2016 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 13–16, 2016, SALERNO, ITALY

ARTICULATORY AND SPECTRUM FEATURES INTEGRATION USING GENERALIZED DISTILLATION FRAMEWORK

Jianguo Yu, Konstantin Markov

Human Interface Lab
The University of Aizu
Fukushima, Japan

Tomoko Matsui

Department of Statistical Modeling
The Institute of Statistical Mathematics
Tokyo, Japan

ABSTRACT

It has been shown that by combining the acoustic and articulatory information significant performance improvements in automatic speech recognition (ASR) task can be achieved. In practice, however, articulatory information is not available during recognition and the general approach is to estimate it from the acoustic signal. In this paper, we propose a different approach based on the generalized distillation framework, where acoustic-articulatory inversion is not necessary. We trained two DNN models: one called "teacher" learns from both acoustic and articulatory features and the other one called "student" is trained on acoustic features only, but its training process is guided by the "teacher" model and can reach a better performance that can't be obtained by regular training even without articulatory feature inputs during test time. The paper is organized as follows: Section 1 gives the introduction and briefly discusses some related works. Section 2 describes the distillation training process, Section 3 describes ASR system used in this paper. Section 4 presents the experiments and the paper is concluded by Section 5.

***Index Terms*—** speech recognition, articulatory features, XRMB, DNN-HMM acoustic model, generalized distillation, privileged information.

1. INTRODUCTION

Current state-of-the-art automatic speech recognition systems represent speech as a sequence of non-overlapping phonetic units while implicitly assuming that speech can be decomposed into a disjoint acoustic segment, which limits the acoustic models ability to properly learn the underlying variations in spontaneous or conversational speech. Although such systems perform fairly well for clearly articulated speech in "controlled" conditions, they suffer from acoustic variabilities in speech. Such variabilities can be due to background noises, speaker differences, differences in recording devices etc.

Many studies [1] [2] [3] [4] [5] have shown that articulatory information can improve the ASR performance and increase its robustness against noise contamination and speaker variation. It can also help model coarticulation in a more systematic way rather than using tri- or quin-phone acoustic models that necessitate a large training database to create all possible models of tri- or quin-phone units if some tri- or quin-phone units are less frequent. Unfortunately, articulatory information is not available during recognition. One direction of utilizing articulatory information is to generate articulatory features given the corresponding acoustic speech signal, known as acoustic-to-articulatory inversion [6]. Several such methods have been attempted, such as, Gaussian mixture model [7], feedforward neural networks [8], Bidirectional LSTMs [9] [10]. Other approaches use articulatory data at training time and attempt to embed the articulatory information inside the model and leave it hidden (i.e., implicitly predict it) at test time. For example, in [1] a hybrid HMM/BN model is adopted, and [11] uses Dynamic Bayesian Network to treat articulatory information as hidden variable. In [12], a MULTI-VIEW method based on canonical correlation analysis(CCA) is proposed, which finds pairs of maximally correlated linear projections of data in two the views.

The paradigm of *machines-teaching machines* has been investigated in studies of Vapnik et al. [13] [14] and Hinton et al. [15]. Motivated by the principles of human education, authors incorporate an "intelligent teacher" into machine learning. It is assumed that for each feature-label pair, there is an additional information about it provided by a teacher to support the learning process. However, teacher information will not be available at test time. This framework is also known as *learning using privileged information*. Such approach allows building a classifier which is better than those built on the regular features alone. On the other hand, Hinton proposed the concept of distilling the knowledge in neural networks [15], where a simple machine learns a complex task by imitating the solution of a more complicated and flexible machine. This can be applied in cases when a fast or real-time operation is required, but using the flexible machine is computationally prohibitive.

In a recent study [16], the learning using privileged in-

978-1-5090-0746-2/16/$31.00 ©2016 IEEE

formation and the distillation methods have been combined into a *Generalized Distillation* framework which utilizes the strengths of both methods. Here, the teacher who has access to the privileged information plays the role of a more complicated machine in the distillation process. After the simpler student is learned through the distillation process, it is used for testing when no privileged information is available. Generalized Distillation is closely related to applications in methods such as semi-supervised learning, domain adaptation, transfer learning, Universum learning [17] and curriculum learning [18].

