A TWO-STAGE DEEP MODELING APPROACH TO ARTICULATORY INVERSION

Abdolreza Sabzi Shahrebabaki[⋆]*, Negar Olfati*[⋆]*, Ali Shariq Imran*[⋆]*,*
Magne Hallstein Johnsen[⋆]*, Sabato Marco Siniscalchi*[⋆,†]*, Torbjørn Svendsen*[⋆]

[⋆]Department of Electronic Systems, NTNU
[†]Department of Computer Engineering, Kore University of Enna

ABSTRACT

This paper proposes a two-stage deep feed-forward neural network (DNN) to tackle the acoustic-to-articulatory inversion (AAI) problem. DNNs are a viable solution for the AAI task, but the temporal continuity of the estimated articulatory values has not been exploited properly when a DNN is employed. In this work, we propose to address the lack of any temporal constraints while enforcing a parameter-parsimonious solution by deploying a two-stage solution based only on DNNs: (i) Articulatory trajectories are estimated in a first stage using DNN, and (ii) a temporal window of the estimated trajectories is used in a follow-up DNN stage as a refinement. The first stage estimation could be thought of as an auxiliary additional information that poses some constraints on the inversion process. Experimental evidence demonstrates an average error reduction of 7.51% in terms of RMSE compared to the baseline, and an improvement of 2.39% with respect to Pearson correlation is also attained. Finally, we should point out that AAI is still a highly challenging problem, mainly due to the non-linearity of the acoustic-to-articulatory and one-to-many mapping. It is thus promising that a significant improvement was attained with our simple yet elegant solution.

Index Terms— Acoustic-to-articulatory inversion, deep learning, DNN, FBE.

1. INTRODUCTION

The acoustic-to-articulatory inversion (AAI) problem is concerned with estimating the vocal tract shape in the form of articulator positions based on the uttered speech. This inversion problem is highly non-linear because of the many-to-one mapping, which refers to the fact that different articulator configurations can produce the same sound. AAI is an important task in speech processing since it can be used in different speech technology applications, including automatic speech recognition (ASR) [1, 2, 3], and speech synthesis [4]. Different machine learning techniques, such as codebook-based models [5], Gaussian mixture models (GMMs) [6], hidden Markov models (HMMs) [7], mixture density networks [8], deep neural networks (DNNs) [9, 10, 11], and deep recurrent neural networks (RNNs) [12, 13, 14] have been used to address the AAI task.

In recent years, DNNs have proven useful in obtaining high accuracy articulatory inversion, but each articulator is estimated independently in the time domain although inter-articulator correlations exist among many of the measured articulators [15]. Such a shortcoming motivated the present investigation, and the use of a two-stage approach based on deep architectures for improving the estimation accuracy. In our solution, a DNN-based AAI is first trained to accomplish acoustic-to-articulatory inversion. This neural architecture represents our *baseline* system. In a second stage, an independent DNN refines the initial estimates provided by the baseline DNN; the second DNN aims to learn the aforementioned correlations leveraging upon the baseline estimates and the input acoustic features. It should be pointed out that the temporal dynamic of the estimated articulators is required for speech production systems, and the proposed method captures temporal dynamics of articulators by exploiting a temporal window of the estimated articulators.

The rest of the paper is structured as follows. In Section 2 deep regression models including the stand-alone DNN and the two-stage DNN achitecture are presented. Section 3 describes the ''Haskins IEEE Production Rate Comparison'' [16] database, feature representation, and the performance metrics used in this study, followed by the experimental results in Section 4. Finally, Section 5 concludes the paper.

2. DEEP REGRESSION MODELS

We discuss the DNN regression method for estimating the articulatory measurements in Section 2.1, and the proposed two-stage DNN model in Section 2.2.

