Deep Architectures for Articulatory Inversion

Benigno Uria[1], *Iain Murray*[1], *Steve Renals*[2], *Korin Richmond*[2]

[1]Institute for Adaptive and Neural Computation, University of Edinburgh, Scotland, United Kingdom
[2]Centre for Speech Technology Research, University of Edinburgh, Scotland, United Kingdom
b.uria@ed.ac.uk, i.murray@ed.ac.uk, s.renals@ed.ac.uk, korin@cstr.ed.ac.uk

Abstract

We implement two deep architectures for the acoustic-articulatory inversion mapping problem: a deep neural network and a deep trajectory mixture density network. We find that in both cases, deep architectures produce more accurate predictions than shallow architectures and that this is due to the higher expressive capability of a deep model and not a consequence of adding more adjustable parameters. We also find that a deep trajectory mixture density network is able to obtain better inversion accuracies than smoothing the results of a deep neural network. Our best model obtained an average root mean square error of 0.885 mm on the MNGU0 test dataset.

Index Terms: Articulatory inversion, deep neural network, deep belief network, deep regression network, pretraining

1. Introduction

The acoustic-articulatory inversion mapping problem (or simply articulatory inversion) involves the inference of the position of the vocal tract articulators from an acoustic speech signal. This nonlinear regression problem is ill-posed: several articulator positions can generate the same sound.

Systems capable of approximating the position of the articulators from the acoustic signal are useful in several domains: speech recognition, where articulatory information can improve the performance of the recognition systems [1]; speech synthesis, where it can be used to improve the quality or to modify the characteristics of the synthesised voice [2]; character animation, where it can be used to automate the facial animation of virtual characters in films and video-games [3].

Rich articulography datasets of precise quantitative articulatory position data along with recordings of the acoustic data produced, make it possible to use standard machine learning methodologies such as artificial neural networks [4] or hidden Markov models [5] to tackle this problem. Deep architectures have recently been used to obtain state-of-the-art accuracies in speech recognition [6, 7]. Motivated by this success, we hypothesised that a deep architecture would be able to obtain high accuracy in articulatory inversion.

In this work we have implemented a deep neural network and a deep trajectory mixture density network and evaluated its performance using the MNGU0 test dataset [8][1]. Both approaches resulted in significant reductions in average root mean squared error (RMSE), with respect to the best previously published results obtained by Richmond [9] using a shallow trajectory mixture density network. We conclude that deep architectures are a suitable approach to articulatory inversion.

[1]We previously presented preliminary results using the deep neural network at the NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning, which does not have official proceedings.

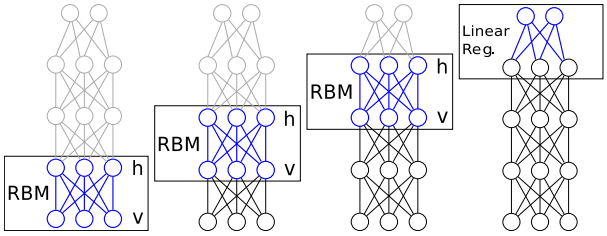

Figure 1: Unsupervised layer-wise initialisation of a regression neural network with three hidden layers. Hidden layers are sequentially trained as Restricted Boltzmann Machines (RBMs). The top layer performs a linear regression.

2. Deep regression neural networks

One hidden-layer neural networks are universal approximators [10]—given enough hidden units they can implement any function to an arbitrary degree of accuracy. However, a network with several hidden layers can implement some functions using an exponentially smaller number of hidden units than in a one hidden layer network [11].

Until recently, training networks with more than one or two hidden layers was usually considered impractical. Back-propagation training progresses very slowly when initialised using very small weights, and may get stuck in poor local optima or weight-space plateaus when randomly initialised using bigger weights [12]. However many-hidden-layer neural networks have been trained successfully through stacked or modular use of back-propagation [13, 14] or by pretraining using restricted Boltzmann machines (RBMs) [15] (see section 2.1).

