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

Direct Neuron-wise Fusion of Cognate Neural Networks

Takashi Fukuda, Masayuki Suzuki, and Gakuto Kurata

IBM Research AI
Chuo-ku Hakozaki, Tokyo, 103-8510, Japan
{fukuda1, szuk, gakuto}@jp.ibm.com

Abstract

This paper proposes a method to create a robust acoustic model by directly fusing multiple neural networks that have dissimilar characteristics without any additional layers/nodes involving retraining procedures. The fused neural networks derive from a shared parent neural network and are referred to as cognate (child) neural networks in this paper. The neural networks are fused by interpolating weight and bias parameters associated with each neuron with a different fusion weight, assuming that cognate neural networks to be fused have the same topology. Therefore, no extra computational cost during decoding is required. The fusion weight is determined by considering a cosine similarity estimated from parameters connecting to the neuron and the fusion is performed for every neuron. Experiments were carried out using a test suite consisting of various acoustic conditions with a wide SNR range, speakers including foreign accented speakers, and speaking styles. From the experiments, the network created by fusing cognate neural networks showed consistent improvement on average compared with the commercial-grade domain-free network originating from the parent model. In addition, we demonstrate that the fusion considering input connections to the neuron achieves the highest accuracy in our experiments.

Index Terms: acoustic modeling, neural network, fusion, cosine similarity, generalization.

1. Introduction

Combining acoustic models with different characteristics to obtain synergistic effects from each model has been well researched in automatic speech recognition (ASR). An example of the combination of neural network based acoustic modeling for ASR includes joint training of multiple neural networks [1, 2, 3], posterior-level combination [4, 5], and hypothesis level combination [6, 7, 8, 9]. The joint training is a method to train a whole network connecting one middle layer in a neural network to a layer in another network. There are two ways of training a whole network: one is training the whole network from scratch, and the other is training each network and then combining the networks and retraining the combined network [10]. Joint training targeting several networks has also been investigated. The posterior-level combination, which is typically applied to an NN/HMM hybrid system, combines posterior probabilities (or logits) with a weighted interpolation obtained from multiple networks trained separately. The interpolation weight is often determined with a development dataset. On the other hand, an example of hypothesis level combination is a voting of output words from each ASR system to form a single word transition network, called ROVER combination [6, 11, 12]. The combinatorial usage of multiple networks described above is a promising way to seek the best performance for a whole system. However, these methods are more computationally expensive and need more resources than single-network systems during decoding time. In addition, approaches leveraging several neural networks that have complicated architecture need many trial-and-error experiments to converge the training for sufficient accuracies.

This paper proposes a method to incorporate synergistic effects from multiple neural networks into a single network by performing direct neuron-level fusion, which does not require any retraining procedures and extra computational cost during decoding. Our method targets an online ASR which requires low computational cost. The neural networks to be fused in our proposed method originate from a unique parent neural network. Because the acoustic property of the parent network is inherited by child networks to some extent, the child neural networks deriving from the same parent neural network are considered as cognate neural networks. We assume that fused neural networks have the same topology and neurons (weight and bias parameters) that exist in the same position are fused by the weighted interpolation. A fusion weight for each neuron is determined on the basis of a cosine similarity metric when connections to the neuron are regarded as a vector. Unless neurons from two cognate networks to be fused are excessively dissimilar, the fusion weight is automatically adjusted to make it large. If the characteristics of two neurons are dissimilar, smaller fusion weight is given to avoid irrelevant neurons being generated by the fusion. To the best of our knowledge, there has been little investigation in the literature into the direct fusion for weight and bias parameters of multiple neural networks without any retraining the network. Similar works to our method include a simple averaging of network parameters on intermediate models during SGD training [13, 14, 15]. However, neither the fusion of dissimilar networks nor the neuron-level fusion with different interpolation weight for each node has not been their focus.

If a neural network composed of a complicated architecture based on several neural networks is trained with a huge amount of training data (e.g. more than 10k hours), training takes a long time to complete because many trial-and-error and hyper parameter tunings are needed to elicit high performance. In some cases, even the convergence of the training itself is difficult. In contrast, the direct fusion of neural networks proposed in this paper not only improves an overall ASR performance by synergistic effects from multiple networks but also reduces the entire neural network development period because of its simpler process. Another advantage in our proposed method is that the fused neural network does not increase computational cost during decoding time because the architecture of the networks is not changed by the fusion at all.

