*INTERSPEECH 2017*
August 20–24, 2017, Stockholm, Sweden

# A Speaker Adaptive DNN Training Approach for Speaker-independent Acoustic Inversion

*Leonardo Badino*[1], *Luca Franceschi*[1,2], *Raman Arora*[3], *Michele Donini*[1], *Massimiliano Pontil*[1,2]

[1] Istituto Italiano di Tecnologia, Italy
[2] University College London, UK
[3] Johns Hopkins University, MD,USA

{leonardo.badino,luca.franceschi,michele.donini,massimiliano.pontil}@iit.it

## Abstract

We address the speaker-independent acoustic inversion (AI) problem, also referred to as acoustic-to-articulatory mapping. The scarce availability of multi-speaker articulatory data makes it difficult to learn a mapping which generalizes from a limited number of training speakers and reliably reconstructs the articulatory movements of unseen speakers. In this paper, we propose a Multi-task Learning (MTL)-based approach that explicitly separates the modeling of each training speaker AI peculiarities from the modeling of AI characteristics that are shared by all speakers. Our approach stems from the well known Regularized MTL approach and extends it to feed-forward deep neural networks (DNNs). Given multiple training speakers, we learn for each an acoustic-to-articulatory mapping represented by a DNN. Then, through an iterative procedure, we search for a canonical speaker-independent DNN that is "similar" to all speaker-dependent DNNs. The degree of similarity is controlled by a regularization parameter. We report experiments on the University of Wisconsin X-ray Microbeam Database under different training/testing experimental settings. The results obtained indicate that our MTL-trained canonical DNN largely outperforms a standardly trained (i.e., single task learning-based) speaker independent DNN.

**Index Terms**: acoustic inversion, acoustic-to-articulatory mapping, multi-task learning, XRMB

## 1. Introduction

Measured vocal tract movements, i.e., articulatory features (AFs), can be beneficial for several speech technology applications, including speech synthesis [1], automatic speech recognition (ASR) [2, 3], pronunciation training [4] and speech-driven computer animation [5]. Techniques for measuring AFs range from electromagnetic articulography (EMA) to ultrasound and functional magnetic resonance imaging (fMRI).

Typically, AFs are much more difficult to collect than audio, require extensive preprocessing steps to reduce noise and interpolate missing data, and, in real-usage scenarios, are often only available at training time. For most of the aforementioned applications (e.g., articulatory ASR [3]), an acoustic inversion (AI) mapping is necessary to recover AFs from the acoustic signal.

In this paper we address speaker-independent AI and propose a speaker adaptive training approach for deep neural network (DNN)-based AI.

Given the scarce availability of multi-speaker articulatory datasets and the limited number of speakers per dataset, previous work has mainly focused on speaker-dependent AI, e.g., [6, 7, 8, 9].

Studies on speaker-dependent AI have proposed methods to appropriately address the non-uniqueness of AI [6, 10, 8]. Non-uniqueness means that identical sounds can be produced by posing the articulators in a range of different positions [11]. As a consequence the conditional probability density function of the position of an articulator given a speech sound can exhibit more than one mode [12]. In other words, AI can be a one-to-many mapping. However, Qin and Carreira-Perpiñán [13] showed that, although the non-uniqueness of AI occurs in human speech, most of the time the vocal tract has a unique configuration when producing a given phone. Non-linearity seems to be a more relevant aspect to address. Indeed, DNNs, which cannot straightforwardly implement one-to-many mappings but are universal approximators of non-linear functions, are often successfully employed to address AI [7, 14].

Most of the existing studies on speaker-independent AI actually consider a cross-speaker setting [4, 15], where simultaneous recordings of acoustic and articulatory data are available for one speaker only (*training* speaker). A speaker-dependent AI is first learned for the *training* speaker. Subsequently, AI for a new speaker (the *target* speaker) is learned by first mapping, e.g., through voice conversion techniques [4], the acoustic space of the *target* speaker into that of the *training* speaker and then applying the speaker-dependent AI learned on the *training* speaker. This approach does not actually recover the AFs of the *target* speaker but that of the *training* speaker. However, in [15] the subset of the recovered AFs can still well correlate with the *target* speaker actual AFs.

