNN, label ensemble in the form of:

$$\begin{cases} T = \lambda \cdot M_s + (1-\lambda) \cdot M'; \\ \mathcal{L} = \mathrm{MSE}(M, T), \end{cases} \tag{4}$$

where T is the ensemble target, M is the mask predicted by DBNN and $\mathcal{L}$ is the loss based on MSE, can be explained that we use the target M' predicted by the teacher model to modify the basic target M_s with a weight λ, while loss ensemble simply uses a weighted average of two different objective functions as follows:

$$\mathcal{L} = \lambda \cdot \mathrm{MSE}(M, M_s) + (1-\lambda) \cdot \mathrm{MSE}(M, M'). \tag{5}$$

The teacher model has more powerful modeling capability, therefore, can extract more structure from the training data and then transfer the distilled knowledge to DBNN at the training phase for a better performance. Furthermore, knowledge distillation only plays part at the training phase, which does not change the runtime of DBNN, compare with BNN. Thus we expect our DBNN trained by the knowledge distilled from the knowledgeable teacher can achieve a better separation performance than the BNN trained with basic target only, using the same runtime.

IV. Experiments

A. Experimental Setting

We evaluate the performance of our proposed approach for speech separation using TIMIT corpus [33], which is widely used in the previous works [5], [6]. There are 630 speakers in the corpus with ten sentences per speaker. We, respectively, choose eight sentences from a male and a female speaker

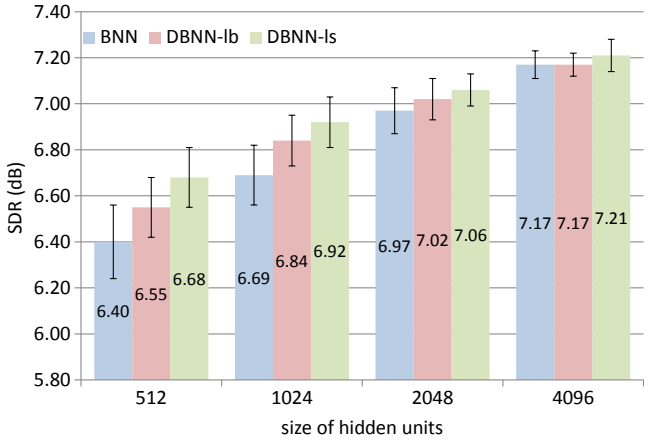

Figure 4. Speech separation performance by the BNN models and DBNN models with different model size.

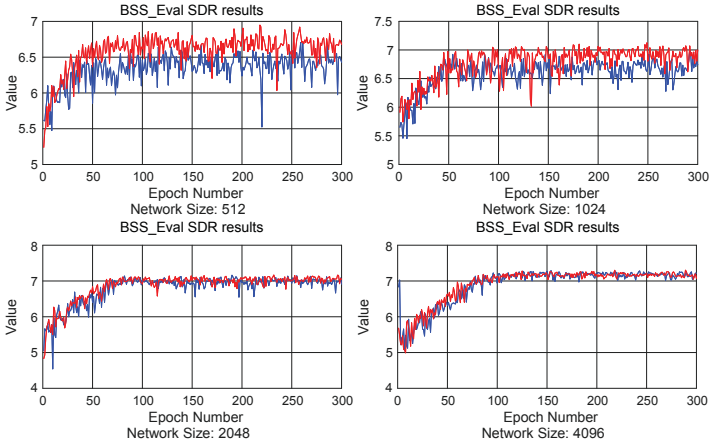

Figure 5. Training process of BNN models and corresponding DBNN models with different model size. Blue lines are for BNN models ,red lines for DBNN models

for training. The other two sentences respectively from the male and the female are used as the development set and the remaining sentences are used as the test set. All sentences are normalized to be with equal power. In order to rich the variety of the training samples, we circularly shift the male speaker's signals and mix them with utterances from the female.

In this paper, we use the most common feature and target, that is using the STFT magnitude spectra as input feature and the ideal ratio mask as target. The STFT magnitude spectra are obtained by using STFT with a frame size of 32 ms and 16 ms shift. Moreover, instead of using a context window, we just predict the mask according to the current frame of the STFT magnitude spectra because the context window could increase the complexity of the computation by increasing the input dimensionality if context window is greater than 1. So the context window conflicts with the motivation of BNNs. What's more, [5] has proved that the context window does not introduce significant differences. Thus, All the separation models, including DNN, BNN and DBNN based models, have 257 input nodes and 257 output nodes. The target mask we use is the IRM, which leads to large speech intelligibility improvements and has been proved better than the Ideal Binary Mask (IBM) [20].