2. Similarities of cognate neural networks

2.1. Similarities based on network initializations

This section addresses the importance of the fusion between neural networks in terms of a similarity metric, which is a key

http://dx.doi.org/10.21437/Interspeech.2019-1930

Table 1: *Cosine similarities between two neural networks. Conv, FC, BN, and OUT mean convolutional layer, fully connected layer, bottleneck layer, and output layer, respectively.*

Layer number	Type	Similarity	
		Init.=(*random*)	Init.=(*parent model*)
Layer=1	Conv	0.07	0.99
Layer=2	Conv	0.16	0.98
Layer=3	FC	0.02	0.94
Layer=4	FC	0.01	0.91
Layer=5	FC	0.01	0.91
Layer=6	FC	0.01	0.90
Layer=7	BN	0.07	0.94
Layer=8	OUT	0.02	0.97

factor in our proposed method. Table 1 shows an example of cosine similarities between every layer in two neural networks when weight and bias parameters connecting to each neuron in the networks are regarded as a vector representing a property of the neuron. This vector is referred to as a *neuron vector* in this paper. To obtain the similarities shown in Table 1, the neuron vector for each neuron is further concatenated to make a super-vector for each layer to compute the layer-wise similarity. In each column for "Similarity" in Table 1, two CNN-based neural networks that have the same topology were first trained with 10000-hour domain-free training data and 2000-hour domain-specific training data, respectively. The domain-free data consists of several public and private corpora covering wide varieties of speakers, speaking styles, and additive and convolutive noises for general-purpose ASR. In contrast, the domain-specific data was collected from a particular acoustic condition mainly focusing on noise robustness. An acoustic coverage greatly differs between domain-free and domain-specific data. See Section 4 for more details of the datasets and the network topology. Two neural networks related to each training dataset were further trained by using different initial parameters for the networks. The neural networks notated as "Init.=(random)" in Table 1 were trained by initializing weights and bias parameters with random values, and those notated as "Init.=(parent model)" were trained by initializing weights from the same parent neural network (Fig. 1). In this paper, the child neural networks originating from the same parent network are referred to as *cognate neural networks*.

As seen in Table 1, cosine similarities of the neural network pair trained with random initialization are much smaller than those of the cognate neural networks, meaning that the neural networks trained with random initialization are entirely dissimilar. On the other hand, the cognate neural networks trained with the initialization from the same parent model are surprisingly similar, and only fully connected (FC) layers have relatively large differences. Because the basic role of each neuron/filter in the cognate networks is carried over from the parent model, the additional training process originating from the parent model does not change the role of a neuron level much even if the training dataset is different, although an acoustic characteristics that can be handled as an entire neural network changes.

2.2. Order of filters

Here we focus on convolutive filters represented in the first convolutional layer after the cross entropy training of the network is performed. Many filters in the first convolutional layer are trained to capture N-th order spectral patterns in the time and frequency plane such as delta features, which are important for

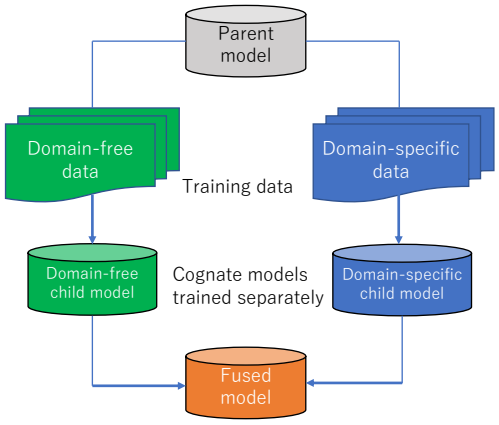

Figure 1: *Constructing flow of the fused model.*

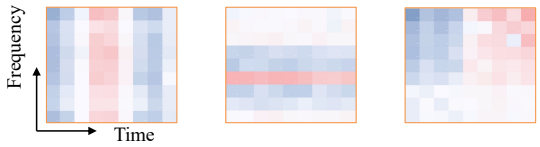

Figure 2: *Examples of* 9×9 *filters after cross entropy training in the first convolutional layer (best viewed in color). Red and blue represent positive and negative values, respectively.*

