Modeling unsupervised phonetic and phonological learning in Generative Adversarial Phonology

Gašper Beguš
Department of Linguistics, University of Washington
begus@uw.edu

Abstract

This paper models phonetic and phonological learning as a dependency between random space and generated speech data in the Generative Adversarial Neural network architecture and proposes a methodology to uncover the network's internal representation that corresponds to phonetic and phonological features. A Generative Adversarial Network (Goodfellow et al. 2014; implemented as WaveGAN for acoustic data by Donahue et al. 2019) was trained on an allophonic distribution in English, where voiceless stops surface as aspirated word-initially before stressed vowels except if preceded by a sibilant [s]. The network successfully learns the allophonic alternation: the network's generated speech signal contains the conditional distribution of aspiration duration. Additionally, the network generates innovative outputs for which no evidence is available in the training data, suggesting that the network segments continuous speech signal into units that can be productively recombined. The paper also proposes a technique for establishing the network's internal representations. We identify latent variables that directly correspond to presence of [s] in the output. By manipulating these variables, we actively control the presence of [s], its frication amplitude, and spectral shape of the frication noise in the generated outputs.

1 Introduction

Modeling phonetic and phonological data with neural networks has seen a rapid increase in the past few years (Alderete et al. 2013; Avcu et al. 2017; Alderete and Tupper 2018; Mahalunkar and Kelleher 2018; Weber et al. 2018; Dupoux 2018; Prickett et al. 2019; Pater 2019, for cautionary notes, see Rawski and Heinz 2019). The majority of existing computational models in phonology, however, model learning as symbol manipulation and operate with discrete units—either with completely abstract made-up units or with discrete units that feature some phonetic properties that can be approximated as phonemes. This means that either the phonetic and phonological learning are modeled separately or one is assumed to have already been completed with a pre-assumed level of abstraction (Martin et al., 2013; Dupoux, 2018). This is true for both proposals that model phonological distributions or derivations (Alderete et al., 2013; Prickett et al., 2019) and featural organizations (Faruqui et al., 2016; Silfverberg et al., 2018).

Most models in the subset of the proposals that operate with continuous phonetic data assume at least some level of abstraction and operate with already extracted features (e.g. formant values) on limited "toy" data (e.g. Pierrehumbert 2001; Kirby and Sonderegger 2015 for a discussion, see Dupoux 2018). Guenther and Vladusich (2012), Guenther (2016) and Oudeyer (2001, 2002, 2005, 2006), for example, propose models that use simple neural maps that are based on actual correlates of neurons involved in speech production in the human brain (based on various brain imaging techniques). Their models, however, do not operate with raw acoustic data (or require extraction of features in a highly abstract model of articulators; Oudeyer 2005, 2006), require a level of abstraction in the input to the model, and do not model phonological processes — i.e. allophonic distributions. Phonological learning in most of these proposals is thus modeled as if phonetic learning (or at least a subset of phonetic learning) had already taken place: the initial state already includes phonemic inventories, phonemes as discrete units, feature matrices that had already been learned, or extracted phonetic values.

Prominent among the few models that operate with raw phonetic data are Gaussian mixture models for category-learning or phoneme extraction

Proceedings of the Society for Computation in Linguistics (SCiL) 2020, pages 138-148.
New Orleans, Louisiana, January 2-5, 2020

(Schatz et al., 2019; Lee and Glass, 2012). Schatz et al. (2019) propose a Dirichlet process Gaussian mixture model that learns categories from raw acoustic input in an unsupervised learning task. The primary purpose of the proposal in Schatz et al. (2019) is modeling perception and categorization: they model how a learner is able to categorize raw acoustic data into sets of discrete categorical units that have phonetic values (i.e. phonemes). No phonological processes are modeled in the proposal.

Recently, neural network models for unsupervised feature extraction have seen success in modeling acquisition of phonetic features from raw acoustic data (Kamper et al., 2015). The model in Shain and Elsner (2019), for example, is an autoencoder neural network that is trained on pre-segmented acoustic data. The model takes as an input segmented acoustic data and outputs values that can be correlated to phonological features. Learning is, however, not completely unsupervised as the network is trained on pre-segmented phones. Thiollière et al. (2015) similarly propose an architecture that extracts units from unsupervised speech data. These proposals, however, do not model learning of phonological distributions, but only of feature representations, and crucially are not generative, meaning that the models do not output innovative data, but try to replicate the input as closely as possible (e.g. in the autoencoder architecture).

As argued below, the model based on a Generative Adversarial network learns not only to generate innovative data that closely resemble human speech, but also learns internal representations that resemble phonological features simultaneously with unsupervised phonetic learning from raw acoustic data. Additionally, the model is generative and outputs both the conditional allophonic distributions in the data and innovative data that can be compared to productive outputs in human speech acquisition.