2.1. DNN

Considering a wide temporal context when estimating the articulatory movements can be useful, as the co-articulation effect often extends beyond the phoneme level. Let $\boldsymbol{x}(i)$ be the acoustic feature vector for the i^{th} frame, the corresponding augmented vector, $\boldsymbol{x}_a(i)$, containing $\boldsymbol{x}(i)$ and its context is obtained as follows:

$$\boldsymbol{x}_a(i) = [\boldsymbol{x}(i-M)^\top, \ldots, \boldsymbol{x}(i)^\top, \ldots, \boldsymbol{x}(i+M)^\top]^\top, \quad (1)$$

where M denotes the number of left and right context frames, which are added to $\boldsymbol{x}(i)$. Letting $\boldsymbol{y}(i)$ be the i^{th} vector of the articulatory estimates, then the regression model for a feed-forward DNN with L hidden layers can be written as:

$$\hat{\boldsymbol{y}}(i) = g_{L+1}\big({W_{L+1}^1}^\top g_L\big({W_L^1}^\top \ldots \big(g_1\big({W_1^1}^\top \boldsymbol{x}_a(i)\big)\big)\big)\big), \quad (2)$$

where $(g_i, W_i^1), (i = 1, \ldots, L+1)$ are the activation function of the i^{th} layer, and the matrix of weights between the $(i-1)^{\text{th}}$ and the i^{th} layer, respectively. The the input layer corresponds to the 0^{th} layer. In regression, g_L is a linear activation function; whereas the remaining layers have hidden units with a non-linear activation function, such as $\mathrm{Sigmoid}$, $\tanh$, or ReLU. DNN parameters are optimized during the training phase using a gradient descent technique and the back propagation algorithm [17] with the goal of minimizing the

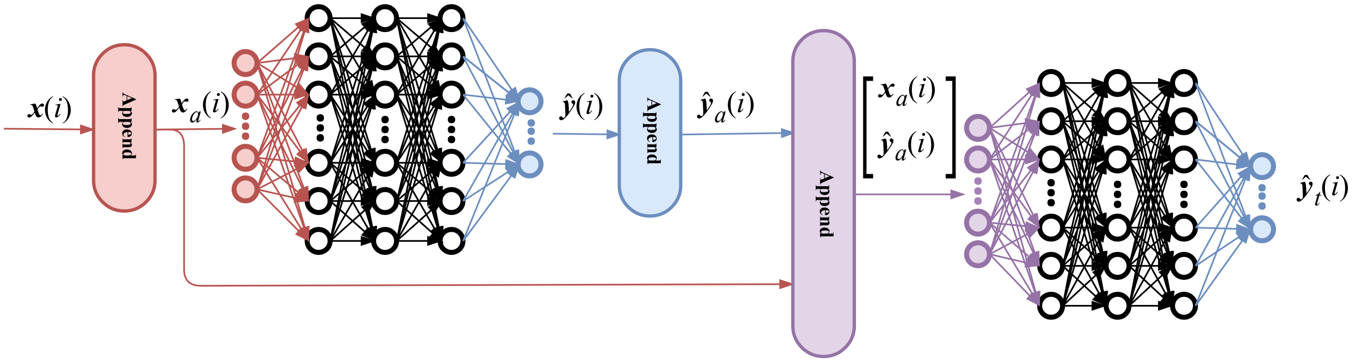
$\boldsymbol{x}(i)$
Append
$\boldsymbol{x}_a(i)$
$\hat{\boldsymbol{y}}(i)$
Append
$\hat{\boldsymbol{y}}_a(i)$
$\begin{bmatrix} \boldsymbol{x}_a(i) \\ \hat{\boldsymbol{y}}_a(i) \end{bmatrix}$
Append
$\hat{\boldsymbol{y}}_t(i)$

Fig. 1. Structure of the two-stage DNN regression model. The first stage performs a baseline AAI, while stage two refines the estimates by combining the first stage estimates with acoustic information. Sections in red, blue, purple and black correspond to acoustic, articulatory, joint acoustic+articulatory and hidden spaces, respectively.

mean squared error (MSE) between the estimated value, $\hat{\boldsymbol{y}}(i)$, and the ground truth value, $\boldsymbol{y}(i)$.