First, we obtain the teacher models by training a DNN with three hidden layers of 4096 hidden units. To get a "knowledgeable" teacher model to extract the spatiotemporal structure from the very large, highly redundant speech datasets, we exploit four tricks during training: first, we use dropout [34] for better performance because dropout can be viewed as a way of training an exponentially large ensemble of models that share weights; second, we also leverage extra data for training the teacher model to provide it with more information; thirdly, data shuffle is used to break through the specific sequence structure; finally we use Batch Normalization [35] to accelerating deep network training. Then we use the teacher model to get the soft mask as the distilled knowledge, which, we believe, has more information than the IRM because the IRM may be a better target on the train set while the soft mask inherits the generalization of the teacher model and is fit for good performance on the test set. Finally, we train (D)BNNs with three hidden layers of various hidden units to evaluate the validity of our method, as Fig. 4 shows. We keep the depth of BNNs same with the teacher model, then evaluate the validity of DBNN with four different hidden units. Just as previous works did, batch normalization layer is used before each binary nonlinear layer and ADAM [36] is used as the optimizer, and the procession of training DBNN with knowledge distillation is as Alg. 1. We adopt strict binary neural network, that is all of the weights and activations except activations in the output layer are binary. If the target is IBM, the output layer activations can be also binary. Actually, we first attempt to use the ideal binary mask as separation target, which is compatible with BNN. However, we find the output layer with binarize$(\cdot)$ function not guarantee a good performance, even when other layers are full-precision, which will be investigated in the future work.

For the hyper-parameters, unless otherwise specified, the initial learning rate is set to 10^{-3} and decay to 10^{-6} with a learning rate decay factor d_c determined by total training epochs. The parameter λ is set to 0.5 for the same contributions of the hard and soft targets and the parameter ℓ is usually set to 0.1, 0.01 or 0.001. Dropout is 0.1 lower than the dropout used in the corresponding DNNs because the BNNs can be regarded as a variant of dropout [12] and the batch size is set to 100.

In this paper, the BSS-EVAL metrics [37] is used to evaluate the performance of speech separation, which includes the Source-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR) and Source-to-Artifacts Ratio (SAR). The separation performance is mainly assessed by SDR because SDR can reflect the overall performance as mentioned in [5], [37].

B. Results and Analysis

We train DBNNs with label ensemble and loss ensemble, respectively, marked as DBNN-lb and DBNN-ls. As Fig. 4

Table I
COMPARISON OF DIFFERENT METHODS ON TIMIT CORPUS.

Model	SDR	SIR	SAR
NMF [6]	4.98	8.37	8.26
DNN+spectra [5]	6.16	8.17	7.91
RNN-best [5]	7.16	11.71	7.28
DRNN-diff [6]	6.71	11.94	8.49
DNN	7.25	12.22	7.48
BNN	7.20	11.94	7.52
DBNN	7.24	11.84	7.54

shows, the power of knowledge on the separation task is akin to previous works on classification tasks if BNN is regarded as a small network, and the knowledge transferred in the form of loss ensemble performs better, which keeps with the observation of [16]. But given that knowledge distillation and BNN are two methods of model compress, Fig. 4 shows important information that we can integrate already existing compression methods to further compress models while maintaining the performance (see DBNN-ls@512 and BNN@1024, respectively DBNN-ls with 512 units per hidden layer and BNN with 1024 hidden units) or improve the performance with a limited size (see the group of each size). Moreover, the standard deviation is reduced by integrating knowledge distillation into the training of BNN, which implies a novel approach to help guarantee the training success of BNN, as shown in Fig. 5. We find the power of knowledge distillation decline with the number of hidden units growing and we think it's due to the recovery of BNN's impaired learning ability as the size increases. When the size of network is small, the network parameters are not so superfluous and directly binarizing the network into BNN will weaken the learning capacity. Hence, DBNN with knowledge transferred from its teacher model will improve the separation performance and the training process of DBNN is more stable when network size is relative small. Table I shows that all DNN-based separation models perform better than the conventional NMF just as previous works demonstrated, and DNN+spectra and RNN-best are the results of DNN+spectra and RNN+logmel+joint+discrim with no context window and soft mask in the work [5]. Moreover, our DNN-based teacher model achieves the state-of-the-art performance due to the deeper and larger network and some efficient tricks. Then we successfully binarize the DNN-based model into BNN forms with a little loss and our DBNN performs better, which does not introduce additional computation at runtime.