In contrast to the work in speech recognition [6, 7], in which deep neural networks are trained for classification, we train deep regression networks in this paper. Figure 1 illustrates a three-hidden-layer network initialised using unsupervised pretraining. The training process begins by pretraining the first hidden layer of the network using an RBM, considering the input and first hidden layers as the visible and hidden layers of the RBM respectively. Each remaining hidden layer is pretrained sequentially by treating the latest pretrained hidden layer as the visible layer of a new RBM, and obtaining the visible inputs to that RBM by feeding the input data through the layers trained previously. A deep regression network is completed by adding a linear regression layer on top, which can also be initialised; in the case of regression this can be done directly using the least-square error solution. Finally, all of the parameters in the network are fitted discriminatively: stochastic gradient descent is used to minimize the mean square error.

2.1. Restricted Boltzmann machines

Restricted Boltzmann machines (RBMs) [16] are undirected graphical models formed by two layers of probabilistic binary units: a visible layer $\mathbf{v}$, and a hidden layer $\mathbf{h}$. All units are fully connected to the units in the other layer and no connections between units of the same layer are present.

Restricted Boltzmann machines are energy-based models. They capture dependencies between variables by assigning an energy value to each configuration, with more probable configurations having a lower energy. Because no connections exist between units of the same layer, the probability of the units in each layer factorise given the state of the other layer.

Maximum likelihood training of an RBM requires the computation of intractable expectations, which could be approximated with Gibbs sampling. A further popular approximation is known as contrastive divergence (CD) learning [16, 17], which uses the following update rules for the weights:

$$\Delta W_{ij} \quad \propto \quad \langle \mathbf{v}_i \mathbf{h}_j \rangle_0 - \langle \mathbf{v}_i \mathbf{h}_j \rangle_n, \tag{1}$$

where $\langle \cdot \rangle_n$ denotes the average after assigning the input data to the visible units and performing just n updates (as in Gibbs sampling) to each layer. Whereas for maximum likelihood training the number of updates required to reach the stationary distribution can be very high, in contrastive divergence learning a very low number of updates is used, usually just one.

2.2. Gaussian-Bernoulli RBM

For problems with real valued input features, having binary visible units is not an appropriate representation of the data. In these cases a Gaussian-Bernoulli RBM [11] can be used. In a Gaussian-Bernoulli RBM visible units are real valued and follow a Gaussian probability distribution with diagonal covariance, while hidden units are binary valued and follow a Bernoulli distribution.

3. Mixture density networks

Because articulatory inversion is ill-posed, Richmond [18] has represented uncertainty in the regression output with mixture density networks. As shown in Figure 2, a mixture density network (MDN) [19] is a regression neural network whose outputs specify a mixture of Gaussians. The MDN maps each input datapoint x, acoustic data, to a probability distribution over the output space t, the position of each articulator; thus implementing a conditional probability distribution $p(t|x) \sim \sum_j \mathcal{N}(\mu_j(x), \Sigma_j(x))$.

The parameters of the mixture of Gaussians are derived from the unconstrained outputs $\mathbf{y}$ of the neural network as follows: the outputs $\mathbf{y}_\alpha$ specifying the mixing fractions are transformed with a softmax, $\alpha_i = \exp(y^i_\alpha) / \sum_j \exp(y^j_\alpha)$; the outputs corresponding to the means are not modified $\mu_i = y^i_\mu$; and the outputs corresponding to the variances are exponentiated, $\sigma_i = \exp(y^i_\sigma)$.

MDNs are trained to minimise the negative log likelihood of its parameters given the data. The gradient with respect to all parameters can be easily calculated and used to train the network using, for example, stochastic gradient descent.

A deep mixture density network can be constructed using the same layer-wise pretraining method as for a deep neural network. However, its last layer cannot be pretrained directly; stochastic gradient descent on its parameters can be used, while clamping all parameters in previous layers.

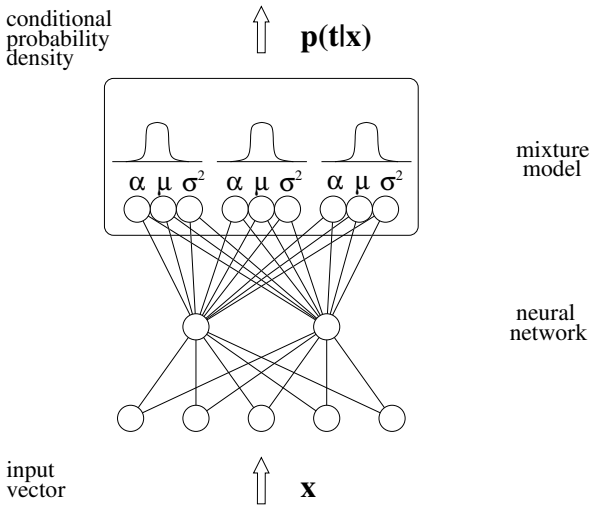

Figure 2: Diagram of a mixture density network (MDN).