Speaker-independent AI learning from multi-speaker articulatory data is carried out in [2] where a DNN is trained and evaluated on data from the Wisconsin X-ray Microbeam Database [16] consisting of 47 speakers. A strong tendency to over-fitting after few training epochs and a relative poor generalization performance are reported. Poor generalization performance is intrinsic to a DNN training based on a pool of data from few speakers. When the number of training speakers is small, the speaker-independent DNN learning can be largely biased by strong speaker-specific AI characteristics.

In this paper, we propose an approach that prevents the speaker-independent DNN, trained on the concatenation of all speakers' data, from deviating from an average model, and thus, from modeling strong speaker peculiarities. We address the AI problem from a multi-task learning (MTL) perspective, where the problem of learning a speaker-dependent AI map for each training speaker is considered as a specific task. Through an iterative procedure, we search for a canonical speaker-independent DNN that is "similar" to all speaker-dependent DNNs and, at the same time, for speaker-dependent DNNs that are similar to the canonical DNN. The degree of

http://dx.doi.org/10.21437/Interspeech.2017-804

similarity is controlled by a regularization hyper-parameter. The proposed approach stems from the well known regularized MTL approach [17, 18] and extends it to DNNs. Its rationale is close in spirit to the Speaker Adaptive Training method proposed for Gaussian Mixture Model-Hidden Markov Model-based ASR, which performs speaker-adaptation of a canonical acoustic model during training [19] .

We evaluate our MTL strategy on the XRMB dataset. We experiment different training settings to assess the benefits of our approach over different numbers of speakers per training set and of data per training speaker.

# 2. Method

In this section, we discuss two methods to train a speaker independent AI function, using data from multiple speakers.

We let $S$ be the number of training speakers. For each $s \in \{1, \dots, S\}$ we consider a speaker-dependent AI function

$$f_s : X \to Y$$

that maps acoustic features (elements of $X$, e.g., MFCCs) into AFs (elements of $Y$). For each speaker $s$ we have a training set $\mathcal{D}_s = \{(\mathbf{x}_s^1, \mathbf{y}_s^1), (\mathbf{x}_s^2, \mathbf{y}_s^2), \dots, (\mathbf{x}_s^{n_s}, \mathbf{y}_s^{n_s})\} \subset X \times Y$, where $n_s$ is the number of training frames of speaker $s$. We also consider the canonical speaker-independent AI function $f_0$, which is trained from the multiple-speaker dataset $\mathcal{D} = \cup_{s=1}^{S} \mathcal{D}_s$. Each function is implemented by a DNN with learning parameter vector $\mathbf{w}$. In its more general formulation the proposed approach aims at finding the learning parameters $\mathbf{w}_1, \dots, \mathbf{w}_S$ (for each speaker-dependent DNN) and $\mathbf{w}_0$ (for the canonical DNN) that minimizes the objective function

$$\sum_{s=1}^{S} E_s(\mathbf{w}_s) + E(\mathbf{w}_0) + R(\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_S) \qquad (1)$$

where, for every $s \in \{1, \dots, S\}$, we defined

$$E_s(\mathbf{w}_s) = \sum_{i=1}^{n_s} \|f_s(\mathbf{x}_s^i, \mathbf{w}_s) - \mathbf{y}_s^i\|_2^2,$$

$$E_0(\mathbf{w}_0) = \sum_{s=1}^{S} \sum_{i=1}^{n_s} \|f_0(\mathbf{x}_s^i, \mathbf{w}_s) - \mathbf{y}_s^i\|_2^2,$$

and $R(\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_S)$ is a multi-task regularization term which leverages commonalities between the different speakers.

Notice that the objective function (1) contains both the regularized loss of the speaker-dependent DNNs and that of the canonical DNN

$$\sum_{s=1}^{S} \sum_{i=1}^{n_s} \|f_0(\mathbf{x}_{i,s}, \mathbf{w}_0) - \mathbf{y}_{i,s}\|_2^2 + \lambda_2 \|\mathbf{w}_0\|_2^2. \qquad (2)$$

In the rest of the paper we will refer to the DNN trained by minimizing the objective (2) as the single task learning (STL) DNN. In our approach below this DNN is used to "pretrain" the canonical DNN in the MTL framework.