1.1 A Generative Adversarial model of phonology

The advantage of the GAN architecture (Goodfellow et al., 2014; Radford et al., 2015; Donahue et al., 2019) is that learning is completely unsupervised and that phonetic learning is simultaneous with phonological learning in its broadest sense. A network that models learning of phonetics from raw data and shows signs of learning discrete phonological units at the same time is likely one step closer to reality than models that operate with symbolic computation and assume phonetic learning had already taken place and is independent of phonology and vice versa. The Generator's outputs can be approximated as the basis for articulatory targets in human speech that are sent to articulators for execution. The latent variables in the input of the Generator can be modeled as featural representation that the Generator learns to output into a speech signal by attempting to maximize the error rate of a Discriminator network that distinguishes between real data and generated outputs. The Discriminator network thus has a parallel in human speech perception, production, and acquisition: the imitation principle (Nguyen and Delvaux, 2015). The Discriminator's function is to enforce that the Generator's outputs resemble (but not replicate) the inputs as closely as possible. The GAN network thus incorporates both the pre-articulatory production elements (the Generator) as well as the perceptual element (the Discriminator) in speech acquisition. While other neural network architectures might be appropriate for modeling phonetic and phonological learning, GAN is unique in that it is a generative model with the production-perception loop parallel and that, unlike for example autoencoders, generates innovative data rather than data that resembles the input as closely as possible. To our knowledge, this is the first proposal that tests whether neural networks are able to learn an allophonic distribution based on raw acoustic data.

We train a Generative Adversarial Network architecture implemented for audio files in Donahue et al. (2019) (WaveGAN; which is based on DCGAN; Radford et al. 2015) on continuous raw speech data that contains information for an allophonic distribution: word-initial pre-vocalic aspiration of voiceless stops ([ˈpʰɪt] ∼ [ˈspɪt]). The data is curated in order to control for non-desired effects, which is why only sequences of the shape #TV and #sTV (T = stop, V = vowel) are fed to the model. This allophonic distribution is uniquely appropriate for testing learnability in a GAN setting, because the dependency between the presence of [s] and duration of VOT is not strictly local. To be sure, the dependency is local in phonological terms, as [s] and T are two segments and immediate neighbors, but in phonetic terms, a pe-

riod of closure intervenes between the aspiration and the period (or absence thereof) of frication noise of [s].

The hypothesis of the computational experiment presented in Section 3 is the following: if VOT duration is conditioned on the presence of [s] in output data generated from noise by the Generator network, it means that the Generator network has successfully learned a phonetically non-local allophonic distribution. Because the allophonic distribution is not strictly local and not automatic, but has to be learned and actively controlled by speakers, evidence for this type of learning is considered phonological learning in the broadest sense. Conditioning the presence of a phonetic feature based on the presence or absence of a phoneme that is not automatic is, in most models, considered part of phonology and is derived with phonological computation. That the tested distribution is non-automatic and has to be actively controlled by the speakers is evident from L1 acquisition: failure to learn the distribution results in longer VOT durations in the sT condition documented in L1 acquisition (McLeod et al., 1996; Bond, 1981). Additional evidence that the GAN's learning resembles phonemic representations (such as presence of [s]) is obtained from recovering the networks' internal representations (see below and Section 3.2).

This paper also proposes a technique for establishing the Generator's internal representations. What neural networks actually learn is a challenging question with no easy solutions. The inability to uncover networks' representations has been used as an argument against neural network approaches to linguistic data (Rawski and Heinz, 2019). We argue that internal representation of a network can be, at least partially, uncovered. By regressing annotated dependencies between the Generator's latent space and output data, we identify values in the latent space that correspond to linguistically meaningful features in generated outputs. This paper demonstrates that manipulating the chosen values in the latent space have phonetic and phonological effects in the generated outputs, such as the presence of [s] and the amplitude of its frication. In other words, the GAN network learns to use random noise as an approximation of phonetic and phonological features. This paper proposes that dependencies, learned during training in a latent space that is limited by some interval, extend beyond that interval. This crucial step allows for the discovery of several phonetic properties.

2 Materials

2.1 The model: Donahue et al. (2019) based on Radford et al. (2015)

Generative Adversarial Networks, proposed by Goodfellow et al. (2014), have seen a rapid expansion in a variety of tasks, including but not limited to computer vision and image generation (Radford et al., 2015). The main characteristic of GANs is the architecture that involves two networks: the Generator network and the Discriminator network (Goodfellow et al., 2014). The Generator network is trained to generate data from random noise, while the Discriminator is trained to distinguish real data from the outputs of the Generator network (Figure 1). The Generator is trained to generate data that maximizes the error rate of the Discriminator network. The training results in a Generator (G) network that takes random noise as its input (e.g. multiple variables with uniform distributions) and outputs data such that the Discriminator is inaccurate in distinguishing the generated from the real data.

Applying the GAN architecture on time-series data such as a continuous speech stream faces several challenges. Recently, Donahue et al. (2019) proposed an implementation of a Deep Convolutional Generative Adversarial Network proposed by Radford et al. (2015) for audio data (WaveGAN); the model along with the code in Donahue et al. (2019) was used for training in this paper. The model takes one-second long raw audio files as inputs, sampled at 16 kHz with 16-bit quantization. The audio files are converted into a vector and fed to the Discriminator network as real data. Instead of the two-dimensional 5×5 filters, the WaveGAN model uses one-dimensional 1×25 filters and larger upsampling (Donahue et al., 2019). The main architecture is preserved as in DCGAN, except that an additional layer is introduced in order to generate longer samples. The Generator network takes as input z, a vector of one hundred uniformly distributed variables ($z \sim \mathscr{U}(-1,1)$) and outputs 16,384 data points, which constitutes the output audio signal. The network has five 1D convolutional layers (Donahue et al., 2019). The Discriminator network takes 16,384 data points (raw audio files) as its input and outputs a sin-

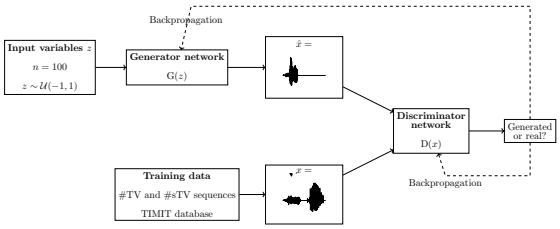

Figure 1: A diagram showing the Generative Adversarial architecture as proposed in Goodfellow et al. (2014); Donahue et al. (2019) and trained on data from the TIMIT database in this paper.