phone classification [10]. Figure 2 illustrates examples of 9×9 filters after training the network. In the figure, red and blue represent positive and negative values, respectively. As illustrated in the figure, the filters are regarded as derivative filters along with the time axis, frequency axis, and both axes. These basic time and frequency derivative filters are formed every training time even if initial weights of the network, training data, network topology including context dependent states related to output layer, and so on are different. Nevertheless, the reason the cosine similarities from the random initialization are small is that the filters from scratch are formed in different positions every training time, that is, the order of filters in each layer varies. As neurons that have greatly different roles in the position are fused, radical characteristics of neurons from each network spoil and the ambiguity of physical meaning in forward-propagated signals via activation functions increases. This results in poor recognition accuracy. This paper defines the closeness of the role of the neuron with a cosine similarity metric and proposes a fusion method that emphasizes the importance of the base neuron by lowering the fusion weight for the neuron pair in which the role between neurons greatly differs. The next section describes the details of the neuron-wise fusion of the networks.

3. Neuron-wise fusion

In Section 2, we discussed the importance of generating cognate neural networks and fusing them to obtain the synergy. This section describes the proposed neuron-wise fusion method and how the cognate neural networks are fused with a different fusion weight.

Let C_A and C_B be fused cognate neural networks. As discussed, these networks should be trained by initializing weight and bias parameters from the same parent neural network. To strengthen the specialty of each cognate network, the cognate

networks are trained with different training data, training procedures, etc. C_A is considered as a base network to be improved whereas C_B is regarded as a network to effectively expand acoustic characteristics of the base network C_A here. A neuron-wise fusion with the cosine similarities is performed by the following equation

$$\mathbf{W}_{lk} = (1 - \gamma_{lk})\mathbf{W}^A_{lk} + \gamma_{lk}\mathbf{W}^B_{lk}, \tag{1}$$

where $\mathbf{W}^A_{lk}$ and $\mathbf{W}^B_{lk}$ are the neuron vector of l-th layer and k-th neuron of C_A and C_B consisting of weights and bias parameters connecting as input to each neuron as

$$\mathbf{W}^A_{lk} = [w^A_{lk1}, w^A_{lk2}, \ldots, w^A_{lkn}, \ldots, w^A_{lkN}, b^A_{lk}]^T \tag{2}$$

$$\mathbf{W}^B_{lk} = [w^B_{lk1}, w^B_{lk2}, \ldots, w^B_{lkn}, \ldots, w^B_{lkN}, b^B_{lk}]^T. \tag{3}$$

w^A_{lkn}, w^B_{lkn}, b^A_{lk}, and b^B_{lk} are n-th weight and bias parameters of each neuron as shown in Fig 3, respectively. γ_{lk} is a fusion weight for each neuron estimated from cosine similarity D_{lk} as

$$\gamma_{lk} = \begin{cases} \alpha \dfrac{(D_{lk} - \beta)}{1.0 - \beta} & (D_{lk} > \beta) \\ 0 & otherwise \end{cases} \tag{4}$$

$$D_{lk} = \frac{\mathbf{W}^A_{lk} \cdot \mathbf{W}^B_{lk}}{|\mathbf{W}^A_{lk}||\mathbf{W}^B_{lk}|}. \tag{5}$$

The fusion weight γ_{lk} ranges $0 < \gamma_{lk} < \alpha$, where α is a parameter for controlling the maximum value of γ_{lk} and β indicates the minimum value (cutoff parameter) considered in the cosine similarity D_{lk} to determine the fusion weight. Instead of utilizing input connections, output connections from the neuron can be utilized as the neuron vector as

$$\mathbf{V}^A_{lk} = [v^A_{lk1}, v^A_{lk2}, \ldots, v^A_{lkm}, \ldots, v^A_{lkM}]^T \tag{6}$$

$$\mathbf{V}^B_{lk} = [v^B_{lk1}, v^B_{lk2}, \ldots, v^B_{lkm}, \ldots, v^B_{lkM}]^T, \tag{7}$$

where $\mathbf{V}^A_{lk}$ and $\mathbf{V}^B_{lk}$ are alternative neuron vectors composed of weight parameters v^A_{lkm} and v^B_{lkm} as output connections. The biases for the current neurons b^A_{lk} and b^B_{lk} can also be used for the similarity computation as a part of neuron vectors.