When, as in the articulatory inversion case, we want to infer a trajectory, a mixture density network can be used to obtain a pdf over the position at each time step. The outputs of such a mixture density network can be augmented with their dynamic components Δ and $\Delta\Delta$, calculated as finite differences. From the sequence of pdfs over the position and dynamics the most probable trajectory can be inferred using the maximum likelihood parameter generation (MLPG) algorithm [20]. When the density network output has just one Gaussian mixture component, the MLPG algorithm only involves solving a linear system. When the Gaussian mixture model has several components, an EM-based algorithm is used.

4. Electromagnetic midsagittal articulography dataset

Electromagnetic midsagittal articulography (EMA) is the most widely used articulography technique for the creation of parallel acoustic and articulator-position recordings. It uses electromagnetic transducer coils glued to the vocal-tract articulators to record precise measurements of their positions [21].

In this paper we use the MNGU0 dataset because it has higher accuracy [9] than other EMA datasets such as MOCHA-TIMIT. MNGU0 consists of 1263 utterances recorded from a single speaker in a single session. Parallel recordings of acoustic data and the position of 6 coils is available. Transducer coil positioning can be seen in Figure 3. Each EMA data frame is made up of 12 coordinates, the x and y position for the six articulators tracked, with a sampling frequency of 200 Hz. The acoustic data consists of frames of 40 frequency warped line spectral frequencies (LSFs) [22] and a gain value, the frame shift step is 5 ms in order to obtain acoustic features at the same frequency as the EMA data.

In our experiments we used a context window of 10 acoustic frames selecting only every other frame. Therefore, each input window spanned a period of about 90 ms. Our target output was the EMA frame for the time at the middle of the current acoustic window, i.e. between acoustic frames 5 and 6.

The dataset is partitioned into three sets: validation and testing sets comprising 63 utterances each, and a training set consisting of the other 1137 utterances.

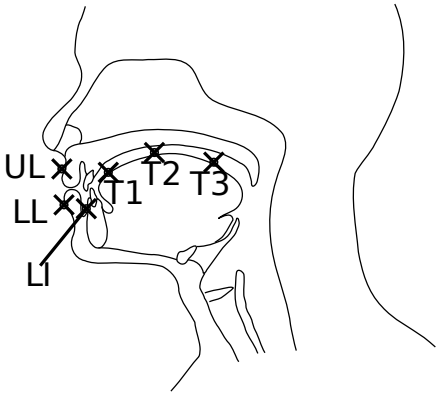

Figure 3: Positioning of electromagnetic coils in the MNGU0 dataset. The articulators tracked are: upper lip (UL), lower lip (LL), lower incisor (LI), tongue tip (T1), tongue blade (T2), and tongue dorsum (T3).

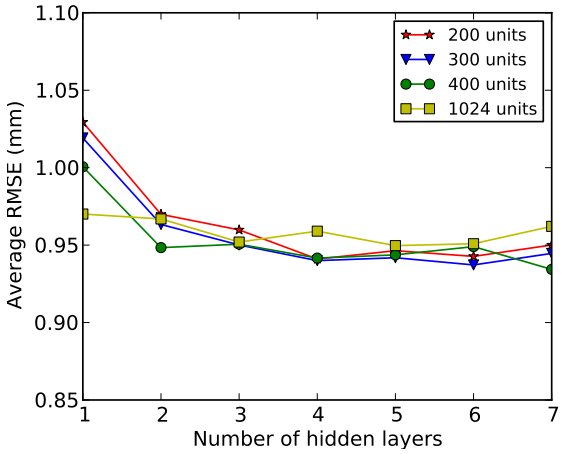

Figure 4: RMSE performance of neural networks as a function of the number of layers. The best results were achieved by networks with many hidden layers.