In this paper, we apply the Generalized Distillation in the speech recognition task in order to integrate articulatory information into speech recognition system in a way that recognition uses the acoustic information only. As privileged information, we utilize both acoustic and articulatory data to learn the teacher machine. The student machine is learned on speech data only and during the training is guided by the teacher which has access to the corresponding articulatory measurements. Our ASR system is a DNN-HMM hybrid where DNN is used to predict HMM state probabilities. Such systems are popular since they allow to utilize the high performance of the DNN with the conventional and well-established decoding and language modeling methods. The HMM transition probabilities are obtained from a traditional GMM-HMM acoustic model and the DNN is learned using the above mentioned generalized distillation framework.

In our experiments, when trained on both the acoustic and articulatory features, the conventional GMM-HMM model achieves about 10% absolute phoneme error rate (PER) reduction with respect to the acoustic only model, which is similar to results from other papers [19] [20]. With the DNN-HMM acoustic model, however, in all cases, 2 to 3 times better PER were obtained.

2. GENERALIZED DISTILLATION

Generalized distillation has been termed in [16] to frame two techniques of Hinton's distillation [15] and Vapnik's privileged information [14] that enable machines to learn from other machines. In the framework, an "intelligent teacher" is incorporated into machine learning and the training data is formed by a collection of triplets

$$(x_1, x_1^*, y_1), \ldots, (x_n, x_n^*, y_n) \sim P^n(x, x^*, y),$$

where x_i, y_i is a feature-label pair and x_i^* is additional information about x_i, y_i provided by an intelligent teacher. The teacher is assumed to develop a language that effectively communicates information to help the student come up with better representation and to enable to learn characteristics about the decision boundary which are not contained in the student samples.

The process is as follows:

1. Learn teacher $f_t \in \mathcal{F}_t$ in eq. (1) using $\{(x_i^*, y_i)\}_{i=1}^n$.

   $$f_t = \arg\min_{f \in \mathcal{F}_t} \frac{1}{n} \sum_{i=1}^{n} l(y_i, \sigma(f(x_i^*))) + \Omega(||f||) \quad (1)$$

   Here, $x_i^* \in \mathcal{R}^d$, $y_i \in \Delta^c$, Δ^c is the set of c-dimensional probability vectors, F_t is a class of functions from $\mathcal{R}^d$ to $\mathcal{R}^c$, $\sigma : \mathcal{R}^c \to \Delta^c$ is a soft-max function, l is a loss function and Ω is an increasing function which serves as a regularizer.

2. Compute teacher soft labels $\{\sigma(f_t(x_i^*)/T\}_{i=1}^n$ using temperature parameter $T > 0$.

3. Learn student $f_s \in \mathcal{F}_s$ in eq. (2) using $\{(x_i, y_i)\}_{i=1}^n$, $\{(x_i, s_i)\}_{i=1}^n$ and imitation parameter $\lambda \in [0, 1]$.

   $$\begin{aligned} f_s &= \arg\min_{f \in \mathcal{F}_s} \frac{1}{n} \sum_{i=1}^{n} [(1-\lambda) l(y_i, \sigma(f(x_i))) \\ &+ T^2 \lambda l(s_i, \sigma(f(x_i)))] \end{aligned} \quad (2)$$

   $$s_i = \sigma(f_t(x_i)/T) \in \Delta^c \quad (3)$$

   Here, $\mathcal{F}_s$ is a function class simpler than $\mathcal{F}_t$.

In this paper, we utilize DNN to learn representation of both f_t and f_s.

3. SYSTEM DESCRIPTION

Our system is a hybrid DNN-HMM system, where DNN is used to predict HMM state posterior probabilities given an input data vector. These probabilities are converted to likelihoods using state priors and standard decoding is performed to obtain the recognition result.

We apply the Generalized Distillation framework for the DNN training only. Targets for the DNN learning are obtained by first training conventional GMM-HMM systems using both articulatory and acoustic features. Then, target states are identified by forced alignment.

Next, we learn the teacher DNN according to the first step of the procedure described in the previous section. The train data x_i^* are concatenated acoustic and articulatory vectors and the "hard" targets y_i are one-hot vectors where the component corresponding to the target state is 1 and all other components are set to 0. After training, the parameters of the teacher DNN are fixed.