## 2.1. Network Weight Regularization

The first approach is identical to that proposed in [17, 18] and is based on the variance regularizer

$$R(\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_S) = \frac{\lambda_1}{S} \sum_{s=1}^{S} \|\mathbf{w}_s - \mathbf{w}_0\|_2^2 + \lambda_2 \|\mathbf{w}_0\|_2^2 \quad (3)$$

where $\lambda_1$ and $\lambda_2$ are hyper-parameters that controls the similarity between all the $\mathbf{w}_s$ and $\mathbf{w}_0$ and the size of the vector $\mathbf{w}_0$ respectively. We refer to this first strategy as W-MTL, to emphasize the fact the weights of the networks are being regularized. That is, the difference between the speaker-dependent DNNs and the canonical DNN is represented by the differences between their learning parameters. Note that, because of the weight-space symmetries of neural networks (see, e.g., [20]), many different choices of weight vectors can generate the same AI function, so defining the similarity between two DNNs in terms of the similarity of their learning parameters may be misleading. To circumvent this problem, the multi-task regularizer (3) is only applied to the parameters of the last layer of the DNNs.

W-MTL approach is based on the following procedure:

1. Learn the weight vectors $\mathbf{w}_0$ by running stochastic gradient descent on the objective function (2).
2. With $\mathbf{w}_0$ fixed, minimize the objective function (1) w.r.t. all the weights $\mathbf{w}_s$, using standard back-prop for a defined number of training epochs.
3. With all $\mathbf{w}_s$ fixed, update $\mathbf{w}_0$ by the formula

$$\mathbf{w}_0 = \frac{\lambda_1}{\lambda_1 + \lambda_2} \overline{\mathbf{w}} \qquad (4)$$

   where $\overline{\mathbf{w}}$ is the average of the vectors $\mathbf{w}_1, \dots, \mathbf{w}_S$.
4. Repeat from Step 2 until an early-stopping criterion is met.

Note that Step 3 consists in minimizing the regularizer (3) over $\mathbf{w}_0$ while keeping $\mathbf{w}_s$ fixed. We choose this step as it is computationally more efficient than the minimization of the full objective function (Eq. 1), which also involves the error term $E(\mathbf{w_0})$ for the speaker-independent DNN.

## 2.2. Network Output Regularization

The second approach, named O-MTL, solves the optimization problem (1) with the regularizer

$$\begin{aligned} R(\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_S) = \lambda_1 \|f_s(\mathbf{x}_s^i, \mathbf{w}_s) - f_0(\mathbf{x}_s^i, \mathbf{w}_0)\|_2^2 \\ + \lambda_2 \|\mathbf{w}_0\|_2^2 + \lambda_3 \|\mathbf{w}_s\|_2^2. \qquad (5) \end{aligned}$$

Here, the speaker-dependent DNNs and the canonical DNN are compared in terms of their outputs. Notice also that in this case weight-space symmetries are not relevant.

The O-MTL approach is given by the following procedure:

1. Learn the weight vectors $\mathbf{w}_0$ by running stochastic gradient descent on the objective function (2).
2. Initialize each $\mathbf{w}_s$ by setting $\mathbf{w}_s = \mathbf{w}_0$.
3. Update the weights $\mathbf{w}_s$ by running stochastic gradient descent on the objective function (1) for a defined number of training epochs.
4. With all $\mathbf{w}_s$ fixed, update $\mathbf{w}_0$ by approximately minimizing objective (1) by running standard back-prop for one training epoch.
5. Repeat from Step 3 until an early-stopping criterion is met.

Finally, note that, unlike for W-MTL, the architectures of the speaker-dependent DNNs used in O-MTL can be different from that of the canonical DNN. For example they may be chosen to be significantly smaller. In this case, it is possible, in principle, to use a model compression technique [21] at Step 2 of the training procedure.