5. Experiments

To measure the accuracy of our system we will use the average over utterances of the root mean-squared error (RMSE), $\sqrt{\frac{1}{N}\sum_i (e_i - t_i)^2}$, where e_i is the estimated tract variable and t_i the actual tract variable at time i.

To put our later results in context, we report here the MNGU0 test set performance of three baselines: 1) A linear model obtains an average RMSE of 1.52 mm. 2) A one-hidden-layer artificial neural network with 300 units, obtains an average RMSE of about 1.13 mm. 3) A state of the art method [9], using twelve (one per articulator dimension) one-hidden-layer trajectory mixture density networks with 1-, 2-, or 4-component Gaussian mixtures (the number is optimised for each channel) obtains an average RMSE of 0.99 mm.

5.1. Pretraining

To serve as an initial configuration for the parameters of both of our deep articulatory inversion systems, we pretrained a set of stacked RBMs using the acoustic data in the MNGU0 dataset. Given that acoustic data is real valued, we used a Gaussian-Bernoulli RBM for the first layer. Higher layers were trained on top by following the conventional procedure [15]: the weights of the layer just trained are frozen and its hidden layer activation probabilities used as pseudo-binary values for the visible units of the new layer. Both the visible and hidden units of higher layers are treated as binary valued, and fitted with the regular RBM procedure.

5.2. Deep neural networks for articulatory inversion

We trained different configurations of deep neural networks to do articulatory inversion. The x and y coordinates of each of the 6 articulators in the MNGU0 dataset corresponds to an output of the neural network. In order to train this neural networks, we initialised its parameters using the values obtained during pretraining. A top regression layer was added on top and initialised with the least-square error solution. Then we performed stochastic gradient descent to fine-tune all parameters.

Generating the output for each time step independently gave rise to jagged trajectories with poor performance (RMSE $>$ 1 mm). To obtain smooth trajectories we applied a zero-shift second-order Butterworth filter to the outputs [23]. Since different articulators can have different dynamical characteristics, we calculated the RMSE on the MNGU0 training dataset using integer cut-off frequencies in the range 1-20 Hz for each channel and chose the optimal for each. Several architectures were trained, with different numbers of hidden layers and units per layer. Figure 4 shows the mean RMSE on the test dataset. The lowest mean RMSE on the validation dataset was obtained by a 5 hidden-layer-network with 300 units per layer, which has an average RMSE of 0.942 mm on the test dataset.

Our results suggest that a deep architecture can obtain better results than a shallow 1-hidden layer system. It could be argued that it is the use of a greater number of parameters and not the hidden layers that matters. However, the 1-hidden layer model with 1024 hidden units has almost twice as many parameters than a 7-hidden-layers model with 200 units per layer, yet still obtains less accurate results.

To check whether we could have achieved similar results without the pretraining phase, we trained a set of neural networks with weights initialised from a Gaussian distribution with mean zero and standard deviation 0.05. In all twenty one cases, a randomly initialised neural network obtained lower inversion accuracy; with an average RMSE 0.051mm higher.

5.3. Deep trajectory mixture density network

We also trained a series of deep trajectory mixture density networks. The training process is very similar to that of our neural networks. We utilised the parameters obtained in the pretraining phase to initialise the network hidden layer weights and biases, and added a linear layer on top whose outputs are adapted to obtain the parameters of a Gaussian mixture model. The EMA data in the MNGU0 dataset was augmented with Δ and $\Delta\Delta$ features in order to use the MLPG algorithm. Therefore, each trajectory MDN models a 36-dimensional probability distribution conditioned on the acoustic inputs. A plot of the mean RMSE on MNGU0's test dataset after applying the MLPG algorithm to the TMDN's output can be seen in Figure 5. The lowest mean RMSE on the validation dataset was obtained by a 6 hidden-layer network with 300 units per layer, which has an average RMSE of 0.885 mm on the test dataset. We repeated