2.2. Two-stage DNN

The regression model described in Section 2.1 does not take into account the temporal continuity for estimated values of the articulatory measurements. To solve this problem, we propose a two-stage DNN-based model that considers a temporal window for the estimated values of the first stage. Let $\hat{\boldsymbol{y}}(i)$ be the output of the first DNN, and we expand its temporal context following the principle of Eq. (1), producing $\hat{\boldsymbol{y}}_a(i)$. By appending the acoustic data, $\boldsymbol{x}_a(i)$ and the concatenated outputs of the first DNN, $\hat{\boldsymbol{y}}_a(i)$, the DNN in the second stage can be trained on those new features as follows:

$$\begin{aligned}\hat{\boldsymbol{y}}_t(i) =& g_{L+1}\big(W_{L+1}^{2}{}^{\top} g_L\big(W_L^{2}{}^{\top} \\ & \dots\big(g_1\big(W_1^{2}{}^{\top}[\boldsymbol{x}_a(i)^\top, [\hat{\boldsymbol{y}}_a(i)^\top]^\top]\big)\big)\big)\big),\end{aligned} \tag{3}$$

where $(g_i, W_i^2), (i = 1, \dots, L+1)$ are the weight matrices and activation functions of the second DNN, respectively, and $\hat{\boldsymbol{y}}_t(i)$ denotes the output trajectory estimates. The structure of the proposed method is shown in Fig. 1.

3. EXPERIMENTAL SETUP

3.1. EMA database

Measurements of the articulatory movements can be done by different techniques such as MRI, microbeam X-ray, and electromagnetic articulography (EMA). The EMA method is one of the most used techniques for simultaneous recording of the speech and the articulatory data. One of the available databases is the ''Haskins IEEE Production Rate Comparison database'' [16], which contains recordings of eight native American English speakers, four female (F01-F04) and four male (M01-M04) speakers. 720 phonetically balanced Harvard sentences[18] are spoken with normal and fast speaking rate (SR) by each speaker. For some of the normal speaking rate utterances, there are repetitions available as well. In our experiments, only the normal speaking rate utterances were used. The amount of data for each data set is shown in Table 1, where ''N1'' and ''N2'' represent the main set of utterances spoken with normal SR and the set of repetitions of some of the normal SR sentences, respectively. The speech waveforms are sampled at the rate of 44.1 kHz, and the synchronously recorded EMA data have the sampling rate of 100 Hz. The EMA data is measured by eight sensors, placed at tongue rear or dorsum (TR), tongue blade (TB), tongue tip (TT), upper and lower lip (UL and LL), mouth left (ML), jaw or lower incisors (JAW) and jaw left (JAWL). The articulatory measurements are aligned to the occlusal plane in X, Y and Z directions, corresponding to movements from posterior to anterior, right to left and inferior to superior, respectively. The movements along the Y axis carry limited information and we thus only employed the measured data along X and Z axis. In this paper we used measurements for TR, TB, TT, UL, LL and JAW which are measured in previous EMA databases as well, such as MOCHA-TIMIT, MNGU0 and USC-TIMIT[19, 20, 21].

Table 1. Available amount of data in different data sets.

Speaking rate	NO. utterances	Amount of data
N1	5756	$\sim$ 244 (minutes)
N2	1379	$\sim$ 55 (minutes)

3.2. Acoustic representation

The acoustic features are extracted from resampled audio of 16 kHz with 25ms frame length and 10ms frame shift. The resulted features have 100 Hz sampling rate, like the articulatory features. The acoustic features are calculated from 40 filters which are linearly spaced on the Mel-scale frequency axis. The energies in the overlapping frequency bands constitute the filter bank energy (FBE) features. The extracted features are concatenated with the $M = 10$ past and future frames to generate the augmented vector $\boldsymbol{x}_a(i)$ specified in eq. (1).