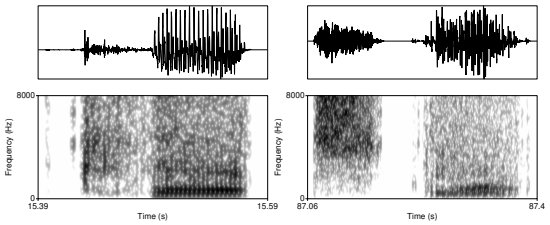

Figure 2: Waveforms and spectrograms (0–8,000 Hz) of a typical generated samples of #TV (left) and #sTV (right) sequences from a Generator trained after 12,255 steps.

gle logit. The initial GAN design as proposed by Goodfellow et al. (2014) trained the Discriminator network to distinguish real from generated data. Training such models, however, faced substantial challenges (Donahue et al., 2019). Donahue et al. (2019) implement the WGAN-GP strategy (Arjovsky et al., 2017; Gulrajani et al., 2017), which means that the Discriminator is trained "as a function that assists in computing the Wasserstein distance" (Donahue et al., 2019). The WaveGAN model (Donahue et al., 2019) uses ReLU activation in all but the last layer for the Generator network, and Leaky ReLU in all layers in the Discriminator network (as recommended for DCGAN in Radford et al. 2015). For exact dimensions of each layer and other details of the model, see Donahue et al. (2019).

2.2 Training data

The model was trained on the allophonic distribution of voiceless stops in English. Voiceless stops /p, t, k/ surface as aspirated [p^h, t^h, k^h] in English in word-initial position when immediately followed by a stressed vowel (Lisker, 1984; Iverson and Salmons, 1995; Vaux, 2002; Vaux and Samuels, 2005; Davis and Cho, 2006). If an alveolar sibilant [s] precedes the stop, however, the aspiration is blocked and the stop surfaces as unaspirated [p, t, k] (Lisker, 1984). A minimal pair illustrating this allophonic distribution is [ˈp^hɪt] 'pit' vs. [ˈspɪt] 'spit'. The most prominent phonetic correlate of this allophonic distribution is the difference in Voice Onset Time (VOT) duration (Abramson and Whalen, 2017) between the aspirated and unaspirated voiceless stops.

The model was trained on data from the TIMIT database (Garofolo et al., 1993).[1] The training data consist of 16-bit .wav files with 16 kHz sampling rate of word initial sequences of voiceless stops /p, t, k/ (= T) that were followed by a vowel (#TV) and word initial sequences of /s/ + /p, t, k/, followed by a vowel (#sTV). The training data includes 4,930 sequences with the structure #TV and 533 sequences with the structure #sTV (5,463 total). Both stressed and unstressed vowels are included in the training data, as this condition crucially complicates learning and makes the task for the neural network more challenging.

3 Experiment

3.1 Model: 12,255 steps

The Generator network after 12,255 steps (∼ 716 epochs) generates an acoustic signal that appears close to actual speech data. The number of training steps was chosen manually as a compromise between output interpretability and the number of epochs, where we try to approximately maximize the first and minimize the latter parameter. Figure 2 illustrates a typical generated sample of #TV (left) and #sTV (right) structures with a substantial difference in VOT durations.

To test whether the Generator learns the conditional distribution of VOT duration, the generated samples were annotated for VOT duration. VOT duration was measured from the release of closure to the onset of periodic vibration with clear formant structure. Altogether 96 generated samples were annotated; 62 in which no period of frication of [s] preceded and 34 in which [s] precedes the TV sequence. The generated data were fit to a linear model with only one predictor: presence of [s] (STRUCTURE). Place of articulation or following phonological learning, because the model is trained on a continuous speech stream and the generated sample fails to produce analyzable results for phonological purposes.

[1] Donahue et al. (2019) trained the model on the SC09 and TIMIT databases, but the results are not useful for model-

lowing vowel were not added in the model, because they are often difficult to recover. STRUCTURE is a significant predictor of VOT duration: $F(1) = 53.1, p < 0.0001$. The estimates for Intercept (duration of VOT when no [s] precedes) are $\beta = 56.2$ ms, $t = 25.74, p < 0.0001$. VOT is on average 26.8 ms shorter if [s] precedes the TV sequence and this difference is significant ($\beta = -26.8$ ms, $t = -7.29, p < 0.0001$).