V. CONCLUSIONS

In this paper, we successfully binarize the DNN-based separation models for monaural speech separation. What's more, we exploit knowledge distilling to train the DBNN for further improving the performance and stability on relatively small BNNs, which is the first attempt to integrate two types of model compression. Finally, we prove the power of our approaches on the TIMIT corpus.

Our future work on this topic will focus on training a more powerful teacher model, adapting our DBNNs for speaker-independent multi-talker speech separation by leveraging other methods such as PIT [9] or ASAM [23] and implementing the DBNNs on embedded devices by analyzing the computational complexity and memory usage.

ACKNOWLEDGEMENTS

We thank the reviewers for their insightful comments. This work was supported by the National Natural Science Foundation of China (61602479, 91720000), the Independent Deployment Project of CAS Center for Excellence in Brain Science and Intelligent Technology (CEBSIT2017-02) and Advance Research Program (6140452010101).

REFERENCES

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the acoustical society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[2] B. Arons, "A review of the cocktail party effect," *Journal of the American Voice I/O Society*, vol. 12, no. 7, pp. 35–50, 1992.

[3] D. Wang and J. Chen, "Supervised speech separation based on deep learning: an overview," *arXiv preprint arXiv:1708.07524*, 2017.

[4] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[5] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1562–1566.

[6] G.-X. Wang, C.-C. Hsu, and J.-T. Chien, "Discriminative deep recurrent neural networks for monaural speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2544–2548.

[7] J. Du, Y. Tu, Y. Xu, L. Dai, and C.-H. Lee, "Speech separation of a target speaker based on deep neural networks," in *Signal Processing (ICSP), 2014 12th International Conference on*. IEEE, 2014, pp. 473–477.

[8] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 31–35.

[9] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 241–245.

[10] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in neural information processing systems*, 2015, pp. 1135–1143.

[11] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[12] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in neural information processing systems*, 2016, pp. 4107–4115.

[13] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 525–542.

[14] I. Hubara, M. Courbariaux, D. Soudry, R. El Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *arXiv preprint arXiv:1609.07061*, 2016.

[15] W. Tang, G. Hua, and L. Wang, "How to train a compact binary neural network with high accuracy?" in *AAAI*, 2017, pp. 2625–2631.

[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[17] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[18] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Ninth International Conference on Spoken Language Processing*, 2006.

[19] M. Cooke, *Modelling auditory processing and organisation*. Cambridge University Press, 2005, vol. 7.

[20] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[21] Y. Wang, J. Du, L.-R. Dai, and C.-H. Lee, "A maximum likelihood approach to deep neural network based nonlinear spectral mapping for single-channel speech separation," *Proc. Interspeech 2017*, pp. 1178–1182, 2017.

[22] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 246–250.

[23] J. Xu, J. Shi, G. Liu, X. Chen, and B. Xu, "Modeling attention and memory for auditory selection in a cocktail party environment," *AAAI*, 2018.

[24] D. Wang, "Deep learning reinvents the hearing aid," *IEEE Spectrum*, vol. 54, no. 3, pp. 32–37, 2017.

[25] E. Ceolini and S.-C. Liu, "Impact of low-precision deep regression networks on single-channel source separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 256–260.

[26] S. K. Esser, P. A. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy, A. Andreopoulos, D. J. Berg, J. L. McKinstry, T. Melano, D. R. Barch *et al.*, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proceedings of the National Academy of Sciences*, p. 201604850, 2016.

[27] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "Finn: A framework for fast, scalable binarized neural network inference," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2017, pp. 65–74.

[28] A. Galloway, G. W. Taylor, and M. Moussa, "Attacking binarized neural networks," *arXiv preprint arXiv:1711.00449*, 2017.

[29] C. Bucilu, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 535–541.

[30] J. Yim, D. Joo, J.-H. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] Z. Wu, D. Lin, and X. Tang, "Adjustable bounded rectifiers: Towards deep binary representations," *arXiv:1511.06201*, 2015.

[32] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *International Conference on Machine Learning*, 2015, pp. 2285–2294.

[33] V. Zue, S. Seneff, and J. Glass, "Speech database development at mit: Timit and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351–356, 1990.

[34] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

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

[37] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.