3.3. Performance measurements

To measure the performance of the AAI methods, the root mean squared error (RMSE) and the Pearson's correlation coefficient (PCC) metrics are used. The first metric reports the deviation and the latter indicates the similarity between the estimated and the ground truth trajectories. These measures are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_i \big(y(i) - \hat{y}(i)\big)^2}, \tag{4}$$

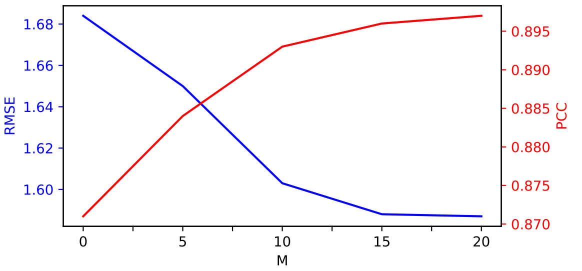

Fig. 2. Performance evaluation of two-stage DNN with different temporal context size over the the first stage DNN estimates. RMSE (Blue line) and PCC (Red line)

$$\text{PCC} = \frac{\sum_i (y(i) - \bar{y})(\hat{y}(i) - \bar{\hat{y}})}{\sqrt{\sum_i \left(y(i) - \bar{y}\right)^2 \sum_i \left(\hat{y}(i) - \bar{\hat{y}}\right)^2}}, \tag{5}$$

where $y(i)$ and $\hat{y}(i)$ are the ground truth and estimated EMA value of the i^{th} frame, respectively; $\bar{y}$, and $\bar{\hat{y}}$ are mean values of $y(i)$, and $\hat{y}(i)$. All results are based on training on the N1 subset, and testing on the N2 subset. 5% of the training data is used as the validation data, which is used to stop the training of DNN to prevent the network from getting over-fitted to the training data.

3.4. Network architecture

For optimizing the neural network parameters, several experiments were performed. We observed that a network configuration having 5 hidden layers with 300 units outperforms all other configurations tested in our laboratory. Moreover, rectified linear units (ReLU) [22] were used as the non-linear activation function for the hidden layers. The stochastic gradient descent (SGD) optimizer was employed to train the networks during the training phase. A batch size of 128 was chosen for all experiments. The actual implementations was carried out using Keras [23] with TensorFlow backend [24]. The same network architecture was used in both the first and the second stage DNN.

Table 2. Performance of AAI systems in terms of RMSE and PCC measures for the baseline (DNN) and the proposed method (2sDNN).

Spk.	RMSE-DNN	RMSE-2sDNN	PCC-DNN	PCC-2sDNN
F1	1.821	**1.594**	0.859	**0.895**
F2	1.959	**1.831**	0.786	**0.818**
F3	1.532	**1.432**	0.774	**0.806**
F4	1.896	**1.727**	0.847	**0.876**
M1	1.501	**1.384**	0.807	**0.841**
M2	1.938	**1.827**	0.819	**0.839**
M3	1.803	**1.676**	0.768	**0.803**
M4	1.516	**1.420**	0.782	**0.811**

4. RESULTS

The N1 subset of the data is used for training and testing is done on the N2 subset. 5% of the training data is used as validation data to avoid overtraining. Evaluation of speaker trained performance is done by training separate models for each speaker and testing on the corresponding speaker's test data. Joint training is done using the training speech from all speakers to train a single model which is tested on the pooled test data from all speakers. In this section, we first evaluate the effect of the temporal context size at the output of the first DNN level, $\hat{\boldsymbol{y}}(i)$, in two-stage DNN. Next, we provide the performance of both baseline and proposed two-stage systems in terms of RMSE and PCC. Finally, results for joint training are presented and compared with speaker trained performance.

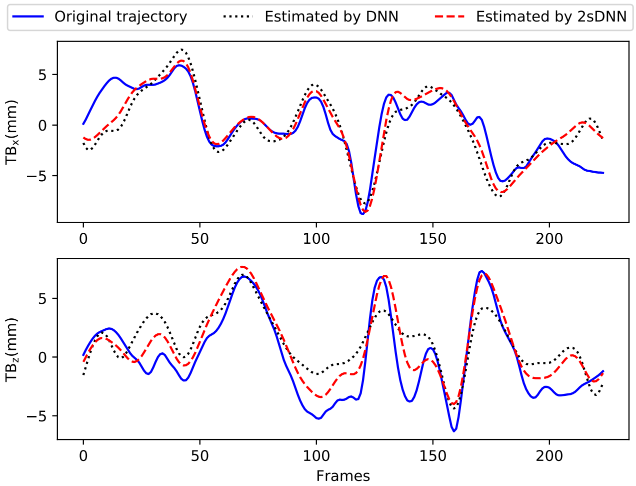

Fig. 3. Illustration of the TB trajectory for sentence ''Glue the sheet to the dark blue background'' and the low pass filtered estimated trajectories from 1st stage (black) and 2nd stage (red) DNNs.