While VOT duration is significantly shorter if [s] precedes the #TV sequence in the generated data, the model shows clear traces that the learning is incomplete and that the generator network fails to learn the distribution *categorically* at 12,255 steps. The three longest VOT durations in the #sTV condition in the generated data are 68.3 ms, 75.7 ms, and 76.2 ms. In all three cases the VOT is longer than the longest VOT duration of any #sTV sequence in the training data (longest is 65 ms). This generalization holds even in proportional terms (i.e. while controlling for "speech rate"): the generated data contains the highest ratio between the VOT duration and the frication duration of [s].

Longer VOT duration in the #sTV condition in the generated data compared to training data is not the only violation of the training data that the Generator outputs and that resembles linguistic behavior in humans. Occasionally, the Generator outputs a linguistically valid #sV sequence for which no evidence was available in the training data. The minimal duration of closure in #sTV sequences in the training data is 9.2 ms, the minimal duration of VOT is 9.4 ms. All sequences containing a [s] from the training data were manually inspected, and none of them contain a #sV sequence without a period of closure and VOT. Homorganic sequences of [s] followed by an alveolar stop [t] (#stV) are occasionally acoustically similar to the sequence without the stop (#sV) because frication noise from [s] carries onto the homorganic alveolar closure which can be very short. However, there is a clear fall and a second rise of noise amplitude after the release of the stop in #stV sequences. Figure 3 shows one case of the Generator network outputting a #sV sequence without any stop-like fall of the amplitude. In other words, the Generator network outputs a linguistically valid sequence #sV without any evidence for existence of this sequence in the training data. Similarly, the Generator occasionally outputs a sequence with two stops (two periods of aspiration noise with intervening short period of closure) and a vowel (#TTV) (Figure 3).

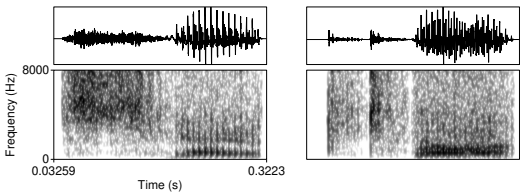

Figure 3: Waveforms and spectrograms (0–8000 Hz) of two innovative generated outputs of the shape #sV and #TTV. The sample on the left was generated after 16,715 steps.

Measuring overfitting is a substantial problem for Generative Adversarial Networks with no consensus on the most appropriate quantitative approach to the problem (Goodfellow et al., 2014; Radford et al., 2015). The danger with overfitting in a GAN architecture is that the Generator network would learn to fully replicate the input. Donahue et al. (2019) test overfitting on models trained with a substantially higher number of steps (200,000) compared to our model (12,255) and presents evidence that GAN models trained on audio data do not overfit even with substantially higher number of training steps. The best evidence against overfitting is precisely the fact that the Generator network outputs samples that substantially violate output distributions.

3.2 Establishing internal representations

Establishing internal representations of a neural network is a challenging task (Lillicrap and Kording, 2019). Below, we propose a technique for uncovering dependencies between the network's latent space and generated data based on logistic regression. This method has the potential to shed light on the network's internal representations: using the proposed technique, we can estimate how the network learns to map latent space into phonetically and phonologically meaningful units in the generated data.

To identify dependencies between the latent space and generated data, we correlate annotations of the output data with the variables in the latent space. As a starting point, we choose to identify correlates of the most prominent feature in the training data: presence or absence of [s]. Any number of other phonetic features can

be correlated with this approach; applying this technique to other features and other alternations should yield a better understanding of the network's learning mechanisms. Focusing on more than the chosen feature, however, is beyond the scope of this paper.

We propose a method based on logistic regression. First, 3,800 outputs from the Generator network trained after 12,255 steps were generated and manually annotated for presence or absence of [s]. 271 outputs (7.13%) were annotated as involving a segment [s]. Frication that resembled [s]-like aspiration noise after the alveolar stop and before high vowels was not annotated as including [s]. Innovative outputs such as an #[s] without the following vowel or #sV sequences were annotated as including an [s].

The annotated data together with values of latent variables for each generated sample (z) were fit to a logistic regression generalized additive model (using the *mgcv* package; Wood 2011 in R Core Team 2018) with the presence or absence of [s] as the dependent variable (binomial distribution of successes and failures) and smooth terms of latent variables (z) as predictors of interest (estimated as penalized thin plate regression splines; Wood 2011). Generalized additive models were chosen in order to avoid assumptions of linearity: it is possible that latent variables are not linearly correlated with features of interest in the output of the Generator network. The initial full model (FULL) includes smooths for all 100 variables in the latent space that are uniformly distributed within the interval $(-1,1)$ as predictors.

To reduce the number of variables, models with different shrinkage techniques are refit and compared: the latent variables for further analysis are then chosen based on combined results of different extratory models. We refit the model with various modifications: with modified smoothing penalty (MODIFIED); with original smoothing penalty, but with an additional penalty for each term if all smoothing parameters tend to infinity (SELECT; Wood 2011); and with manual removal of non-significant terms by Wald test for each term (EXCLUDED).

The estimated smooths appear mostly linear. We also fit the data to a linear logistic regression model (LINEAR) with all 100 predictors. To reduce the number of predictors, another model is fit (LINEAR EXCLUDED) with those predictors removed that do not improve fit.

To identify latent variables with highest correlation with [s] in the output, we extract estimates for each term from the generalized additive models and estimates of slopes from the linear model. Figure 4 plots those values. The plot points to a substantial difference between the highest seven predictors and the rest of the latent space. Seven latent variables are thus identified (z_5, z_{11}, z_{49}, z_{29}, z_{74}, z_{26}, z_{14}) as potentially having the largest effect on presence or absence of [s] in output. Lasso regression (Simon et al., 2011) and Random Forest models (Liaw and Wiener, 2002) give almost identical results.