4. Experiments

4.1. Model Topology

All acoustic models used throughout this paper are CNNs of the same size. The CNN is trained with 40 dimensional log Mel-frequency spectra augmented with Δ and $\Delta\Delta$ as inputs with 16 kHz sampling frequency. Each frame of speech is also appended with a context of 11 frames after applying a speaker independent global mean and variance normalization. The CNN system uses 2 convolutional layers with 128 and 256 hidden nodes each in addition to 4 fully connected layers with 2048 nodes per layer to estimate posterior probabilities of 9300 context-dependent states as output targets. Before the output layer, the bottleneck layer with 512 nodes is also attached. All 128 nodes in the first feature extracting layer are attached with 9×9 filters that are 2 dimensionally convolved with the input log Mel-filterbank representations. The second feature extracting layer with 256 nodes has a similar set of 3×4 filters that processes the non-linear activations after max pooling from the preceding layer. The non-linear outputs from the second feature extracting layer are then passed onto the subsequent fully connected layers. All the layers use the ReLU non-linearity.

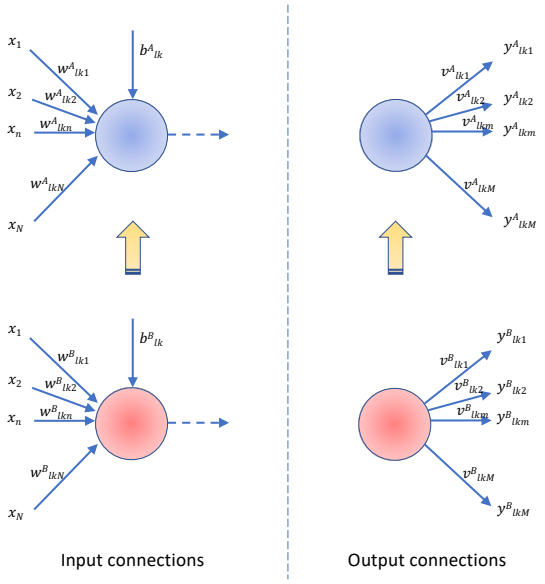

Figure 3: *Input and output connections of neuron.*

4.2. Data and model builds

The training data for the parent CNN model used in the experiments consists of 3600 hours of audio data. One-third of this training corpora is clean audio from several public corpora such as Broadcast News, Mixer 6 [16], and AMI corpus [17]. These corpora are further augmented with realistic environmental noises and impulse responses at various SNR range to total 3600 hours. Before the parent model is trained, this data was used to train the VGG model comprising 10 convolutional layers, with a max-pooling layer inserted after every 3 convolutional layers, followed by 4 fully connected layers [18]. Batch normalization is also applied to every fully connected layer in the VGG model. The VGG model has the same context dependent phones in the output layer as the parent CNN model. Then, the student CNN model was trained as the parent CNN model with a teacher-student training framework with the same 3600 hours of audio data after soft labels were generated from the VGG model [19].

After the parent CNN model construction, two child networks were created from the parent network as cognate networks in the experiments. One child network C_A was initialized by weight parameters from the parent network and trained with more than 10000 hours created by adding in-house farfield speech and foreign accented speech data to the 3600-hour dataset to enhance the robustness to the various acoustic conditions. This 10000-hour dataset is called a domain-free dataset because of its diversity of speakers, speaking styles, and acoustic conditions. In contrast, the other child network C_B was trained with 2000 hours of domain-specific data that is a subset of the above datasets. The child network C_B was initialized by the parent model, too. The training dataset for C_B focuses more on the noise robustness.

We explore the effectiveness of the proposed method using several well-known English corpora including ASpIRE [20] and Broadcast News as large vocabulary continuous speech recognition (LVCSR) tasks, and Aurora 4 which is primarily used to evaluate noise robust algorithms [21]. In addition, foreign accented English speech consisting of Asian (2.1 hours) and Latin American accented speakers (2.4 hours) was also used. The utterances in the foreign accented speech data set are a mix of dig-

Table 2: *Baseline performance for test sets.*

Test set	ChildNet C_A (Domain-free)	ChildNet C_B (Domain-specific)
ASpIRE	38.7	39.9
BN-Dev04f	15.4	14.5
Aurora-4	12.2	11.9
Accented	18.3	25.3

its and letters in insolation, command phrases, and short dialogs seen in spoken language systems [22]. Baseline performance from two child networks is tabulated in Table 2. The decoder uses a vocabulary comprising 250K words and the language model is a 4-gram LM with 200M *n*-grams. The child network C_A was trained with an acoustically well-balanced dataset, so the model works robust on average for every test set. In contrast, the child network C_B was constructed as noise-robust model, and it worked very well for BN-Dev04f and Aurora-4 tasks but had a large negative impact on accented data.