4.1. Temporal context size

As discussed above,the outputs of the first stage DNN, are concatenated together to form the context vector, $\hat{\boldsymbol{y}}_a(i)$.

$$\hat{\boldsymbol{y}}_a(i)^\top = \{\hat{\boldsymbol{y}}(i-M)^\top, \ldots, \hat{\boldsymbol{y}}(i)^\top, \ldots, \hat{\boldsymbol{y}}(i+M)^\top\} \tag{6}$$

which is then fed into the second stage DNN. We evaluate the performance of temporal context size for $M = [0, 5, 10, 15, 20]$ frames on each side of the current output. The experimental results for the F1 speaker is shown in Figure. 2. From the figure, we can observe that introducing a temporal context of $M = 20$ causes a decrease in the RMSE by 0.1 mm and an increase in the PCC by 0.03 relative to using only static information. It can also be observed that the performance improves with increasing M, but flattens out for $M > 10$. Moreover, comparing the speaker F1 performance of the proposed two-stage system with the baseline system in Table 2, we can see from Fig. 2 that the proposed system without temporal context ($M = 0$), , attains 0.14 mm improvement in terms of RMSE, and a 0.01 improvement in terms of PCC when compared against the baseline system.

4.2. Performance evaluation

From Figure 2, we can observe that no significant improvement is achieved using $M \geq 15$. Therefore, $M = 15$ is chosen for generating the augmented vector, $\hat{\boldsymbol{y}}_a(i)$. Table 2 shows the experimental results with this setting for each speaker. Here, the AAI model is trained on data from the same speaker as the test speaker. The experimental evidence clearly demonstrates that a smaller RMSE is

Table 3. Performance of AAI systems in terms of RMSE and PCC measures for the estimated articulators.

Articulator positions	TD_x	TD_z	TB_x	TB_z	TT_x	TT_z	UL_x	UL_z	LL_x	LL_z	JAW_x	JAW_z
RMSE(mm)	1.692	2.181	1.991	2.125	2.103	2.214	0.758	1.332	0.859	0.845	1.079	1.992
PCC	0.838	0.841	0.849	0.868	0.858	0.863	0.839	0.849	0.763	0.787	0.826	0.844

Table 4. Performance of AAI systems in terms of average RMSE and PCC measures for speaker trained and jointly trained systems.

Spk.	RMSE-DNN	RMSE-2sDNN	PCC-DNN	PCC-2sDNN
Speaker trained	1.745	1.611	**0.805**	**0.836**
Jointly trained	**1.728**	**1.582**	0.802	0.835

attained using the two-stage DNN system compared to the baseline DNN system, and an averaged improvement of 0.13 mm is attained. Moreover, PCC of the two-stage DNN system is 0.03 better, on average, than that of the baseline DNN system. In Figure 3, we visualize the TB trajectories for both the baseline and the proposed two-stage systems for a randomly spoken utterance from the test set. The original EMA trajectories are the TB positions in X and Z axis. We can argue that the two-stage DNN approach achieves better estimation results for the peaks of TB articulator. The recordings contained initial and final silences, which were removed from the training data. Figure 3 shows the complete utterance including initial and final silences. The trajectories at the beginning and end are thus not indicative of the performance during speech.

We also implemented a system using joint training of speakers. RMSE and PCC average values for each of the articulators obtained by joint training is reported in Table 3. The overall RMSE and PCC average values are given in Table 4. From those results, we can conclude that that performance of the joint training over all speakers is slightly better than that of the per speaker training, in terms of RMSE for both architectures.