To conduct an independent generative test of whether the chosen values correlate with [s] in the output data of the Generator network, we set values of the seven identified predictors (z_5, z_{11}, z_{49}, z_{29}, z_{74}, z_{26}, z_{14}) to the marginal value of 1 or -1 (depending on whether the correlation is positive or negative) and generated 100 outputs. Altogether seven values in the latent space were thus manipulated, which represents only 7% of the entire latent space. Of the 100 outputs with manipulated values, 73 outputs included a [s] or [s]-like element, either with the stop closure and vowel or without them. The rate of outputs that contain [s] is thus significantly higher when the seven values are manipulated to the marginal levels compared to randomly chosen latent space. In the output data without manipulated values, only 271 out of 3800 generated outputs (or 7.13%) contained an [s]. The difference is significant ($\chi^2(1) = 559.0, p < 0.00001$).

High proportions of [s] in the output can be achieved with manipulation of single latent variables, but the values need to be highly marginal, i.e. extend well beyond the training space. Setting the z_{11} value outside the training interval to -15, for example, causes the Generator to output [s] in 87 out of 100 generated (87%) sequences, which is again significantly more than with random input ($\chi^2(1) = 792.7, p < 0.0001$). When z_{11} is -25, the rate goes up to 96 out of 100, also significantly different from random inputs ($\chi^2(1) = 959.8, p < 0.0001$).

While there is a consistent drop in estimates of the regression models after the seven identified variables (Figure 4) and while several independent generation tests confirm that the seven variables correspond the to presence of [s] in the output,

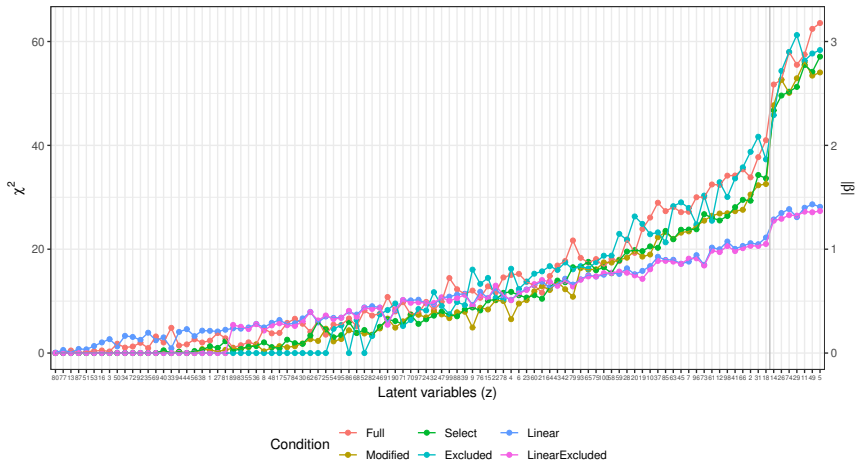

Figure 4: Plot of χ^2 values (left scale) for the 100 predictors across the four generalized additive models. For the two linear models (LINEAR and LINEAR EXCLUDED), estimates of slopes in absolute values ($|\beta|$) are plotted (right scale). The blue vertical line indicates the division between the seven chosen predictors and the rest of the predictor space with a clear drop in estimates between the first seven values (z_5, z_{11}, z_{49}, z_{29}, z_{74}, z_{26}, z_{14}) and the rest of the space.

the cutoff point between the seven variables and the rest of the latent space is still somewhat arbitrary. It is likely that other latent variables directly or indirectly influence the presence of [s] as well: the learning at this point is not yet categorical and several dependencies not discovered here likely affect the results. Nevertheless, further explorations of the latent space suggest the variables identified with the logistic regression (and other) models (Figure 4) are indeed the main variables involved with the presence or absence of [s] in the output.

3.3 Interpolation and phonetic features

We further explore whether the mapping between the uniformly distributed input (z) variables can be associated with specific phonetic or phonological features in that output. The crucial step in this direction is to explore values of the latent space beyond the training interval, i.e. beyond $(-1, 1)$. Crucially, we observe that the Generator network, while being trained on latent space limited to the interval $(-1, 1)$, learns representations that extend this interval. Even if the input latent variables (z) exceed the training interval, the Generator network outputs samples that closely resemble human speech. Furthermore, the dependencies learned during training extend outside of the $(-1, 1)$ interval. Exploring phonetic properties at these marginal values might reveal the actual underlying function of each latent variable.

To explore phonetic correlates of the seven latent variables, we set each of the seven variables separately to the marginal value -4.5 and interpolate to its opposite marginal value 4.5 in 0.5 increments, while keeping randomly-sampled values of the other 99 latent variables z constant. The ± 4.5 value was chosen based on manual inspection of generated samples: amplitude rises of [s] gradually weaken when variables have a value greater than ± 3.5. Seven sets of generated samples are thus created, one for each of the seven z values (with the other 99 z-values randomly sampled, but kept constant for all seven manipulated variables). Each set contains a subset of 19 generated outputs that correspond to the interpolated variables from -4.5 to 4.5 in 0.5 increments. Twenty-nine such sets containing an [s] in at least one set are extracted for analysis.