# 3. Experimental Setup

### 3.1. Dataset

All experiments were carried out on data from the XRMB corpus [22], consisting of simultaneous recordings of audio and articulatory movements of American English speakers. The dataset and the extracted acoustic and articulatory features are those reported in [2] with the only difference being one less testing speaker (JW58). JW58 was removed as we observed outlier AI results of the STL DNN (Eq. 2) on a subset of JW58's utterances.

Each speaker's recording consists of approximately 20 minutes of read speech divided into a maximum of 53 utterances. The 46-speaker dataset is divided into disjoint subsets of 35/8/3 speakers for AI training/validation/testing respectively as in [2].

The acoustic features are the first 13 MFCCs, plus deltas and delta-deltas, computed every 10ms from 25ms Hamming windows. AFs were obtained after down-sampling to 100Hz (equal to the acoustic frame rate) and missing data recovering [22]. AFs refer to the x-y positions in the sagittal plane of 8 pellets on tongue, lips and jaw.

### 3.2. DNNs, STL- and MTL-based training

Speaker-dependent and canonical DNNs were all 3-hidden layer feed-forward neural networks with 300 units each. As reported in previous work [2, 8] we did not observe significant improvements when increasing the number of nodes or hidden layers of the STL DNN.

The input to the DNNs consisted of 5 concatenated MFCC vectors (a context size used in all our previous work [3]) used to estimate a vector of 16 AFs. The 39 MFCCs were previously normalized to have 0 mean and 1 standard deviation. The 16 AFs were first per-speaker normalized and then normalized after pooling all training speakers.

All DNNs were implemented and trained using TensorFlow [23]. Optimizer and hyper-parameters shared by the canonical and the speaker-dependent DNNs were tuned according to the STL-trained canonical DNN performance on the validation set. After random search we found that stochastic gradient descent with 0.01 learning rate and 0.9 momentum (Tensorflow's MomentumOptimizer) and $\lambda_2 = 0.001$ gave the best results. The parameter $\lambda_3$ is assumed to be equal to $\lambda_2$ if not stated differently. The batch size was fixed to 200 training examples. The SLT-trained DNNs (also used to "pre-train" the MTL-trained canonical DNNs, see Step 1 of W-MTL and O-MTL procedures) were trained for 25 epochs. In both MTL strategies, speaker-dependent DNNs were trained for 3 epochs per each MTL iteration. We did not experiment with different numbers of epochs. The maximum number of MTL iterations were fixed to 20 but could be early stopped if no improvement in the AFs reconstruction error, measured as root mean squared error (RMSE) of the canonical DNN, was observed in the validation set over 3 consecutive iterations. In O-MTL, we set $\lambda_2 = 0$ after Step 1. At each O-MTL iteration, the $\mathbf{w_0}$ parameters of the canonical DNN were update for one epoch.

# 4. Results

Figures 1 and 2 show the reconstruction RMSE for the normalized AFs (i.e., Normalized RMSE in both figures, shortened to RMSE in the text) of the W-MTL and O-MTL strategies respectively on the validation set for some $\lambda_1$ values. As a reminder, $\lambda_1$ is the regularization hyperparameter that controls the similarity between the canonical DNN and the speaker-dependent DNNs (cf. Eqs. (3) and (5) for W-MTL and O-MTL, respectively). During STL training, the validation RMSE does not significantly decrease and stabilizes after approximately 15 epochs.

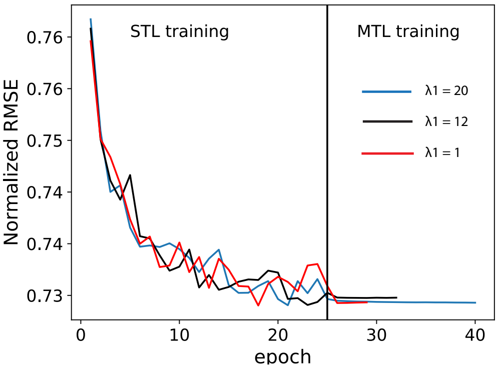

Figure 1: *Normalized Root Mean Square Error on the validation set of W-MTL at different $\lambda_1$ values. When STL stops and W-MTL starts, epochs are actually iterations of the last three steps in the procedure described in section 2.1.*

When W-MTL follows STL, it usually produces a slight RMSE reduction for a couple of iterations and then keeps reducing the error, but reductions are tiny (and barely visible in the figure).