5. CONCLUSION

This paper has addressed the issue of AAI, and proposed utilizing a two-stage DNN regression model. The architecture is designed using a two-stage DNN approach that allows inter-articulator correlations by leveraging temporal information from articulator estimates captured in the first stage. By appending the acoustic data to the context augmented output of the first stage DNN architecture the trajectory estimates are subsequently refined by the second DNN stage architecture. The model is evaluated on a EMA database for six articulatory locations using FBE acoustic features computed using 25ms frame length and 10ms frame shift. Employing this database is of interest because multi-speaker measurements are available for both male and female speakers which will be useful for further investigations on the speaker independent training AAI systems. Compared to the baseline model, r a higher Pearson correlation and a significant reduction in the RMSE is observed for each of the male and female speakers.

For the future work, we will consider exploration of recurrent layers in our structure which leads to have a smoother estimated trajectory compared to the feed forward networks. Also, having convolutional layers on top of estimated trajectories before concatenating them to the input vectors, will be of interest to see what will be the frequency response of the filters.

6. ACKNOWLEDGEMENTS

This work has been supported by the Research Council of Norway through the project AULUS, and by NTNU through the project ArtiFutt. Prof. Siniscalchi is supported by PRIN2017 JNKCYZ_002 project.

7. REFERENCES

[1] Simon King, Joe Frankel, Karen Livescu, Erik McDermott, Korin Richmond, and Mirjam Wester, "Speech production knowledge in automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 723–742, 2007.

[2] Prasanta Kumar Ghosh and Shrikanth Narayanan, "Automatic speech recognition using articulatory features from subject-independent acoustic-to-articulatory inversion," *The Journal of the Acoustical Society of America*, vol. 130, no. 4, pp. EL251–EL257, 2011.

[3] 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.

[4] Zhen-Hua Ling, Korin Richmond, and Junichi 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, no. 1, pp. 207–219, 2013.

[5] B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey, "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1535–1555, 1978.

[6] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Communication*, vol. 50, no. 3, pp. 215–227, 2008.

[7] Z. Le and S. Renals, "Acoustic-articulatory modeling with the trajectory HMM," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008.

[8] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Ninth International Conference on Spoken Language Processing*, 2006.

[9] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[10] P. L. Tobing, H. Kameoka, and T. Toda, "Deep acoustic-to-articulatory inversion mapping with latent trajectory modeling," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec 2017, pp. 1274–1277.

[11] Nadee Seneviratne, Ganesh Sivaraman, Vikramjit Mitra, and Carol Espy-Wilson, "Noise robust acoustic to articulatory speech inversion," in *Proc. Interspeech 2018*, 2018, pp. 3137–3141.

[12] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *ICASSP*, 2015, pp. 4450–4454.

[13] P. Zhu, X. Lei, and Y. Chen, "Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings," in *Interspeech*, 2015, pp. 2192–2196.

[14] X. Xie, X. Liu, and L. Wang, "Deep neural network based acoustic-to-articulatory inversion using phone sequence information," in *Interspeech 2016*, 2016, pp. 1497–1501.

[15] P. J.B. Jackson and V. D. Singampalli, "Statistical identification of articulation constraints in the production of speech," *Speech Communication*, vol. 51, no. 8, pp. 695 – 710, 2009.

[16] M. Tiede, C. Y. Espy-Wilson, D. Goldenberg V. Mitra, H. Nam, and G. Sivaraman, "Quantifying kinematic aspects of reduction in a contrasting rate production task," *The Journal of the Acoustical Society of America*, vol. 141, no. 5, pp. 3580–3580, 2017.

[17] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning internal representations by error propagation," Tech. Rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[18] "IEEE recommended practice for speech quality measurements," *IEEE No 297-1969*, pp. 1–24, 1969.

[19] A. Wrench, "The MOCHA-TIMIT articulatory database," `http://www.cstr.ed.ac.uk/artic/mocha.html`, 1999.

[20] R. Korin, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the MNGU0 articulatory corpus," in *Interspeech*, Florence, Italy, August 2011, pp. 1505–1508.

[21] S. Narayanan and et al, "Real-time magnetic resonance imaging and electromagnetic articulography database for speech production research (tc)," *The Journal of the Acoustical Society of America*, vol. 136, no. 3, pp. 1307–1311, 2014.

[22] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947, 2000.

[23] François Chollet et al., "Keras," `https://keras.io`, 2015.

[24] M. Abadi and et al., "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.