A clear pattern emerges in the generated data: the latent variables identified as corresponding to the presence of [s] via regression (Figure 4) have direct phonetic correlates and cause changes in amplitude and presence/absence of frication noise of [s] when each of the seven values in the latent space are manipulated to the chosen values, including values that exceed the training interval. In other words, by manipulating the identified latent variables, we control the presence/absence of [s] in the output as well as the amplitude of its frication noise.

Figure 5 illustrates this effect. Frication noise of

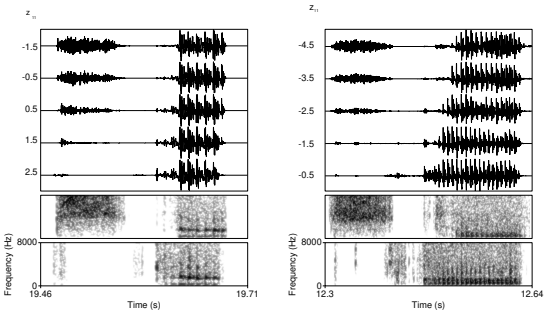

Figure 5: Waveforms and two spectrograms (both $0-8,000$ Hz) of generated data with z_{11} variable manipulated and interpolated. The values on the left of waveforms indicate the value of z_{11}. The two spectrograms represent the highest and the lowest value of z_{11}. A clear attenuation of the frication noise is visible until complete disappearance.

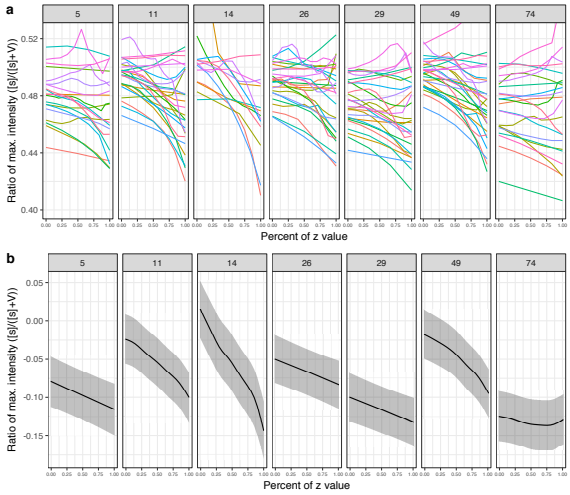

Figure 6: **(a)** Plots of ratios of maximum intensity between the frication of [s] and phonation of the vowel in #sTV sequences across the seven variables and **(b)** predicted values with 95% CIs of the ratio based on beta regression generalized additive model.

[s] gradually decreases by increasing the value of z_{11} until it completely disappears. The exact value of z_{11} for which the [s] disappears differs across examples and likely interacts with other features. It is possible that frication noise in the training has a higher amplitude in some conditions, which is why such cases require a higher magnitude of manipulation of z_{11}. The figure also shows that as the frication noise of [s] disappears, aspiration of a stop in what appears to be a #TV sequences starts surfacing and replaces the frication noise of [s]. Occasionally, frication noise of [s] gradually transforms into aspiration noise. The exact transformation is likely dependent on the 99 other z-variables held constant and their underlying phonetic effect. Regardless of the underlying phonetic effect of the other variables in the latent space, we can force [s] in the output when generating data and manipulating the chosen variables.

To test the significance of the effects of the seven identified features on the presence of [s] and the amplitude of its frication noise, the 29 generated sets of 19 outputs (with z-value from -4.5 to 4.5) for each of the seven variables were analyzed. The outputs were manually annotated for [s] and the following vowel. Outputs gradually change from #sTV to #TV. Only sequences containing an [s] were analyzed; as soon as [s] stops in the output, annotations were stopped and the outputs were not further analyzed. For each data point, maximum intensity of the fricative and the vowel was extracted in Praat (Boersma and Weenink, 2015; Lennes, 2003) with a 13.3 ms window length.

To test whether the decreased frication noise is not part of a general effect of decreased amplitude, we perform significance tests on the ratio of maximum intensity between the frication noise of [s] and the following vowel in the #sTV sequences. Figure 6 plots the ratio of maximum intensity of the fricative divided by the sum of two maximum intensities: of the fricative ([s]) and of the vowel (V). The manipulated z-values are additionally normalized to the interval [0,1], where 0 represents the most marginal value with [s] (usually ± 4.5; referred to as STRONG henceforth) and 1 represents the last value before [s] disappears (WEAK). Note that the point at which [s] is not present in the output anymore, but the vowel still surfaces (which would yield the ratio at 0) is not included in the model.

The data were fit to a beta regression generalized additive mixed model (Wood 2011) with random smooths for (i) trajectory and for (ii) value of other variables in the latent space of the Generator network, see Figure 6. All smooths (except for z_{74}) are significantly different from 0 and the plots show a clear negative trajectory.

The seven variables thus strongly correspond to the presence or absence of [s] in the output; by manipulating the chosen variables to the identified values we can attenuate frication noise of [s] and cause its presence or complete disappearance in the generated data. Again, the discovery of these

features is possible because we extend the initial training interval and test predictions on marginal values.