On the other hand, O-MTL significantly reduces the validation RMSE over a few iterations. The error reductions over the validation and test set are largely correlated, as it is shown in the example of Figure 3 for $\lambda_1 = 12$. Given the much better performance of O-MTL w.r.t. W-MTL, all the next results will refer to the O-MTL method only.

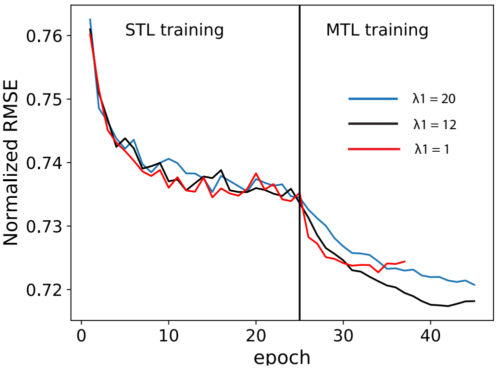

Figure 2: *Normalized Root Mean Square Error on the validation set of O-MTL at different $\lambda_1$ values. When STL stops and O-MTL starts, epochs are actually iterations of the last three steps in the procedure described in section 2.2.*

Figure 4 shows the dependency of the validation and test RMSE on $\lambda_1$. As expected, this dependency tends to be U-shaped. Very small and very large $\lambda_1$ values make O-MTL very close to STL. When $\lambda_1$ is very small the training of the canonical DNN (Step 4) is barely affected by the multi-task regulariza-

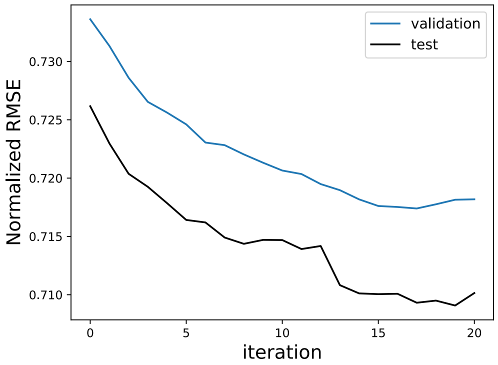

Figure 3: *Normalized RMSE of O-MTL on validation and test data with $\lambda_1 = 12$. Iteration 0 corresponds to the last STL training epoch.*

tion term $\lambda_1 \| f_s(\mathbf{x}, \mathbf{w}_s) - f_0(\mathbf{x}, \mathbf{w}_0) \|_2^2$, which becomes close to 0. When $\lambda_1$ is large the multi-task regularization term forces all $f_s(\mathbf{x}, \mathbf{w}_s)$ to be almost identical to $f_0(\mathbf{x}, \mathbf{w}_0)$, thus the multi-task regularization term is again very small.

Finally, we studied the benefits of O-MTL over two new training conditions: (i) reduced number of training speakers (first 13 speakers); (ii) reduced number of utterances (10) per training speaker. We kept the same validation and test sets of the full XRMB. Results of the best O-MTL systems (determined by the $\lambda_1$ value that produced the lowest RMSE on the validation set) are shown in Table 1 along with results on the full XRMB training set. In the second condition, we increased $\lambda_3$ to 0.005, as the DNNs where trained on $\approx 80\%$ smaller datasets.

O-MTL improves results in both the new training settings. However, the relative improvement over STL is smaller than that produced on the full XRMB training set. That may due to the fact that we did not thoroughly tune optimizer hyperparameters and $L^2$ weight penalties ($\lambda_2$ and $\lambda_3$) on these two new conditions.