4.3. Results

The goal here is to improve the overall performance of the child network C_A by mixing the particular classification ability of C_B which is robust to noisy environments. A system combination is a popular approach to obtain a benefit from multiple networks, but the computational cost increases in proportion to the number of models. This is a critical problem in an online ASR system which requires low computation cost. Combining multiple networks with additional computational cost during decoding is not our target in this paper.

In this section, three methods of neural network fusions are first compared. Table 3 shows the experimental results. In the table, "Flat fusion" means using the same fusion weight for all layers and "Layer-wise fusion" indicates using different fusion weights for each layer. For the flat fusion, 0.35 was used as the fusion weight that showed the best accuracy. The layer-wise fusion weight was estimated from one similar to the weight of the neuron-wise method, but layer-wise fusion uses supervector concatenating neuron vectors whereas the neuron-wise method does not. The neuron-wise fusion is the method described in Section 3. For the neuron-wise and layer-wise fusion, α and β were set to 0.3 and 0.7, respectively. From Table 3, we can see consistent improvements in three sets (ASpIRE, BN-Dev04f, and Aurora-4) from all types of fusion methods. Because the child network C_B works poorly for accented speakers, the flat fusion greatly degrades performance compared with the general-purpose child network C_A although they are both robust to noisy conditions. In contrast, since layer-wise and neuron-wise fusions can control fusion weights on the basis of the cosine similarity between networks, a side-effect from the fusion was much reduced over the flat fusion technique, further improving the averaged WER compared with the domain-free network C_A.

Next we consider neuron-wise fusions performed by the different neuron vectors. As mentioned in Section 3, there are several choices to create the neuron vectors. One neuron-wise fusion focuses on input-side connection, and the other considers output-side connections. An additional candidate to test here is to use both input and output connections. Table 4 compares performances using three types of neuron vectors. All cases effectively transferred acoustic knowledge from C_B to the domain-free child network C_A and outperformed the strong base child network C_A by the fusion. Using the neuron vector composed of input connections achieved the best results in our experiments. In addition, we investigate how important the bias term is in the neuron vector. Table 5 compares the performance with and without the bias term for each neuron when the input-side connections are used as the neuron vectors. The results suggest that the bias term is useful when the performance gap between two cognate models is big, such as in accented data.

Table 3: *Comparisons between fusion methods.*

Test set	Flat fusion	Layer-wise	Neuron-wise
ASpIRE	38.3	38.5	38.4
BN-Dev04f	15.1	15.0	14.9
Aurora-4	11.9	12.0	11.9
Accented	19.3	18.9	18.5

Table 4: *Comparisons between input and output connections as neuron vector.*

Test set	Input-side	Output-side	Both-side
ASpIRE	38.4	38.5	38.5
BN-Dev04f	14.9	15.0	15.1
Aurora-4	11.9	12.0	12.0
Accented	18.5	18.8	18.7

Table 5: *Comparisons neuron vector with and without the bias term.*

Test set	w/ bias	w/o bias
ASpIRE	38.4	38.4
BN-Dev04f	14.9	15.1
Aurora-4	11.9	12.0
Accented	18.5	19.0

Table 6: *Neuron-wise fusion of three networks.*

Test set	Neuron-wise (C_A+C_B)	ChildNet C_C (Accented)	Neuron-wise (C_A+C_B+C_C)
ASpIRE	38.4	38.8	38.4
BN-Dev04f	14.9	15.6	14.9
Aurora-4	11.9	12.8	12.0
Accented	18.5	17.1	17.6

Lastly, we tried to fuse one more child network focusing on foreign accented speakers, where the best model shown in Table 3 did not perform well, originating from the parent network. The additional child network C_C was further fused after the fused network was first created from two child networks C_A and C_B. From Table 6, we can see improved performance on accented data with 3.8% relative improvement over the base domain-free child network C_A.