The student DNN learning procedure is illustrated in Fig.1. Outputs of the teacher DNN are used as soft targets s_i and together with the hard targets y_i act as arguments of the student DNN loss function as in Eq.(2). In addition, teacher DNN outputs are smoothed with the temperature parameter T according to Eq.(3). The input training data for the student DNN are acoustic features only and are fed in batches. The corresponding concatenated acoustic and articulatory data,

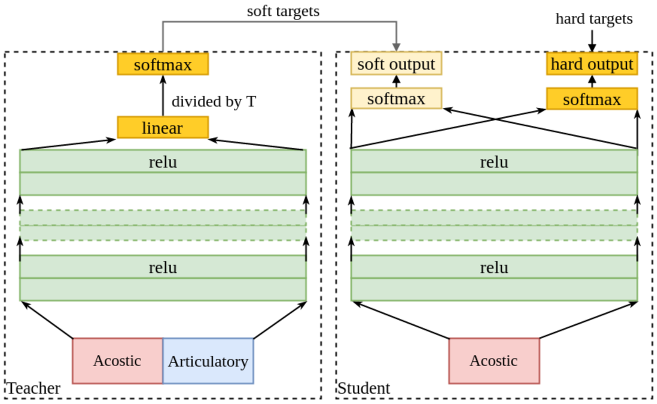

Fig. 1. *Student training block diagram.*

also in batches, are given to the teacher DNN input. However, only student DNN parameters are updated during this procedure.

During the test, only student DNN is used and the state probability predictions from the "hard" output, i.e. the output that was compared with the hard targets during training, are fed to the HMM decoder as shown in Fig.2. Unlike the articulatory inversion approach, during the test, DNN model trained using distillation approach doesn't need extra computational cost and can test as fast as the standard DNN-HMM system.

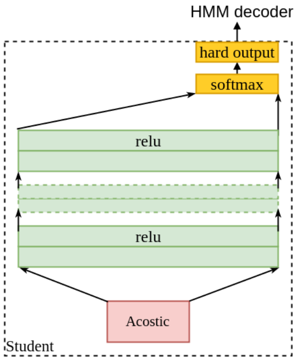

Fig. 2. *Testing with student DNN.*

4. EXPERIMENTS

We experimented with the University of Wisconsin X-ray microbeam database (XRMB) [21] which consists of simultaneously recorded acoustic and articulatory measurements from 47 American English speakers (22 males, 25 females). Each speaker's recordings comprise at most 118 tasks whose type can be number sequence, TIMIT sentences, isolated word sequence, paragraph as well as non-speech oral motor. Only normal speed sentences and number sequence tasks were used in our experiments. The articulatory measurements are horizontal and vertical displacements of 8 pellets on the tongue, lips, and jaw as shown in Fig.3 [21].

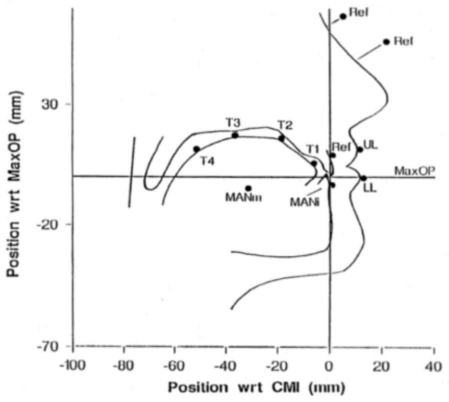

Fig. 3. *Placement of the 8 pellets on T1,T2,T3,T4,MANm, MANi,UL,LL points.*

We downsampled the acoustic signal from 21.74 kHz to 16 kHz, and our acoustic features are 13-dimensional Mel-frequency cepstral coefficients (MFCCs) computed every 10ms over a 25ms window, along with their first and second derivatives, resulting in 39-dimensional frames. We also downsampled the articulatory data from the original rate of 145.7Hz to 100Hz to match the frame rate of acoustic features and use the x,y coordinates of the 8 articulators along with their first and second derivatives as articulatory feature vectors of 48 dimensions. Including the first and second derivatives of the articulatory data is helpful since the movement itself can't tell apart speech pause from other phones. Finally, all feature vectors are mean and variance normalized on per utterance basis.

Due to limitations in the recording technologies, articulatory measurements contain missing data when individual pellets are mistracked. Though there are methods to reconstruct missing data [20], we decided to use only complete data. Phoneme alignment was done using the Penn Phonetics Lab Forced Aligner [22] and the missing entries, as well as data that are not consistent with orthographic transcripts, were cut off. Utterances are split into files, each containing only one sentence with silence parts of at the beginning and end reduced to 150ms. We split our data into several subsets as shown in Table.1.