Interpolation of latent variables reveals that the presence of [s] is not controlled by a single latent variable, but by at least seven of them. The different latent variables that correspond to the presence of [s], however, are not phonetically vacuous: individually, they have distinct phonetic correspondences. The generated samples reveal that the variables' secondary effect (besides outputting [s] and controlling its intensity) is likely reflected in spectral properties of the frication noise. The seven variables are thus similar in the sense that manipulation of their values results in the presence of [s] by controlling its frication noise. They crucially differ, however, in the effects on the spectral properties of the outputs.

To test this prediction, spectral properties of the output fricatives are analyzed in the same 29 sets of generated samples. Spectral properties of the generated fricatives are generally not significantly different at the value of z right before [s] disappears from the outputs. As values of z increase toward the marginal levels (in most cases, ± 4.5), however, clear differentiation in spectral properties emerge between the seven z-variables. The trajectory for center of gravity, for example, significantly differs between z_{11} and most of the other six variables. Overall kurtosis is significantly different when z_{11} is manipulated, compared to, for example, z_{26} and z_{29}. Similarly, while z_{74} does not significantly attenuate amplitude of [s], it significantly differs in skew trajectory of [s]. The main function of z_{74} is thus likely in its control of spectral properties of frication of [s] (e.g. skew).

In sum, manipulating the latent variables that correspond to [s] in the output not only attenuates frication noise (when vocalic amplitude is controlled for) and causes [s] to surface or disappear from the output, but the different z-variables likely correspond to different phonetic features of the frication noise. By setting the values to the marginal levels well beyond the training interval, however, significant differences emerge both in overall levels as well as in trajectories of COG, kurtosis, and skew. It is thus likely that the variables collectively control the presence or absence of [s], but that individually, they control various phonetic features — spectral properties of the frication noise.

4 Conclusion

The results of this paper suggest that we can model phonology not only with rules (Chomsky and Halle, 1968), finite-state automata (Heinz, 2010; Chandlee, 2014), input-output optimization (Prince and Smolensky, 1993/2004), or with neural network architecture that already assumes some level of abstraction (see Section 1), but as the dependency between the latent space and generated data in Generative Adversarial Networks that are trained in an unsupervised manner from raw acoustic data. We train a Generative Adversarial Network (as implemented in Donahue et al. 2019 based on DCGAN architecture; Radford et al. 2015); the results of the computational experiment suggest that the network learns the conditional allophonic distribution of VOT duration. To the author's knowledge, this is the first paper testing learning of allophonic distributions in an unsupervised manner from raw acoustic data using neural networks. This paper also proposes a technique that identifies variables that correspond to the presence of [s] in the output and shows that by manipulating these values, we can generate data with or without [s] in the output as well as control its intensity and spectral properties of its frication noise. While at least seven latent variables control the presence of [s], each of them has a phonetic function that controls spectral properties of the frication noise. The proposed technique thus suggests that the Generator network learns to encode phonetic and phonological information in its latent space.

Training GAN networks on further processes and on languages other than English should yield more information about learning representations of phonetic and phonological processes. This paper outlines methodology for establishing internal representations and testing predictions against generated data, but represents just a first step in a broader task of establishing learning representation of phonetic and phonological data in a Generative Adversarial framework of phonology.

Acknowledgments

This research was funded by a grant to new faculty at the University of Washington. I would like to thank Sameer Arshad for slicing data from the TIMIT database and Heather Morrison for annotating data. All mistakes are my own.

References

Arthur S. Abramson and D.H. Whalen. 2017. Voice onset time (vot) at 50: Theoretical and practical issues in measuring voicing distinctions. *Journal of Phonetics*, 63:75 – 86.

John Alderete and Paul Tupper. 2018. Connectionist approaches to generative phonology. In Anna Bosch and S. J. Hannahs, editors, *The Routledge Handbook of Phonological Theory*, pages 360–390. Routledge, New York.

John Alderete, Paul Tupper, and Stefan A. Frisch. 2013. Phonological constraint induction in a connectionist network: learning ocp-place constraints from data. *Language Sciences*, 37:52 – 69.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223.

Enes Avcu, Chihiro Shibata, and Jeffrey Heinz. 2017. Subregular complexity and deep learning. In *Proceedings of the Conference on Logic and Machine Learning in Natural Language (LaML)*.

Paul Boersma and David Weenink. 2015. Praat: doing phonetics by computer [computer program]. version 5.4.06. Retrieved 21 February 2015 from http://www.praat.org/.

Z. S. Bond. 1981. A note concerning /s/ plus stop clusters in the speech of language-delayed children. *Applied Psycholinguistics*, 2(1):55–63.

Jane Chandlee. 2014. *Strictly local phonological processes*. Ph.D. thesis, University of Delaware.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row, New York.

Stuart Davis and Mi-Hui Cho. 2006. The distribution of aspirated stops and /h/ in American English and Korean: an alignment approach with typological implications. *Linguistic*, 41(4):607–652.

Chris Donahue, Julian McAuley, and Miller Puckette. 2019. Adversarial audio synthesis. In *ICLR*. `github.com/chrisdonahue/wavegan`.

Emmanuel Dupoux. 2018. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43 – 59.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, San Diego, California. Association for Computational Linguistics.

J. S. Garofolo, Lori Lamel, W M Fisher, Jonathan Fiscus, D S. Pallett, N L. Dahlgren, and V Zue. 1993. Timit acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Frank H Guenther. 2016. *Neural control of speech*. MIT Press.