Results in terms of $r$ Pearson's correlation and "real" (i.e., not normalized) RMSE reported in Table 1 show that the performance increase due to MTL (i) allows an MTL DNN trained on less than half the training speakers to almost match the performance of a STL DNN trained on the full training set ($r = 0.679$ vs. $r = 0.682$ on the test set); (ii) is comparable to AF reconstruction improvements over STL DNNs produced by recently proposed alternative machine learning techniques for speaker-dependent AI, e.g., trajectory mixture density networks [8] and general regression neural networks [14]. Note that our SLT baseline performs worse than the SLT DNN reported in [2] in terms of validation RMSE (2.00mm vs 1.96mm), but it uses a smaller context (5 vs. 7 frame window) and a different output.

Future work will include automatic MTL hyperparameter optimization [24], study of the effect of O-MTL on critical vs. non-critical AFs [25], application of O-MTL to ASR with hundreds of training speakers, and use of O-MTL-based speaker-independent AI for "articulatory" ASR [2, 3].

## 5. Conclusions

We have proposed a regularized multi-task learning approach for deep neural networks to learn a speaker-independent acoustic inversion. Given multiple training speakers we considered each speaker-dependent AI as a separate task and trained a DNN for each task. Then, we explicitly searched for a canonical speaker-independent DNN that shares commonalities with each speaker-dependent DNN. Results on the XRMB dataset and different training settings shows significant AI improvements over standard DNN training.

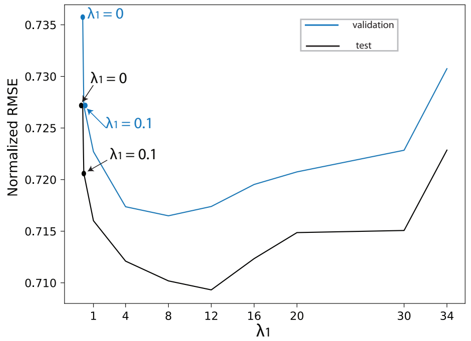

Figure 4: *O-MTL Normalized RMSE on the on validation and test set vs. $\lambda_1$. $\lambda_1 = 0$ correspond to STL training. The RMSE values reported correspond to the O-MTL iteration where the normalized RMSE was lowest on the validation set. The optimization procedure diverged for $\lambda_1 > 34$, most probably because of numerical computation issues.*

## 6. Acknowledgements

We thank Raman Arora and Karen Livescu for providing a preprocessed version of XRMB with reconstructed missing articulatory entries. The first author acknowledges the support of the European Union's Horizon2020 project ECOMODE (grant agreement No 644096).

Table 1: *STL (baseline) and O-MTL results over 3 different training sets: (i) the full XRMB training set, (ii) a reduced XRMB training set with 10 utterances per training speaker, and (iii) a reduced XRMB training set with 13 speakers (with no reduced number of utterances). $r$ refers to the Pearson's correlation coefficient. In this table RMSE is computed on not normalized AFs and is in millimeters*

| Full XRMB Training set | | | | | |
|---|---|---|---|---|---|
| | Validation | | Test | | |
| Approach | $r$ | $RMSE$ | $r$ | $RMSE$ | $\lambda_1$ |
| STL | 0.665 | 2.00 | 0.682 | 2.19 | |
| O-MTL | 0.684 | 1.93 | 0.700 | 2.14 | 8 |

| 10-utterance x speaker Training set | | | | | |
|---|---|---|---|---|---|
| | Validation | | Test | | |
| Approach | $r$ | $RMSE$ | $r$ | $RMSE$ | $\lambda_1$ |
| STL | 0.637 | 2.08 | 0.658 | 2.29 | |
| O-MTL | 0.646 | 2.05 | 0.667 | 2.26 | 12 |

| 13-speaker Training set | | | | | |
|---|---|---|---|---|---|
| | Validation | | Test | | |
| Approach | $r$ | $RMSE$ | $r$ | $RMSE$ | $\lambda_1$ |
| STL | 0.642 | 2.07 | 0.662 | 2.28 | |
| O-MTL | 0.654 | 2.04 | 0.679 | 2.23 | 20 |

## 7. References

[1] Z.-H. Ling, K. Richmond, and J. Yamagishi, "Articulatory control of hmm-based parametric speech synthesis using feature-space-switched multiple regression," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, pp. 207–219, 2013.