We built two conventional GMM-HMM recognizers. One uses only acoustic features (39D), and the other one uses both MFCCs and articulatory features (87D). They are both standard 3-state left-to-right monophone HMM models. The phoneme language models is a simple bi-gram trained on data transcriptions including the paragraph task. We use 39 distinct phoneme and one silence HMMs. In total, there are 120 states. For this acoustic model, the best results were achieved with 38 Gaussian components per state. The recog-

Table 1. *Details of the datasets.*

	Train	**Test**	**Validation**	**Total**
Speakers	36	5	4	45
Female	19	3	2	24
Male	17	2	2	21
Utterances	3040	379	256	3675
Words	33883	4167	2758	40808
Phonemes	144863	17815	11580	174258
Hours	2:33:14	19:46	12:16	3:05:16

nition results (phone error rate%) are summarized in Table.2. With this GMM-HMM system, we generated frame level DNN training targets.

Table 2. *Phone error rates for conventional GMM-HMM system.*

LM weight/penalty	**MFCC**	**MFCC+ART**
0/0	29.95	12.01
7.0/2.0	18.85	9.67
7.0/1.0	18.73	9.72

4.1. Teacher DNN

Following some other studies [23] [24] and [25], we set the input window of 17 feature vectors, resulting in 663 or 1479 input nodes when using MFCC or MFCC+ART features. The output layer always has 120 nodes as the number of HMM states.

In contrast to some other approaches, we don't use layer-wise pre-training. Weights in each layer are uniformly initialized and the activation function is Rectifying Linear (ReLU). The output layer uses SoftMax activation. Since the DNN operates in classification mode, the standard objective is Categorical Cross-entropy. We also compared several optimization methods such as SGD+Nesterov Momentum(0.9) [26], rmsprop [27] as well as Adam [28], but found no significant differences in results. Adam and Rmsprop are faster, but the initial learning rate has to be smaller than SGD's. For the following experiments, we choose Adam optimization, where the learning rate starts from 1e-4 and is multiplied with 0.1 if validation data loss doesn't go down for 3 epochs. The entire training procedure is stopped when the learning rate is smaller than 1e-6 or the maximum of 100 iterations is reached. We varied four parameters to select optimal DNN structure, the number of hidden layers [3,**5**,6], the number of hidden nodes [1024,2048,3072], batch size [128,**256**,512] as well as drop out probability [0.1,0.2,0.3,**0.4**]. In total 144 DNN models were trained. We found that more layers and nodes increase the model's learning power, whereas smaller batch size and higher dropout probability tend to prevent the model from over-fitting. The best results we got are from the models that are well-balanced between this two trends.

First, we tested the teacher DNN in frame level state classification mode. The number of hidden layers did not have a big effect on the accuracy which was around 82%. We found, however, that the dropout influences the performance. Several training configurations and the corresponding frame level state classification accuracies, as well as the phoneme error rates for a DNN trained with 256 batch size, are shown in Table.3. The best PER result of **4.56**% was achieved with 5 hidden layers, 3072 nodes, and 40% dropout. Compared with the GMM-HMM system, this performance is 2 to 3 times better. Thus, we chose this DNN as our teacher model.

Table 3. *Training conditions and performance of the teacher DNN with 256 batch size*

Layers	**Nodes**	**Drop**	**PER%**	**Acc%**
3	2048	0.3	5.13	82.2
3	3072	0.3	5.19	82.3
3	2048	0.4	4.96	82.4
3	3072	0.4	4.98	82.6
4	2048	0.3	4.86	82.3
4	3072	0.3	5.01	82.4
4	2048	0.4	4.69	82.0
4	3072	0.4	4.71	82.5
5	2048	0.3	4.82	82.2
5	3072	0.3	5.03	82.1
5	2048	0.4	4.67	81.8
5	**3072**	**0.4**	**4.56**	**82.4**
6	2048	0.3	4.78	81.9
6	3072	0.3	4.61	82.3
6	2048	0.4	4.75	81.8
6	3072	0.4	4.90	81.5

4.2. Student DNN

The structure and training parameters of the student DNN are similar to the teacher DNN. Based on the results of teacher model, we set the batch size to 256. Both teacher and student PER were obtained using the MFCC+ART HMM model.

First, we learned the student DNN without distillation, i.e. without the help of the teacher DNN. As training data, we use only MFCC features. Table 4 shows the student DNN training conditions and performance for the different number of hidden layers and nodes.