Frank H. Guenther and Tony Vladusich. 2012. A neural theory of speech acquisition and production. *Journal of Neurolinguistics*, 25(5):408 – 422. Is a neural theory of language possible? Issues from an interdisciplinary perspective.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc.

Jeffrey Heinz. 2010. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.

Gregory K. Iverson and Joseph C. Salmons. 1995. Aspiration and laryngeal representation in germanic. *Phonology*, 12(3):369–396.

Herman Kamper, Micha Elsner, Aren Jansen, and Sharon Goldwater. 2015. Unsupervised neural network based feature extraction using weak top-down constraints. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5818–5822.

James Kirby and Morgan Sonderegger. 2015. Bias and population structure in the actuation of sound change. *arXiv e-prints*, page arXiv:1507.04420.

Chia-ying Lee and James Glass. 2012. A nonparametric Bayesian approach to acoustic model discovery. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 40–49, Jeju Island, Korea. Association for Computational Linguistics.

Mietta Lennes. 2003. f0-f1-f2-intensity_praat_script. praat script. Modified by Dan McCloy, Esther Le Grésauze, and Gašper Beguš.

Andy Liaw and Matthew Wiener. 2002. Classification and regression by randomforest. *R News*, 2(3):18–22.

Timothy P. Lillicrap and Konrad P. Kording. 2019. What does it mean to understand a neural network? *arXiv e-prints*, page arXiv:1907.06374.

Leigh Lisker. 1984. How is the aspiration of english /p, t, k/ "predictable"? *Language and Speech*, 27(4):391–394.

Abhijit Mahalunkar and John D. Kelleher. 2018. Using regular languages to explore the representational capacity of recurrent neural architectures. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 189–198, Cham. Springer International Publishing.

Andrew Martin, Sharon Peperkamp, and Emmanuel Dupoux. 2013. Learning phonemes with a proto-lexicon. *Cognitive Science*, 37(1):103–124.

S McLeod, J van Doorn, and V Reed. 1996. Homonyms and cluster reduction in the normal development of children's speech. In *Proceedings of the Sixth Australian International Conference on Speech Science & Technology*, pages 331–336.

Noël Nguyen and Véronique Delvaux. 2015. Role of imitation in the emergence of phonological systems. *Journal of Phonetics*, 53:46 – 54. On the cognitive nature of speech sound systems.

Pierre-Yves Oudeyer. 2001. Coupled neural maps for the origins of vowel systems. In *Proceedings of the International conference on artificial neural networks. Lecture notes in computer science*, pages 1171–1176. Springer. Volume: 2130.

Pierre-Yves Oudeyer. 2002. Phonemic coding might result from sensory-motor coupling dynamics. In *From animals to animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, pages 406–416. MIT Press.

Pierre-Yves Oudeyer. 2005. The self-organization of speech sounds. *Journal of Theoretical Biology*, 233(3):435 – 449.

Pierre-Yves Oudeyer. 2006. *Self-organization in the evolution of speech*. Studies in the evolution of language ; 6. Oxford University Press, Oxford.

Joe Pater. 2019. Generative linguistics and neural networks at 60: Foundation, friction, and fusion. *Language*.

Janet Pierrehumbert. 2001. Exemplar dynamics: Word frequency, lenition, and contrast. In Joan L. Bybee and Paul J. Hopper, editors, *Frequency Effects and the Emergence of Lexical Structure*, pages 137–157. John Benjamins, Amsterdam.

Brandon Prickett, Aaron Traylor, and Joe Pater. 2019. Learning reduplication with a variable-free neural network. Ms., University of Massachusetts, Amherst. http://works.bepress.com/joe_pater/38/ (accessed 23 May 2019).

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell, Malden, MA. First published in Tech. Rep. 2, Rutgers University Center for Cognitive Science.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Jonathan Rawski and Jeffrey Heinz. 2019. No free lunch in linguistics or machine learning: Response to pater. *Language*.

Thomas Schatz, Naomi Feldman, Sharon Goldwater, Xuan Nga Cao, and Emmanuel Dupoux. 2019. Early phonetic learning without phonetic categories – insights from machine learning.

Cory Shain and Micha Elsner. 2019. Measuring the perceptual availability of phonological features during language acquisition using unsupervised binary stochastic autoencoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 69–85, Minneapolis, Minnesota. Association for Computational Linguistics.

Miikka P. Silfverberg, Lingshuang Mao, and Mans Hulden. 2018. Sound analogies with phoneme embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 136–144.

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. 2011. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13.

Roland Thiollière, Ewan Dunbar, Gabriel Synnaeve, Maarten Versteegh, and Emmanuel Dupoux. 2015. A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling. In *Proceedings of Interspeech*.

Bert Vaux. 2002. Aspiration in English. Ms., Harvard University. Accessed on June 27, 2019.

Bert Vaux and Bridget Samuels. 2005. Laryngeal markedness and aspiration. *Phonology*, 22(3):395–436.

Noah Weber, Leena Shekhar, and Niranjan Balasubramanian. 2018. The fine line between linguistic generalization and failure in Seq2Seq-attention models. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 24–27, New Orleans, Louisiana. Association for Computational Linguistics.

S. N. Wood. 2011. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1):3–36.