5. Conclusions

A neuron-wise fusion by adjusting interpolation weights on the basis of a cosine similarity metric significantly improves performance and reduces the side effect caused by a bad fusion. The fusion can be performed very well when the neural networks are trained by starting from the same parent model. We evaluated the neuron-wise fusion with four kinds of test sets consisting of various acoustic conditions, speakers including foreign accented speakers, and speaking styles. In this paper, experiments were conducted with the fusion between child networks, but the fusion between parent and child networks was considerable as well. In long-run service such as cloud-based ASR, it is general to periodically update an acoustic model, for example, by adding new training data. Our method would be well suitable for such scenario.

6. References

[1] H. Soltau, G. Saon, and T. N. Sainath, "Joint training of convolutional and non-convolutional neural networks," in *Proc. IEEE ICASSP*, 2014, pp. 5572–5576.

[2] Y. Qian, T. Tan, D. Yu, and Y. Zhang, "Integrated adaptation with multi-factor joint-learning for far-field speech recognition," in *Proc. IEEE ICASSP*, 2016, pp. 5770–5774.

[3] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *Proc. IEEE ICASSP*, 2013, pp. 8619–8623.

[4] K. Audhkhasi, A. M. Zavou, P. G. Georgiou, and S. S. Narayanan, "Theoretical analysis of diversity in an ensemble of automatic speech recognition systems," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 3, 2014, pp. 711–726.

[5] P. Swietojanski, A. Ghoshal, and A. Renals, "Revisiting hybrid and gmm-hmm system combination techniques," in *Proc. IEEE ICASSP*, 2013.

[6] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. IEEE ASRU*, 2014, pp. 347–354.

[7] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," in *Computer Speech and Language*, 2000, pp. 373–400.

[8] O. Siohan, B. Ramabhadran, and B. Kingsbury, "Constructing ensembles of ASR systems using randomized decision trees," in *Proc. IEEE ICASSP*, vol. 1, 2005, pp. 197–200.

[9] X. Zhang, D. Povey, and S. Khudanpur, "A diversity-penalizing ensemble training method for deep learning," in *Proc. Interspeech*, 2015.

[10] T. Fukuda, O. Ichikawa, G. Kurata, R. Tachibana, T. Samuel, B. Ramabhadran, and G. Saon, "Effective joint training of denoising feature space transforms and neural network based acoustic models," in *Proc. IEEE ICASSP*, 2017, pp. 5190–5194.

[11] H. Schwenk and J. Gauvain, "Combining multiple speech recognizers using voting and language model information," in *Proc. ICSLP*, pp. 915–918.

[12] Y. Huang, D. Yu, Y. Gong, and C. Liu, "Semi-supervised gmm and dnn acoustic model training with multi-system combination and confidence re-calibration," in *INTERSPEECH*, 2013.

[13] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *arXiv:1803.05407v2*, 2018.

[14] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson, "Improving consistency-based semi-supervised learning with weight averaging," in *arXiv:1806.05594v2*, 2018.

[15] K. Xu, H. Mi, D. Feng, H. Wang, C. Chen, Z. Zheng, and X. Lan, "Collaborative deep learning across multiple data centers," in *arXiv:1810.06877v1*, 2018.

[16] L. Brandchain, "The mixer 6 corpus: Resource for cross-channel and text independent speaker recognition," *LREC*, 2010.

[17] J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus," *Language Resources and Evaluation*, vol. 41, no. 1, pp. 181–190, 2007.

[18] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for lvcsr," in *Proc. Interspeech*, 2016.

[19] T. Fukuda, M. Suzuki, G. Kurata, T. Samuel, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers," in *Proc. Interspeech*, 2017, pp. 3697–3701.

[20] M. Harper, "The automatic speech recognition in reverberant environments (ASpIRE) challenge," *Proc. IEEE ASRU*, pp. 547–554, 2015.

[21] D. Pearce and J. Picone, "Aurora working group: DSR front end LVCSR evaluation au/384/02," *Inst. for Signal & Inform. Process., Mississippi State Univ., Tech. Rep*, 2002.

[22] T. Fukuda, R. Fernandez, A. Rosenberg, T. Samuel, B. Ramabhadran, S. A., and G. Kurata, "Data augmentation improves recognition of foreign accented speech," in *Proc. Interspeech*, 2018, pp. 2409–2413.