Based on the results from this table, for the distillation training experiments, we choose student DNN with 4 hidden layers, 2048 hidden layer nodes, 40% dropout. The temperature parameter in Eq.(3) was varied from 1 to 5 and the imitation value was changed from 0 to 1 in steps of 0.2. Learning of the distilled student was performed as illustrated in Fig.1. The results in terms of PER are summarized in Fig.4. The

Table 4. *Training conditions and performance of the student DNN when learned alone, i.e. without distillation.*

Layers	Nodes	Drop	PER%	Acc%
3	2048	0.3	9.18	79.5
3	3072	0.3	9.11	79.4
3	2048	0.4	8.50	80.0
3	3072	0.4	8.68	79.9
4	2048	0.3	9.02	79.5
4	3072	0.3	8.98	79.5
4	**2048**	**0.4**	**8.18**	**79.7**
4	3072	0.4	8.46	79.7
5	2048	0.3	8.43	79.7
5	3072	0.3	8.92	79.2
5	2048	0.4	8.23	79.6
5	3072	0.4	8.34	79.4

blue dashed line shows the result of the student when trained alone, so it doesn't depend on the temperature or imitation parameters. The teacher result serves as the lower bound distilled student can achieve. As can be seen from the figure, for $T = 1$ and $\lambda = 0.6$, distillation result is 6.74% PER, which is 17.6% better than the result of the student alone. For the temperature $T = 2$, the performance is still better than the student alone, but not as good as temperature $T = 1$. Temperatures of 10 and higher, however, performed worse which can be explained by the smoothing effect they have on the teacher output.

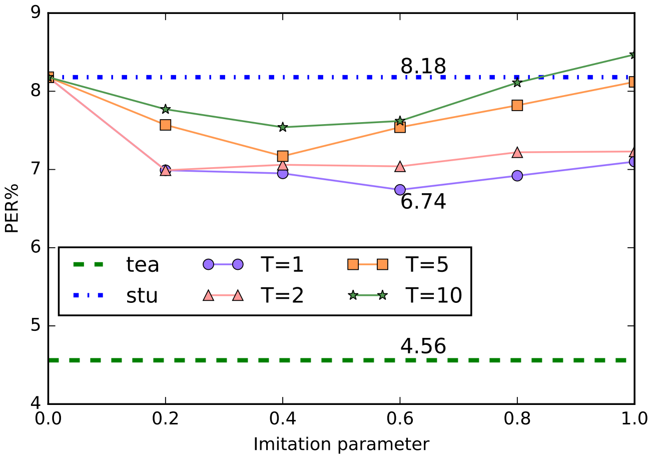

Fig. 4. *Results of distillation training*

5. CONCLUSIONS

In this work, we proposed an ASR system with integrated articulatory and acoustic features, where the acoustic model DNN is trained using the Generalized Distillation framework. It is an example of the *machines-teaching-machines* paradigm where machines learn from other machines. The teacher DNN trained on "rich" data and provides guidance to the student which learns from acoustic data only. Using DNN in the acoustic model provides big performance boost compared to the conventional GMM-HMM systems and we have confirmed this observation is our experiments as well. The student DNN trained without distillation achieves 8.18% average PER, while the GMM-HMM system's result is 18.73%. When we applied the distillation framework, additional 17.6% performance improvement was achieved leading to PER as low as 6.74%.

This is the first attempt to apply the generalized distillation framework for integration of articulatory and acoustic data for ASR. The results are encouraging, though we expected a higher gain in the performance. We believe there are other issues to be investigated within this framework including the effect of the teacher performance on training set, more sophisticated ways of "teaching", not just linear combination of loss functions, as well as utilizing other sophisticated DNN structures such as deep Long-Short Term Memory (LSTM) networks.

6. REFERENCES

[1] Konstantin Markov, Jianwu Dang, and Satoshi Nakamura, "Integration of articulatory and spectrum features based on the hybrid hmm/bn modeling framework," *Speech Communication*, vol. 48, no. 2, pp. 161–175, 2006.

[2] Karen Livescu, Omer Cetin, Mark Hasegawa-Johnson, Simon King, Christopher Bartels, Nash Borges, Amir Kantor, Pyare Lal, Lisa Yung, Ari Bezman, et al., "Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 jhu summer workshop," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 4, pp. IV–621.