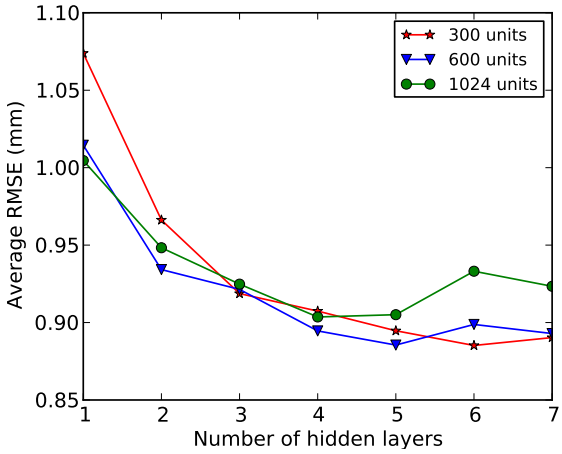

Figure 5: RMSE performance of mixture density network as a functions of the number of layers. The best results were achieved by networks with many hidden layers.

the experiments initialising the TMDN weights randomly from a Gaussian distribution with mean zero and standard deviation 0.05. In all cases, a randomly initialised TMDN obtained lower inversion accuracy, with an average RMSE 0.054mm higher.

6. Discussion

We have implemented two different kinds of articulatory inversion systems: a neural network and a trajectory mixture density network. In both cases, we have found a deep architecture is able to obtain better inversion accuracy than a one-hidden-layer architecture.

All TMDN results shown in this paper used only one Gaussian component, making them a simple Gaussian model. We repeated our experiments with 2 and 4 components, but their accuracy was slightly worse in almost all cases. Richmond's previous work selected different number of components for each articulator, which remains a possible improvement to the method implemented in this paper.

We also repeated most experiments initialising our network weights randomly. In all cases, the networks that had been pretrained obtained better results.

Finally, articulatory data exists as a time series, and it is not possible to obtain competitive results by applying regression methods independently to narrow time windows. This work provides a first demonstration of applying deep architectures to a complex, time-varying regression problem, and defines the new state-of-the-art for this task.

7. Acknowledgements

This work was supported by an EPSRC CASE studentship in collaboration with Novauris Technologies, and by EPSRC Healthcare Partnerships grant EP/I027696/1 (Ultrax).

8. References

[1] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 121, pp. 723–743, 2007.

[2] Z. Ling, K. Richmond, J. Yamagishi, and R. Wang, "Integrating articulatory features into HMM-based parametric speech synthesis," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 6, pp. 1171–1185, 2009.

[3] G. Hofer and K. Richmond, "Comparison of HMM and TMDN methods for lip synchronisation," in *Proc. Interspeech*, Makuhari, Japan, September 2010, pp. 454–457.

[4] K. Richmond, S. King, and P. Taylor, "Modelling the uncertainty in recovering articulation from acoustics," *Computer Speech and Language*, vol. 17, pp. 153–172, 2003.

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

[6] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modelling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[7] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[8] K. Richmond, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus," in *Proc. Interspeech*, 2011.

[9] K. Richmond, "Preliminary inversion mapping results with a new EMA corpus," in *Proc. Interspeech*, Brighton, UK, Sep. 2009, pp. 2835–2838.

[10] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[11] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems*, vol. 19. Vancouver, Canada: The MIT Press, Dec. 2007, p. 153.

[12] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[13] F. Grézl, M. Karafiát, and M. Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *Proc. IEEE ASRU*, 2011.

[14] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. IEEE ASRU*, 2011.

[15] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[16] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[17] M. Carreira-Perpiñán and G. Hinton, "On contrastive divergence learning," in *Proc. Artificial Intelligence and Statistics*, 2005, pp. 33–40.

[18] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Proc. Interspeech*, Pittsburgh, USA, Sep. 2006.

[19] C. Bishop, "Mixture density networks," Neural Computing Research Group, Aston University, Tech. Rep. NCRG/94/004, 1994.

[20] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, Istanbul, Turkey, Jun. 2000, pp. 1315–1318.

[21] J. Perkell, M. Cohen, M. Svirsky, M. Matthies, I. Garabieta, and M. Jackson, "Electromagnetic midsagittal articulometer systems for transducing speech articulatory movements," *The Journal of the Acoustical Society of America*, vol. 92, p. 3078, 1992.

[22] H. Strube, "Linear prediction on a warped frequency scale," *The Journal of the Acoustical Society of America*, vol. 68, p. 1071, 1980.

[23] K. L. Poort, "Stop consonant production: An articulation and acoustic study," Master's thesis, MIT, 1995.