[2] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "Unsupervised learning of acoustic features via deep canonical correlation analysis," in *Proc. of ICASSP*, Brisbane, Australia, 2015.

[3] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "Integrating articulatory data in deep neural network-based acoustic modeling," *Computer Speech and Language*, vol. 36, pp. 173–195, 2016.

[4] T. Hueber, G. Bailly, P. Badin, and F. Elisei, "Speaker adaptation of an acoustic-articulatory inversion model using cascaded gaussian mixture regressions," in *Proc. of Interspeech*, Lyon, France, 2013.

[5] A. Ben-Youssef, H. Shimodaira, and D. A. Braude, "Articulatory features for speech-driven head motion synthesis," in *Proc. of Interspeech*, Lyon, France, 2013.

[6] K. Richmond, S. King, and P. Taylor, "Modelling the uncertainty in recovering articulation from acoustics," *Computer Speech and Language*, vol. 17, no. 2, p. 153172, 2003.

[7] V. Mitra, H. Nam, C. Espy-Wilson, E. Saltzman, and L. Goldstein, "Retrieving tract variables from acoustics: a comparison of different machine learning strategies," *IEEE J. of Selected Topics in Signal Processing*, vol. 4, no. 6, p. 10271045, 2010.

[8] B. Uria, Murray, S. I., Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *Proc. of Interspeech*, Portland, Oregon, USA, 2012.

[9] C. Canevari, L. Badino, L. Fadiga, and G. Metta, "Cross-corpus and cross-linguistic evaluation of a speaker-dependent dnn-hmm asr system using ema data," in *Workshop on Speech Production for Automatic Speech Recognition*, Lyon, France, 2013.

[10] T. Toda, A. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model," *Speech Communication*, vol. 50, no. 3, p. 215222, 2007.

[11] B. Lindblom, J. Lubker, and T. Gay, "Formant frequencies of some fixed-mandible vowels and a model of speech motor programming by predictive simulation," *Journal of Phonetics*, vol. 7, p. 146161, 1979.

[12] S. Roweiss, "Data driven production models for speech processing," phD thesis, California Institute of Technology, Pasadena, California, 1999.

[13] C. Qin and M. Carreira-Perpiñan, "An empirical investigation of the nonuniqueness in the acoustic-to- articulatory mapping," in *Proc. of Interspeech*, Antwerp, Beglium, 2007.

[14] S. Najnina and B. Banerjee, "Improved speech inversion using general regression neural network," *The J. of Acoustical Society of America*, vol. 138, 2015.

[15] P. K. Ghosh and S. S. Narayanan, "An subject-independent acoustic-to-articulatory inversion," in *Proc. of ICASSP*, Prague, Czech Republic, 2011.

[16] J. R.Westbury, "X-ray microbeam speech production data- base users handbook version 1.0," waisman Center on Mental Retardation and Human Development, University of Wisconsin, Madison, WI, June 1994.

[17] T. Evegniou and M. Pontil, "Regularized multi-task learning," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[18] M. Donini, D. Martinez-Rego, M. Goodson, J. Shawe-Taylor, and M. Pontil, "Distributed variance regularized multitask learning," in *IEEE International Joint Conference on Neural Networks*, Vancouver, Canada, 2016.

[19] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. of ICSLP*, 1996.

[20] C. Bishp, *Neural Networks for Pattern Recognition*. Oxford, UK: Oxford University Press, 1995.

[21] R. C. Baciluă Cristian and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.

[22] W. Wang, R. Arora, and K. Livescu, "Reconstruction of articulatory measurements with smoothed low-rank matrix completion," in *IEEE SLT*, Lake Tahoe, Nevada, USA, 2014.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[24] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil, "Forward and reverse gradient-based hyperparameter optimization," *arXiv preprint arXiv:1703.01785*, 2017.

[25] C. Canevari, L. Badino, L. Fadiga, and G. Metta, "Relevance-weighted reconstruction of articulatory features in deep neural network- based acoustic-to-articulatory mapping," in *Proc. of Interspeech*, Lyon, France, 2013.