[3] Katrin Kirchhoff, Gernot A Fink, and Gerhard Sagerer, "Combining acoustic and articulatory feature information for robust speech recognition," *Speech Communication*, vol. 37, no. 3, pp. 303–319, 2002.

[4] Joe Frankel and Simon King, "Asr-articulatory speech recognition," in *Eurospeech*, 2001.

[5] Joe Frankel, Korin Richmond, Simon King, and Paul Taylor, "An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces," in *The Proceedings of the 6˜(th) International Conference on Spoken Language Processing (Volume)*, 2000.

[6] Sankaran Panchapagesan and Abeer Alwan, “A study of acoustic-to-articulatory inversion of speech by analysis-by-synthesis using chain matrices and the maeda articulatory model,” *The Journal of the Acoustical Society of America*, vol. 129, no. 4, pp. 2144–2162, 2011.

[7] Tomoki Toda, Alan W Black, and Keiichi Tokuda, “Acoustic-to-articulatory inversion mapping with gaussian mixture model.,” in *INTERSPEECH*, 2004.

[8] Benigno Uria, Iain Murray, Steve Renals, and Korin Richmond, “Deep architectures for articulatory inversion.,” in *INTERSPEECH*, 2012, pp. 867–870.

[9] Peng Liu, Quanjie Yu, Zhiyong Wu, Shiyin Kang, Helen Meng, and Lianhong Cai, “A deep recurrent approach for acoustic-to-articulatory inversion,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4450–4454.

[10] Pengcheng Zhu, Lei Xie, and Yunlin Chen, “Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[11] Vikramjit Mitra, Hosung Nam, and Carol Y Espy-Wilson, “Robust speech recognition using articulatory gestures in a dynamic bayesian network framework,” in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 131–136.

[12] Rajkumar Arora and Karen Livescu, “Multi-view cca-based acoustic features for phonetic recognition across speakers and domains,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7135–7139.

[13] Vladimir Vapnik and Akshay Vashist, “A new learning paradigm: Learning using privileged information,” *Neural Networks*, vol. 22, no. 5, pp. 544–557, 2009.

[14] Vladimir Vapnik and Rauf Izmailov, “Learning using privileged information: Similarity control and knowledge transfer,” *Journal of Machine Learning Research*, vol. 16, pp. 2023–2049, 2015.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.

[16] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik, “Unifying distillation and privileged information,” in *ICLR*, 2016.

[17] Jason Weston, Ronan Collobert, Fabian Sinz, Léon Bottou, and Vladimir Vapnik, “Inference with the universum,” in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 1009–1016.

[18] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[19] Leonardo Badino, Claudia Canevari, Luciano Fadiga, and Giorgio Metta, “Integrating articulatory data in deep neural network-based acoustic modeling,” *Computer Speech & Language*, vol. 36, pp. 173–195, 2016.

[20] Weiran Wang, Raman Arora, and Karen Livescu, “Reconstruction of articulatory measurements with smoothed low-rank matrix completion,” in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 54–59.

[21] J Westbury, “X-ray microbeam speech production database user’s handbook. 1994,” *Waisman Center, University of Wisconsin: Madison, USA*, pp. 1–100, 1994.

[22] Jiahong Yuan and Mark Liberman, “Speaker identification on the scotus corpus,” *Journal of the Acoustical Society of America*, vol. 123, no. 5, pp. 3878, 2008.

[23] Bo Li and Khe Chai Sim, “Modeling long temporal contexts for robust dnn-based speech recognition,” in *INTERSPEECH*, 2014, pp. 353–357.

[24] Jessica Ray, Brian Thompson, and Wade Shen, “Comparing a high and low-level deep neural network implementation for automatic speech recognition,” in *High Performance Technical Computing in Dynamic Languages (HPTCDL), 2014 First Workshop for*. IEEE, 2014, pp. 41–46.

[25] Akihiro Abe, Kazumasa Yamamoto, and Seiichi Nakagawa, “Robust speech recognition using dnn-hmm acoustic model combining noise-aware training with spectral subtraction,” in *INTERSPEECH*, 2015.

[26] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, “On the importance of initialization and momentum in deep learning,” in *Proceedings of the 30th international conference on machine learning (ICML-13)*, 2013, pp. 1139–1147.

[27] Tijmen Tieleman and Geoffrey Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural Networks for Machine Learning*, vol. 4, pp. 2, 2012.

[28] Diederik Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.